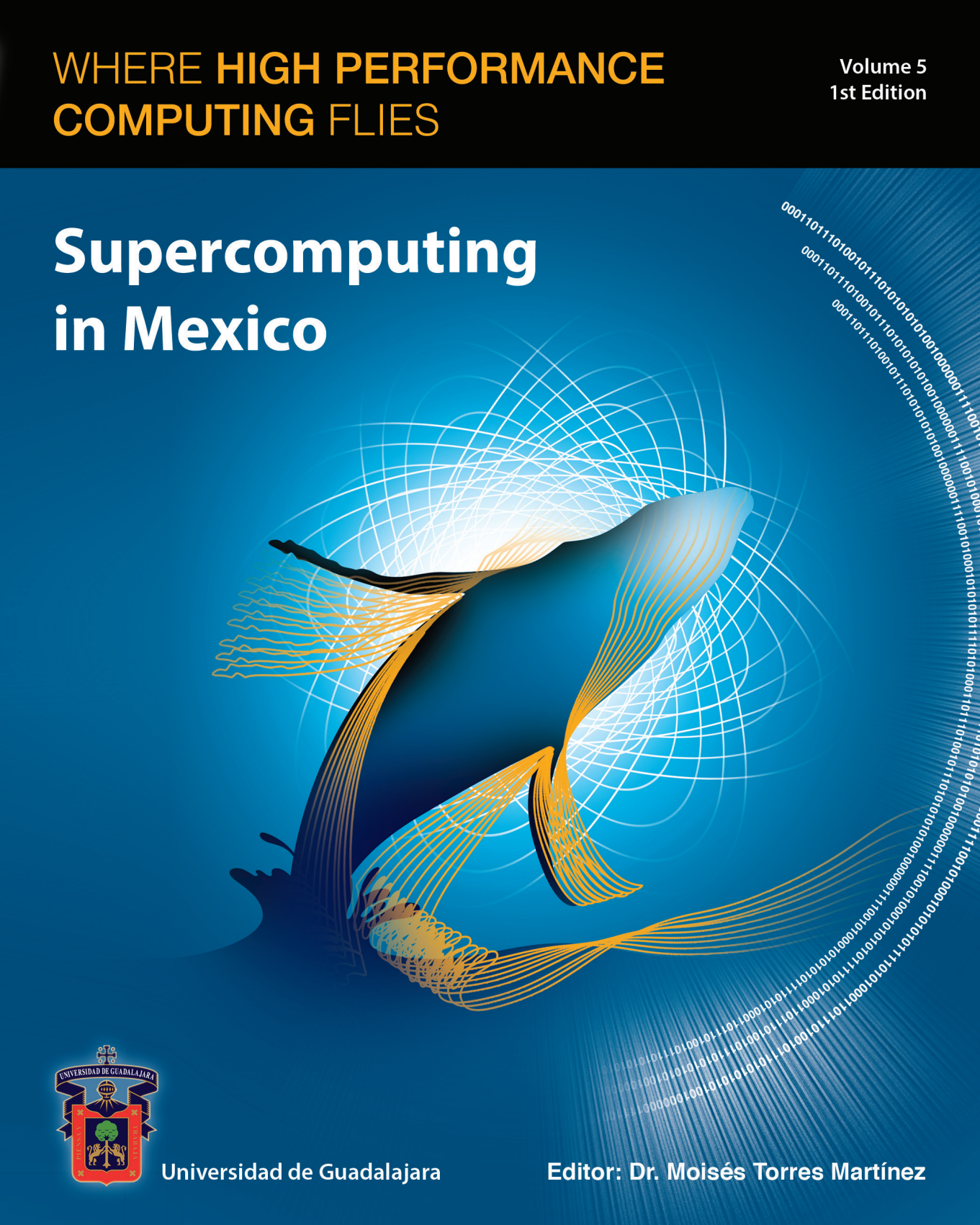
WHERE **HIGH PERFORMANCE COMPUTING** FLIES

Volume 5
1st Edition

Supercomputing in Mexico



Universidad de Guadalajara

Editor: Dr. Moisés Torres Martínez

Supercomputing in Mexico

ISUM

WHERE HIGH PERFORMANCE COMPUTING FLIES

ISUM 2014 Conference Proceedings

Editorial Board

ISUM 2014 Conference proceedings are published by the *Coordinación General de Tecnologías de Información*, Volume 5, 1st edition, March 26, 2015. Printed in Guadalajara, Jalisco México, March 26, 2015 in los Talleres Gráficos de Transición.

Supercomputing in México

WHERE HIGH PERFORMANCE COMPUTING FLIES

Volume editor:

Dr. Moisés Torres Martínez

University of Guadalajara
General Coordination of Information Technologies
México 2015

ISBN: 978-607-742-205-1

Ave. Juárez No. 976, Piso 2
Col. Centro, C.P. 44100
Guadalajara, Jal., México

Volume V:
1st Edition

Obra Completa: 978-607-450-347-0

Printed and Edited in México

Contents

Cloud Computing

Grids and GPU´s

Infrastructure

Parallel Computing

Scheduling

Appendix I

Foreword

Supercomputers play an important role in computational science, and are used for a wide range of computationally intensive tasks in graphical visualizations, climate research, weather forecasting, molecular modeling, and physical simulations just to mention a few uses of the myriad of possibilities.

We know that Supercomputing is a key enabler for most recent discoveries in science and technology, providing researchers computational power for analyzing really large datasets in a timely manner. It became crucial for the recent discovered Higgs boson particle in CERN. As well as, the recent climate change initiatives outlined by the US President Barack Obama that requires large-scale analysis of climate data. Now, supercomputers are considered the modern science labs.

There are still many computational and technical challenges ahead that must be addressed and we need supercomputers to help researchers obtain faster results. Whether it is analyzing a mammogram to determine if a woman has cancer or not, the time needed to analyze the problem can mean the difference between life and death. For iterative processes you may find that your competitor, who is using a faster computer, comes up with a better answer or a better product faster than you do. These are only a few examples of the importance of supercomputers in our everyday lives. But, as we continue to produce large digital sets of data, these systems become more important in our society to be able to analyze these big data at faster speeds to obtain new knowledge.

Thus, the knowledge obtain from these powerful computers is not useful unles it is shared with the scientific community. The 5th International Supercomputing Conference in México (ISUM) hosted in Ensenada, Baja California, México, is a space that allows researchers to share their work in the uses of supercomputing. It is also an open forum to exchange ideas, to discuss and provide feedback on the work of research and industry centers, for networking and consolidate collaborative networks of researchers, as well as industry and academia. Works presented in this book addresses research from the analyisis of deterministic timetabling strategies for the generation of university timetables, a tcp/ip replication with a fault tolerance scheme for high availability, and to the parallel metaheuristics.

In addition to the works presented in ISUM 2014, the conference also provided a space for dialogue among the academic community and society in general, about computing capabilities available in the country, while emphasizing its importance in the lives of people.

For example, at ISUM 2014, Cinvestav, under the ABACUS project, announced the acquisition of the most powerful computer in Mexico. This is great news for the country´s advancement in research and development and certainly great news for the supercomputing community. This news among the many presentations by the keynote speakers like Dr. Uwe Schwiegelshohn from TU Dortmund University from Germany enriched the conference sharing his knowledge on how to design a resource management method for large data centers. As Dr. Schwiegelshohn enriched the conference with his keynote, the many presentations created an atmosphere of knowledge throughout the four days of the conference. It was breathtaking to see the ambiance created by all the participants and presenters. Simply incredible!

The introduction of this book, Dr. Torres Martinez addresses the growth of the the supercomputing community in the country, and the establishment of the new RedMexSu (Red Mexicana de Supercomputo)—Mexican Supercomputing Network, which is now an official organization under the auspices of CONACYT (Nationa Council for Science and Technology-*Consejo Nacional de Ciencia y Tecnología*). He also provides a brief description of the publishable works of this 5th edition of the ISUM book.

All this work is certainly the result of the national ISUM conference committee leadership´s success in convening some of our best researchers, undergraduate, and graduate students from across the country to present their work and publish it in this book. My congratulations to the Centro de Investigación Científica y de Educación Superior en Ensenada (CICESE) for an excellent job in hosting the 5th edition of the International Supercomputing Conference in Méxco (ISUM 2014). Kudos to all those who contributed to the success of this conference because without the commitment from all the individuals and academic institutions and industry sponsors that participated, this event would not have had the success it did, and these works are the real testament of that success.

The University of Guadalajara is most proud to have been a contributor to the success of this event through the vast participation of its researchers, graduate and undergraduate students who presented their research work. It is also proud to see the broad participation of researchers from and throughout the country who are contributing significantly to research and development via their work and uses of supercomputers to achieve their research results. It is gratifying to see that ISUM is providing a space for researchers in supercomputing to share their important work with the national and international scientific community. It is also great to see that it is making a real impact in fostering the uses of supercomputing in México and Latin America.

On behalf of the University of Guadalajara and the General Coordination Office in Information Technology, it is my pleasure to congratulate all the contributing authors to this publication for their work and commitment to advance research in science and technology. Lastly, I cordially invite you to read through the articles of your interest and to participate in upcoming ISUM conferences.

Dr. Luis Alberto Gutiérrez Díaz de León
University of Guadalajara
General Coordinator,
General Coordination Office in Information Tecnologies (CGTI)

Preface

Supercomputing in México continues to have a steady growth in its uses in research and development. We are now seeing this growth in institutions of higher learning and national laboratories who are valuing the power of these machines to advance science and technology in the country. For example, CINVESTAV recently announced the most powerful computer in Mexico under the ABACUS project with 8,904 nodes-Intel® Xeon® E5-2697v3 (Haswell) processors, 100 GPU´s Nvidia K40, 400 Tflops linpack Rmax and with 1.2 petabytes of disc space and 40.7 Terabytes of RAM. This system is being long awaited by CINVESTAV. The state of Mexico and the federal government have invested significantly to establish the ABACUS center to address the many needs in solving some of the most complex research problems in academia and industry. Similarly, the Benemérita Universidad Autónoma de Puebla (BUAP) is also in the planning to create a supercomputing center with similar capabilities as the CINVESTAV-ABACUS project. The BUAP promises to be of high caliber and will address some of the most complex problems needing the power of a supercomputer. The country ten years ago was not investing enough to acquire these systems to accelerate research and development as it is doing today. This wave of new centers emerging around the country is certainly a positive step towards the advancement of science and technology in México; however we hold a huge responsibility as researchers to capitalize on the capacity of these machines to resolve some of the most complex problems in academia, industry and society in general.

Thus, the 5th International Supercomputing Conference in México (ISUM 2014) has served as a space for researchers from Mexico, Latin America, United States, Asia and Europe to share their work related to supercomputing whether it is in architectures, parallel computing, scientific visualization, Grids or applications. This edition of the ISUM is certainly a stepping stone to promoting the uses of these powerful machines in research, to make long lasting partnerships between academic institutions and industry, and collaborations among researchers, undergraduates and graduate students. It is refreshing to see that these past five years of ISUM many of these connections remain strong and continue to mature in the work that evolves from these partnerships and collaborations. As we see new computer centers emerging around the country, we also see the significant growth of researchers interested in participating in the ISUM to share their work and to make those important connections with colleagues and the industry sector.

This book is a collection of works that were presented in the ISUM 2014 hosted by the Centro de Investigación Científica y de Educación Superior en Ensenada (CICESE) in Ensenada, Mexico. The academic review committee received more than 40 articles to review. It was not an easy task for the review committee to select the works to be published in this book, although they felt that many of the works were worthy of publishing, the selected ones demonstrated a high quality and relevance to supercomputing. So, you will find among these works themes in cloud computing, scheduling, Grids/GPU´s, parallel computing, infrastructures and applications. You will also find abstracts presented in ISUM 2014 from the keynote speakers from academia, industry and individual presentations by researchers from and throughout the globe. It is with great pleasure that I congratulate all the presenters who participated in this 5th edition of ISUM presenting their work. It is certainly these works what set the stage in ISUM in creating a constructive dialogue of a new body of research work reinforcing and challenging the current knowledge. This interchange of ideas is what made this event such a success after the four days of presentations and the

more than 450 attendees. Congratulations to CICESE for a work well done in coordinating this event and to the ISUM national committee for all their contributions that made it a success.

This event is certainly complementary to the growth of supercomputing centers that are emerging around the country. As we continue to evolve in hosting the ISUM around the country, we do hope that these new and existing supercomputing centers find a space where they can build long-lasting collaborations, share their important work and promote the innovations that rise from each center.

On behalf of the ISUM national committee, I invite you to read through this book and find the article of your interest and do hope you find it of importance to move forward on your own particular work. You are also invited to join us at the ISUM 2015 in México City and share your own particular work, it would be our pleasure to have the opportunity to see your research work published in the next edition of the ISUM book: Supercomputing in México.

Dr. Moises Torres Martinez
ISUM National Committee, Chair

Acknowledgements

This publication could not be possible without the contributions of participants representing institutions from México, Latin America, Asia, United States, and European Union whom participated in this "5th International Supercomputing Conference in México 2014" with presentations and paper submittals. It is a great honor to have had the participation of the many authors who contributed to this publication and conference attendees for their participation, presentations, questions, and interaction making this conference a success.

In addition, this conference was also possible due to the many important contributions from the following people who made this event a success. We gratefully thank everyone for their individual contribution.

Universidad de Guadalajara

Mtro. Itzcóatl Tonatiuh Bravo Padilla,	Rector General
Dr. Miguel Ángel Navarro Navarro,	Vicerrector Ejecutivo
Lic. José Alfredo Peña Ramos,	Secretario General
Mtra. Carmen Enedina Rodríguez Armenta,	Coordinadora, Coordinación Administrativa
Dr. Luis Alberto Gutierrez Diaz de Leon,	Coordinador, Coordinación General de Tecnologías de Información

Executive Committee in Ensenada

Dr. Federico Graef Ziehl	Director General, Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)
Dr. Oscar Lopez Bonilla	Vicerrector, Universidad Autónoma de Baja California Campus Ensenada (UABC)
Mtro. Carlos Garcia Alvarado	Director, Cetys Campus Ensenada
Jose Lozano Rizk	Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)
Salvador Castañeda	Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)
Andrei Tchernykh	Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE)

Special recognition to the ISUM National Committee 2014 because without their participation and dedication in the organization of this event, the ISUM 2014 could not have been possible.

Andrei Tchernykh	(CICESE)
Afredo Santillan	(UNAM)
Axel Aviles Pallares	(UAM)
César Carlos Diaz Torrejón	(IPICyT-CNS)
Cynthia Lynnette Lezama Canizales	(IPICyT-CNS)
Jaime Klapp Escribano	(ININ)
Raul G. Hazas Izquierdo	(UNISON)
René Luna García	(IPN)
José Lozano	(CICESE)
Juan Carlos Chimal Enguía	(IPN)
Juan Carlos Rosas Cabrera	(UAM)
Juan Manuel Ramirez Alcaraz	(UCOLIMA)
Jesús Cruz	(UNAM)
Lizbeth Heras Lara	(UNAM)
Mauricio Carrillo Tripp	(CINVESTAV)
Manuel Aguilar Cornejo	(UAM)
Ma. Del Carmen Heras Sanchez	(UNISON)
Moisés Torres Martínez	(UdeG)
Salma Jalife	(CUDI)
Salvador Botello Rionda	(CIMAT)
Salvador Castañeda	(CICESE)
Verónica Lizette Dueñas	(UdeG)

Introduction

A Perspective on the Self Sustainability Model of Supercomputer Centers and Review of Research Works in Supercomputing in Mexico

by
Dr. Moises Torres Martinez
Centro Nacional de Supercomputo (CNS-IPICYT)
moises_torres10@hotmail.com

Abstract

This paper provides a brief perspective on the new model of self-sustainability being implemented in new supercomputing centers emerging around the country. It gives a view on the importance of collaboration among these centers and how they need to be careful in implementing these models in order to sustain a balance between research and services provided. It also introduces the importance of the ***Red Mexicana de Supercomputo*** *(RedMexSu) to complement the work done in these centers and promote supercomputing in the country. Finally, it gives an overview of the works presented in this book.*

A Perspective in a Self-Sustainable Model in Supercomputers Centers

The country is finally beginning to make significant investments in acquiring supercomputers to move along research and development. The last time it made these significant investments and encouraged academic institutions to acquire this power in computing was in 1991 when the first systems were acquired in the country. Since then, supercomputing in Mexico grew very slowly and the support from the Consejo Nacional en Ciencia y Tecnología (CONACYT)- National Council in Science and Technology was sporadic and did not have a significant impact nationally. Although in these twenty four years centers like the National Center for Supercomputing (CNS-IPICYT), Supercomputing Laboratory and Parallel Visualization from the Universidad Autónoma Metropolitana (UAM) and others were established, these few centers did not meet the needs of the growing number of researchers needing supercomputers to advance their research. However, what some of these centers, like CNS-IPICYT, did is established a new paradigm in Mexico that focused not only on the needs of academia but also on the growing needs in industry and government in solving problems needing the processing power of these machines. At the same time, established a business plan that focused on self-sustainability to ensure that the center was going to last and be on the cutting edge, since the funds received from the federal government were not enough to keep up this center, this model was established.

Interestingly enough, CNS-IPICYT has become a model to follow for the success it has had in its self-sustainability. The fact that this model is being replicated in new rising centers and encouraged by CONACYT, we must keep in mind that the federal government must continue to support these centers despite of their success, because they continue to be centers that serve academia and who´s interest is to advance research and development in the country.

As new centers around the country begin to rise, like the CINVESTAV-ABACUS project and the BUAP Supercomputer Center and implement a self-sustainability model, it is important to realize that it takes time for this model to evolve and it takes a unique team of people who need to focus in bringing in contracts from the industry and government sectors. This team of people will not necessarily be researchers or technologist as these centers are accustomed to have in their staff. Although, they remain as the fundamental staff necessary to solve many of the problems from academia, industry and government, they cannot be the ones seeking new contracts for the center. Thus, these centers need to bring in a new set of expertise beginning from its leadership, sales persons, project managers, marketing experts etc. The mind set of this new set of collaborators is to attract contracts that will eventually help reach their goals in becoming a self-sustainable center. This process takes time and a new thinking in the way how these centers must be led, because the new thinking has to be to solve problems immediately for these customers and keep them satisfied. At the same time, the centers must maintain a focus on the academic side in advancing the works and meeting the needs of researchers from their institution and from around the country.

These new centers are not going to be an easy task to manage and will need time to evolve and find their own niche in meeting the needs of industry and government sectors, while maintaining a strong hold on the academic research conducted. Thus, the federal government must maintain a strong funding support to ensure these centers evolve and are successful as they implement a self-sustainable model. If there is no strong support from the federal government and the individual institutions for the implementation of this new model these new centers and future ones will be difficult to achieve the success that CNS-IPICYT has had the past few years; Especially that there will be greater competition among them for contracts with the country´s primary industry and needless to say government.

Thus, it is important for the centers implementing a self-sustainable model to work together and establish collaborations among each other to capitalize on the strength of each to be able to compete for contracts that may be beneficial to all and minimize any fierce competition among each other. It is not to say that there will not be competition, but what is important is that there exists a high level of collaboration to ensure that the human and technical resources are abundant to meet the needs of industry and government and ensure a trust is built among these sectors. Otherwise there exists the risk for these centers to offer more than what they are equipped to do and break the trust among the industry and government sectors that may harm everyone involved in offering solutions that require high processing power and a unique expertise.

Building a strong supercomputing community is a key element for existing and new centers to achieve their success. We all know that working in silos is not the best approach to be successful in what one does, especially when it involves organizations that require a high and unique expertise to function. More than ever, the supercomputing community in México must work together to continue to make a significant impact in research and development in the country and ensure all supercomputer centers are successful and on the cutting edge.

One approach to be able to strengthen this community is the new network that the National Council for Science and Technology (CONACYT) funded to establish the Mexican Supercomputing Network (*Red Mexicana de Supercomputo* (RedMexSU)). This network has as the objective to support an e-infraestructure, advance connectivity, services, high level training in these areas, and a strong collaboration among institutions of higher learning in the country. For the first time CONACYT recognized (with fund support) that there is a growing need of researchers, technologists and engineers involved in supercomputing. And it approved the RedMexSu to ensure academic institutions, industry, and government work together to build capacity in the country to meet the needs of this growing research community in México. Since the RedMexSu is in its infancy stages, it is of greater importance that the supercomputing community in México joins this group to ensure that its activities evolve and meet their needs. Also, to make certain it brings to the table the researchers, technologists and engineers who need to be there to help move the group along in its deeds.

This is a significant opportunity to be able to work together and define the state of the art in supercomputing in the country where the infrastructure, applications, training, and the strategic need for supercomputing centers are researched and defined. As the activities of this group are well-defined we will see greater growth in the uses of this powerful machines to advance the research conducted in the country and eventually evolve to be a greater player in supercomputing internationally. Most importantly is that the country is equipped with the sufficient computing power and trained personnel to solve some of their most impacting problems whether it is in health, climate, and/or economics as an example. The RedMexSu will be a space for the supercomputing community to come together and define the direction in which we want this area to grow to address some of the needs of the country in research and development. The success of this network can only be accomplished by an active participation of this community and work towards a common goal which is to advance supercomputing in research and development in México. And as the new supercomputer centers continue to rise around the country as they implement a self-sustainable model, the RedMexSu will be of greater importance to help in their success.

A Review of Research Works Presented in this Book

As important as the supercomputer centers are to the country, their importance is defined by the works that these centers produce. In this book we show case fourteen articles that were selected by the ISUM academic committee out of 40 that were reviewed at the 5th edition of the International Supercomputing Conference in México (ISUM 2014). Although the academic committee felt that most of the works deserved to be published, they only selected a few that were most relevant to supercomputing. The works that this book presents include the following themes, applications, cloud computing, Grids/GPU´s, infrastructure, parallel computing, and scheduling.

Applications

In the article *"HPC applied to Fluorescence Fluctuation Analysis: Contributing to Unravel Hidden Dynamical Processes"* the authors' share the application of scientific high performance computing

techniques to complement the fluorescence fluctuation analysis using stochastic simulation. They found that the execution time analysis demonstrated that using the approach proposed, the parallel algorithm was able to compute 24000 iterations and generate 4000 times more data than the sequential version, while demanding the same time than the sequential algorithm requires to compute 16000 iterations.In another article *"Application of an Adaptive Inversion Frequencies Algorithm for Router Bandwidth Improvement"* Kravtsunov et al., study the practical application of the inversion frequencies algorithm for the compression of IP network data. They present a modified version of the algorithm with adaptability. And show that their algorithm can be effectively used for DNS/UDP traffic compression, allowing a 3-15% increase in router bandwidth.

Cloud Computing

In this theme of cloud computing there are two articles that were selected, in the first article *"Cost Optimization of Virtual Machine Provisioning in Federated IaaS Clouds"*, Armenta Cano, Tchnerykh, Yahyapour and Nabrzyski discuss cost optimization model in cloud computing, and formulate the cost-aware resource allocation problem that provides cost-efficiency in the context of the cloud federation. Their model assumes a cloud provider with multiple heterogeneous resources or data centers. The provider needs to control amount of resources to avoid overprovisioning and increasing capital costs. To reduce an importance of known Build-To-Peak approach that means building infrastructures for top demands with over-provisioning in total operating time, cloud provider has to collaborate with other providers to be able to fulfil requests during peak demands by using idle resources of other peers. They conclude by showing how none of these works directly addresses the problem space of the considered problem, but do provide a valuable basis for their work

The article *"Model of Video on Demand Service Provisioning on Multiple Third Party Cloud Storage Services"* presented by the authors Barba Jimenez, Ramirez Velarde, and Tchnerykh share a solution model to tackle the problem of providing Video on-Demand (VoD) using cloud computing storage service composition. They also present related works, the problem motivation and some preliminary results. As part of the problem, they study the performance and scalability model for this VoD cloud service by performing a statistical analysis and Principal Component Analysis (PCA) on real cloud data.

Grids and GPU´s

Under the Grids and GPU´s section there´s five articles that share interesting findings. In the article *"An Optimization Method for Lennard-Jones Clusters using Conjugate Gradient and Molecular Dynamic Simulations"*, the authors Alonzo Velázquez, Valdéz Peña, and Botello Rionda discuss the trends of current computational systems that have been created based in grids methods, which avoid computing all pairs

of distances in problem related with short range potentials speeding up Molecular Dynamic simulations, but with a little more cost in memory. In this work they implemented an optimization algorithm based in Conjugate Gradient (CG) and Molecular Dynamic (MD) Simulations. The results provide information to believe they had a good optimization method in the problem of optimization of Lennard-Jones clusters.

In the article *"GPU's Molecular Dynamics Simulation of a One Million particles"* the authors Zamora, Cruz-Santiago et al., describe the main algorithms involved in a molecular dynamics simulation and present a strategy of parallelization using CUDA to accelerate the computations in GPUs. They show several application examples of their implementations for 500, 2048, 10000 and 106 particles. They also have very good findings in terms of computation time and accuracy of the results.

An interesting article entitled *"Semi-automatic Historical Climate Data Recovering Using a Distributed Volunteer Grid Infrastructure"* by Nesmachnow and Da Silva. They present the Digi-Clima project, whose main objective is to design a semi-automatic tool for digitalizing and recovering historical climate data using distributed computing techniques on grid and cloud infrastructures. The deploy of the processing tool for historical climate records on the Ourgrid middleware for grid and cloud is described, and the experimental evaluation using a volunteer-based Ourgrid infrastructure is reported. Accurate speedup values are reported for the distributed application.

The authors Hazas Izquierdo, Rodriguez Navarro, Faz Gutierrez and Salazar Orozco in their work entitled *"Straightforward DSP Algorithm Suitable for GPU Computation"* describe that a Current Graphic Processing Units (GPUs) have achieved a deep level of programming parallelism. Filtering of discrete sequences often requires an inordinate amount of computing resources. Infinite Impulse Response (IIR) filter structures often are broken down into simpler and concatenated elements called biquads for ease of implementation. Analysis of data flow across one such simplified structure, prompts of feasible digital signal processing (DSP) algorithm simplification. Favorable comparison of outcome brought up by forthright C-based implementation of prospective DSP algorithm versus industry standard application suggests a filtering method likely adaptable for CUDA coding and execution.

The article *"Performance Evaluation of Cellular Genetic Algorithms on GPU"* by López-Juárez, Barrón Fernández, and Godoy-Calderon evaluates the performance of two models of cellular genetic algorithms, which find the optimal number of partitions for a data set, using a cluster validation index as objective function. One of their findings is the GPU mainly improves speedup and serial fraction because of shared memory is able to increase the number of blocks without sacrificing speedup.

Infrastructure

The theme in infrastructure has only one article that is very distinct and provides interesting findings in the research conducted. The authors Medina, Tchernykh and Ramos Paz in their article entitled *"A TCP/IP Replication with a Fault Tolerance Scheme for High Availability"* propose a TCP/IP Replication scheme for a fault tolerance system to provide high availability. And Quantitative results of protocol behavior in real scenario are analyzed with special attention to network performance.

Parallel Computing

This theme usually presents interesting articles and this is not the exception, there´s three articles selected by the committee that do just that. The article *"Parallelization of filter BSS/WSS on GPGPU for classifying cancer subtypes with SVM*" the authors Castro Liera, Luna Taylor, Merecías Pérez, and Meléndrez Carballo propose a parallel version of BSS/WSS (Between Sum Square-BSS, Within Sum Square-WSS) filter using GPGPU (General Purpose Graphics Processing Units). The application processes genetic expression data to select statistically relevant genes. A SVM (Support Vector Machine) is applied on the expression data of genes selected by the filter, to diagnose cancer. With the use of this combination of algorithms, a success rate of 92% in the diagnosis was achieved. With the parallel implementation on GPGPU with CUDA technology, a dramatic reduction of execution time of approximately 18 times compared to the sequential implementation on CPU was achieved.

In the article *"Application of Parallel Processing Based on Multithreading to the Optimal Trajectory Planning for Robot Manipulators"*, Pérez Bailón and Ramos-Paz present a novel algorithm that uses eighth-degree polynomial functions to generate smooth trajectories for the parametric representation of a given path. In this contribution it is introduced the application of parallel processing based on multithreading to the optimal trajectory planning for robot manipulator in order to improve the execution time of the optimization algorithm proposed. The algorithms presented minimize the mechanical energy consumed by the actuators of the robot manipulator (RM) or the path traveling time of the RM. To solve the optimization model of mechanical energy consumed, a genetic algorithm is implemented, and to solve the optimization model of path traveling time, a method based on a combination of a genetic algorithm and the numerical algorithm known as bisection method has been implemented.

In addition, the article *"Simulating a Hyperbolic-PDE using a 2-dimensional CA"*, Huerta-Trujillo and Chimal Eguía present a parallel model based on 2-dimensional Cellular Automaton (CA); likewise the process for obtaining the evolution rule. The model obtained is compared with the analytical solution of a partial differential equation of the two-dimensional linear and homogeneous hyperbolic type. This models a vibrating membrane with specific initial and border conditions. The frequency spectrum and the error between the data obtained from the CA model are analyzed versus the data provided by the solution evaluation to the differential equation. On the other hand, CA complexity is analyzed and the implementation using parallel computing to reduce the computational complexity is guaranteed.

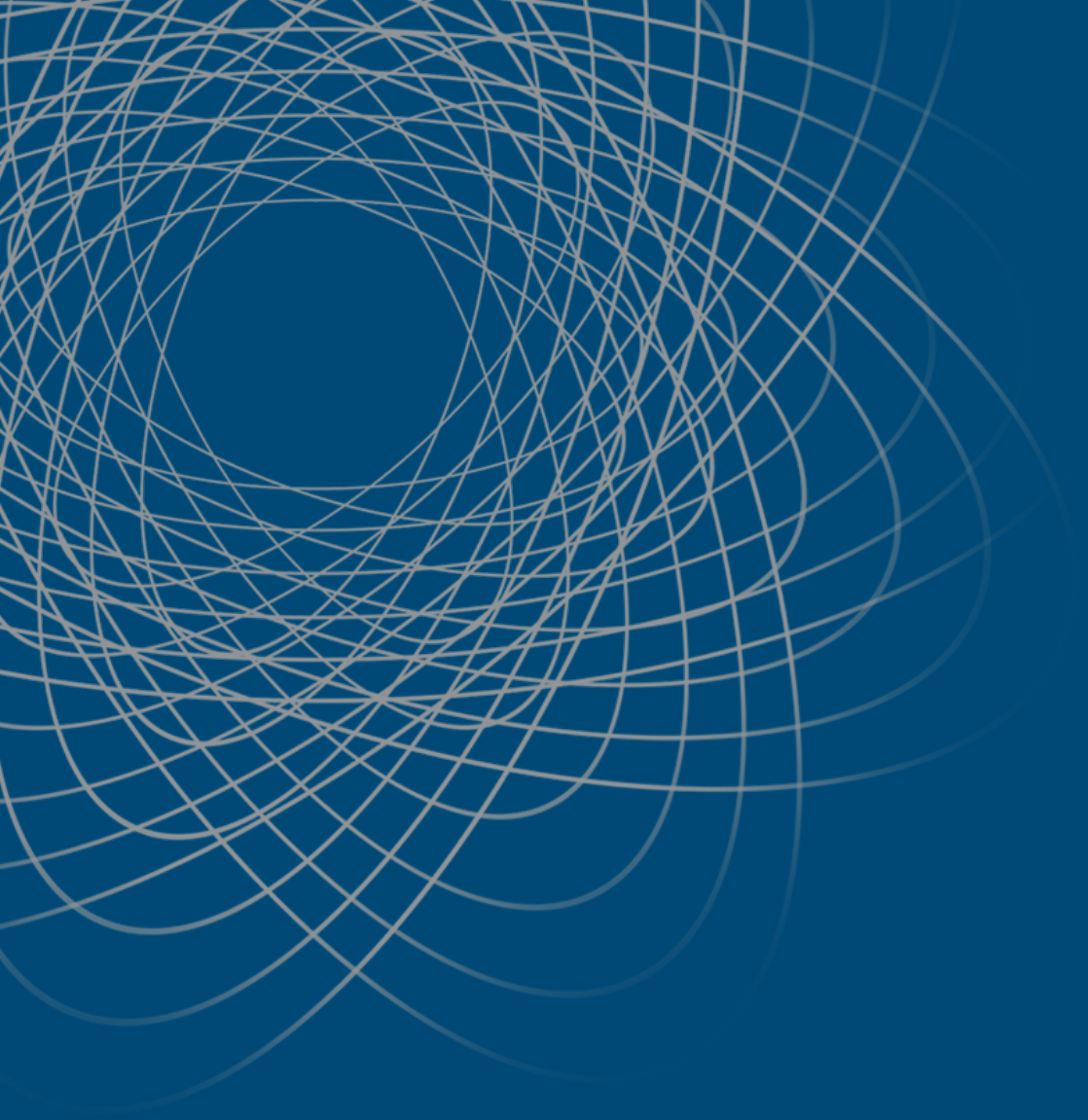
Scheduling

Last but not least in this section you´ll find a fascinated article by Peña-Luna, Hirales Carbajal, Tchernykh, and Salinas Yañez entitled *"Analysis of Deterministic Timetabling Strategies for the Generation of University Timetables*" in which they present an experimental study of deterministic timetabling strategies for the generation of university timetables. They distinguish three prioritization criteria, which differ by the type and amount of information required to select the courses to schedule. They also analyze the performance of the time tabling strategy and prioritization criteria by using a workload with 126 courses, 65 professors and 12 classrooms from a real scenario. To analyze the performance, they conducted a joint analysis of three metrics and evaluate the quality of the obtained solutions in terms of their degradation and performance profiles.

Conclusion

The summary of these studies gives us a brief lens of the work presented in this 5th volume of Supercomputing in México: Where High Performance Computing Flies. However, the contents of each article provide the essence of the work conducted, which is not reflected in this summary. This collection of studies is a real testament of the quality of works presented in ISUM 2014, giving us a brief snap shot of some the works done in and out of the country in supercomputing. It is breathtaking to see the growth of supercomputing in México and the course that is taking. It is evident in the multiple studies presented in this book that scientists are making greater use of supercomputing to achieve their results. We know that as supercomputing continues to evolve in México and Latin America we will see studies that are focused on the evolution of HPC and not so much on the uses of HPC, and contribute significantly to the progress of these systems.

It is with much gratitude that I thank the many authors who contributed to this publication, and the review committee who contributed their time to make this book a collection of quality work. On behalf of the National Committee, I invite you to read about this pioneering work and to participate in the upcoming International Supercomputing Conference in México and share your research work with the scientific community by presenting and submitting your work for publication.



APPLICATIONS

HPC applied to fluorescence fluctuation analysis: contributing to unravel hidden dynamical processes

Marcelo Ortega, Sergio Nesmachnow
Centro de Cálculo, Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
{mortega,sergion}@fing.edu.uy
Juan Angiolini, Valeria Levi, Esteban Mocskos
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Buenos Aires, Argentina
vlevi@qb.fcen.uba.ar, emocskos@dc.uba.ar

Abstract

The fluorescence microscopy techniques and the capability of labeling proteins in the cellular environment with fluorescent tags constitute a significant breakthrough in the way the behavior of cells is studied. Fluorescence correlation spectroscopy and related techniques are extremely useful tools to measure quantitatively the motion of molecules in living cells. This article presents the application of scientific high performance computing techniques to complement the fluorescence fluctuation analysis using stochastic simulation.

A parallel master-slave algorithm is introduced for processing the output data from microscopes studying cells expressing a fluorescent protein. Accurate speedup values are reported in the experimental analysis of the parallel master-slave implementation proposed over a cluster infrastructure.

Resumen

Las técnicas de microscopia por fluorescencia y la capacidad de etiquetar proteínas en ambientes celulares han marcado un antes y un después en la forma de estudiar las células. La espectroscopia de correlación de fluorescencia y las técnicas relacionadas son extremadamente útiles para medir cuantitativamente el movimiento de moléculas en células vivas. Este artículo presenta la aplicación de técnicas de computación científica de alto desempeño para complementar el análisis de las fluctuaciones de fluorescencia por medio de simulación estocástica.

Se presenta un algoritmo paralelo maestro-esclavo para procesar datos de salida de microscopios al estudiar células que expresan proteínas fluorescentes. Valores de aceleración muy interesantes se reportan en el análisis experimental de la implementación maestro-esclavo que se propone, sobre una plataforma cluster.

Keywords: *high performance computing; cellular processes simulation, fluorescence fluctuation analysis*

I. Introduction

Recent advances in fluorescence microscopy techniques and the actual capability of labeling proteins in the cellular environment with fluorescent tags, such as genetically encoded fluorescent proteins, have constituted a significant breakthrough in the way of studying the cell biology. While traditional biochemical experiments require the isolation of the studied biomolecules and provide information in an artificial milieu, these new technologies allowed scientists to explore relevant biological processes in situ.

Since microscopy started to be used to observe cells, it became clear that the distribution of cellular components ranging from small proteins to organelles was not homogeneous in space and time. Trying to understand the rules governing the intracellular organization and its response to specific stimulus has since become a priority.

Fluorescence correlation spectroscopy (FCS) and related techniques (reviewed in [1]) are extremely useful tools to measure quantitatively the motion of molecules in living cells (reviewed in [2]). A schema of the fluorescence correlation spectroscopy measurements is presented in Figure 1.

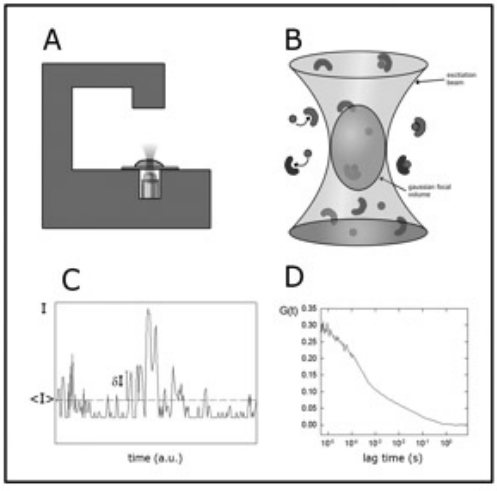


Figure 1: Fluorescence correlation spectroscopy measurements. (A) Cartoon schematically representing a confocal or two photon microscope. The excitation laser is focused by the objective to a small region of interest within a cell. (B) Fluorescent molecules (red) diffuse through the small observation volume (gray) defined in these microscopes. These molecules may, for example, interact with other cellular components (blue and green) altering their dynamics. (C) Representative fluorescence intensity time trace obtained in a FCS experiment. The dotted line shows the average intensity from which the fluorescence fluctuations (dI) are calculated. (D) Autocorrelation function obtained from the fluorescence intensity trace.

Figure 1A shows a cartoon of the experimental setup required in these experiments. The sample (e.g. cells expressing a fluorescent protein) is placed on top of the microscope stage of a confocal or two-photon excitation microscope. The excitation laser is focused in a diffraction-limited spot on the sample and fluorescence photons produced in the small observation volume (~1 femtoliter, as presented in Figure 1B) are collected as a function of time.

Fig. 1C shows a representative fluorescence intensity trace obtained in an FCS experiment. This trace shows fluctuations due to fluorescent molecules moving in and out of the observation

volume. It can be demonstrated that the amplitude of the fluctuations are inversely related to the number of molecules in the observation volume while their duration is given by the dynamics of these molecules. This information can be recovered calculating the autocorrelation function (ACF), as is shown in Equation 1, where $\delta I(t) = I(t) - \langle I(t) \rangle$ represents the fluctuation of the intensity, the brackets indicate the time average, and τ is a lag time. Figure 1D shows a typical example of an ACF plot.

Since this technique captures fluctuations of the fluorescence due to the motion of single molecules, it is necessary to acquire in the order of 105 to 106 data points to recover statistically significant information of the underlying dynamical process.

For simple models such as Brownian diffusion, the ACF analysis yields analytical functions that can be fitted to the experimental data to recover parameters related to the mobility of the molecules. Unfortunately, many dynamical processes in cells cannot be interpreted with these simple models, and in many instances it is not even possible to obtain an analytical function via theoretical analysis of a more complex model.

$$G(\tau)=\frac{\langle \delta I(t)\cdot\delta I(t+\tau)\rangle}{\langle I(t)\rangle^2} \quad (1)$$

In this line of work, the main contributions of the research reported in this article are: i) the application of high performance scientific computing techniques to solve the problem of simulating reaction-diffusion processes in complex biological environments; ii) the development of a parallel master-slave application to solve the problem, deployed in a high performance cluster computing infrastructure; iii) the experimental analysis performed using a set of representative data corresponding to a complex biological phenomena.

The rest of the article is organized as follows. Section II describes the numerical techniques used to simulate the underlying biological processes. Section III introduces the main concepts about high performance scientific computing, performance metrics and cluster infrastructures. After that, the approach using high performance computing techniques for solving the problem is presented in Section IV, just before reporting the experimental evaluation in Section V. The last section summarizes the main conclusions and formulates the main lines for future work.

II. Simulating Complex Biological Phenomena

At cellular scales, a finite number of molecules interact in complex spaces defined by the cell and organelle membranes. In order to simulate stochastic cellular events (i.e. movements, interactions, diverse reactions) with spatial realism at reasonable computational cost, specific numerical and optimization techniques should be employed [3-5]. In the case of typical biological systems and using these optimization techniques in conjunction with Monte Carlo reaction probabilities, it is nowadays possible to study biological systems considering their evolution during a wide range of time, from milliseconds to minutes [6].

The standard approximation for reaction-diffusion systems ignores the discrete nature of the reactants and the stochastic character of their interactions. The techniques based on the chemical master equation, such as the Gillespie algorithm [7], assume that at each instant the particles are uniformly distributed in space. In order to take into account both the full spatial distribution of

the components and the stochastic character of their interactions, a technique based on Brownian dynamics is used. The Mcell [6] simulation package is based on an event-driven algorithm, named Green's function reaction dynamics (GFRD), which uses Green's functions to combine in one step the propagation of the particles in space with the reactions between them [8].

In the GFRD algorithm, the particles move diffusively; it is assumed that if a reaction exists, it follows a Poisson process and it happens instantaneously. This means that the reaction event can be decoupled from the diffusive motion of the particle. The time step of the algorithm is determined such that only single particles or pairs of particles have to be considered, avoiding complex reaction rules.

Due to the nature of the underlying process (tracking of single molecules), it is necessary to simulate a large number of data points to obtain a statistically sound simulation. These numerical experiments demand extremely large computing times: each molecule (data point) should be followed in its motion, and the interaction with other molecules (i.e. reactions) should be modeled and solved in every simulated time step. High performance computing (HPC) techniques come to play a key role to obtain the computing efficiency needed to process a considerably large amount of data points during sufficient time steps, in order to capture the biological process.

Using HPC for modeling and solving reaction diffusion systems is presented in related works like [9, 10]. However, the design and implementation of a specific HPC application to be used in the context of fluorescence fluctuation analysis is novel at the best of our knowledge. Moreover, the use of the techniques described in this paper, which allow performing the simultaneous analysis of several experiments, is a practical contribution in this line of research that considerably increases the capability of generating and validating the proposed models for biological complex phenomena such as fluorescence fluctuation analysis.

III. High Performance Computing

This section introduces the main concepts about parallel high performance computing techniques, the metrics used to evaluate the computational efficiency and the models for communication and synchronization of parallel processes.

A. Introduction

High performance computing is an umbrella term for a set of computational procedures and programming strategies used to efficiently solve complex problems that demand very large computing power [11]. HPC is a steadily growing research field our Latin American region [12]

The kind of problems solved using HPC techniques is frequent in science, especially when simulating physical and natural phenomena, such as tracking single molecules in complex biological environment tackled in this article. The main sources of complexity in these simulations are related to complex methods and mathematical functions, very large search spaces, or handling large volume of data for realistic problem instances, such as in the fluorescence fluctuation analysis problem tackled in this article.

B. Parallel Programming

By using several computing resources simultaneously, parallel HPC techniques allows implementing a cooperative strategy based on dividing the workload into several processes, in order to efficiently solve complex problems in reasonable execution times. The original problem is then replaced by set of (less difficult to solve) sub-problems, which can be tackled in parallel. This cooperative approach allows exploiting the current availability of computing resources in both multicore and parallel/distributed computing environments.

Two main programming techniques are applied in HPC:

- Domain decomposition (or data parallel), focusing on dividing the data handled by the application into (preferable) disjoint pieces, to be processed in parallel by different processes executing on different computing resources.
- Functional decomposition (or control parallel), based on identifying functional units that perform different tasks in the program/application. These tasks are then executed in parallel in different computing resources.

Both HPC techniques rely on communication and synchronization, via a shared resource (usually a shared memory) or explicit message passing, to implement the cooperation that allow solving the sub-problems and integrate the partial results to build a solution of the original problem.

In this work, we apply a domain decomposition technique to perform the simulation of the reaction-diffusion processes in a complex biological scenario.

Two main paradigms exist for parallel programming: shared-memory and distributed-memory.

The shared-memory paradigm is based on a number of processes (usually light processes or *threads*) executing on different cores of a single computer, while sharing a single memory space. This model is easy to implement, and it has a low communication/synchronization cost via the shared memory resource. However, the scalability of the shared-memory approach is limited by the number of cores integrated in a single multicore server and it cannot take advantage of parallel environments composed by many computers. Several libraries are available to implement the thread creation, management, communication and synchronization, including the POSIX thread library for C, OpenMP, and others.

On the other hand, the distributed-memory paradigm relies on many processes executed on different computers. The communication and synchronization is performed by explicit message passing, since there is no other shared resource available than the network used to interconnect the computers. The main feature of this model is that it can take advantage of parallel environments with a large number of computer resources available (i.e. large clusters and grids of computers). Nevertheless, the communication between processes is more expensive than in the shared-memory approach, and carefully implementation is needed to achieve maximum efficiency. Several languages and libraries have been developed for designing and implementing parallel programs using the distributed-memory approach, including the Message Passing Interface (MPI) standard [13].

C. Metrics to Evaluate Performance

The most common metrics used to evaluate the performance of parallel algorithms are the *speedup* and the *efficiency*. The speedup evaluates how much faster a parallel algorithm is than its corresponding sequential version. It is computed as the ratio of the execution times of the sequential algorithm (T_1) and the parallel version executed on m computing elements (T_m) (Equation 2). The ideal case for a parallel algorithm is to achieve linear speedup ($S_m = m$), but the most common situation is to achieve sublinear speedup ($S_m < m$), mainly due to the times required to communicate and synchronize the parallel processes.

The efficiency is the normalized value of the speedup, regarding the number of computing elements used to execute a parallel algorithm (Equation 3). This metric allows the comparison of algorithms eventually executed in non-identical computing platforms. The linear speedup corresponds to $e_m = 1$, and in the most usual situations $e_m < 1$.

D. Models for Communication and Synchronization

$$S_m = \frac{T_1}{T_m} \quad (2) \qquad e_m = \frac{S_m}{m} \quad (3)$$

Three main communication models have been proposed for parallel algorithms:

1. *master-slave*: a distinguished process (the master) controls a group of slave processes. The master process creates and manages the execution of the slave processes, by sending information regarding the sub-problem or function to perform. The slaves perform the computing and communicate back to the master process the results of their execution when they finish the computation. Both the communication and control are centralized in the master process.
2. *lient-server:* two classes of processes are used; the clients, which request services from a process of another kind, and the servers, that handle requests from clients and provide the services. Each server is always waiting for requests, and typically many clients consume services from only one server process. This paradigm provides a model to communicate distributed applications that simultaneously behave as clients or servers for different services.
3. *peer-to-peer*: each client process has the same capabilities to establish communication. Nowadays, this model is well-known by its use in P2P networks to share files. This model can be implemented by developing the logic of a both a client and a server process within the same peer process.

The parallel algorithm proposed in this work supports the fluorescence fluctuation analysis by simulating reaction-diffusion processes in complex biological environment. The algorithm follows the master-slave model for communication, using a distributed-memory approach for synchronization and communication.

IV. Design And Implementation Of The Parallel Algorithm

This section describes the existing approach to the problem, and introduces the proposed parallel computing solution.

A. Design and parallel model

The existing sequential solution for computing the photons emissions is based on executing the *Mcell* application [6] and waiting until the end of the execution in order to get the positions of the molecules for an specified number of iterations (i.e. time steps). In this sequential approach, Mcell generates one output file per time step, containing the list of the spatial coordinates for each molecule in the simulation. This straightforward approach does not scale up to solving problems where a large number of iterations and molecules are required, because the number of generated output files is larger than the maximum number of i-nodes available on the filesystem for all the known operating systems.

In the existing sequential approach, a caveat has been implemented to deal with the previous issue: using a Mcell feature that allows checkpointing and an auxiliary script that controls the Mcell execution. This script executes Mcell for a predefined (small) number of time steps, generating a checkpoint file that allows restarting the Mcell execution. This script performs a concatenation of each individual output, storing the computed iterations in a single but very large output file. In this way, it is possible to run Mcell for a large number of time steps and not running out of i-nodes.

The next phase consists in the data processing. In this stage, each molecule in a predefined zone is tested to check if it emits photons. The sum of all the photons emitted is obtained after processing each of the individual molecules in each time step.

The parallel algorithm was designed to exploit the parallelism present in the problem. There are two main characteristics that the solution should consider:

- The data generation for each iteration is inherently sequential. Each new molecule position in the random walk (Brownian motion) depends on the previous one; therefore the data of one iteration cannot be generated until the previous one is available.
- The processing of the generated data for each iteration is independent from the other ones. The only information required for computing the number of photons emitted at one moment is the molecules positions at that moment.

The parallel algorithm is designed based on overlapping the data generation (Mcell execution) and the data processing phases. Since the processing of the generated data for each iteration is independent of the other ones, the processing program does not need to wait for the whole data set of molecule positions to start. Thus, a pipeline processing strategy can be applied, by overlapping the data generation process and the (already computed) data processing stages.

The parallel algorithm applies a data parallel approach, considering the data from the set of iterations generated by the Mcell program as the problem domain to be split into different computing processes executing on different computing resources. In addition, the algorithm does not work with each iteration separately, but with an aggregation of data from multiple iterations (a data chunk), in order to reduce the overhead in data communication between processes.

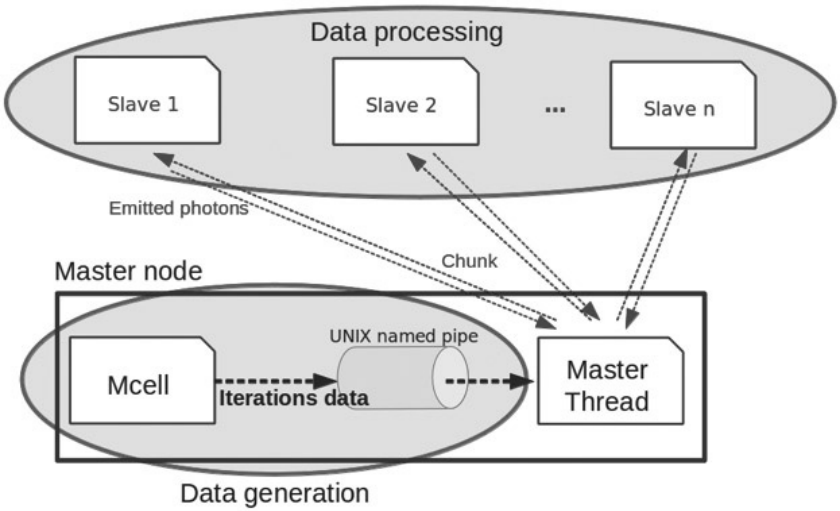
Each of those chunks (i.e. the iterations they have) can be processed independently, so the domain decomposition can be easily applied in this

case. However, the independence does not hold for the data generation, since each iteration depends on the previous one, thus the Mcell execution is inherently sequential, and it cannot be parallelized.

The already commented problem features makes it possible to solve it using a master-slave model for the parallel algorithm using a domain decomposition approach. In the proposed master-slave hierarchy, the master process executes Mcell and generates the data for each iteration. It also creates the data chunks during the program execution, by grouping the iteration data in pieces of a given size. The master is also in charge of distributing the chunks among the available slave processes which performs the data processing. Each slave process performs the calculation of the number of emitted photons in the iterations corresponding to each data chunk received and sends the result back to the master. A diagram of the proposed master-slave parallel algorithm is presented in Figure 2.

In the proposed solution, the original Mcell source code was modified to execute in the master following the concatenation approach already described. The data generated for each iteration is written on the output file, which is in fact a special file named pipe in the operating system. The pipe is used as the interprocess communication (IPC) mechanism to communicate the Mcell program and the master process, allowing the master to perform a significantly more efficient utilization of the input/output resources.

Figure 2: Diagram of the master-slave algorithm for photon emission processing.



B. Implementation Details

The proposed solution was implemented using a distributed-memory approach, in order to be able to execute over a distributed computing resource (i.e. a cluster of computers and/or a grid computing infrastructure) and take advantage of the large processing capability provided by the aggregation of multiple computing resources.

Communication and synchronization between processes executing on different computing resources was implemented using the features offered by the MPI library for parallel and distributed computing [13].

The master process gets the iteration data from the named pipe, and then it prepares the data on grouped named chunks, ready to be sent to a slave process available to start the processing. Each slave process indicates the master when it is free and ready to receive a new piece of work (a chunk) to start working on it. Such indications are made by sending explicit MPI messages between processes/nodes.

The MPI library offers tags as a way to identify types of messages. In our implementation, several tags are used in messages, including:

- WORK_SOLICITATION: when a slave is idle and willing to receive a new piece of work from the master, a message with this tag is sent from the slave to the master.
- WORK_LOAD: the message contains a new chunk being sent from the master to the slave.
- WORK_DONE: once the slave process finishes processing a chunk, it sends the computed results on a message of this type. Note that this message type has the same effect of sending a new WORK_SOLICITATION, because the slave is now idle, so the master can proceed to send a new load of work to be processed. For this reason, the WORK_SOLICITATION tag is only used the first time a slave request data to process, while in later iterations the solicitation is implicit in a message type WORK_DONE.
- END_JOURNEY: when all the required iterations were generated and all data has been processed, the master sends an END_JOURNEY tagged message to each slave, indicating that the program finishes. All slaves then proceed to finish their execution properly.

A similar implementation could be developed without using the first message type, but we decided to keep it, thinking in developing a comprehensive peer-to-peer implementation, able to execute in a large volunteer grid computing system, such as the one provided by the Ourgrid project [14].

In fact, in a cluster-based MPI parallel implementation, the master knows exactly how many and who are the participating slave processes, thus a chunk could be sent to each slave when initializing the program. This option will make the WORK_SOLICITATION message type completely useless, as all the future messages sent from the slave to the master will have the tag WORK_DONE, as explained. The use of the WORK_SOLICITATION message type was preferred, thinking on such future implementations suited for other execution platforms (such as a grid or P2P computing environment), where the master process does not have knowledge of the process participating in the computation. In those

scenarios, the WORK_SOLICITATION message types are needed to be sent from the slaves when they start the execution, in order to let the master know the participating slaves.

As the experimental results reported on the next section shows, the bottleneck on this problem is the generation of data. As this is sequentially executed, the Mcell performance cannot be improved by using parallel machines, limiting the achievable speed up. To overcome this problem, the slaves compute the photons emitted on many areas of the generated space, as if many microscopes were focused at different places of the sample. With this approach, more compute time is used by the slaves on each iteration, solving the Mcell bottleneck issue, while still contributing to the experiments.

V. Experimental Analysis

This section introduces the high performance computing platform used to implement and evaluate the proposed parallel algorithm. After that, the experimental analysis is described and the main efficiency results are commented.

A. Development and execution platform

The parallel master-slave algorithm was developed in C/C++ programming language. The distributed-memory implementation was developed using the MPICH implementation (version 1.2.7) of the MPI library for parallel and distributed computing [13].

The experimental evaluation of the proposed parallel algorithm was performed on a cluster of Dell Power Edge servers, with Quad-core Xeon E5430 processors at 2.66GHz and 8 GB of RAM, from the Cluster FING high performance computing infrastructure [15.16].

B. Experimental Results and Discussion

The results reported in Figure 3 show that the proposed parallel implementation halved the execution times of the original sequential implementation. Even when using just two computing nodes (one master and one slave), the parallelization of data generation and data processing have a notorious impact on the performance. However, the execution with 16 nodes shows no gain in performance, obtaining the same execution time as the two-node case. Indeed, the obtained time for 32000 iterations and 16 nodes is even larger, due to the impact of the parallel implementation overhead.

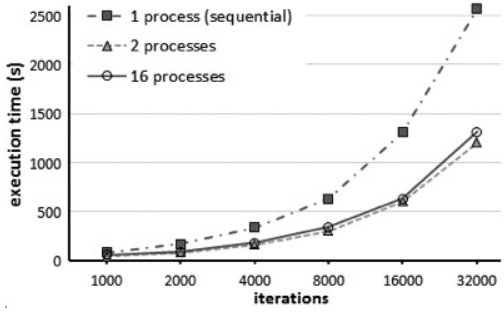


Figure 3: Execution time comparison for different number of processes.

The poor speedup behavior is due to the bottleneck on the data generation, which limits the capabilities of the master to send new data to idle slaves, and therefore the performance is highly dependent on the Mcell data generation speed.

When introducing multiple microscopes (i.e. a larger number of observations) on the data analysis, a better speedup behavior is observed.

Table 1 reports the execution time comparison (in minutes) for both the sequential and the parallel implementation using 2 and 24 computing nodes. Both versions compute the same number of iterations (24000) for a different number of staged microscopes, which allows computing the photons emitted on different areas.

In this way, a better relation between the effort needed to deliver a new chunk of data from the master to each slave and the computing time spent on processing the data it is achieved, overcoming the Mcell bottleneck and reducing the communication overhead.

Table I. Execution Time Analysis

# processes	Number of virtual microscopes (24000 iterations)					
	1	125	500	1000	2000	4000
sequential	34.2	83.0	238.8	432.1	835.9	1633.1
2	13.2	16.2	17.4	25.5	57.5	121.0
24	14.1	14.0	14.0	15.3	18.8	20.9
speedup	2.4	5.9	17.1	28.2	44.5	78.1

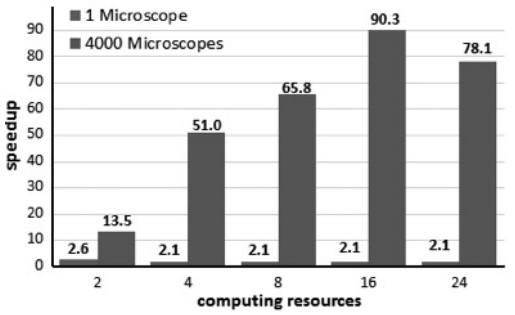


Figure 4: Speedup analysis for one microscope and for 4000 microscopes (all executions with 24000 iterations).

The speedup analysis reported in Figure 4 for different number of computing resources demonstrates that while the executions with a single staged microscope maintain the same speed up for any number of nodes, significantly better results are obtained when using a large number (up to 4000) of microscopes. In this case, increasing the number of computing nodes produce a notable increase on the speedup factor, always achieving a super lineal value ($S_m > m$).

The best speedup values are obtained when using 16 computing resources. A slight lower value is obtained value for 24 computing resources. When using a larger number of slave processes, the Mcell bottleneck problem raises again, as the data generation speed is not enough to have all slaves working permanently.

Table II summarizes the execution time analysis for different number of parallel slave processes (in minutes), for one and 4000 microscopes. We distinguish between total time (the total time demanded to execute the application), and effective processing time (which evaluates the time spent in data processing). The results in Table II clearly indicates how the ratio between processing time over number of parallel processes reduces when using a larger number of parallel processes, demonstrating the scalability of the proposed approach when dealing with very large volumes of data (in the case of study including 4000 microscopes, up to 1.31 TB of data, including input and output, are processed). This feature is very important for any parallel algorithm and clearly states the contribution of the proposed parallel implementation.

Table II. Execution Times Analysis

# parallel processes	total time		processing time		processing time/ # processes	
	1M	4000M	1M	4000M	1M	4000M
2	13.0	121.2	0.0	108.2	0.0	60.6
4	16.2	32.0	3.2	19.0	0.8	8.0
8	16.4	24.8	3.4	11.8	0.4	3.1
16	15.9	18.1	2.9	5.1	0.2	1.1
24	16.4	19.0	3.4	6.0	0.1	0.8

VI. Conclusions

The fluorescence microscopy techniques and the actual capability of labeling proteins in the cellular environment with fluorescent tags are a significant breakthrough in the way the behavior of cells is studied. The distribution of cellular components ranging from small proteins to organelles is not homogeneous in space and time. The rules governing the intracellular organization and its response to specific stimulus have become a priority. Fluorescence correlation spectroscopy and related techniques are extremely useful tools to measure quantitatively the motion of molecules in living cells.

The Mcell simulation package is based on the GFRD event-driven algorithm, which uses Green's functions to combine in one step the propagation of the particles in space with the reactions between them. It is necessary to simulate a large number of data points to obtain a statistically sound simulation.

This article presented the application of a master-slave parallel algorithm for the simulation of reaction-diffusion problem, applied in this case to the fluorescence fluctuation analysis. By using multiple computing resources, the proposed technique allows efficiently performing the simultaneous analysis of several experiments, which is a practical contribution in this line of research, in order to generate and validate models for biological complex phenomena such as fluorescence fluctuation analysis.

The proposed parallel algorithm was implemented using a distributed-memory approach and taking advantage from the processing capability provided by the aggregation of multiple computing resources. Additionally, the original Mcell source code was modified to avoid writing a large amount of output files, which was a serious drawback of the sequential implementation. In the parallel version, the data generated in each iteration is sent via IPC to the master process, who then delivers this data to the slave processes. This way, each slave receives a chunk containing the data for a given number of iterations, computes the reaction-diffusion simulation and returns the number of emitted photons.

The experimental analysis of the proposed parallel algorithm was performed over a cluster HPC infrastructure. Introducing parallel computing to the existing sequential solution allowed reducing the execution times of reaction-diffusion simulations to values lower than 50% of the original ones. Moreover, the use of parallelism allowed the analysis of several simulations simultaneously, increasing the number of valuable results obtained on each execution.

The experimental results showed a large reduction on the execution times when executing the developed parallel implementation on two computing resources. However, the times do not reduce further when using more computing resources. The program used for data generation (Mcell) uses a sequential algorithm, and therefore the potential benefit the entire solution had from

parallel computing was limited.

Despite the previous commented results, the parallel algorithm emerged as a powerful tool for processing very large volumes of data. Indeed, notably speedup values were obtained when modifying the data processing on the slave in order to achieve a better ratio between the effort needed to deliver a data chunk from the master to the slave and the computing time it spend on processing. Simulating the case where a large number of microscopes are staged at different places of the sample allows the slave to make a more computing intensive job for each chunk, generating more experimentation values useful to the analysis.

Simulating 4000 staged microscopes to cause a greater computing effort made at the slave for each chunk, the proposed parallel implementation achieved notably superliner speedup values, up to 90.3 when using 16 nodes. Keeping a similar relation between nodes and microscopes staged, according speedup values can be obtained for execution on a larger number of nodes.

Summarizing, the execution time analysis demonstrated that using the approach proposed in this article, the parallel algorithm was able to compute 24000 iterations and generate 4000 times more data than the sequential version, while demanding the same time than the sequential algorithm requires to compute 16000 iterations.

The main lines for future work include further improving the parallel model to scale up and solve efficiently all the problem variants, and applying a fully distributed computing approach to solve similar problems dealing with information processing in biology by using large grid infrastructures.

Acknowledgment

The work of S. Nesmachnow is partly supported by ANII and PEDECIBA, Uruguay. J. Angolini has a scholarship from CONICET, Argentina. E. Mocskos and V. Levi are members of CONICET, Argentina.

References

[1] M. Digman and E. Gratton, “Lessons in fluctuation correlation spectroscopy”. Annual Review of Physical Chemistry. 62:645-668, 2011.

[2] E. Haustein and P. Schwille, “Ultrasensitive investigations of biological systems by fluorescence correlation spectroscopy”. Methods 29:153-166, 2003.

[3] T. Bartol, B. Land, E. Salpeter, and M. Salpeter. “Monte Carlo simulation of MEPC generation in the vertebrate neuromuscular junction”. Biophysical Journal 59:1290–1307, 1991.

[4] J. Stiles, D. Van Helden, T. Bartol, E. Salpeter, and M. Salpeter. “Miniature endplate current rise times < 100μs from improved dual recordings can be modeled with passive acetylcholine diffusion from a synaptic vesicle”. In Proceedings of National Academy of Sciences, 93:5747–5752, 1996.

[5] J. Stiles and T. Bartol, “Monte Carlo methods for simulating realistic synaptic microphysiology using MCell”. In: DeSchutter, E., editor. Computational Neuroscience: Realistic Modeling for Experimentalists. CRC Press; Boca Raton, FL. p. 87-127, 2001.

[6] R. Kerr, T. Bartol, B. Kaminsky, M. Dittrich, J. Chang, S. Baden, T. Sejnowski and J. Stiles, “Fast Monte Carlo Simulation Methods for Biological Reaction-Diffusion Systems in Solution and on Surfaces”. SIAM Journal on

Scientific Computing 30(6), 3126-3149, 2008.

[7] D. Gillespie, “A general method for numerically simulating the stochastic time evolution of coupled chemical reactions”. Journal of Computational. Physics 22, 403, 1976.

[8] J. van Zon and P. ten Wolde, “Simulating Biochemical Networks at the Particle Level and in Time and Space: Green’s Function Reaction Dynamics”. Physics Review Lettters 94(12), 128103, 2005.

[9] F. Molnár, F. Izsák, R. Mészáros, and I. Lagzi, “Simulation of reaction-diffusion processes in three dimensions using CUDA”, Chemometrics and Intelligent Laboratory Systems, 108(1), 76-85, 2011.

[10] E. Martínez, J. Marian, M. Kalos, and J. Perlado, “Synchronous parallel kinetic Monte Carlo for continuum diffusion-reaction systems”, Journal of Computational Physics, 227(8), 3804-3823, 2008.

[11] I. Foster, “Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering”. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[12] E. Mocskos, S. Nesmachnow, G. Hernández, “HPCLatAm: Towards the integration of the research communities on High Performance Computing in LA Southern Cone”. VI Conferencia Latinoamericana de Computación de Alto Rendimiento, San José, Costa Rica, 2013.

[13] W. Gropp, E. Lusk, and A. Skjellum, Using MPI: Portable Parallel Programming with Message-Passing Interface. MIT Press, 1999.

[14] W. Cirne, F. Brasileiro., N. Andrade, L. Costa, A. Andrade, R. Novaes, and M. Mowbray, “Labs of the world, unite!!!”. Journal of Grid Computing, 4:225–246, 2006.

[15] Cluster FING. On line www.fing.edu.uy/cluster, January 2014.

[16] S. Nesmachnow, “Computación científica de alto desempeño en la Facultad de Ingeniería, Universidad de la República”, Nesmachnow S. Revista de la Asociación de Ingenieros del Uruguay 61:12-15, 2010 (in Spanish).

Application of an Adaptive Inversion Frequencies Algorithm for Router Bandwidth Improvement

[1]Evgeniy Kravtsunov, [1]Timur Mustafin,
[2]Andrei Tchernykh, [1]Valery Perekatov, [1]Alexander Drozdov
[1]MCST, Moscow, Russia
{kravtsunov_e, Timur.R.Mustafin, perekatov}@mcst.ru
alexander.y.drozdov@gmail.com
[2]CICESE Research Center, Ensenada, Baja California, México. chernykh@cicese.mx

Abstract

In this article, we study the practical application of the inversion frequencies algorithm for the compression of IP network data. We present a modified version of the algorithm with adaptability. We show that our algorithm can be effectively used for DNS/UDP traffic compression, allowing a 3-15% increase in router bandwidth.

Keywords. *bandwidth, Compression. Inversion Frequencies Algorithm*

1 Introduction

The Inversion Frequencies Algorithm (IF) is used to obtain an output sequence, which is better compressible than the input sequence. The IF scans the input sequence for each alphabet symbol, and produces zero output values for the last symbol of the alphabet. Since a sequence of zeroes does not give any information, it can be omitted. Therefore, the output sequence of the IF stage is shorter than the input sequence.

On the other hand, the IF adds more overhead for the transmission of the symbol distribution or terminator symbols, which makes IF less attractive for small amounts of compressible data.

In this paper, we propose and analyze a new algorithm to increase router bandwidth and data transmission speed using compression of IP network data. The aim was to implement lossless data compression meeting the following requirements:

- The compression algorithm should encode the stream in one pass.
- The amount of main memory required to implement the algorithm, should not exceed 32 KB.
- The costs associated with the transfer of the control information needed to decode the data should be minimized.
- The encoding time for one data block must not exceed the network transmission time saved by the compression.

An additional purpose of the study is to determine the type of network traffic, for which the use of this compression algorithm is effective. The compression algorithm must complete data stream coding in one pass of the input data, which corresponds to an algorithm complexity of *O(n)*. The Stream compression and memory minimization requirements mean that the input stream cannot

be modeled as a source with memory [1]. This means that compression algorithms based on search coincident substrings and dynamic dictionary [2] cannot be used as solution of the problem. The input stream has to be modeled as a source without memory.

The use of the general interval transformation algorithm is theoretically justified for this kind of sources. For sources without memory, [3] shows that general interval transformation encoding with Rice-Golomb codes provides good compression ratio, which tends to approximate the theoretical limit of compression. Practical implementation of the general interval transformation called inverted frequency coding (IF-transformation) is described in [4].

The implementation in [4] is promising; based on the standard test suite compression results, and an analysis of the proposed algorithm. However, it cannot be used as solution for our problem. To provide an efficient compression for small blocks of the input data, we propose the following improvements: adaptability of IF- transformation, and supporting alphabet letter sizes of 2 and 4 bits.

The adaptability allows an improvement of compression ratio, making it almost equal to the theoretical limit. Supporting letters of sizes 2 and 4 bits minimizes the costs associated with the additional information needed for decoding and transmitting the compressed code.

Compression efficiency for the algorithm is studied based on the Calgary Corpus standard test, and on the real data obtained on the collecting of outgoing internet traffic of MCST. Measurements made on the Calgary Corpus test confirmed the improvement of the compression ratio by adding adaptability to the algorithm.

Measurements on real data have revealed traffic types that can use our small blocks data compression algorithm. Based on these positive results, we have developed a generalized description of the main functions of the algorithm, which can be used to develop an instruction set and internal logic for a compressing ASIC (Application-specific integrated circuit.)

2 Compression Algorithm

The compression algorithm is based on the inverted frequency transformation by Rice-Golomb encoding. Detailed description of the inverted frequency transformation algorithm is given in [Kravtsunov]. Let us present the main improvements related to the adaptability of inverted frequency transformation. The main features of the algorithm are the following:

Alphabet A[N] is a set of symbols (letters) of the text. Each alphabet symbol can be represented by 8, 4 or 2 bits. The Power of alphabet N is a quantity of letters (symbols). So, if each symbol is represented by 8 bits, the power of alphabet is $N=2^8=256$ symbols. For 2-bit and 4-bit symbols power of the alphabet is 4 and 16 symbols respectively.

Entropy Q is the minimal quantity of bits necessary for encoding one symbol of the text, with certain statistics of the symbols. Entropy defines the theoretical limit of the compression. This limit of compression, K, is calculated with the next equation:

$$K = \left(1 - \frac{Q}{S(N)}\right) * 100\% \qquad (1)$$

Where N is the power of alphabet, *S(N)* is the number of bits for representation of one symbol of

the alphabet with power equals to N.

Alphabetical order is the order of symbols in which $F(A[N])>F(A[n+1])$, where F(A[n]) is a frequency of symbol A[N] appearing in the text, $F(A[N+1])$ is a frequency of symbol $A[N+1]$ appearing in the text, n belongs to interval $[0;N-1)$.

Lexicographically older symbol, the symbol $A[i]$ is lexicographically older than the $A[j]$ symbol of the $A[N]$ alphabet, if $i > j$ and the $A[N]$ symbols are sorted in alphabetical order.

IF transforms a set of text symbols to N sets of offsets values. Offset is a quantity of symbols lexicographically older than the current symbol, which appeared from previous position of current symbol or from the beginning, if current symbol meets at the first time. If the text presents alphabetical order, long series of small offsets values appear, which can be effectively compressed by Rice-Golomb codes. The key point is the alphabetical order of the statistics of the text. In [4], alphabetical order is calculated for each text block of 65536 bytes, and transmitted with the encoded text block.

From the complexity point of view, this method is acceptable. The downside is that with alphabetical order, statistical heterogeneity within the text block is not considered, reducing the compression ratio. Reducing the size of the block in the implementation of [4] does not solve the problem, and increases the costs associated, since we need to transmit the alphabetical order for each block.

This problem is solved by dynamically changing the alphabetical order in the conversion process, in accordance to the changing statistics of the text.

2.1 Dynamic Alphabetical Order Changing

In our algorithm, as in [4], a full binary tree is used for inverted frequency transformation (Figure 1). There is a counter C in each tree node, initialized to zero. The tree leaves are initialized by an A[N] symbol, following an alphabetical order. The G' value (the number of lexicographically older symbols from the beginning of the text) is set initially to 0.

Direct transformation of each letter of the input text in [4] is performed by the next sequence of actions (Figure 2):

1. Determinate tree leafs, which correspond to input letters. Calculated offset from the beginning of the text, G, is initialized to zero.
2. Visit each node from the selected leaf to the root with 2 actions on each node:
   • Value G is increased by right brother tree node counter value
   • Current tree node counter is increased by 1
3. Offset value is calculated by Offset = G – G';
4. Previous G' value for the symbol is replaced with current G value.
5. Offset is transmitted to the Rice-Golomb algorithm input

Our main improvement is to add an alphabet sorting, which changes alphabetic order after encoding each symbol (Figure 3). The Sorting, in decreasing order of appearance counters, is done after the calculated value Offset is passed to the Rice-Golomb algorithm input. The Quicksort algorithm is used for rebuilding. The proposed method is called "Adaptive rebuilding of binary

coding tree". The inverse transformation is also performed using the structure of a balanced binary tree. Rebuilding the tree during the reverse transformation is performed after decoding each letter. Sorting criteria in the reverse and forward transformations coincide, which allows you to decode text.

2.2 2 and 4 bits Letter Sizes Support

In [4], the authors considered alphabet power equals to 256. The letters of that alphabet are 8 bit symbols and the offset was represented by an unsigned int type, which size is 4 bytes. An array of offsets numbers comes out from direct transformation too.

In order to obtain the source text from a compressed format, the array of offsets obtained by direct conversion has to be provided as the input of the inverse algorithm.

The array of offsets numbers is sent with bit-code as a sequence of 1024 bytes (256 symbols * sizeof(int)). Thereby, using eight-bit symbols for each text block introduces costs estimated at 1024 bytes. It forbids effective use of the algorithm from [4] for small data block compression.

For solving the problem of small data block compression, we introduce the support of alphabet powers of 4 and 16 symbols, two and four-bit symbols. We also changed the data type for storing offsets numbers from unsigned *int* to unsigned short. The data type has limited maximal block size of input text of 65536 bytes, but for alphabet power reduction it allows decreasing the costs. The costs are equal to 8 and 32 bytes per block.

2.3 Entropy Calculation Function

Entropy is the minimal bit quantity, required for encoding of one input symbol of a text. We implemented an entropy calculation function, which is used for input blocks of any size and alphabet powers of 4, 16 and 256 symbols. It is applied for estimating the compression efficiency. It is also helpful to compare compression ratios against the theoretical limit.

There are two ways to do entropy calculation:

1. Modeling text as a source with memory.
2. Modeling text as a source without memory.

Which one we choose depends on type of compression algorithm. A source with a memory model is focused on algorithms that do not have a fixed alphabet, but instead have a dynamically created dictionary. A source without a memory model is suitable for algorithms which have a constant alphabet. It is not extended in the transformation process by a dictionary.

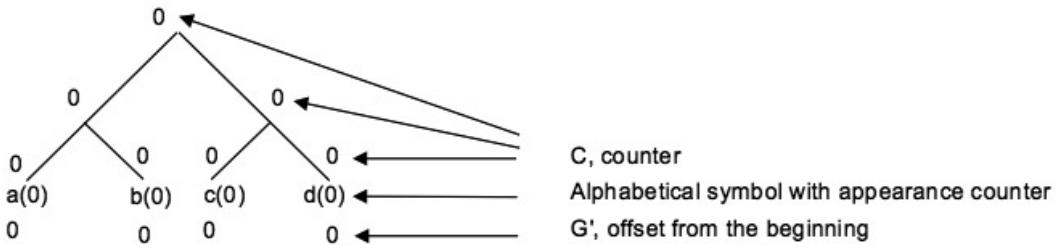


Figure 1: Full binary tree after initialization

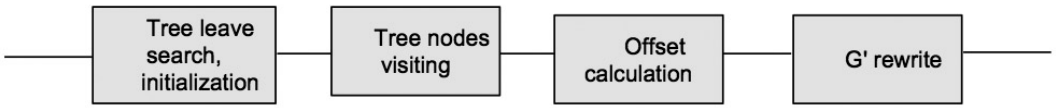


Figure 2: Direct inverted frequency coding

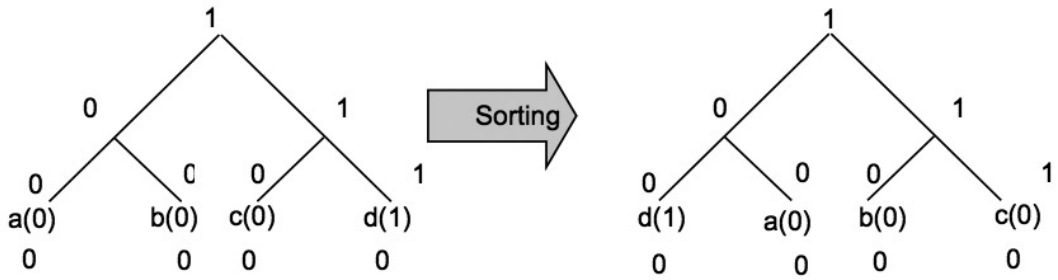


Figure 3: Adaptive rebuilding of binary tree

Because the inverted frequency transformation is based on using of constant alphabet (alphabetic order can change, but the set of symbols is fixed), a model of source without memory is used for estimating the efficiency of the algorithm. In that case, the entropy function H for text with length D looks like:

$$H = \sum_{i=0}^{N} \left(\frac{q_i}{D}\right) * \log_2\left(\frac{q_i}{D}\right) \qquad (2)$$

Where q_i is the quantity of i letter in text.

3 Measurement Results

In this section, we present the compression ratios of the adaptive inverted frequency transformation with Rice-Golomb coding for two types of test data:

1. Standard test set for compression algorithms Calgary Corpus;
2. Real data – net traffic files of different types.

The measurements on Calgary Corpus aimed to compare of obtained compression ratios with theoretically calculated compression limits. The entropy calculation function was used for theoretical compression limit calculation. Calgary Corpus measurements were also used to measure the effectiveness of adaptive rebuilding of binary coding tree method, and estimated usefulness of alphabet powers of 4 and 16 symbols.

The analysis on real data shows that it is possible to determine traffic type, which can be compressed. The data stream contains a lot of small data blocks with different length and statistics.

3.1 Calgary Corpus Measurements

Table 1 contains the results produced on Calgary Corpus test set by adaptive interval frequency transformation with Rice-Golomb coding. Table 1 shows compression ratios obtained by algorithms with rebuilding and without rebuilding with the theoretical compression limit for each file from the Calgary Corpus test set. The results were obtained with an alphabet power of 256 (symbol size is 8 bits).

Table 1: Comparison of compression ratios in percent of Calgary Corpus files by algorithms without rebuilding and with adaptive rebuilding the alphabet order. Symbol size is 8 bits.

File	Without rebuilding	With rebuilding	Theoretical limit
bib	33,51	33,52	34,99
book1	41,73	42,43	43,41
book2	37,00	39,02	40,09
geo	28,14	28,38	29,42
news	33,65	33,74	35,13
obj1	18,29	22,13	25,65
obj2	19,64	19,83	21,75
paper1	35,14	36,31	37,71
paper2	39,73	40,94	42,48
paper3	36,70	39,95	41,69
paper4	21,26	36,80	41,25
paper5	31,27	33,09	38,30
paper6	33,12	35,36	37,38
pic	77,93	78,69	84,87
progc	32,63	32,52	35,01
progl	38,73	38,48	40,37
progp	37,56	37,18	39,14
trans	29,35	28,58	30,84
Average	34,74	36,50	38,86

Table 1 shows that an adaptive rebuilding of the binary coding tree improves compression of standard files by 2 % and brings compression ratios closer to theoretical limits.

Results with compression ratios for alphabet powers 4 and 16 (symbol sizes equals 2 and 4 bits) are given in Table 2. Results for alphabet power 256 (symbol size is 8 bits) is also given to facilitate comparison.

In Table 2, we find that the theoretical limit of compression value and real compression are reduced as the alphabet power decreases. This situation is not good for compressing big data blocks, where the costs for transmission of control information in compressed format can be neglected. The costs cannot be neglected for smaller data blocks with sizes from 12 to 1500 bytes. Costs obtained with alphabet power 256 do not allow compressing small data blocks. Table 2

shows that a 16 symbols power alphabet provides near to theoretical compression. Hence, we conclude suitability of 16 symbols power alphabet for network traffic compression.

3.2.1 Different Net Traffic Types, DNS Traffic

Table 3 contains results of theoretical compression limits for files with different traffic types. The entropy is calculated for the set obtained by merging all data segments into one. This allowed us to determine compression feasibility of either traffic type.

Type of traffic and compressible DNS traffic were recorded in the file of size 2607767 bytes, consisting of 19102 small packets of different lengths, from 12 to 1500 bytes. This type of traffic represents a suitable set to test the conversion algorithm and Rice-Golomb encoding for small blocks.

For the experiment, data segments of the size specified in the header were extracted from traffic packets. Measurements were carried out on an alphabet capacity of 4 letters (costs are minimal) using the adaptive refinement of the binary tree. For each block of data, the entropy is also calculated. The experimental results are shown in Table 4.

It is clear that despite the theoretical calculations showing that the packets can be compressed (column “entropy”),only 30% of the available sample packs were compressed. This discrepancy is due to the presence of costs, that is, the presence of additional information in the compressed code needed to decode the data. Thus, the total effective compression of network traffic can be achieved by compression of the packets with a good compression ratio, and the remaining packets will be transmitted uncompressed. To determine which packets are suitable for compression, we propose a decision taking method based on a comparison of the calculated values of the entropy of the data segments with the criterion R obtained by the experimental method.

R value defines the boundary values of entropy. In fact, the R value characterizes non-optimality of compression and allows to evaluate the possibility of real compression of the data.

3.2.2 Method of calculating the decision criterion for package compression

The proposed method of deciding about compression is based on the assumption of dependency between the entropy and compression rate. If the entropy is greater than R, it is decided not to compress the package, otherwise it is compressed.

The size of compressed files is calculated in advance. This allows to known what packages have to be compressed. This information allows us to judge the correctness of the decision.

We tested the algorithm on 19102 packets with different values of R ranging from 0.6 to 2. The result of the simulation is presented as a function of R. Figure 4 shows the total size of packets of the network after the application of the algorithm.

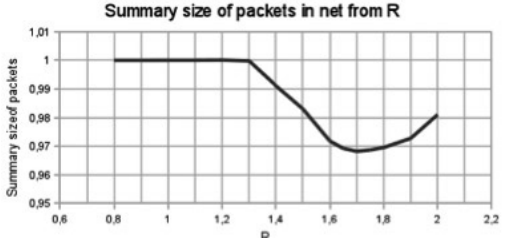


Figure 4: Compressed Size of packets

We show that the optimal value of R is 1.7. This value provides maximum compression.

3.2.3 Compression DNS packet flow using the decision method

Table 5 shows the characteristics of the decision-making method for the DNS compression packages. Total compression ratio of all flow coincides with what is shown in Table 3 for DNS traffic. The discrepancy between the total compression ratio as a set of small packets and entropy presented in Table 3 is due to the fact that the compression algorithm does not have enough statistics of such small packages.

Table 2: Compression ratios of Calgary Corpus files for alphabets with different powers

	Alphabet Power			Theoretical Compression Limits		
File	4 letters	16 letters	256 letters	4 letters	16 letters	256 letters
bib	-2,46	9,67	33,52	2,74	11,77	34,99
book1	-2,53	15,51	42,43	2,45	16,41	43,41
book2	-2,57	14,21	39,02	2,20	15,25	40,09
geo	12,19	14,38	28,38	16,81	17,92	29,42
news	-2,94	11,28	33,74	1,91	12,75	35,13
obj1	6,08	15,40	22,13	9,85	16,25	25,65
obj2	0,13	7,88	19,83	5,02	10,18	21,75
paper1	-2,58	13,77	36,31	2,19	14,70	37,71
paper2	-2,40	15,23	40,94	2,43	16,45	42,48
paper3	-3,11	15,06	39,95	2,41	15,96	41,69
paper4	-2,69	13,96	36,80	2,35	15,26	41,25
paper5	-2,98	12,39	33,09	2,13	14,34	38,30
paper6	-2,67	12,68	35,36	2,10	13,75	37,38
pic	44,20	67,06	78,69	73,82	80,78	84,87
progc	-3,45	10,07	32,52	2,13	12,41	35,01
progl	-3,82	10,21	38,48	1,98	12,11	40,37
progp	-2,25	10,70	37,18	2,87	12,44	39,14
trans	-3,53	6,13	28,58	1,90	8,29	30,84
Average	1,26	15,31	36,50	7,63	17,61	38,86

Table 3: Results of the analysis of various types of network traffic for compressibility

Traffic type	TCP/UDP	Port	Theoretical compression limit: 4 alphabet letters	Theoretical compression limit: 16 alphabet letters
HTTP	TCP	80	0,00	0,50
HTTPS	TCP	443	0,00	0,00
SMTP	TCP	25	0,00	0,00
DNS	UDP	53	14,75	18,75

Table 4: Results of experiments on the compression of data segments packet DNS traffic.

Data segment size	Total	Compressed	Uncompressed	Entropy
< 300 byte	17041	5016	12025	0,83-1,96
300-500 byte	814	489	325	1,29-1,95
> 500	1247	246	1001	1,26-1,97
	19102	5751	13351	0,83-1,97

Table 5: Degree of compression using different decision-making principles

All packets	Method of making a decision with R = 1,7	Using a posteriori knowledge about compression package
1,9	3,18	4,73

4. Hardware Support for Compression

Modern trends in the network to accelerate the traffic to increase the MTU to 9000 and 16000 bytes (Jumbo and SuperJumbo frames) [5], allow to assume that the hardware implementation of adaptive algorithm Literal interval transformation and coding Rice-Golomb compression can be useful for blocks of medium size. For these sizes, one can use the alphabet of power 16, allowing us to compress DNS traffic up to 15%. Hardware implementation of the algorithm should provide the following APIs:

- Initialization of the tree
- Tree traversal - increase counters
- Rebuilding of the tree
- Calculation of bit code conversion Rice –

Golomb
- Adaptation of the Rice – Golomb factor
- The decision based on the value of the entropy

5. Conclusions and Future Work

This article describes the implementation of the algorithm for adaptive Literal interval transformations, which allows compressing small blocks of data.

The implementation uses an adaptive reconstruction of binary tree literally equal interval transformation and alphabets capacity of 4 and 16 characters.

The authors experimentally determined the type of traffic being compressed by the algorithm. A method for estimating the compressibility of the block, based on the theoretical calculation and comparing it with the criteria characterizing the algorithm was presented. APIs for hardware compression are also proposed.

References

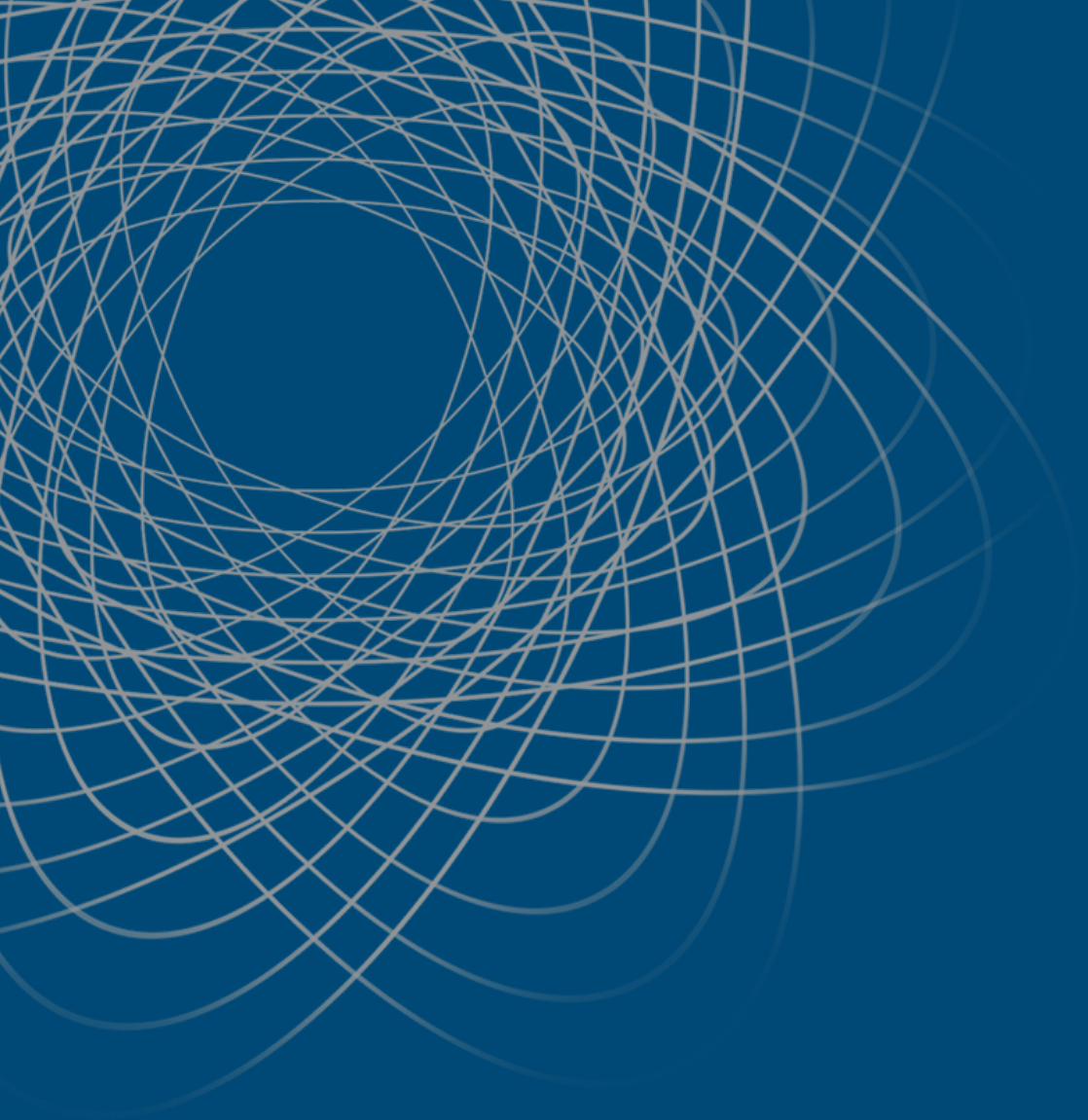
1. Krichevsky, R. (1989). Compression and information retrieval. *Radio and Communication*, Moscow.

2. Nelson, M. & Gailly, J. (1996). The Data Compression Book. *M & T Books*, New York.

3. Brailovski, I.V. (2003). Effective Data Compression Using the Generalized Interval Transformations. Ph.D. Thesis, Moscow.

4. Kravtsunov, E.M. (2004). Effective Methods of Data Compression for Embedded Systems. High *Performance Computing and Microprocessors* , ,IMVS RAS, Issue 6 , p. 103 , Moscow.

5. Leitao, B. (2009). Tuning 10Gb network cards in Linux, a basic introduction to concepts used to tune fast network cards. *Proceedings of the Linux Symposium*, p.169, Quebec, Canada.



CLOUD COMPUTING

Cost Optimization of Virtual Machine Provisioning in Federated IaaS Clouds

Fermin A. Armenta-Cano and A. Tchernykh are with *Computer Science Department, CICESE Research Center, Ensenada, B.C., México (e-mail: jcortes@cicese.edu.mx, chernykh@cicese.mx).*
Ramin Yahyapour *with GWDG - Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, Germany, (e-mail: ramin.yahyapour@gwdg.de)*
Jarek Nabrzyski *with Center for Research Computing, University of Notre Dame, (e-mail: naber@nd.edu)*

Abstract.

In this paper, we present cost optimization model in cloud computing, and formulate the cost-aware resource allocation problem that provides cost-efficiency in the context of the cloud federation. Our model assumes a cloud provider with multiple heterogeneous resources or data centers. The provider needs to control amount of resources to avoid overprovisioning and increasing capital costs. To reduce an importance of known Build-To-Peak approach that means building infrastructures for top demands with over-provisioning in total operating time, cloud provider has to collaborate with other providers to be able to fulfil requests during peak demands by using idle resources of other peers. In this scenario, it is important to find a trade-off that allows reducing the total investment and operational cost. We address cost minimization problem in the hierarchical federated cloud environment, where external clouds are parameterized by renting costs per time unit. We discuss several cost optimization algorithms in distributed computer environments with the goal to understand the main characteristic of the cost optimization. We conclude by showing how none of these works directly addresses the problem space of the considered problem, but do provide a valuable basis for our work.

Keywords. Cloud computing, IaaS, Operational Cost, Service Level Agreement.

I. Introduction

Cloud Computing is an innovative distributed computing paradigm that is widely accepted by public and private organizations. It focuses on providing three main types of services through the Internet with quality of services to their customers: SaaS, PaaS and IaaS.

Software as a Service (SaaS) is a model of software deployment whereby a provider offers a software application on the internet rather than a software package to be buying it for the costumer. Examples are online email providers like Google Gmail, Microsoft hotmail, Google docs, and Microsoft Office 365. Platform as a Service (PaaS) fills into the system level, which provides platform to run end user's applications without downloads or installation. Examples are the Google App Engine, which allows applications to be run on Google's infrastructure.

Infrastructure as a Service (IaaS) model the provider offers hardware resources such as storage capacity and power computing resources like CPU capacity and memory capacity as a service over the

internet. In this way the costumers rent only the necessary resources for they need instead to buy all the equipment. One of the leading vendors that provide this service is Amazon Web Services (EC2 and S3) for processing and storage.

In this paper, we focus on the IaaS type of clouds.

Service Level Agreement (SLA) is a business component of extreme importance in Cloud computing, which represents a contract that specifies the minimum obligations of the provider to its customers, or expectations of the customers to receive in exchange of the price paid.

Two most important IaaS cloud challenges that providers must addressed are the energy efficiency and cloud provider costs.

IT companies must meet global and national goals for carbon-footprint reduction, and must compensate for noticeable increases in energy expenditures. Thus, technological energy saving measures are mandatory ingredients for any emerging information and communication technology. In this context, IaaS cloud providers must evaluate and assess different strategies to improve energy efficiency in their data centers, including computing, cooling, and power supply equipment. This involves defining and using unified metrics, such as the power usage effectiveness (PUE) or data center infrastructure efficiency (DCIE) metrics, among others. These help cloud providers measure their data centers' energy efficiency, compare the results against other cloud infrastructures, and decide what improvements are necessary.

Some open issues include developing more efficient energy-proportional servers that consume power proportionally to utilization level and improving cloud resource allocation, consolidation, and migration strategies that consider each particular service's workload profile. For example, the reallocation of service components for consolidating purposes can be efficient from a power-saving perspective, but can be counterproductive for service performance when workloads have tightly coupled communications requirements. Another future direction is to study mechanisms for advanced resource provisioning, based on a service's historical execution profile, to predict the resources that the service will consume, allowing for optimal provisioning those results in lower energy consumption.

II. Provider Costs

The costs are changing slightly depending on whether the cloud is public or private, their general structures are similar.

Provider costs are primarily tied to their assets and the maintenance of these assets. For example, providers have an infrastructure that needs to be powered and cooled. Similarly, storage providers have storage arrays containing storage disks, and these arrays are connected to chassis which are all housed in data centers. So, major provider costs can be categorized as follows [1]:

1. Servers cost (compute, storage, software)
2. Infrastructure cost (power distribution and cooling, data center building, etc.)
3. Power draw cost (electrical utility costs)
4. Network cost (links, transit, equipment)

A number of other costs exist.

Optimization is very important for providers to

offer competitive prices to prospective customers. Inefficient resource management has a direct negative effect on performance and cost.

In the shared environments, it is often difficult to measure costs, usage, and value of virtual and physical resources and the services they provide. It is a challenge optimizing the costs of running workloads, usage costs, and defining billing rates to cover total cost of ownership. Detailed cost management can optimize resource usage and improve profitability. An effective step, then, is to measures resource usage at granular levels.

The main objective of IaaS providers is to obtain maximal profits and guarantee QoS requirements of customers. Efficient resource allocation strategies should be exploited in dynamic environment to provide needed quality of service. Graham [2] demonstrated that the problem of scheduling jobs on a set of heterogeneous resources is NP-complete.

On the other hand, more challenges should be met, when providers do not have enough resources to satisfy all customers. Several policies could be used:

- The provider may buy services from other providers to meet the customer‘s requirements established in the SLA. In this scenario, if a new task arrives, the scheduler must analyse whether it is more efficient to allocate the task to other cloud providers or reallocate existing tasks on external resources.
- The provider could invest in additional computational resources.
- Redistribute own computational resources from other services to increase cloud service capability. The main idea is to set a cloud resource admissibility threshold, and dynamically adapt it to cope with different objective preferences, and workload properties.

The multi objective nature of the scheduling problem in clouds makes it difficult to solve. In different cases, we have to consider different performance criteria such as response time, number of deadline violations, resource cost, operational cost, income, energy consumption, etc.

Federated cloud unifies different cloud resources and computing capability, even if they are owned by different organizations, to overcome resource limitations of each cloud, and to enable an unlimited computing capability.

Due to this technology, enterprises can choose an on-demand computing environment.

Cloud federation allows reducing an importance of known Build-To-Peak approach that means building infrastructures for top demands with over-provisioning in total operating time. The infrastructure can be scalable up to the certain limit but not for peak requirements, when resources of other providers can be used. In this scenario, it is important to find a trade-off that allows reducing the total investment and operational cost. Providers must propose efficient resource allocation strategies to guarantee QoS based on the SLA with customers, and make an intelligent decision on outsourcing requests to other providers.

On the other hand, providers should propose efficient cost minimization strategies and optimize different operating expenses that will be generated during the execution of a given workload due to most of the resource allocation strategies today

are non-provider-pricing-based. To this end, we have to look at the total costs in details to discover how much expense is incurred in different usage categories.

In this study, we consider four categories:

1. The cost of resources that are turned off.
2. The cost of resources that are turned on but not used.
3. The cost of resources that are in use.
4. The cost of resources of other providers in the federated cloud.

The first category includes capital costs on hardware/software, upgrades, etc. The second one includes running and maintenance costs, electricity billing, etc. The third category includes additional expenses for extra power consumption of loaded processors, extra cooling, for using hard disks, memory, databases, repairing, etc. The last one includes costs of the use of resources of other providers when local provider requests during peak demands.

Taking a resource offline does not mean shutting off services, and the operating spends for unused or under-utilized resources.

Cloud providers need to control amount of resources to avoid overprovisioning and increase capital costs. Also they need to optimize provisioning local and external resources requested by own customers and by other providers, so they can stay within budgets.

They need also to be able to manage their pricing plans to cover costs and meet profitability targets. This issue is not addressed here.

III. Problem definition

We address cost minimization problem in the hierarchical federated cloud environment, where independent clouds of different providers collaborate to be able to fulfill requests during peak demands and negotiate the use of idle resources with other peers.

A. Infrastructure model

Cloud computing infrastructure model for resource management typically assumes a homogeneous collection of hardware in one data center. Here, we extend this model to provide a cloud-based access to several data centers of one provider, which contain heterogeneous architectures, with different number of cores, execution speed, energy efficiency, amount of memory, bandwidth, operational costs, etc.

Let us consider that the cloud C consists of m nodes (data centers, sites) $D_1, D_2, \ldots, D_m$. Each node D_i, for all i=1..m, consists of b_i servers (blades, boards) and p_i processors per board. We assume that processors in the data center are identical and have the same number of cores. Let m_i be the number of identical cores of one processor in D_i. We denote the total number of cores belonging to the data center D_i by $m_i = b_i \cdot p_i \cdot m_i$, and belonging to all data centers of the cloud C by $m = \sum_{i=1} m_i$. The processor of data center D_i is described by a tuple $\{m_i, s_i, mem_i, band_i, eff_i\}$, where s_i is a measure of instruction execution speed (MIPS), mem_i is the amount of memory (MB), $band_i$ is the available bandwidth (Mbps), and eff_i is energy efficiency (MIPS per watt). We assume that data centers have enough resources to execute any job but their resources are limited.

In addition, to satisfy requests during the peak demands that exceed the capacity of the cloud C, it collaborates with k external independent clouds

(sites) $C_1,C_2,\ldots,C_k$. Each cloud C_i is characterized by the given price per time unit of the allocated instances on a pay-as-you-go basis $q_1,q_2,\ldots,q_k$.

$$\overline{q} = \overline{q_m}$$

B. Cost model

In cloud computing, a critical goal is to minimize the cost of providing the service. In particular, this also means minimizing energy consumption and maximizing resource utilization. In this section, we first present the cost model of running workloads in the cloud C. Then we define the cost model of cloud federation.

As we mentioned in Section 2, we look at the total costs in details to discover how much expense is incurred in different usage categories.

In this study, we consider three costs of resources: turned off (off); turned on but not used (idle), in use (used).

Let $q\_off_i^{core}$, $q\_idle_i^{core}$, $q\_used_i^{core}$, $q\_off_i^{proc}$, $q\_idle_i^{proc}$, $q\_off_i^{server}$, $q\_idle_i^{server}$, $q\_off_i^{site}$, $q\_idle_i^{site}$ be the operational costs (prices) per time unit of resources (core, processor, server, data center) when they are turned off (standby mode), turned on, but not used, and in use, respectively.

Let q_i be a price per time unit of the request from local cloud C to cloud C_i to use external resources.

The operational cost of a core at time t consists of a constant part $q\_off_i^{core}$ (cost in the off state) and two variable parts $q\_idle_i^{core}$, and $q\_used_i^{core}$: $q_i^{core}(t)=q_{off_i}^{core} + o_i(t)*(q_{off_i}^{core} + w_i(t)*q\_used_i^{core})$, where $o_i(t)=1$, if the core is on at time t, otherwise, $o_i(t)=0$, and if the core is in operational state at time t, $w_i(t)=1$, otherwise $w_i(t)=0$.

When a core is off, it has an operational cost $q\_off_i^{core}$; when it is on, it has extra cost $q\_idle_i^{core}$, even if it is not performing computations. Therefore, the model assumes that cost of all system components has a constant part regardless of the machine activity. Hence, core in the idle state includes the cost of the core and extra costs related with power consumption, cooling, and maintenance cost. In addition, the core has extra cost $q\_used_i^{core}$, when the core is loaded (in operational mode).

The operational cost of all cores in the processor is

$$q^{cores_{proc}}(t) = \sum_{i=1}^{mi} q_i^{core}(t)$$

The operational cost $q_i^{proc}(t)$ of processor at time t consists of a constant part $q\_off_i^{proc}$ (cost in the off state) and one variable parts $q\_idle_i^{proc}$:

$$q_i^{proc}(t) = q_{off_i}^{proc} + o_i(t) * \left(q\_idle_i^{proc} + q^{cores}(t)\right)$$

where $o_i(t) = 1$, if the processor is on at time t, otherwise, $o_i(t) = 0$.

The operational cost of processors in a server is

$$q^{proc}_{server}(t) = \sum^{pi} q^{proc}_{i}(t)$$

The operational cost $q_i^{server}(t)$ of a server at time t consists of a constant part $q\_off_i^{server}$ (cost in the off state) and one variable parts $q\_idle_i^{server}$:

$$q_i^{server}(t) = q_{off\,i}^{server} + o_i(t) * \left(q\_idle_i^{server} + q^{cores}_{server}(t)\right)$$

where $o^i(t)=1$, if the server is on at time t, otherwise, $o^i(t)=0$.

The operational cost of the servers in the site is

$$q^{server}_{site}(t) = \sum^{pi} q_i^{server}(t)$$

The operational cost $q_i^{site}(t)$ of a site at time t consists of a constant part $q\_off_i^{site}$ (cost in the off state) and one variable parts $q\_idle_i^{site}$:

$$q_i^{site}(t) = q_{off\,i}^{site} + o_i(t) * \left(q\_idle_i^{site} + q^{server}(t)\right)$$

Total operational cost of the site D_i^{site} is

$$Q_i = \sum^{Cmax} q_i^{site}(t)$$

The total cloud C operational cost

$$Q^{cloud} = \sum^{m1} Q_i$$

In addition, we consider costs associated with using resources from other providers.

The cost of execution of n_i jobs in cloud C_i is

$$Q_i^{extcloud} = \sum^{n_i} q_i \cdot p_j$$

where q_i is a price per time unit in cloud C_i, and p_j is the job execution time.

The total cost Q^{fed} is calculated as follows

$$Q^{fed} = \sum^{k} Q_i^{extcloud}$$

The total cost that will be generated during the execution of a given workload is defined as

$$Q = Q^{cloud} + Q^{fed}$$

In this paper, in order to evaluate the provider's expenses we consider total operational cost Q criterion. This metric allows the provider to measure the system performance in terms of parameters that helps him to establish utility margins.

In this paper, we consider only the part of the total cost optimization problem assuming given infrastructure.

We do not consider capacity planning for clouds that address the finding near optimal size of the cloud (best size/configuration of the infrastructure) that minimize total investment and operational costs.

C. Job model

We consider n independent jobs $J_1, J_2, \ldots, J_n$ that must be scheduled on federation of clouds [3]. The job J_j is described by a tuple $J_j=(r_j, p_j, d_j, SL_j)$, where r_j is the released time, p_j is the processing time of the job, SL_j is the SLA from a set $SL=\{SL_1, SL_2, \ldots, SL_j, \ldots, SL_k,\}$ offered by the provider [4]. Each SLA represents a SL guarantee, and d_j is the deadline. The release time of a job is not available before the job is submitted, and its processing time is unknown until the job has completed its execution.

Due to the virtualization technique and resource sharing the resources are constantly changed, which causes uncertainty in the assignation of the jobs. A job can be allocated to one cloud only, replication of jobs is not considered. Jobs submitted to the one cloud can be migrated to another one.

The admissible set of data centers for a job J_j is defined as a set of indexes $\{a^j_1, \ldots, a^j_1\}$ of data centers that can be used for allocation of the job J_i.

D. Allocation strategies

To design cost efficient allocation strategies we have to answer several important questions: How much allocation strategies can minimize the operation cost of the provider? How intelligent decisions on outsourcing requests or renting

resources to other providers could be made in the context of multiple IaaS providers? Which additional metrics are useful to describe and analyse trade-off between minimization of the provider expenses and efficiency of the service providing? What type of performance guarantees can be secured when the scheduler is based on the provider cost model? How a balance between under-provision and over-provision be obtained in cloud computing? What changes have to be made to known resource allocation algorithms to use them in cloud environment? How cost-aware scheduling algorithms can change the model complexity?

Table 1 shows the summary of cost-aware allocation strategies.

Table 1.
Cost-Aware Allocation Strategies

Strategy	Description
Random	Randomly allocates jobs to the admissible cloud
MQp -(Min Cost per processor)	Allocates job j to the cloud with the least cost of the cloud per processor at time r_j: $min_{i=1..m}\left(\frac{Q_i}{m_i}\right)$. The motivation behind this strategy is to balance the cost between clouds.
MinQ (Min Cost)	Allocates job j to the cloud with the least cost for the job
LBal_Q (Load Balance Cost)	Allocates job j to the cloud with the least standard deviation of the cost per processor (taking into account all clouds) when job j is assigned to it. $\min_{q=1..m}\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(Q_i^q-\bar{Q}\right)^2}$, where $Q_{i=1..m}^q = \frac{1}{m}\sum_{g_k=i}(q_k + q_i^q)$ and q_i^q is the cost of job j

IV. Related work.

Several works have addressed operational cost reduction in grid and cloud computing environments. Most of these works evaluated user operational cost, while few of them have considered a provider cost optimization.

CESH-Cost-Efficient Scheduling Heuristics. In [5] authors propose a set of heuristics to cost-efficiently schedule deadline-constrained computational applications on both public cloud providers and private infrastructure. They focus on the optimization problem of allocating resources from both a private cloud and multiple public cloud providers in a cost-optimal manner, with support for application-level quality of service constraints such as minimal throughput or completion deadlines for the application's execution.

FVPM-Federated VEE Placement Optimization Meta-Problem [6]. Authors address efficient provisioning of elastic cloud services with a federated approach, where cloud providers can subcontract workloads among each other to meet peaks in demand without costly overprovisioning. They define a novel Federated Virtual Execution Environments Placement Optimization Meta-Problem (FVPM), where each cloud autonomously maximizes the utility of Virtual Execution Environment (VEE) placement using both the local capacity and remote resources available through framework agreements, and present integer linear program formulations for two local VEE placement optimization policies: power conservation and load balancing.

CMHC- Cost Minimization on Hybrid Cloud [7]. Authors address the problem of task planning on multiple clouds formulated as a mixed integer nonlinear programming problem. The optimization criterion is the total cost, under deadline constraints. Their model assumes multiple heterogeneous compute and storage cloud providers, such as Amazon, Rackspace, ElasticHosts, and a private cloud, parameterized by costs and performance. Results show that the total cost grows slowly for long deadlines, since it is possible to use resources from a private cloud. However, for short deadlines

it is necessary to use the instances from public clouds, starting from the ones with best price/ performance ratio.

Table 2.
Areas Of Algorithms Application

	Algorithm	Data centers	Cloud Computing	Cloud federation	Net data storage	Two level (tier)
CESH	Cost-Efficient Scheduling Heuristics	•		•		•
FVPM	Federated VEE Placement Optimization Meta-Problem	•		•		•
CMHC	Cost Minimization on Hybrid Cloud			•	•	•
AMAW	Autonomic Management of application Workflow			•		•
TBPC	Tradeoffs between Profit and Customer Satisfaction	•		•		•
SSSC	Six Scheduling Strategies on Clouds	•	•			•
RAGF	Resource Allocation Game in Federated Cloud			•		•

Table 3.
Areas Of Algorithms Application

	Centralized	Decentralized	Hybrid	On line	Off line	Clairvoyant	Nonclairvoyant	QoS	Migration cost	Network delay	Static	Dynamic	Migration
CESH	•			•		•		•		•		•	
FVPM		•			•	•		•	•		•		•
CMHC			•		•	•		•	•	•	•		•
AMAW			•		•		•					•	
TBPC	•			•		•		•		•		•	
SSSC			•		•	•		•				•	
RAGF	•		•	•			•	•				•	

Table 4.
Evaluation Criteria

	Utilization	Performance	Response time	Scalability	Provider Cost	User Cost	Throughput	Energy	Deadline
CESH					•		•		•
FVPM	•				•			•	
CMHC		•		•		•			•
AMAW		•		•		•			•
TBPC			•		•				
SSSC		•	•			•			•
RAGF			•			•			•

V. Conclusion

We address the problem of the resource allocation on cloud computing with provider cost-efficiency in the context of the cloud federation, considering QoS.

We present cost model and discuss several cost optimization algorithms in distributed computer environments.

A fundamental design decision in the cloud is how many servers in one data center and how many data centers are optimal. In the future work, we apply our cost model to optimize capacity planning for clouds and find investment cost and operational cost trade off. This is difficult problem to answer even when simple cloud architecture is considered.

References

1. **Greenberg, A., Hamilton, J., Maltz, D. A., & Patel, P. (2008).** "The cost of a cloud: research problems in data center networks," SIGCOMM Comput Commun Rev, vol. 39, no. 1, pp. 68–73.

2. **Graham, R. L., Lawler, E. L., Lenstra, J. K. & Rinnooy Kan, A. H. G. (1979).** "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey," in Annals of Discrete Mathematics, vol. Volume 5, E. L. J. and B. H. K. P.L. Hammer, Ed. Elsevier, pp. 287–326.
3. **Schwiegelshohn, U. & Tchernykh A. (2012)**. "Online Scheduling for Cloud Computing and Different Service Levels," in Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International, pp. 1067–1074.
4. **Barquet, A. L., Tchernykh, A. & Yahyapour R. (2013).** "Performance Evaluation of Infrastructure as Service Clouds with SLA Constraints," Comput. Sist., vol. 17, no. 3, pp. 401–411.
5. **Van den Bossche, R., Vanmechelen, K., & Broeckhove, J. (2011)**. "Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds," in 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pp. 320–327.
6. *Breitgand, D., Marashini, A., & Tordsson, J. (2011).* "Policy-Driven Service Placement Optimization in Federated Clouds," IBM Haifa labs technical report h-0299 .
7. **Malawski, M., Figiela, K., & Nabrzyski, J. (2013).** "Cost minimization for computational applications on hybrid cloud infrastructures," Future Gener Comput Syst, vol. 29, no. 7, pp. 1786–1794.
8. **Kim, H., El-Khamra, Y., Rodero, I., Jha, S., & Parashar, M. (2011)**. "Autonomic management of application workflows on hybrid computing infrastructure," Sci Program, vol. 19, no. 2–3, pp. 75–89.
9. **Chen, J., Wang, C., Zhou, B. B., Sun, L., Lee, Y. C., & Zomaya, A. Y. (2011)**. "Tradeoffs Between Profit and Customer Satisfaction for Service Provisioning in the Cloud," in Proceedings of the 20th international symposium on High performance distributed computing, New York, NY, USA, pp. 229–238.
10. **De Assunção, M. D., Di Costanzo, A., & Buyya, R. (2009).** "Evaluating the CostBenefit of Using Cloud Computing to Extend the Capacity of Clusters," in In Proceedings of the International Symposium on High Performance Distributed Computing HPDC 2009, pp. 11–13.
11. **Xu, X., Yu, H., & Cong, X. (2013)**. "A QoS-Constrained Resource Allocation Game in Federated Cloud," in 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 268–275.

Andrei Tchernykh is a researcher in the Computer Science Department, CICESE Research Center, Ensenada, Baja California, Mexico. From 1975 to 1990 he was with the Institute of Precise Mechanics and Computer Technology of the Russian Academy of Sciences (Moscow, Russia). He received his Ph.D. in Computer Science in 1986. In CICESE, he is a coordinator of the Parallel Computing Laboratory. He is a current member of the National System of Researchers of Mexico (SNI), Level II. He leads a number of national and international research projects. He is active in grid-cloud research with a focus on resource and energy optimization. He served as a program committee member of several professional conferences and a general co-chair for International Conferences on Parallel Computing Systems. His main interests include scheduling, load balancing, adaptive resource allocation, scalable energy-aware algorithms, green grid and cloud computing, eco-friendly P2P scheduling, multi-objective optimization, scheduling in real time systems, computational intelligence, heuristics and meta-heuristic, and incomplete information processing.

Ramin Yahyapour is the executive director of the GWDG University of Göttingen. He has done research on Clouds, Grid and Service-oriented Infrastructures for several years. His research interest lies in resource management. He is a steering group member and on the Board of Directors in the Open Grid Forum. He has participated in several national and European research projects. Also, he is a scientific coordinator of the FP7 IP SLA@SOI and was a steering group member in the CoreGRID Network of Excellence.

Fermin Alberto Armenta-Cano received a bachelor's degree in Electronics Engineering from Technological Institute of Sonora, Mexico in 2005. He earned the M.S. degree in Computer Sciences from CICESE Research Center Ensenada, Baja California, Mexico in 2011. Actually he is a Ph.D. student working on distributed computing and Cloud computing scheduling problems.

Jarek Nabrzyski is the director of the University of Notre Dame's Center for Research Computing. Before that he led the Louisiana State University's Center for Computation and Technology, and earlier he was the scientific applications department manager at Poznan Supercomputing and Networking Center (PSNC). His research interests are resource management and scheduling in distributed and cloud computing systems, decision support for global health and environment and scientific computing. Nabrzyski was involved in more than twenty EC funded projects, including GridLab, CrossGrid, CoreGrid, GridCoord, QoSCoSGrid, Inteligrid and ACGT. Within his last five years at Notre Dame Nabrzyski has been focused on building the Center for Research Computing, a unique, interdisciplinary environment, where strong groups of computational and computer scientists and research programmers work side by side with

Model of Video on Demand Service Provisioning on Multiple Third Party Cloud Storage Services

Carlos Barba Jimenez and Raul V. Ramirez Velarde *are with Computer Science Department, Monterrey Institute of Technology and Higher Education, ITESM, Monterrey, NL, México (e-mail: a00613412@itesm.mx , rramirez@item.mx).*
A. Tchernykh is with Computer Science Department, CICESE Research Center, Ensenada, B.C., México (e-mail: chernykh@cicese.mx).

Abstract

In this paper, we present a solution model to tackle the problem of providing Video-on-Demand (VoD) using cloud computing storage service composition. We present related works, the problem motivation and some preliminary results. As part of the problem, we study the performance and scalability model for this VoD cloud service by performing a statistical analysis and Principal Component Analysis (PCA) on real cloud data. In order to simplify the stochastic modeling, we created a Characteristic Cloud Delay Time trace (using PCA), and determined the self-similarity nature of the data itself to pursue modeling using heavy-tailed probability distributions.

Index Terms Video on Demand, Cloud Computing, Service Composition, heavy-tails, PCA.

I. Introduction

The cloud is a term that has become increasingly popular in IT and consumer trends, generating buzz. Each year we see new products coming out that take advantage of it, are based on it, or even named after them. The scope of the concept has reached enterprises, journalism, research, software development, and business models; as well as the outlook for technology products. Commercially we have seen the rise of new IT services described as the cloud or cloud computing, with companies like Amazon offering it as elastic computing and storage. These services have increased interoperability, usability and reduced the cost of computation, application hosting, and content storage and delivery by several orders of magnitude [3]. These conditions open the door to creating new services that satisfy existing and future user's demands.

The trends on IP traffic are changing constantly, posing new challenges to deliver, as the amount of newer mobile internet enabled devices increases and content consumption rises [2]. Cisco [1] reports that:

- Global IP traffic has increased eightfold over the past 5 years, and will increase fourfold over the next 5 years;
- In 2015, the gigabyte equivalent of all movies ever made will cross global IP networks every 5 minutes;
- Internet video is now 40 percent of consumer Internet traffic, and will reach 62 percent by the end of 2015, not including the amount of video exchanged through P2P file sharing

- Video-on-demand traffic will triple by 2015

Content services, like Video-on-Demand (VoD), are usually supported by a centralized delivery architecture based on private or rented severs with fixed costs and little flexibility [9]. However, facing the problem of performance bottlenecks caused by an unpredictable usage of the service, they had to adopt a Content Distribution Network (CDN) model, usually operated by another company [9]. CDNs have a central entity that can enforce Quality of Service (QoS) levels, but this comes at a non-negligible financial cost [2].

There are also Peer-to-Peer (P2P) alternatives, some even free, but they can rarely provide guaranteed services [2]. There have been complimentary or hybrid solutions, like [4] or [5].

They are used in live streaming, where they help with flash crowds, in events, where certain content needs to be accessed at the same time by several users [5], which is no always the case in a VoD service. As of now, the usual CDN centralized model is still the most prevalent, with services like Akamai [6].

CDN providers are mostly priced out of reach for small to medium enterprises (SMEs), smaller government agencies, universities and charities [6]. A very small scale business could in theory use a private server or contract space in a hosting service. However, it is difficult to size them properly initially. They are either expensive, because of an overprovision, or they are not robust enough and will have problems under heavy demand. They could be under the Slashdot effect or some other popularity phenomenon.

There are other options like third party services (e.g. Youtube, Vimeo, etc.), but content rights management and QoS are not their priorities, even if they have big infrastructures behind them. This situation could not be suitable for all businesses.

To help us to cope with some of these factors, we go back to the concept of the cloud and different available services. There are new solutions commercially available that take the characteristics and advantages of the cloud, like the one proposed in [6] with the MetaCDN project. The authors published some of the characteristics of this model, but left out the proprietary code and algorithms that make it work. Their statistical analysis is also limited, if we are to generate a predictive model.

In this paper, we use different cloud services available in composition to create an Edge Cloud Service (similar to MetaCDN [6]), but in a fashion as [8] describes it: a gateway. This gateway will create a CDN like functionality sitting in a layer atop the cloud, and will be taking into account some of the main challenges of VoD delivery like response time.

The idea is to tackle a similar problem motivation and approach as established in [6]. However, the objective of our work is not to solve the programmatic problem, but rather tackle the mathematical performance predictive model for the final service. This model has to determine maximum number of clients that can be served under certain cloud conditions and a fixed maximum delay time. In addition, it has to handle request redirections to different cloud

providers, under different conditions keeping a given QoS. These aspects were not fully explained and explored in [6].

As a metric for the meta VoD service, we take mainly into account the times for the abandonment rate described in [20], which found that VoD viewers start leaving after a startup delay of 2000ms (milliseconds) losing 5.8% of users for

each additional second of is total, it includes all network times, server times, and additional overhead.

This paper proposes a scalability model and analysis using self-similarity and heavy tails similar to ones used in [15][19]. We start it by characterizing real cloud data using statistical analysis, and dimensionality reduction using Principal Component Analysis. The , which will enable us to make predictions for the probability of successful service under a certain threshold of abandonment rate.

II. Cloud Computing

The cloud is defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort and service provider interaction [7].

Cloud computing offers three service models according to [7]: SaaS, PaaS, IaaS.

Cloud Software as a Service (SaaS): The capability provided to the consumer is to use the providers applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Cloud Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.

Cloud Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

We focus on the storage part of Cloud Infrastructure as a Service (IaaS), where VoD is potentially a very heavy user. Cloud storage providers offer service level agreements (SLAs), which guarantee a level of Quality of Service (QoS) and operate on a utility computing model. This model is defined in [7] as the metering capability for a pay-per-use-basis. Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service, as explored in [14]. This situation makes the idea of using the existing cloud services for content distribution very attractive, as we will only pay for the amount of storage and computing we use, instead of a potentially expensive contract with a CDN or an under provisioned private server.

III. Related Works

The basic idea we want to address is the video CDN based on the cloud as explained in [6]. One of the most interesting figures it presents is the price comparison of delivering content through a CDN versus different cloud providers (Figure 1). It also has interesting points related to QoS provisions, but the model in [6] is still a little broad for a proper mathematical definition.

On the topic of CDN and modelling, the authors in [10] explore resource discovery and request redirection in a multi provider content delivery network environment.

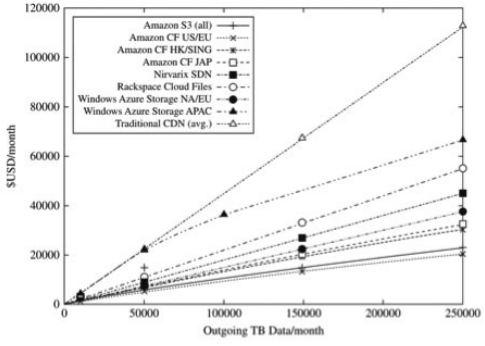


Fig. 1. CDN and cloud Costs [6]

They explain that CDNs evolved as a solution for internet service degradations and bottlenecks due to large user demands to certain web services. In addition, they address some of the internal problems that CDN providers face, like the required break down of system locations, the need to increase utilization rates and the over provisioning and external resource harnessing need that can happen to satisfy a certain SLA. The proposition is a constellation of CDNs that collaborate for short or long term periods to handle the different workload situations. This constellation uses a load distribution scheme with a request redirection approach where there is a mediator or coordinator agent that redirects load according to the state of the different CDNs [10].

Continuing on the topic of cloud computing service aggregations, the authors in [11] describe a QoS aware composition method supporting cross-platform service invocation in cloud environments, which explores a web service composition and how to attain a QoS optimal solution.

There have also been more recent developments where a coordinator for scaling elastic applications across multiple clouds is used, as proposed in [12]. The main focus is in elastic web applications where the application characteristics and usage in the cloud providers are better known (although provisioning is still a challenge).

The authors in [12] don't address much on content delivery, like in a VoD service, and the application characteristics of a web application and content storage in the cloud are not the same. However, it does give ideas to take into consideration for the work of this research. This paper also takes into consideration previous work related to VoD specific topics like the specification of an integrated QoS model for VoD applications found in [13].

IV. Solution Model And Preliminary Results

Given that we are developing a CDN-like service, it is fitting to use similar techniques to the ones that this type of systems use and apply them to Cloud Environments. The model stems from the basis of peering CDNs where several CDN networks cooperate to fulfil requests. In this case the CDNs will be the different clouds available, but without the concepts of authoritative rights.

Our model works by sitting on a layer atop the

different Cloud Storage providers (SaaS) and using IaaS for some of the calculations and computing required. The solution model to the problem consists of the following main components:

- Asynchronous Resource Discovery: Since each Cloud providers operates individually this module keeps track of different points of presence and resources available.
- Asynchronous Resource Monitoring: This nodule takes care of probing the different providers to keep historic information about different performance parameters (this are explained with more detail later).
- QoS Aware Resource Redirector: This agent is the one in charge of weighting in the information available from the clouds and redirecting the load to the optimal resource location and provider..

Figure 2 shows the basics of the solution model, with all the principal elements of the gateway and how most of the data and connections flow. User requests go through the gateway, which has the redirection and resource monitoring logic inside, then that gateway gives back an address redirection for the final content in one of the several cloud providers. In this model the several cloud providers and their infrastructure are working in a similar manner as the Content Distribution Inter networking model, in which the gateway will consider the clouds and its networks as black boxes [10].

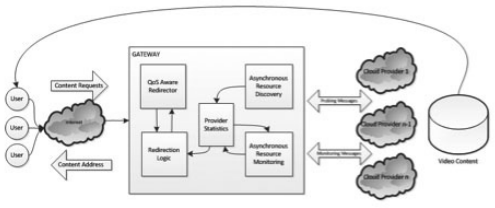


Fig. 2. Cloud VoD System

However, it differs from the main model in that there really aren't mechanisms implemented from the cloud providers that will allow peering, and there is no central entity or supervisor with access to intra cloud information.

This is one of the main challenges, working without knowing the request and response loads that are present in each of the inter cloud networks in real time; essential to work within the peering CDN scheme and algorithms [10]. In order to overcome these limitations, we draw conclusions from the information we can get.

We use the total delay times, since they can be monitored and discovered from outside the black boxes. Additionally, we make assumption based on the published requests/second statistics from some cloud providers [27]. The performance parameters that the redirector has access for the different cloud providers available are:

- Delay Time: measured in ms as captured by probing done by the monitor service.
- Request Origin/Destination similarity: Established in a [0,1] scale where 1 matches the origin of the request to an available destination cloud available and progressively gets reduced as the distance difference between both is greater.
- Rejection rate: number of dropped requests due to service unavailability or timeouts as captured by the monitor service.
- Completion rate: number of completed services that each of the providers have on record.
- Provider QoS: the QoS as stated by the service level agreement that each provider gives.

- Mean Peak requests/second and the percentage that VoD type requests represent.

We need to be able determine how many users the service can handle under a maximum delay time, taking into consideration the abandonment rate times and data from [20]. This will be the scalability model.

Once we can make predictions about users and system scalability, the problem that we must solve inside the redirector logic is to minimize the delay time that the end user has when connecting to the final cloud provider, taking into consideration the redirection time caused by the decision making time in the gateway. We must also add the wildcard variables of the downtime status and QoS levels each provider has, as a continually changing modifier. Finally we evaluate them all under the values and statement of abandonment rate and time described by [20] and different cloud load scenarios using the requests/second information.

To begin, we define a basic model where the delay time T_{Dij} that the user i needs to access the content in a certain provider j is determined by:

$$T_{Dij}=T_{rij}+T_{nij}+T_{Og} \qquad (1)$$

T_{rij} is the response time from the cloud storage (disk access, memory and computations needed), T_{og} is the delay time in the gateway caused by the overhead of redirection, and T_{nij} is the network time the request takes to travel from client i to cloud j. From here, we follow a simple scheme similar to [10], minimizing the redirection time and cost, using the following:

$$min\sum^{n}_{i=1}\sum^{m}_{j=0} R_c\ (i,j)\ S_{ij}\ C_j\ Q_j \qquad (2)$$

In this case $R_c = T_{Dij}$ if the cloud provider is not rejecting requests at that moment, else it is ∞. S_{ij} can take a value between [0,1] according to how close the locations from user and cloud provider are together geographically, C_j is the completion rate of the provider j and can take a value between [0,1], and finally Q_j is the quality of service that provider j has and can take a value between [0,1].

However, this solution model (2) would be naïve, since it would in theory base its results in the most recent data from the different clouds available. It does not consider the stochastic nature of the response and network times; its probability distribution and chaos measurements.

Thus, considering the probabilistic nature of T_D, and taking into account [20] and the 2s abandonment start time θ as a measure of a successfully redirected petition, we could create a new model for a client i connected to m different clouds using the following premise:

$$P(T_D<\theta)<\phi \qquad (3)$$

ϕ is the probability that the delay time is under θ or 2s (in our case). Using (3) as a base and assuming T_D is heavy-tailed we could model it, using one of the Pareto, Lognormal or Weibull distributions. However, we first must do a statistical analysis of different real life cloud delay times, to properly characterize them.

A. Cloud Data Analysis

In order to have a good estimation and idea of the extreme values and distribution that the clouds delays can exhibit, we used real cloud delay times. From now on, when we talk about Delay Time we only refer to $T_{nij}+T_{rij}$, since only the network times and cloud response times interest us to model from the cloud data.

Real cloud delay times were obtained using the PRTG Network Monitor from Paessler [26]. The measurements are taken from an Amazon EC2 instance (small) located in N. Virginia, as an analog for a client, which then connects to different cloud providers and gathers the delay time (encompassing, disk access times, memory access times, and network times) to get a 65kb file (representing a worst case video frame size). The measurements were taken every 60s over a period of 40 days leaving us with over 52,000 measurements for 6 cloud storage providers. In Table 1, we see different clouds and some descriptive statistics:

Table I. Cloud Providers Statistics

Cloud Provider	Mean Delay (ms)	Max Delay (ms)	Kurtosis
Google	65.74	3425	170
GoGrid	22.88	11768	959.6
Rackspace	118.17	8790	1168
CloudFront	37.26	3001	152.4
S3 USA	188.24	27059	2725.9
S3 EU	652.48	22653	386.66

In Figure 3, we see an example time series, which we call trace, for the Google cloud.

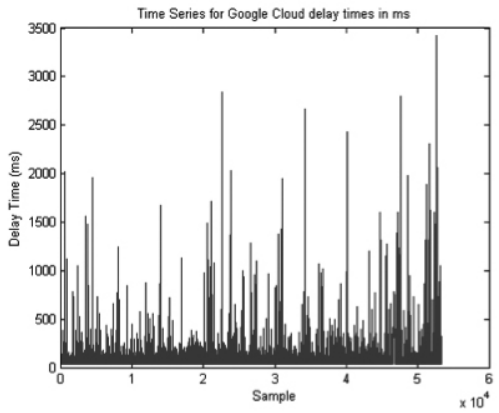


Fig. 3. Time Series for Google Delay Times

We clearly see the chaotic behavior of the delay times; however we still need to look at the distribution of the values. Figures 4 and 5 have the histograms for each of the clouds.

Figures 4 and 5 show that the delay times for the different cloud provider samples exhibit a heavy-tailed distribution which is corroborated in combination with the high kurtosis values presented in Table 1.

Moreover, if we take into consideration that the histograms presented are not symmetrical and the high variability in the data, (with max delay values several times the mean value), we conclude in concordance with [15] that the data cannot be modeled by a symmetric probability such as a Gaussian or a short-term memory process like a Markov one. This would be a good sign if a scalability analysis similar to [19][15], shows that the data are self-similar.

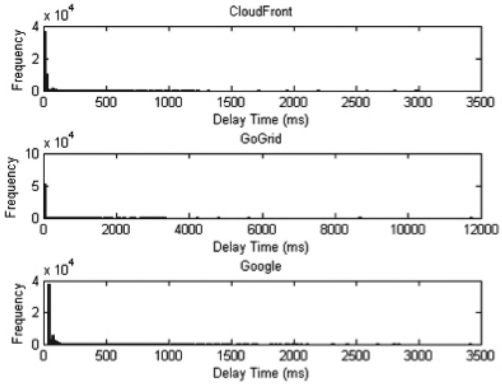


Fig. 4. Cloud Histograms part 1

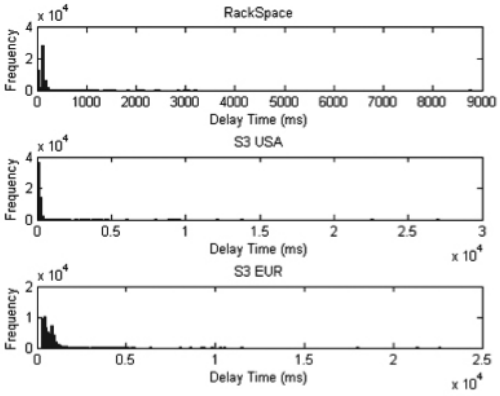


Fig. 5. Cloud Histograms part 2

The problem is now converted to finding a cloud to model the delay. Clouds have similar statistics and histograms, but some have different kurtosis and different means, etc.

In this case, we treat the existence of several different clouds as a problem of multidimensionality [19]. We consider each cloud as a dimension for the total delay time. Therefore, we use Principal Component Analysis (PCA) to determine a Characteristic Cloud Delay Trace (CCDT), as in [19][16] to capture as much of the variability of all clouds as possible. The goals is to reduce dimensionality to at most 2 components, from our available 6, and then regenerate a single trace that can be analyzed further and used for modeling.

To find the Principal Components we followed a procedure similar to [16][19], and built a correlation matrix using the delay traces for each cloud. Traces are used as columns to indicate the dimensions with ~50k different observations. The procedure then subtracts the overall mean μ from each element. We used the mean from all values, not in a per cloud basis (180.83 ms), and then used Pearson's correlation coefficient, since all the observations are in the same ms units [19].

The results for the PCA (for the first 3 components) are visualized in Figure 6, where we show the orthonormal principal component coefficients for each variable and the principal component scores for each observation in a single plot.

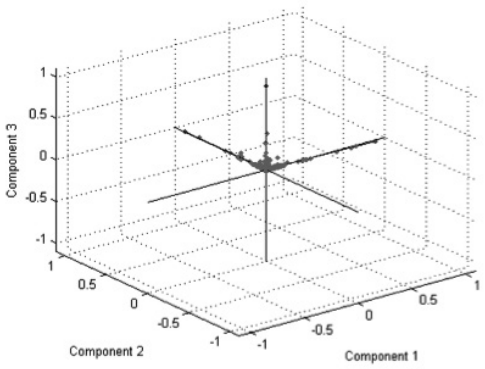


Fig.6. Component Scores

The resultant principal components (PCs) and their respective effect are shown in Table 2:

TABLE 2. PCA Results

Variables	PC1	PC2	PC3
Eigenvalue	169231.8	80962.5	30646.5
Variability(%)	54.920	26.274	9.947
Cumulative (%)	54.920	81.200	91.140

Variables	PC4	PC5	PC6
Eigenvalue	10249.8	9555.8	7490.3
Variability(%)	3.326	3.102	2.431
Cumulative (%)	94.470	97.570	100.000

Figure 7 shows the Scree and cumulative variance plots for the PCs.

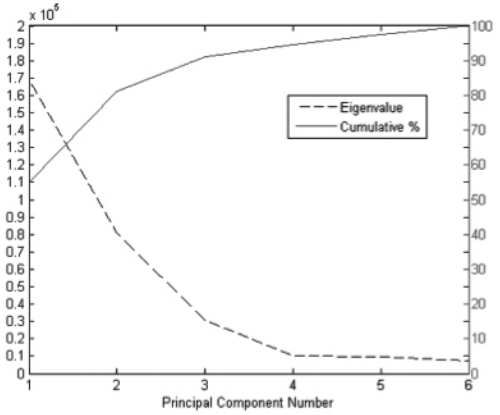


Fig. 7. PCA Scree plot and cumulative variance plots

Following [19], we have to select m number of components that will keep most of the variability present in p variables, where we want m<<p, effectively doing a dimensionality reduction. In the case of our data, we see that selecting 1 PC will give us ~50% of the original data variability, but if we go to 2 PCs we have ~81% of it. After that, the percentage of added variability of each extra PC is < than 10%.

Considering these conditions, choosing 2 PCs will give us significantly more information than the other 4, so the Characteristic Cloud Delay Trace will be generated using PC1 and PC2.

Using these PCs we reconstruct a single trace that will have the same variance that the whole set of cloud delay time traces: $\sigma^2_{original}$= 98894.582 and $\sigma_{original}$ = 314.475. The reconstruction also has to take into account that we subtracted the overall mean μ while doing the PCA. To reconstruct it, we start from the following:

$$E[PC1]=0 \quad E[PC2]=0 \qquad (4)$$

$$VAR[PC1]=\lambda_1 \quad VAR[PC2]=\lambda_2 \qquad (5)$$

Where λ_i is the eigenvalue for the corresponding PCi. We propose creating a new variable C :

$$C=PC1+PC2 \quad (6)$$

This variable C has the following expected value, variance, and standard deviation:

$$E[C]=E[PC1]+E[PC2]=0 \quad (7)$$

$$VAR[C]=VAR[PC1]+VAR[PC2]=\lambda_1+\lambda_2 \quad (8)$$

$$\sigma\_c=\sqrt{(\lambda_1+\lambda_2)} \quad (9)$$

Then we create ~~the final~~ reconstructed trace X', which must have a $\sigma^2_{x'} = \sigma^2_{original} = 98894.582$ using the following transformation, and taking into account μ:

$$X' = \mu + \frac{\sigma_{original} * c}{\lambda^1+\lambda^2} \quad (10)$$

Obtaining the expected value and variance for X' we get:

$$E[X']=E[\mu]+\left(\frac{\sigma_{original} * c}{\sqrt{\lambda^1+\lambda^2}}\right) \quad (E[C]=\mu \quad (11)$$

$$VAR[X']=VAR[\mu]+\left(\frac{\sigma_{original} * c}{\lambda^1+\lambda^2}\right) VAR[C] = \sigma_{original} \quad (12)$$

Since the standard deviation is the same in our new reconstructed trace X' as in the σ_original^2, evidenced by (12) and numerically tested, we apply the transformation to each row of the PC1+PC2 and get our Characteristic Cloud Delay Trace. We see the result in Figure 8.

Figure 8 shows that the CCDT's behavior is similar to the Google delay times in Figure 3. Figure 9 shows the histogram for the new reconstructed trace.

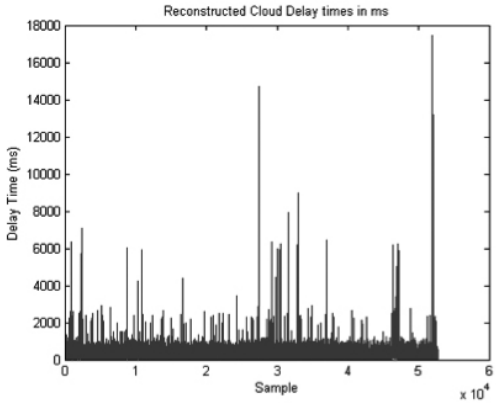


Fig. 8. Reconstructed Characteristic Cloud Delay Trace

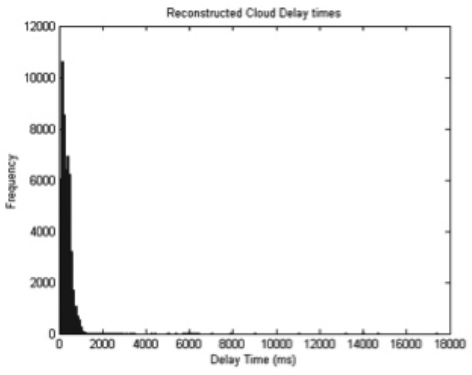


Fig. 9. Reconstructed Characteristic Cloud Delay Times Histogram

The data has a heavy-tailed distribution, which is not symmetric. The kurtosis parameter is 472.451, so in accordance to [15], we assume that the data cannot be modeled using traditional symmetric probability distributions and Markovian processes. Therefore, we use the Pareto, Lognormal and Weibull probability distributions and fit them to characterize the reconstructed data and obtain the cdf's that we see in Figure 10.

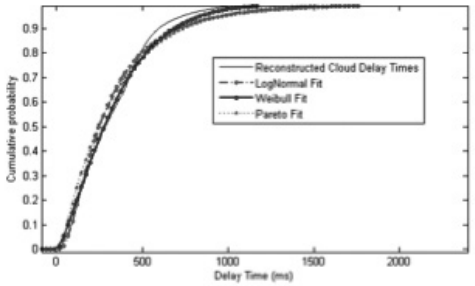


Fig. 10. Comparison of Lognormal, Pareto, and Weibull cdf against data trace

Going back to the scree plot in Figure 7, even if limited to only 6 components, it shows that there's one mayor PC and a second one that has still some considerable percentage of variation. After them, the third to sixth component exhibit a tail of slowly decaying eigenvalues with what [19] calls a rule similar to K^{-B}, power law rule. This is an indication of self-similarity.

According to [15], there are several, not equivalent, definitions for self-similarity. The most commonly accepted for flows related to networks is found in [25], it states that a continuous-time process $Y=\{Y(t),t\geq 0\}$ is self-similar, with a self-similarity or Hurst parameter H, if it satisfies the following conditions.

$$Y(t) \quad a^{-H} Y(t) \quad t \geq 0 \quad a > 0, \; 0.5<H<1 \qquad (13)$$

This self-similarity measurement is further studied by [21][22][23][24], having similar conclusions about the values of the Hurst parameter that indicate self-similarity ($0.5<H<1$). However, from [21], we also know that even if the Hurst parameter is perfectly defined mathematically, measurements and estimations using real traces is not easy. This calls for the use of several Hurst parameters estimators rather than just one, if we are to avoid false conclusions [21]. It must be said, that all estimators are vulnerable when the data has a trend and periodic component [21][24].

In this paper, we use the following methods to estimate the Hurst parameter: the R/S (Rescaled/ Range), the aggregate variance, the boxed periodogram, the Abry Veitch wavelet method [23] and Peng's variance of residuals. Table 3 summarizes results.

Table 3. Hurst Parameter Estimations

Method	Hurst Parameter
R/S	0.777
Aggvar	0.789
Box Per	0.679
Peng's	0.786
Abry-Veitch	0.680
Mean	0.742

We concluded that $0.670 <H <0.789$, which is a definite indication of self-similarity.

V. Conclusions And Future Work

The use of cloud computing services is a fundamentally new approach to solve the increased VoD and online video content demand. The main idea is to use some of the knowledge that the CDN paradigm has to offer and adapt it to the challenges and advantages of cloud computing.

We presented a solution model and its elements to create a CDN like service that exists in a layer atop of the Clouds, using storage service composition from different providers. We concluded that it can be based on a gateway redirector that accepts requests from incoming clients and gives out

the optimal cloud redirection based on a set of available metrics.We also show that this system can treat each cloud as a black box where we have very little visibility, and that in order to make a proper redirection scheme, we would need to make a proper performance and scalability model of real life cloud data.

We demonstrate that PCA is a powerful tool. It allows making a dimensionality reduction of the multiple cloud data, and generating a Characteristic Cloud Delay Time Trace. This trace can act as a proxy, as we showed that it retains the same variance as the original cloud traces, and that it contains up to 80% of the variability from all the data. This not only simplifies statistical analysis, but enables future work in the devising of the scalability and performance model.

Finally, we found that cloud delay times present a heavy-tailed probability distribution, and that they exhibit self-similarity (taking advantage of different Hurst parameter estimators to achieve the measurement). This are important results, since we can use probability distributions like the Weibull, Lognormal and Pareto to model performance and establish the request redirector gateway logic for the proposed solution model.

As a future work we want to evaluate the different probability distributions goodness of fit to either select one or two of them to generate the final scalability model. This model will then be validated through simulation, using the real cloud delay time traces and several load assumptions. Finally, further work is needed in the model to account for the stochastic nature of the delay times and to integrate the abandonment rate and times into it, as another measure of success.

References

1. Cisco Systems (2011). Cisco Visual Networking Index: Forecast And Methodology, 2010 2015. Tech. Report.
2. Passarella, A. (2012). A Survey On Content-Centric Technologies For The Current Internet: Cdn And P2p Solutions. Computer Communications, 35(1), 1 – 32.
3. Buyya, R., Broberg, J., & Goscinski, A. M. (2011). Cloud Computing Principles And Paradigms. Wiley Publishing.
4. Farinha, D., Pereira, R. & Vazao, T. (2008). Performance Analysis of VoD Architectures. In Proc. International Workshop on Traffic Management and Traffic Engineering for the Future Internet, Porto, Portugal.
5. Mansy, A. & Ammar, M. (2011). Analysis of adaptive streaming for hybrid CDN/P2P live video systems. In Network Protocols (ICNP), 2011 19th IEEE International Conference, 276-285.
6. Broberg, J., Buyya R. & Tari, Z. (2009). Metacdn: Harnessing Storage Clouds For High Performance Content Delivery. Journal of Network and Computer Applications, 32(5), 1012 – 1022, Next Generation Content Networks.
7. Mell, P. & Grance, T. (2011). The NIST Definition Of Cloud Computing. NIST Special Publication, v800, p145.
8. Islam, S. & Gregoire, J.-C. (2012). Giving Users An Edge: A Flexible Cloud Model And Its Application For Multimedia. Future Generation Computer Systems, 28(6), 823-832.
9. Buyya, R., Yeo, C., Venugopal, S., Broberg, J. & Brandic, I. (2009). Cloud Computing And Emerging It Platforms: Vision, Hype, And Reality For Delivering Computing As The 5th Utility. Future Generation computer systems, 25(6), 599–616.
10. Pathan, M. & Buyya, R. (2009). Resource

Discovery And Request-Redirection For Dynamic Load Sharing In Multi-Provider Peering Content Delivery Networks. Journal of Network and Computer Applications, 32(5), 976–990.

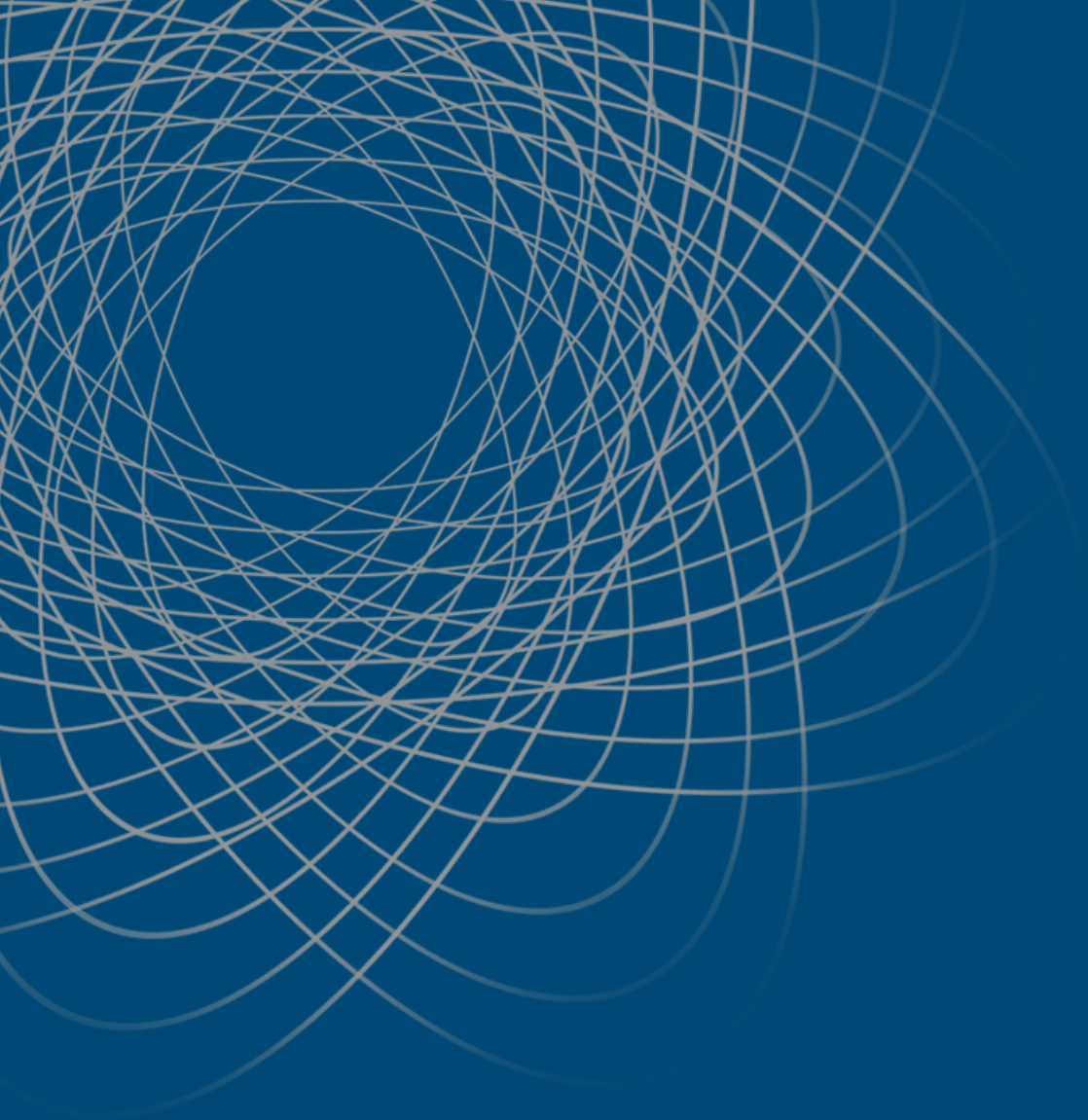
11. Qi, L., Dou, L., Zhang, X. & Chen, J. (2011). A Qos-Aware Composition Method Supporting Cross Platform Service Invocation In Cloud Environment. Journal of Computer and System Sciences, 78(5), 1316-1329.
12. Calheiros, R., Toosi, A., Vecchiola, C. & Buyya, R. (2012). A Coordinator For Scaling Elastic Applications Across Multiple Clouds. Future Generation Computer Systems, 28(8), 1350-1362.
13. Sujatha, D., Girish, K., Venugopal, K. & Patnaik, L. (2007). An Integrated Quality-Of-Service Model For Video-On-Demand Application. IAENG International Journal of Computer Science, 34(1), 1–9.
14. Espadas, J., Molina, A., Jimenez, G., Molina, M., Ramirez, R. & D. Concha (2011). A Tenant-Based Resource Allocation Model For Scaling Software-as-a-Service Applications Over Cloud Computing Infrastructures. Future Generation Computer Systems, 29(1), 273-286.
15. Ramirez-Velarde, R. & Rodriguez-Dagnino, R. (2010). From Commodity Computers To High Performance Environmets: Sacalability Analisis Using Self-Similarity, Large Deviations And Heavy Tails. Concurrency and Computation: Practice and Experience, v22, 1494-1515.
16. Ramirez-Velarde, R. & Pfeiffer-Celaya, R. C. (2009). Using Principal Component Analysis On High-Definition Video Streams To Determine Characteristic Behavior And Class Grouping For Video-On-Demand Services. Congreso Iberoamericano de Telematica, Gijon, Espana.
17. Pathan, M. & Buyya, R. (2009). Resource Discovery And Request-Redirection For Dynamic Load Sharing In Multi-Provider Peering Content Delivery Networks. Journal of Network and Computer Applications, 32(5), 976–990.
18. Hasslinger, G. & Hartleb, F. (2011). Content Delivery And Caching From A Network Providers Perspective. Computer Networks, 55(18), 3991 – 4006.
19. Ramirez-Velarde, R., Martinez-Elizalde, L. & Barba-Jimenez, C. (2013). Overcoming Uncertainty On Video-On-Demand Server Design By Using Self-Similarity And Principal Component Analysis. Procedia Computer Science, v18, 2327 – 2336, International Conference on Computational Science.
20. Krishnan, S.S. & Sitaraman, R. K. (2012). Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs. Proceedings of the 2012 ACM conference on Internet measurement conference, 211–224.
21. Clegg, R. G. (2005). A Practical Guide To Measuring The Hurst Parameter. 21st UK Performance Engineering Workshop, p. 43.
22. Lobo, A., Garca, R., Paeda, X. G., Melendi, D. & Cabrero, S. (2013). Modeling Video On Demand Services Taking Into Account Statistical Dependences In User Behavior. Simulation Modelling Practice and Theory, v31, 96 – 115.
23. Abry, P. & Veitch, D. (1998). Wavelet Analysis Of Long-Range-Dependent Traffic. IEEE Transactions on Information Theory, 44(1), 2–15.
24. Kirichenko, L., Radivilova, T. & Deineko, Z. (2011). Comparative Analysis For Estimating Of The Hurst Exponet For Stationary And Nonstationary Time Series. International Journal of Information Technologies and Knowledge, 5(1).
25. Willinger, W. & Paxson, V. (1998). Where Mathematics Meets The Internet. Notices of the AMS, 45(8), 961–970.
26. Paessler (2014). Prtg Public Demo. https://prtg.paessler.com/index.htm. Accessed:2014-01-15.
27. Barr, J. (2012). Amazon s3 - 905 billion objects and 650,000 requests/second. http://aws.typepad.com/

aws/2012/04/amazon-s3-905-billion-objects-and-650000-requestssecond.html. Accessed: 2014-01-15.

Carlos Barba Jimenez obtained his Bachelor's Degree in Electronic Systems Engineering from Tecnologico de Monterrey, Campus Monterrey in 2009. Currently he is a PhD student at the same institution. His interests include cloud computing, performance modeling, and microprocessor manufacturing, developing, and testing.

Raúl V. Ramírez Velarde is a researcher in the Computer Science Department, Tecnológico de Monterrey, Monterrey, Mexico. He is part of the Consejo Nacional de Acreditación Informática y Computación (CONAIC) and Asociación Nacional de Instituciones de Educación Tecnología de Información (ANIEI) in Mexico. He received his PhD in Distributed Multimedia Systems in 2004. His research interests include analysis, modeling and design of multimedia systems.

Andrei Tchernykh is a researcher in the Computer Science Department, CICESE Research Center, Ensenada, Baja California, Mexico. From 1975 to 1990 he was with the Institute of Precise Mechanics and Computer Technology of the Russian Academy of Sciences (Moscow, Russia). He received his Ph.D. in Computer Science in 1986. In CICESE, he is a coordinator of the Parallel Computing Laboratory. He is a current member of the National System of Researchers of Mexico (SNI), Level II. He leads a number of national and international research projects. He is active in grid-cloud research with a focus on resource and energy optimization. He served as a program committee member of several professional conferences and a general co-chair for International Conferences on Parallel Computing Systems. His main interests include scheduling, load balancing, adaptive resource allocation, scalable energy-aware algorithms, green grid and cloud computing, eco-friendly



GRIDS AND GPU´S

An Optimization Method for Lennard-Jones Clusters using Conjugate Gradient and Molecular Dynamic Simulations

José Luis Alonzo Velázquez, Sergio Ivvan Valdéz Peña, Salvador Botello Rionda
Centro de Investigación en Matemáticas, CIMAT A.C., Guanajuato, Gto., México.
E-mail: {pepe, ivvan, botello}@cimat.mx.

Abstract

The development of new technologies has reduced the computing power limits. This allows handling large volumes of data, but nowadays there are many problems that cannot be solved in a reasonable time with the actual computing power. It is necessary to have suitable optimization methods for functions in high dimensions. Several optimization methods using computers with multiple processors using OpenMP technology have been implemented. The trend of those implementations there are linked and restricted to the evolution of the computational systems. Following the trends of current computational systems has been created methods based in grids, which avoid computing all pairs of distances in problem related with short range potentials speeding up Molecular Dynamic simulations, but with a little more cost in memory. In this work have been implemented an optimization algorithm based in Conjugate Gradient(CG) and Molecular Dynamic(MD) Simulations. The results provide information to believe we have a good optimization method in the problem of optimization of Lennard-Jones clusters.

Keywords: *Molecular Dynamics, Optimization, Conjugate Gradient, Grids, OpenMP.*

1. Introduction

Computation power has increased widely in the span of about 20 years. There are many areas requiring computational speed including numerical simulation of scientific and engineering problems. Computations must be completed within a suitable time. In manufacturing realm and engineering, calculations and other simulations must be achieved within minutes or even seconds, it is the reason we need to improve optimizations methods.

Optimization algorithms are an important component of many applications. There is a wide variety of optimization methods. On the other hand, parallel algorithms have been studied extensively in the last four decades. Therefore, many of the optimization methods have a parallel version. However the parallelization itself is not enough to reach optimal results. We consider essential to develop optimizations methods for choosing a suitable one based on the specific application and the available computer power. In this paper, a method based in conjugate gradient and molecular dynamic simulations (MD) is implemented in parallel using OpenMP platform to solve the problem of optimization of Lennard-Jones clusters. Their performance for various clusters sizes is measured taking the number of iterations.

The optimization problem of Lennard-Jones clusters consist in finding a configuration of atoms that minimize the energy of a given particle system considering the pair potential of Lennard-Jones. The importance of this problem resurfaced on 90's with applications and extensions of molecular conformation research such that computational prediction of protein tertiary structure in the protein folding. In the 90's there were many algorithms enough efficient to reach a minimum for system above of 100 atoms such that basin-hopping methods [18], algorithms based in annealing evolutionary algorithms [16, 17] or LBFGS method [21]. Another kind of algorithms are based in symmetries using tetrahedral, decahedral, icosahedral clusters of atoms [15, 19, 20] that reach the minimum of bigger systems, those last algorithms permitted to explore the configurational space and find the minimum energy on system above 1600 atoms [14, 22].

In this article is considered the temperature in a cluster of particles, a physical property of materials that permits create a faster algorithm more efficient, saving time at the moment to choose a good initial point for a local method. The global method is based in cooling the system, the main difference with the annealing evolutionary algorithm based on a Monte Carlo step and a metropolis decision is not to use a random perturbation of the system in all the steps of the algorithm, only in the first step would be defined a random velocity for each particle that depends on a initial temperature, the rest of this phase is a molecular dynamic simulation with a thermostat that is cooling the system, speeding down the velocities, but the direction moves only depends of molecular interactions, avoiding many unstable positions that methods like annealing evolutionary algorithm evaluate saving computational time in our algorithm.

This paper is structured as follows. In section 2 is given a briefly explain classical gradient conjugate method and basics notions of Molecular Dynamics Simulations. In section 3 is given a description how to implement variants of both methods to get a good optimization method. The results of these implementations are presented on the section 4. Conclusions and future work are discussed on section 5.

2. Methods

A brief description of Conjugate Gradient Method is presented [13]. This problem is a nonlinear problem, thus actual method than we are talking about is the nonlinear conjugate gradient method that generalizes the conjugate gradient method to nonlinear optimization. In general the method consists in:

Given a function *f(x)* of *N* variables to minimize, its gradient $\nabla_x f$ indicates the direction of maximum increase.

We define the steepest descent direction, with $\Delta x_0 = -\nabla_x f(x_0)$ an adjustable step length α and performs a direction until it reaches the minimum of f, begin with, $x_1 = x_0 + \alpha_0 \Delta x_0$, the following steps constitute one iteration of moving along a subsequent conjugate direction s_n, where . $s_0 = \Delta x_0$. The steps to repeat are:

1. Calculate the steepest direction: $\Delta x_n = -\nabla_x f(x_n)$.
2. Compute , $\beta_n = \frac{\Delta x_n^T \Delta x_n}{\Delta x_{n-1}^T \Delta x_{n-1}}$ in my implementation a set this value in the beginning for all *n*.
3. Update the conjugate direction: $s_n = \Delta x_n + \beta_n s_{n-1}$.
4. Perform a line search: optimize $\alpha_n = arg\, min_\alpha f(x_n + \alpha s_n)$.

5. Update the position: .
$$x_{n+1} = x_n + \alpha_n s_n.$$

In general is a good method for quadratic functions, but our function is non-quadratic, and then we get a slower convergence, to accelerate this convergence will use.

Molecular Dynamics Simulations

In molecular dynamics simulations, the successive configurations of the system can be obtained from the integration of Newton's equation of motion. As the result, the trajectory presents the variations of the positions and velocities of the particles moving in the system. Newton's laws of motion can be stated as follows:

1. If one body is not influenced by any forces, it will go on moving straight in constant velocity.
2. Force equals the rate of change of momentum.

The trajectory can be obtained by solving Newton's second law

$$\frac{d^2 x_i}{dt^2} = \frac{F_i}{m_i}$$

This equation describes the motion of a particle of mass m_i along the coordinate x_i with the force F_i acting on the particle.

Under the influence of potentials, the motion of all particles are correlated which makes an intractable many-body problem. For this reason the equations of motion are integrated using a finite difference method.

Finite Difference Method

The finite difference method also called Gauss method can generate MD trajectories with continuous potential models. The essential idea is that the integration is divided into many small steps, each separated by a fixed time interval Δt. The total interactions on each particle at time can be calculated from the sum of interactions from other particles. The force is assumed to be constant during the time step t and $t + \Delta t$. . The forces and accelerations of the particles in new positions can be determined, and so on.

There are many algorithms for integrating the equations of motion using finite difference methods, which are commonly used in MD simulations. All algorithms assume that the positions and dynamics properties can be approximated in Taylor series expansions, in this particular case would be used Velocity Verlet Integration:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2$$

$$v\left(t + \frac{\Delta t}{2}\right) = v(t) + \frac{1}{2}a(t)\Delta t$$

$$a(t + \Delta t) = -\frac{1}{m}\nabla V(r(t + \Delta t))$$

$$v(t + \Delta t) = v\left(t + \frac{\Delta t}{2}\right) + \frac{1}{2}a(t + \Delta t)\Delta t$$

where v is the velocity, α is the acceleration and V is the potential. The Verlet algorithm is the most broadly used method for integrating the trajectories of motion in MD simulations. The Verlet algorithm uses the positions and accelerations at time t and the positions from the previous step to calculate the new positions. In molecular dynamics, the force calculation occupies the majority of the simulation time. In every loop, the force

computation time is proportional to the square of the particle number.

Lennard-Jones Potential

As a model system for the method of MD simulation, a system interacting with the Lennard-Jones pair potential has long been used to elucidate the fundamental properties of many body interacting particles. The standard Lennard-Jones 12-6 potential for argon is suitable for measuring all the computational efficiency of the parallel MD simulation.

The common expression for the Lennard-Jones potential is:

$$V(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right],$$

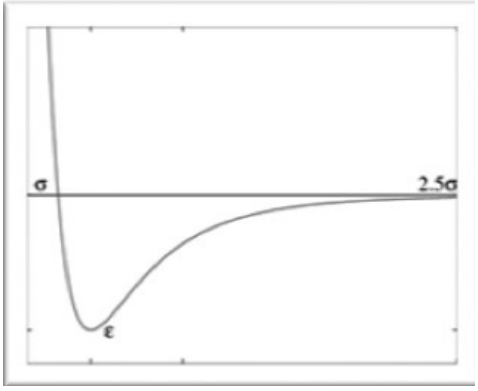


Figure 1 Lennard-Jonnes Potential

with two parameters: σ the diameter, and ϵ the depth.

The potential minimum occurs at $r_{min} = 2^{\frac{1}{6}}$. and $V(r_{min}) = -\epsilon$. An important aspect that would be used is when $r > r_{min}$ the potential switches from being repulsive to being attractive. As $r \to \infty$, $V(r)$ is attractive and decays r^{-6}, as to get an idea how fast $V(r)$ decays, note that $V(r = 2.5\sigma) = -0163\epsilon$ and $V(r = 3.5\sigma) = -.00217\epsilon$. .

Cutoff schemes

In a system of N particles interacting via pair potential, there are $\frac{N}{\frac{N(N-1)}{2}}$ calculations in total. Thus, the cost of N force calculation increases with the square of , and almost all execution time of MD simulation is taken up in the calculation of force as the system size becomes larger.

The cutoff distance Rc is introduced in the Lennard-Jones system to save the computational cost, a good Rc is between 2.5σ and 3.5σ

The cutoff is simply a truncation distance for the Lennard-Jones potential. In this work $Rc = 3\sigma$, then the potential can be described as follow:

$$V(r) = \begin{cases} 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], & r < 3\sigma \\ 0, & r \geq 3\sigma \end{cases}$$

Grids Methods to accelerate Molecular Dynamics Simulations

From particles position, velocity and acceleration the position and the velocity for the next time step can be calculated. Due to computational complexity of this method the amount of computer power required is quite substantial. To reduce computation time the system can be divided into several cells. One particle interacts with all the particles in the same cell as well as the surrounding cells as a whole.

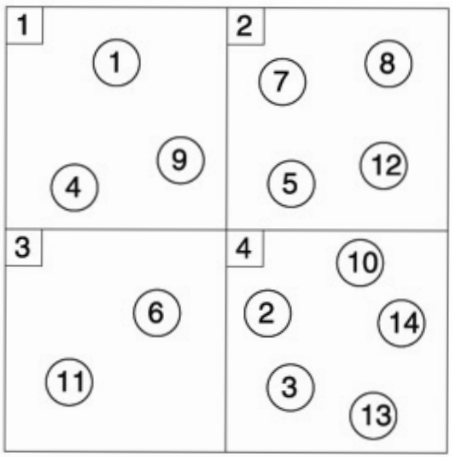
Grid Search Method

To perform the time evolution of a system, we first have to find particle pairs such that the distance between the particles is less than the cutoff length.

Since the trivial computation leads to $O(N^2)$ computation, where N is the number of particles, we adopt a grid algorithm, which is also called the neighbor list method to reduce the complexity of the computation to $O(N)$.

The algorithm finds interacting pairs with in a grid by the following three steps:

1. Divide a system into small cells; in this work every cell has side of length Rc.
2. Register the indices of particles on cells to which they belong.
3. Search for interacting particles pairs using the registered information.



4. Figure 2 Domain divided in cells.

Now it is described step by step how to create a grid:

- First divides the simulation box into small cells of equal size.
- To every particle is assigned a position respect the cell in the direction x, y and z, with the three assigned positions a number of cell can be assigned.
- An auxiliary array is created to save a global index of every particle.
- Sort the arrays respect at cell number, to have sequential labels in every cell, the global index save the real position in memory.
- An array with initial and final position of every cell is created, to know which particles are in every cell.

At final on the Molecular Dynamic simulation, the only important change is how to compute the force, every particle has 27 interacting cells, and then our algorithm has to find the pair interactions in those cells, transforming our quadratic problem in a linear problem.

Temperature in Molecular Dynamics Simulations

In this subsection we describe how to enforce a certain temperature on a three dimensional system of particles. The temperature T of a system and its kinetic E_{kin} energy are related by the equipartition theorem of thermodynamics [1, 2, 3] as

$$E_{kin} = \frac{3N}{2} k_B T.$$

k_B is the proportionality constant called the Boltzmann constant. Thus, the temperature is given by

$$T = \frac{2}{3Nk_B} E_{kin} = \frac{2}{3Nk_B} \sum_{i=1}^{N} \frac{m_i}{2} v_i^2.$$

Then the first idea is already change the kinetic energy of the system by a scaling of the velocities of the particles with a multiplication of the velocity

by the factor $\alpha = \sqrt{\frac{T^D}{T}}$ which will transform the temperature T of the system in the temperature T^D, then only we have to do is to set $v_i = \alpha v_i$ at chosen times of the simulation, for example every 10 steps. This procedure is called thermostat. This can be so rude to the system affecting the energy of the system quite strongly, and then we have to be careful with this aspect.

In this specific case we will cold the system to get a system with less energy, the temperature T^D would be a factor of the current temperature of the particle's system. The advantage of this procedure is its simplicity to move particles with much energy faster than other particles with less energy. In the intermediate steps the system is integrated without scaling, this allows the system to restore the equilibrium between potential and kinetic energy.

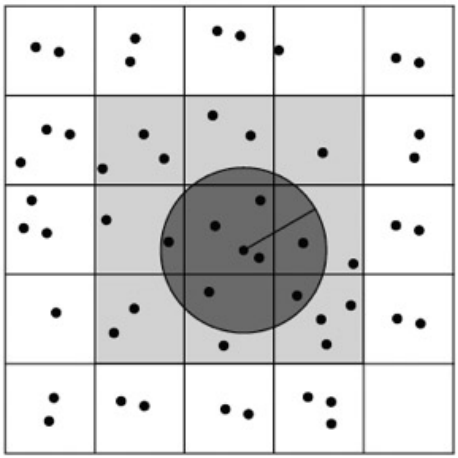


Figure 3 Example of cutoff radius and neighbors list.

3. Implementation

Force computation with OpenMP

For the simulation of a system of N particles on a parallel computer with P processors, the replicated data method operates as follows. The N particles are partitioned into P subsets. Every processor works only on the $\frac{N}{P}$ particles assigned to it.

For example, for the basic force computation this means that each processor computes all the sums for the subset of particles i associated to it $F_i = \sum_{j=1, j \neq i}^{N} F_{ij}$ where F_{ij} denotes the force from particle i on particle j. To be able to execute these computations, each processor needs the positions, and parameters for the potentials of all N particles as copies. Therefore, every processor needs in each time step a current copy of all particles. Each processor then has to receive N pieces of data. The replicated data approach is not ideally suited for the parallelization of grid algorithm, because the method does not scale with P. It's the reason because for huge system I will use a domain decomposition technique. For Molecular Dynamics Simulation of Lennard-Jones cluster we have a short-range potential, and the domain is divided into cells with edges that are at least as long as the cutoff radius R_c. The interactions with particles inside one of the cells can be computed in one loop over the particles in the cell and the particles in directly adjacent cells. Those aspects lead an efficient force computation.

We have to remark than conjugate gradient method and molecular dynamic simulation spend mayor computation time in the force. Then we can do comparison between both methods only counting how many steps spend everyone, doing the force computation with the same algorithm.

Then my first propose is mix both algorithms to solve the optimization Lennard-Jones cluster problem, when the conjugate gradient (CG) is lacking of speed in its convergence I will use molecular dynamic simulation (MD) to escape from the local minimum.

4. Results and Discussion

These implementations were made on C and was used OpenMP for executing the parallel versions. All the test cases were run on the host described below.

Table 1

Time and String Metrics		Constant Metrics	
Machine Type	x86_64	CPU Count	16 CPUs
Operating System	Linux	CPU Speed	2395 MHz
Operating System Release	3.2.0-24-generic	Memory Total	32938000 KB
		Swap Space Total	1020 KB

Results of parallelization methods in molecular dynamics.

Table 2 This table shows how the execution time is better when the number of particles increase.

No. Particles	No. Iterations	Time Normal	Time with Grid
125	100000	18.18s	86.03s
1000	10000	208.98s	109.78s
1000	100000	2650.99s	1363.41s
32767	10	120.49s	9.61s
32767	100	1103.79s	76.21s
125000	1	113.14s	11.58s

It's clear that use Grids when we have more of 1000 particles is very useful to speed up the molecular dynamic simulation.

Table 3 This table shows the execution time of parallelized molecular dynamic simulations of 1000 particles during 1000 time steps.

No. Threads	Time Normal	Time with Grid
1	20.73s	12.99s
2	11.77s	8.51s
4	6.17s	5.10s
8	3.37s	3.80s

When the number of particles is small the parallelization with OpenMP over grids is not efficient sometimes.

Results on Optimization Methods

Method	Dx	No. Iterations	END	Energy
		0	YES	-6.5116213103
CG	1.00E-01	1	YES	-12.5551407146
CG	1.00E-02	24	YES	-24.5712801727
CG	1.00E-03	2951	YES	-33.2478966199
CG	1.00E-04	28006	YES	-33.2478966199
CG	1.00E-05	266384	YES	-33.2478966197
CG	1.00E-06	2520285	YES	-33.2478966179
CG	1.00E-07	23924122	YES	-33.2478965994
CG	1.00E-08	225902094	YES	-33.2478963867

Table 4 This table shows the classical conjugate gradient over a cluster of 12 particles.

We already have a benchmark database for Lennard-Jones clusters in The Cambridge Energy Landscape Database [14]. This database report -37.967600 for the cluster of 12 particles, that show us CG is not enough efficient. The word title END tell us if the conjugate gradient method reach some point were is not improving the potential, then the CG method stop in that iteration the method and return the value reached.

Method	Dx	No. Iterations	END	Energy
		0	YES	-6.5116213103
MD+CG	1.00E-01	0	YES	-36.8984154236
MD+CG	1.00E-02	1	YES	-36.9013776554
MD+CG	1.00E-03	1252	YES	-37.9675995624
MD+CG	1.00E-04	11818	YES	-37.9675995624
MD+CG	1.00E-05	113383	YES	-37.9675995623
MD+CG	1.00E-06	1087170	YES	-37.9675995616
MD+CG	1.00E-07	10377772	YES	-37.9675995542
MD+CG	1.00E-08	98749599	YES	-37.9675994676

Table 5 This table shows an optimization where first was implemented molecular dynamics during 1000 times steps with a thermostat and after the classical conjugate gradient over a cluster of 12 particles.

After do a short simulation with only 1000 steps and cooling the system every 10 steps, we get a better approach and the CG can continue the work in many cases given us reason to believe that MD is a good global method at least to reach a good zone of search. I have initialized this method in the same point taken in Table 4. The Table 6 was initialized too in the same point.

Table 6 This table shows an optimization where first was implemented molecular dynamics during 1000 times steps with a thermostat and after the classical conjugate gradient and repeat again the same process over a cluster of 12 particles.

Method	Dx	No. Iters	END	Energy
		0	YES	-6.5116213103
MD+CG+MD+CG	1.00E-01	0	YES	-37.9667829880
MD+CG+MD+CG	1.00E-02	1	YES	-37.9675847289
MD+CG+MD+CG	1.00E-03	207	YES	-37.9675995624
MD+CG+MD+CG	1.00E-04	1518	YES	-37.9675995624
MD+CG+MD+CG	1.00E-05	10498	YES	-37.9675995623
MD+CG+MD+CG	1.00E-06	61814	YES	-37.9675995618
MD+CG+MD+CG	1.00E-07	366833	YES	-37.9675995587
MD+CG+MD+CG	1.00E-08	2591428	YES	-37.9675995430

Another good application of MD is to escape from a local minimum. How can see in Table 6 This table shows an optimization where first was implemented molecular dynamics during 1000 times steps with a thermostat and after the classical conjugate gradient and repeat again the same process over a cluster of 12 particles.

Table 7 This table shows the classical conjugate gradient over a cluster of 250 particles.

Method	dx	No. Iters	END	Energy
CG		0		-96.23562519
CG	1.00E-01	2	YES	-115.71474
CG	1.00E-02	21	YES	-127.7035226
CG	1.00E-03	1000	NO	-592.6551057
CG	1.00E-04	1000	NO	-104.7639176
CG	1.00E-05	1000	NO	-96.87157531
CG	1.00E-06	1000	NO	-96.29798414
CG	1.00E-07	1000	NO	-96.24184922
CG	1.00E-08	1000	NO	-96.23624748

Table 8 This table shows an optimization where first was implemented molecular dynamics during 1000 times steps with a thermostat and after the classical conjugate gradient over a cluster of 250 particles.

Method	dx	No. Iters	END	Energy
MD+CG		0		-96.23562519
MD+CG	1.00E-01	0	YES	-1470.159752
MD+CG	1.00E-02	0	YES	-1470.159752
MD+CG	1.00E-03	1000	NO	-1470.588317
MD+CG	1.00E-04	1000	NO	-1470.542998
MD+CG	1.00E-05	1000	NO	-1470.443566
MD+CG	1.00E-06	1000	NO	-1470.253546
MD+CG	1.00E-07	1000	NO	-1470.172212
MD+CG	1.00E-08	1000	NO	-1470.16104

Table 9 This table shows an optimization where first was implemented molecular dynamics during 1000 times steps with a thermostat and after the classical conjugate gradient and repeat again the same process over a cluster of 250 particles.

Method	Dx	No. Iters	END	Energy
MD+CG+MD+CG		0	YES	-96.23562519
MD+CG+MD+CG	1.00E-01	0	YES	-1480.161207
MD+CG+MD+CG	1.00E-02	0	YES	-1470.461333
MD+CG+MD+CG	1.00E-03	1000	NO	-1470.596265
MD+CG+MD+CG	1.00E-04	1000	NO	-1470.589122
MD+CG+MD+CG	1.00E-05	1000	NO	-1470.54366
MD+CG+MD+CG	1.00E-06	1000	NO	-1470.482569
MD+CG+MD+CG	1.00E-07	1000	NO	-1470.458663
MD+CG+MD+CG	1.00E-08	1000	NO	-1470.455332

Table 10 This table shows the classical conjugate gradient over a cluster of 1000 particles.

Method	Dx	No. Iters	END	Energy
CG		0		-393.1458244
CG	1.00E-01	5	YES	-782.7501996
CG	1.00E-02	27	YES	-474.5103793
CG	1.00E-03	1000	NO	-1970.434898
CG	1.00E-04	1000	NO	-409.6512408
CG	1.00E-05	1000	NO	-394.4868945
CG	1.00E-06	1000	NO	-393.2778812
CG	1.00E-07	1000	NO	-393.1590102
CG	1.00E-08	1000	NO	-393.1471428

Table 11 This table shows an optimization where first was implemented molecular dynamics during 1000 times steps with a thermostat and after the classical conjugate gradient over a cluster of 1000 particles.

Method	Dx	No. Iters	END	Energy
MD+CG		0		-393.1458244
MD+CG	1.00E-01	0	YES	-6258.623079
MD+CG	1.00E-02	0	YES	-6258.623079
MD+CG	1.00E-03	1000	NO	-6351.782221
MD+CG	1.00E-04	1000	NO	-6291.317211
MD+CG	1.00E-05	1000	NO	-6278.666914
MD+CG	1.00E-06	1000	NO	-6265.28216
MD+CG	1.00E-07	1000	NO	-6259.53062
MD+CG	1.00E-08	1000	NO	-6258.717269

Table 12 This table shows an optimization where first was implemented molecular dynamics during 1000 times steps with a thermostat and after the classical conjugate gradient and repeat again the same process over a cluster of 1000 particles.

Method	Dx	No. Iters	END	Energy
MD+CG+MD+CG		0		-393.1458244
MD+CG+MD+CG	1.00E-01	0	YES	-30.83542373
MD+CG+MD+CG	1.00E-02	0	YES	-6614.177313
MD+CG+MD+CG	1.00E-03	1000	NO	-6613.213415
MD+CG+MD+CG	1.00E-04	1000	NO	-6600.480363
MD+CG+MD+CG	1.00E-05	1000	NO	-6633.987865
MD+CG+MD+CG	1.00E-06	1000	NO	-6606.972223
MD+CG+MD+CG	1.00E-07	1000	NO	-6631.338996
MD+CG+MD+CG	1.00E-08	1000	NO	-6606.688904

5. Conclusions and Future Work

In this article were implemented some optimization algorithms only mixing two strategies the first was the classic algorithm of conjugate gradient and the second is similar to annealing evolutionary algorithm, using a classic thermostat that cooling the system, saving several steps of conjugate gradient. The conjugate gradient algorithm depends on the initial point, it is the reason because is only a local method that only minimize but do not find a minimum by itself in the major of the cases. Using molecular dynamics

simulation to cool the system was gotten a better initial configuration to use the conjugate gradient method, getting the minimum energy for 12 particles as is reported in [22]. In the case of 250 and 1000 particles, the method only was used with 1000 steps to show that a better approach to the minimum was gotten than only using conjugate gradient. Another conjecture was to use molecular dynamics to escape from a local minimum; this can be observed comparing table 11 with table 12.

We focused on studying their performance using parallel computing and domain decomposition, getting a very good option to accelerate the compute of the energy and the force of a particle system, this option use the grid technique parallelized with OpenMP technology, both result useful to accelerate molecular dynamic simulations and the method of conjugate gradient, permitting to work with huge clusters of particles.

Now we have reasons to believe in Molecular Dynamic Simulations like a good global method to solve the problem of optimization of Lennard-Jones clusters.

As future work, would be implemented the algorithms using MPI and CUDA technology which allow us to increase the size of the number of particles, furthermore we have to optimize the parameter of the thermostat. Also I have to look for a better local algorithm and consider system with more complex potential like could be potential associated to biological systems.

References

[1] Allen, P. and Tildesley, D.J., "Computer Simulation of Liquids", Oxford Science Publ, (1989).

[2] Frenkel, D. and Smit, B., "Understanding Molecular Simulation: From Algorithms to Applications", Computational science, (2002).

[3] Rapaport, D.C., "The Art of Molecular Dynamics Simulation", Cambridge University Press, (2004).

[4] Hiroshi Watanabe, Masaru Suzuki, and Nobuyasu Ito, "Efficient Implementations of Molecular Dynamics Simulations for Lennard-Jones Systems", Progress of Theoretical Physics, Vol. 126, No. 2, August 2011.

[5] B. J. Alder and T. E. Wainwright, J. Chem. Phys. 27 (1957), 1208.

[6] A. Rahman, "Correlation in the motion of atoms in liquid argon". Physical Review 136(2A):405(1964).

[7] L. Verlet, "Computer Experiments on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules", Phys. Rev. 159, 98 (1967),

[8] B. Chapman, G. Jost, R. van der Pas, *Using OpenMP: Portable Shared Memory Parallel Programming*, The MIT Press, (2008).

[9] B. Wilkinson, M. Allen, *Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers*, 2nd edition. Pearson Education Inc. (2005).

[10] B. Barney. OpenMP. (2012) https://computing.llnl.gov/tutorials/openMP.

[11] C. Breshears. *The Art of Concurrency: A Thread Monkey's Guide to Writing Parallel Applications* , O'Reilly Media, Inc., (2009).

[12] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson, *Introduction to Algorithms* (2nd ed.), McGraw-Hill Higher Education. (2001)

[13] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery ,"Numerical Recipes

in C - The Art of Scientific Computing", Cambridge University Press; 2nd edition (1992).
[14] http://www-wales.ch.cam.ac.uk/CCD.html

[15] C. Barron, S. Gomez, and D. Romero, "Lower energy icosahedral atomic clusters with incomplete core," Appl. Math. Lett., vol. 5, pp. 25–28(1997).

[16] Wensheng Cai and Xueguang Shao, "A Fast Annealing Evolutionary Algorithm for Global Optimization", J. Comput Chem 23: 427-435 (2002).

[17] Wensheng Cai, Haiyan Jiang, and Xueguang Shao, "Global Optimization of Lennard-Jones Clusters by a Parallel Fast Annealing Evolutionary Algorithm", J. Chem. Inf. Comput. Sci., 42(5), pp 1099-1103(2002).

[18] Lixin Zhan, Bart Piwowar and Wing-Ki Liu, "Multicanonical basin hopping: A new global optimization method for complex systems", Journal of Physics, volume 120, number 12, pp 5536-5542(2004).

[19] Yuhong, Longjiu, Wensheng and Xueguang, "Structural Distribution of Lennard-Jones Clusters Containing 562 to 1000 Atoms", J. Phys. Chem. A., 108(44), pp 9516-9520(2004).

[20] XiangJing Lai, RuChu Xu, WenQi Huang, "Prediction of the lowest energy configuration for Lennard-Jones clusters", Science China Chemistry, Volume 54, Issue 6, pp 985-991(June 2011).

[21] Wayne Pullan, "An Unbiased Population-Based Search for the Geometry Optimization of Lennard-Jones Clusters: $2 \leq N \leq 372$", J Comput Chem 26: 899-906(2005).

[22] http://chinfo.nankai.edu.cn/chmm/pubmats/LJ/ljstructures_e.html

GPU's molecular dynamics simulation of a one million particles

José María Zamora and Pedro D. Cruz-Santiago
Lufac Computación®
México, D.F., México
Email: josema zamora@lufac.com
Enrique Díaz-Herrera
Departamento de Física
Universidad Autónoma Metropolitana
México, D.F., México
Email: diaz@xanum.uam.mx
Luis M. de la Cruz
GMMC, Depto. de Recursos Naturales
Instituto de Geofisica – UNAM
M´exico, D.F., M´exico
Email: luiggi@unam.mx

Abstract

In this work the main algorithms involved in a molecular dynamics simulation are described. We present an strategy of parallelization using CUDA to accelerate the computations in GPUs. We show several examples of application of our implementations for 500, 2048, 10000 and 106 particles. We show very good results in terms of computation time and accuracy of the results.

I. Introduction

The advancement in technologies of high-performance computing, both hardware and software, have made it possible to improve the approximation of solutions to complex problems in science and engineering. For example, in [1] it is shown that the use of GPUs to solve optimization problems using genetic algorithms, dramatically reduces the time to compute the solutions.

In this work we address the use of the Graphic Processor Unit (GPU) architecture. This architecture was originally designed and developed to solve problems within the computer graphics, particularly in the area of video games. However in the last years the GPUs have found crucial application in the scientific computing. Some algorithms for molecular dynamics, which are based on the simulation of N-bodies are reviewed in this paper. Since several decades ago, this area of study has had many advances in both the algorithms and implementations. One of the first work in this direction is the Allen and Tildesley [2]. There are three very important components that are highlighted in this area:

1. The solution to the problem of N-bodies requires a intensive numeric calculation, so it is necessary to optimize the algorithms reduce as much as possible the number of

operations.

2. At the same time it reduces the number of operations and generate solutions increasingly more accurate, with the aim of comparing the results of the simulations against experimental results.
3. The correct application of architectures for highperformance computing, such as GPUs in this area, it is very important because it is through these that can be generate solutions in reasonably short times, which is the main objective of this work.

A comprehensive survey of new algorithms, as well as of recent computing architectures, allows you to generate very useful tools to obtain precise solutions in short times [3].

II. Background

In the past 50 years the solutions of problems in many areas of science, by computational methods have expanded rapidly in many fields, due to the fast and stable growth of the computing power. The price-performance ratio has increased by an order of magnitude approximately every 5 years, and there are no signs of any change in this trend. On the contrary, with the incursion of new technologies it is possible to predict that in 2019 you can reach the barrier of exaflops. This means that could be simulated large-scale problems, for example very complex molecular systems over a very long period of time.

Similarly, it will be possible to use interactions increasingly complex in biological systems, such as the DNA and the folding of proteins.

For decades, have used a wide variety of techniques to accelerate the Molecular Dynamics simulations. Clusters have been used for High Performance Computing (HPC) and also computer architectures quite novel. In general it is possible to classify these strategies into two categories: coarse-grained and fine grain.

The architectures in the category of coarse-grained include general-purpose supercomputers, clusters of PC's, Bewolfs and Grids. Usually the supercomputers such as the Blue Gene, which in 2006 was ranked first in the TOP500 with 478.2 teraFLOP/s, in Molecular Dynamics can give stunning performance [4]. However, in some cases can be quite expensive and inaccessible to some researchers from lesser developed countries. It is important to emphasize that the fact of using clusters suffers from problems of scalability to grow the number of processors due to the high latencies in the communications between computers. Cloud-based projects such as Folding@home [5] and Predictor@Home [6]1 have attracted several hundreds of thousands of volunteers around the world to collaborate with CPU time on their PCs and Playstations. Unfortunately, these compute resources of volunteer only allow communication between master servers and clients, which is why this approach is only desirable for simulations with a large number of particles, to steps of times short so that the scales of integration are large.

All these approaches to perform a molecular simulation, they can be viewed as algorithms that attempt to obtain the more computing power at the lowest economic cost. The main advantage of the GPUs architectures compared with the above-mentioned components is that they are quite flexible.

In particular, the majority of users with an ordinary computer already have access to a fairly modern graphics cards. For these users this solution

has a very low cost. In addition, the installation of these devices is trivial (the majority are plug & play). To write software that takes advantage of the features of a GPU, it requires quite specialized knowledge. However, the emergence of new models of high-level programming, such as CUDA, offer a programming environment similar to C.

A. Molecular dynamics

Computational thermodynamics is an area of study whose interest is on the rise. In this discipline the problems are solved by numerical methods that are implemented in computer systems. One way of achieving the foregoing is using molecular dynamics simulations, where the dynamics of systems of objects (particles, atoms, molecules, colloids, disks, polymers, etc.) distributed in a physical space are modeled. Each object has a set of physical properties and its dynamics depend on a model of the interactions between them. Using a model it is possible to obtain, several quantities, for example, the internal energy, pressure, thermal diffusivity, viscosity, etc. Subsequently, these quantities are compared with data that are determined experimentally. This comparison is made to validate the model of interactions. Once the model has been validated, this can be used to study relationships between physical quantities of the same and make predictions. In the best case, it is possible to obtain non-measurable properties.

In this work, we are trying to answer two basic questions.

Is there a relationship between molecular structure and some thermodynamic properties as the pressure? If one varies some property as for example, the density of the system, the molecular structure changes?

Here the density is defined as: $\rho = N/V$, where N is the number of particles and V is the volume that deals with the system, both finite quantities. As is known, thermodynamics considers the infinite limit. Therefore, a Molecular Dynamics, only give approximate results.

Molecular systems are generally quite heterogeneous and complex to be treated with analytical methods. As we know, the field is present in our around in three classic thermodynamic states: gas, liquid and solid, and now science has only two ways to address the study of these states from a molecular or atomics point of view as follows:

1. One is the classical statistical mechanics, which is a branch of physics which derive properties of matter that emerge at macroscopic level from the atomic structure and of the microscopic dynamics of the same. Some of the properties that emerge at macroscopic level of a material are: temperature, pressure, flow, the constant magnetic, dielectric, etc. These properties are essentially determined by the interaction of many atoms or molecules. The key point of statistical mechanics is the introduction of probabilities in a classical physics inherently deterministic [7].
2. The other alternative is quantum mechanics, the branch that can predict the behavior of the material from their electrical properties and their atomic energies. This branch of physics considers the particles, which constitute the atom, as well as the associated energy, as objects and quantities to be discrete. Thus, the molecular structures formed by several atom, depend on the properties of charge and mass of elementary objects (proton,

electron, etc.) [8].

The treatment of molecular systems in the gas phase using methods of quantum mechanics along with statistical mechanics is simple. This is due to the possibility of reducing the number of particles on the basis of the low density of such system. In the crystalline solid state, the face problems with quantum mechanics and statistical is made possible by the reduction of the system of many particles to few particles due to symmetry properties of the solid state. Between these two extremes there are liquids, such as macromolecules, solutions, amorphous solid, etc. These state of matter can only be addressed as a system of many particles. There is no single reduction, due to the fact that all the degrees of freedom, translation, rotation and vibration, provide information to describe the macroscopic properties of the system. These facts have two implications for the study of systems of this type:

1. The analytical study of these systems is virtually impossible. As an alternative is the numerical simulation of the behavior of a molecular system using a computer, which,
2. A simulation will produce a statistical ensemble of configurations to represent the thermodynamic state of the system under study.

A statistical ensemble refers to a collection of system configurations obtained in different time steps. In this work configuration is used as a synonym for the structure of the system.

B. Simulation of molecular systems

The simulation of molecular systems to different temperatures, require the generation of a set of statistically representative configurations, this is known as assemble. The properties of a system are defined as averages in the assemble or integral in the configuration space. This configuration space is also known as phase-space. For a system of many particles the averaging or integrate will involve several degrees of freedom of each particle. The translation is defined by the spatial location of the particles, while the rotation is defined when the particle has a different shape to a sphere. The accuracy of a molecular model depends entirely on the force field or interaction, define only between pairs of particles, in later sections will be given a more formal definition, in the meantime it is important to know that there is a wide variety of these. Some interactions are very complex to implement, but very accurate, while others are very simple but not very accurate, some that occupy many computing resources and are very complex to implement.

When we look at a molecular system by computational simulations there are three factors that should be considered:

1. The properties of a molecular system that one is interested in measuring (pressure, temperature, density, energy, etc.) and the configuration space. Should be define the relevant configurations to be estimated.
2. The precision required by the properties specified in item 1.
3. The computational time should also be estimated. The incorrect estimate of this point you may end up in extremely long simulations as the age of the universe.

III. N-Body Algorithms

The N-bodies algorithm used in this work is as follows:

Algorithm 1 N-Body Algorithms

```
N = Number of Bodies
T_max = Total number of time steps
for t = 1 to T_max do
    for i = 1 to N do
        for j = 1 to N do
            Compute f(B_i, B_j)
        end for
    end for
    for i = 1 to N do
        Integrate any equation of motion B_i
    end for
    for i = 1 to N do
        Update the dynamic properties of B_i
    end for
end for
```

IV. Theoretical Aspects On Molecular Dynamics

A. Lennard Jones interactions

The molecular dynamics simulations are based on classical physics and, by consequence, in the equations of motion of Newton of N bodies (see [9]). In this case a body is interpreted as a particle. Thus, every particle in the system is seen as an object with different intrinsic characteristics to their behavior, which are: position, velocity, and acceleration defined within the set of dynamic properties, and the mass as a static property.

The equation of motion of the i-th object is defined by Newton's second law that is written as:

$$\mathbf{f_i} = m_i \ddot{\mathbf{r}}_i = m_i \mathbf{a}_i \quad , \qquad (1)$$

where a_i represents the acceleration of the i-nth particle. Here $\ddot{}_i$ represents the second derivative with respect to the time of the position ri of the particle.

The position of each particle is located in a threedimensional space of Cartesian coordinates, and is denoted by r_i. Thus, the distance between two particles is:

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad . \qquad (2)$$

In this work, in order to define a relationship of interaction between the particles, we will implement a molecular model based on the potential of Lenard-Jones (see [9]), which is written as:

$$U_{ij} = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right] \quad . \qquad (3)$$

B. Calculating the forces

The vector force between the particle i and particle j, can be written as:

$$\mathbf{F}_{ij} = -\frac{\partial U_{ij}(r_{ij})}{\partial x_{ij}} \hat{e_x} - \frac{\partial U_{ij}(r_{ij})}{\partial y_{ij}} \hat{e_y} - \frac{\partial U_{ij}(r_{ij})}{\partial z_{ij}} \hat{e_z} \qquad (4)$$

Therefore, the component x of F_{ij} is :

$$\frac{\partial U_{ij}(r_{ij})}{\partial x_{ij}} = -48 \left[\frac{1}{r_{ij}^{14}} - \frac{1}{2 r_{ij}^{8}} \right] x_{ij}. \qquad (5)$$

Extending the equation 5 on the components *x,y,z* we have:

$$F_{ij}^{x} = \frac{\partial U_{ij}(r_{ij})}{\partial x_{ij}} = -48 \left[\frac{1}{r_{ij}^{14}} - \frac{1}{2}\frac{1}{r_{ij}^{8}} \right] x_{ij} \qquad (6)$$

$$F_{ij}^{y} = \frac{\partial U_{ij}(r_{ij})}{\partial y_{ij}} = -48 \left[\frac{1}{r_{ij}^{14}} - \frac{1}{2}\frac{1}{r_{ij}^{8}} \right] y_{ij} \qquad (7)$$

$$F_{ij}^{z} = \frac{\partial U_{ij}(r_{ij})}{\partial z_{ij}} = -48 \left[\frac{1}{r_{ij}^{14}} - \frac{1}{2}\frac{1}{r_{ij}^{8}} \right] z_{ij} \qquad (8)$$

Thus, the force on the particle i due to all the N-particles of the system can be written as:

$$\vec{F_i} = \vec{F_{i1}} + \vec{F_{i2}} + \vec{F_{i3}} + ... + \vec{F_{iN}} \qquad (9)$$

Note that the equations 6, 7 and 8 only positions are required to obtain the strength in each component. These positions will be spatially ordered arrangements as shown in the Table I and are inputs to the algorithms.

TABLE I. Memory arrays for all particles

particles	1	2	3	...	N
r_x	x_1	x_2	x_3	...	x_N
r_y	y_1	y_2	y_3	...	y_N
r_z	z_1	z_2	z_3	...	z_N

C. Numerical integration of velocity Verlet

This method allows to obtain a direct solution to equation

1. The equations that govern this method are [10]:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \Delta t \mathbf{v}(t) + (1/2)\,\Delta t^2\,\mathbf{a}(t) \qquad (10a)$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \frac{1}{2}\Delta t\,[\mathbf{a}(t) + \mathbf{a}(t+\Delta t)] \qquad (10b)$$

These equations can be obtained from the Verlet algorithm, eliminating the speeds. This algorithm only requires memory save r , v and a of each particle; also, this algorithm minimizes rounding errors [11]. In this algorithm the new positions at the time $t + \Delta t$ are calculated using the equation 10a; then the speeds at half step of time are calculated using:

$$\mathbf{v}\left(t + \frac{\Delta t}{2}\right) = \mathbf{v}(t) + \frac{1}{2}\Delta t\,\mathbf{a}(t) \qquad (11)$$

Now it is possible to compute the forces and accelerations at time $t + \Delta t$ and then the movement of the speeds can be completed by:

$$\mathbf{v}(t+\Delta t) = \mathbf{v}\left(t + \frac{\Delta t}{2}\right) + \frac{1}{2}\Delta t\,\mathbf{a}(t+\Delta t) \qquad (12)$$

At this point, the kinetic energy at the time $t + \Delta t$ can be calculated using:

$$E_K = \frac{1}{2}\sum_{i=1}^{N} m_i \|\mathbf{v}_i(t)\|^2 \qquad (13)$$

While the potential energy in this time step will be assessed through the calculation of the forces with the equation 3.

The total process is shown in Figure 1. This method uses $9N$ words of memory space, it is numerically stable, convenient and easy to implement this algorithm is one of the most widely used in molecular dynamics simulations (see [2]).

D. Calculation of the pressure and temperature in a canonical ensemble

The equation to determine the pressure of a homogeneous fluid inside a box of simulation will be given by:

$$P = \frac{N k_B T}{V} - \frac{1}{3V}\left\langle \sum_i \sum_{j>i} \mathbf{r}_{ij} \left.\frac{dU(r)}{dr}\right|_{r_{ij}} \right\rangle \quad ,, \qquad (14)$$

where N is the number of particles, k_B is Boltzmann's constant, V is the volume of the box, r_{ij} is the distance between particles and $\left.\frac{dU(r)}{dr}\right|_{r_{ij}}$ is the derivative of the potential between particles.

The temperature is controlled during the simulation method ulilizando Nose-Hoover [12].

E. Molecular dynamics algorithm

In this section is describes the algorithms that are introduced during the development of this study. These algorithms take ideas from different sources, in which explains the processes that are required to perform simulations of molecar dynamics.

Below is a basic algorithm to generate a dynamic of spherical objects that evolve in time, where it is considered a general method of integration:

The cycle of the dynamic is considered on the variable t and has to perform T_{max} changes over time. This cycle generates the change of the system.

The Algorithm 2 must be modified in accordance with the needs of the computational implementation (programming language, parallelizing), even the order of the sections within the cycle of the dynamic may vary. In the case of the integration of the equations of motion (step 1 and 6), are optional in the sense that they can appear in the step 1 or 6, but even this operation is not necessarily a integration. In addition, the method of integration may depend on the assemble that you are using (NVT, NVE, NPT, etc).

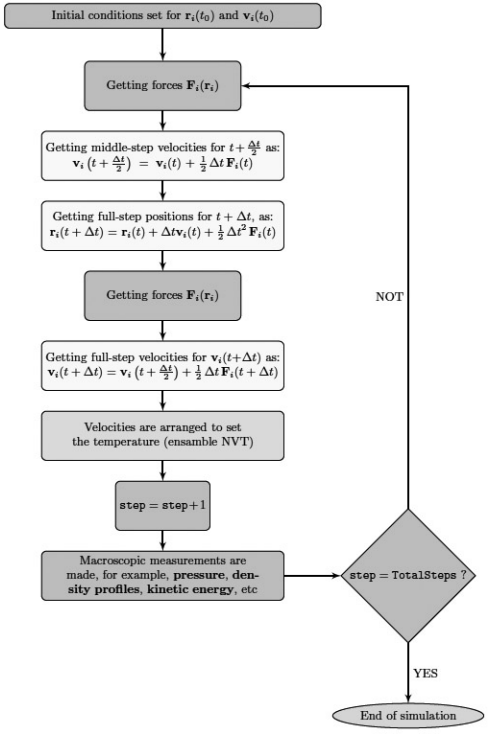


Fig. 1. Flowchart for a Molecular Dynamic using Velocity Verlet

Algorithm 2 General algorithm of a Molecular Dynamics

You choose the initial conditions of the simulation.
Calculation of forces
for $t = 1$ to T_{max} **do**
1. Integration of the equations of motion.
2. Calculation of boundary conditions.
3. Calculation of interactions.
4. Measurement of any property.
5. Integration of the equations of motion.
6. Reporting quantities snapshots.

end for

V. Parallel Programming In Gpu

For some years there is an increase in scientific applications that make use of GPUs to reduce the computation time of its processes. Several authors report that their accelerated codes are 5 to 10 times faster than using CPUs (see [13]). The reason for this acceleration is the fundamental difference in the design of GPUs and CPUs. On a CPU, we try to optimize the sequential operations using a very sophisticated control logic that enables arithmetic instructions are executed on a thread, plus how to handle countless operations: device management, input-output and interrupts. Now, CPUs reach up to 20 cores and shared memory programming (parallel) is increasingly contested but N-body applications remains much lower than the yield on a GPU. On the other hand, the philosophy of design and construction of a GPU focuses on maximizing the area of the chip (ALUs) dedicated to massive floating point operations. The basic idea is to optimize the parallel execution of a very large number of threads in many arithmetic units. At present, GPUs are designed as engines numerical calculation, so that this type of architecture will not be optimal in other tasks that have nothing to do with floating point operations.

Therefore, it is desirable to use a CPU-GPU combination, executing the sequential parts of the code in the CPU and numeric-intensive parts in the GPU. It is for this reason that CUDA (Compute Unified Device Architecture) programming model

designed for NVIDIA GPUs, supports executions of a hybrid application (CPU-GPU).

A. CUDA

From a hardware standpoint, CUDA is using the GPU as a set of SIMD (Single Instruction Multiple Data) processors. Today, the number of processors on the GPU is between 1536 and 2880 processors (CUDA cores) per graphics card. cite teslaweb. In the CUDA programming model we have multiprocessors SMs (streaming multiprocessors), which contain a defined number of processors or SPs (Stream processors). Each SM is viewed as a multi-core device that is capable of executing a large number of parallel threads. These threads are organized in blocks. The threads in the same block can cooperate effectively together because they share data and synchronize your execution at memory access with other threads. Each thread has a unique identifier, similar to a coordinate in 2D or 3D, which means that they can read and modify data from memory simultaneously without conflict with other threads. However, threads in different blocks can not communicate and synchronize. Having occupied the mostly threads can help reduce gaps in data transfer rate to and from the global memory. Data and numerical computation portion to be held in parallel, are isolated in subprograms known as kernels. During execution of a CUDA program, the kernel is loaded into the hardware of the GPU to be used later during intensive numerical calculations.

A SM having a built-in chip is divided into four memory types:

1) A set of registers per processor;
2) A sheared memory;
3) A constant memory-cache of read only;
4) A texture memory-cache of read only;

Each of the previous memories are easily accessible and should be used in arithmetic operations of high demand. Data exchange between these memories well managed resulting in good performance. Furthermore, although the global memory has a speed lower access, can be harnessed to share data, because all threads have direct access to it.

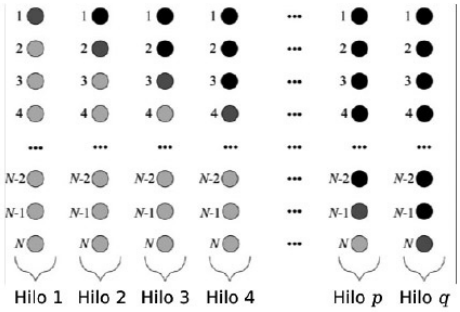


Fig. 2. Distribution of particles in the threads.

B. Calculation of forces in molecular dynamics based in CUDA

Many parallel algorithms for Molecular Dynamics simulation have been proposed and implemented by different researchers ([14],[15]). The details of these algorithms vary too, as they are dependent on the application and computational architecture that will be used. In [15] the algorithms are classified from the point of view of the decomposition of data. This paper takes advantage of the inherent parallelism in the simulations and in particular uses the atom-decomposition (AD) method. The main reasons for choosing this method are:

- We can achieve good scalability and good load balancing.

- According to the model of CUDA that describe in section V-A, GPU hardware can be seen as a shared memory multiprocessor system, therefore, the DA method can give a good performance in such systems (see [15]).

As shown in Figure 2, the idea is that a thread is in charge of the particle number circled in red, and calculate all interactions of his column, both black and gray interactions (unlike the algorithm in a CPU), the end only split in half all contributions to the force components.

C. Preparation of the forces calculation

As seen in previous sections the computation intensive part is the calculation of forces. In [16] is available the software for this work supported by Lufac Computación®. The explanation of some key parts of this code is given below:

1) The kernel for this calculation receives as parameters two arrays of floating-point type with 4 positions in memory, i.e., each element occupy 16 bytes. which represent Cartesian coordinates. An array stores the positions and other saves the forces of all particles:

```
__global__ static void \
MDGPU_ljForcesN2(float4* pPos, \
                 float4* pFor)
```

2) Subsequently, we get each thread identifier with the following macro:

```
tid = threadIdx.x+blockDim.x * \
(blockIdx.x+gridDim.x * blockIdx.y)
```

We use blockDim.x and texttt blockDim.y, because the positions of the particles are stored in a twodimensional arrays of texture memory. Thus, the coordinates of the block gives us the index of the thread which can be represented as a one-dimensional array. Established by [17], we say that the more sophisticated memory in a GPU, is the texture memory. Fortunately, this memory can be used for both graphics processing as for general purpose computing.

The texture memory is constituted as a memory texttt cache within the GPU processing chip, this makes access to it has a much higher rate of transfer to other devices. Besides constant access to this type of memory improve performance of programs. Specifically, the texture caches are designed for processing graphics applications (optimized OpenGL [18] and DirectX [19]), which is because displaying great performance for applications where the threads execute tasks that are spatially divided, that is, that a thread requires data from adjacent yarns. In the application that we developed this spatial distribution is governed by the particle and position.

3) We obtain the value of the position of the particle pointed by the thread tid. Is important to remember that this kernel will be executed by all the threads, so each thread will get values of position (see Figure 2) texture, to bring them to the cache.

```
f4Pos = \
tex1Dfetch(g_tPosTex, (long int) tid);
```

4) The variable for storing the positions of the particles are defined as accommodated globally shared memory variables on GPU:

```
// Copy to shared memory
s_fvPosX[threadIdx.x] = f4Pos.x;
s_fvPosY[threadIdx.x] = f4Pos.y;
s_fvPosZ[threadIdx.x] = f4Pos.z;

__syncthreads();
```

Function `__syncthreads` is a barrier to where they have to get all the threads to continue running the kernel. If the threads are not synchronized at this point, then we can read incorrect data and subsequent calculations would also be incorrect.

The Tesla GPU K20Xm has 49152 bytes of shared memory per block. If a floating-point data has 4 bytes, then we can store at most 4096 particles in each block.

5) Each thread has a variable to accumulate the contributions of the calculated forces, the name of this variable is texttt f4For, which is a data structure of 4 floating-point, and each element represents a Cartesian coordinate.

D. Loops of forces calculation

The calculation of the forces of each particle is divided into two loops:

1) Computation of particle interactions (threads) of the same
block (Inner Loop):

Listing 1. Inner Loop on the particles within the first block

```
1   for (i=0; i<nBlockSize; ++i)
2   {
3     //get the positions of the shared memory
4     f4Pos2.x = s_fvPosX[i];
5     f4Pos2.y = s_fvPosY[i];
6     f4Pos2.z = s_fvPosZ[i];
7
8     //Calculation of distance
9     f4R.x = f4Pos.x - f4Pos2.x;
10    f4R.y = f4Pos.y - f4Pos2.y;
11    f4R.z = f4Pos.z - f4Pos2.z;
12
13    //Apply periodic baundary conditions
14    MDGPU_pbcDistance( f4R );
15
16    //Compute LJ force which is a constant
17    fC = MDGPU_ljForce( f4R );
18
19    //The contribution of force is added to the
          particle tid
20    f4For.x += f4R.x * fC;
21    f4For.y += f4R.y * fC;
22    f4For.z += f4R.z * fC;
23  }
```

This first loop is quite simple: we first obtain the positions of the particles of the other blocks (lines 4 to 6), which are stored in shared memory.

In line 17 is calculated the force constant of type Lennard Jones. This routine requires only the distance of the particles stored in `f4R`.

The function `MDGPU_ljForce`, calculates a constant that is required to climb the positions and obtain the contribution of the interaction in the 3 cartesian coordinates of the particles (line 17). The details of this routine are made according to what was shown in the section about the potential of Lennard-Jones (see IV-A). It is noteworthy that within the code of the routine `MDGPU_ljForce`, discriminate particles that are outside of the radius cut, and the interaction of the particle with herself.

2) Calculating particle interactions with other blocks:

Listing 2. Loop on the particles of missing blocks

```
1   const int nBlocks = gridDim.x * gridDim.y;
2   for (k=1; k<nBlocks; ++k)
3   {
4     // get the indices of k-th block
5     i = tid + k * nBlockSize;
6
7     // i = i mod N
8     if (i >= c_nParticles) i -= c_nParticles;
9
10    // We ensure that all threads have
          successfully
11    // uploaded your data in shared memory
12    __syncthreads();
13
14    // Obtain the positions of the particles of
          another block
15    f4Pos2 = tex1Dfetch( g_tPosTex, (int) i );
16
17    // Copy to shared memory
18    s_fvPosX[threadIdx.x] = f4Pos2.x;
19    s_fvPosY[threadIdx.x] = f4Pos2.y;
20    s_fvPosZ[threadIdx.x] = f4Pos2.z;
21
22    __syncthreads();
23
24    // loop on the all particles in the block
25    for (i=0; i<nBlockSize; ++i)
26    {
27      f4Pos2.x = s_fvPosX[i];
28      f4Pos2.y = s fvPosY[i];
29      f4Pos2.z = s_fvPosZ[i];
30
31      ....
32
33      reference to Inner Loop on the particles
            within the first block
34
35    }
36  }
37  //write out the total of forces computed
38  pFor[tid] = f4For;
39 }
```

On line 1 of this code, the total number of blocks of the simulation is calculated. The number of blocks is a two-dimensional form.

In this second cycle will involve interactions with the particles out of the block. To do this requires two nested loops: the first to point to a block that lack using the variable `k` and the inner loop to get the threads of the current block using the variable `i`. The idea of this second loop is to move the positions of the particles corresponding to a block to the shared memory of another block.

In line 6 is used provisionally the variable i to store the index *one-dimensional* of the k-th block. This index depends on the thread that is running the code, because their block address is different, so this line can be read as the index of the k-th block executed relative to the thread. The last threads will have an address greater than the number of particles, this may not be possible. One way to deal with this is to use the module operation with the number of particles (line 8), with this we make sure the particles corresponding are emulating a queue. Once calculated the index of the particle in another block can read the positions of this particle in texture memory and copy it to shared memory (lines 18-20). Subsequently, the threads are synchronized, not to read incorrect data in the inner loop. hat is already exchanged the memory spaces between the blocks, each thread within the block calculates their interactions with the particles of the same as explained in previous steps. This is done in the cycle that goes inside of the lines 25 to 35. It is also necessary to calculate the minimum conditions of image and the contribution of the force.

At the end of these loops will take all the contributions of all the particles, so each thread writes its total force sum in the array `(pFor)` that will be used to perform the integration.

E. Coding highlights

- The calculation of the forces is done in T number of steps, i.e. kernel listing in 2 will run T times.
- To achieve good performance in the GPU, there is a trade of blocks between threads, exploiting its shared memory to the maximum. This is essential, otherwise it gets a bottleneck in the GPU, this can ensure that the code could be slower than on a CPU.
- The use of routines of device as are MDGPU_pbcDistance and MDGPU_

ljForce make the code more readable in CUDA and decrease the coding errors.

- The indexing of the wires is essential. And in these codes are macros to do this and don't have to be repeating code, this also avoids coding errors. Using textures also accelerates the calculation because the GPU is optimized to access memory in this way.
- Integration is done as shown by the algorithm given by equations 10a, 10b, 11 and 12. Which is pretty simple in the GPU because it is a sums of vectors.
- All the simulation is calculated within the GPU there is only a transfer from the CPU to the GPU at the start of the simulation where are copied some constants, the positions and the initial velocities. Subsequently, performed all the steps of integration within the GPU. And finally there are some scalar measures such as: the kinetic energy, potential energy, pressure and temperature.

VI. Results

In this section we present the results of molecular dynamics in GPU's corresponding to the study of liquid-vapor equilibrium of a Lennard-Jones fluid.

A. Details of the simulations

We studied the effects of finite size of the molecular simulations. To do this were considered systems of 500, 2048, 10,000 and a million particles. All results are presented in units with reduced $\sigma = 1$ and $\epsilon = 1.$. The simulations were performed in the NVT ensemble and the temperature was set to be 0.85. We used a Nose-Hoover thermostat with parameter Q=100. The interaction potential was cut in a radius of 3.0 for all simulated systems, less than 500 particles where the radius cut was 2.5. Periodical boundary conditions were used and as initial configuration was fcc-type crystal. The integration step was 0,003.

All the simulations were performed in a GPU, Tesla K20Xm whit compute capability 3.5. With the interest of testing the accuracy and measure the performance of some result system of N=2048 were simulated on a GPU Tesla K10Xm and a GPU Geforce 760.

B. van der Waals's isotherm

In the Figure 3 are presented the results of the behavior of pressure as a function of the average density for a temperature of 0.85, this graph is known in the literature as isotherm of van der Waals [7]. It is widely known, that for small systems of particles (N=500 and 2048) the isotherm presents a maximum at low densities and a minimum to high densities and indicate respectively the start and end of region known as thermodynamic instability and that border is known as spinodal. In addition is presented in this graph the values of the pressure and density (square symbols) corresponding to the values of the liquid-vapor equilibrium, which will be discussed below as were obtained. For N=500, the regions between the values corresponding to the liquid-vapor equilibrium and the maximum and minimum value of pressure is called in the literature the metastable region.

In particular, the isotherm of 500 particles is a smooth function, but not so the case of N=2048, where there are regions well-localized in density and steps in the pressure, discussed in the recent literature [20]. However the corresponding isotherms of N=10000 and one million, show a marked reduction in the maximum and minimum

values of the pressure, which can be interpreted as the disappearance of the spinodal curve.

In the Figura 4 is shown for clarity a zoom of the region of low density.

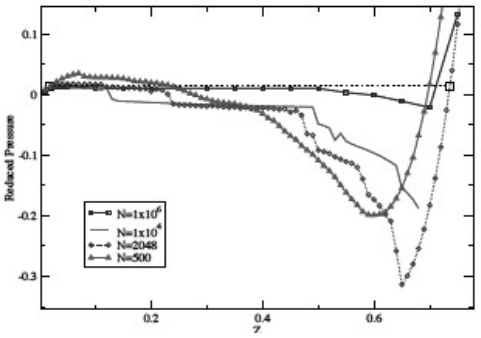


Fig. 3. The finite size effects in the isotherms of T=0.85

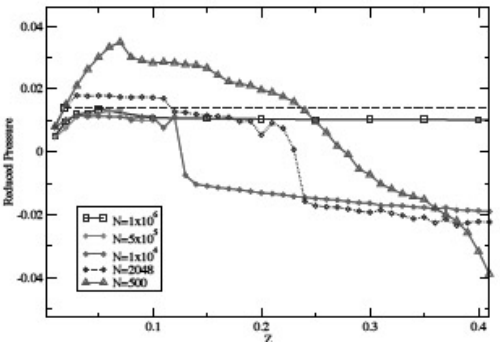


Fig. 4. Zoom of the low density region in the isotherm of T=0.85

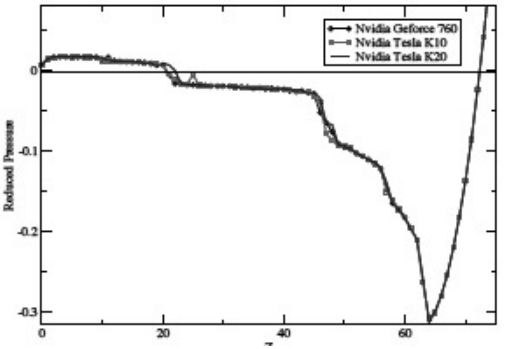


Fig. 5. Isotherm of N=2048 particles, simulated in the more commercial graphics cards

values of the densities are obtained by making slices in the three directions of bandwidth = 0:01 and have no meaning quantitative, qualitative only. A high density in the three curves represents a region in space where has condensed the fluids (liquid), this is a drop ball. A low density (constant function) represents the regions of gas respectively, which surrounds the drop of liquid. At higher densities that 0.5, the liquid-vapor interface is a spherical cavity, this is a bubble of gas (see bottom panel of Figure 6). Snapshots of these density profiles are shown in Figures 7 and 8 respectively.

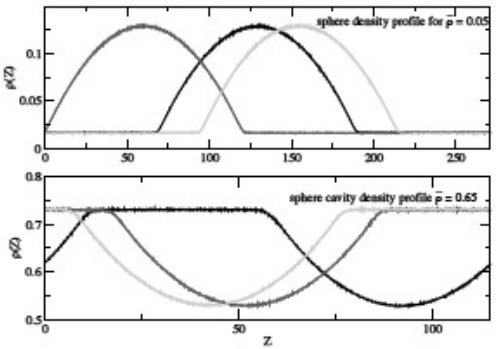


Fig. 6. Density profiles of a system of a million particles simulated for: T=0.85, mean density of 0.05 (top panel) and 0.65 (bottom panel), X-direcction (black line), Y-direccion (red line) and Z-direccion (green line)

The Figure 9 shows the profiles of density of a system of a million particles for average densities of 0.2,0.3 and 0.4. In these average densities the simulated system in the NVT ensemble must relax to a state of balance between two phases (liquid-vapor) with planar geometry, in this case only displaying the density profile in the direction in which the liquid-vapor interface is normal to the interface, Z-direction. In the other two directions the density profiles are constant functions. In these cases it is possible to measure both the density as the pressure of coexistence, which corresponds to the component of the pressure tensor in the normal direction to the flat interface. This is the way to achieve explicitly both, the coexistence pressure and densities, which are shown in the Figure 3 with the square symbols. Also in Figure 9 it can be seen, that as the average density increases, the amount

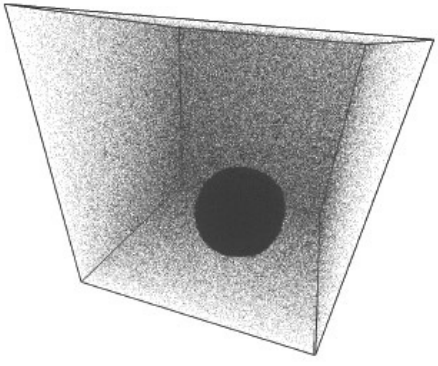


Fig. 7. Snapshot of a million particles for T=0.85 and average density of 0.05

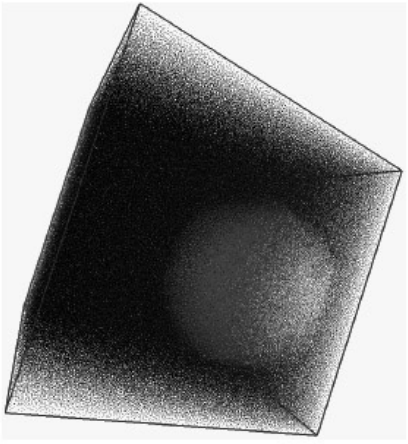


Fig. 8. Snapshot of a million particles for T=0.85 and average density of 0.65

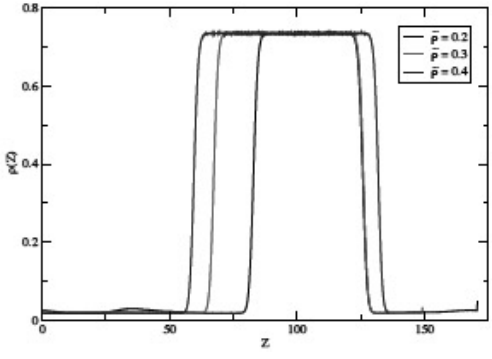


Fig. 9. Density profiles of a system of a million particles simulated for: T=0.85 , mean density of 0.20 (black line), 0.30 (red line) and 0.40 (blue line)

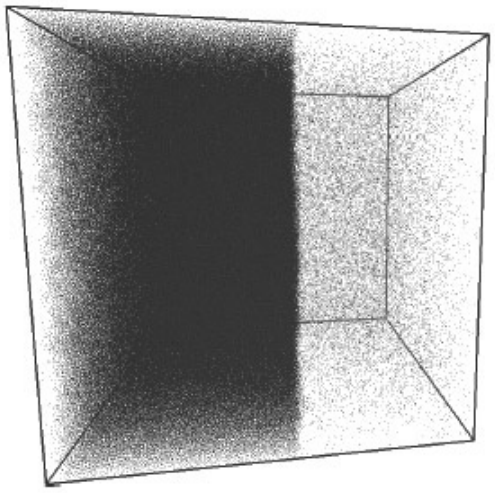


Fig. 10. Snapshot of a million particles for T=0.85 and average density of 0.40

used in the past, due to the limited computing capacity of the last years.

Acknowledgment

EDH thank CONACYT-Mexico [Project No. 178963] for the financial support given to this work.

References

[1] G. A. L. S´anchez, M. O. Carbajal, N. C. Cort´es, R. Barr´on, and J. A. A. Cedillo, "Comparative Study of Parallel Variants for a Particle Swarm Optimization Algorithm Implemented on a Multithreading GPU," vol. 7, no. 3, pp. 292–309, 2009.

[2] M. P. Allen and D. J. Tildesley, *Computer simulation of liquids*. New York, NY, USA: Clarendon Press, 1989.

[3] S. R. Alam, P. K. Agarwal, M. C. Smith, J. S. Vetter, and D. Caliga, "Using FPGA Devices to Accelerate Biomolecular Simulations," Computer, vol. 40, pp. 66–73, Mar. 2007.

[4] M. Giampapa, Y. Zhestkov, M. C. Pitman, F. Suits, A. Grossfield, J. Pitera, W. Swope, R. Zhou, and S. Feller, "Blue Matter : Strong Scaling of Molecular Dynamics on Blue Gene / L," *Proc. International Conf. on Computational Science (ICCS 2006)*, vol. 3992, pp. 846–854, 2006.

[5] V. S. Pande, I. Baker, J. Chapman, S. P. Elmer, S. Khaliq, S. M. Larson, Y. M. Rhee, M. R. Shirts, C. D. Snow, E. J. Sorin, and B. Zagrovic, "Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing.," *Biopolymers* , vol. 68, pp. 91– 109, Jan. 2003.

[6] M. Taufer, C. An, A. Kerstens, and C. L. Brooks III, "Predictor@Home: A "Protein Structure Prediction Supercomputer' Based on Global Computing," *IEEE Trans. Parallel Distrib. Syst*., vol. 17, pp. 786–796, Aug. 2006.

[7] L. Garc´ıa-Col´ın Scherer, *Introducción a la física estadística*. 2008.

[8] R. G. Mortimer, *Physical chemistry*. Harcourt/ Academic Press, 2000.

[9] D. Frenkel and B. Smit, *Understanding Molecular Simulation, Second Edition: From Algorithms to Applications (Computational Science).* Academic Press, 2 ed., Nov. 2001.

[10] A. L. Garcia, *Numerical Methods for Physics (2Nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2nd ed., 1999.

[11] W. C. Swope, “A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters,” *The Journal of Chemical Physics*, vol. 76, no. 1, p. 637, 1982.

[12] D. C. Rapaport, *The Art of Molecular Dynamics Simulation.* Cambridge University Press, 2nd editio ed., 2004.

[13] NVIDIA, “http://www.nvidia.com/object/molecular dynamics.html.”

[14] R. Murty and D. Okunbor, “Eficient parallel algorithms for molecular dynamics simulations,” *Parallel Computing*, vol. 25, pp. 217–230, Mar. 1999.

[15] S. Plimpton, “Fast Parallel Algorithms for Short-Range Molecular Dynamics,” *Journal of Computational Physics*, vol. 117, pp. 1–19, Mar. 1995.

[16] J. M. Zamora Fuentes, “https://github.com/ch3m/moldynamics.”

[17] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming.* Addison-Wesley Professional, 1 ed., July 2010.

[18] OpenGL, “http://www.opengl.org/.”

[19] DirectX, “http://www.microsoft.com/download/en/details.aspx?id=35.”

[20] L. G. MacDowell, V. K. Shen, and J. R. Errington, “Nucleation and cavitation of spherical, cylindrical, and slablike droplets and bubbles in small systems.,” *The Journal of chemical physics,* vol. 125, p. 34705, July 2006.

Semi-automatic Historical Climate Data Recovering Using a Distributed Volunteer Grid Infrastructure

Sergio Nesmachnow, Miguel Da Silva
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
{sergion,mdasilva}@fing.edu.uy

Abstract

This article presents the Digi-Clima project, whose main objective is to design a semi-automatic tool for digitalizing and recovering historical climate data using distributed computing techniques on grid and cloud infrastructures. The deploy of the processing tool for historical climate records on the Ourgrid middleware for grid and cloud is described, and the experimental evaluation using a volunteer-based Ourgrid infrastructure is reported. Accurate speedup values are reported for the distributed application.

Resumen

Este artículo presenta el Proyecto Digi-Clima, cuyo principal objetivo es diseñar una herramienta semi-automática para digitalizar y recuperar registros histórico-climáticos utilizando técnicas de computación distribuida en infraestructuras grid y cloud. Se describe el desarrollo de la herramienta de procesamiento de registros histórico-climáticos utilizando el middleware para grid y cloud Ourgrid y se reporta el análisis experimental utilizando una plataforma de computación voluntaria basada en Ourgrid. Valores razonables de speedup se reportan para la aplicación distribuida.

Keywords: *distributed computing; volunteer grid/cloud computing; historical climate records*

I. Introduction

Studying the behavior of climate variables through time is crucial for many different fields of science, and also for industrial production, disaster prediction, and warning systems, among many other applications. The potential impact of predicting the climate behavior is very important in the short-term for decision making, i.e. in agriculture, but also for long-term situations, i.e. to know whether a certain island will sink into the ocean in the next hundred years or not. For specific applications, such as the bombing control on sewer systems for large cities, or the prediction of water level for a river in a flood warning system, usually is it enough to know the actual conditions of meteorological variables or at last their recent past values. However, many other problems require knowing long-term series of climate variables records, and when considering the climate time scale, long-term could mean about 30 years and even more.

Taking into account the previous considerations, the scientific community is currently interested on recovering climate data stored through the years.

When the search looks back in time, climate records are very scarce and more difficult to recover. Furthermore, the preservation of climate records gathered in the pre-digital era is in danger of destruction.

In Uruguay, the systematic and extensive recording of climate variables has been performed from the early years of the 20th century, most of them stored in paper. All this data is of great value for science, but it has limited utilization in nowadays computerized systems, mainly due to the paper-format storage, and of course its preservation is in danger.

For the information to be applicable for a proper use on nowadays systems, a transcription process of the historical records is needed. Climate records have information stored in natural language in some cases. In other cases, just like in the case of pluviometer records, the transcription process can be automatized by using digitalization and software.

The previous approach is applied by the Digi-Clima project [1], which proposes developing an efficient application for digitalizing the scanned output of pluviometers, and storing the digital data in a database of historical climate records. The historical records are stored in graph paper bands containing the pluviometer output for a certain time period, which can be divided into intervals with each one looking like a continuous growing monotone function.

By using ad-hoc image-processing techniques, the Digi-Clima application requires about 10-15 minutes of execution time in a standard computer in order to process the image corresponding to one day of pluviometer records. Considering that more than a century of pluviometer records are available from meteorological stations situated in different locations across the country, applying high computing performance techniques is a promising choice−and in some cases a needed option such as the capital city Montevideo, with almost one third of the whole records−in order to get the results of the processing in reasonable execution times.

The Digi-Clima Grid project [2] proposes applying distributed computing techniques for efficiently performing the digitalization, by using grid/cloud computing platforms, including the Cluster FING, from Universidad de la República, Uruguay, the Latin-American/European grid infrastructure from the GISELA project [3], and the Latin-American grid/cloud computing platform from the Ourgrid project [4].

The Digi-Clima Grid project is carried out by the research groups on Computational Fluid Mechanics, Superficial Hydrology, and Numerical Computing Center at Engineering Faculty, Universidad de la República, Uruguay; and the Department of Computer Systems at Universidade Federal de Campina Grande (UFCG), Brazil.

The project has been partly funded and received financial support from the World Meteorological Organization; the National Direction of Water and the Departmental Government of Montevideo, Uruguay, and the Latin American and Caribbean Collaborative ICT Research Federation (LACCIR).

In previous articles, we have described the application of parallel scientific computing techniques to solve the problem tackled in the Digi-Clima project. In [5,6] the solutions developed by applying parallel computing techniques in cluster platforms and the GISELA grid infrastructure were

described. Both approaches were appropriate to solve the problem, achieving accurate speedup values when executing in dedicated computing infrastructures. In this article, we describe in detail an approach based on using volunteer-computing platforms.

The main contributions of the research reported in this article are: i) the application of high performance scientific computing techniques to solve the problem of processing and digitalizing historical climate records is described; ii) the development of the software application over a volunteer grid computing infrastructure in Argentina, Brazil, México, and Uruguay is presented; iii) the experimental analysis performed using a set of representative images over computing elements located in the aforementioned four countries is reported.

The rest of the article is organized as follows. Section II describes the Digi-Clima project and reviews related works about historical climate data processing. Section III introduces the main concepts about volunteer grid computing and the Ourgrid middleware. After that, the approach using distributed computing techniques for solving the problem is presented in Section IV. Section V reports the experimental evaluation. The last section summarizes the conclusions and formulates the main lines for future work.

II. Digitalization Of Historical Climate Data

This section introduces the Digi-Clima project, describes the problem of digitalizing historical pluviometer records, and reviews the related works on the topic.

A. The Digi-Clima project

In Uruguay, the National Direction of Meteorology has kept systematic records of climate variables from the beginning of the 20th century, including tables of manuscript data, descriptive reports, and also graphic records of pluviometers and other devices. These data are very useful for understanding climate phenomena and for climate/weather forecasting using sophisticated numerical models. However, the analysis and utilization of the historical climate data is very difficult, mainly due to the paper-format in which they are stored. Furthermore, their preservation is in danger, because of the inherent deterioration of the material when the records are handled.

In order to guarantee the preservation of the historical climate records, the Digi-Clima project was conceived to provide a semi-automatic method for digitalizing the rain intensity records from weather stations across the country. The resulting data is stored in a database of historical climate records, making possible both the preservation and the utilization of the data in numerical models.

The rain intensity records are stored in millimeter-marked paper bands, in which a pluviometer recorded for a certain time period the amount of rain accumulated in a given zone of the country. Each band contains the pluviometer records, and on the other side manuscript annotations that indicate the dates of the beginning and the end of the register, as well as the scale used to compute the maximum value of the rain records for the band, among other significant information. Figure 1 describes the recording procedure for historic rain data.

The pluviometer records draw a continuous line, reasonably smooth and (piecewise)

monotonically increasing, indicating the amount of rain accumulated on the device. The value grows until the pluviometer capacity is reached. Then, a vertical fall to zero indicates that the device is emptied, and the measuring process starts again. Figure 2 presents an example of the historical rain records.

In the sample band in Figure 2, several of the inherent troubles that make the digitalization problem difficult are shown, including a discontinuous ink line, ink stains, different intensity levels on the record line, and other defects that are consequences of the measuring procedure using an analogic device. Furthermore, manuscript annotations are also present in the band to account for additional data, usually written by operators when the band was removed from the pluviometer.

The Digi-Clima application is a software program developed in the Matlab scientific programming language, for extracting the data of the temporal series for accumulated rain records from a pluviometer band just like the one in Figure 2. By using scientific computing techniques, including segmentation and chromatic separation methods, counting and interpolation, the Digi-Clima application is able to successfully perform the reconstruction of the temporal series for a significant majority of the images available. The current version of the Digi-Clima application uses several functions from specific Matlab toolboxes, but right now it is been migrated to an open-source code using the Octave language.

Specifically, the Digi-clima application performs the following tasks:

1. color separation: separates the rain traces from the background and other information by color tagging, getting a black and white image which the active pixels from the rain traces;
2. band identification: analyzes the frame information layer to separate and scale the individual bands;
3. trace identification: for each band, identifies the traces of rain records as continuous lines;
4. trace analysis: analyzes each trace to obtain its footprint in pixel scaling, by using a spline fitting (Matlab) or a median estimator (Octave).
5. trace grouping: orders the traces in each band, since these records should be monotonic.
6. rain intensity computation: obtains the rain intensity data from the discrete time derivative of the grouped traces.

Figure 3 presents a diagram of the processing steps of the Digi-Clima application.

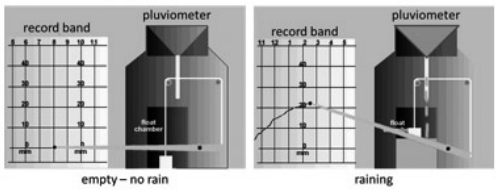


Figure 1. Recording pluviometer records in data record bands.

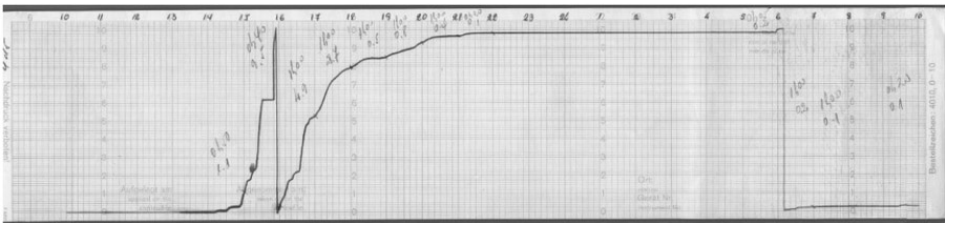


Figure 2. An example of a pluviometer record band, (Artigas, Uruguay, year 1980)

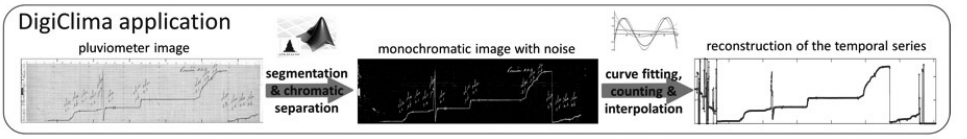


Figure 3. Diagram of the processing steps of the Digi-Clima application.

Other manuscript data are recorded at the reverse of each band, including beginning/end dates for the recording period, and the total amount of accumulated rain. All this information is needed to properly determine the pluviometer records. A specific application was designed using OCR techniques to extract/recover these data, and transcript them manually by a human operator.

The images corresponding to both sides of pluviometer bands were digitalized using scanners, and stored in a database in JPEG format. 8000 pluviometer bands corresponding to the cities of Montevideo, Rocha, Paysandú, and Artigas, in Uruguay, recorded from 1968 and 2010, were digitalized in the first stage of the project. Nowadays, the images database includes more than 30.000 bands from many meteorological stations, and new digitalized data are periodically required by the National Direction of Meteorology.

The Digi-Clima application requires from 10 to 15 minutes for processing each pluviometer image, depending on the complexity of the counting, trace identification, and interpolations needed for the time series reconstruction.

Using a traditional sequential approach, processing the images on the first stage of the project would require several months when executing on a standard computer. Taking into account these long execution times and considering that the national archive of historical pluviometer records has more than 50.000 bands, the Digi-Clima Grid project proposes using high performance computing techniques over large distributed grid/cloud computing infrastructures in order to cope with the processing tasks and solving efficiently the problem of digitalization of historical climate records.

B. Related Work

Some previous works have been reported in the literature, which have tackled similar problems than the one faced by the Digi-Clima Project, but using different approaches than the semi-automatic

processing proposed by Digi-Clima.

A number of those previous works on the topic were exclusively focused on analyzing specific climate phenomena for short time periods. Thus, these works analyzed significantly fewer volumes of data than the ones to be processed in the Digi-Clima project.

One of the first initiatives that tackled the problem of recovering historical climate data from a systematic perspective was the CLIWOC project [7], funded by the EU, and developed by researchers from Spain, UK, Netherlands, and Argentina. The main achievement of CLIWOC was building a database of climate observations based on hundreds of thousands records stored manually and in natural language from logbooks of Dutch, Spanish, French, and English vessels between 1750 and 1850.

Several work groups and human forces were involved in CLIWOC, each one specialized on the analysis of different climate data in the logbooks. The transcription included several data for each observation: date, geographic location, weather status, wind speed and direction, sea status, ice reports, and air temperature and pressure. CLIWOC also built a dictionary that allowed the researchers to unify the several criteria and terminology used in the logbooks, corresponding to four languages in more than 100 years.

A specific feature of the CLIWOC Project, which is also relevant for the Digi-Clima Project, is the quality control applied to the data transcription process. A random set of 5% of the transcriptions was selected for different periods; they were (manually) checked for possible significant errors, and in the presence of errors all the transcriptions for the same period were processed again.

On the same research line, the RECLAIM project [8] (under development) proposes continuing the work started in the CLIWOC project, by processing data that were not included in the original process. At the long-term, RECLAIM proposes processing records from the instrumental era, from 1853 to the current years.

The Old Weather Project [9] is focused on the transcription phase of climate records. It proposes extracting data from digital images of the documents by using volunteer human operators. In order to make more attractive the task and gathering more collaborators, the project developed an online game. A player in the game subscribes and chooses a ship, which corresponds to a real ship in the past with climate data logbooks to digitalize. The player starts with a low rank in the navy, and he earns experience and points by transcribing data, which allows him to be promoted to higher ranks. By using this game, the Old Weather project collected a group of human 'digitalizers'. They implement a simple control correction mechanism by performing multiple transcriptions of the same records and deciding by voting, thus achieving high quality results. This approach follows the "human computing" paradigm, which attempts to solve computing problems using human operators, and it is proposed as a promising line for future work in the Digi-Clima project.

The three previously commented projects are somehow related to Digi-Clima, since they propose the transcription of historical climate records. The projects demonstrate that the international research community is currently interested in the problem of recovering historical climate data stored in non-digital format.

The review of the related work allowed our

research group to understand the state-of-the-art about recovering historical data from manuscripts. However, none the similar projects reviewed have developed efficient methods for the automatic or semi-automatic transcription of data, because all of them are based on human operators or human computing techniques. Furthermore, there are no antecedents of using high performance or distributed computing techniques in order to speed up the recovering process. As a consequence, the Digi-Clima Grid project is a contribution in this line of research, by proposing the use of distributed computing techniques to efficiently solve the problem.

III. Grid Computing And The Ourgrid Middleware

This section introduces the paradigm of grid computing and describes the Ourgrid middleware for developing applications over volunteer-based grid/cloud platforms.

A. Grid Computing

Grid computing is a paradigm for parallel/distributed computing that allows the integration of many computer resources from diverse locations worldwide, to provide a powerful computing platform that allows solving applications with high computing demands. This paradigm has been increasingly employed to solve complex problems (i.e. e-Science, optimization, simulation, etc.) in the last ten years [10]

Grid infrastructures are conceived as a large loosely-coupled virtual supercomputer formed by many heterogeneous platforms of different characteristics, usually working with non-interactive workloads with a large number of files. Grid infrastructures have made it feasible to provide pervasive and cost-effective access to distributed computing resources for solving hard problems [11]. Starting from small grids in the earlier 2000's, nowadays grid computing is a consolidated field of research in Computer Science and many grid infrastructures are widely available. As of 2012, more than 12 PFLOPS are available in the current more powerful grid system, from the Folding@home project.

B. The OurGrid Middleware

OurGrid is an open source grid middleware based on a peer-to-peer architecture, developed by researchers at UFCG, Brazil [4]. This middleware enables the creation of peer-to-peer computational grids, and it is intended to speed up the execution of Bag-of-Tasks (BoT) applications.

The OurGrid architecture is built by aggregating several participants in a grid environment, allowing them to use remote and local resources to run their applications. OurGrid uses the eXtensible Messaging and Presence Protocol (XMPP), an open technology for real-time communication which powers a wide range of applications, including instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data. XMPP allows federation, it is Internet-friendly, and efficient, since several services can use the same XMPP server.

The main components of the OurGrid architecture are:

- the broker: implements the user interface to the grid. By using the broker, the users can submit jobs to the grid and also track their

execution. All the interaction between the user and the grid infrastructure is performed through the broker.

- the workers: used for processing the jobs submitted to the grid. Each worker represents a real computing resource. OurGrid workers support virtualization, and so they offer an isolated platform for executing jobs comprising no risks to the local system running the component.
- the peers: have a twofold role; from the point-of-view of the user, it searches and allocates corresponding computing resources for the execution of his jobs. From the point-of-view of the infrastructure (implicitly, for the site administrator) the peer is responsible for determining which workers can be used to execute an application, and also how they will be used. Normally, it is enough to have one peer per site. Communication between peers makes possible to execute jobs remotely; in case that the local resources are not enough for satisfying the requirements of a job, the peer seeks for additional resources available in remote sites.
- the discovery service: keeps updated information about the sites comprising the grid, and it is used to find out the end points that peers should use to directly communicate with each other.

All these components are integrated in a transparent way to the user, allowing the grid to provide a single-image of an infrastructure with a large computing power. A description of the Ourgrid architecture is shown in Figure 4.

The Ourgrid middleware provides support for implementing the volunteer computing paradigm [12,13]. Volunteer computing is based on individual users making available their computing resources to execute applications and projects. The first volunteer computing projects were proposed in the mid-1990s. In 1999, the well-known SETI@home and Folding@home distributed projects were launched. Both became pioneering projects to demonstrate the power of gathering volunteer computing resources to solve very large scientific problems.

The main features of volunteer computing are: i) often unaccountable and potentially anonymous users voluntarily provide their resources for computation; ii) users can join and leave the volunteer computing platform at any time; iii) users "credits" are accounted in order to know how much computing time h in the systemas been used and provided by every user; iv) replication is usually applied as the preferred technique for fault-tolerance, in order to cope with incorrect results or anomalous events that occur when volunteers unexpectedly leave the system.

The middleware for volunteer computing is a software layer that provides support for creating, managing, and using the volunteer distributed computing infrastructure, independently from the scientific computing applications to execute. The Berkeley Open Infrastructure for Network Computing (BOINC) [14] is the most widely used middleware system for volunteer computing. The general architecture of a middleware for volunteer computing is like the one presented for Ourgrid: a client program runs on the volunteer side, which periodically contacts servers over the Internet, requesting jobs and reporting the results of completed jobs. In Ourgrid, the standard volunteer

computing model is extended to support a full P2P architecture. This feature allows Ourgrid to be used in cloud infrastructures too.

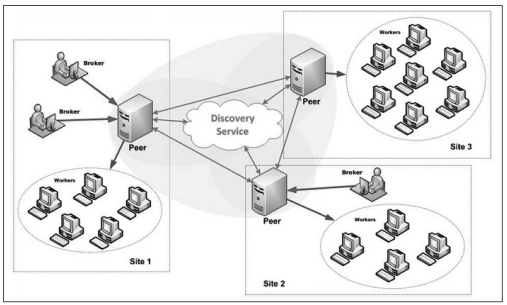


Figure 4. Description of the Ourgrid architecture.

IV. Digi-Clima Grid Using Ourgrid

This section presents the proposed solution for recovering historical climate records over a distributed computing infrastructure using Ourgrid. A conceptual description of the parallel model is presented, and the specific details of the implementation over the volunteer grid platform using Ourgrid are described.

A. Parallel model

The proposed approach is based on considering the Digi-Clima application (as programmed in Matlab) as a black box, in order to take full advantage of the distributed computing resources in the volunteer platform.

The processing of each digitalized image is independent of the other images, so a traditional data-parallel approach can be applied on the input data, using a sets-of-images partition.

The parallel model is based on executing multiple instances of the Digi-Clima application simultaneously, working on different input data, without communications between the distributed processing tasks. Communications are only used to get the images and to keep track of the shared information about the images processing status, in order to avoid processing the same image more than once.

In this parallel model, the only overhead in the processing is due to the mutual exclusion between processes when loading images from the image database and when accessing the shared information about the processing status. These accesses are performed before (load, access to shared information) and after (set processing status) processing each image, but the negative impact of these communications on the computational performance of the distributed application can be reduced by assigning a BoT (in our application case, is a bag of images) to each distributed process executing in a computing element of the volunteer grid.

A master-slave parallel architecture was applied to execute Digi-Clima over the volunteer grid. The master process launches and assigns the work to several slave processes that perform the processing by execution the Digi-clima application on a set of images assigned by the master, execution on computing elements of the volunteer grid infrastructure. Once a slave finishes its work, the master process collects the output data to include it in the final job output. Figure 5 presents a schema of the parallel model.

The proposed strategy allows scaling up achieving almost-linear speedup in case of applying an efficient access to the database for load

balancing and synchronization.

The black-box approach allows focusing on the data parallelism itself, making the grid development independent of the Digi-Clima application. This strategy allows the continuous improvement in the processing tool, by using updated versions of the image-processing algorithms, which are in constant improvement by our research group.

There are two options to execute the Digi-Clima application: i) interpreted Matlab execution and ii) execution of a binary application compiled for a given platform by using a free runtime (the Matlab Compiler Runtime). In addition, the application can be run using a single thread or using multiple threads.

The grid implementation presented in this article supports using both the interpreted and the compiled executions of Digi-Clima, as well as both the single and multithread versions. In order to analyze the benefits of applying distributed computing techniques, we have decided to report the results using the single-thread compiled option in the experimental analysis section. This decision allows working with the black-box approach, since it does not require performing modifications to the program code to handle threads, external functions and/or intermediate data.

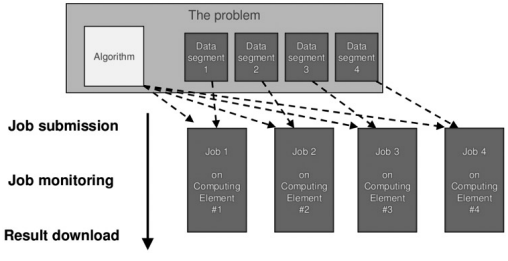


Figure 5. Data-decomposition approach used in the distributed Digi-Clima Grid application

B. Implementation in Ourgrid

One of the main features of the Ourgrid middleware consists in working with virtual machines, allowing a flexible way for efficiently using the computing resources in the distributed volunteer infrastructure [15].

Following this approach, the parallel/distributed implementation of Digi-Clima in Ourgrid is based on virtual machines for implementing the domain decomposition strategy for parallel/distributed computing. Images are stored in a database in our local Ourgrid node at Cluster FING, Universidad de la República, Uruguay [16]. Processing tasks execute in a distributed volunteer-based Ourgrid infrastructure created for this project, including computing resources from four Latin-American countries (see next section for a description of the grid infrastructure). In the proposed master-slave implementation, the master process assigns a set of images to each slave for processing.

A general diagram of the master-slave implementation for executing Digi-Clima in the distributed volunteer based Ourgrid infrastructure is presented in Figure 7.

Each slave process executes a pre-configured virtual machine that is turned on in order to process images using the Digi-Clima application. Slaves knows how to access the database of images and also how to get other mandatory files that they will necessary have to process images; such files include both the Digi-Clima application and the MATLAB Compiler Runtime (both of them available for 32-bit and 64-bit architectures).

The master process searches for images that have not yet been processed and assign them to

a slave process. Images are assigned based on partitions of the whole set of images, following the BoT approach. The corresponding slave process, which is responsible for processing the bag of images, receives the images IDs and uses this data to download images, keep track of their processing and update the database.

After selecting a bag of images for processing, the master process starts a new thread (the execution thread) which is in charge of launching the slave processes to digitalize the set of images selected. The main master thread (the monitoring thread) remains in execution, controlling the slaves and checking for possible errors in the processing or in the distributed environment.

Before creating a new thread, the master execution thread creates on-the-fly a Job Description File (JDF) with proper information for the images processing. This JDF will be used by the corresponding slave in order to run an Ourgrid task. An example of JDF created on-the-fly for Digi-Clima is shown on Figure 6. A timestamp is used to avoid issues related to repeated names for images. Two variables provided by the Ourgrid middleware are also used to make easier identify the slave process responsible for processing such images.

No special requirements are specified for the processing in the example presented in Figure 6. Nevertheless, the Ourgrid middleware allows specifying requirements inside JDF files to comply with specific user's needs (for example: a 64-bits architecture for the host, etc.). These specifications will be taken into consideration when the middleware search for computing resources for allocation.

```
job:
label: slave_digiclima_13948073

task:

init: put slave_digiclima.pl
slave_digiclima_13948073.pl

remote: perl slave_digiclima_13948073.pl
<parameters>

final: get slave_digiclima_13948073.tar.bz2
slave_digiclima_13948073_$JOB_$TASK.tar.bz2
```

Figure 6. An example of JDF for Digi-Clima in Ourgrid.

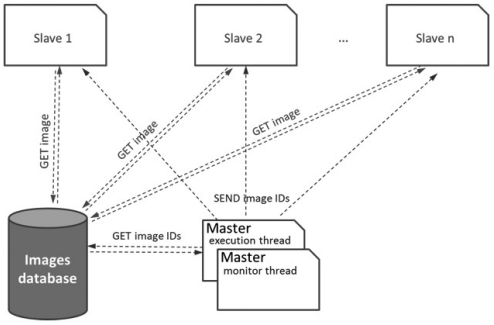


Figure 7. Diagram of the master-slave algorithm for Digi-Clima in Ourgrid

The JDF format is simple. A group of stanzas define the items and features needed for the processing, including: i) the files to be transferred in and out of slave machines; ii) the commands to be executed; iii) the labels to be used to identify jobs and iv) other features. A complete list of options available can be found in the Ourgrid documentation.

The example shown in Figure 6 will start a job labeled 'slave_digiclima_13948073'. This job will transfer the file 'slave_digiclima.pl' to the Slave and its name on the remote host will be 'slave_digiclima_13948073.pl'. Once the transfer is performed, this application (a perl script) will be executed by the slave using a group of parameters.

For each new thread launch by the master execution process the parameters are different, and they are a timestamp (when the thread was created) and the IDs of the images to be processed by the slave.

Slave processes are in charge of processing images. For such a task, each slave downloads the files needed to handle the processing request. Files are accessed through a local webserver. In case the Digi-Clima application or the Matlab Compiler Runtime has been used before, they will not be downloaded again. In this situation only the images are transferred, thus improving the execution time and the computational efficiency of the parallel/ distributed algorithm.

Each image can be in four different stages: (1) waiting, (2) assigned, (3) processing and (4) done. Figure 7 presents the finite state machine that characterize the image processing. Transitions can be carried out by both master and slaves processes; for instance, once a image is assigned to a specific slave by the master, it will marked as assigned.

Errors in the processing are handled in two levels: (1) Ourgrid middleware errors and (2) the master-slave solution proposed. The Ourgrid middleware works with replicas of each defined task, and in case of error, only after all the replicas have failed, Ourgrid will stop trying to process a specific image. In such a case, the slave process will mark the image as 'waiting', in order to be processed again (by another slave) in the future.

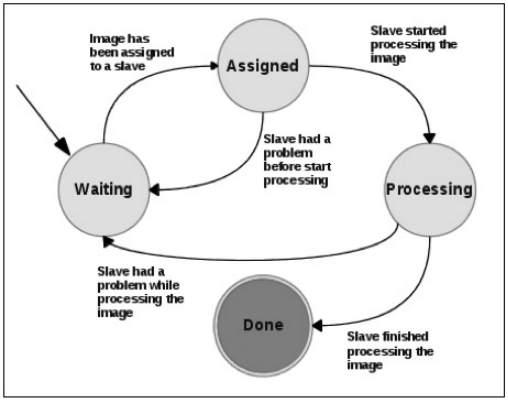


Figure 8. Finite state machine used in the master-slave implementation.

As mentioned before, two key files must be transferred to guarantee a correct execution: i) the Digi-Clima application and ii) the Matlab Compiler Runtime. The Digi-Clima application is about 2 MB, but the Matlab runtime is significantly larger (about 220 MB), so it could have a significant negative impact on performance. To avoid transferring the runtime each time a new process is launched, at the beginning of execution the slave process verify if a local version of the runtime is already available on the remote host, and it also checks if it is suitable for the Digi-Clima algorithm provided. Being so, only images have to be transferred.

For the implementation of master and slave processes, the Perl programming language was chosen. Perl is highly available in most Linux distributions, and the virtual machines used by Ourgrid already have an operative version of this programming language. Perl offers a large set of modules that make possible the interoperation of applications with very different features, including thread handling, file compression, database access

and HTTP connections.

A database (using the open source mysql manager) is used to keep track of the processed images and their corresponding state. The database is structured in rows corresponding to each single image. The data stored for each image are: image ID, file name, file URL (used by the slave to download the image), current state of the image, output path and descriptive notes.

The status in the database is updated by each slave process, when one of the events in the finite state machine in Figure 8 occurs. The monitor thread of the master process is continuously monitoring the database to check for new slave threads to be launched, or if there are images to be assigned. Once all the images are marked as 'done', the monitor thread stops and the parallel program ends the execution.

V. Experimental Evaluation

This section introduces the distributed volunteer computing platform built over Ourgrid to evaluate the proposed algorithm. After that, the experimental analysis is described and the main efficiency results are commented.

A.Volunteer Computing Infrastructure

A dedicated volunteer computing infrastructure was built using computing resources from four Latin America research institutions: Universidad de Buenos Aires (UBA) in Argentina, Universidade Federal de Campina Grande (UFCG) in Brazil, Universidad Veracruzana (UV) in México, and Universidad de la República (UdelaR), in Uruguay, as it is shown in Figure 7.

The volunteer computing infrastructure gathers a heterogeneous collection of computing resources, including:

- Argentina: IBM Express x3650 M4, Xeon E5-2620 12 cores, (2.0GHz), 128GB RAM, Gigabit Ethernet.
- Brazil: : Intel i5-3470S, 4 cores (2.9 GHz) & i7-2600, 8 cores (3.40GHz), 8 GB RAM, Gigabit Ethernet.
- México: Intel i5-3470S, 4 cores (2.3 GHz), 48 GB RAM, Ethernet.
- Uruguay: AMD Opteron 6172, 24 cores (2.1 GHz), 24 GB RAM, Gigabit Ethernet, from Cluster FING [16]

Figure 9. The dedicated volunteer Ourgrid infrastructure used in the Digi-Clima grid project.

B. Computational Efficiency Results and Analysis

Table I reports the times required to process a prototype database containing 400 images. The time required to process an individual image

differs, depending on the specific characteristics of each image. Processing times between 4 and 20 minutes have been observed.

Four different numbers of Ourgrid worker processes were used as shown in Table 1. Using one worker the images were processed sequentially. When using 8, 16, and 24 workers, Ourgrid allocated the computing resources as required. In these cases, each worker is assigned a bag of images to process. We performed experiments using bags containing 5 and 10 images. This way, different numbers of jobs have been scheduled in the Ourgrid infrastructure; allowing performing experiments with different bag sizes, and analyzing the behavior of the parallel algorithm in each case.

Table I. Execution Times And Speedup Analysis

	metrics			
	bag size = 5 images		*bag size = 10 images*	
#processes	***time***	***speedup***	***time***	***speedup***
1 (sequential)	60 hours	-	60 hours	-
8	10 hours	6	12 hours	5
16	8 hours	7.5	9 hours	6.6
24	4.5 hours	13.5	6 hours	10

The results on Table I demonstrate that using a parallel algorithm on a grid environment allows reducing the processing times considerably. In addition, using a smaller bag of images (5 images in each assignment) gets a better level of parallelism, thus improving the execution times.

The speedup analysis reported in Figure 10 shows that promising computational efficiency values and scalability behavior are obtained when using up to 24 slave processes. Taking into account the obtained efficiency results, the current whole database of 30.000 pluviometer records can be digitalized in about 14 days, while the sequential processing would demand more than 187 days (more than 6 months).

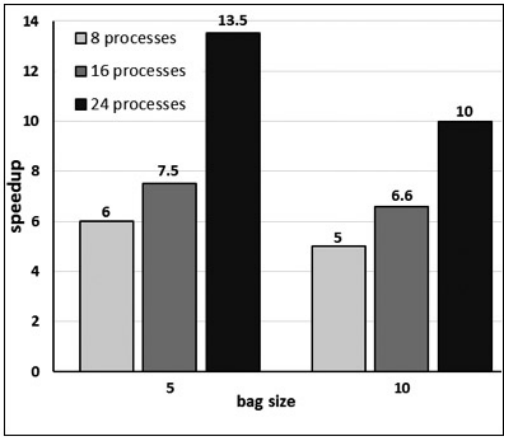


Figure 10. Speedup analysis.

VI. Conclusions

This article presented the application of image-processing techniques over a distributed volunteer grid system for the semi-automatic recovering of historical rain data in Uruguay.

The experimental analysis performed over a volunteer infrastructure including computing resources from four Latin America research institutions, demonstrated that accurate speedup values are achieved when applying a domain-decomposition master-slave implementation for the image processing. Using this approach, the current whole database of pluviometer records can be digitalize in about 14 days, instead of the 187 days required by a traditional sequential implementation. The implemented solution also demonstrated the simplicity and the versatility of the Ourgrid middleware for distributed computing.

The main lines for future work include improving the image processing techniques within the Digi-Clima application, possibly by using graphic processing units and human computing techniques for verification, and applying the distributed computing approach to solve similar problems dealing with information recovery.

Acknowledgment

The work of S. Nesmachnow is partly supported by ANII and PEDECIBA, Uruguay. The Digi-Clima Grid project is funded by the LACCIR Virtual Institute.

References

[1] Digi-Clima. Online www.fing.edu.uy/inco/grupos/cecal/hpc/digiclima, January 2014.

[2] G. Usera, S. Nesmachnow, F. Brasileiro, M. Da Silva, and S. García. "Recovering historical climate records through grid computing". In *Latin American eScience Workshop*, São Paulo, Brazil, 2013.

[3] GISELA. Grid Infrastructures for e-Science virtual communities in Europe and Latin-America. Online www.gisela-grid.eu, January 2014.

[4] W. Cirne, F. Brasileiro., N. Andrade, L. Costa, A. Andrade, R. Novaes, and M. Mowbray. "Labs of the world, unite!!!" Journal of Grid Computing, 4:225–246, 2006.

[5] S. García, S. Iturriaga, and S. Nesmachnow. "Scientific computing in the Latin America-Europe GISELA grid infrastructure" In *Proceedings of 4th HPC Latin America Symposium*. Córdoba, Argentina, 2011.

[6] S. García, S. Iturriaga, S. Nesmachnow, M. da Silva, M. Galnarés, and G. Rodriguez. "Developing parallel applications in the GISELA grid infrastructure". In *Proceedings of the Joint GISELA-CHAIN Conference*, 9–16, México, 2012.

[7] R. García-Herrera, G. Können, D. Wheeler, M. Prieto, P. Jones, and F. Koek. "CLIWOC: A climatological database for the World's oceans 1750–1854". Climatic Change, 73:1–12, 2005.

[8] C. Wilkinson, S. Woodruff, P. Brohan, S. Claesson, E. Freeman, S. Lubker, C. Marzin, and D. Wheeler. "Recovery of logbooks and international marine data: the RECLAIM project", International Journal of Climatology, 31(7):968–979, 2011.

[9] Old weather project. Online www.oldweather.org, January 2014.

[10] J. Joseph and C. Fellenstein. Grid Computing. Pearson Education, 2004.

[11] I. Foster and C. Kesselman. The Grid 2: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann, 2003.

[12] O. Nov, D. Anderson, and O. Arazy. "Volunteer computing: a model of the factors determining contribution to community-based scientific research". In *Proceedings of the 19th international conference on World wide web* . New York, USA, 741-750, 2010.

[13] D. Anderson and G. Fedak. "The Computational and Storage Potential of Volunteer Computing". In *Proceedings of the Sixth IEEE International Symposium on Cluster*

Straightforward DSP Algorithm Suitable for GPU Computation

Raúl G. Hazas-Izquierdo[a]., José L. Rodríguez-Navarro[b]., Alan D. Faz Gutiérrez[c]., Raymundo Salazar Orozco[c].

[a]*Universidad de Sonora*
[b]*Centro de Investigación Científica y de Educación Superior de Ensenada*
[c]*Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Sonora Norte*
Email: rhazas@difus.uson.mx, navarro@cicese.mx, alanfaz@hotmail.com, salazarorozco@gmail.com

Abstract

Current Graphic Processing Units (GPUs) have achieved a deep level of programming parallelism. Filtering of discrete sequences often requires an inordinate amount of computing resources. Infinite Impulse Response (IIR) filter structures often are broken down into simpler and concatenated elements called biquads for ease of implementation. Analysis of data flow across one such simplified structure, prompts of feasible digital signal processing (DSP) algorithm simplification. Favorable comparison of outcome brought up by forthright C-based implementation of prospective DSP algorithm versus industry standard application suggests a filtering method likely adaptable for CUDA coding and execution.

Keywords: *DSP, Filtering, GPU, CUDA Programming.*

1. Introduction

Digital treatment of sequences resulting from a sampled physical phenomenon has many applications, particularly in speech, image and video processing, as well as on environmental and life sciences. A digital filter is a formula for going from one digital sequence to another whose characteristics have been suitably modified. It may exist as an equation on paper, as a small loop in a computer subroutine, or as a series of integrated circuit chips properly connected [1].

Migration of signal processing functions from Digital Signal Processors (DSPs) to Graphical Processing Units (GPUs) is currently under consideration due to the deep level of integration achieved in newer generations of GPUs aimed at solving massively parallel computations [2]. In hardware, typical DSPs implement low-order filter structures that when combined together (in cascade for example) produce a specified higher-order filter's characteristics [3, 4].

In this paper we suggest a filtering algorithm suited to exploit GPUs' massive programmable pipelines based on the analysis of the data flow across a low-order filter structure.

This paper is organized as follows: Section 2 discusses IIR Filters and biquad structure realizations. The analysis of the data flow across

a biquad structure is described in Section 3. The proposed method is elucidated in Section 4. Experimental results from the filtering algorithm implementation are shown in Section 5. Concluding remarks are given in Section 6.

2. IIR Digital Filters, Biquad Structures in Direct Form I and II

Realizing a digital filter corresponds to computing the output of the filter in response to a given input. For Infinite Impulse Response (IIR) filters [5], the general input-output equation is of the form

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] + \sum_{k=1}^{N} a_k y[n-k] \qquad (1)$$

Where the feed-forward coefficients are b_k while a_k are the feedback coefficients.

If we assume a causal filter with initial conditions equal to zero, the z-transform of (1) becomes

$$Y(z) = X(z)\left(\sum_{k=0}^{M} b_k z^{-k}\right) + Y(z)\left(\sum_{k=1}^{N} a_k z^{-k}\right) \qquad (2)$$

$$Y(z)\left(1 - \sum_{k=1}^{N} a_k z^{-k}\right) = X(z)\left(\sum_{k=0}^{M} b_k z^{-k}\right)$$

The filter's transfer function $H(z)$ is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_N z^{-N}} = \frac{B(z)}{A(z)} \qquad (3)$$

Assuming $M \leq N$, $H(z)$ could be factored as

$$H(z) = \prod_{k=1}^{N_s} \frac{b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}} \qquad (4)$$

Where $N_s = \lfloor (N+1)/2 \rfloor$. Each of the second order kth factors in the product above constitutes a biquadratic section or biquad [6].

Bringing forth a linear and time invariant discrete-time system from a difference equation like (1) requires that the input, output, and intermediate sequence values be available at once. A block diagram representation of a biquad section is shown below.

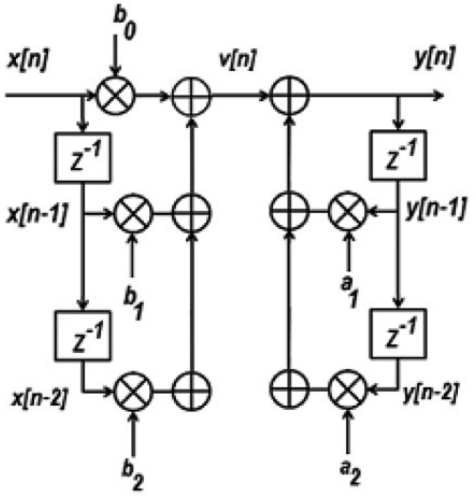


Figure 1 Biquad section Direct Form I realization

Without loss of generality, we could rearrange the filter structure if we factor (3) as:

$$H(z) = \frac{B(z)}{A(z)} = \frac{1}{A(z)} * B(z) \qquad (5)$$

The above would result in the following structure:

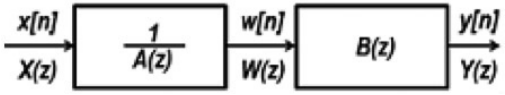


Figure 2 Block diagram of a rearranged biquad filter section

Therefore we could envision the procedure as performing the recursive filtering first, and then carrying out the ensuing moving average computation, which would result in a Direct Form II realization as shown in figure 3 where minimum storage and computing requirements are being met [7].

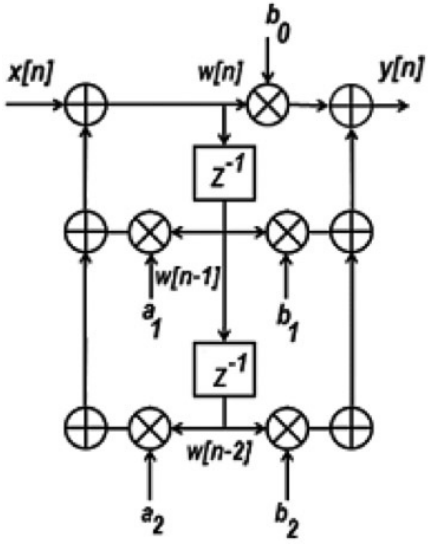


Figure 3 Biquad section Direct Form II realization

3. Data flow across the biquad structure

We follow the flow of a given sequence $x[n]$ across the structure as shown in the table below:

Table 1 Data flow across the biquad structure

n	$n<0$	0	1	2
$x[n]$	0	$x[0]$	$x[1]$	$x[2]$
$w[n]$	0	$x[0]$	$a_1x[0]+x[1]$	$a_1{}^2x[0]+a_1x[1]+x[2]+a_2x[0]$
$w[n-1]$	0	0	$x[0]$	$a_1x[0]+x[1]$
$w[n-2]$	0	0	0	$x[0]$
$y[n]$	0	$b_0x[0]$	$b_0(a_1x[0]+x[1])+b_1x[0]$	$b_0(a_1{}^2x[0]+a_1x[1]+x[2]+a_2x[0])+b_1(a_1x[0]+x[1])+b_2x[0]$

We recognize how sequence $x[n]$ is being recursively shaped by the feedback coefficients a_1 and a_2

While $w[n]$ takes its place as the input to the feed-forward section, where an apparent inner vector product takes place i.e.

$$w[n] = x[n] + a_1w[n-1] + a_2w[n-2]$$

4. Proposed method

There has been a great deal of previous work on the parallel implementation of digital filtering by parallel block processing methods [8]. Yet, most of the speed-up efforts are hindered because

$$y[n] = b_0w[n] + b_1w[n-1] + b_2w[n-2].$$

of data dependency problems where re-shuffling the intermediate results take up the bulk of computation time [9, 10]. Clever algorithms have been recently devised on GPUs to try to uncouple the data dependencies by multi-block processing paired with fast look-ahead [11].

Due to the simplicity of the biquad structure, our filtering algorithm is rather straightforward: for a given input sequence $x[n]$, replace the contents at

each memory location by the contents of $w[n]$, the recursively processed intermediate sequence. On completion, perform a vector inner product with the b_k feed-forward coefficients.

We anticipate such algorithm could take advantage of the massive parallelism, shared memory and thread synchronization in a GPU and be easily coded and put into effect by well-established approaches [12].

5. Experimental results

Matlab[1] and C++ programs were devised to create a series of input sequences consisting of a sampled sinusoid signal augmented with random values. The C++ program was coded according to the algorithm described above. Filter coefficients were chosen from textbook examples for low-pass (LPF), and high-pass (HPF) filters [7 ibid, 13]. Output sequence lengths were adjusted for comparison and display.

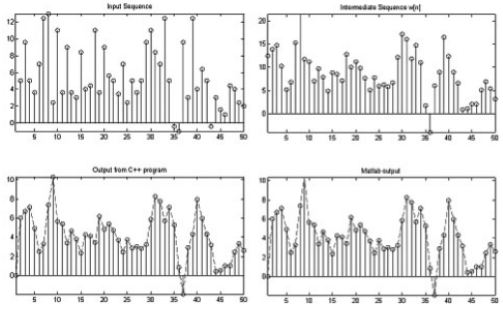


Figure 4 Output from Chebyshev 2nd. order LPF filter (see explanation below)

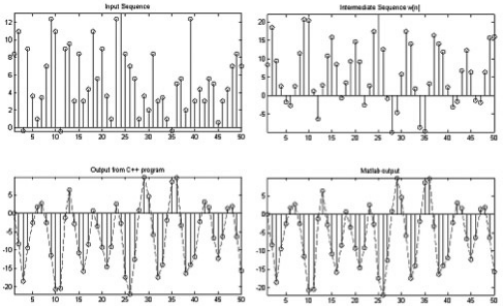


Figure 5 Output from 2nd. order HPF filter (see explanation below)

On figures 4 and 5, top left panes show the input sequences. Top right panes show the intermediate $w[n]$ sequences. Bottom panes show the point-by-point and contoured output for both the proposed IIR filter algorithm implementation (bottom left panes), and from Matlab filter(bb,aa,x) function (bottom right panes).

Comparing both figures' bottom panes we observe that the outputs from the proposed filter implementation completely match the ones delivered by Matlab. We also observe that the filter's characteristic is portrayed by the w[n] sequences (top right panes): it either suppresses (LPF) or enhances (HPF) high rate transitions. The filters' ensuing moving average stages trim the sequences contour.

6. Conclusions

We have implemented a forthright filtering algorithm whose response is in agreement with industry standard application. Although recursive filtering incurs in a data dependency problem, we uncoupled the recursive operation from the feed-forward phase and came up with a method we

presume could take advantage of GPU's massive parallelism and synchronization to readily carry out DSP processes.

7. Acknowledgements

Authors would like to thank the technical and material help provided by the Área de Cómputo de Alto Rendimiento de la Universidad de Sonora (ACARUS), and by the Dirección de Informática support desk personnel at Instituto Tecnológico y de Estudios Superiores de Monterrey campus Sonora Norte.

8. References

[1] Smith III, Julius O., Introduction to Digital Filters with Audio Applications, W3K Publishing, 2007.

[2] Hallmans, D., Sandstrom, K., Lindgren, M., Nolte, T., GPGPU for industrial control systems, IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), 2013, pp. 1 - 4.

[3] Karam, Lina J., James H. McClellan, Ivan W. Selesnick, C. Sidney Burrus, Digital Filtering, Chapter 11, The Digital Signal Processing Handbook, Vijay K. Madisetti and Douglas B. Williams, editors. CRC Press LLC, 1998.

[4] Srisangngam, P., Chivapreecha, S., Dejhan, K., Even Order Biquad Digital Filter Design Using Pascal Matrix, IEEE International Symposium on Communications and Information Technologies, ISCIT 2008, pp. 327 – 330.

[5] Lutovac, Miroslav D., Dejan V. Tošić, Brian L. Evans, Filter Design for Signal Processing Using Matlab and Mathematica, Prentice Hall, Inc., 2001.

[6] Chassaing, Rulph, Donald Reay, Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK, Second Edition, John Wiley & Sons, Inc., 2008.

[7] Oppenheim, Alan V., Ronald W. Schafer, Discrete-Time Signal Processing, Third Edition, Pearson Education, Inc., 2010.

[8] Wonyong Sung, Mitra, S.K., Jeren, B., Multiprocessor implementation of digital filtering algorithms using a parallel block processing method, IEEE Transactions on Parallel and Distributed Systems, Volume: 3, Issue: 1, 1992, pp. 110 – 120.

[9] Proakis, J.G., Manolakis, D.G., Digital Signal Processing: Principles, Algorithms, and Applications, Third Edition, Prentice Hall, Inc., 1996.

[10] Embree, P.M., Danieli, D., C++ Algorithms for Digital Signal Processing, Second Edition, Prentice Hall, Inc., 1999.

[11] Lee, D.H., Sung W., GPU based implementation of recursive digital filtering algorithms, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 2684 – 2687.

[12] Sanders, J., Kandrot, E., CUDA by example: an introduction to general-purpose GPU programming, Pearson Education, Inc., 2010.

[13] McClellan, J.H., Schafer, R.W., Yoder, M.A., DSP First: A Multimedia Approach, Prentice Hall, Inc. 1998.

Performance Evaluation of Cellular Genetic Algorithms on Gpu

Migdaled López-Juárez, Ricardo Barrón Fernández and Salvador Godoy-Calderón
Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional (IPN)
Mexico City, Mexico

Abstract

Cellular genetic algorithms were initially designed for working on massively parallel machines. However, their performance can be affected by population size, the kind of neighborhood and other issues about implementation itself. Nowadays, GPU cards allow us to execute a task in their fundamental computing units (called threads), which are completely independent among them. In addition to use concurrent threads, there are also memories that help to improve the performance. In this paper we evaluate the performance of two models of cellular genetic algorithms, which find the optimal number of partitions for a data set, using a cluster validation index as objective function.

Keywords: *cellular genetic algorithms, parallelism, GPU*

1. Introduction

GENETIC algorithms (GA) described by Holland have led to the solution of search and optimization problems [1]. They are based on the genetic process of living organisms where populations evolve, throughout generations, according to the principles of natural selection and the survival of the strongest. Years later, Whitley introduced the term cellular genetic algorithms (CGA) which were initially designed to run on massively parallel machines [2].

CGA can have an arbitrary structure, but is commonly used a two-dimensional grid structure, which facilitates its implementation on parallel processors [3].

In this paper, we evaluate the performance of CGA on a Graphics Processing Unit (GPU). The problem to be solved is to find an optimal partitioning of a dataset, using the Davies-Bouldin index as objective function. Solving this problem is very useful in algorithms such as k-means, since it is the starting point for determining the number of representatives in each partition or class. In contrast with proposed works [4, 5] we suggest two parallel implementations designed to run on a GPU.

This paper is structured as follows. CGA and CGA models that can be implemented in parallel are described in Section 2. Section 3 describes the measures used to evaluate the performance of CGA. The results of the experiments are presented in Section 4. Finally, in Section 5 we present our conclusions.

2. Parallel Cellular Genetic Algorithms

2.1 Cellular genetic algorithms

CGA are part of the evolutionary algorithms, in which each of the individuals in the population communicates with individuals closest to them forming neighborhoods [2, 6]. The neighborhood can be linear type or compact type, as shown in Figure 1.

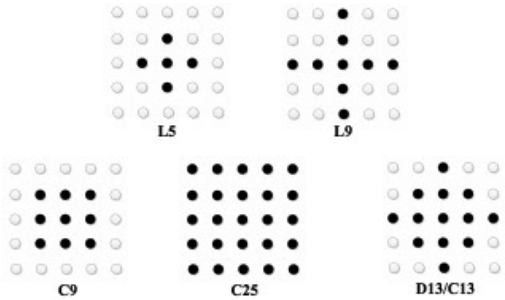


Figure 1. Types of neighborhood used in CGA for a population of 25 individuals. From top to bottom, we can see the linear neighborhoods L5 and L9 with 5 and 9 neighbors respectively. Compact neighborhoods C9, C25 and C13 with 9, 25 and 13 neighbors, respectively, are presented in the bottom. The C13 neighborhood is also known as D13.

The overlap among neighborhoods enables a slow diffusion of the best solutions produced, while the diversity is maintained for longer time [6, 7], as seen in Figure 2.

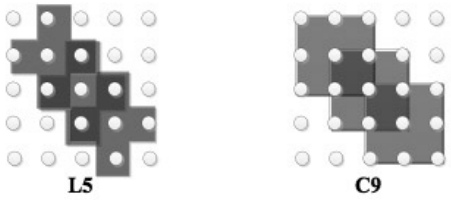


Figure 2. The overlap can be observed in L5 and C9 neighborhoods of CGA with a population of 25 individuals. The neighborhood overlapping provides to CGA an implicit migration mechanism, thus, the best solutions are smoothly spread through the population

We propose the following aspects in order to solve the problem of cluster validation:

- Each individual, who is part of our search space, is encoded by coordinates (centroids) of real numbers in Rn and represents clusters. This encoding is suitable for this kind of problems [8]. A chromosome represents each individual and its length is variable. The length depends on the number of clusters and the size of the centroids. The chromosome structure used is illustrated in Figure 3.

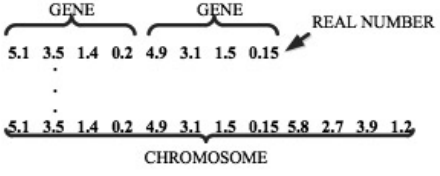


Figure 3. It is illustrated the encoding of the individuals. A gene corresponds to centroid coordinates and the number of genes depends on the number of clusters used. As well as, a chromosome is the set of genes of an individual.

- The selection of individuals is determined by considering the current individual that is being operated (P1) and the other is an individual selected by binary tournament among its neighbors (P2). In Figure 4 is exemplified a binary tournament selection.

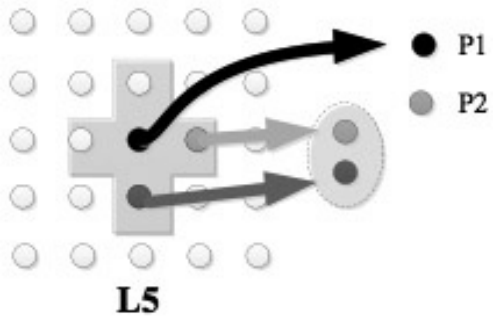


Figure 4. One of the two parents is selected by the current individual (P1) and the other is selected using binary tournament (P2).

- During recombination, both parents' centroids are considered. The recombination between P1 and P2 centroids is shown in Figure 5.

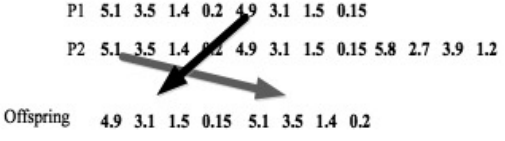


Figure 5. A new individual is generated by the recombination of parents P1 and P2. This individual will have at least 2 partitions.

- We consider a uniform mutation with random increments up to ±2δ. Where δ is a number generated between 0 and 1 with a uniform distribution. In the same way the + and - signs occur with equal probability.
- The replacement of individuals is performed if the offspring generated by P1 and P2 were better than P1.
- Evaluation of individuals is based on the Davies-Bouldin index. This index considers the error caused by the centroids and the distance among partitions [9]. As the index is an average, this gives us a general measure of the relation between cluster scatter for clusters and their separation among them.

$$Disp(Ci) = \sqrt{\frac{1}{Ci}\sum_{oi,oj \in Ci,\ oi \neq oj} \|oi - oj\|} \quad (1)$$

$$DB = \frac{1}{k}\sum \max_{i,j=1,..,k,\ i \neq j} \left\{ \frac{Disp(Ci) + Disp(Cj)}{d(Ci,Cj)} \right\} \quad (2)$$

Where:

Disp (Ci)	is the cluster scatter for a cluster,
i and j	are the i^{th} and j^{th} cluster,
k	is the number of clusters.

2.2 Simple and distributed cellular genetic algorithms Models

From the Whitley's review of parallel implementations in GA, two important models are relevant: the islands or distributed model and the cellular model [2].

The distributed or islands model divides the population into subpopulations. Thus, the independent evolution of each subpopulation is performed and an operator called migration is used to exchange information among subpopulations after a specific number of generations. In Figure 6 is exemplified the interaction among islands.

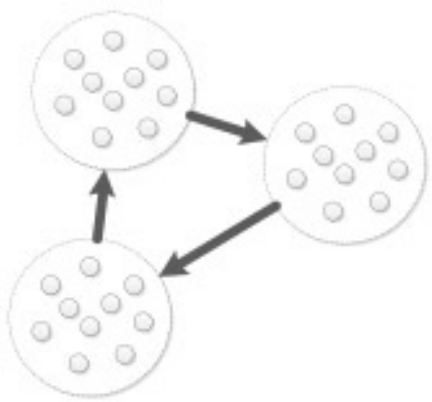


Figure 6. The distributed model has isolated subpopulations, which after a specific number of generations, exchange information using the migration operator.

On the other hand, the cellular model divides the population until just all the individuals evolve independently. This model does not need the migration operator because of exchanging information is achieved with the interaction of the neighborhoods.

The result of combining the cellular model with the islands model is described in works [10, 11]. This model divides the population into subpopulations, and each subpopulation is divided using a cellular structure. The main advantage of this model is that enhances search space diversity. In Figure 7 is shown the islands' subpopulations cellular structure.

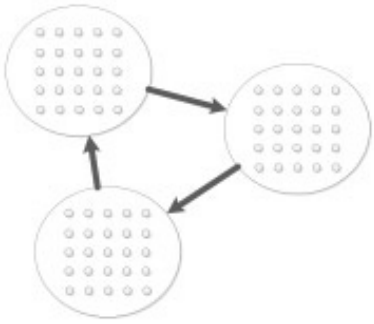


Figure 7. It is shown that the distributed CGA model has some characteristics of the cellular model, while it allows the exchange of information using the migration operator.

2.3 CGA implementation on GPU

Nowadays GPU cards offer a good option to deal with general problems that can be expressed as parallel data computations. In this case GPU is adapted to the necessity of running an individual on a thread, and making use of shared memory to avoid read/write data access bottlenecks [12].

Algorithm 1 contains the pseudocode for a simple CGA model. Figure 8 shows the GPU configuration to implement this model. The implementation contemplates the use of an individual per thread. The set of threads belongs to a single GPU block unit. In this case the use of a single block is limitated because the maximum supported population is 100 individuals. It is worth noting that shared and constant memories were used to provide efficient access to data.

ALGORITHM 1
Simple Cellular Genetic Algorithm Pseudocode

```
Initialize the population
Evaluate the population
While stop condition is not met ...
    For each individual in the population ...
        Get its neighbors
        Select its parents (P1, P2)
        Recombine its parents
        Mutate its offspring
    Replace the population
    Evaluate the population
```

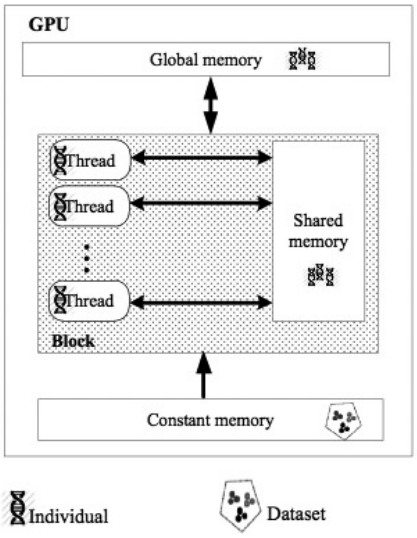


Figure 8. For a simple CGA model implementation, an individual is run in a thread of a GPU block unit. The shared memory contains the population that is going to be replaced whereas the constant memory contains the data to be partitioned and is going to be used for read only. The global memory is used to exchange data between CPU and GPU.

Algorithm 2 presents a pseudocode for a distributed CGA model, and unlike Algorithm 1, Algorithm 2 has an extra loop to deal with islands, and the migration operator to exchange information among islands. Figure 9 shows the GPU configuration to implement a distributed CGA model. The implementation contemplates the use of an individual per thread, while a GPU block unit corresponds to an island. As in the simple CGA model, shared and constants memories are used to provide efficient access to data.

ALGORITHM 2
Distributed Cellular Genetic Algorithm Pseudocode

```
Initialize the population
Evaluate the population
While stop condition is not met ...
    For each island …
        For each individual in the subpopulation ...
            Get its neighbors
            Select its parents (P1,P2)
            Recombine its parents
            Mutate its offspring
    Replace the population
    Evaluate the population
    Migrate individuals among islands
```

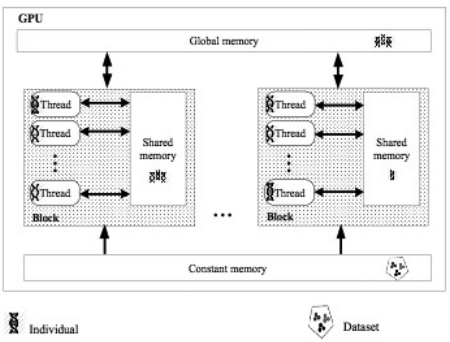


Figure 9.The distributed CGA implemented model as the simple CGA one, considers that an individual runs on a thread, but, in this case, there are as many blocks as islands. The shared memory contains the population that is going to be replaced and the constant memory contains the data that is going to be partitioned and it is used for read only. The global memory is used to exchange data between CPU and GPU.

The implementation of these algorithms was performed using CUDA (Compute Unified Device Architecture), release 5.5 in a NVIDIA GeForce 610M card with 2 GB memory.

3. Performance Measurement

Performance measures used for this experiment were: speedup, efficiency and serial fraction.

Speedup and efficiency are important because they indicate how fast the algorithm runs in a parallel implementation in comparison to the sequential one. The average time is considered for the GA [13].

Serial fraction is important because it allows identifying how efficient could result the addition of more processing units to run an algorithm.

3.1 Speedup

This measure is the most important, because of expreses the relationship between the CPU runtime with respect to the GPU runtime. Ideally it is desired that the speedup could be greater than the number of processors.

$$S_m = \frac{T_1}{T_m} \quad (3)$$

Where:

S_m is the speedup (soft) of the algorithm,

T_1 is the average runtime of the sequencial algorithm version,

T_m is the average runtime of the parallel algorithm version on GPU,

m is the amount of threads.

3.2 Efficiency

Efficiency is a normalization of the speedup. The efficiency values are between 0 and 1, where 1 is the best value. This measure enables the comparison among different implementations of an algorithm.

$$e_m = \frac{S_m}{m} \quad (4)$$

3.3 Serial fraction

It is a measure given by Karp and Flatt that allows identifying subtle effects in comparison with using only acceleration or efficiency measurements [14]. The importance of this measurement is based on the suposition that not all of the processors run an algorithm in the same amount of time. Small values are better.

$$f_m = \frac{1/S_m - 1/m}{1 - 1/m} \quad (5)$$

Donde:

f_m is the serial fraction,

S_m is the speedup (soft) of the algorithm,

m is the amount of threads.

4. Experimental Results

The implemented models were run 30 times each one, for a dataset that had been used in several studies, such as [4]. It was observed the size of

the population, the type of neighborhood and the number of islands affects the performance of CGA.

Iris dataset was used, it has four dimensional data [15], and this dataset can be seen in Figure 10.

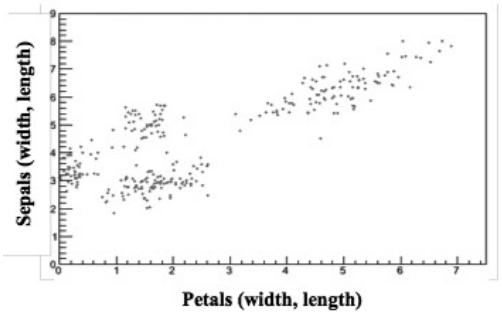


Figure 10.It is shown the Iris dataset in two-dimension.

In Figure 11 is illustrated the fitness convergence graph of individuals according to the number of generations.

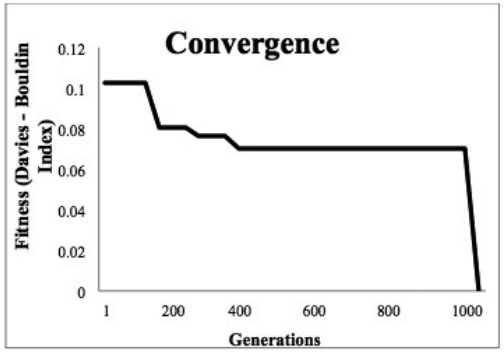


Figura 11. Converge graph is shown for the simple and distributed CGA models, these with 5 and 9 neighbors.
Table 1 shows the parameters used in the simple and distributed CGA models executions.

TABLE 1
Parameters Used In Simple Cga Execution

Cross probability: 0.8
Mutation probability: 0.1
Generations limit: 1000
Neighborhoods: linear with 5 (X-L5) and 9 (X-L9) neighbors

The execution performance of the simple (S) and distributed (D) CGA models are summarized in tables from 2 to 4, which contains the average runtime, speedup, efficiency and serial fraction, where linear 5 (L5) and 9 (L9) neighborhoods are used.

The best speedup and serial fraction values, for a population of 100 individuals and a L9 neighborhood for the simple CG model, are shown in Table 2.

TABLE 2
Performance Results Of The Simple Cga Model

Algorithm	Threads	T_1	T_m	S_m	e_m	f_m
S-L5	9	63.9	58.011	1.102	0.122	0.896
S-L9	9	105.1	38.562	2.725	0.303	0.288
S-L5	36	66.2	65.747	1.007	0.028	0.993
S-L9	36	236	49.247	4.792	0.133	0.186
S-L5	81	60.4	63.833	0.946	0.012	1.058
S-L9	81	522	80.489	6.485	0.080	0.144
S-L5	100	601.3	95.305	6.309	0.063	0.150
S-L9	100	1580.4	88.562	17.845	0.178	0.047

The speedup, efficiency and serial fraction performance graphs of the simple CGA model are shown in Figures from 12 to 14.

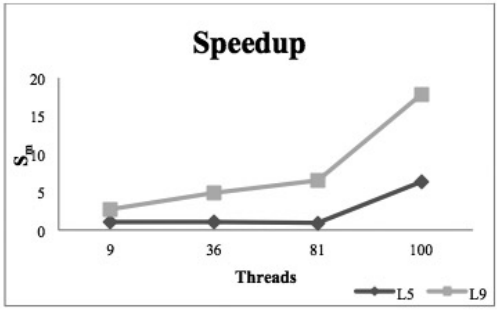


Figure 12.Speedup of the simple CGA model using 5 and 9 neighbors.

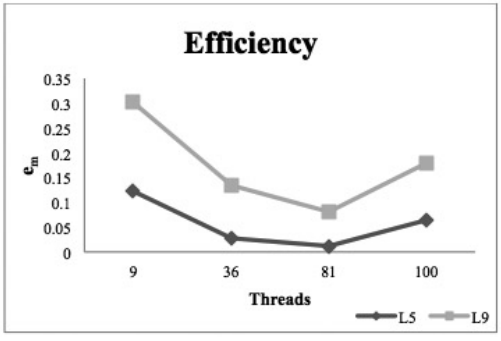


Figure 13.Efficiency of the simple CGA model using 5 and 9 neighbors.

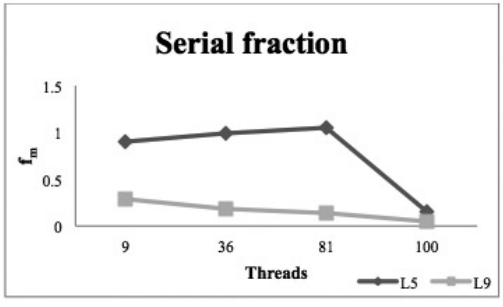


Figure 14.Serial fraction of the simple CGA model using 5 and 9 neighbors.

The best speedup and serial fraction values, for a population of 100 individuals and a L9 neighborhood for the distributed CGA model, are shown in Table 3.

TABLE 3
Performance Results Of The Distributed Cga Model With Resizable Islands

Algorithm	Threads	T_1 Seg	T_m Seg	S_m	e_m	f_m	Island size
D-L5	36	66.2	100.824	0.657	0.018	1.538	9
D-L9	36	236	115.016	2.052	0.057	0.473	9
D-L5	100	601.3	128.540	4.678	0.047	0.206	25
D-L9	100	1580.4	114.221	13.836	0.138	0.063	25
D-L5	400	435.2	513.455	1.848	0.005	0.540	100
D-L9	400	2209.9	451.464	31.877	0.080	0.029	100

The speedup, efficiency and serial fraction performance graphs of the distributed CGA model (with resizable islands), are shown in Figures from 15 to 17.

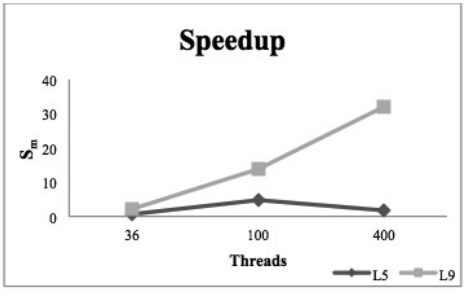


Figure 15.Speedup of the distributed CGA model with 4 resizable islands.

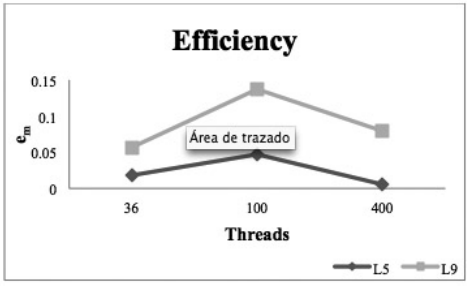


Figure 16.Efficiency of the distributed CGA model with 4 resizable islands.

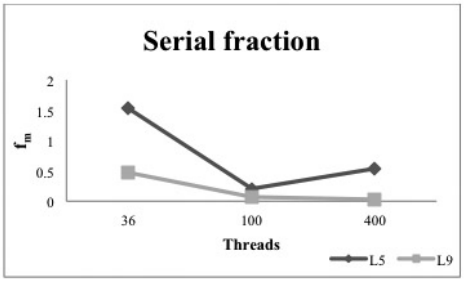


Figure 17. Serial fraction of the distributed CGA model with 4 resizable islands.

The best speedup and serial fraction values, for a population of 400 individuals and a L9 neighborhood for the distributed CGA model (with 25-size islands), are shown in Table 4.

TABLA 4
Performance Results Of The Distributed Cga Model With 25-Size Islands

Algorithm	Threads	T_1 Seg	T_m Seg	S_m	e_m	f_m	Islands
D-L5	100	601.3	128.540	4.678	0.047	0.206	4
D-L9	100	1580.4	114.221	13.836	0.138	0.063	4
D-L5	225	435.2	263.887	1.649	0.007	0.605	9
D-L9	225	2209.9	217.229	10.173	0.045	0.094	9
D-L5	400	949	423.304	2.242	0.006	0.445	16
D-L9	400	14391.5	334.465	43.028	0.108	0.021	16

The speedup, efficiency and serial fraction performance graphs of the distributed CGA model (with 25-size islands), are shown in Figures from 18 to 20.

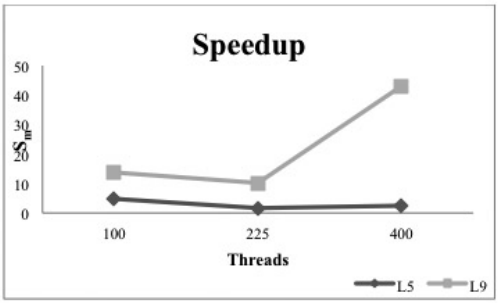


Figure 18.Speedup of the distributed CGA model with 25-size inslands.

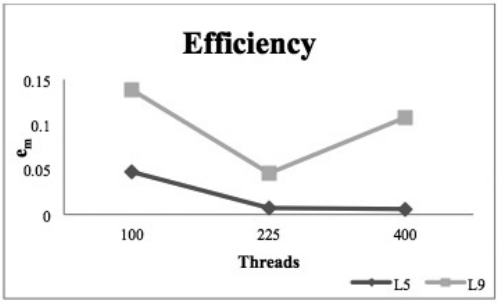


Figure 19.Efficiency of the distributed CGA model with 25-size inslands.

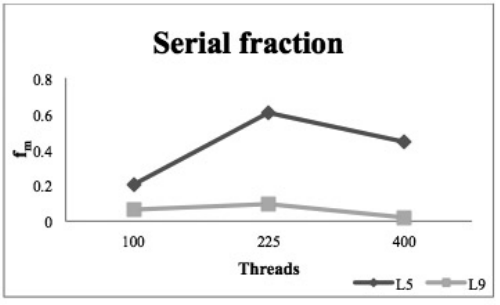


Figure 20.Serial fraction of the distributed CGA model with 25-size inslands.

5. Conclusions

An empirical study of the performance for the simple and distributed CGA models on GPU was presented.

The experimental population of 400 individuals used is greater than the 50 individuals reported in paper [4], enhancing the search space diversity.

The serial fraction allows reviewing the workload of each thread in comparison to its speedup and efficiency.

The speedup and efficiency have better values when L9 neighborhood is used, in both models.

Particularly, the efficiency is affected when the population size is increased in both models. However, it shows an improvement when L9 neighborhoods is used. On the other hand, the distributed CGA model has better efficiency with 100 individuals.

For the distributed model the serial fraction gets improved when the population size is 400 individuals. This may be due to the use of the shared memory in each one of the GPU blocks, allowing workload balance among threads.

Using the GPU mainly improves speedup and serial fraction because of shared memory is able to increase the number of blocks without sacrificing speedup.

References

[1] J. Holland, "Adaption in Natural and Artificial Systems", University of *Michigan Press*, 1975.

[2] D. Whitley, "A genetic algorithm tutorial", Computer Science Department, *Colorado State University, Fort Collins, CO 80523, USA, Statistics and Computing* 4, 65 – 85, 1994.

[3] P. Vidal y E. Alba, "A Multi-GPU Implementation of a Cellular Genetic Algorithm", *Evolutionary Computation (CEC), 2010 IEEE Congress on, Barcelona, 2010.*

[4] S. Bandyopadhyay y U. Maulik, "Nonparametric genetic clustering: Comparison of validity indices", *Systems, Man and Cybernetics, Part C, IEEE Transactions on: Applications and Reviews*, 2001.

[5] G. Xiufeng y X. Changzheng, "K-means Multiple Clustering Research Based on Pseudo Parallel Genetic Algorithm", *Information Technology and Applications (IFITA), 2010 International Forum on.*

[6] B. Dorronsoro y E. Alba, "Cellular genetic algorithms", *Operations Research/Computer Science Interfaces Series 42, Springer US*, 2008.

[7] E. Alba y J.M. Troya, "Cellular evolutionary algorithms: Evaluating the influence of ratio", *Parallel Problem Solving from Nature PPSN-6, volume 1917 of Lecture Notes in Computer Science, pp. 29-38, Springer, 2000.*

[8] N. Murilo C., A. C. P. L. F. de Carvalho, R. J. Gabrielli B., Campell, E. R. Hruschka. "Genetic Clustering for Data Mining",*Soft Computing for Knowledge Discovery and Data Mining*, Universidade de São Paulo, Brazil, 2008, pp. 113-132.

[9] D. L. Davies y D. W. Bouldin, "A Cluster Separation Measure". IEEE

Transactions on Pattern Analysis and Machine Intelligence, 1979, pp. 224–227.

[10] E. Alba, "Análisis y Diseño de Algoritmos Genéticos Paralelos Distribuidos", Tesis Doctoral, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, 1999.

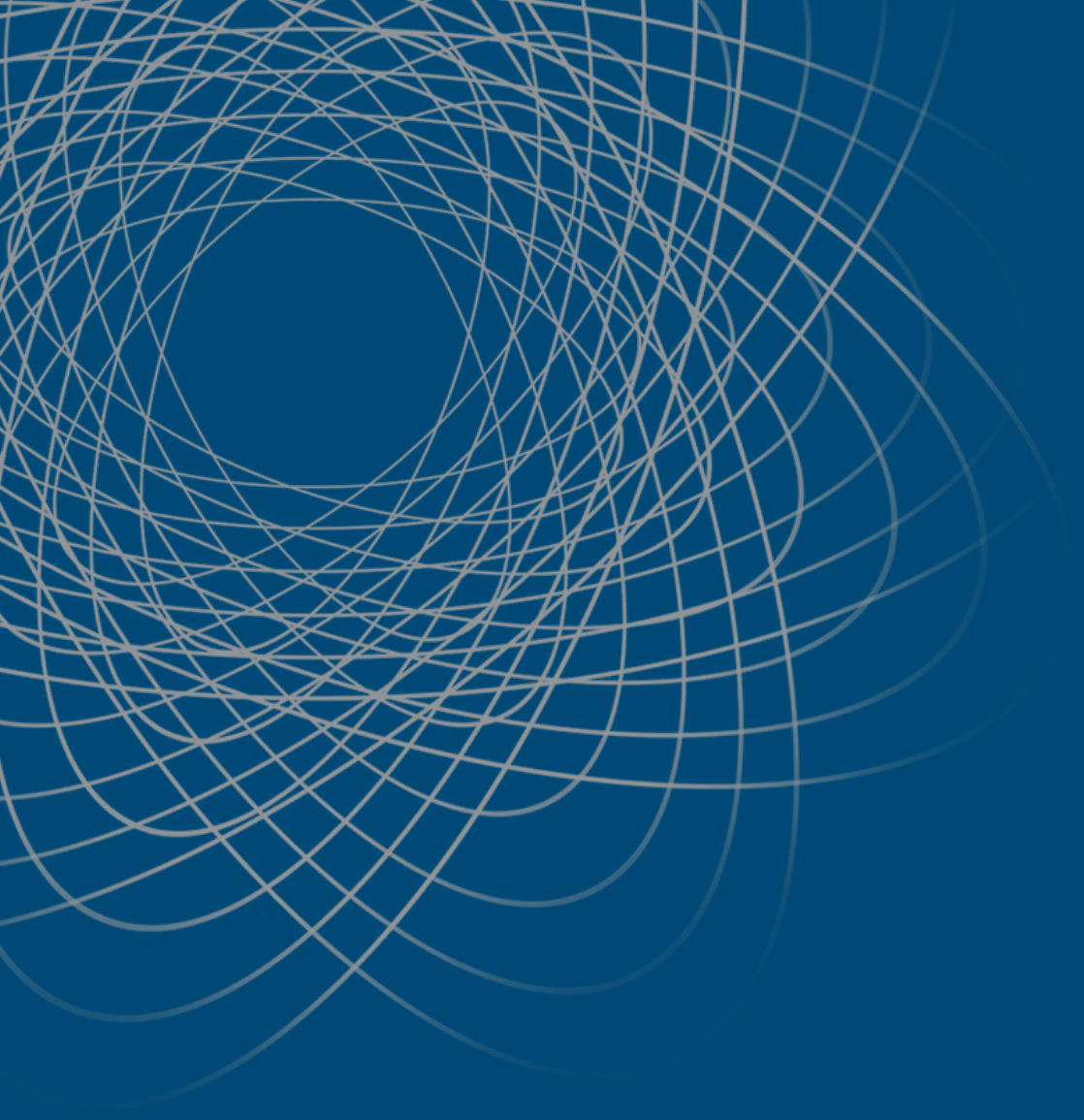
[11] T. Nakashima, T. Ariyama, T. Yoshida y H. Ishibuchi, "Performance evaluation of combined cellular genetic algorithms for function optimization problems", *Computational Intelligence in Robotics and Automation, Proceedings. 2003 IEEE International Symposium on, pp. 295-299.*

[12] D. B. Kirk y W. W. Hwu, "Programming Massively Parallel Processors", Elsevier Inc, segunda edición, 2013.

[13] G. Luque y E. Alba, "Parallel Genetic Algorithms", Springer-Verlag Berlin Heidelberg, 2011, pp. 41-45.

[14] A.H. Karp, H.P. Flatt,"Measuring Parallel Processor Performance", *Communications of the ACM 33*, 1990, pp. 539–543.

[15] UCI Machine Learning Repository, Iris Dataset, http://archive.ics.uci.edu/ml/datasets/Iris



INFRASTRUCTURE

A TCP/IP Replication with a Fault Tolerance Scheme for High Availability

Violeta Medina1, Andrei Tchernykh2, Antonio Ramos Paz1
[1]División de Estudios de Posgrado, Facultad de Ingeniería Eléctrica, Universidad Michoacana de San Nicolás de Hidalgo, Morelia, Mich.,México
[2]Computer Science Department,CICESE Research Center, Ensenada, BC, México
violeta mr@yahoo.com.mx, chernykh@cicese.mx, arpaz@umich.mx

Abstract

In this paper, we propose a TCP/IP Replication scheme for a fault tolerance system to provide high availability. This protocol has been implemented on virtual machines managed by Xen. Each package received by a virtual machine called original is replicated to a virtual machine called replica. The main purpose of this protocol is to preserve the same state of the original virtual machine in the replicated virtual machine for high availability. Original machine is substituted by a replica dynamically adapting to failures. It does not require modifications of TCP/IP protocol, host or guest operating systems. Quantitative results of protocol behavior in real scenario are analyzed with special attention to network performance.

Index Terms *: Replication, TCP/IP, virtual machines, virtualization.*

I. Introduction

Over the last years, high availability issues have been the subject of intensive research and experimentation. However, there are still many open problems in the field. One of the big challenges is the development of high availability mechanisms for distributed systems. They have to ensure a prearranged level of operational performance during a contractual measurement period. Distributions of operations create extra difficulties. The Transmission Control Protocol (TCP) complementing the Internet Protocol (IP) [1] is one of the core protocols of the shared public distributed infrastructure. It is used by major Internet applications, and provides reliable mechanism of data exchange between computers.

Virtualization technology was beginning to emerge in the 1960s. The origin of virtual machines was in mainframe computers with multiprogramming operating systems, which made use of time-sharing and resource-sharing on expensive hardware [2]. The principal architectural characteristics were the dual-state hardware organization with privileged and non-privileged modes [3]. Nowadays, interest in virtualization has resurged since computers have sufficient processing power to use virtualization as a technique of partitioning resources of a single server into multiple separated execution environments. As each environment is isolated, then, multiple operating systems can run on the same hardware [4], [5], [6]. The current Virtual Machine Monitors (VMMs) offer such characteristics as isolation, server consolidation, cloning, server migration

and replication [7]. These characteristics can be used to provide high availability in a virtualized system. Many databases, file systems, mail servers, and virtualized Cloud systems have included replication and migration mechanisms to reach high availability. In this paper, a replication protocol with quality of service for high availability is proposed and analyzed. This protocol replicates the state of one virtual machine to a second virtual machine via TCP/IP commands.

Detecting a failure, it switches one machine to another one providing high availability and quality of service. We continue this paper by presenting our protocol and discussing related work in Section II. In Section III, we describe architecture details. In Section IV, we discuss details of the protocol implementation. In Section V, experimental setup, test cases, and experimental results are presented. Finally, we conclude with a summary in Section VI.

II. Related Work

Several service replication mechanisms based on TCP/IP protocol have been published. In [8], a TCP fault tolerance scheme for cluster computing named ER-TCP is proposed. In the cluster, there is a primary server, unique logger server, and one or multiple backup servers. To synchronize nodes ER-TCP uses a multicast tunnel. The primary server waits for the logger response, and answers to clients. The logger server maintains the synchronization of TCP connections with the backup servers by gathering responses during the establishment and termination phases. Logger server is synchronized with the primary server at any moment, while the backup servers are synchronized at the beginning and at the end of every conversation. A fault on the primary or in a backup server can be attended. It is assumed that primary and logger servers are not failed simultaneously.

In [9], a transparent TCP Connection Failover is presented. The failover is achieved by modifying the server's TCP/IP stack. No modifications are required to the client's TCP/IP stack, the client application or the server application. IP datagrams that the client sends to the primary server must be redirected to the secondary server. The secondary server receives and acknowledges all TCP segments sent by the client, first that the primary server. The secondary server must synchronize its TCP sequence numbers with the sequence numbers used by the primary server. The scheme allows the system continues working if the primary or the secondary server fails.

In [10], a fault-tolerant TCP (FT-TCP) protocol is described. Two types of backups are considered: hot and cold. FT-TCP uses a component called south-side wrapper (SSW) to intercept, modify, and discard packets between TCP/IP stacks and network drivers. FT-TCP intercepts and changes semantics of system calls (between an application and kernel) made by server applications through a component called the north-side wrapper (NSW). Both components communicate with a stable buffer located in the physical memory of backup machines. The stable buffer acknowledges received data, and returns them to a requester in an orderly way. In the hot-backup scheme, after a primary server failure, the backup server reaches the state of the primary server before its failure. This process uses TCP segments and system call records from the stable buffer. Failover with a cold backup consumes memory for stored packets and system calls. The recovery process from a cold backup can

be handled by a checkpoint mechanism.

In [11], [12], [13], a fault-tolerant Web service scheme based on Connection Replication and Application-level Logging (CoRAL) is presented. The basic idea of CoRAL is to use a combination of active replication and logging. Two servers are considered: a primary and backup. Each server can play a dual-role as a primary for some requests and backup for others. At the connection level, the server-side TCP state is actively replicated on the primary and backup servers. However, the standby backup server logs HTTP requests, but does not process requests unless the primary server fails. At the application level, HTTP request and reply messages are logged. The key of this scheme is that the backup server obtains every TCP packet (data or acknowledgment) from the client before the primary server. Thus, the only way the primary obtains a packet from the client is when there is a copy in the server backup.

In [14], an infrastructure HYDRANET-FT for dynamic replication of services across an internetwork is presented. It modifies TCP communication protocol on the server side to allow one-to-many message delivery from a client to service replicas, and many-to-one message delivery from the replicas to the client. The main objective of HYDRANET-FT is to provide fault-tolerant services across an internetwork of different network providers. It supports atomic multicasting [15] to ensure that all servers process the same operations and keep the same state. Backups are connected in a daisy chain to the primary host. In this schema, all replicas receive TCP packets from the client, but only the primary responds to the client.

In [16], Migratory TCP (M-TCP) support for live connection migration is proposed. Hosts in a pool of similar servers can accept migrating connections and continue service on them. M-TCP transfers the last connection checkpoint state, along with protocol specific state, to the destination host to migrate the server endpoint. The protocol ensures that the destination server resumes the service while preserving the same state, without disrupting the traffic on the connection. M-TCP provides a mechanism called cooperative service model, in which an Internet service is represented by a set of geographically dispersed equivalent servers. M-TCP has some limitations such as lack of fault tolerance support.

In Table I, we summarize TCP replication protocols and their main characteristics. Active replication indicates that client requests are simultaneously processed by multiple replicas. Message logging states that all incoming client messages are stored in a file. Checkpointing indicates that up-to-date copies of the server state are maintained on a backup or stable storage. Failover shows an automatic switching to a redundant or standby computer server. Multi-backup indicates more than one simultaneously maintained backup server. Scope shows maintenance of local or geographically separated replica servers. Modifications of kernel, TCP/IP original source code and application source code are shown in next columns.

III. Architecture

The replication process is supported between VM (the Client machine) and one or more replica virtual machines. At the connection level, the TCP state of an original machine, called Original Virtual Machine (OVM) is actively replicated on Replica

Table I

TCP REPLICATION PROTOCOLS AND THEIR CHARACTERISTICS

Mechanism	Active replication	Message logging	Checkpointing	Failover	Muti-backup	Scope	Kernel	TCP/IP source	Application	References
ER-TCP	•	•	•	•	•	•[1]				[8]
Transparent TCP Connection	•			•	•[3]	•[1]		•		[9]
FT-TCP	•[4]	•[5]			•		•[1]		•	[10]
CoRAL	•			•	•	•[1]	•	•	•	[11], [12], [13]
HYDRANET-FT	•			•	•	•[2]	•	•	•	[14]

1 = LAN, 2 = WAN, 3 = Daisy chaining mechanism, 4 = Hot-backup, 5 = Cold-backup

Virtual Machines (RVMs). Clients communicate with a service through a unique principal server using a single server address called Principal Server Connection (PSC). At the TCP/IP level, all messages sent by the client have the address of the PSC as the destination and all messages received by clients have the address of the PSC as the source address. The protocol stores a list all available servers, including the single PSC IP address. This PSC address is different to the IP address of the OVM Principal Server. The protocol intercepts the packets with PSC destination and this address is substituted by the IP address of the OVM.

RVM can substitute actual OVM to cope with failures, different network workloads and communication delays. When OVM sends packets to the client, these packets are intercepted by the replication protocol. Only IP source address is substituted by PSC IP address. Replica servers process these packets and update their own TCP state. The client realizes a selection of the initial TCP sequence number. To achieve client transparency at the TCP level, all replica servers choose identical sequence numbers during connection establishment. The ACK numbers are different for the conversation with each replica server. The proposed protocol replicates packets transmitted to a destination host, and redirects them to local and remote VMs. The objective is to maintain equal states of all VM, including memory, local storage disk, and network connections. Interactions with OVM based on TCP/IP are replicated to the replica VM. The protocol is installed on an intermediate machine, which controls the packet replication. The complete implementation of this protocol has been made with open source code: Xen, Libnet [17] and Libpcap [18]. Sequence numbers and acknowledgments are maintained according to RFC793 [19].

The network packets associated with launched

commands in RM are captured and stored in a queue. The initial sequence and acknowledgment numbers are generated when the connection is established according to the three-way handshake.

Data from RM to OVM are sent in the same order as from RM to RVM. Every time when data are sent by the client or server, the countpart should send a signal that the information has been received, and acknowledge how many bytes have been received. Due to the fields in the TCP header are modified, checksum is recalculated according to RFC 793 specifications. Besides, the subfield Timestamp is modified based on RFC 1323. The timestamp clock frequency is set in the range from 1ms to 1 sec per tick. When a replication of a SYN package is detected, the field TSVal is initialized with the valued of the original package.

The field TSec is set to zero as initial value. When a last TSVal is sent, TSVal is calculated as a time when the last detected duplicated package is appeared plus hundredths seconds. The value of the field TSecr of a duplicated package corresponds to the value of the TSVal of the last package captured with direction from RVM to OVM. To avoid that unsolicited packages are reset, it is necessary to configure a firewall rule to drop packages from RVM. A package filtering is applied. Only packages that belong to the TCP protocol and conversations RM- OVM or RM-RVM are caught.

IV. Protocol Implementation

One of characteristics that distinguishes distributed systems from systems with only one machine is the concept of partial fault. In a distributed system, a partial fault can occur when a component fails. A fault can affect the operation of some components and, at the same, time others can not be affected.

On the other hand, in a not distributed system, a fault can be total, because it affects all components and leaves a system inactive. An important objective in the distributed system is the construction of mechanisms that allow the automatic recuperation from partial faults without affecting seriously the total performance. When a fault occurs, the distributed system should continue working in an acceptable mode while the repairs are realized. An important concept in fault tolerance is atomicity. For example, in distributed transactions it is necessary to guarantee the realization of all involved operations in a process [20].

A. Fault Tolerance Protocol

In this configuration, the client sends petitions to a unique IP address. This address is a virtual address that belongs to the machine where the replication protocol is installed. The client sends requirements to this virtual IP and the protocol redirections them to the OVM and at the same time, these requirements are sent to the RVMs. In a general way, the protocol follows next steps:

1. The client sends a requirement to the Principal Server Connection, PSC.
2. The packets with the client source IP address and with PSC destination IP address are caught for the protocol.
3. The protocol modifies the trapped packets. The source IP address is changed by the PSC source address. The destination IP address is changed by the OVM IP address.

4. The packets are replicated a number of times equal to the number of replicas in the system. These packets change the source IP by the PSC IP address and the destination IP by the IP address of the corresponding RVM.
5. The OVM receives packets with PSC source IP address; it processes and responds to them.
6. When the protocol detects a response package with OVM source IP address and PSC destination, the package is caught and modified to be redirected to the client machine. The source IP address is changed by the PSC IP address and the destination address is changed by the client IP address (RM).
7. The client receives responses like directly interacting with the PSC.
8. The responses from replicas are processed in the intermediate machine which has the protocol installed.

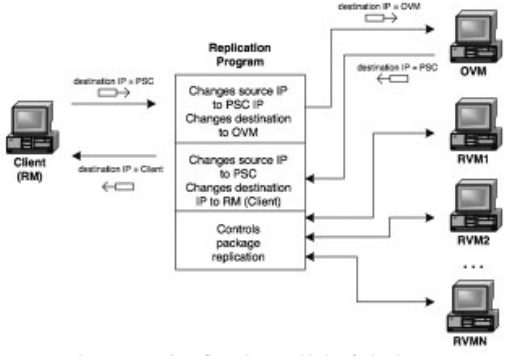


Fig. 1. Protocol configuration considering fault tolerance

The protocol configuration is shown in the Fig. 1. The protocol stores a queue of the available servers and their IP addresses including the unique IP address of the PSC server. It exists a queue of RVMs that allows to know where replicate commands. The redirection package is possible, because the client machine enables a virtual IP through ARP commands for the machine where the protocol is installed; this virtual address corresponds to the PSC IP address. In each RVM, the MAC address of the protocol machine is associated with the PSC IP address.

This configuration allows the client to communicate with a fixed address without knowing the OVM IP address. So, the OVM could be replaced transparently by a RVM if a fault is produced.

The protocol considers two fault types: OVM fault and RVM fault. It is considered that OVM and RVM do not fail simultaneously.

B. OVM Fault

When an OVM fails, the first RVM in the available replicas queue takes the place of the OVM. This queue stores the IP address associated with the physical machine and its function, for example:

-- 0, identifies the PSC IP address
-- 1, client IP address
-- 2, an OVM
-- 3, a RVM
-- 4, a machine that has taken the place of an OVM and it is on recuperation state.

If an OVM fails, next steps are followed: The OVM is eliminated from the available machines queue. The first RVM is set in the state 4 indicating that takes the OVM place. Before that RVM can be identified as the new OVM, the before the fault. So, the number of sent and received bytes in the replica conversation should be the same that

number of sent and received bytes in the original conversation before the fault. If at the moment of the fault, there are packages in transit, when the RVM is at the same state of the OVM before the fault, the RVM enters in a recuperation state until the current remote command is completed. During the recuperation, if the protocol identifies a packet with a source IP address that belongs to the replica virtual machine that will take the place of the OVM and PSC destination IP address, it is caught and modified:

1. The source IP address changes from the RVM on recuperation address to the PSC address.
2. The destination IP address changes from PSC IP address to the client address
3. Fields in the TCP header like the sequence and ACK numbers should be replaced by the corresponding numeration in the original conversation.
4. Timestamp Subfields in TCP options should be changed to keep the conversation between the client and the PSC.

Besides, the sliding window registers of the original conversation should be modified to preserve the sequence and ACK numerations in the correct state.

During the recuperation, the conversation between the PSC and the RVM that takes the place of the OVM should be maintained active. Therefore, this conversation is handled like an active normal replica. When the recuperation process finishes, the machine in recuperation state changes to OVM state.

C. RVM Fault

When a RVM fails, the recuperation process follows the next steps:

1. The replica in fault is searched in the replicas queue.
2. The memory that belongs to the replica is eliminated; such the Received Bytes Queue and all the internal variables that maintain the replica state are released.
3. The number of active replicas decremented by one.

V. Experimental Setup

To probe the fault tolerance configuration, there is a Latitude D410 computer client, Pentium M processor 2.0 GHz, 2 GB RAM, 80 GB DD. An intermediate computer, Xeon quad-core 2.0 GHz, con 3GB RAM, 72 GB DD, where the replication protocol is installed. OVM is installed on a Xeon server and the computer that assumes the client role has 3 RVMs installed. The operating system base is Debian 4.0 and the hypervisor Xen 3.1.3. The OVM and replicas have Ubuntu 8.04 operating system, con 512 de RAM y 2GB of storage. To realize the performance tests and fault tolerance tests a shell script, which executes random transactions like insertions, deleting or modification of registers is used.

The client machine has the IP address address `192.168.2.2.` There is a PSC unique address `192.168.2.1` and through the command arp -s the client associates this virtual address with the MAC address `(00:21:5A:44:DA:BA),` which is

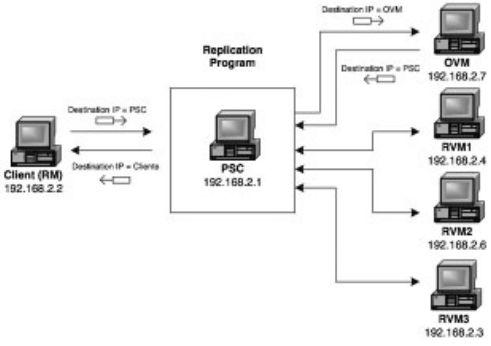


Fig. 2. Configuration for the case study

the machine with the active protocol. The arp command is executed in each active RVM. All replicas are in the same network segment. The Fig. 2 shows this configuration. The script was run to execute 500 random transactions each 8 seconds on the database installed on the OVMand replicas.

A. Results

To continue the average RTTs for the realized transactions between the client and the OVM with no replicas, one, two and three active replicas are shown. The Table II shows the obtained results.

The results in the Table II are measured in client-server sense. In column one, it can be observed that the RTT was measured from the RM (client) to some virtual machine (to OVM or replica). Column 2 indicates the number of active replicas during the measurement. Column 3 indicates the average RTT in milliseconds and column 4, the percentage of RTT degradation.

In Table II, it is observed that the replication mechanism introduces an overload on the system and the average RTT suffers degradation in comparison with the average RTT of a TCP system without replication of remote commands. It can be observed that the conversation between the client and the OVM suffers a degradation of 81.16% with one replica in comparison with the conversation without replicas. The RTT is degraded a 80.64% with two active replicas and a 82.92%

Table II

AVERAGE RTT, CONFIGURATION WITH PSC, CLIENT-SERVER

Direction	Replicas	Av. RTT (ms)	% RTT Degradation
RM-OVM	0	0.026990	--
	1	0.048890	81.16 *vs* 0 replicas
	2	0.048750	80.64 *vs* 0 replicas
	3	0.050710	87.92 *vs* 0 replicas
PSC-RVM1	1	6.238400	--
	2	6.418100	2.88 *vs* 1 replica
	3	7.519844	20.54 *vs* 1 replica
PSC-RVM2	2	9.778470	65.63 *vs* PSC-RVM1 2 replicas
	3	13.68177	39.92 *vs* 2 replicas
PSC-RVM3	3	13.56890	0.82 +efficient *vs* PSC-RMV2 3 rep.

with three active replicas. With PSC-RVM1 replica, the average RTT with two replicas suffers a 2.88% with two active replica in comparison with the RTT with one replica and 20.54% if there are three active replicas.
In Table II, it can be observed the PSC-VM2 conversation with two or more active replicas, is a 39.91% slower with three active replicas than the average RTT with two active replicas. It can be noted that the average RTTs of the PSC-RVM2 and PSC-RVM3 conversations are around 13.5 ms in both cases.

The average RTT suffers degradation when the system incorporates the replication protocol operation. The intermediate processing of the original package introduces a delay too. The original package is processed on the intermediate machine PSC, the source and destination IP addresses are modified and the packet is sent to the OVM. The packet processed in the OVM is trapped by the PSC again and processed. This packet changes the source and destination IP and returns to the RM (client). This processing introduces a RTT increment.

When there is an active replica, the RTT measurement includes an additional processing time for the packet that travels to RVM1, the package is removed from the waiting queue, necessary fields are modified and it is sent to RVM1, where and ACK package is waited. When the number of replicas is increased, the time to do replicas, send to the corresponding replica and wait for the ACK is incremented too.

OVM was located on a server with better performance than the server for RVMs. However, the RTT from the point of view of the client (RM) suffers up to 87.92% of degradation with three active replicas, the average RTT is 0.050710 ms which is imperceptible to the user. In related work, there are not comparable experiments with those in this paper, but in [13] a duplex replication is mentioned; a replication of a primary web server and its corresponding secondary server is realized. The client launches requirements of approximately 50 bytes.

The responses are in the range of 50 bytes. An average latency time of 6.40 ms is reported. This time is similar to the average RTT in the experiments when the replication protocol duplicates transactions from OVM in RVM1, which is 6.2384 ms, even when there are two active replicas, the average RTT is 6.4181 ms. In [8], latency increments from 15% up to 90% when the replication of a secondary server is introduced. The proposed protocol is within these parameters. However, in the before mentioned publication, it is not clear how the latency was measured. In Table III, the average RTTs from server to client without replicas and with one, two and three active replicas are shown.

Table III shows the average RTTs from server to client, the proposed protocol introduces an overhead and produces an increment to the average

RTT. In the case of OVM-Client, the average RTT with two replicas is 3.89 times greater than the average RTT without replicas; the average RTT with two replicas is 4.68 times greater and with three replicas is 6.29 times greater. In the conversation RVM1-PSC, the average RTT with two replicas increases 1.15 in comparison with the average RTT with one replica and 1.57 times with two active replicas. The average RTT in the conversation RVM2-PSC

Table III

AVERAGE RTT, WITH A PSC, SERVER-CLIENT

Direction	Replicas	Av. RTT (ms)	% RTT Degradation
OVM-RM	0	6.908490	--
	1	26.86444	388.86 *vs* 0 replicas
	2	32.29798	467.51 *vs* 0 replicas
	3	43.43767	628.76 *vs* 0 replicas
RVM1-PSC	1	43.37405	--
	2	49.71785	114.63 *vs* 1 replica
	3	68.00150	156.78 *vs* 1 replica
RVM2-PSC	2	43.50354	0.12 + efficient *vs* RVM1-RM replicas
	3	57.12202	131.30 *vs* 2 replicas
RVM3-PSC	3	58.53689	2.48 *vs* RMV2-RM 3 rep.

increases a 31% if there are three active replicas in comparison with time when there are two replicas in operation. Times for RVM2-PSC and RVM3-PSC show RTTs around 58 ms.

RTT from server to client suffers greater degradation than the results for the conversation client to server. This is because the acknowledgment of packages could be delayed by intermediate replication processes. In this configuration, the replicas are installed in a machine with less processing power, which affects the measurement of average RTT.

The fault tolerance scheme allows the automatic recuperation of the system during a fault and the reubication of the OVM in transparent way

B. OVM Fault

In this experiment, freeze fault of OVM was simulated. An interval of time was measured, it starts when it is detected that OVM let to respond requirements from PSC and stops when the first machine in the list enters in a recuperation state and PSC begins to redirect packages from the machine that will replace OVM to the client. In this moment, the intermediate machine, which operates the protocol, recognizes that RVM1

Table IV

TIMES TO REDIRECT TRAFFIC FROM OVM-TO-NEW OVM

Replicas	Times to redirect OVM-new OVM (ms)
1	5.05
2	5.07
3	5.64

will attend the requirements from client. The simulation was made with one, two and three active replicas. The number of samples was 30 in each case. The results are shown in Table IV:

The time between the protocol recognizes the fault and the RVM1 reaches the state of OVM had before let to responding is around 5 ms, which is a imperceptible time. These recuperation times allow an uninterrupted communication between client and the new OVM. Replicas in operation are not affected by the OVM change.

C. 5.3 RVM Fault

If the protocol detects that a replica does not respond requirement from PSC, the replica is eliminated from the available replicas queue and the memory of the connection state is released. The worst time registered releasing memory, when there are three active replicas and one of them fails was 0.003526 ms.

VI. Conclusions

The proposed protocol allows the automatic recuperation from partial fault in the system. It allows the recuperation to OVM or RVM faults, assuring the availability of the system and data integrity. The availability of the system implies a performance penalty.

According to the results, the conversation between the client and the PSC suffers a degradation of 81.16%, when the system operates with an active replica. This increment in RTT is because the RTT considers the time of sending the packet between the client and the intermediate machine (PSC). This packet is resent to the OVM; the OVM returns the answer to PSC and finally, this answer is resent to the client. High availability carries out a degradation in the system performance.

However, in the proposed protocol the degradation is imperceptible from the point of view of the client, which is up to 0.050710 ms, in the worst case, with three active replicas. In related work, it is mentioned replication cases with maximum of active replicas, where the latency average time is 6.40 ms. [13], which is a time greater than the result for the proposed protocol.

Our protocol reports an average RTT of 6.2384 ms with one active replica and 6.4181 ms with two active replicas. In [8], a degradation of 15%-80% with one active replica was reported. The proposed protocol has better limits of penalization for the current replication schemes on TCP/IP. The proposed protocol is based on active replication, which set up all structures on memory.

This could cause system degradation if the number of replicas increases. As a future work we will analyze the network performance with a greater number of replicas and the design of the mathematical model of this replication protocol.

Acknowledgment

The authors would like to thank Universidad Michoacana de San Nicolás de Hidalgo and Consejo Nacional de Ciencia y Tecnología (CONACYT) by the support for the development of this project.

References

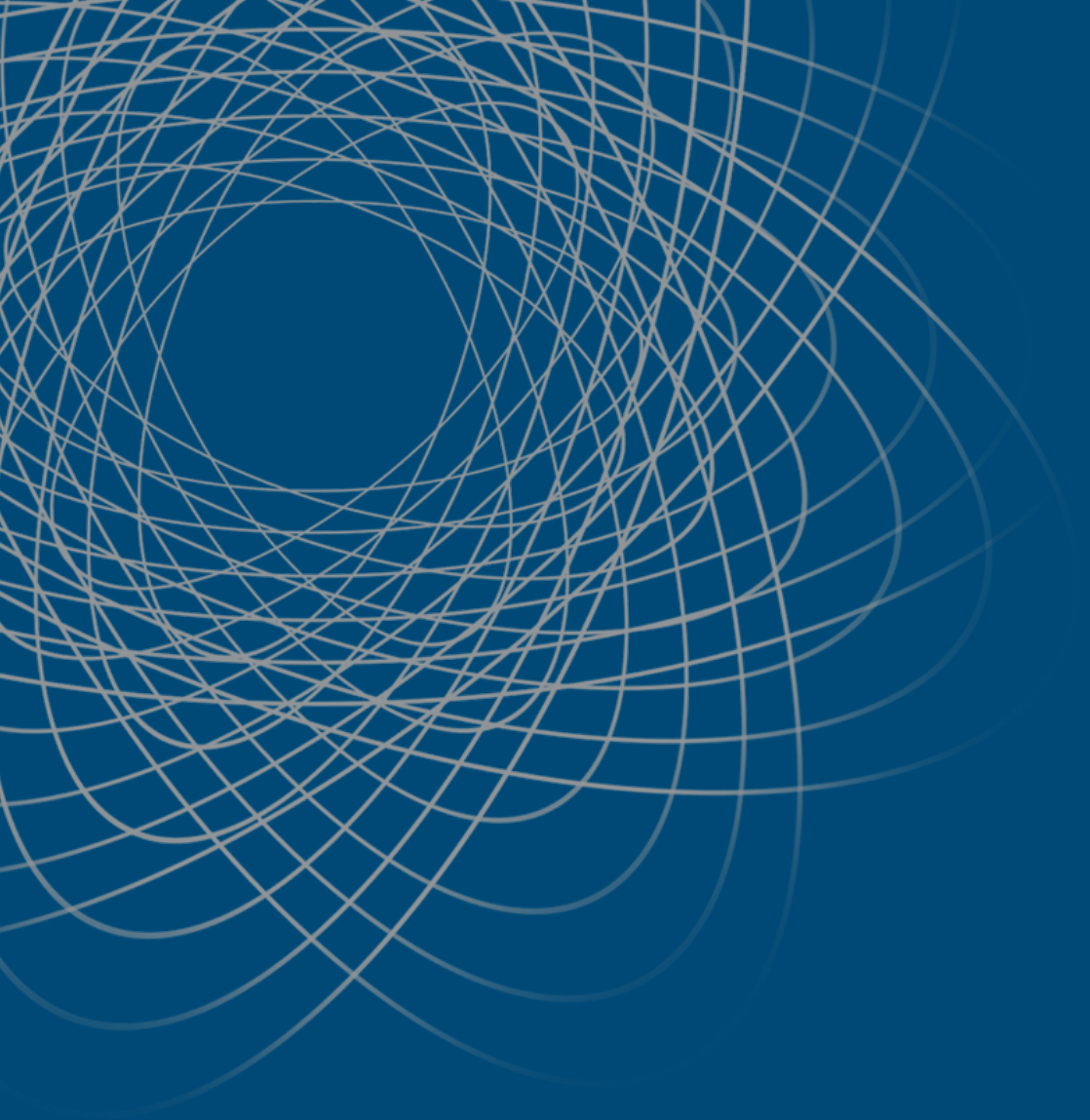
1. B. A. Forouzan, TCP/IP Protocol Suite, 2nd ed. New York, NY, USA: McGraw-Hill, Inc., 2003.
2. N. Susanta and C. Tzi-Cker, “A survey on virtualization technologies,” ECSL-TR-179, Stony Brook University, Tech. Rep., 2005. [Online]. Available: http://www.ecsl.cs.sunysb.edu/tr/TR179.pdf
3. R. P. Goldberg, “Survey of Virtual Machine Research,” IEEE Computer Magazine, pp. 34–45, Jun. 1974.
4. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neuge- bauer, I. Pratt, and A. Warfield, “Xen and the art of virtualization,” in SOSP ’03: Proceedings of the nineteenth ACM symposium on Operating systems principles. New York, NY, USA: ACM, 2003, pp. 164–177.
5. P. Changarti, Xen Virtualization: A Practical Handbook, 1st ed. Birm- ingham, B27 6PA, UK.: Packt Publishing Ltd., 2007
6. M. Rosenblum, “The reincarnation of virtual machines,” Queue, vol. 2, no. 5, pp. 34–40, 2004.
7. P. Padala, X. Zhu, Z. Wang, S. Singhal, and K. Shin, “Performance evaluation of virtualization technologies for server consolidation,” Re- port HPL-2007-59, HP Labs Tech, Tech. Rep., 2007.
8. Z. Shao, H. Jin, B. Cheng, and W. Jiang, “Er-tcp: an efficient tcp fault- tolerance scheme for cluster

computing," The Journal of Supercomput- ing, pp. 127–145, 2008.
9. R. R. Koch, S. Hortikar, L. E. Moser, and P. M. Melliar-Smith, "Trans- parent tcp connection failover," Dependable Systems and Networks, International Conference on, vol. 0, p. 383, 2003.
10. D. Zagorodnov, K. Marzullo, L. Alvisi, and T. C. Bressoud, "Practical and low-overhead masking of failures of tcp-based servers," ACM Trans. Comput. Syst., vol. 27, pp. 4:1–4:39, May 2009. [Online]. Available: http://doi.acm.org/10.1145/1534909.1534911
11. N. Aghdaie and Y. Tamir, "Client-transparent fault-tolerant web service," in 20th IEEE International Performance, Computing, and Communications Conference, 2001, pp. 209–216.
12. ——, "Fast Transparent Failover for Reliable Web Service," in International Conference on Parallel and Distributed Computing and Systems, Nov. 2003, pp. 757–762.
13. ——, "Coral: A transparent fault-tolerant web service," J. Syst. Softw., vol. 82, no. 1, pp. 131–143, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2008.06.036
14. G. Shenoy, S. K. Satapati, and R. Bettati, "Hydranet-ft: Network support for dependable services," in Proceedings of the The 20th International Conference on Distributed Computing Systems (ICDCS 2000), ser. ICDCS '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 699–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850927.851754
15. S. Mishra and L. Wu, "An evaluation of flow control in group communication," IEEE/ACM Trans. Netw., vol. 6, no. 5, pp. 571–587, Oct. 1998. [Online]. Available: http://dx.doi.org/10.1109/90.731193
16. F. Sultan, K. Srinivasan, D. Iyer, and L. Iftode, "Migratory tcp: Con- nection migration for service continuity in the internet," in ICDCS'02, 2002, pp. 469–470.
17. Libnet, "Libnet home page," http://libnet.sourceforge.net, 1999.
18. Libpcap,"Tcpdumplibpcappublicrepository,"http://www.tcpdump.org/, 2010.
19. I. S. Institute, "Rfc 793-transmission control protocol," sep 1981, edited by Jon Postel. Available at http://www.rfc-es.org/rfc/rfc0793-es.txt. [Online]. Available: http://www.rfc-es.org/rfc/rfc0793-es.txt
20. B. Forouzan, TCP/IP Protocol Suite, ser. McGraw-Hill Forouzan Networking. McGraw-Hill Education, 2010. [Online]. Available: http://books.google.com.mx/books?id=yNthPwAACAAJ

Violeta Medina received the Master degree from Universidad Michoacana de San Nicolás de Hidalgo (UMSNH) in 2008 and a PhD degree in Science in Electrical Engineering option Computation from UMSNH in 2013.

Andrei Tchernykh is a researcher in the Computer Science Department, CICESE Research Center (Centro de Investigación Científica y de Educación Superior de Ensenada), Ensenada, Baja California, México and teaches in CICESE's postgraduate programs in computer science, a position I have held since 1995. I graduated from the Sevastopol Technical University in 1975. From 1975 to 1990 I worked at the Institute of Precise Mechanics and computer Technology of the Russian Academy of Sciences, and took part in the parallel supercomputer ELBRUS Project development and implementation. I received my Ph.D. degree in Computer Science from the Institute of Precise Mechanics and Computer Technology of the Russian Academy of Sciences in 1986. In CICESE I lead the Parallel Computing Laboratory.

Antonio Ramos Paz received his B.Sc. degree in Electrical Engineering and the M. Sc. and Ph. D. degrees in 2001, 2002 and 2007, respectively all from the Universidad Michoacana de San Nicolás de Hidalgo. Currently, he is an Associate Professor in the División de Estudios de Posgrado, FIE, UMSNH, Morelia, México. His research interests are power systems analysis and modeling, as well as parallel processing.



PARALLEL COMPUTING

Parallelization of filter BSS/WSS on GPGPU for classifying cancer subtypes with SVM

Iliana Castro-Liera, J. Enrique Luna-Taylor,
Jacob E. Merecías-Pérez, Germán Meléndrez-Carballo
Departamento de Sistemas y Computación, Instituto Tecnológico de La Paz,
Boulevard Forjadores No. 4720, La Paz, B.C.S., México;
icastro@itlp.edu.mx, eluna@itlp.edu.mx

Abstract

A parallel version of BSS/WSS (Between Sum Square-BSS, Within Sum Square-WSS) filter using GPGPU (General Purpose Graphics Processing Units) is proposed. The application processes genetic expression data to select statistically relevant genes. A SVM (Support Vector Machine) is applied on the expression data of genes selected by the filter, to diagnose cancer. With the use of this combination of algorithms, a success rate of 92% in the diagnosis was achieved. With the parallel implementation on GPGPU with CUDA technology, a dramatic reduction of execution time of approximately 18 times compared to the sequential implementation on CPU was achieved.

Keywords: *genetic expression, Support Vector Machine, parallel programming, GPGPU, CUDA.*

I. Introduction

Cancer is a serious and complex problem that has caught the attention of the scientific community and is among the 10 main causes of death in the world [1]. Among the many fronts and efforts that are being made to address this problem, exists the development of algorithms that aim predict and classify different types of cancers using gene expression data. One of the main technologies that have provided large volumes of biological data are the DNA microarrays [2]. The datasets, on which the classification algorithms work, contain the expression levels of thousands of genes from a group of patients. For instance, the dataset for the analysis of Leukemia cancer [3], consists of expression values of 7129 genes for 72 patients.

This dataset is divided in two groups according to the specific subtype of Leukemia cancer: AML (Acute Myeloid Leukemia) and ALL (Acute Lymphoblastic Leukemia). Within the group of patients with AML subtype, there is another division corresponding to subgroups of patients that were successful in the treatment and those that were not successful in the treatment. With these data, a classification algorithm has as goals either diagnoses the specific subtype of Leukemia, or predict the likelihood of success of treatment.

Previous works for the classification and prognosis of cancer have been done. Golub et al. [3], use the *"neighborhood analysis"* method for the distinction of classes, grouping the genes with similar expression patterns. Next, using the levels of expression, and the degree of correlation of genes, a method is applied for each gene to assign

a score associated with each of the classes.

With the sum of the scores, the winning class is determined. The results of this method are validated through a *"cross-fold."* For the classification experiments they used 50 *"informative genes"* from Leukemia cancer dataset, obtained these previously, achieving a total of 36 correct predictions from 38 patients. Nasimeh and Russel [4], use an algorithm to identify biclusters, based on the mathematical method called *"rank-1 matrix approximation"*, which is applied to reduce the size of the dataset.

Each bicluster is a subset of genes and a subset of patients. The genes in a bicluster have expression values which are correlated. Based on the discovered biclusters, a Support Vector Machine is used to perform the classification. The Leukemia, Colon [5], Prostate [6], and Lung cancer datasets [7] were used to the diagnosis experiments, achieving a success rate of 84.72% for the Leukemia cancer dataset. The runtime of the method was approximately of 1 minute for obtaining each bicluster, from a data matrix of around 20,000 genes of 100 patients. Hernández et al. [8], use three filters to reduce the number of genes used in the classification.

The filters are: BSS/WSS, T-Statistic and Wilcoxon. After applying the filters, the subset of genes obtained is passed to a genetic algorithm (GA), which with the help of an SVM, performs the final selection of genes to be used in the classification. The Leukemia and Colon cancer datasets were used for the experiments. With the Leukemia dataset, a success rate of 98.61% was obtained. It is unclear if in the final experiment performed, were used patients' data that were also used in the filtering phase. None of the methods mentioned above, have a parallel computing technique. Our proposed method uses two algorithms to deal with the problem of classification and prognosis of cancer.

The first is the BSS/WSS filter [9], which was used to select the most relevant genes for classification. The second algorithm is a Support Vector Machine [10], to perform the classification based on selected genes by the filter. Due SVM is unable to perform a correct classification when the number of genes is greater than the number of patients [11], it is necessary to reduce the number of genes before applying the SVM for classification or prognosis of cancer. In the experiments performed, a success rate of 92% on the Leukemia cancer dataset was obtained.

Due to high cost of time that this kind of analysis involves on large volumes of data, a parallel BSS/WSS filter implementation was done. Using CUDA technology on a GNU/Linux Fedora environment with CUDA C, a reduction of the execution time of approximately 18 times was achieved, with respect to a sequential implementation on a CPU with the C programming language. Both implementations of the filter were performed 1000 times using different sets of patients in each iteration.

II. Development

A. BSS/WSS Filter

It has been shown that the use of BSS/WSS filter, is a useful method in the classification and prediction of cancer [9]. The function of this filter is to identify, statistically, those genes that behave differently between groups of patients of different classes.

In the filter, the selection of genes is based on the ratio of the sums of squares differences between groups (BSS), and within groups (WSS), calculated for each gene *j*. The filter's formula is shown in (1).

$$\frac{BSS(j)}{WSS(j)} = \frac{\Sigma_i \Sigma_k I(y_i=k)(\bar{x}_{kj}-\bar{x}_j)^2}{\Sigma_i \Sigma_k I(y_i=k)(x_{ij}-\bar{x}_{kj})^2} \quad (1)$$

Where y_i represents the class of the subtype of cancer for the *ith* patient. $\bar{x}_j$ represents the mean expression level of gene *j* for all patients. $\bar{x}_{kj}$ represents the mean of expression level of gen *j* for all patients that belong to the class *k*. x_{ij} represents the expression level of gene *j* for the patient *i*. The function $I(y_i = k)$ return 1 if the class of patient *i* is equal to the class *k*, and 0 otherwise.

Since our problem is to identify only two classes, and we know in advance the range of indexes of patients of both classes, we can omit the function $I(y_i = k)$ from (1), as shown on (2).

$$\frac{BSS(j)}{WSS(j)} = \frac{\Sigma_i \Sigma_k (\bar{x}_{kj}-\bar{x}_j)^2}{\Sigma_i \Sigma_k (x_{ij}-\bar{x}_{kj})^2} \quad (2$$

B. Filter implementation in CUDA

Since the formula of BSS/WSS filter is applied independently for each gene, the parallel implementation was designed so that each gene is processed by a thread. Due that the CUDA device must process 7129 genes, the algorithm was designed to run on 14 thread´s blocks of 512 threads each one, giving a total of 7168 threads, where the last 30 threads are not used.

Each thread has its own ID, whose value is calculated as shown in (3).

Where *BLK_DIM* represents the block´s dimension, *BLK_ID* is the ID of the thread´s block and *THR_ID* is the ID of the set of threads.

The Leukemia cancer dataset was loaded into memory in a one-dimensional array, including the class labels for each patient. Figure 1 shows the representation of the data in a one-dimensional array.

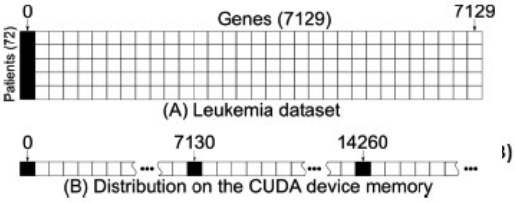


Figure 1.

Representation of the Leukemia dataset. The black cells contain the labels of the patients, and white cells contain the expression levels of the genes. A) The two-dimensional Leukemia array representation. B) The distribution of the Leukemia dataset on the CUDA device memory, where each segment of 7130 cells correspond to the label and genes of a patient.

In the same way, the indexes of the training patients, used in each iteration, were stored in a one-dimensional array. The size of this array is equal to the number of patients multiplied by the number of iterations to be executed. The formula used to get the index of a patient, for a given iteration, is shown in (4).

$$rowlabel = (k * patients\_length) + i \quad (4)$$

Where *k* is the number of iteration, *i* is the number of patient, and *patients_length* is the number of patients used for each iteration of the filter. The formula for get the index of the expression level of a patient is shown in (5).

$$rowdata = ((rowlabel - 1) * COLS) + tid \quad (5)$$

Where *COLS* is the total of columns that has the dataset, that is, the number of genes, plus one, for the label. A one-dimensional array to store the results was created. The size of the array is equal to the number of genes plus one, multiplied by the number of iterations to be executed. The formula for get the index to store a result is given in (6).

$$(num\_iter * COLS) + tid \quad (6)$$

Where *num_iter* is the number of the iteration. The results are real numbers that vary in the range from 0 to 3 for Leukemia cancer dataset. A result of 0, or close to 0, is considered a poor value, and is associated to a gene irrelevant for classification. A value close to 3 is considered as good, and its gene associated is relevant for the classification.

The Algorithm 1, shows the pseudocode of the CUDA kernel.

Each thread *j* starts getting the mean of the expression levels of its associated gene for all patients of each class. The mean of expression levels for all patients is obtained too. After, the square of the mean of each class minus the mean of gene *j* is obtained, and is multiplied by the number of patients in the corresponding class. These squares are summed and divided by the following summation. A summation of the squared, of the expression level of the gene *j* for the patient *i* minus the mean of the class of such patient, is performed.

Algorithm 1: BSS/WSS CUDA implementation

Input: one-dimensional array of gene expression, AML patients indexes, ALL patients indexes, # of iterations

Output: one-dimensional array with the scores obtained by the filter for each gene

1. allocate memory on GPU and CPU for the one-dimensional array of gene expression, AML/ALL patients indexes and one-dimensional array for the scores
2. fill the one-dimensional array of gene expression with the data of the Leukemia cancer dataset
3. fill the AML/ALL arrays with the round-cross method
4. copy the one-dimensional array of gene expression and AML/ALL patients indexes from CPU to GPU
5. invoke the CUDA kernel for perform the filter (Algorithm 2).
6. copy back the one-dimensional array with the scores from GPU to CPU
7. free both memories, GPU and CPU

C. Support Vector Machine

The Support Vector Machine (SVM) was introduced in 1992 [10], and is widely used in classification problems. The SVM works through of the construction of an N-dimensional hyperplane that is used to separate the data of different classes of a dataset.

Through an initial set of data used as *"training"* , where the classes are known in advance, the SVM creates a function, or a mathematical model, able to separate elements of different classes. Is recognized as *"support vectors"* to the subset of data which are taken as the basis for create the model.

Once generated the mathematical model, the test data are entered for the classification. This process is shown in Figure 2.

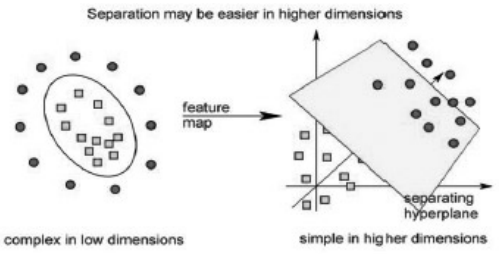


Figure 2. Process of a SVM. Gets training data, generates the mathematical model and applies it on testing data.

In our implementation, the set of patients used as training for the SVM, corresponds to the same patients used in the filter. The values entered to the SVM for training and classification, are the expression values of the 10 "best" genes obtained by the filter. With these data the training of the SVM was performed, and a function or mathematical model able to separate patients from different classes was obtained. After building the model, the final classification tests were performed. For these tests was selected a set of patients whose expression data, and class, were not used in the execution of the filter, nor in some stage prior to these final tests.

On these test patients, and based on the generated model, the SVM estimates the subtype of cancer of each patient. Finally, with the information of the classes estimated by the SVM, and with the information of the actual classes of the patients, we calculate the success rate obtained by the method.

The SVM functions used in this project were taken from [13]. These functions are implemented in R programming language, and work on GPGPU using CUDA technology. To apply these functions, the dataset was scaled to a range of -1 to 1. This range is the recommended in [14]. The kernel used in the SVM was of linear type, and its "C" parameter was set to 2.0. This value of "C" was obtained running the script "grid.py" that belongs to the LIBSVM library [12].

III. Experiments and Results

A. Design of the Experiment

For the tests of the BSS/WSS filter, we structured the Leukemia cancer dataset according to the LIBSVM format [12]. In this format the columns correspond to the genes, and the rows to the patients. The first 25 patients have Leukemia type AML, and the last 47 patients have type ALL. The first column is reserved to identify the class of each patient. Figure 3 shows a representation of the structure of the Leukemia cancer dataset.

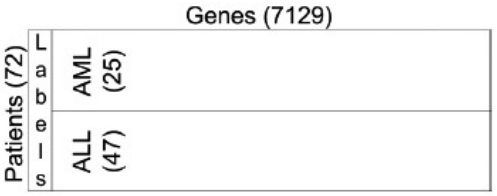


Figure 3. Structure of the Leukemia cancer dataset.

The evaluation of the CUDA implementation consisted in tests of 1000 iterations, performed on a set of training data, which were selected by a method which we called "round-cross". In each iteration were selected a total of 15 patients of class AML, and 27 patients of ALL class. The first 15 patients selected of the AML class correspond to the indexes 1 to 15 of the dataset, and for the ALL class the indexes 26 to 52. In each iteration was incremented, by one, the start of the indexes of the subsets, so that, for the second iteration, the indexes of the subset of AML patients are from

2 to 16 and from 27 to 53 for the subset of ALL patients. In Table 1, are shown the selected sets by the *"round-cross"* method for the first 5 iterations.

Table 1. The first 5 subsets of AML/ALL patients obtained from round-cross for the BSS/WSS filter.

AML	ALL
1 – 15	26 – 52
2 – 16	27 – 53
3 – 17	28 – 54
4 – 18	29 – 55
5 – 19	30 – 56

The experiments were performed with the following equipment:

- Graphic card: Nvidia GeForce GTX 670 with 2GB RAM memory, 1344 CUDA cores, compute capability of 3.0.
- Driver version: 319.49.
- CUDA Toolkit version: 5.5.
- GCC version: 4.7.0 20120507 (Red Hat 4.7.0-5).
- Kernel version: 3.3.4-5.fc17.x86_64.
- GNU/Linux distribution: Fedora release 17 (Beefy Miracle) 64 bits.
- Processor: Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz.
- System ram memory: 4GB.
- B. Results
- The genes

B. Results

The genes considered as "best" for classification, and for which the filter assigns a higher score, have a similar behavior for patients of the same class, and dissimilar behavior for patients of different classes. Figure 4 shows a graph with the expression levels of the 10 best qualified genes by the filter. It is clear that the behavior of these 10 genes is very different for the first 25 patients, that are of type AML, with respect to the remaining 47 patients that are of type ALL. Therefore these 10 genes are most suitable for the classification of subtypes of Leukemia.

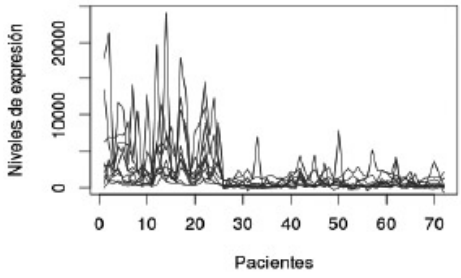


Figure 4. Graph of the expression levels of the best 10 genes qualified by BSS/WSS filter.

Figure 5 shows a graph of the expression levels of the 10 worse qualified genes by the filter. It is appreciable that the behavior of the genes cannot distinguish between the groups of patients with type AML from those of type ALL. Therefore these 10 genes are not suitable to perform the classification of subtypes of Leukemia.

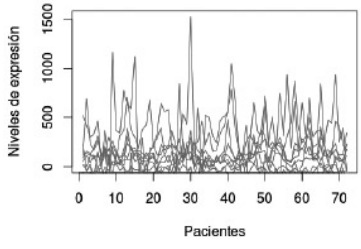


Figure 5. Graph of the expression levels of the worse 10 genes qualified by BSS/WSS filter.

In Figure 6 is shown a comparative graph of the execution time for 1000 iterations, between the parallel implementation with CUDA technology and the sequential implementation with C. It is appreciable that CUDA C had a greater performance compared to C, resulting a difference of performance of about 18 times faster.

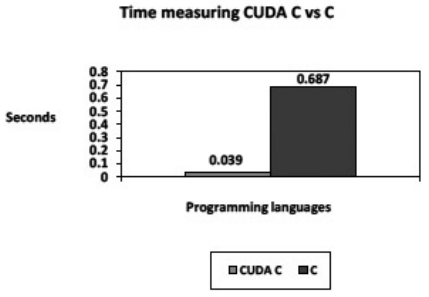


Figure 6. Time measuring of BSS/WSS filter execution. CUDA C vs C for 1000 iterations.

The final classification tests were performed with the SVM, based on the 10 best genes discovered by the filter. These tests were performed using information from 42 patients taken as training, and 15 patients used as test. The results show a success rate of classification of 92%, that is, in average in each iteration, 13.8 of 15 patients of test were correctly classified.

IV. Conclusions

The use of the BSS/WSS filter is very useful for the reduction of the size of the gene expression datasets used for the classification of subtypes of cancer. The significant reduction of the number of genes achieved for the filter makes possible the use of a Support Vector Machine, to perform the classification of cancer with an acceptable success rate.

With the parallel implementation of the BSS/WSS filter on GPGPU using CUDA technology, we have achieved a significant reduction of the execution time, compared to the sequential implementation.

As future work, we plan to implement the filters TStatistic and Wilcoxon in parallel. It is expected that with the use of these two filters, together with the BSS/WSS filter, we could achieve best success rates in the classification. It is also proposed perform tests of classification, for both, diagnostic and prognostic of cancer, with others cancer datasets.

References

[1] World Health Organization, Fact Sheet 310.

[2] Ying Lu, and Jiawei Han, "Cancer Classification Using Gene Expression Data," Journal Systems, vol. 28, no. 4, 2003, pp. 243─268.

[3] Golub et al., "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," Science, vol. 286, 1999, pp. 531─537.

[4] Nasimeh Asgarian, and Russell Greiner. "Using Rank- 1 Biclusters to Classify Microarray Data," Bioinformatics, vol. 00, 2007, pp. 1─10.

[5] U. Alon et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," Proceedings of the National Academy of Sciences of USA, vol. 96, no. 12, 1999, pp. 6745─6750.

[6] Dinnesh Singh et al., "Gene expression correlates of clinical prostate cancer behavior," Cancer Cell, vol. 1, 2002, pp. 203─209.

[7] Gavin G. Gordon et al., "Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma," Cancer Research, vol. 62, 2002, pp. 4963─4967.

[8] Hernández Montiel L.A., Bonilla Huerta E., and Morales Caporal R., "A multiple-filter-GA-SVM method for dimension reduction and classification of DNAmicroarray data," Revista Mexicana de Ingeniería Biomédica, vol. 32, no. 1, 2011, pp. 32─39.

[9] Sandrine Dudoit, Yee Hwa Yang, Matthew J. Callow, and Terence P. Speed, "Statistical Methods For Identifying Differentially Expressed Genes In Replicated cDNA Microarray Experiments," Statistica Sinica 12, 2002, pp. 111─139.

[10] B.E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," Proceedings of the Fifth Annual Workshop on Computational Learning Theory 5, 1991, pp. 144─152.

[11] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, "A Practical Guide to Support Vector Classication," 2010.

[12] Chih-Chung Chang, and Chih-Jen Lin, "LIBSVM: a library for support vector machines." ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[13] (2009) The R's MVS website. [Online]. Available: http://www.r-tutor.com/

[14] Part 2 of Sarle's Neural Networks FAQ Sarle, 1997.

Application of Parallel Processing Based on Multithreading to the Optimal Trajectory Planning for Robot Manipulators

Waldemar Pérez Bailón
Member, IEEE, Departamento de Ingeniería Eléctrica y Electrónica del Instituto Tecnológico de Lázaro Cárdenas. (e-mail: waldemar.perez@itlac.mx).
Antonio Ramos-Paz,
Member, IEEE División de Estudios del Posgrado de la Facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo (e-mail: arpaz@umich.mx).

Abstract

Optimal trajectory planning for robot manipulators is a very important issue in the research field of robotics. Many applications require smooth trajectories and the minimization of a performance index, usually the traveling time or the mechanical energy of the actuator. This paper presents a novel algorithm that uses eighth-degree polynomial functions to generate smooth trajectories for the parametric representation of a given path. In this contribution it is introduced the application of parallel processing based on multithreading to the optimal trajectory planning for robot manipulator in order to improve the execution time of the optimization algorithm proposed. The algorithms presented minimize the mechanical energy consumed by the actuators of the robot manipulator (RM) or the path traveling time of the RM. To solve the optimization model of mechanical energy consumed, a genetic algorithm is implemented, and to solve the optimization model of path traveling time, a method based on a combination of a genetic algorithm and the numerical algorithm known as bisection method has been implemented.

Keywords: *Robot Manipulator, Trajectory Planning, Eighth-Degree Polynomials, Genetic Algorithm, Parallel Processing, Multithreading.*

I. Introducction

The goal of trajectory planning is to generate the reference inputs to the motion control system that ensures that the manipulator executes the planned trajectories. A path denotes the locus of points in the joint space, or in the operational space [1-2]. On the other hand, a trajectory is a path on which a time law is specified, for instance in terms of velocities and/or accelerations at each point. Planning consists of generating a time sequence of the values attained by a polynomial function interpolating the desired trajectory [1-2]. Optimal control of RM requires an optimal trajectory planning. Many applications, especially those handling delicate loads, require smooth trajectories and the minimization of a performance index, usually the traveling time or the mechanical energy of the actuators [3-4]. Polynomials are a common method for defining robot trajectories.

The most commonly used polynomials are the linear interpolation, the third degree polynomials

and the fifth degree polynomials [5]. With linear interpolation, the position is continuous but not the velocity. The cubic polynomial ensures the continuity of position and velocity but not of acceleration. In order to avoid exciting resonances in the mechanics, it is worth ensuring the continuity of acceleration as well. To get this requirement, the interpolation needs a polynomial of at least fifth degree [6]. Higher degree polynomials have been used in the literature, e.g., Amirabdollahian et al. introduced a seventh degree polynomial [7].

Our paper presents a novel algorithm that uses eighth-degree polynomial functions to generate smooth trajectories for the parametric representation of a given path. This eighth degree polynomial includes a parameter that defines the maximum velocity of the trajectory and allows the generation of a big number of different smooth trajectories for a given path. The optimization algorithm presented in this paper minimizes the mechanical energy consumed by the RM actuators or the path traveling time of the RM with the constraint that torque limits of the actuators are satisfied.

To solve the optimization model of mechanical energy consumed, a genetic algorithm is implemented, and to solve the optimization model of the path traveling time, a method based on a combination of a genetic algorithm and the numerical algorithm known as bisection method has been implemented [8-9]. This contribution incorporates parallel processing technology based on multithreading processes to get a speed-up in the execution of the optimal trajectory planning algorithms [10-13].

II. Generation Of Trajectories

The trajectory planning process generates the reference inputs to the motion control system that ensures that the manipulator executes the planned trajectories. Typically, the user specifies a number of parameters to describe the desired path. Planning consists of generating a time sequence of the values attained by a polynomial function interpolating the desired trajectory. The locus of the trajectory is the set of Cartesian points. It may be a line, arc, ellipse, etc. or a combination of these paths. The locus should preferably be expressed in parametric form. A linear segment in parametric form can be formulated as follows: If $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$ are the initial and final points of the linear segment, respectively, $p_1 \in \Re^3$, and $p_2 \in \Re^3$ are the vectors that describe each point, and $P(t) \in \Re$ is a time variable parameter existing in the interval $[0, \|p_2 - p_1\|]$, the parametric representation of the linear segment is.

$$\bar{P}(t) = \bar{p}_1 + \left(\frac{\bar{p}_2 - \bar{p}_1}{\|\bar{p}_2 - \bar{p}_1\|} \right) P(t) \qquad (1)$$

where $\bar{P}(t) \in \Re^3$ is the vector associated with each point in the linear segment. Expanding (1),

$$\bar{P}(t) = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + \frac{\begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix}}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} P(t) \qquad (2)$$

If $\bar{P}(t) = (x(t)\ y(t)\ z(t))^T$, the equations of the linear segment in parametric form are defined by (3), (4) and (5)

$$x(t) = x_1 + \left(\frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \right) P(t) \quad (3)$$

$$y(t) = y_1 + \left(\frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \right) P(t) \quad (4)$$

$$z(t) = z_1 + \left(\frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \right) P(t) \quad (5)$$

III. Eighth-Degree Polynominal Interpolation

Eighth-degree polynomial functions can be used to generate smooth trajectories for the parametric representation of a given path. The Eighth-degree polynomial is defined by (6), its first derivative is given in (7) and its second derivative in (8).

$$P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 + a_7 t^7 + a_8 t^8 \quad (6)$$

$$\dot{P}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 + 6a_6 t^5 + 7a_7 t^6 + 8a_8 t^7 \quad (7)$$

$$\ddot{P}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 + 30a_6 t^4 + 42a_7 t^5 + 56a_8 t^6 \quad (8)$$

Table I shows the constraints that must be satisfied by the eighth-degree polynomial interpolation. t_f, is the total traveling time, q_f is the total length of the path, and V_{max} is the maximum velocity during the trajectory.

Table I: Constraints To Be Satisfied By The Eighth Degree Polynomial Interpolation.

$t = 0$	$t = \frac{t_f}{2}$	$t = t_f$
$p(t) = 0$	$p(t) = \frac{q_f}{2}$	$p(t) = q_f$
$\dot{P}(t) = 0$	$\dot{P}(t) = v_{max}$	$\dot{P}(t) = 0$
$\ddot{P}(t) = 0$	$\ddot{P}(t) = 0$	$\ddot{P}(t) = 0$

Evaluating (6), (7) and (8) to satisfy the constraints, the coefficients of (6) are obtained. These coefficients are given by equations from (9) to (15).

$$a_0 = a_1 = a_2 = 0 \quad (9)$$

$$a_3 = \frac{70 q_f}{t_f^3} - \frac{32 v_{max}}{t_f^2} \quad (10)$$

$$a_4 = \frac{-315 q_f}{t_f^4} + \frac{160 v_{max}}{t_f^3} \quad (11)$$

$$a_5 = \frac{546 q_f}{t_f^5} - \frac{288 v_{max}}{t_f^4} \quad (12)$$

$$a_6 = \frac{-420 q_f}{t_f^6} + \frac{224 v_{max}}{t_f^5} \quad (13)$$

$$a_7 = \frac{120 q_f}{t_f^7} - \frac{64 v_{max}}{t_f^6} \quad (14)$$

$$a_8 = 0 \quad (15)$$

Substituting (9) – (15) in (6), (7) and (8), the position, velocity, and acceleration, respectively, can be obtained. The position is given by (16).

$$P(t) = q_f \left(\frac{70}{t_f^3} t^3 - \frac{315}{t_f^4} t^4 + \frac{546}{t_f^5} t^5 - \frac{420}{t_f^6} t^6 + \frac{120}{t_f^7} t^7 \right) + v_{max} \left(\frac{-32}{t_f^2} t^3 + \frac{160}{t_f^3} t^4 - \frac{288}{t_f^4} t^5 + \frac{224}{t_f^5} t^6 - \frac{64}{t_f^6} t^7 \right) \quad (16)$$

If V_{max}, is defined by using a parameter that it will be called *index_m*, as it is shown in (17), a big number of different eighth-degree polynomial trajectories can be generated for the parametric representation of a given path.

$$v_{\max} = (index\_m)\frac{q_f}{t_f} \qquad (17)$$

To generate smooth trajectories, the *index_m* parameter must be in the interval [1.458334, 2.187500], as it is illustrated in Fig. 1. If the *index_m* parameter is outside this interval, the eighth degree polynomial generates non-natural movements, (Fig. 2).

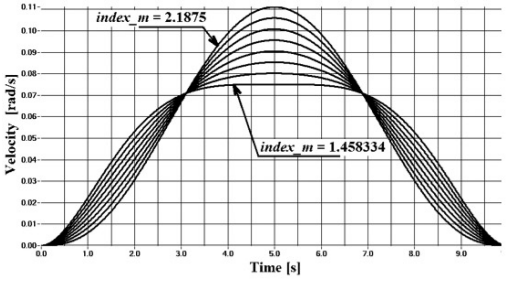


Fig. 1. Speed profiles of the robot manipulator when *index_m* is within the interval of natural movements.

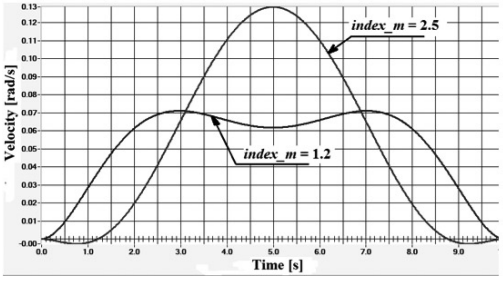


Fig. 2. Speed profiles of the robot manipulator when *index_m* is outside the interval of natural movements

It can be noted that if *index_m*,= 1.458334, the slope of the velocity curve generated is very sharp at the beginning and at the end of the trajectory, and that this trajectory has the lowest V_{max},. It can be noted also that if *index_m*= 2.187500 the slope of the velocity curve is very soft at the beginning and at the end of the trajectory, and that this trajectory has the highest V_{max}, [14].

IV. Inverse Manipulator Kinematic

The objective of the inverse kinematics is to compute the set of joint angles corresponding to a given end-effector position and orientation [1-2]. The implementation of the manipulator kinematics is based on the Denavit-Hartenberg notation. A six-degree of freedom (DOF) robot manipulator was simulated to test the trajectory-planning algorithm proposed in this paper. Fig. 3 shows the structure of this robot manipulator and the coordinate reference systems associated with each joint. Table II shows the Denavit-Hartenberg parameters for this robot.

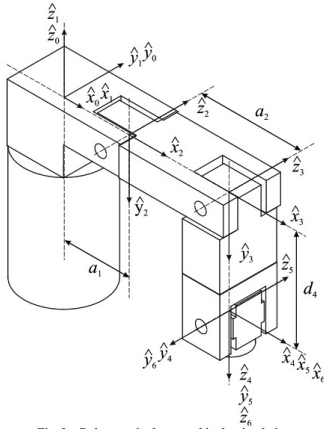


Fig. 3. Robot manipulator used in the simulation

Table II: Denavit-Hartenberg Parameters Of The Robot Manipulator

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	–90	a_1	0	θ_2
3	0	a_2	0	θ_3
4	–90	0	d_4	θ_4
5	90	0	0	θ_5
6	–90	0	0	θ_6

V. Robot Manipulator Dynamics

The dynamic model of a RM provides a description of the relationship between the joint actuators torques and the motion of the structure [1-2]. The dynamic model plays an important role for simulation of motion, analysis of manipulator structures, and design of control algorithms. Simulating manipulator motion allows testing control strategies and trajectory planning algorithms without the need to use a physically available system. The software platform that has been developed to test the optimal trajectory-planning algorithm considers that the robot manipulator, has rigid links and that the inertia tensor and link masses are specified. To calculate the joint actuator torques during a trajectory, the Newton-Euler algorithm [1-2] has been implemented.

VI. Mechanical Energy Optimization

One of the algorithms presented in this paper minimizes the mechanical energy consumed by the robot manipulator actuators with the constraint that torque limits of the actuators are satisfied. To solve the optimization model, a genetic algorithm is implemented. The use of eighth-degree polynomial functions allows the generation of a big number of smooth trajectories for a given path. This is very convenient for the genetic algorithm. Fig. 4 shows the pseudo-code of the software designed to implement the optimal trajectory-planning algorithm.

```
1. Begin
2.    tf = path traveling time
3.    Randomly generate an initial population of 100 index_m
parameters (P)
4.    Repeat
5.   Calculate the trajectory for each index_m parameter (P)
6.      Calculate the robot inverse kinematics for each trajectory
P)
7.      Calculate the robot dynamics for each trajectory P)
8.      Calculate the mechanical energy consumption of the robot
for each trajectory P)
9.      Rank population members based on the mechanical energy
consumption (P)
10.     Randomly select a new population (NP)
11.     Mate best members of population (NP)
12.     Mutate resulting members (NP)
13.     P ← NP
14.   Until convergence criterion is reached or loop=40
15. End
```

Fig. 4 Pseudo-code of the trajectory planning algorithm whit minimization of mechanical energy

Given a desired path, the optimization process uses 100 individuals, i.g. 100 smooth trajectories in each iteration. The *index_m* parameters are represented using 20 bits. The population members are ranked based on the mechanical energy consumed by the six joint actuators of the robot manipulator. The couples to generate the new population are randomly selected using a method that emulates the darts game [8]. For the mutation process a mutation operator of 0.25% is applied. The convergence criterion of the algorithm is that the iteration process stops when the difference of the energy consumed by the first 96 ranked trajectories is smaller than a given resolution.

VII. Path Traveling Time Minimization

The other algorithm presented in this paper minimizes the path traveling time of a RM. To solve the optimization model, a method based on a combination of a genetic algorithm and the numerical algorithm known as bisection method has been implemented, with the constraint that torque limits of the actuators are satisfied. This algorithm also uses eighth-degree polynomial functions to generate a big number of smooth trajectories for a given path. Fig. 5 shows the pseudo-code of the software designed to implement the path traveling time minimization algorithm.

```
1.  Begin
2.  flag_start = 1
3.  tf = initial guess for the path traveling time
4.  tf_minimum = initial value of the searching internal left limit
5.  tf_maximum = initial value of the searching internal right limit
6.  delta_t = tf_maximum – tf _minimum
7.  torque_ratio_maximum = 0
8.  Repeat
9.    If (delta_t > 0.001 or torque_ratio_maximum < 0.999) {
10.   Randomly generate an initial population of 100 index_m
      parameters (P)
11.       Repeat // for the path traveling time tf
12.           Calculate the trajectory for each index_m parameter (P)
13.           Calculate the robot inverse kinematics for trjectory (P)
14.           Calculate the robot dynamics for each trajectory (P)
15.           Calculate the maximum torque ratio given by
                  max { torque_ratio[i] = torque_real[i] / torque_nominal[i] }
                  i = 1,2,...,6
16.           Rank population members based on the maximum torque
              ratio
17.           Randomly select a new population (NP)
18.           Mate best members of population (NP)
19.           Mutate resulting members (NP)
20.           P ← NP
21.       Until global maximum of the torque ratio is found for the path
          traveling time tf
22.   If(flag_start ==1)
23.   If( torque_ratio_maximum > 1) {
24.   flag_start = 0
25.   torque_ratio_maximum = 0
26.   tf = 10tf  }
27.   If(torque_ratio_maximum < 1){
28.   flag_start = 0
29.   tf_maximum = tf
30.   tf = (tf_maximum – tf_minimum)/2 + tf_minimum  }
31.   Else {
32.   flag_start = 0
33.   tf_minimum = tf
34.   tf = (tf_maximum – tf_minimum)/2 + tf_minimum  }
35.   delta_t = tf_maximum – tf minimum
36. Until (delta_t ≤ 0.001)  }
37. End
```

Fig. 5 Pseudo-code of the trajectory planning algorithm whit minimization of the path traveling time

The minimization algorithm uses the genetic algorithm in the internal loop to find the trajectory with the maximum torque for a given traveling time. The external loop uses the bisection method, where the path traveling time is incremented if the maximum torque found by the genetic algorithm is smaller than the corresponding torque limit. The traveling time is reduced if the maximum torque found by the genetic algorithm is smaller than the corresponding torque limit. The minimization process stops when the maximum torque of the RM is close to the corresponding nominal torque with a resolution of 0.1 %.

IX. Parallel Processing

In order to improve the execution time of the mechanical energy and the path traveling time optimization algorithms, a parallel processing technique based on multithreading has been used in both algorithms. Multithreading [10-11] is the application of lightweight sub processes executed within a process sharing code and data segments, but with their own program counter, machine registers and stack. Global and static variables are common to all threads. To measure the improvement in the execution time of the optimization algorithms when the parallel processing technique is applied, the concept of speed-up (S) is used. This concept is defined as follows [16]:

$$S = \frac{T(1)}{T(P)} \tag{18}$$

Where, *T(P)* is the total execution time when a parallel processing system with (*P*) processing elements are used. This factor indicates how many times the parallel program is faster than its sequential counterpart. Figure 6 shows the application of the parallel processing technique to the mechanical energy minimization algorithm.

The master thread randomly generates a population of 100 *index_m* parameters. Then the master thread divides this population between the number of processing elements, i.e. the number of threads available. Each thread receives a determined number of *index_m* parameters. For each of these *index_m*parameters, each thread calculates a trajectory, computers the inverse kinematics, solves any existing interior singularity, computes the dynamics, and calculates the mechanical energy consumed by the robot manipulator actuators.

After all threads finish, the master thread receives the mechanical energies corresponding to the population of 100 *index_m* parameters. The master thread ranks the population members based on the mechanical energy consumed by the six joint actuators of the RM. If the convergence criterion is not reached, the master thread generates a new population of 100 *index_m* parameters. The process is repeated until the convergence criterion is reached or a maximum number of iteration has been completed.

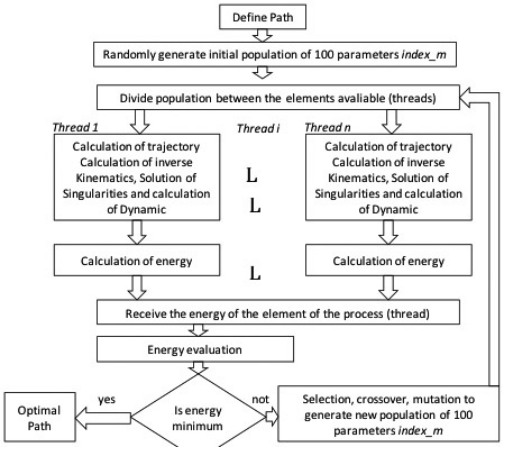


Fig. 6 Mechanical energy optimization algorithm using parallel processing

Figure 7 shows the application of the parallel processing technique to the path traveling minimization time algorithm. The algorithm starts by setting the initial guess for the path traveling time and the searching time interval for the successive bisection method. In this path traveling time optimization algorithm, the genetic algorithm is used to find the maximum torque ratio corresponding to a path traveling time. If the maximum torque ratio is greater than, i.e. if there is a torque violation, equations (19) and (20) are applied.

$$t_{f\min} = t_f \tag{19}$$

$$t_f = \frac{t_{f\max} - t_{f\min}}{2} + t_{f\min} \tag{20}$$

If it maximum torque ratio is smaller than1, equations (21) and (22) are applied.

$$t_{f\max} = t_f \tag{21}$$

$$t_f = \frac{t_{f\max} - t_{f\min}}{2} + t_{f\min} \tag{22}$$

The process is repeated until the maximum torque ratio is equal to 0.999 or t_{fmax}-t_{fmin}=0.001 s.

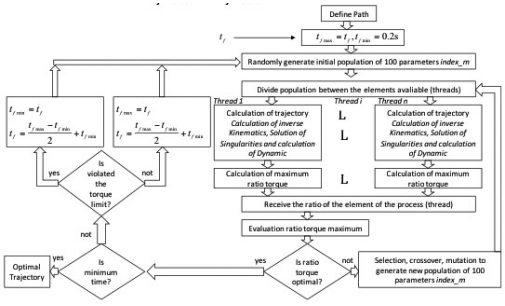


Fig. 7 Path traveling time optimization algorithm using parallel processing.

X. Test Cases

The optimal trajectory planning algorithm has been implemented on the operating system Linux Ubuntu [15]. A two processors 3.06 GHz quad core computer was used to execute the developed software platform. The two test cases presented in this paper were carried out for the RM shown in Fig. 3. In the case of the mechanical energy optimization algorithm, the test was carried out for the linear segment path between the points p_1 =(0.4,0.0,0.5) and p_2=(0.15,0.0,0.05), with a total traveling time of t_f=1.0 s, and a sampling period of one millisecond.

Table III shows the optimal trajectory found with the mechanical energy optimization algorithm. The results of the optimization trajectory planning algorithm was compared with an exhaustive algorithm that evaluates all possible smooth trajectories generated varying the *index_m* parameter in the interval [1.458334, 2.187500] with increments of $\mathbf{1x10^{-6}}$.

Table III: Optimal Trajectory Found With The Mechanical Energy Optimization Algorithm

	Computer time [s]	*Index_m*	Mechanical Energy [J]	Speed-up
Exhaustive algorithm	8672.102	1.925408	25-143368	-------
Genetic algorithm	45.14	1.925412	25.143368	1.000
Parallel genetic algorithm	7.236	1.925412	25.143368	6.239

The optimal trajectory found with this algorithm has *index_m*=1.925408 and the corresponding minimal mechanical energy consumed in the robot manipulator is 25.0143368 J. The mechanical energy optimization algorithm executed with one processor has an execution time 192 times faster than the exhaustive algorithm. This is achieved within the constraints to meet the proposed.

In the case of the path traveling time optimization algorithm, the test was carried out for the linear segment path between the points p_1 =(0.3,0.2,0.05) and p_2=(-0.3,0.2,0.05), with a total traveling time of t_f=1.0 s and a sampling period of one millisecond.

Table IV shows the trajectory optimal found with the path traveling time optimization algorithm. The optimal trajectory found with this algorithm has *index_m*=1.490444 and the corresponding minimal path traveling time in the robot manipulator is 0.579970 s.

Table IV : Optimal Trajectory Found With The Path Traveling Time Optimization Algorithm

	Computer time [s]	*Index_m*	Minimum traveling $T_{fmin}[s]$	Speed-up
Genetic algorithm	924.878	1.490444	0.57997	1.000
Parallel genetic algorithm	123.794	1.490444	0.57997	7.458

Figure 8, and 9, show the torques and angular velocities of joint 1, y 3 respectively, corresponding to the optimal trajectory. It can be seen that the constraint for torque limits has been satisfied.

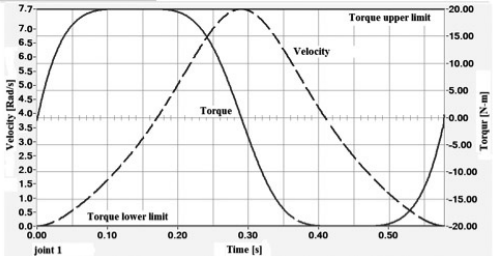


Fig. 8. Torque and rotational velocity in joint 1

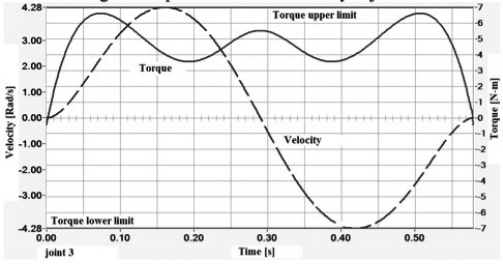


Fig 9. Torque and rotational velocity in joint 3

XI. Conclusions

A software platform with two optimal trajectory planning algorithms has been presented. The algorithms are based on eight-degree polynomial functions, a genetic algorithm and a parallel processing technique. The eighth-degree polynomial functions include a parameter that defines the maximum velocity during the trajectory and the velocity waveform. The use of this parameter allows the generation of a big number of smooth trajectories for a given path. This is very convenient for the genetic algorithm. One of the algorithms presented in this paper minimizes the mechanical energy consumed by the robot manipulator actuators with the constraint that torque limits of the actuators are satisfied.

The other algorithm presented in this paper minimizes the path traveling time of a robot manipulator. The cases study presented shows the effectiveness of the optimal trajectory planning algorithm and the software platform developed. In the case of the mechanical energy optimization algorithm the application of parallel processing technology to the optimal trajectory planning algorithm showed a speed-up close to 6 when 8 threads multithreading ware applied to the case study presented in this paper.

In the case of path traveling time optimization algorithm the application of parallel processing technology to the optimal trajectory planning algorithm showed a speed-up close to 7.5 when 8 threads multithreading ware applied to the test cases presented in this paper. This paper introduced the application of parallel processing techniques based on multithreading to the optimal trajectory planning for robot manipulator in order to minimize the execution time of the optimization algorithm proposed and its application in real time studies.

XII. References

[1] John J. Craig, Introduction Robotics Mechanics & Control, Addison-Wesley Publishing Company, 1986.

[2] L. Sciavicco, B. Siciliano. Modeling and Control of Robot Manipulators, 2nd edition, Springer-Verlag London Limited, 2000.

[3] Kang G. Shin and Neil D. McKay. "Minimum-Time Control of robotic Manipulator with Geometric path Constraints". IEEE Transaction on Automatic Control, vol. AC-30, N0 6, junio/1985.

[4] Shugen Ma and Mitsuru Watanabe. "Minimum Time Path-tracking Control of Redundant Manipulator". Proceedings of the 2000 IEEE/RsJ International Conference on Intelligent Robots and Systems.

[5] Kee-Whan Kim, Hyun-Sink and Young-Kiu Choi. "Optimization of Cubic Polynomial Joint Trajectories and Sliding Mode Controllers for Robots Using Evolution Strategy". IEEE International conference On Industrial Electronic Control and Instrumentation, 1997. IECON 97 23rd.

[6] W. Khalil and E. Dombre, Modeling, Identification and Control of Robots, Kogan Page Science, London, 2002.

[7] F. Amirabdollahian, R. Loureiro, and W. Harwin, "Minimum Jerk Trajectory Control for Rehabilitation and Haptic Applications, 2002 IEEE International Conference on Robotics and Automation, May, 2002.

[8] Z. Michalewicz, Genetic Algorithms + Data Structure = Evolution Programs, Springer, 1999

[9] D. C. Monteiro and M. K. Madrid. "Planning of Robot Trajectories with Genetic Algoritms". Robot Motion and Control, 1999. Proceeding of the First Worsohp. 28 Junio 1999.

[10] N. Herlihy, N. Shavit. The Art of Multiprocessor Programming", Morgan Kaufmann 1 edition , 2012.

[11] P. Pacheco. An Introduction to Parallel Programming. Morgan Kaufmann 1 edition , 2011

[12] Salman Yussof, Rina Azlin Razali, Ong Hang See, "A Parallel Genetic Algorithm for Path Routing Problem", International Conference on Future Computer and Comunication, IEEE CONFERENCE, 2009.

[13] Yi Liu, Xin Zhang, He Li, Depei Qian, Allocasting Tasks in Multi-core Processor based parallels Systems", Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on. IEECONFERENCE, 2007.

[14] I. Juarez Campos, Sobre las Regiones Geométricas Articulares y su Aplicación Mediante Estrategias Evolutivas, Doctoral Thesis (In Spanish), Universidad Autónoma de México, 2002.

[15] [online]. Avaliable: www.ubuntu.com

[16] Foster I., "Designing and Building Parallel Programs", Addison Wesley 1994.

Simulating a Hyperbolic-PDE using a 2-dimensional CA

[1] Huerta-Trujillo, [2] J.C. Chimal-Eguía
[1]Simulation and Modeling Lab. Computing Research Center,
National Polytechnic Institute, (e-mail: ihuerta@ipn.mx).
[2]*Simulation and Modeling Lab. Computing Research Center,*
National Polytechnic Institute, (e-mail: chimal@cic.ipn.mx).

Abstract

This paper presents a parallel model based on 2-dimensional Cellular Automaton (CA); likewise the process for obtaining the evolution rule. The model obtained is compared with the analytical solution of a partial differential equation of the two-dimensional linear and homogeneous hyperbolic type. This models a vibrating membrane with specific initial and border conditions. The frequency spectrum and the error between the data obtained from the CA model is analyzed versus the data provided by the solution evaluation to the differential equation.

On the other hand, CA complexity is analyzed and the implementation using parallel computing to reduce the computational complexity is guaranteed.

***Index Terms**: Cellular automaton, parallel computing, partial differential equations, vibrant membrane.*

I. Introduction

A cellular automaton (CA) is a discrete dynamical system, consisting of an array of cells (nodes) in some d-dimension [1]. Wolfram [2] defined it as mathematical idealization of physical systems whose time and space are discrete in which the physical quantities can be a finite set of values.

The CA definition is associated with others concepts, space and influence area. It is assumed that the representing system is distributed into space; closer regions have more influence between each other than other distant regions in the system [3].

The elements of a CA are:

- A regular lattice of N finite identical state machines or cells [4], filling up the n-dimensional space; each one has identical patterns and local connections to the others.
- A set of states that can be assigned to each cell.
- An evolution rule that specifies the transition of cells between states over time.
- The local interactions of each cell is with others only in its neighborhood

A. Cellular Automaton: A Formal Definition

Def 1: A lattice is generally a regular infinity array [2] formed by identical objects called cells. This array can be n-dimensional; usually for natural

systems simulations can be of 1, 2 or 3 dimensions of finite size.

Def 2: A CA is a 4-tuple $CA=(L,S,V,\ \Phi)$ where:

L: Is a regular d-dimensional lattice and $L=\{c \in C^d\}$

S: Is the finite set of all possible states for the cells, $c \in L$.

V: Is the finite set of cells which form the neighborhood to a cell.

Φ: Is the evolution function which is applied simultaneously to all cells in the lattice.

The update of the cells state requires knowing the states of neighbor cells [5].

Def 3: A neighborhood for $c \in L$, is a set $V(c)=\{k_1,K_2,\ldots,k_n \mid k_j \in L, j=1,2,\ldots,n\}$;

II. Classical Vibrant Membrane Model

The moving equation of a membrane [6] is based on the assumption of being thin and regular with negligible stiffness; it is perfectly elastic without cushioning and its vibration has small amplitude displacements. This equation is represented [7] by:

$$\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} \qquad (1)$$

Where $u(x,y,t)$ is the initial deflection of the membrane and $c^2= T/\rho$, being ρ the mass density of the membrane [8] and T the stress membrane per unit length. The equation (1) is called *two-dimensional wave equation.*

Taking a square membrane (Fig. 1), border conditions are defined as follows:

$$u(x,y,t)=0 \text{ for } x=0, x=b, y=0, y=b \forall t \qquad (2)$$

Initial conditions are defined as follows:

$$u(x,y,t) = f(x,y) \qquad (3)$$

$$\frac{\partial u(x,y,t)}{\partial t}\Big|_{t=0} = g(x,y) \qquad (4)$$

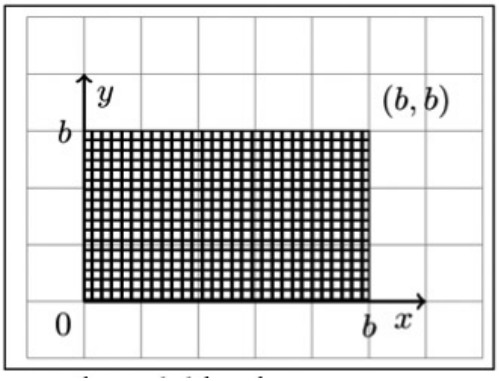


Fig. 1. Square membrane, *bxb* length.

Where $f(x,y)$ and are the displacement and the initial speed of the membrane respectively.

If the initial conditions are defined as:

$$u(x,y,t) = xy(x-b)(y-b) \qquad (5)$$

$$\frac{\partial u(x,y,t)}{\partial t}\Big|_{t=0} = 0 \qquad (6)$$

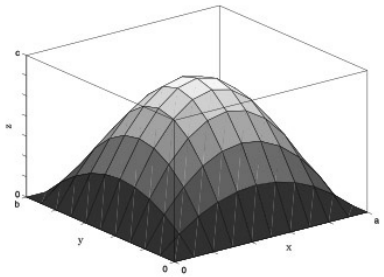


Fig. 2. Initial and border conditions for membrane system.

Then the membrane starts from the rest with a spatial distribution that is possible as seen in Fig. 2.

The solution to the differential equation (1) with the initial conditions described above is obtained as follows:

Suppose that $u(x,y,t)=F(x,y)T(t)$, replacing in (1) it has:

$$F\frac{\partial^2 T}{\partial t^2} = c^2\left(\frac{\partial^2 F}{\partial x^2}T + \frac{\partial^2 F}{\partial y^2}T\right) \quad (7)$$

Clearing $T(t)$:

$$F\ddot{T} = c^2 T\left(\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2}\right) \quad (8)$$

$$\Rightarrow \frac{\ddot{T}}{c^2 T} = \frac{F_{xx} + F_{yy}}{F} \quad (9)$$

If $F(x,y)=X(x)Y(y)$ and substituting in (9), then:

$$\frac{1}{c^2}\frac{\ddot{T}}{T} = \frac{X''Y + XY''}{XY} \quad (10)$$

$$\Rightarrow \frac{1}{c^2}\frac{\ddot{T}}{T} = \frac{X''}{X} + \frac{Y''}{Y} \quad (11)$$

Because both equation (11) sides are dependent on different variables, its reduction can be equal to a constant called $-k^2$, then:

$$\Rightarrow \frac{1}{c^2}\frac{\ddot{T}}{T} = \frac{X''}{X} + \frac{Y''}{Y} = -k^2 \quad (12)$$

Which leads to the next ordinary differential equations (ODE's):

$$\ddot{T} + c^2k^2T = 0 \quad (13)$$

$$\& \frac{X''}{X} + \frac{Y''}{Y} = -k^2 \Rightarrow \frac{X''}{X} = -\frac{Y''}{Y} - k^2 \quad (14)$$

In the same manner both sides of (14) depend on different variables. This one must to be equal to a constant called -l2, then:

$$\frac{X''}{X} = -\frac{Y''}{Y} - k^2 = -l^2 \quad (15)$$

Which leads to:

$$\frac{X''}{X} = -l^2 \Rightarrow X'' + l^2X = 0 \quad (16)$$

$$\& -k^2 - \frac{Y''}{Y} = -l^2 \Rightarrow Y'' + \alpha^2 Y = 0 \quad (17)$$

Where $a^2=k^2+l^2$. Given (13), (16) and (17) are ODE's, it is possible to obtain a general solution, which is:

$$X(x) = A\cos(lx) + B\sin(lx) \quad (18)$$

$$Y(y) = C\cos(\alpha y) + D\sin(\alpha y) \quad (19)$$

$$T(t) = E\cos(ckt) + F\sin(ckt) \quad (20)$$

In order to make a unique solution, the boundary conditions described in (2) are taken into account and this obtains $u(x,y,0)=0$, given that $u(x,y,0)=X(x)Y(y)T(0)=x(x-b)y(y-b)$ then $T(0)\neq 0$, there are two options $X(x)=0$ or $Y(y)=0$. This only can be possible in four cases, when $x=0$, $x=b$, $y=0$ and $y=b$, because of this it follows that:

$$X(0) = A\cos(l0) + B\sin(l0) = A = 0$$

$$(b) = B\sin(lb) = 0 \Rightarrow lb = m\pi \Rightarrow l = \frac{m\pi}{b} \text{ for } m = 1,2,3,\ldots$$

Similarly:

$$Y(0) = C\cos(\alpha 0) + D\sin(\alpha 0) = C = 0$$

$$Y(b) = D\sin(\alpha b) = 0 \Rightarrow \alpha b = n\pi \Rightarrow \alpha = \frac{n\pi}{b} \text{ for } n = 1,2,3,\ldots$$

With $a^2=k^2+l^2$ implying that $k^2=l^2-\alpha^2$, replacing the values for α and l, has the following:

$$k^2 = \left(\frac{m\pi}{b}\right)^2 + \left(\frac{n\pi}{b}\right)^2 \Rightarrow k_{mn}^2 = \frac{(m^2+n^2)\pi^2}{b^2}$$

The solution to the general equations (18), (19) and (20), depends on the value taken by m and n; the particular solution is:

$$X_m(x) = B_m\sin\left(\frac{m\pi x}{b}\right); m = 1,2,3,\ldots \quad (21)$$

$$Y_n(y) = D_n\sin\left(\frac{n\pi y}{b}\right); n = 1,2,3,\ldots \quad (22)$$

From this, it follows that the solution for (20) is:

$$T_{m,n}(t) = E_{m,n}\cos(ck_{m,n}t) + F_{m,n}\sin(ck_{m,n}t) \quad (23)$$

With $m=1, 2, 3,\ldots$ and $n=1, 2, 3,\ldots$

The particular solution to (1) is constructed from the equations (21), (22) and (23), obtaining:

$$u_{m,n}(x,y,t) = [G_{m,n}\cos(ck_{m,n}t) + H_{m,n}\sin(ck_{m,n}t)] \sin(\frac{m\pi x}{b})\sin(\frac{n\pi y}{b}) \quad (24)$$

Where $m=1,2,3, \ldots, n=1,2,3, \ldots$ and $k_{m,n}=\frac{\pi\sqrt{m^2+n^2}}{b}$

The solution (24) depends directly on m and n values; then each combination of these is a particular solution to (1). By the superposition principle, a linear combination of the solutions is also a solution to the differential equation having:

$$u(x,y,t) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} u_{m,n}(x,y,t) \quad (25)$$

Using Fourier series, the final solution is:

$$u(x,y,t) = \frac{64b^4}{\pi^6} \sum_{\substack{m=1 \\ m=odd}}^{\infty} \sum_{\substack{n=1 \\ n=odd}}^{\infty} \frac{1}{m^3n^3} \cos\left(\frac{\pi\sqrt{m^2+n^2}}{b} ct\right) \sin(\frac{m\pi x}{b})\sin(\frac{n\pi y}{b}) \quad (26)$$

The next section shows the process to make the discrete analytical model.

III. Discretization Of Analytical Model

A. Previous Analysis

Suppose that a membrane is a sequence of points connected by a specific mass with springs (see Fig. 3), each point of the membrane is attached to the four orthogonal neighbors, where the mass of the membrane is spread over the attachment points rather than springs, and the edge of the membrane is subject to a surface.

Call d_e the equilibrium distance which is the spring length to join two masses at rest. This will be the length to separate the masses.

For a vibrating membrane system with initial conditions the position described in equation (5) and from rest, each junction of the membrane is subjected to four forces acting in the direction of each orthogonal neighbor, called m_c, m_n, m_s, m_e, m_w for the central particle, north, south, east and west respectively, as shown in. Fig. 4

It is necessary to know the force exerted for each neighbor over m_c. Following the procedure defined by [9], it proceeds to calculate for one neighbor.

Fig. 3. Membrane representation as a succession of masses and springs, each mass is attached to its neighbors north, south, east and west.

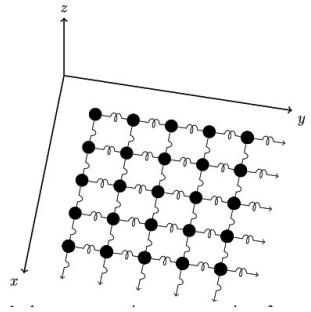
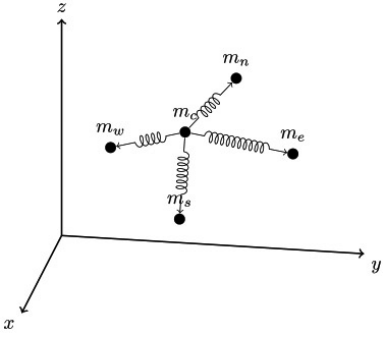


Fig. 4. Von Newman's neighborhood representation for CA, the central cell, its four orthogonal neighbors and the direction in which the force is exerted by the neighbors.



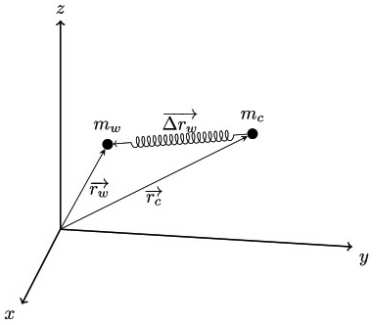


Fig. 5 Representation of a neighbor and the force direction exercised by the spring over the central cell.

Considering Fig. 5 has:

$$\vec{r_c} + \vec{\Delta r_w} = \vec{r_w} \quad (27)$$
$$\Rightarrow \vec{\Delta r_w} = \vec{r_w} - \vec{r_c}$$

The $\vec{\Delta r_w}$ vector can be expressed as the product of a unit vector multiplied by its module, then:

$$\vec{\Delta r_w} = |\vec{\Delta r_w}|\hat{\Delta r_w}$$

The vector $|\vec{\Delta r_w}|$ represents the length between masses; this scalar can be written as the sum of the equilibrium distance or the spring length plus the deformation undergone by the spring due to the change of position of the central mass. so it has:

$$|\vec{\Delta r_w}|\hat{\Delta r_w} = (d_e + \Delta d_e)\hat{\Delta r_w}$$

Where Δd_e is the value of the spring deformation, which must be calculated in order to know the increased force from the rest point to the analysis point, thus:

$$|\vec{\Delta r_w}|\hat{\Delta r_w} = (d_e + \Delta d_e)\hat{\Delta r_w}$$
$$\Rightarrow |\vec{\Delta r_w}|\hat{\Delta r_w} = d_e\hat{\Delta r_w} + \Delta d_e\hat{\Delta r_w}$$
$$\Rightarrow \Delta d_e\hat{\Delta r_w} = |\vec{\Delta r_w}|\hat{\Delta r_w} - d_e\hat{\Delta r_w}$$
$$\Rightarrow \Delta d_e\hat{\Delta r_w} = (|\vec{\Delta r_w}| - d_e)\hat{\Delta r_w} \quad (28)$$

Since both elements of the equation (28) are vectors, it is possible to represent its components as follows:

$$\Delta d_e\hat{\Delta r_w} = (\Delta x_w, \Delta y_w, \Delta z_w) \quad (29)$$
$$(|\vec{\Delta r_w}| - d_e)\hat{\Delta r_w} = (|\vec{\Delta r_w}| - d_e)\left(\frac{x_w - x_c}{|\vec{\Delta r_w}|}, \frac{y_w - y_c}{|\vec{\Delta r_w}|}, \frac{z_w - z_c}{|\vec{\Delta r_w}|}\right) \quad (30)$$

Equating, member to member from the vector equations (29) and (30) has:

$$\Delta x_w = (|\vec{\Delta r_w}| - d_e)\frac{x_w - x_c}{|\vec{\Delta r_w}|} \quad (31)$$
$$\Delta y_w = (|\vec{\Delta r_w}| - d_e)\frac{y_w - y_c}{|\vec{\Delta r_w}|} \quad (32)$$
$$\Delta z_w = (|\vec{\Delta r_w}| - d_e)\frac{z_w - z_c}{|\vec{\Delta r_w}|} \quad (33)$$

Thus obtaining the rectangular components Δx_w, Δy_w, and Δz_w that correspond to increases in displacement in the coordinate axes x, y and z for mass m_c to $\vec{\Delta r_w}$, the components $\vec{\Delta r_n}$, $\vec{\Delta r_s}$, and $\vec{\Delta r_e}$ vectors are calculated in a similar way.

Continuing the analysis by Hooke's law for a mass-spring system in one dimension we have:

$$F = -k\Delta x \quad (34)$$

In general, for the particle mc, there are three forces exerted by m_w in the $\vec{\Delta r_w}$ direction because of its components x, y and z, and so for each of the m_n, m_s and m_e in the directions $\vec{\Delta r_n}$, $\vec{\Delta r_s}$, and $\vec{\Delta r_e}$ respectively. Substituting the values found in equation (31) and the respective neighboring m_c, in the equation (34) obtains:

$$F_x = -k_n\Delta x_n - k_s\Delta x_s - k_e\Delta x_e - k_w\Delta x_w$$

Supposing that the spring that joins two masses of the membrane are equal, then $-k_n=-k_s=-k_e=-k_w$, gives:

$$F_x = -k(\Delta x_n + \Delta x_s + \Delta x_e + \Delta x_w) \quad (35)$$

Analogously:

$$F_y = -k(\Delta y_n + \Delta y_s + \Delta y_e + \Delta y_w) \quad (36)$$
$$F_z = -k(\Delta z_n + \Delta z_s + \Delta z_e + \Delta z_w) \quad (37)$$

These being the forces acting on m_c, define its acceleration at the time that m_c is oscillating. As the strength is known, by using Newton's second law $\vec{F} = m\vec{a}$, and the equation for a uniformly accelerated motion, the final speed in m_c is given in terms of its initial speed and the acceleration experienced by the particle at time t, therefore:

$$\begin{aligned}\vec{v_f} &= \vec{v_i} + \vec{a}t \\ &= \vec{v_i} + \frac{\vec{F}}{m}t\end{aligned}$$

Separating the speed vector into its rectangular components and returning force values of equations (35), (36) and (37) finds the speed components for mc these being:

$$v_{fx} = v_{ix} + \frac{F_x}{m}t \quad (38)$$

$$v_{fx} = v_{ix} + \frac{F_x}{m}t \quad (39)$$

$$v_{fz} = v_{iz} + \frac{F_z}{m}t \quad (40)$$

This gives the speed of m_c after a t time, which allows the calculation of the new position of the particle at the same instant of time; using the equation of uniformly accelerated motion, obtains:

$$x_{fi} = x_i + v_{ix}t + \frac{1}{2}\frac{F_x}{m}t^2 \quad (41)$$

$$y_{fi} = y_i + v_{iy}t + \frac{1}{2}\frac{F_y}{m}t^2 \quad (42)$$

$$z_{fi} = z_i + v_{iz}t + \frac{1}{2}\frac{F_z}{m}t^2 \quad (43)$$

These speed and position equations are used in the definition of evolution function for the proposed CA.

IV. Model Of The Vibrant Membrane Using A 2-Dimensional Ca

Based on the analysis developed in Section III-A, The CA model is defined for a length $l \times l$, vibrant membrane system fixed at the edges as a 4-tuple $CA = (L, S, V, \Phi)$ where each m_c in L cells is defined by its mass, initial position and speed when the membrane is at rest. being:

L: 2-dimensional regular lattice

$$S: = \left\{ \begin{array}{c} \text{fixed cell} \\ \vec{P}^t_{m_{i,j}} : \text{position vector at time } t \\ \vec{V}^t_{m_{i,j}} : \text{speed at time } t. \end{array} \right\} \forall m_{i,j} \in \mathbb{L}^2$$

$$V: \quad V = \{(m_n, m_s, m_c, m_e, m_w)\}$$

$$\Phi: \quad \mathbb{R}^3 \to \mathbb{R}^3,$$

$$\Phi: \left\{ \begin{array}{l} a)\vec{P}^{t+1}_{m_{cf}} = \vec{P}^t_{m_{ci}} + \vec{V}i^t_{m_{ci}}t + \frac{\sum_{v=1}^{4}\vec{F}^t_{v_c}}{2m}t^2 \\ \\ b)\vec{V}f^{t+1}_{m_c} = \vec{V}i^t_{m_{cf}} + \frac{\sum_{v=1}^{4}\vec{F}^t_{v_c}}{m}t \end{array} \right\}$$

where : $\sum_{v=1}^{4}\vec{F}^t_{v_c}$ it is the force over m_c because of its neighbors at time t.

$\vec{P}^{t+1}_{m_{cf}}$, is the final position of the cell into the space, and

$\vec{V}f^{t+1}_{m_c}$, is the final speed at time $t+1$.

The evolution function Φ, is composed of two fundamental rules, both being applied simultaneously to all the cells that form the lattice.

The rule a) defines the position of the cell at time $t+1$; taking the speed at time t, this position is updated, with the new start position $t+2$ and so on. Similarly, for rule b) the final speed at time $t+1$ is updated, being the initial speed at time $t+2$.

An important point in the model definition is the constant restitution for the springs employed in equation (34), unlike other models [9] in which the constant restitution was relatively simple to calculate. In this model we face a problem in which the arrangement of masses and springs do not have a serial pattern but present an array of grids. For this value we proceed similarly to the method for

reducing power grids, for a constant restitution generally defined as start arrangements that can reduce type "Delta" to "Star" and vice-versa, based on Fig. 6.

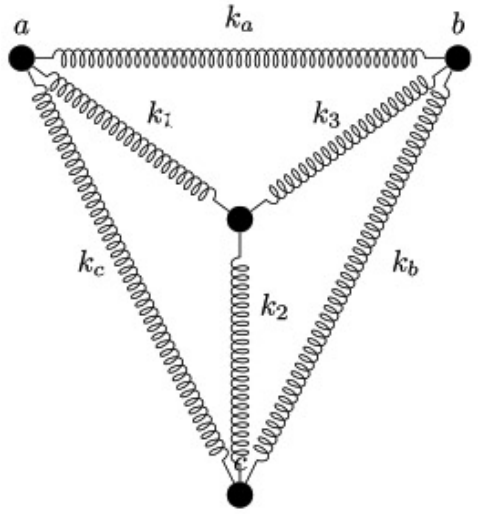


Fig. 6 Delta-Star spring array.

Def 4: (Delta to Star transformation) Defined as the transformation of spring array in Delta pattern to a Star pattern such as:

$$k_1 = \frac{k_a k_c}{k_a + k_b + k_c} \quad (44)$$

$$k_2 = \frac{k_b k_c}{k_a + k_b + k_c} \quad (45)$$

$$k_3 = \frac{k_a k_b}{k_a + k_b + k_c} \quad (46)$$

Where k_1, k_2, k_3 are the new restitution constants that will make the springs perform the transformation and k_a, k_b, k_c are the restitutions constants that serve as a basis for change.

Def 5: (Star to Delta transformation) Defined as the transformation of spring array in Star pattern to a Delta pattern such as:

$$k_a = \frac{k_1 k_2 + k_1 k_3 + k_2 k_3}{k_2} \quad (47)$$

$$k_b = \frac{k_1 k_2 + k_1 k_3 + k_2 k_3}{k_1} \quad (48)$$

$$k_c = \frac{k_1 k_2 + k_1 k_3 + k_2 k_3}{k_3} \quad (49)$$

Where k_a, k_b, k_c are the new restitution constants that will make the springs perform the transformation and k_1, k_2, k_3 are the restitutions constants that serve as a basis for change.

It is necessary to know how to treat the springs found connected in series or in parallel patterns. For this we define the following:

Def 6: (Spring in series pattern) Given a group of joined springs in series pattern, we define the total restitution constant of array as:

$$\frac{1}{k_t} = \frac{1}{k_1} + \frac{1}{k_2} + \cdots + \frac{1}{k_n} \quad (50)$$

Where k_t is the total restitution constant array and k_1, k_2,… k_n are restitution constants of each one of the springs that form the array.

Def 7: (Springs in parallel pattern): Given a group of join springs in a parallel pattern, we define the total restitution constant of array as the sum of the restitution constants of all springs that form the array.

The next section shows how to use these definitions.

A. Computing the k constant

Proceed as follows to calculate the value of k restitution constant of springs that joins the corresponding masses.

The process begins by assuming that the membrane has a constant elasticity with a value of kt. If started with a membrane represented by the proposed CA, consisting of 2×2 nodes, following the reduction of springs connected in series and in parallel obtains:

$k=k_t$

Following the process for a CA consisting of 3×3 nodes, gives:

$$k = \frac{3k_t}{2}$$

Continuing this, the relationship take the form:

$$k = \frac{n}{2}k_t$$

Where k is the restitution constant, k_t is the membrane elasticity constant and n the number of nodes to $n\times n$ CA.

V. Simulation

To develop the simulation with the proposed CA, it uses a membrane with the following characteristics:

- Length: $10\ cm\ \times\ 10\ cm$ and stretched *20%* of its length.
- Density: *0.1*
- Tension: *20 N/m*
- Nodes: 50×50 nodes for the CA.

The initial conditions for the CA, are the same as those described for the PDE from which the model was derived.

It shows the displacement obtained by the PDE of vibrant membrane, and is oscillated for *1s* and took 1×10^4 samples. The offset was obtained from the CA from the same cell during the same time and with the same number of samples.

Fig. 7 and Fig. 8 show the overlapping of the graphs obtained by the CA oscillations (continuous line) and the PDE (in dotted line) as well as a zoom-in to the same overlay.

Qualitatively, there is a phase matching the two models for the same cell.

Obtaining the frequency spectra for both signals and plotting them, it can be observed that there is a congruence between the frequency spectra from the CA and EDP (see Fig. 9). Fig. 10, shows that the fundamental frequency for both models is around 85Hz.

A. Error

To calculate the mean squared error between measures generated by the CA and the reference values provided by the PDE, uses the formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Vac_i - Vr_i)^2$$

Where Vac_i is the i-th estimated value of the CA and Vr_i is the reference value taken from the PDE. Performing the calculation obtains $MSE=4.4189\times10\text{-}11$.

VI. Conclusions

This paper shows the process of scaling from the vibrating string CA model to a two-dimensional CA in order to model membranes. It is feasible to extend the model to three dimensions; the difficulty lies in obtaining the relationship between the elasticity of the material models and the restitution constant of the internal CA's springs.

The CA membrane model is released from the initial conditions and it is not necessary to redefine it due to any change, so it is possible to define the

initial conditions and simulate the system behavior immediately. Difference to the PDE, it is sensitive to initial conditions for its solution.

The model expansion from one to two dimensions, brings an increased computational model complexity, from linear in the one-dimensional case to $O(n^2)$ complexity because the CA space is now a $n \times n$ matrix. In this sense the parallelization model is justified. If we define m as the number of threads that help the evolution of CA per unit time, then the CA computational complexity will be $O(n^2/m)$. If we make m large enough such that $m \rightarrow n$, then the complexity is reduced to $O(n)$ which justifies the model parallelization.

Acknowledgements

The authors wish to thank CONACYT, of the COFAA-IPN (project number 20143959) and to EDI-IPN for the support provided for this work.

References

[1] R. Espericueta, "Cellular Automata a Dynamics Explorations in Parallel Processing", USA: Bakersfield College, 1997.

[2] S. Wolfram, «Statical mechanics of cellular automata,» Reviews of Modern Physics, nº 55, pp. 601-644, 1983.

[3] Y. Bar-Yam, "Dynamics of Complex Systems", USA: Addison-Wesley, 1997.

[4] M. Mitchell, «Computation in cellular automata: A selected review,» de Nonstandard Computation, weinheim, VCH Verlagsgesellschaft, 1998, p. 95–140.

[5] B. Chopard, "Cellular Automata Modeling of Physical Systems", USA: Cambridge University Press, 1998.

[6] L. Kinsler, "Fundamentals of acoustics", USA: Wiley,, 2000.

[7] H. Hsu, "Análisis de Fourier", México: Prentice Hall, 1998.

[8] M. Van-Walstijn y E. Mullan, «Time-domain simulation of rectangular membrane vibrations with 1-d digital waveguides,» pp. 449-454, 2011.

[9] I. Huerta-Trujillo, J. Chimal-Eguía, N. Sánchez-Salas y J. Martinez-Nuño, «Modelo de autómata celular 1-dimensional para una edp hiperbólica,» Research in Computing Science, vol. 58, pp. 407-423, November 2012.

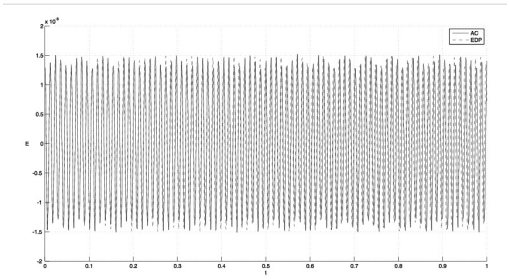


Fig. 7 Overlay graph of simulation performed by the CA and the PDE. The displacement of the cell (n/2,n/2) is presented in z-axis for both cases.

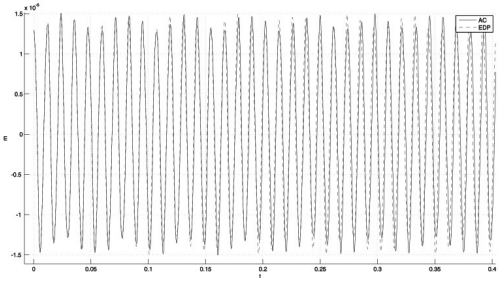


Fig. 8 Zoom-in of the overlay graph of simulation performed by the CA and the PDE. The displacement of the cell (n/2,n/2) is presented in z-axis for both cases.

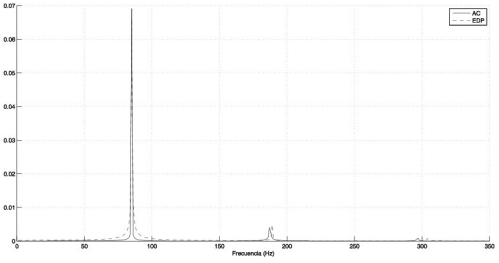


Fig. 9: Overlay graph of the frequency spectrum of the simulation performed by the CA and PDE.

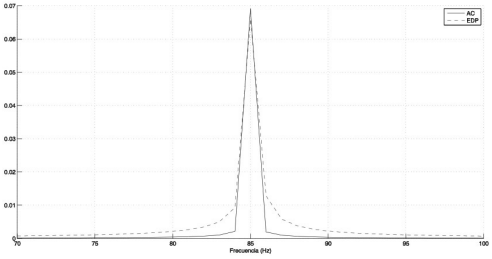
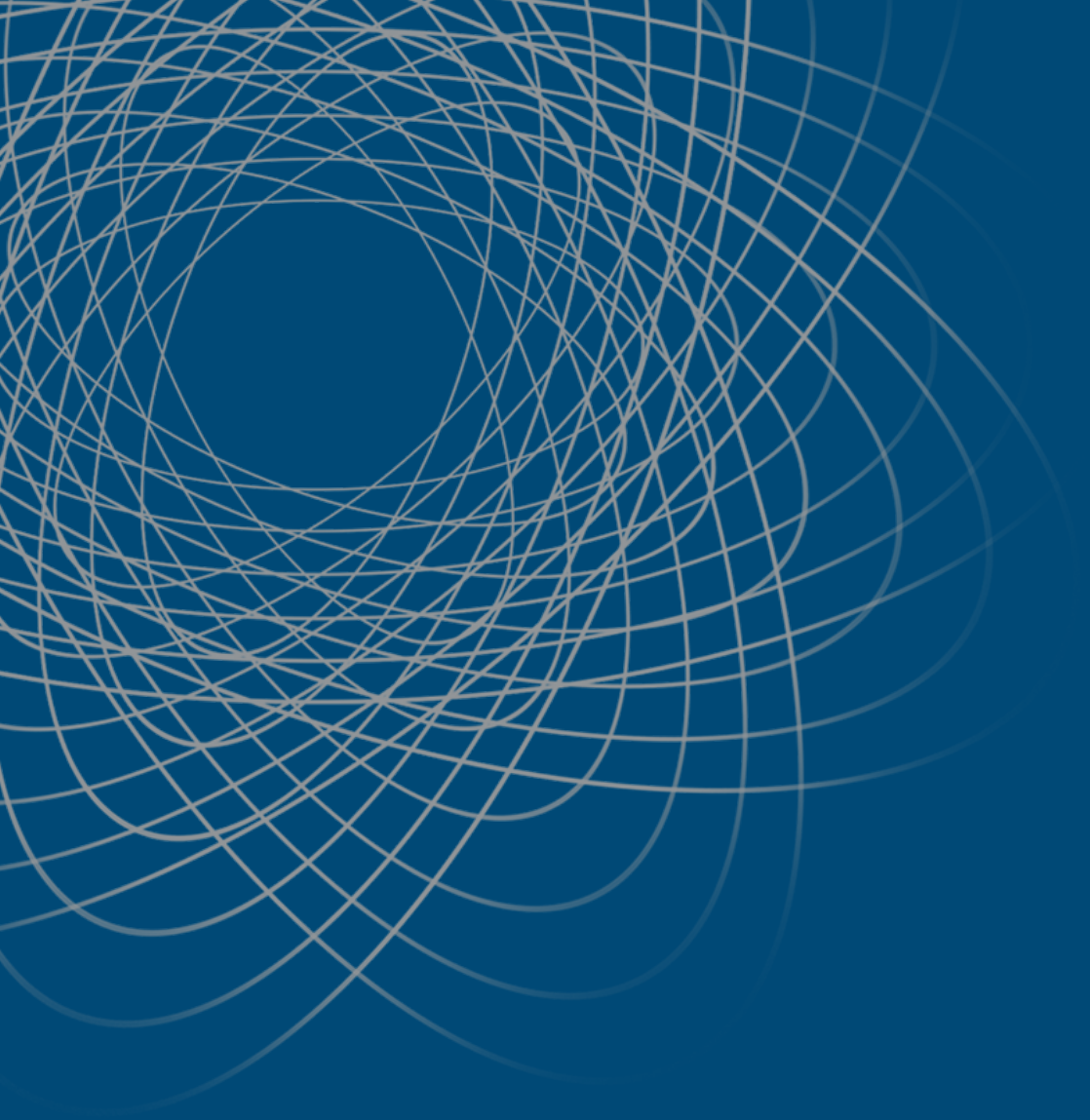


Fig. 10: Zoom-in of the overlay graph of the frequency spectrum for the simulation performed by the CA and PDE, shows the fundamental frequency.

I. Huerta-Trujillo earned a Computational Systems engineering degree with specialization in Systems at 2004 in the ESCOM-IPN, his Master's degree in Computer Science in the CIC-IPN at 2009. At present is PhD student in the Simulating and Modeling Lab at Computing Research Center from IPN.

His interests are complex systems, parallel computing and Cellular Automata for modelling physical systems.

J.C. Chimal-Eguía He obtained the title of Bachelor of Mathematical Physics at the ESFM-IPN in 1994, the master's degree with a major in physics in 1996 at the ESFM-IPN and the degree of Doctor of Science degree in Physics in ESFM-IPN in 2003. Postdoctoral Fellow at the University of Alberta Canada. Currently holds the rank of professor "C" in the Computer Research Center of the IPN in Mexico. He has authored more than 50 articles in journals and international conferences. His interests are the modeling of complex systems and nonlinear dynamics. Dr. Chimal-Eguia is fellow of COFAA, EDI and member of the National Research System (SNI) level I of Mexico.



SCHEDULING

Analysis of Deterministic Timetabling Strategies for the Generation of University Timetables

[1]Marco A. Peña-Luna, [1,2]Adán Hirales Carbajal, [2]Andrei Tchernykh, [1]Miguel Salinas Yañez,
[1]Center for Applied Engineering, CETYS, Tijuana, B.C. 22210, Mexico
{marco.pena, adan.hirales, miguel.salinas}@cetys.mx
[2]Computer Science Department, CICESE Research Center, Ensenada, B.C. 22860, Mexico
{chernykh, ahiralesc}@cicese.mx

Abstract

We present an experimental study of deterministic timetabling strategies for the generation of university timetables. We distinguish three prioritization criteria, which differ by the type and amount of information required to select the courses to schedule. We analyze the performance of the time tabling strategy and prioritization criteria by using a workload with 126 courses, 65 professors and 12 class rooms from a real scenario. To analyze the performance, we conducted a joint analysis of three metrics. We evaluate the quality of the obtained solutions in terms of their degradation and performance profiles.

Keywords. *University timetabling, Scheduling.*

Análisis de estrategias determinísticas de calendarización, para la generación de horarios universitarios

Resumen.

Presentamos un estudio experimental de una estrategia de calendarización determinística para la generación de horarios universitarios. Distinguimos tres criterios de priorización que se diferencian por el tipo y cantidad de información que requieren para seleccionar materias a calendarizar. Analizamos el desempeño de la estrategia de calendarización y de los criterios de priorización al aplicar una carga de trabajo con 126 cursos, 65 profesores y 12 salones correspondientes a un programa educativo universitario real. Con el fin de analizar su desempeño realizamos un análisis conjunto de tres métricas. Evaluamos la calidad de las soluciones obtenidas en términos de su degradación y perfil de desempeño.

Palabras clave: *Calendarios universitarios, calendarizacion.*

1 Introduction

This work studies the university timetabling problem. It addresses a deterministic version of the problem, since knowledge needed to produce a solution is known. Only valid solutions are considered. Solutions that violate hard constraints are discarded. The quality of solutions is evaluated under the perspective/objectives of two actors: the university as the owner of the infrastructure (classrooms), and the professors. We conduct a joint analysis of metrics, giving equal importance to the interest of both actors.

We propose a heuristic for university timetabling, namely Deterministic Time Tabling Strategy (DTTS). DTTS systematically applies a course reservation mechanism called Resource Reservation Strategy (RRS). RRS creates a set of reservations, if the hard constraints can be satisfied for a given course.

We evaluate DTTS performance in terms of its capacity to schedule a set of 126 courses and 65 professors over 12 resources corresponding to a university department. Solutions are evaluated in terms of three metrics: scheduled courses rate, resource utilization and professor's schedule dispersion. DTTS robustness is analyzed by evaluating its capacity to schedule the workload under different resource capacity conditions.

2 Previous work

The timetabling problem consists of the allocation of a sequence of jobs on a set of resources, while satisfying a set of constraints [21]. In its more generic form, it implies the assignment of courses and professors to classrooms in specific time periods, so that no single professor is assigned to more than one classroom at the same moment, and there is no classroom with more than one course at the same time instance [20]. In summary, the objective of timetabling is to find the maximum set of time periods that satisfy space and time restrictions given by professors, resources, and students.

In the context of university course timetabling, the following elements are disti nguished: academic program, a composition of courses within the same field of knowledge; the course with a series of lectures associated to it. If it is possible, courses are allocated to slots or time windows, and professor, a human resource associated to a university department. The professor teaches one or more courses; student belongs to a university department and can be a part of a group, classroom or physical resource, where lectures take place.

In [18, 7, 21], two types of constraints are distinguished: soft and hard. A solution must meet all hard constraints, but it might omit soft constraints. A solution is considered valid, if all hard constraints are satisfied [7]. It is considered feasible, if it is valid and all courses are scheduled [17]. In addition, it is termed high quality solution, if it satisfies soft constraints [21].

Schaerf [19] distinguished three variants of the timetabling problem: School timetabling (STT), Course Timetabling, and Examination timetabling (ETT). STT deals with weekly scheduling of classes. It considers two hard constraints: a professor cannot conduct simultaneous lectures and concurrent lectures cannot occur in groups. Course Timetabling, deals with weekly scheduling of university courses, it includes STT constraints and aims to minimize overlapping of courses shared by common sets of students. Examination timetabling refers to the scheduling of university departmental exams. ETT aims to avoid overlapping of examination days. Its objective is to distribute exam application dates as much as possible within the exam period.

The timetabling problem has been treated as a deterministic problem, because all characteristics of the problem are known. That is, the properties of courses, professors and resources are known. The timetabling problem, even for simple formulations, is NP-Complete [13, 12, 6, 19, 21]. A solution to the problem can be verified in polynomial time. However, there is no known efficient way to find solutions in polynomial time, in terms of

the size of the problem, unless NP equals to P. To our knowledge, there is no approximation algorithm capable of reducing the solution space. In the literature, the timetabling problem has been addressed via: sequential methods, clustering heuristics, constraint programming, and heuristics, such as genetic algorithms, Tabu search, simulated annealing, and constraint satisfaction [5, 6].

Gans [11] proposed a combinatorial heuristic used to schedule 21 school programs. Foundsolutions satisfied all hard constraints. Glover y McMillan [14] utilized administrative science and AI to schedule a workload with more than 100 employees. Aubin and Ferland [4] proposed the generation of a master timetable by grouping independent timetables. They proposed a heuristic that penalizes overlapping and excessive use of resources. Hertz [15] and Costa [10] used Tabu search to reduce the number of conflicts on shared resources. Colorni [8] evaluated the quality of solutions produced by a genetic algorithm with those of different versions of simulated annealing and Tabu search strategies. Predominant characteristic of previous works are the use of AI techniques and single criteria evaluation of solutions.

Constraint satisfaction approaches have also been applied [16]. They are characterized by the type of constraints (hard and soft), domain (discrete and continuous); and the amount of constrains to satisfy (binary, unitary, multiples and n-ary). CSP are studied using operation research and AI technics, such as: search trees, look back algorithms (back-jumping, conflict-directed back-jumping), look-ahead algorithms, and stochastic methods (hill climbing, Tabu search, simulated annealing, genetic algorithms, among others).

3 Problem definition

University timetabling problem is formulated as follows.

There exists a planning horizon [0,T] divided in periods [t, t + 1], with t = 0,1,⋯,T-1. Period duration is constant, one hour in our study. There exist m resources, each has its own availability and planning horizon, both of size T. We assume full availability of resources.Therefore, no reservations exist prior the scheduling process. Resource capacity is not considered.

n courses j = 1,⋯, n are scheduled within the planning horizon. A course is described by the tuple (ps_j, pl_j, pc_j, pt_j, hcc_j, id_j) with duration of lecture $ps_j \geq 1$; lab hours $pl_j \geq 0$, workshop hours $pt_j \geq 0$, clinical hours $pc_j \geq 0$; release time $r_j = 0$, and course identifier id_j Initially all courses are released at time 0. During the scheduling process their release times are set according to availability of the professor and the resources available.

The number of employees $D_j(t)$ who teaches course t_j in the period [t, t + 1] is one. Given a set E of employees, with |E| = k , each employe e ∈E is qualified to teach a subset Q_e of courses. For example, Q_e {1,2} implies that employee is qualified to teach only courses 1 and 2.

The availability pattern of employee e is defined as following.

A binary array $\langle w_e(t)\rangle_{t=0}^{T-1}$. $w_e(t) = 1$ife is available in the time slot [t, t + 1], otherwise $W_e(t) = 0$, which means that there is no availability.

When a course is allocated into Q_e at a time slot [t, t + 1], the value of $W_e(t)$is changed from 1 to 0.

A planning horizon is a composition of days from Monday to Saturday. Other days combinations are possible. We consider a daily window of 16 hours (1:00am to 11:00pm), hence T = 96 hours.

A working pattern in a time window is a binary array $\alpha = (\pi\ (j,t))$, such as $\pi(j,t) = 1$ if in the time slot $[t,\ t + 1]$ course t_j is imparted, otherwise its value is cero. After the scheduling process, professors and each resource have a working pattern.

The following hard constraints are considered:

1. There is no preference for employees. Employee selection depends on the course list to schedule.

2. Scheduling is done for one university department programs. Resources are not shared among departments.

3. Hours of Consecutive Class HCC_j of a course t_j are finite. Furthermore, we limit the length HCC_j of to be no greater than 16 hours.

4. It is not allowed to have two or more HCC_j in the same day. HCC_j models class hours, lab hours, workshop hours, or clinic hours.

5. Summation of all HCC_j corresponding to course must be equal to the sum of the hours specified in its tuple.

6. An employee $e \in E$ cannot teach courses t_j and t_i, $t_j, \neq t_i$, at the same time period $[t, t + 1]$.

7. Similarly, courses t_j and t_i, $t_j \neq t_i$, cannot be taught during the same time period $[t, t + 1]$ in the same resource $r \in \{1, 2, ..., m\}$.

A solution that satisfies these constraints is valid, even, if it does not schedule all courses in the list. The concept of the group is not considered.

4 Scheduling heuristics

Requirements of a course are satisfied when its vector of durations (ps_j, pl_j, pc_j, pt_j) is assigned to one or more resources. During the scheduling process durations are fragmented into blocks of size HCC_j and scheduled as early as possible in the employee availability window. For each block, a reservation is made in the resource that satisfies constraints 1 to 7. This scheduling process is denominated Resource Reservation Strategy (RRS).RRS selects a resource among the m resources available, by means of a selection criterion. For example: the resource with less reservations, randomly, with less of more capacity, etc.

Given the course t_j, RRS produces a set of reservations, if and only if all of its durationsare scheduled. Otherwise reservations are cancelled.

Pseudocode 1: RSS

1. RRS reads the identifier e of the employee asociated to t_j and makes the release time r_j of t_j equal to the first available time of employee e.
2. Given course t_j, RRS iterates while durations or some constraints of t_j are not satisfied.
3. RRS selects the duration d_j from duration vector $(ps_j, pl_j, pc_j, pt_j)$of t_j. The vector is visited sequentially starting with ps_j. Subsequent duration is not read until the number of hours from the immediate predecessor is cero.
4. If d_j is cero, RRS removes the tag from t_j, return to 2.

5. if duration d_j is bigger than cero. RRS checks if course t_j was previously assigned to a resource. A scheduled task is tagged with the identifier of resource r if it was previously assigned to that resource. Tag is removed when duration class d_j is cero.
6. RSS creates a search area $A = [r_j, r_j + HCC_j]$
7. if$A$ do not belong to the horizon of employee e, $A \notin H_e$, RSS increases $r_j = r_j + 1$, return to 6.
8. if$A$ do belongs to e horizon, RRS checks if there exist availability in time interval A in each resource u. Let $s_u = \{(r_j, r_j + HCC_j), null\}$ and let $disp$ a no null availability vector $disp = \{u | u = 1, \cdots, mys_u \neq nulo\}$. Note that $0 \leq |disp| \leq m$.
9. if $|disp| > 0$, that is, there exists one or more resources available in interval $[r_j, r_j + HCC_j]$
    a. if r is not null and $r \in disp$, RRS selects resource r
    b. if$r$ is not null and $r \notin disp$, RRS selects a resource r applying a selection criteria. For example, resource with minor number of reservations; randomly; resource with smaller or larger capacity, etc.
    c. if$r$ is null, RRS selects a resource r applying a selection criteria (same as 9.b.)
10. RRS makes reservation permanent in r; reduce duration of d_j to $d_j - HCC_j$; eliminates profesor availability in time interval $[r_j, r_j + HCC_j]$; tags t_j with r; and modifies release time r_j of t_j to the next employee e available time (release time r_j should correspond to the next day in the employee availability); set $w_e(t) = 0$, $t = r_j, \cdots, r_j + HCC_j$. Return to 2.

We assume an atomic reservation process. Whenever possible, course t_j reservations are made at the same resource, but at different days. Initially there is no knowledge about resourcesfor reservations.

Once the first HCC_j block is allocated the resulting reservation is tagged with the id of the selected resource. The tag is used to keep memory of previous scheduling decisions. Subsequent reservations, for the same course, are allocated if possible to the tagged resource. The tag is removed if a reservation can no longer be done at the tagged resource. Pseudocode 1 shows the RSS heuristic.

The scheduling heuristic consists of two phases: reservation and confirmation. Given a course, the reservation phase tries to create a set of reservations. Reservations are regarded as tasks in subsequent paragraphs. In the confirmation phase, tasks are sent to their respective resources, thus making reservations permanent.

Given n courses j = 1,⋯, n, the DTTS systematically takes one course t_j, applies RRS and produces a set of reservations or none. DTTS keeps two logs: courses with reservations (TcR) and courses that could not be scheduled (Task with Failures, TcF). RRS generated reservations are registered in TcR, while courses with no reservations are stored in TcF. Once all courses are processed by RRS, DTTS sends the tasks in TcR to their corresponding resources. The local scheduler at each resource receives the task and makes the reservation permanent.

DTTS receives a list of courses to schedule. Each entry of the list contains the following fields: class hours ps_j, lab hours pl_j, workshop hours pt_j , clinic hours pc_j, release time r_j, identifier id_j of course t_j and identifier e ∈ E of the employee qualified to teach t_j. DTTS produces a schedule for each resource. Each schedule contains the following fields: release time r_j, completion time c_j , identifier of the course , identifier of employee e.

Pseudo code 1 describes the execution of the RSS. Alternate flows try to find time intervals compatible between course requirements, employee and resource availability window. When course requirements are not satisfied both availability windows are restored to previous state.

Employee and resource horizons are updated each time a task is assigned to them. Reservations made over one or more horizons are cancelled when, given the requirements of a course, compatible windows between employee and resources do not exist.

Pseudo code 2 describes DTTS execution flow, which uses RRS and generates TcF and TcR logs.

Pseudocode 2. DTTS

```
1. For each course t_j in the workload
2. res = call RSS with t_j
3. if the set res (reservations) is empty then
4.   store t_j in log TcF
5. otherwise
6.   store reservations res in log TcR
7. end
```

5 Methodology

5.1 Ordered lists

Previous to the scheduling process, all courses are inserted into a list. The course in the head of the list is scheduled first. One course is scheduled at a time. Three different orderings are studied: Random (A), there is no preference in the selection of courses; Part Time – Full Time (AP), courses taught by part time professors are scheduled first; Proportion of Hours Required from the Professor (PHRP), courses corresponding to professors with higher proportion of required hours are scheduled first. We evaluate DTTS and RSS analyzing its performance on the three list orderings. We evaluate the algorithm robustness analyzing its performance varying the available number of resources: 6, 8, 10, …, 26 and 28 resources.

5.2 Experimental design

Table 1 summarizes different aspects of the experimental design.The number of faculty and courses to teach is 65 and 126 respectively. 18 full time professors are considered, each with an availability window of 8 hours with recess from 1:00 PM to 3:00 PM. The rest of the professors are part time. Their availability varies. A total of 541 hours should be scheduled. The workload corresponds to real requirements of an Engineering School, Mexicali, CETYS University, January-June 2013 semester.

Table 1. Experimental design

Factors	Levels
Workload	Course Priority list. Three orderings: Random (A), Part Time – Full Time (AP); Proportion of Hours Required from the Professor (PHRP)
# of resources	6, 8, 10, …, 28
Dependent Variables	Utilization, time dispersion and y scheduled courses rate
Replication	400
Design	Factorial (2 factors)
ProblemModel	Deterministic

5.3 Metrics

A good solution aims to satisfy expectations from both the owner of the resources and form faculty. University administrators aim to maximize resource utilization and scheduled courses rate, while faculty prefers to minimize dispersion in their course schedule.

We distinguish three metrics: utilization $U = \frac{1}{HT*m}\sum_{g(t_j)=\{m_k\}} p_j$, where $p_j = ps_j + pl_j + pc_j + pt_j$

represents the duration of course t_j, HT is the total number of hours to schedule, and $g(t_j)$ is a function that assigns course t_j to a set of resources $\{m_k\}$; scheduled course rate PCC =|{scheduled courses} / |{courses}|; and normalized average time dispersion.

Let t_i y t_j be two tasks corresponding to one or more courses in Q_e such that i α jwhere i ≠ j, that is, is an immediate predecessor of in the same working day. Average time dispersion of employee e is $d_e = \frac{\sum_{(i,j)} s_j - c_i}{n_e}$, where s_j is the start time of t_j; c_i is the end time of t_i; and n_e is the number of tasks in the working window of employee . We normalize the employee average time dispersion $dn_e = \frac{\bar{d}_e}{\max_{e \in E}\{\bar{d}_e\}}$, such that when dn_e tends to one average dispersion in employee *e* working window is big. It is desirable that dn_e tends to cero. Finally, normalized average time dispersion is $dn = \frac{\sum_{e \in E} \overline{dn}_e}{|E|}$.

For professor , we define PHRP as the rate of the number of hours required to teach all courses in Q_e over the number of hours in employee *e* availability window.

The workload is composed of a sample of 400 lists of the 126! possiblelists. A degree of variability of 50%, 95% confidence level, and +/- 5% precision level is used to determine the sample size. A list L with AP ordering is generated by creating two sub-lists: L_a for courses taught by part time professors and L_p for courses taught by full time professors.

Elements in lists y are permutated using the algorithm that randomizes in place [9], then L_a y L_p are concatenated in . A list with PHRP ordering is generated using the set of lists $\{L_1, L_2, \ldots, L_k\}$. List L_i contains courses of professors with same PHRP. Lists are sorted in descending order according to their PHRP index and are concatenated into L.

6 Results

6.1 Increasing number of resources

Figure1 shows the average of scheduled course rate. DTTS performance using PHRP is marginally superior to AP and Random. 60% of courses are programed with only six resources, independently of the list ordering. Average of course rate using PHRP approximates 100% by increasing the number of available resources. With 14 resources AP and Random rate approximates 92%, while PHRP attains 100%.

Results presented in Figure 1 consider valid solutions, that is, those which satisfy all hard constraints. However, AP and Random do not find feasible solutions with a reduced number of resources. Result presented in Table 2 show that PHRP, AP, and Random produce feasible solutions with 14, 20 and 24 resources respectively. From the 400 course lists ordered under PHRP, 194 produce feasible solutions. Additionally, PHRP finds the higher number of feasible solutions.

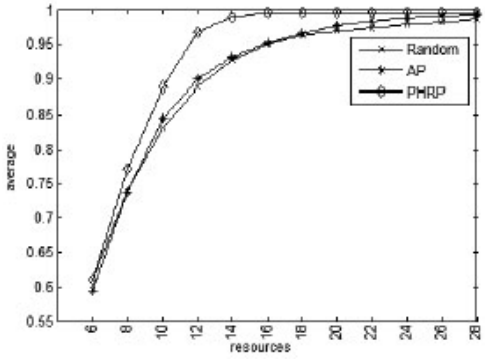


Figure 1. Average scheduled course rate

Table 2. Number of generated solutions

	Feasiblesolutions		
Resources	Random	AP	PHRP
6	0	0	0
8	0	0	0
10	0	0	0
12	0	0	0
14	0	0	71
16	0	0	193
18	0	0	194
20	0	2	194
22	0	16	194
24	3	51	194
26	6	89	194
28	17	131	194

DTTS systematically takes an availability pattern and tries to find a subset of time slots to assign the professor courses. PHRP selects availability patterns where a higher number of courses can be scheduled. We denominate patterns with this characteristic as congested. PHRP produces solutions with higher scheduled course rate by creating opportunities for a higher number of compatible congested patterns. We say that an availability pattern is compatible if it does not overlap with other patterns.

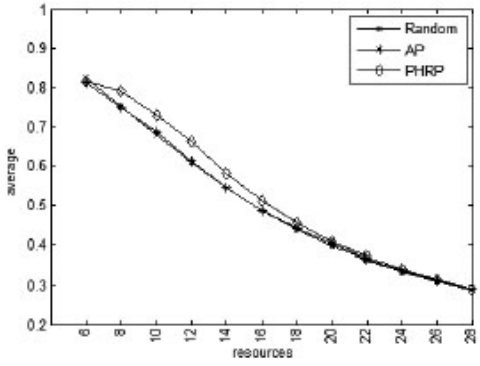


Figure 2. Average resource utilization

Independent of the list ordering, a maximum utilization of 80% is obtained. It is achieved, when 6 resources are used (See Figure 2). As expected, utilization decreases when the number of available resources increases. PHRP utilization is higher than the one obtained using AP and Random in the interval between 8 and 18 resources. This happens because PHRP finds a higher number of compatible congested windows. With 12 and 14 resources DTTS+PHRP achieves utilization between 66% and 58%, which is comparable with results presented in **[Error! Reference source not found., Error! Reference source not found., Error! Reference source not found.]**. Therefore, we can conclude that timetables generated by DTTS produce reasonable resource utilization.

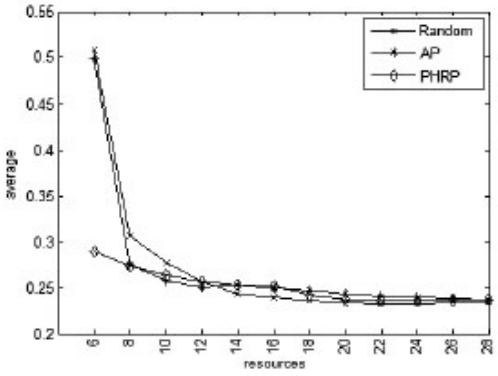


Figure 3. Normalized average professor time dispersion

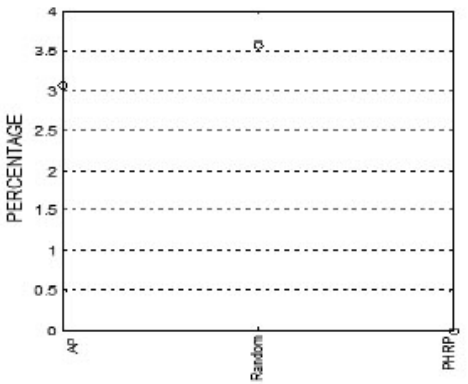


Figure 4. Degradation of Avg. scheduled course rate

Figure 3 shows professor time dispersion. When 6 resources are used, AP and Random produce solutions with normalized average dispersion of approximately 0.5, while PHRP have a dispersion of 0.3. Solutions generated by PHRP have 20% less dispersion than the other two orderings. With 12 resourcesor more, dispersiontends to 0.25.

6.2 Degradation

In this section, we evaluate DTTS degradation using three input list orderings, 126 courses, 65 professors and 12 classrooms (resources). We select 12 resources from real scenario of 12 resources in the semester timetable.

Degradation is defined as the rate between a generated solution and the best generated solution for one metric of interest:

$= \left(\frac{\text{metric value}}{\text{best found metric value}} - 1\right) * 100$. A degradation of cero is desirable.

From Figures 4, 5, and 6, it is not possible to measure, with a reasonable precision, how better one ordering with respect to another one. However, figures show that DTTS produces better solutions using PHRP for all metrics.

Solutions generated by AP and Random are 3% and 3.5% worse than those found by PHRP in terms of scheduled course rate. AP and Random utilization are approximately 3.6% and 4% worse than PHRP. Time dispersion in solutions generated by AP and Random are approximately 7% worse than those generated by PHRP. In summary, AP and Random obtain an average degradation of 4.5 and 5, while PHRP obtains the best average degradation, see Figure7.

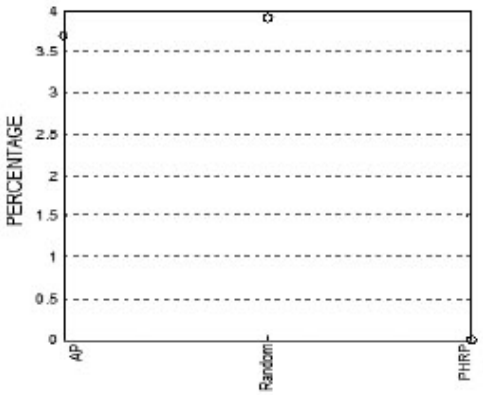


Figure 5. Degradation of Avg. resource utilization

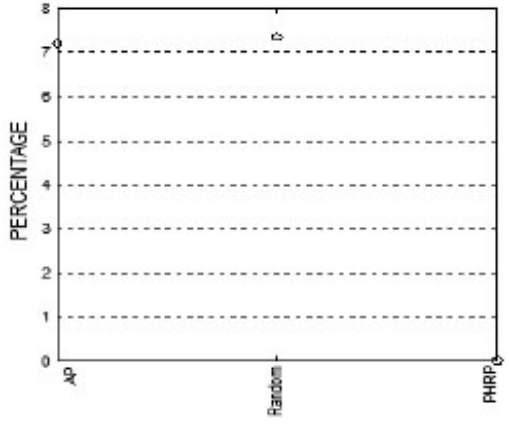


Figure 6. Degradation of Avg. professor time dispersion

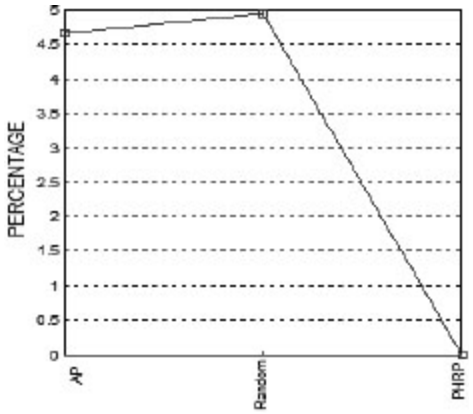


Figure 7. Avg. degradation of the three metrics

6.3 Performance profiles

We analyze DTTS performance profile with Random, AP and PHRP using 12 resources. We found that 50%, 30% and less than 10% of the solutions generated by PHRP, AP and Random are equal to the best generated solution in terms of the scheduled course rate (see Figure 8). However, within a factor of 0.02 from the best generated solution, the percentage of AP and Random generated solutions quickly increase to 80% and 70% respectively. It is important to remark that this happens because of the capacity of the strategies to find valid solutions, although not necessarily feasible solutions.

Generated solutions are marginally 0.02 times worse or equal to the best generated solution. However, it does not imply that solutions are feasible.

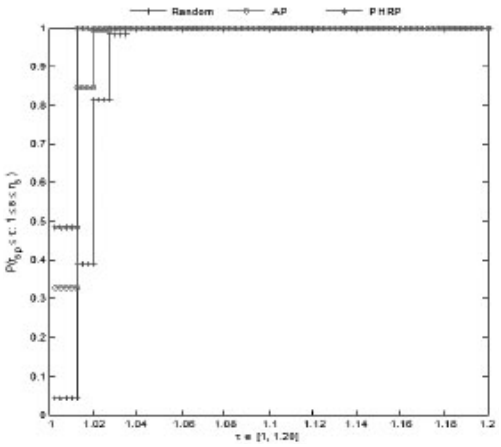


Figure 8. Schedule course rate

We found that, in terms of utilization, approximately 50%, 30% and less than 10% of PHRP, AP and Random, respectively, generate solutions equal to the best generated solution (Figure 9). Within a factor of 0.02, AP and Random increase to 85% and 45% respectively. PHRP had the best dispersion, see degradation analysis. However, the amount of solutions with equal professor time dispersion is near cero (Figure 10). Approximately 40% of Random generated orderings are equal to or up to 1.05 times worse than the best generated solution.

Almost 100% of the solutions generated by PHRP are within a factor of 1.06 from the best generated solution.

From the previous analysis, we conclude that the advantage of PHRP over AP and Random is marginal, when considering valid solutions.

In general AP and Random can produce numerous solutions close to the best generated solution within a factor of 0.03. These results cannot be considered conclusive, if problem restrictions are modified or amounts of resources vary, then response variables can be affected. This case will be studied in the future work.

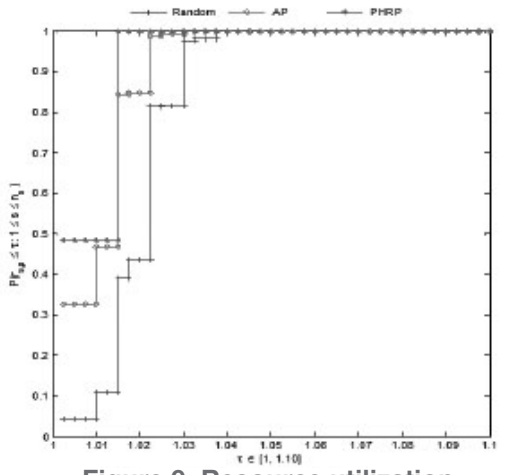


Figure 9. Resource utilization

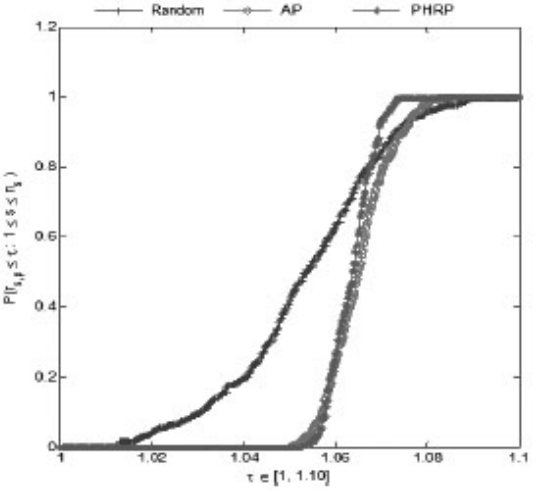


Figure 10. Dispersion

7 Conclusions

University timetabling requires an effective allocation of courses to resources, given a set of hard constraints, different professor availabilities, and frequently a reduced number of resources. It is a combinatory optimization problem and it is classified as NP-Complete, therefore, there is no knowledge of efficient algorithms to find an optimal solution.

In this work, we propose a deterministic heuristic called Deterministic Timetabling Strategy (DTTS) to generate university timetables. We analyze its performance under three different input list ordering conditions (Random, AP and PHRP) to determine which input produce best solutions. We evaluate the quality of the solutions in terms of three metrics: scheduled course rate, utilization, and professor time dispersion. For the selection of the best strategy, we conduct a joint analysis of the metrics in terms of the mean and degradation. In multi-criteria analysis, we assume equal importance of each metric. To address the possibility of biased results, where a small number of instances of the problem have large deviations that affect the average results, we analyze the performance profile of the strategies.
Our study results in these contributions:

- We developed and analyzed the performance of DTTS using different input criteria
- *We proposed and evaluated a mechanism for resource selection, specifically selection in terms of the Least Number of Reservations (LNR)
- We demonstrated that PHRP + DTTS + MNR strategy finds the largest number of feasible solutions via proof of concept. We conclude that it results beneficial to allocate courses of professors with more academic load relative to their availability first.
- We demonstrated that AP + DTTS + MNR and Random + DTTS + MNR possess the capacity to produce high percentage of valid solutions, although not feasible solutions.

When analyzing DTTS performance using real data, we found that an appropriate distribution of courses is achieved by prioritizing the allocation of courses in terms of the course load of professors. At the same time, we useknowledge

of the number of reservations held by each resource; that is, the state of the resource. This is opposed to the idea of giving higher priority to the allocation of full time professors first.

By varying the amount of resources PHRP + DTTS produces the best results in terms of the three metrics and generates the largest number of feasible solutions (194). Using Random and AP orderings, 2 and 3 feasible solutions are generated when utilizing 20 and 24 resources, respectively. PHRP accomplishes this by finding a larger number of independent congested availability patterns. On average, the solutions generated by AP and Random are approximately 4.5% and 5% worse than those generated by PHRP, as both are able to find a greater number of valid solutions.

When analyzing the performance of DTTS evaluating their performance profile, we found that approximately 50% of the solutions found by PHRP + DTTS are equal to the best solution found in terms of utilization and scheduled courses rate. However, being only a factor of 0.03 from the best found solution, about 80% of the solutions generated by Random and AP are valid. From the above perspective, the advantage of PHRP over AP and Random is marginal when valid solutions are considered.

Regarding the amount of knowledge required by the scheduling strategy, we conclude the following: Since DTTS uses MNR, it is necessary to have knowledge of the total number of resources and their status. Having state information is favored to PHRP that allows to find 194 feasible solutions. Knowledge of the professor course load is also useful, as it allows prioritization for allocating courses giving better results by issuing higher importance to professors with bigger academic load with respect to their availability. Computationally, prioritization of resources (MNR) and professors results in a higher computational cost, which increases the execution time of the heuristics but it allows us to find a larger number of feasible solutions.

7.1 Future work

From this study a number of questions arise. We underline the following observations and unknowns.

- DTTS does not consider the concept of group, in future implementations we will incorporate it.
- During the scheduling process, resources are selected in ascending order of the number of reservations they have, giving higher priority to the resource with fewer reservations. Such a mechanism is intrusive since it requires knowledge of the characteristics of the resources. From this, the following questions arise: What is the impact in the quality of generated solutions, having more or less informations about the resources?
- So far, we can only say that we found valid and feasible solutions, and that a valid solution meets all constraints of the problem. However, what are the differences between valid and feasible solutions? Why a valid solution did not schedule all courses?
- The input list is prioritized in terms of each professor academic load. It is interesting to study the possibility of prioritizing the workload together. For example, all professors teaching courses ini-th semester receive the same priority. During the scheduling process, the heuristic tries to assign professors of such a groupfirst.

8 References

1. University of Oregon classroom space utilization study fall term 2012.
2. University of Victoria classroom utilization report 2004/05, November 2005.
3. University of Wisconsin-Madison administrative excellence classroom space utilization, June 2012.
4. J. Aubin and J. A. Ferland (1989).A large scale timetabling problem. Comput. Oper. Res., 16(1):67–77.
5. Edmund Kieran Burke and SanjaPetrovic (2002). Recent research directions in automated timetabling. European Journal of Operational Research, 140(2):266 – 280.
6. Michael W. Carter and Gilbert Laporte (1998). Recent developments in practical course timetabling. In Practice and Theory of Automated Timetabling II, volume 1408 of Lecture Notes in Computer Science, pages 3–19. Springer Berlin Heidelberg.
7. Sara Ceschia, Luca Di Gaspero, and Andrea Schaerf (2012).Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. Comput.Oper. Res., 39(7):1615–1624.
8. Alberto Colorni, Marco Dorigo, and Vittorio Maniezzo (1998). Metaheuristics for high school

timetabling. Comput.Optim. Appl., 9(3):275–298, March.
9. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (2009). Introduction to Algorithms (3rd ed.). MIT Press and McGraw-Hill.
10. Daniel Costa (1994). A tabu search algorithm for computing an operational timetable. European Journal of Operational Research, 76(1):98–110.
11. Onno B. de Gans (1981).A computer timetabling system for secondary schools in the netherlands. European Journal of OperationalResearch, 7(2):175–182.
12. D. de Werra (1985). An introduction to timetabling. European Journal of Operational Research, 19(2):151–162.
13. S. Even, A. Itai, and A. Shamir (1976).On the complexity of timetable and multicommodity flow problems. SIAM Journal on Computing, 5(4):691–703.
14. Fred Glover and Claude McMillan (1986). The general employee scheduling problem: an integration of ms and ai. Comput.Oper. Res., 13(5):563–573.
15. A Hertz (1991). Tabu search for large scale timetabling problems. European Journal of Operational Research, 54:39–47.
16. Tomáš Müller, Hana Rudová, and Roman Barták (2005).Minimal perturbation problem in course timetabling. In Practice and Theory of Automated Timetabling V, volume 3616 of Lecture Notes in Computer Science, pages 126–146. Springer Berlin Heidelberg.
17. Clemens Nothegger, Alfred Mayer, Andreas M. Chwatal, and Günther R. Raidl (2012).Solving the post enrolment course timetabling problem by ant colony optimization.Annals of Operations Research, 194(1):325–339.
18. Ender Ozcan, Andrew J. Parkes, and AlpayAlkan (2012).The interleaved constructive memetic algorithm and its application to timetabling. Comput.Oper. Res., 39(10):2310–2322.
19. A. Schaerf (1999).A survey of automated timetabling.Artif.Intell. Rev., 13(2):87–127.
20. Kate Smith-Miles and Leo Lopes (2012). Measuring instance difficulty for combinatorial optimization problems.Computers& Operations Research, 39(5):875 – 889.
21. Ioannis X. Tassopoulos and Grigorios N. Beligiannis (2012).Solving effectively the school timetabling problem using particle swarm optimization.Expert Syst. Appl., 39(5):6029–6040.

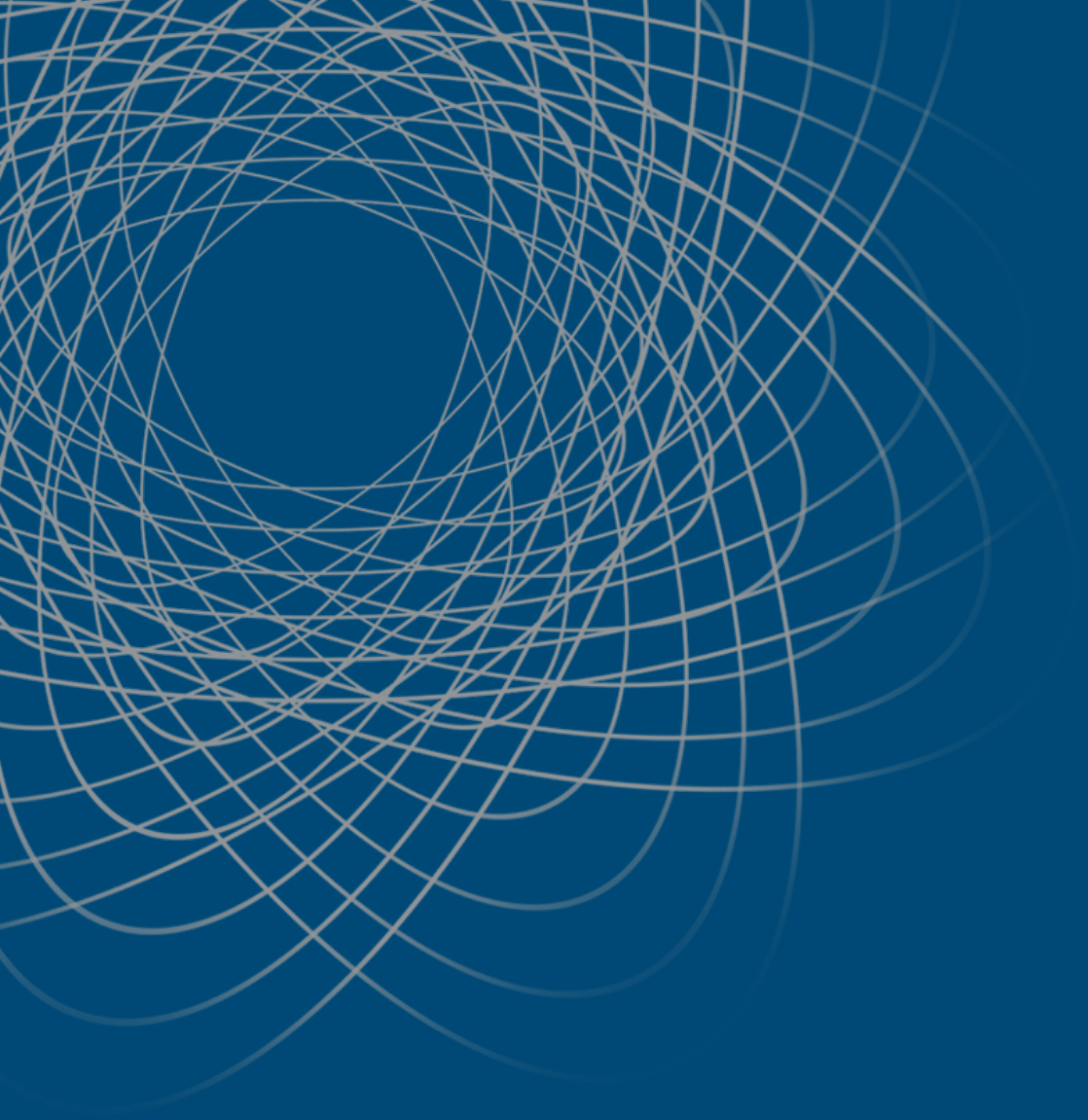
Marco A. Peña Luna is a professor in the college of engineering at CETYS University. He received the MS degree in Computer Science and Engineering from the University of Texas at Arlington in 1995 and a Doctoral degree in Engineering with specialization in computer networks and communication systems in 2013 from CETYS University. His interest lies in the area of scheduling, distributed systems, parallel computing and computer networks.

Adan Hirales Carbajal graduated from the Faculty of Sciences of the Autonomous University of Baja California, Ensenada B.C. Mexico, with specialization in Computer Sciences in 1999. He received the MS degree in Computer Science from CICESE Research Center in 2001 and the Ph. D. also in Computer Science from CICESE in 2012. He is a current member of the National System of Researchers of Mexico (SNI), Candidate.His research deals with different aspects of parallel, Grid computing, and cloud computing. His main interests include autonomic computing, scheduling, distributed systems and algorithms, multi-objective optimization, Grid and Cloud computing, parallel computing, and performance evaluation.

Andrei Tchernykh is a researcher in the Computer ScienceDepartment, CICESE Research Center, Ensenada, Baja California, Mexico. From 1975 to 1990 he was with the Institute of Precise Mechanics and Computer Technology of the Russian Academy of Sciences (Moscow, Russia). He received his Ph.D. in Computer Science in 1986. In CICESE, he is a coordinator of the Parallel Computing Laboratory. He is a current member of the National System of Researchers of Mexico (SNI), Level II. He leads a number of national and international research projects. He is active in grid-cloud research with a focus on resource and energy optimization. He served as a program committee member of several professional conferences and a general co-chair for International Conferences on Parallel Computing Systems. His main interests include scheduling, load balancing, adaptive resource allocation, scalable energy-aware algorithms, green grid and cloud computing, eco-friendly P2P scheduling, multi-objective optimization, scheduling in real time systems, computational intelligence, heuristics and meta-heuristic, and incomplete information processing.

Miguel Alberto Salinas Yáñez is the director of the Engineering College at CETYS University. He holds a Ph. D in Engineering with specialization in Computer Networks and Communication Systems. His main interests include autonomic computing, pervasive networking, combinatorial problems, and different aspects of distributed systems.

APPENDIX I



KEYNOTE SPEAKERS

Trusted and Secure Cloud Computing

Pascal Bouvry
Faculty of Sciences, Technology
Luxembourg University, Luxembourg

Abstract

Cloud computing is definitely the way to go. What are the new security requirements in the context of cloud computing? Would you trust another company to host your IT and their other customers sharing the same hardware? The presentation intends to highlight the challenges of cloud computing in terms of security and confidentiality issues. Current systems limitations will be described as well as compliance issues to security and auditing standards. Finally new generations of hardware and software based solution based on crypto-chips, ARM processors and new security protocols and services will be introduced.

Pascal Bouvry earned his undergraduate degree in Economical & Social Sciences and his Master degree in Computer Science with distinction (‘91) from the University of Namur, Belgium . He went on to obtain his Ph.D. degree (‘94) in Computer Science with great distinction at the University of Grenoble (INPG), France. His research at the IMAG laboratory focussed on Mapping and scheduling task graphs onto Distributed Memory Parallel Computers. Next, he performed post-doctoral research on coordination languages and multi-agent evolutionary computing at CWI in Amsterdam.DrBouvry gained industrial experience as manager of the technology consultant team for FICS (SONE) a world leader in electronic financial services. Next, he worked as CEO and CTO ofSDC, a Saigon-based joint venture between SPT (a major telecom operator in Vietnam), Spacebel SA (a Belgian leader in Space, GIS and Healthcare), and IOIT, a public research and training center. After that, DrBouvry moved to Montreal as VP Production of Lat45 and Development Director for MetaSolv Software (ORCL), a world-leader in Operation Support Systems for the telecom industry (e.g. AT&T, Worldcom, Bell Canada, etc). Dr. Bouvry is currently serving as Professor in the (CSC) research unit of the Faculty of Sciences, Technology of Luxembourg University. Pascal Bouvry is also faculty of the Interdisciplinary Center of Security, Reliability and active in various scientific committees and technical workgroups (IEEE CIS Cloud Computing vice-chair, IEEE TCSC GreenIT steering committee, ERCIM WG,ANR, COST TIST, etc.)

How to Design a Resource Management Method for Large Data Centers

Uwe Schwiegelshohn
TU Dortmund University, Germany

Abstract

Compared to the past, today's resource management for large computer systems has become more complex since it must consider various additional constraints, like virtualization, failure tolerance, and energy efficiency. Therefore it is not surprising that almost every conference on supercomputers or on parallel processing covers the topic resource management with one or more sessions. Although a very large number of different algorithmic approaches has already been proposed to improve efficiency of these computer systems, only very few of them are actually used in real machines, like EASY backfilling. In this talk, we discuss the reasons for this disparity. Then we suggest some rules that future work should consider in order to point out the applicability and the benefit of a new scheduling algorithm. These rules are relevant for research that emphasizes practical relevance of the presented algorithms. Exemplarily, we show the application of these rules when developing a new method to manage computing resources in the Infrastructure-as-a-Service (IaaS) model of Cloud Computing.

Uwe Schwiegelshohn received the Diploma and the Ph.D. degrees in Electrical Engineering from the TU Munich in 1984 and 1988, respectively. He was with the Computer Science department of the IBM T.J. Watson Research Center from 1988 to 1994 before becoming full Professor at TU Dortmund University. In 2008 he was appointed vice rector finance of this university. Also in 2008 he became managing director of the Government sponsored D-Grid corporation to coordinate the Grid research effort in Germany. From 2002 to 2012 he was organizer of the Workshop on Job Scheduling Strategies for Parallel Processing. In addition, he has been chairman and member of the program committees for various other conferences and workshops. His present research interests are scheduling problems, resource management for large computer systems and virtual research environments.

Big Displays for Big Data-the Next WAVE

Thomas DeFanti
Research Scientist, Qualcomm Institute
University of California, San Diego

Abstract

The advent of tiled ultra-narrow bezel commercial signage displays a few years ago dramatically altered the way information is shown at large scale. These video walls have become ubiquitous in advertising and in TV newsrooms, although they are employed as wall-sized, very bright HDTVs, not for display of big data. However, tiled video walls used as means to visualize big data coming over big networks have become integral to scientific communication, artistic performances, and exhibitions at universities. International video wall collaborations have been ongoing for several years. At UCSD, both 2D and stereo 3D walls are in use, displaying big data up to 64 megapixels resolution, as are the new generation of 4K UHD (Ultra-High-Definition) LCD displays. Specific effort has been invested to optimizing high-speed local and distance file serving and collaboration using multiple 10Gb/s and 40Gb/s networks and software tuned to synchronized image sharing (SAGE), extremely high-resolution static and streaming image viewing (MediaCommons), and immersive virtual reality experiences (CalVR), as well as to accurately handling with focused SoundBender audio speaker arrays and advanced echo cancellation. Recent results adapting flash memory big data technology championed by the San Diego Supercomputer Center to SSD-based "FIONA" PCs driving 2D/3D big data displays locally with up to 40Gb/s network interfaces, attached to 100Gb/s wide-area networks, along with their applications will be presented. Applications in omics and archaeology, are two UCSD examples with great international potential. The latest big displays at UCSD, the WAVE and WAVElet, and use of emerging UHDTV (4K) panels will also be described in detail.

Thomas A. DeFanti, PhD, is a research scientist at the Qualcomm Institute, a division of the California Institute for Telecommunications and Information Technology, University of California, San Diego, and a distinguished professor emeritus of Computer Science at the University of Illinois at Chicago (UIC). He is principal investigator of the NSF IRNC Program TransLight/StarLight project. He is recipient of the 1988 ACM Outstanding Contribution Award and was appointed an ACM Fellow in 1994. He shares recognition with fellow UIC professor emeritus Daniel J. Sandin for conceiving the CAVE virtual reality theater in 1991.

Parallel metaheuristics: efficient optimization for the supercomputing era

Sergio Nesmachnow
Universidad de la República
Multidisciplinary Center for High Performance Computing
Uruguay

Abstract

Metaheuristics are high-level soft computing strategies that define algorithmic frameworks and techniques able to find approximate solutions for search, optimization, and machine learning problems. They are highly valuable techniques, which allow researchers and practitioners to meet realistic resolution delays in many fields of application, ranging from informatics (combinatorial optimization, bioinformatics, software engineering, etc.) to industrial and commercial (logistics, telecommunications, engineering, economics, etc.). Parallel models for metaheuristics have been conceived to enhance and speed up the search. By splitting the search workload into several computing elements, parallel metaheuristics allow reaching high quality results in a reasonable execution time even for hard-to-solve optimization problems. This talk introduces the main concepts about metaheuristics as problem solvers and provides a general view of the field of parallel implementations for metaheuristics, including the topics of implementation in new computing devices and supercomputing infrastructures, and also the main lines of application in real-world problems.

Sergio Nesmachnow is a Full Time Professor at Universidad de la República, Uruguay, with several teaching and research duties. He is Researcher at National Research and Innovation Agency (ANII) and National Program for the Development of Basic Sciences (PEDECIBA), Uruguay. His main research interests are scientific computing, high performance computing, and parallel metaheuristics applied to solve complex real-world problems. He holds a Ph.D. (2010) and a M.Sc. (2004) in Computer Science, and a degree in Engineering (2000) from Universidad de la República, Uruguay. He has published over 90 papers in international journals and conference proceedings. Currently, he works as Director of the Multidisciplinary Center for High Performance Computing (Universidad de la República, Uruguay) and as Editor-in-Chief for International Journal of Metaheuristics, while he is also Guest Editor in Cluster Computing and The Computer Journal. He also participates as speaker and member of several technical program committees of international conferences and is a reviewer for many journals and conferences. E-mail: sergion@fing.edu.uy, Webpage: www.fing.edu.uy/~sergion.

Cyberinfrastructure Tools for Science, Engineering, and Humanities

Jarek Nabrzyski
University of Notre Dame
Center for Research Computing, USA

Abstract

Purdue's HUBzerois growing leaps and bounds these days, powering everything from simple collaborative websites to compute intensive portals such as nanoHUB. Does thismean it's a great fit for any science portal? When should HUBzero be used and when other tools are a better fit? In this talk I will present Notre Dame's Center for Research Computing experiences with building science gateways, where both, HUBzero and non-HUBzero solutions are used.

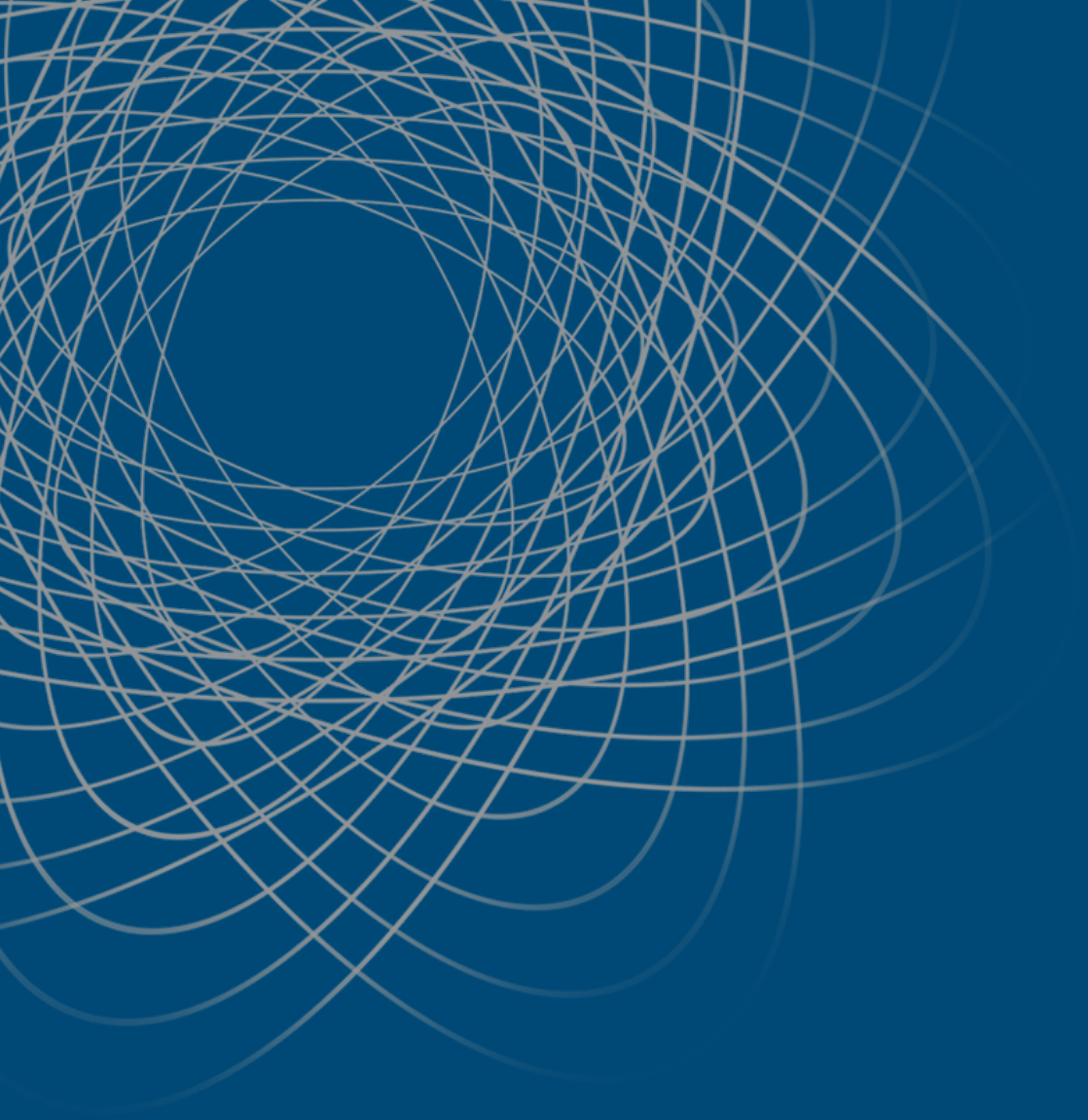
Jarek Nabrzyski is the director of the University of Notre Dame's Center for Research Computing. Before coming to Notre Dame Nabrzyski led the Louisiana State University's Center for Computation and Technology, and before that he was the scientific applications department manager at Poznan Supercomputing and Networking Center (PSNC), where he got interested in building science gateways and distributed computing middleware tools. Nabrzyski has built a team that developed and supported the GridSphereportlet framework, and later the VineToolkit framework. Both were used in many grid computing collaborations worldwide. While in Europe Nabrzyski was involved in more than twenty EC funded projects, including GridLab, CrossGrid, CoreGrid, GridCoord, QoSCoSGrid, Inteligrid and ACGT. Within his last five years at Notre Dame Nabrzyski has been focused on building the Center for Research Computing. The Center, a 40+ staff and faculty research enterprise has been involved in many research projects funded nationally and internationally. Nabrzyski has received his M.Sc. and Ph.D. in Computer Science and Engineering from the Poznan University of Technology in Poland. His research interests cover distributed resource management and scheduling, cloud computing, scientific portals, and decision support systems for global health and environmental applications.

Building a Robot Cloud

Dr. Zhihui DU
Department of Computer Science and Technology
Tsinghua University

Abstract

Dr. Zhihui DU is an associate professor in the Department of Computer Science and Technology at Tsinghua University. His principal research interests lie in the field of High Performance Computing and he participated in building several cluster systems, including one TOP 500 supercomputer. He has published more than 100 academic papers and authored/coauthored 5 books on parallel programming and grid computing. He has served on numerous program committees and is vice program chair for IPDPS, and associate editor for the Journal of Parallel and Distributed Systems and the International Journal of Parallel, Emergent and Distributed Systems. Cloud computing, a key computing platform which can share resources that include infrastructures, platforms, software applications, and even everything, known as "X as a Service ", is shaping the cyber world. But the existing cloud computing is limited in addressing the problems which can only be modeled in the cyber world. There are more problems in the physical world and we need a bridge which can connect the cyber world with the physical world seamlessly. Based on the concept of robot as a service in cloud computing, we provide a design of robot cloud. In this talk, I will give the details of our design on robot cloud architecture and an application scenario to explain how to provide high quality service for users and obtain more benefit for the robot cloud center by optimizing the robot services.



TECHNOLOGY SPEAKERS

Cray Solutions for Higher Education and Research

Dr. David Barkai
Sponsored by Cray

Abstract

Cray has always been known for its high-end, high-performance, compute systems. It has a strong presence at the top national academic computation centers in the world. This talk will highlight some of the pioneering science done on these systems. The focus of the talk, however, will be on some of the less known products Cray brings to the academic research. This includes a standard cluster offering (the result of the Appro acquisition over a year ago) that combines high quality standard components with the software and services Cray is known for; and a full range of storage products, including a Cray-design Lustre appliance and a Hadoop product. The talk will cover Cray's end-to-end solutions that are optimized for both compute and data intensive workloads.

Bio.

Dr. David Barkai is Business Development Director for Higher Education at Cray Inc. He joined Cray after it acquires Appro, where David worked after leaving Intel. Until 2011 he was an HPC (High Performance Computing) computational architect for Intel Corporation during most of his 15 years there. He also held a number of positions within Intel research labs involving peer-to-peer and investigations of the impact of emerging technologies on society. Before joining Intel in 1996, David worked for over 20 years in the field of scientific and engineering supercomputing for Control Data Corporation, Floating Point Systems, Cray Research Inc., Supercomputer Systems Inc., and NASA Ames Research Center. David received his B.Sc. in physics and mathematics from the Hebrew University in Jerusalem and a Ph.D. in theoretical physics from Imperial College of Science and technology, London University, in the UK. Over the years David has published in various forums on subjects in physics, numerical methods, and computer applications and architectures. He authored the book "Peer-to-Peer Computing: Technologies for Sharing and Collaborating on the Net" (Intel Press, 2001) and other articles on related topics.

CERN openlab: a Model for Research, Innovation, and Collaboration

Dr. (Ph.D.) Alberto Di Meglio
Sponsored by Intel

Abstract

CERN runs one of the most technologically advanced machines in the world, the Large Hadron Collider (LHC), and is at the heart of the Worldwide LHC Computing Grid (WLCG), required to process its data. Thousands of scientists and engineers collaborated to solve challenges in particle physics, but also in massive data storage and distribution, data analysis, simulation, computing resource provisioning, or distributed software development. For more than 10 years, CERN openlab has established partnerships between CERN and leading IT companies to support the LHC programme. This presentation highlights the current achievements and the future technical challenges in the ongoing LHC programme and how CERN openlab and its partners are contributing to its success. The role of companies such as Intel in current and future activities is highlighted as an example of innovation and collaboration between CERN and leading commercial companies.

Bio.

Alberto Di Meglio is the CERN openlab CTO and a senior project manager in the CERN IT Department with more than 15 years of experience in IT systems and software engineering. Following an M.Eng. in Aerospace Engineering and a Ph.D. in Electronic and Electrical Engineering, he worked as university research associate and founded and run for three years a commercial software company before joining CERN as lead IT systems engineer. Before joining openlab Alberto was the leader of the European Commission funded EMI project that coordinated the development of most of the grid software used worldwide to process LHC data.

AMD Technology for HPC Solutions

David Garza Marín, Engineering Director in AMD Latin America
Sponsored by HP y AMD

Abstract

Technical elements will be presented that will allow to give serial considerations to ADM technologies for HPC solutions.

Bio.

David Garza joined AMD June of 2002 as a Field Applications Engineer. Previously, David Garza served as Director General of his own enterprise PRO-2. In his 28 years career in the technologies industry, David Garza has accomplished a great variety of field posts in México, that include companies like Microsoft, IBM, Pearson Educación, Sayrols, Vendor, Neodata, Lotería Nacional para la Asistencia Pública y Bolsa Mexicana de Valores, among others.

Has authored and coauthored three books on Computation and many papers and columns in different hard-copy and electronic media. Also has participated in televised and radio presentations about computer technologies both, nationally and internationally. Also, has participated as keynote speaker about technology in México, United States, Argentina, Brazil, Colombia, Dominican Republic, Costa Rica, Uruguay and several other countries. David lays his basic communication techniques on a maxim that says: "There's no worst technology, that the one you don't understand," so he has made his mission in making sure that technology becomes people's real ally. As a matter of fact, his post in AMD is greatly related to this, given that among his responsibilities, one is to make sure that technology is adequately applied by his field clients. He is a Computer Engineer with specialty in Information Technologies (UNAM), and Programmer Analyst in Computing Systems and Architectures (Instituto de Computación y Métodos). Has been distinguished as "Most Valuable Professional" by Microsfot since 2000 to 2003, and received the award "Employee of the Year" by AMD in 2008.

Supercomputing Macroprojects in CNS

Dr. César Díaz Torrejón, General Coordinador of CNS
Sponsored by CNS-IPICYT

Abstract / Bio.

César Carlos Díaz Torrejón, graduated B. Sc in Physics from UNAM's Faculty of Sciences, where he later obtained M.Sc. in Computer Modelling of Molecular Systems and Extended Solids. And Ph.D. en Materials Science by the Centro de Investigación en Materiales Avanzados A.C.

He has authored several scientific articles and science divulgation in national as international journals in the fields of Molecular Modelling, Computational Quantic Chemistry, Amorphous Semiconductors and High Performance Computing (HPC). He has participated in numerous congresses at national and international venues in Molecular Modelling and Supercomputing. Since 2005 is member of Sistema Nacional de Investigadores (SNI).

Faculty member from 1998 to 2005 of Science Faculty of UNAM in the schools of Mathematics, Physics and Computer Sciences. From June of 2001 to June 2005, was coordinator of computing and supercomputing in the Instituto de Investigaciones Materiales of UNAM, time in which he incorporated Clusters Beowulf technologies in the areas of computational modelling, as well as incorporation of Linux Operating System in several development areas of the institute.

His areas of expertise include: design, building and make operational Linux Clusters Beowulf type; use and administration of supercomputers, GRID technologies, and Unix-Linux Operating Systems in general, as well in computer programming scientific languages C, and FORTRAN, parallel computing OpenMP, MPI, etc. As of now, he is General Coordinator of the Centro Nacional de Supercómputo del Instituto Potosino de Investigación Científica y Tecnológica.

Accelerating Development of Cures: BigData Solutions for Biomedical Research

Alejandro Martínez Carmona,
HPC specialist

Abstract

The convergence of efforts form software engineers and investigators, has rendered fruits. The utilization of BigData solutions for analysis and DNA sequencing reduces search time of hereditary-illness from years to just days. Dell's BigData solutions help medical researchers analyze DNA sequences faster than ever. In this talk we'll be able to analyze how Dell's BigData solutions help researchers around the world find cures in a faster way.

Bio.

Alejandro Martínez graduated in Cibernetic Engineering and Computer Systems. Then obtained a M.Sc. in Economic and Financial Engineering form La Salle University. Has 16 years of experience in the field of Information Technology (IT) as consultor in the areas of programming and hardware architecture solutions from 2000 to date. As of now, he is HPC Champion en HPC for Hispanic speaking area, and has been at Dell for the past four years.

Exit Case: Ocotillo - UNISON Scientific Cluster

Carmen Heras
Sponsored by Qualisys-DELL

Abstract / Bio.

The careers of Computer Sciences and Information Systems, where besides collaborating in the projects like Laboratorio Nacional de Grids de Supercómputo para el Soporte de la E-Ciencia, and other grid initiatives in México. Member of the JRU-MX (Mexican Joint Research Unit). Coordinator of the Consejo Consultivo de Supercómputo of Universidad de Sonora, has presided over two Northwest Supercomputer Symposiums in 2006 and 2008. Assisted and participated as presenter in congresses and academic events national and international level related wit High Performance Computing (HPC). Has been promoter of collaborations between Mexican and American universities. Member of the Organizing Comity for the International Congresses on Supercomputing: ISUM2010, ISUM2011 and ISUM2012. Director of Iniciativa de la Asociación Mexicana de Cómputo Avanzado and member of Consejo Especial de E-infraestructuras in México.

Jose Carlos Huescas, Products Manager, HPC specialist
BM Sponsored

Abstract / Bio.

Jose Carlos Huescas is IBM's Products Manager of the line 'X86 - Pure Systems' in IBM. Has an ample trajectory and experience in the world of Information Technologies. Among his areas of expertise are Cloud Computing, High Performance Computing and Business Analytics and Virtualization. Inside IBM, he has participated and lead in solution's design for México and Central America, where there are more than 300 consolidation studies of data centers including the biggest case of virtualization in Latin America, recently he participated on the implementation of Latin America's biggest computer located in San Luis Potosí, México. He is a Systems Engineer by Instituto Politécnico Nacional (IPN) with specialty in Neural Networks and Projects Management.

The present and future of computing on NVIDIA GPUs

Arnaldo Tavares
Sales Manager of NVIDIA for Professionals Products in Latin America
Sponsored by Nvidia-LUFAC

Abstract / Bio.

Armando Tavares is responsible of defining the sales strategy and increasing business with Tesla, Quadro and GRID, besides CUDA Technology, looking for partnerships with computer manufacturers and software providers for implementing Centers for learning and research in CUDA. Tavares graduated in Engineering from the Centro Federal de Educación Tecnológica of Río de Janeiro and has an MBA from University of Southern California (USC) in Los Angeles, California.

LUFAC: Advances in Areas of Research and Development (R+D)

M. en C. José María Zamora Fuentes
Sponsored by LUFAC

Abstract

In the last months, LUFAC® applying HPC technologies state of the art (Xeon PHI®, GPU, Infiniband®), has consolidated his area of research generating solid scientific results in a diverse fields of knowledge, among which outstand dynamic molecular simulations on a record number of particles. The advancements on the investigations realized at LUFAC® show the potential of the new HPC technologies that can be applied in any area of investigation. In this presentation we will show the results and motivations as well as future projects inside the R&D Department at LUFAC®. LUFAC® is a 100% mexican Enterprise with 17 years in the development of solution for HPC.

Bio.

José María Zamora Fuentes graduated as Electronics Engineer by Universidad Autónoma Metropolitana. Later he obtained M.Sc. from Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS-UNAM). Has realized several academic investigation visits including at Notre Dame University. For the past eight years he has occupied posts as programmer, administrator and software designer for several organizations and institutions, outstanding for his interest in the scientific area, in particular: molecular dynamics. As of right now, he is project leader in Research and Development inside Lufac Computación S.A. de C.V.

Round Table: HPC Across Industries

Participants:
Sponsored by Cadgrafics-DELL
Moderator: Jorge Marrón Director de Soluciones Empresariales de DELL México
Industry: Energy / Instituto Mexicano del Petróleo / Ing. Arturo Díaz Bustos

Industry: Education and Investigation / Universidad Autónoma Metropolitana / Mtro. Juan Carlos Rosas Cabrera /Coordinador Técnico del Laboratorio de Supercómputo y Visualización en Paralelo y Responsable Técnico del Laboratorio Nacionalde Cómputo de Alto Desempeño

Industry: Entertainment / Huevocartoon / Ignacio Martínez Casares / Productor Huevocartoon Ing. Arturo Díaz Bustos

Summarized resume Ing. Arturo Díz Bustos

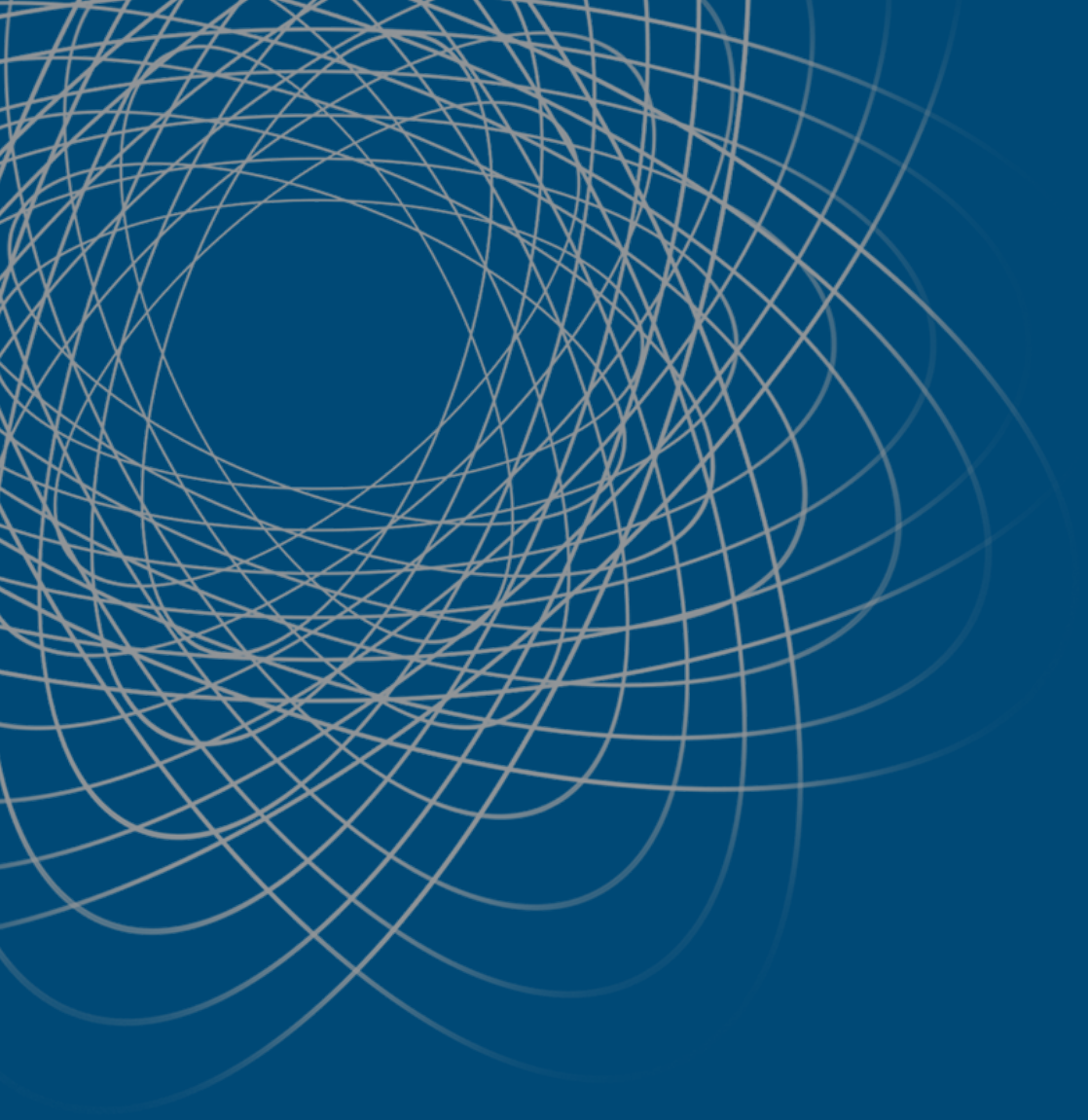
Ing. Arturo Díaz obtained B.Sc. and M.Sc. in Electronics Engineering from Instituto Politécnico Nacional (IPN). As of right now, is representative of the specialized processes domain of the Instituto Mexicano del Petróleo (IMP), where he is in charge of identifying and proposing emerging technologies applicable to the institution in matter of specialized processing for performance improving of the domain elements. Besides he is responsible of the administration of the Centro de Tecnología en Informática Especializada para Interpretación y Procesamiento de Información Geológica y Geofísica, with the objective of generating infrastructure in specialized last-generation software and hardware, among which there is High Performance Computing (HPC). Has participated in public tenders conducted by the IMP for the supply of high performance computing for the past 6 years; has been administrator of different HPC equipment for exploration and production for the past 15 years; and has imparted courses in Mathematics, Computer Sciences and Programming in Mexico's three principal academic institutions: Universidad Autónoma de México, Instituto Politécnico Nacional and Universidad Autónoma Metropolitana.

Summarized resume Mtro. Juan Carlos Rosas Cabrera

Ing. Juan Carlos Rosas Ehas a multidisciplinary formation in Engineering and Organizational Studies. He is Technical Coordinator of Laboratorio de Supercómputo of Universidad Autónoma Metropolitana. Has participated and coordinated important institutional projects of High Performance Computing and Connectivity: like Study of Factibility of Wide Band Project and development of NIBA Network (Red Nacional de Impulso a la Banda Ancha) for the Secretary of Communications and Transport (SCT), en 2012. Participated and supervised the infrastructure and connectivity development of the Proyecto de Conectividad de la Delta Metropolitana de Cómputo de Alto Rendimiento between la Universidad Nacional Autónoma de México (UNAM), Centro de Estudios Avanzados (Cinvestav) del Instituto Politécnico Nacional (IPN), y la Universidad Autónoma Metropolitana (UAM) in 2010-2011. In 2009 supervised and coordinated the construction of the biggest HPC in Latin America with more tan 2000 nucleus, putting it in the Top500 category; Has implemented strategies for Human Resources formation with specialty on Information Technologies and Communications. Has had important inter-institutional collaborations relating his expertise in: el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), La Universidad de Sonora (UNISON), Universidad de Guadalajara (UDG), Universidad Veracruzana, Universidad de Colima (UCOL), UNAM, Cinvestav, Corporación de Universidades para el Desarrollo de Internet 2 (CUDI), Instituto Mexicano del Petróleo (IMP), Petróleos Mexicanos (PEMEX), Coordinación de la Sociedad y el Conocimiento de la Secretaria de Comunicaciones y Transportes (CSIC-SCT), Sistema de Transporte Colectivo "Metro", Instituto de Ciencia y Tecnología del Gobierno del Distrito Federal, Consejo Mexiquense de Ciencia y Tecnología del Estado de México (COMECyT), Consejo Nacional de Ciencia y Tecnologiía (CONACyT), among many others; Has participated in collaboration on international projects with: La Universidad de Andrés Bello de Santiago de Chile and Instituto Lingüístico Computacional de CERN de PISA Italia; Has promoted liasons between the production sector and private initiatives for business development and sustentability strategies between UAM and: Hewlett Packard de México, IBM, Dell, Cisco, Brocade, Intel, AMD, EMC, Vmware, Mellanox, APC-Schnneider, Sonda-NextiraOne SA de CV, Migesa SA de CV, Proatech SA de CV, Infoglobal SA, Comnet SA de CV, Root Technologies SA de CV, among others otras. Has participated as member of the Organismo Nacional de Normalización y Certificaciones de la Construcción y Edificación SC (ONNCCE) for the design of High Performance Datacenters; Has realized and coordinated several Training Courses, Workshops, Conferences and Lectures in the area of HPC and High Speed Communications.

Summarized resume Ignacio Martínez Casares, Productor Huevocartoon

Graduated in Communication Sciences (UNLA 2005), Nacho Casares studied in Madrid, Spain, Master in Audovisual Production (UCM 2007). In Spain, he worked in the area of Chanel+ Production, and collaborated with several international companies like: Thinking Heads, Creative Society and Imagina. In México, he has been Chief of Production of Channel 13 of Michoacán, (Televisa 2008-2010), Director assistant in the Novel "Amar de Nuevo" (Argos-TeleMundo 2011) and Production Assistant for the film "Morelos" (Astillero Producciones, 2011). Has produced and directed six independent projects among others: "B7aacute;rbara" (Short Film), "Tierra Dentro" (Serie Documental), "Fábrica de Huevos" (Azteca 7 and Azteca América 2012). Has collaborated in diverse publicity companies like: 35 AM, Monster GC y MacCann. Right now, is producing the film "Un Gallo con Muchos Huevos" (Huevocartoon 2012-2014) along several other projects with his production Company +1 Producciones.



ABSTRACTS

A GPU implementation of Markov Decision Process for Path Planning

Sergio Ruiz[1], Benjamín Hernández[2], Isaac Rudomin[2]
[1]*Tecnológico de Monterrey, Campus Ciudad de México, México*
[2]*Barcelona Supercomputing Center, Barcelona, Spain*

Abstract

Navigation refers to the ability of agents to effectively avoid collisions against static or dynamic objects within an environment while moving toward their goal. Classical path finding methods [1,2,3] had been used extensively; however, in real world planning problems where environment information is incomplete or dynamically changing, Markov Decision Process (MDP) has been used as an alternative [4,5]. Nevertheless, the problem with the MDP formalism is that the state space grows exponentially with the number of domain variables, and its inference methods grow in the number of actions [6]. To overcome this problem, we present a MDP implementation and its related data structures using Thrust [7] to produce obstacle-free paths (best policy) with an application in crowd simulation. The algorithm makes use of Value Iteration technique to find the best policy in order to steer virtual agents. We will show that our algorithm is suitable when real time response is needed. In addition, we will present an analysis of the execution of our algorithm on different mobile, gaming and scientific Kepler architectures under CUDA 5 and CUDA 6.

References

[1] RINA, D., AND PEARL, J."Generalized best-first search strategies and the optimality of A*". Journal of the ACM(1985).

[2] KOENIG, S. AND LIKHACHEV, M."D* lite". 18th Conference on Artificial intelligence (2002).

[3] KOENIG, S. AND LIKHACHEV, M. AND FURCY, D."Lifelong planning A*".Journal of Artificial Intelligence(2004).

[4] FOKA, A.F.; TRAHANIAS, P.E."Predictive autonomous robot navigation".Intelligent Robots & Systems(2002).

[5] FOKA, A.F.; TRAHANIAS, P.E., "Predictive control of robot velocity to avoid obstacles in dynamic environments," Intelligent Robots & Systems(2003).

[6] REYES, A., AND SUCAR, L. "An operation auxiliary system for power plants based on decision-theoretic planning". International Conference on Intelligent Systems Application to Power Systems(2005).

[7] BELL and HOBEROCK. "Thrust: A Productivity-Oriented Library for CUDA". GPU Computing Gems, Jade Edition (2011).

Developing a Cloud Computing Testbed Using OpenNebula.

A. Espinoza Godínez [a], J.L. Garza Rivera [b], J.J. Cruz Guzmán [c]
Universidad Nacional Autónoma de México. Facultad de Estudios Superiores Cuautitlán.
Carr. Cuautitlán-Teoloyucan Km. 2.5, San Sebastián Xhala, Cuautitlán Izcalli, Edo. Mex. C.P. 54714.
E-mail: anesgo@comunidad.unam.mx [a], jlgr@unam.mx [b], cruz@unam.mx [c].

Abstract

In the last few years, cloud computing has occupied a very important role in several scopes into organizations, for instance, in technological and economic scopes. Knowing, using and developing cloud computing paradigm is becoming more imminent for letting new ways of collaboration among organizations and communities. We present the procedure followed to develop a cloud computing testbed to promote further collaboration among some research and education communities in Latin America. This cloud computing testbed, built with open source software, is based on OpenNebula project. It is defined as a private cloud deployment model and Infrastructure as a Service(IaaS) model.

References

1. P. Mell and T. Grance (2011). The NIST Definition of Cloud Computing. [Online]. National Institute of Standards and Technology (NIST), U. S. Department of Commerce. Retrieved: August 28, 2013 from http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

2. OpenNebula Project (2002-2013). OpenNebula.org. Retrieved: September 20, 2013 from http://www.opennebula.org/

3. L. Wang, R. Ranjan, J. Chen, and B. Benatallah (2012). Cloud Computing: Methodology, Systems and Appications. Boca Raton, FL (USA): CRC Press.

4. G. Toraldo. (2012). OpenNebula 3 Cloud Computing. Birmingham, Reino Unido: Pack Publishing.

Monitoreo y pronóstico de condiciones meteorológicas para Baja California

Luis M. Farfán[1], Julián Delgado[2], Ismael Villanueva[2], José Lozano[2],
Salvador Castañeda[2] y Sylvia Camacho[2]

1: Unidad La Paz Centro de Investigación Científica y de Educación Superior de Ensenada, B.C. Miraflores #334, La Paz, Baja California Sur E-mail: farfan@cicese.mx

2: Dirección de Telemática Centro de Investigación Científica y de Educación Superior de Ensenada, B.C. Carretera Ensenada-Tijuana No. 3918, Ensenada, B.C

E-mail: judelga@cicese.mx, ivillanu@cicese.mx, jlozano@cicese.mx, salvador@cicese.mx, sylvia@cicese.mx

Resumen

Se describe la estructura y funcionamiento de un sistema de análisis de las condiciones meteorológicas, en tiempo real, para la península de Baja California. Este sistema cuenta con una componente de recepción de datos y procesamiento de la información para generar productos gráficos que se ponen a disposición del público en el portal http://met-wrf.cicese.mx. Por otra parte, se presentan pronósticos del tiempo en plazos de 72 horas y esto aplica un modelo numérico regional en equipo de alto rendimiento. Estos pronósticos permiten hacer una estimación de cambios en las condiciones de temperatura, humedad, viento y lluvia.

La información se recibe de una fuente que maneja varios tipos de datos. Esto incluye satélites geoestacionarios que capturan la estructura y movimiento de la atmósfera mediante imágenes visibles, infrarrojas y vapor de agua. Otra fuente son las estimaciones de lluvia por medio de los radares del Servicio Meteorológico de Estados Unidos. Adicionalmente, el portal presenta reportes de una red de estaciones meteorológicas del Servicio Meteorológico de México para determinar las condiciones de la atmósfera en 25 estaciones a lo largo de la península y estado de Sonora, así como varias islas en el Océano Pacífico. Un modelo numérico, de cobertura global, se utiliza para revisar predicciones del tiempo sobre Norteamérica y zonas marítimas que le rodean. Estos productos están a disposición del público en general, investigadores y estudiantes así como de las autoridades con la responsabilidad de proporcionar alertas a la población de la región.

Finalmente, se presenta un resumen de los resultados de aplicar un modelo de alta resolución espacial y temporal para la simulación de huracanes. Se han analizado varios eventos de las temporadas 2006-2013 y se describen los resultados de varios casos relevantes que han tenido impacto en tierra tanto en la población como en el medio ambiente.

Design optimization of 2D-airfoils shapes for micro-aerial vehicles

Armando Oropeza-Osornio [1], Alfredo González [1], Ivvan Valdez-Peña [2],
Eusebio Hernández-Martínez [1]
[1] *SEPI UP Ticoman, Instituto Politécnico Nacional*
[2] *CIMAT Centro de Investigación en Matemáticas A.C.*

Abstract

In this paper a robust methodology for the optimized design of two-dimensional low speed airfoils is presented. We propose a methodology in which the automatic design process starts when the user imposes basic geometrical characteristics, such as: the airfoil chord, thickness, camber and some operating parameters such as the Reynolds number and angles of attack for specific operating states Reynolds number and control points.

Based on this initial geometry the airfoil shape is parametrized by using Bezier curves. These generated points on the curves are used as inputs for aerodynamical evaluations which are executed by the Xfoil flow solver. An optimization tool introduced in this work and named Optimate is used to improve the airfoil shape. It is a set of optimization methods, mainly global evolutionary optimizers, programmed in C++ which allows to externally evaluate the proposed solutions, it is well known that evolutionary algorithms have the advantage of being highly parallelizable, hence, we used and MPI version of Optimate in order to reduce the computational time.

The results are promising, and show the capacity of the optimizer for tackling real-world optimization problems with a minimum effort. The resulting shapes from the optimization are shown to be better than the initial ones; Optimate improves the aerodynamic performance of airfoil surfaces.

Cálculo de Similitud de Artículos de Wikipedia Usando Cómputo Grid

Sidorov, Grigori, Soto Ramos Manuel Alejandro, Melara Abarca Reyna Elia, Barbosa Hernandez Flavio
Instituto Politécnico Nacional, Centro de Investigación en Computación
Instituto Politécnico Nacional, Escuela Superior de Cómputo

Resumen

El presente documento trata sobre el uso de un recurso de cómputo en Grid para trabajar con la información de los artículos de la enciclopedia en línea Wikipedia y a partir de ello, realizar un cálculo de la similitud entre artículos, por medio del modelo de espacio vectorial y la similitud coseno[2].

Las Wikipedias en diferentes idiomas, , han sido utilizadas para procesamiento de lenguaje natural[3][4][5] . En este sentido, se pretende recopilar los textos disponibles de los respaldos de la Wikipedia en español, integrar un recurso léxico grande[1] sin marcas Wiki que sea disponible para su explotación y realizar el cálculo de la similitud entre artículos de la colección.

Dado el volúmen de la información, una expectativa es reducir el tiempo de preprocesamiento del texto por medio del recurso Grid[6], en relación al tiempo que lleva hacerlo en un equipo de escritorio.

Referencias

[1] Procesamiento automático del español con enfoque en recursos léxicos grandes, Alexander Gelbukh y Grigori Sidorov, segunda edición, 2010.

[2] An Introduction to Information Retrieval, Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Online edition, 2009

[3] Using Wikipedia for Automatic Word Sense Di sambiguation, Rada Mihalcea, 2007.

[4] Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary, Torsten Zesch, Christof Müller, Iryna Gurevych, 2008.

[5] Learning to Link with Wikipedia, David Milne, Ian H. Witten, 2008.

[6] Supercomputing and High Energy Physics

Heuristics for Nested Dissection toReduce Fill-in in Sparse Matrix Factorizations

J. Miguel Vargas-Felix, Salvador Botello-Rionda
Center for Mathematical Research (CIMAT)

Abstract

Let A $R^{n\times n}$ be a sparse matrix, symmetric positive definite, with n large. The solution of a linear systems of equations Ax=b using the Cholesky factorization A=L L^T can be inefficient if it is applied directly, because L, although sparse, could have a large fill-in. We can work with less fill-in if, instead, we solve the system of equations (P AP^T)(P x)=(P b), where P is a permutation matrix. Cholesky factorization is then applied to the permuted matrix, L LT=(PAPT). The choice of P has a considerable effect on the amount of non-zero entries of L. The two most common heuristics used to reduce fill-in are the nested dissection method and the minimum degree method.

The nested dissection method consists on split the graph in two parts by selecting a set of vertices called separator. This procedure is applied recursively to each part, forming a dependency tree. P is created numbering the vertices of the tree from the top to the root. In this work we propose a new way to find separators by finding an analogous to a plane that cuts the graph in two.

The minimum degree method finds P looking local features of the graph instead of global ones like the nested dissection [1]. In our work, we use the minimum degree method when the amount of vertices is below a threshold, thus we use combine the two methods to try to get a better fill-in.

We will present some results of our work on the parallel solution of systems of equations with tens of million variables, on multi-core computers. We will also present a comparison of our method with other variants of the nested dissection.

References

[1] C. Ashcraft, J. W. H. Liu. “Robust Ordering of Sparse Matrices using Multisection”. York University. Department of Computer Science. 1996.

Análisis de rendimiento en simulaciones de N-cuerpos utilizando diferentes modelos de programación paralela

Irving Álvarez Castillo1, Octavio Valenzuela Tijerino2
1: Dirección General de Cómputo y de Tecnologías de la Información y Comunicación
Universidad Nacional Autónoma de México
2: Instituto de Astronomía Universidad Nacional Autónoma de México

Resumen

Las simulaciones de N-cuerpos ayudan al estudio de diversos fenómenos astronómicos como: agujeros negros y su interacción con estrellas o con las galaxias que los albergan, colisiones de galaxias, etc. En este trabajo se evalúan versiones de un código de N-cuerpos partícula-partícula[1] que utiliza un integrador simpléctico Leapfrog[2] en diferentes modelos de programación paralela (memoria compartida y distribuida). Los programas tienen diversos ajustes como: diferentes tipos de distribuciones de carga en OpenMP, traslape en el cálculo y comunicación en MPI2, entre otros. Se evalúa el rendimiento mediante las métricas de: tiempo de ejecución, speed up y eficiencia; con el fin de cuantificar la mejoría en los ajustes realizados y determinar cuáles son los más útiles. Se explorará otro integrador simpléctico que puede manejar una jerarquía de pasos en bloque[3].

Referencias

[1] Building a better leapfrog. Piet Hut, et al.

[2] Understanding the Behavior and Performance of Non-blocking Communications in MPI. Taher Saif, et al.

[3] N-body integrators with individual time steps from Hierarchical. Federico I. Pelupessy, et al.

Accessing HPC resources through Science Gateways

J.L. Garza Rivera, A. Espinoza Godínez, J.J. Cruz Guzmán, M. Álvarez Pasaye
Universidad Nacional Autónoma de México. Facultad de Estudios Superiores Cuautitlán.

Abstract

Accessing and managing of HPC(High Performance Computing) resources is a challenge for both Virtual Research Community members and HPC infrastructure administrators.

Science Gateways are environments that bring up a set of tools, data and applications available to advanced computing research communities through a graphical environment [2]. The aim of the Science Gateways is to unify and simplify the use and control of HPC resources, grid [1] , cloud and other kind of computational resources and applications [1,2] together. Access control issues are also simplified, or even improved, by implementing interaction of Science Gateways with Identity Federations.

We describe the procedure that we followed to develop the portlet[3] for HPC access through a Science Gateway, from making the development environment to implement it into the Science Gateway, reaching distributed HPC infrastructures located in several academic research centers in Mexico.

References

[1] J.L. Garza Rivera, A. Espinoza Godínez, V.K. Kharchenko, A. V. Lara Sagahon, J.J. Cruz Guzmán, "Experiences of gridification of a Right Coideal Subalgebras application and its implementation in Science Gateway", Proceedings of the Joint GISELA-CHAIN, R. Barbera et al(eds). COMETA 2012. ISBN: 978-88-95892-05-4.

[2] Catania Science Gateway Framework. [On line]. http://www.catania-science-gateways.it/

[3] Sarin, Ashish. "Portlets in Action", Manning Publications. New York. 2012. ISBN : 9781935182542.

Aplicación del Kinect para la prevención de allanamientos en casa-habitación

Licenciado Rubén Antonio Gallegos, Alexis Adrián Puentes Valencia
José Antonio Leyva Morando
Instituto Tecnológico Superior de Coatzacoalcos.

Resumen

El allanamiento en los hogares es un tipo de alta inseguridad para las personas, el índice de inseguridad demuestra ser un factor común en negocios y hogares los cuales presentan ser una de las principales víctimas del allanamiento. El índice de allanamientos a casa-habitación en el municipio de Coatzacoalcos del estado de Veracruz, ha ido en aumento en el presente año de manera alarmante, el observatorio ciudadano presentó un boletín hemerográfico de delitos; se presentaron 573 delitos dentro de los que se registra un incremento del 130 por ciento de robos a casa-habitación. En México La percepción de inseguridad relacionada con la ola de violencia subió a 72.3 % de la población entre marzo y abril, los primeros meses de la gestión del presidente Enrique Peña Nieto, mostró una encuesta del Instituto Nacional de Estadística y Geografía (Inegi). La percepción de inseguridad que mide el Inegi alcanzaba en el 2012 a un 66.6 % de la población y en el 2011 a 69.5% de los habitantes de México.

La tendencia general es que la inseguridad en México crece de manera exponencial, el robo a casa-habitación dentro del estado de Veracruz pasaron de 10 a 23 casos, casi similar a la cifra que se registró el mes de noviembre del año pasado. Cinco de los siete eventos registrados han sido con violencia, además de que la mayoría de ellos se presentaron durante la mañana o el medio día. Para prevenir el allanamiento en la casa-habitación, las personas recurren a la adquisición de cámaras de vigilancia y dispositivos de seguridad, pero debido a la gran gama de dichos dispositivos, estos tienden a ser poco factibles y altamente costosas dependiendo su tipo de funcionalidad.

Las cámaras vigilancia son dispositivos que manejan funciones limitadas dentro de su aspecto de seguridad tales como grabar video y reproducir vídeo. El implementar diferentes dispositivos de seguridad aumenta el grado de seguridad en las casa-habitación pero también genera un alto costo de adquisición ya que estos dispositivo genera una única función específica, se deben adquirir varios con distintas funciones para complementar un sistema de seguridad completo, a estos hay que sumar los gastos adicionales en instalación, actualización y mantenimiento del sistema de seguridad y los dispositivos externos para mantener un óptimo funcionamiento. Dentro de las cámaras de vigilancia con más demanda en el mercado se encuentran las cámaras DVR y las cámaras IP, dentro de los dispositivos de seguridad se encuentran los sensores de movimiento.

Las cámaras DVR presentan un monitoreo y grabación de video en tiempo real, pero el problema de estas cámara es que necesitan un servidor dedicado para digitalizar los videos capturado, por ende requieren un gran nivel de almacenamiento, llegando a ser altamente costoso. Las cámaras IP presentan un monitoreo y grabación de video en tiempo real, el

problema de este tipo de cámaras es su necesidad de internet por su función principal de acceso remoto. Esto presenta una vulnerabilidad de seguridad, ya que la cámara IP puede ser atacada y accesada por otro usuario de diferente parte.

Los sensores de movimiento son uno de los dispositivos de seguridad más utilizados, estos presentan un sensor dentro de un escenario establecido por el usuario; alertando mediante una alarma sonora cuando detecta un movimiento. Sin embargo estos sensores tienden a ser altamente sensibles y puede llegar a ocasionar un grave problema, ya que al ser sensibles pueden detectar el movimiento de algún animal o alimañana y disparar una alarma falsa. Por otra parte, se encuentra el Kinect es un dispositivo de videojuegos XBOX 360 desarrollado por Microsoft, cuenta con una cámara RGB, sensor de movimiento, sensor de profundidad y un lector de cámara infrarrojo y su precio de adquisición no es muy alto. Este dispositivo contiene características superiores a distintas cámaras convencionales lo que lo lleva a ser una buena opción para convertirse en un sistema de video vigilancia.

Los sistemas de video vigilancia presentan soluciones para los problemas de inseguridad en casa-habitación, sin embargo como son planteados anteriormente presentan ciertas deficiencias las cuales llegan a ser perjudiciales para las personas, por ello se crean numerosos sistemas de video vigilancia ofreciendo funcionalidades diferentes a las de la competencia, características únicas y garantías de una mayor seguridad que las de sus competidores. Pero esto resulta ser excesivamente caro por todos los componentes que involucran y el mantenimiento de dichos sistemas.

Es por esto que el proyecto actual presenta un sistema de video vigilancia que da soluciones para disminuir los índices de inseguridad en casa-habitación y sea económico para las personas, abarcando aspectos básicos y combinar las distintas funcionalidades de varios componentes en un solo sistema de video vigilancia. El sistema "Aplicación del Kinect para la prevención de allanamientos en casa-habitación" combina las distintas características de las cámaras DVR, las cámaras IP, sensores de movimiento y el dispositivo Kinect, logrando el reconocimiento y detección de personas dentro de un escenario preestablecido mediante un esqueleto lineal y una asignación de color única, una alarma sonora, captura de imágenes y grabación de video con copia a el correo electrónico del usuario.

El proyecto actual utiliza el sistema operativo de Linux distribución Ubuntu 12.04 por su bajo consumo de memoria RAM y velocidad de procesamiento, por ende es accesible a compra y consigue ser un Software óptimo para ser empleado como sistema de vigilancia. Su característica de detección de personas es única en cuanto a los demás sistemas de vigilancia en el mercado se refiere, esto se debe al sensor de movimiento que trae integrado el dispositivo Kinect, en cuanto un intruso entra en el escenario vigilado por el Kinect rápidamente es detectado mediante el esqueleto lineal y su código de color, la detección no ocurre en animales logrando una mayor vigilancia y eliminando las alarmas falsas que ocurren con otros dispositivos, también se cuenta con una alarma sonora que es disparada para alertar al usuario en el momento que un intruso es detectado, se realizan una serie de foto capturas de el intruso y se comienza una grabación en video del ilícito, con copias de las foto capturas para almacenarlas en la memoria del sistema y posterior mente enviarlas al correo electrónico del usuario brindando una mayor seguridad, todo esto en un corto tiempo.

El sistema también demuestra una alta capacidad de monitoreo en tiempo real con una

calidad de imagen que se compara a las cámaras de seguridad convencionales, por su componente de cámara RGB (Red, Green, Blue) y su sensor de profundidad. El sistema convierte al dispositivo Kinect en una cámara de video vigilancia multifuncional con la ayuda del Software libre Linux y librerías dedicadas para adaptar en el sistema operativo libre o superando a las cámaras convencionales, mediante la manipulación de imágenes y video en tiempo real así como la detección de intrusos.

Las pruebas realizadas al sistema fueron dentro de ciertos parámetros los cuales comprende en una casa-habitación con medidas de 6.90 mts. De ancho, 2.40 mts. De alto y 3.90 mts. De largo; el Kinect fue puesto en un lugar estratégico, abarcando un rango de visión de 5 metros a partir de su posición dentro de un escenario principal para la prevención del allanamiento. analizado las cámaras convencionales para el uso de seguridad dentro de una casa-habitación, se observa que el empleo del dispositivo Kinect para un sistema de seguridad en una vivienda o negocio es muy óptimo, dado las diversas ventajas que este proporciona, tales como un precio accesible, fácil instalación, no necesita mantenimiento el dispositivo a diferencias de las cámaras IP y las DVR, que a pesar de ser las más populares dentro del mercado, Kinect muestra una buena mejora para esta rama.

Para el desarrollo del proyecto se usa metodología RUP la cual nos ayuda disciplinadamente asignar tareas y responsabilidades, dicha metodología logra que cumplan con las tareas asignada en tiempo y forma. Mediante la metodología RUP el proyecto hace uso de un estudio observacional que es percibir, detectar, mirar precisa y detenidamente en orden y de forma detallada. Para poder cumplir con los propósitos del proyecto, es indispensable observar si los accesos dentro de la casa o negocio son suficientemente seguros para evitar el allanamiento de personas externas.

Se determina que la naturaleza del proyecto es de forma experimental, es decir, un investigador manipula conscientemente las condiciones de una o de diversas situaciones precedentes, ya que es necesario confrontar la solución propuesta a su campo de acción real requiriendo una contrastación entre la misma y la solución que se implementa en la actualidad, evidenciando su grado de funcionalidad.

Para llevar a cabo un análisis más a fondo, también se mide la seguridad en distintos escenarios junto con diferentes posiciones estratégicas de la cámara Kinect y técnicas de intrusión controlados previamente, gracias a los cuales se pueden dejar al descubierto vulnerabilidades y fortalezas del proyecto. Además, se hace uso del estudio comparativo de efecto-causa que analiza los robos en vista a su prevención y detectar factores recurrentes con el fin de controlar o eliminar los riesgos en su misma fuente.

"La inseguridad ciudanía es un problema grave y la tecnología nos brinda un apoyo para reducirla de manera drástica, pero esta debe ser accesible para todas las personas". Por ende el sistema "Aplicación del Kinect para la prevención de allanamientos en casa-habitación" aplica las mejores cualidades de distintos sistemas de seguridad, así como diferentes dispositivos de seguridad proporcionando un sistema de alta gama en video vigilancia. Las principales características que cubre el sistema son:

1. *Detecta a una o varias personas dentro de un escenario establecido.*
2. *Asigna un color específico a la persona identificada dentro de un escenario establecido.*
3. *Traza un esqueleto lineal a la persona identificada dentro de un escenario*

establecido.

4. *Toma de fotografías.*
5. *Empieza la grabación cuando se detecte a una persona.*
6. *Dispara una alarma cuando se detecte a una persona dentro de un escenario establecido.*
7. *Discrimina animales y objetos que no estén establecidos dentro del área establecida.*
8. *Envió de fotografías al correo personal del usuario.*

Haciendo uso del Kinect y el sistema se puedo realizar una simulación en tiempo real donde el sujeto fue detectado cuando entro al escenario preestablecido dentro de la casa-habitación gracias al esqueleto lineal y la asignación de color en el sujeto de pruebas por medio del sistema y los componentes del Kinect. Su entorno de desarrollo fue mediante el software libre, que nos permite utilizar librerías existentes para la manipulación completa del Kinect, lo que nos proporciona un sistema de video vigilancia bastante completo y de bajo costo, que es adecuado para personas de clase media a media baja; las cuales pueden implementar sistemas o dispositivos de seguridad para la protección de sus patrimonios sin realizar gastos excesivos.

Por otra parte el uso del Kinect y tecnologías de software libre no sólo se limitan a aplicaciones en el área de la seguridad dentro de la casa - habitación, también se puede implementar en diferentes áreas tales como educación, medicina, etc… esto nos brinda la oportunidad de abrir nuevos proyectos de investigación para poder dar soluciones a problemas en la vida real.

Referencias

CodelcoEduca. La observación y el método científico. [En línea]. Disponible en: https://www.codelcoeduca.cl/biblioteca/naturales/1_naturales_NB5-7B.pdf

Alonso Serrano, A., García Sanz, L., León Rodrígo, I., García Gordo, E., Gil Álvaro, B. & Ríos Brea, L. Métodos de investigación de enfoque experimental. [En línea]. Disponible en: http://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso_10/Experimental.pdf

Díaz Flores Milagros, Flores. (2010). Metodología Rational Unified Process (RUP). [En línea]. Disponible en: http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf

Instituto Nacional de Estadística y Geografía (INEGI) [En línea]. Disponible en:http://www.inegi.org.mx/inegi/contenidos/espanol/prensa/boletines/boletin/comunicados/especiales/2013/diciembre/comunica9.pdf

Construcción alterna de los diagramas de Voronoi empleando cómputo paralelo GPU

Netz Romero, Ricardo Barrón
Centro de Investigación en Computación del Instituto Politécnico Nacional

Resumen

Los diagramas de Voronoi son empleados en diversas áreas, como: problemas geométricos, construcción de modelos, sistemas de información geográfica, planeación de rutas entre otras más. Generalmente para construirlos computacionalmente se parte de su estructura dual que es la triangulación Delaunay. Ahora para generar las regiones de Voronoi partiremos de un concepto diferente llamado el método de la "Esfera Hueca", el cual no consiste en la generación de triángulos, pero empleará una de sus propiedades fundamentales llamada el "Círculo Vacío" publicado en uno de los trabajos de Boris Deloné. Lo más importante de este trabajo será incluir el cómputo paralelo para el diseño de los diagramas de Voronoi, en donde se explican los algoritmos y procedimientos necesarios para llevar a cabo su construcción. La herramienta necesaria para llevar a cabo el paralelismo en los algoritmos, será el modelo de programación CUDA, por lo que de esta manera se aprovecharán las capacidades de operaciones del GPU. Estas características nos permitirá el procesamiento de grandes volúmenes de datos, los cuales se utilizarán en la generación de los diagramas de Voronoi con tiempos deseados y sean posibles de aplicar en los campos de la ciencia y la ingeniería.

Palabras claves. diagramas de Voronoi, triangulación Delaunay, Esfera Hueca, Círculo Vacío, CUDA y GPU.

Referencias

1. Aurenhammer, F., Klein, R.: Voronoi Diagrams, Handbook of Computational Geometry, Elsevier Science, Amsterdam, pp. 201–290, 2000.
2. Delone , B. N.: Sur la sphere vide, Bulletin of the Academy of Sciences of the U. S. S. R., no. 6, 793-800, 1934.
3. Fischer, I., Gotsman, C.: Fast approximation of high order Voronoi diagrams and distance transforms on the GPU. Journal of Graphics Tools 11, 4, pp 39–60, 2006.
4. Guodong, R., Tiow-Seng, T., Thanh-Tung, C.: Computing two-dimensional Delaunay triangulation using graphics hardware. Proceedings of the 2008 symposium on Interactive 3D graphics and games, pp 89-97, 2008.
5. NVIDIA CUDA Compute Unified Device Architecture - Programming Guide version 1.1, November 2007.
6. Osami, Y.: An acceleration technique for the computation of Voronoi diagrams using graphics hardware. Computational Science and Its Applications - ICCSA 2005, pp 786-795, 2005.
7. Traversoni, L.: Interpolación con esferas de cobertura., Tesis Doctoral en la UAM Iztapalapa, 1993.
8. Zhan, Y., Guodong, R., Xiaohu, G, Wenping, W..: Generalized Voronoi Diagram Computation on GPU, Eighth International Symposium on Voronoi Diagrams in Science and Engineering, pp 75 - 82, 2011.

El Álgebra Lineal del PageRank

Ernesto Ortega Trujillo, Salvador Botello Rionda
Centro de Investigación en Matemáticas A.C., Jalisco S/N, Col. Valenciana, C.P. 36240, Guanajuato, Gto., México, email: ortegae, botello@cimat.mx, web: www.cimat.mx

Resumen

Los motores de búsqueda empleados para encontrar en internet sitios web con la información relevante para el usuario, se componen de lo siguiente: 1. Bots que buscan en internet páginas con acceso público; 2. un catálogo que contiene las páginas que han sido indexadas por el bot y 3. un algoritmo para determinar la importancia de cada página del catálogo.

Uno de los buscadores que ha sobresalido es el de la empresa Google que como se menciona en la referencia [1] se distinguió de los demás porque sus búsquedas arrojaban los "buenos resultados". Gran parte de este éxito se debe a su algoritmo de rankeo conocido como PageRank el cual determina la importancia de una cierta página principalmente en función de la importancia de las otras páginas que se conectan a ella y del número de links salientes de cada una de estas. El problema para poder hacer el rankeo consiste en la solución de un problema de valores y vectores propios generalmente para matrices de grandes dimensiones y muy ralas.

En este trabajo se presentan comparaciones de tiempos de ejecución empleando el método de la potencia tradicional y una forma adaptativa de éste [2]. Las matrices se trabajan con el formato renglón comprimido y dado el tamaño de los sistemas a resolver se paralelizó el algoritmo usando el esquema OpenMP, el cual permite reducir estos tiempos de manera considerable.

Palabras clave: *PageRank, eigenvectores, eigenvalores, paralelización, páginas web.*

Referencias

[1] Kurt Bryan and Tanya Leise, "The $25,000,000,000 Eigenvector. The Linear Algebra behind Google".

[2] Kristen Thorson, "Modeling the Web and the Computation of PageRank", April 2004.

In Situ Crowd Simulation and Visualization

Hugo Pérez[1], Benjamín Hernández[2], Isaac Rudomin[2]
[1]*Universidad Politécnica de Catalunya, Barcelona, Spain*
[2]*Barcelona Supercomputing Center, Barcelona, Spain*

Abstract

Multi-GPU clusters are suitable for in situ visualization applications, e.g. when significant computational power for concurrent execution of simulation and visualization is required [1]. Pipelined parallelism can be used to overlap the execution of simulation and rendering among multiple GPUs [2, 3], but is sensitive to load imbalance that can significantly impact performance [4]. The aim of this paper is to analyze crowd simulation and visualization under different system architectures: simulation on slaves, rendering on master; simulation and rendering on slaves; simulation on cluster and remote visualization; in order to optimize the performance for each particular resources availability. Scaling, data copying and load balancing are also discussed.

Finally, we present experimental results obtained in the Barcelona Supercomputing Center's Minotauro cluster using CUDA 5.

References

[1] Fogal, T., Childs H., Siddharth S., , Krüger J., Bergeron R., and Hatcher P., "Large Data Visualization on Distributed Memory Multi-GPU Clusters". High Performance Graphics (2010)

[2] D. Jacobsen, J. C. Thibault, and I. Senocak."An MPI-CUDA Implementation for Massively Parallel Incompressible Flow Computations on Multi-GPU Clusters". In American Institute of Aeronautics and Astronautics (AIAA) 48th Aerospace Science Meeting Proceedings (2010)

[3] Kirk, David B. and Hwu, Wen-mei W. "Programming Massively Parallel Processors: A Hands-on Approach". Morgan Kaufmann Publishers Inc. (2010)

[4] Yong Kao, Robert Hagan, "Multi-GPU Load Balancing for In-Situ Simulation and Visualization". Journal of Supercomputing (2013)

Securities System at Banco de México, High Availability Architecture for A Mission-Critical Business

Blanka Selenne Berthely García, Carlos Pérez Leguízamo
Banco de México Av. 5de Mayo #2, Mexico City, 06059, MEXICO
sberthely@banxico.org.mx, cperez@banxico.org.mx

Abstract

Banco de México (Banxico) is the Central Bank of Mexico and its daily critical mission is to provide a sound development of the country's economy through payment systems. Thus, Banxico convey operations in the securities market with several financial institutions, interacting with different systems and beware of the following requirements: immediate response to their requests, a 24/7 operation, and an effective solution to the continual changes of the market. Accordingly, Banxico requires a system that can keep track of each security's life cycle and characteristics like high availability, online management, and timeliness for giving immediate response, is needed. Today, the existing system has developed several difficulties in its daily operation due to the business changing nature; consequently, the bank is working in a new one that can support low impact changes, an efficient use of the communication resources to ensure the assertiveness of the information so to avoid wrong financial decisions. All these considerations can be overcome and controlled by the implementation of an architecture based on Services that can bear with adjustments to the code without affecting the one that is already tested. The architecture also implements a Data Field and a protocol to ensure message delivery, and data integrity. Finally, the architecture by means of using different replicas, and an auto recovery mechanism, guarantees availability that safeguard the general status of the system. This paper will contribute to show the benefits and opportunity areas in the implementation of ADSOA (Autonomous Decentralized Service Oriented Architecture). Moreover, the potential use for future projects at the bank, making a wise use of this architecture.

References

[1] Mori K. et al, 1984, "Proposition of Autonomous Decentralized Concept", Journal of IEEE Japan, Vol. 104, No. 12, pp. 303-310. **[2]** Mori K, 1993, "Autonomous Decentralized Computer Control Systems", First International Symposium on Autonomous Decentralized Systems (ISADS'93), Kawasaki, Japan, pp. 28-34. **[3]** Catherine Mansell Carstens, 1992, "Las Nuevas Finanzas en México", Ed. Milenio. **[4]** Alan S. Blinder, 1999, "El banco central: teoría y práctica", Ed. Antoni Bosch. **[5]** Coronado-García L. C. et al, 2011, "An Autonomous Decentralized Service Oriented Architecture for High Reliable Service Provision", The 10th International Symposium on Autonomous Decentralized Systems (ISADS'11), Kobe, Japan. **[6]** Thomas Erl, 2005, "Service-Oriented Architecture (SOA): Concepts, Technology, and Design", Ed. Prentice Hall. **[7]** García-Gomez R. et al, 2013, "Autonomous Recovery Technology for Fault Tolerance in Distributed Service-Oriented Mission Critical Systems", The 4th International Supercomputing Conference in Mexico (ISUM'13), Manzanillo, México.

High Reliable Graphic User Interface for Mission Critical Systems

Raymundo García Gómez, Juan Sebastián Guadalupe Godínez Borja,
Pedro Josué Hernández Torres, Carlos Pérez-Leguízamo
Banco de México Av. 5de Mayo #2, Mexico City, 06059, MEXICO
rgarciag@banxico.org.mx, jgodinez@banxico.org.mx, pjhernan@banxico.org.mx, cperez@banxico.org.mx

Abstract

Mission Critical Systems (MCS) requires continual operation, due to fails which can cause fatal losses, such as human lives or a huge economic damage. The Autonomous Decentralized Service Oriented Architecture (ADSOA) was proposed as an approach to design and to implement MCS. This architecture provides a framework to develop business processes; nevertheless it does not have a model to design a Graphic User Interface (GUI) which works with MCS. This model should take into consideration the possibility of offline work, as well as guarantying data consistency, in order to improve the response time and make sure that online management does not interrupt the operation of its graphic services. Today, many GUI technologies do not allow development of the kind of systems with the features mentioned above. Accordingly to this absence we propose an architecture that uncouples deployment and functionality through graphic services. These services are loading on demand in order to optimize the computational resources, besides persistence data is guaranteed by a local repository. Finally, it is proposed a technology to manage online graphic services which are sent by a communication bus where data is highly efficient transported by its size. The feasibility of this proposal is demonstrated by an implemented prototype.

References

[1] Mori K. et al, 1984, "Proposition of Autonomous Decentralized Concept", Journal of IEEE Japan, Vol. 104, No. 12, pp. 303-310.

[2] Mori K, 1993, "Autonomous Decentralized Computer Control Systems", First International Symposium on Autonomous Decentralized Systems (ISADS'93), Kawasaki, Japan, pp. 28-34.

[3] Coronado-García L. C. et al, 2011, "An Autonomous Decentralized Service Oriented Architecture for High Reliable Service Provision", The 10th International Symposium on Autonomous Decentralized Systems (ISADS'11), Kobe, Japan.

[4] Thomas Erl, 2005, "Service-Oriented Architecture (SOA): Concepts, Technology, and Design", Ed. Prentice Hall.

[5] García-Gomez R. et al, 2013, "Autonomous Recovery Technology for Fault Tolerance in Distributed Service-Oriented Mission Critical Systems", The 4th International Supercomputing Conference in Mexico (ISUM'13), Manzanillo, México.

[6] Craig Buckler, 2013, "HTML5 Browser Storage: The Past, Present and Future", http://www.sitepoint.com/html5-browser-storage-past-present-future/

Simulación de una EDP-Hiperbólica usando un AC 2-dimensional

I. Huerta-Trujillo1, J. C. Chimal-Eguía[1], N. Sánchez-Salas[2] and J.A. Martinez-Nuño[3]

[1]*Centro de Investigación en Computación, Instituto Politécnico Nacional, UP Adolfo López Mateos, México, D.F.*

[2]*Escuela Superior de Física y Matemáticas, Instituto Politécnico Nacional, UP Adolfo López Mateos, México, D.F.*

[3]*Escuela Superior de Computo, Instituto Politécnico Nacional, UP Adolfo López Mateos, México, D.F.*

Resumen

En este trabajo se propone un modelo paralelo basado en un Autómata Celular (AC) 2-dimensional, así como el proceso de obtención de la regla de evolución del mismo, el modelo obtenido se compara contra la solución analítica de una ecuación diferencial parcial de tipo hiperbólica de dos dimensiones, lineal y homogénea, la cual modela una membrana vibrante con condiciones iniciales y de frontera específicas, se analiza el espectro de frecuencia así como el error entre los datos arrojados por el modelo de AC contra los datos proporcionados por la evaluación de la solución a la ecuación diferencial.

Por otro lado se analiza la complejidad del AC y se justifica la implementación del modelo usando cómputo paralelo a fin de disminuir la complejidad computacional.

References

[1]Ilachinski, A. "Cellular Automata: a discrete universe". World Scientific, 2001.

[2] J. Johannes and R. Shannon. "Systems simulation: the art and science". IEEE Transactions on Systems, Man and Cybernetics, vol. 6(10):pp. 723–724., 1976.

[3] P. Fishwick. "Simulation Model Design". Proceeeding of the Winter Simulation Conference, 1995.

[4] B. Chopard and et al. "Cellular Automata Modeling of Physical Systems". Cambridge University Press, USA, 1998.

[5] T. Toffoli. "Occam, Turing, von Neumann, Jaynes: How much can you get for how little? (A conceptual introduction to cellular automata)". InterJournal, MIT, December 1994. Cambridge, MA.

[6] S. Motta and V. Brusic. "Mathamatical Modelling of the Immune System". Springer, USA, 2004.

[7] S. Wolfram. "cellular automata as models complexity". Nature, (311):419–424, 1984.

[8] R. Espericueta. "Cellular Automata Dynamics Explorations in Parallel Processing". Bakersfield College, USA, 1997.

[9] S. Wolfram. "statistical mechanics of cellular automata". Reviews of Modern Physics, (55):601–644, 1983.

[10] Y. Bar-Yam. "Dynamics of Complex Systems". Addison-Wesley, USA, 1997.

[11] M. Mitchell. "Computation in cellular automata: A selected review". Nonstandard Computation, pages 95–140, 1998. Weinheim: VCH Verlagsgesellschaft.

[12] S. Hoya and et al. "Modeling epidemics using cellular automata". Applied Mathematics and Computation, 2006. doi:10.1016/j.amc.2006.06.126.

[13] Norman Margolus. Physics-like models of computation. Physica D: Nonlinear Phenomena, 10(1–2):81 – 95, 1984.

[14] L.E. Kinsler. Fundamentals of acoustics. Wiley, 2000.

[15] H. P. Hsu. "Análisis de Fourier". Prentice Hall, México, 1998.

[16] Maarten Van Walstijn and Eoin Mullan. Time-domain simulation of rectangular membrane vibrations with 1-d digital waveguides. pages 449–454, 2011.

[17] Huerta-Trujillo I. et. al. Modelo de autómata celular 1-dimensional para una edp hiperbólica. Research in Computing Science, 58(ISSN: 1870-4069):407 – 423, November 2012.

Un método de integración numérica en paralelo basado en el algoritmo de Particle Swarm Optimization

Mario Iván Jaen-Márquez, Arturo Hernández-Aguirre, Salvador Botello-Rionda
Centro de Investigación en Matemáticas, A. C.,
Jalisco S/N, Col. Valenciana, Zip 36240, Guanajuato, Gto. M´exico.
{mario.jaen,artha,botello}@cimat.mx, http://www.cimat.mx

Resumen

La integración numérica juega un rol importante en áreas como ciencias de la computación e ingeniería cuando se trabaja con funciones de las cuales se conoce su expresión analítica pero es difícil (o imposible) encontrar su antiderivada. Para ello se han propuesto métodos que hacen más preciso el cálculo aproximado de integrales definidas. En general, la forma tradicional de integraci´on numérica consiste en construir un polinomio de interpolaci´on utilizando puntos equiespa-ciados dentro del intervalo para aproximar la función a integrar. Los métodos de Riemann, Trapecios, Cuadratura de Gauss y las f´ormulas de Newton-Cotes están basados en esta idea. Sin embargo, un mejor método de integración debería utilizar puntos espaciados a distancias variables dentro del intervalo de acuerdo a la forma de la función misma, con el objetivo de reducir una cota del error y obtener mayor precisión.

En este trabajo se plantea el uso del algoritmo Particle Swarm Optimization (PSO) para elegir la partición que subdivide el dominio de integración y que mejor aproxima el valor de la integral de la función dada minimizando la cota del error. Cada partícula del swarm se mueve dentro de un espacio de búsqueda N-dimensional utilizando las reglas definidas en el PSO, siendo N el número de puntos de la partición. Leila Djerou et al. presentan un enfoque similar utilizando el PSO en serie para resolver integrales en una dimensión [1], Juan Xie et al. proponen un método de integración basado en el algoritmo de Artificial Bee Colony (ABC) [2], Li Bei propone una Estrategia Evolutiva (ES) basada en las características de un Sistema Inmune Artificial (AIS) y la utiliza para calcular numéricamente la integral de una función [3].

Para el cálculo de integrales en varias dimensiones de funciones complejas la evaluación de la función de aptitud llega a ser muy costosa, por lo que el tiempo de cómputo se incrementa notablemente si no se utilizan técnicas de Cómputo de Alto Rendimiento (HPC), por ello se propone un esquema en paralelo del método usando un paradigma de memoria compartida, implementado con OpenMP.

Palabras clave: *Integración Numérica, Particle Swarm Optimization, PSO, Cómputo Paralelo, OpenMP*

Referencias

1. Leila Djerou, Naceur Khelil, andMohamed Batouche. Numerical integration method based on particle swarm optimization. In Advances in Swarm Intelligence, volume 6728 of Lecture Notes in Computer Science, pages 221–226. Springer Berlin Heidel- *berg, 2011.*

2. Juan Xie and Jianfeng Qiu. A novel numerical integration method based on artificial bee colony algorithm. In Control Engineering and Communication Technology (ICCECT), 2012 International Conference on, pages 531–534, 2012.

3. Li Bei. Research on the evolutionary strategy based on ais and its application on numerical integration. In Ran Chen, editor, Intelligent Computing and Information Science, volume 135 of Communications in Computer and Information Science, pages 183–188. Springer Berlin Heidelberg, 2011.

A Comparison of the convergence and performance of parallel Monte Carlo integration method on n-dimensional probability functions.

Jonathan Rafael-Patiño, Salvador Botello.
Centro de Investigacion en Matematicas A.C.,
Jalisco S/N, Col. Valenciana, Zip 36240,
Guanajuato, Gto., Mexico,
email: jonhrafe, botello web: www.cimat.mx,

Abstract

The Monte Carlo numerical methods are used in a variety of problems in science. These methods relay on a large number of random samples to obtain numerical results. Since Monte Carlo methods do not require an explicit analytical expression, it has become a very popular method in practice to explore complex and high-dimensional systems.

Compute-intensive features within multidimensional numerical integration methods requires a highly ecient implementation. The rapid growth of Graphics Processing Units (GPU's for short) have allowed the implementation of computationally ecient methods for various areas of scientic computing. Using a number of optimization techniques for implementation within a GPU, is possible to achieve a high eciency, which is maintained despite of the increase in the dimensionality.

This paper presents a series of algorithms implemented in parallel for the Monte Carlo integration problem, applied to n-dimensional probability functions. Given the high-paralleling characteristics of the problem, we takes advantage of the possibility of a parallelization based on multiple threads (CUDA) and multiple processors (OpenMp). Within the results, we used several algorithms in order to generate parallel random samples for the comparison in the eciency and accuracy of the results. The algorithms were implemented in the CUDA programming language and benchmarked under the exibility of various parameters as the number of threads, selection of the memory interaction and the random samples generating functions. Eciency results are presented for the numerical accuracy and execution time. Also a comparisons of the convergence (average number of samples required) to the expected numerical result is presented.

Keywords: *Monte Carlo, Integration, Paralell Algorithms, CUDA, OpenMp, Performance.*

Resumen

Los metodos de aproximacion numerica por Monte Carlo son utilizados en una gran variedad de problemas en la ciencia. Estos metodos dependen de un gran numero de muestreos aleatorios para obtener resultados numericos. Dado que los metodos Monte Carlo no requieren una expresion analtica explcita del sistema, estos se ha convertido en un metodo muy popular en la

practica para explorar sistemas complejos y en alta dimension.

Las caractersticas intensivas de computo dentro de los metodos de integracion numerica multidimensional requieren de implementacion altamente eficiente. El rapido crecimiento de las unidades graca de procesamiento (GPU's por sus siglas en ingles) han permitido la implementacion de metodos computacionalmente ecientes para diversas areas del computo cientco. Usando varias técnicas de optimizacion para una implementacion dentro de una GPU es posible alcanzar una gran eciencia la cual se mantiene a pesar del incremento de dimension del problema.

En este trabajo se implementan una serie de algoritmos en paralelo para la integracion por Monte Carlo, aplicado a funciones n-dimensionales de probabilidad. Aprovechando las caractersticas del problema se toma ventaja de la posibilidad de una paralelizacion basada en multiples threads (CUDA) o multiples procesadores (OpenMp). Dentro de los resultados obtenidos, se utilizaron varios algoritmos de generacion de muestras aleatorias en paralelo para su comparacion en la eciencia y precision de los resultados. Los algoritmos fueron implementados en el lenguaje CUDA con exibilidad en los parametros como el numero de threads, seleccion de la interaccion de la memoria así como para las funciones de generacion de muestras aleatorias.

Se presentan resultados de eciencia en la precision numerica, as como el tiempo de ejecucion, ademas de comparaciones en la convergencia (numero de muestras promedio necesarias) al resultado numerico esperado.

Palabras clave: *Monte Carlo, Integracion, Algoritmos en Paralelo, CUDA, OpenMp, Performance.*

References

Alerstam E, Svensson T, Andersson-Engels S; Parallel computing with graphics processing units for high-speed monte carlo simulation of photon migration. J. Biomed. Opt. 0001;13(6):060504-060504-3.

K. Esselink, L. D. J. C. Loyens, and B. Smit Shell Research B.V., Koninklijke/Shell-Laboratorium, Amsterdam, P.O. Box 38000, 1030 BN Amsterdam, The Netherlands

Análisis de Desempeño de una Grid Móvil Ad Hoc

Anabel Pineda Briseño[1], Manuel Alejandro Soto Ramos[2], Miguel Rodríguez Martínez[3], Rolando Menchaca Méndez[3]
[1]Departamento de Sistemas y Computación, ITM., [2]Escuela Superior de Cómputo, IPN.
[3]Lab. de Comunicaciones y Redes de Computadoras, CIC-IPN.
E-mails: anabel.pineda@itmatamoros.edu.mx, msotoa06@yahoo.com.mx,
miguelonrm1@gmail.com, rmen@cic.ipn.mx

Resumen

Los sistemas de cómputo actuales permiten integrar diversas tecnologías para la obtención de recursos de cómputo de altas prestaciones. Asimismo, los avances en cómputo en grid han permitido el procesamiento de millones de datos en un tiempo menor debido a la distribución de la carga de trabajo entre sus elementos que lo conforman.

De forma paralela, el crecimiento del número de dispositivos móviles y sus capacidades de cómputo han permitido la comunicación y el intercambio masivo de información entre ellos. Considerando los aspectos anteriores, se describe una propuesta de infraestructura de cómputo grid sobre una red móvil ad hoc para la distribución de carga de procesamiento y los algoritmos requeridos para el descubrimiento de recursos, el envío de las tareas y la recolección de los resultados para su entrega.

El diseño de grid que se propone en este trabajo utiliza un control centralizado con un planificador de tipo Fisrt Come First Serve (FCFS) para la distribución de la carga. Se utiliza un algoritmo de descubrimiento de nodos disponibles para conformar la grid utilizando un modelo de propagación y de movilidad 2D. Los experimentos son basados en simulaciones a nivel capa de red y los resultados obtenidos consideran la utilización de los recursos grid, el tiempo de finalización máximo de la carga de trabajo, el porcentaje de tareas realizadas satisfactoriamente y la sobrecarga.

Referencias

[1] Andrei Tchernykh, Uwe Schwiegelshohn, Ramin Yahyapour, and Nikolai Kuzjurin. On-line hierarchical job scheduling on grids with admissible allocation. J. Scheduling, 13(5):545–552, 2010.

[2] Ian Foster and Carl Kesselman. The Grid 2. Blueprint for a New Computing Infrastructure. The Elsevier Series in Grid Computing. Morgan Kaufmann, 2nd edition, 2004.

[3] J. MacLaren. Advance reservations: State of the Art. In Global Grid Forum, volume 9, 2003.

[4] R. Baldoni, R. Beraldi, and A. Mian. Survey of service discovery protocols in mobile ad hoc networks. Technical Report MidLab 7/06, Dip. Dip. Informatica e Sistemistica, Universita di Roma, 2006.

[5] Zhuoqun Li, Lingfen Sun and Emmanuel C. Ifeachor. Challenges of mobile ad hoc grid and their application in e-healthcare. University of Plymouth, UK.

[6] Zhi Wang, Bo Yu, Qi Chen, Chunshan Gao. Wireless Grid Computing over Mobile Ad Hoc Networks with Mobile Agents. 0-7695-2534-2/05 IEEE 2006.

[7] Antonio Tadeu A. Gomes, Artur Ziviani, DICHOTOMY: A Resource Discovery and Scheduling Protocol for Multihop Ad Hoc Mobile Grids. In Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGRID), pages 719-724, 2007.

[8] V. Hamscher, U. Schwiegelshohn, A. Streit, R. Yahyapour. Evaluation of Job-Scheduling Strategies for Grid Computing
In Proc. of 1st IEEE/ACM International Workshop on Grid Computing (Grid 2000) at 7th International Conference on High Performance Computing (HiPC-2000), Bangalore, India, LNCS 1971, Springer, 2000, pages 191-202, ISBN 3-540-41403-7

[9] The network simulator - NS-2 Available: http://www.isi.edu/nsnam/ns

Straightforward DSP algorithm suitable for GPU computation

Raúl G. Hazas-Izquierdo [a]., José L. Rodríguez-Navarro [b]., Alan D. Faz Gutiérrez [c]., Raymundo Salazar Orozco [c].

[a] *Universidad de Sonora*

[b] *Centro de Investigación Científica y de Educación Superior de Ensenada*

[c] *Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Sonora Norte*

Abstract

Current Graphic Processing Units (GPUs) have achieved a deep level of parallelism. Filtering of discrete sequences often requires an inordinate amount of computing resources. Infinite Impulse Response (IIR) filter structures often are broken down into simpler and concatenated elements called biquads for ease of implementation. Analysis of data flow across one such simplified structure, prompts of feasible digital signal processing (DSP) algorithm simplification. Favorable comparison of outcome brought up by forthright C-based implementation of prospective DSP algorithm versus industry standard application suggests a filtering method likely adaptable for CUDA coding and execution.

Keywords: *DSP, Filtering, GPU, CUDA Programming.*

References:

Matei, R., Two-Dimensional IIR Filter Design Using Prototype Biquad Transformation, 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2010, pp. 375 – 378.

Monsurro, P. et. al., Design Strategy for Biquad-Based Continuous-Time Low-Pass Filters, 20th European Conference on Circuit Theory and Design (ECCTD), 2011, pp. 385 – 388.

Kuo, Sen M., Bob H. Lee, Wenshun Tian, Real-Time Digital Signal Processing: Implementations and Applications, Wiley, 2nd. Edition, 2006, pp. 258-262.

Bozejko, W. et. al., Parallelizing of digital signal processing with using GPU, Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings (SPA), 2010, pp. 29 – 33.

Paralelización de filtro BSS/WSS sobre GPGPU para la clasificación de subtipos de cáncer con MVS

Iliana Castro-Liera, J. Enrique Luna-Taylor, Jacob E. Merecías-Pérez, Germán Meléndrez-Carballo
Departamento de Sistemas y Computación, Instituto Tecnológico de La Paz,
Boulevard Forjadores No. 4720, La Paz, B.C.S., México;
icastro@itlp.edu.mx, eluna@itlp.edu.mx

Resumen

En este trabajo proponemos la paralelización sobre GPGPU del filtro BSS/WSS (Between Sum Square-BSS, Within Sum Square-Wss), el cual trabaja con datos de expresión genética, para seleccionar genes relevantes estadísticamente. Posteriormente aplicamos una MVS (Support Vector Machine) sobre los datos de expresión de los genes seleccionados por el filtro, para realizar diagnósticos de cáncer. Los resultados obtenidos han alcanzado un porcentaje de éxito del 92% en el diagnóstico; y con la implementación en paralelo sobre GPGPU utilizando tecnología CUDA (Compute Unified Device Architecture), se ha logrado una reducción del tiempo de ejecución de aproximadamente 18 veces con respecto a la implementación secuencial sobre CPU.

Palabras claves: *expresión genética, MVS, programación paralela, GPGPU, CUDA.*

GPU's molecular dynamics simulation of a one million particles

José María Zamora and Pedro D. Cruz-Santiago
Lufac Computación, México, D.F., México, Email: josema zamora@lufac.com
Enrique Díaz-Herrera
Departamento de Física, Universidad Autónoma Metropolitana, México, D.F., México, Email: diaz@xanum.uam.mx
Luis M. de la Cruz
GMMC, Depto. de Recursos Naturales, Instituto de Geofísica – UNAM Email: luiggi@unam.mx

Abstract

In this work the main algorithms involved in a molecular dynamics simulation are described. We present a strategy of parallelization using CUDA to accelerate the computations in GPUs. We show several examples of application of our implementations for 500, 2048, 10000 and 106 particles. We show very good results in terms of computation time and accuracy of the results.

Prototipo Aplicación Web Móvil para docentes ITESI utilizando la plataforma Twitter Bootstrap

José Antonio López Quiroz, Nelly Beatriz Santoyo Rivera, Daniel Jorge Montiel García.
Instuto Tecnológico Superior de Irapuato

Resumen

La tecnología es un área que evoluciona día con día de manera muy acelerada, en los últimos años los dispositivos móviles están transformando la forma en que los seres humanos nos comunicamos, estos dispositivos tienen capacidades similares a las de una computadora personal con la gran ventaja de que las podemos portar en nuestros bolsillos (Girones, 2012).

Este prototipo es un primer paso que permitirá a los sistemas de información de ITESI dar sus primeros pasos en tecnologías móviles, por este motivo solo se muestra un modelo parcial de la base de datos, similar al real ya que por políticas de seguridad no se da el acceso a la base de datos real o copias de la misma.

Para el desarrollo de este prototipo se utilizó la Tecnología Web para móviles de twitter bootstrap donde la interfaz del usuario se adapte a cualquier dispositivo móvil utilizando la característica "responsive design" del bootstrap o jQuery Mobile.

La aplicación le permite a los profesores consultar sus horarios y la información de seguimiento docente una vez que haya iniciado sesión en el sistema. Este prototipo permitirá a los docentes consultar y administrar las principales actividades de su quehacer docente como lo son:

- *Consulta de horarios de clase.*
- *Consulta de listas de alumnos de los grupos que tiene asignados.*
- *Consulta de Mensajes que recibe*
- *Envío de mensajes a alumnos*

Este proyecto se desarrollo tomando en cuenta conceptos de sistemas de información, que ayudan a la creación de un producto de software de calidad, el cual, satisfaga las necesidades del cliente en su parte funcional, así como también asegurarse que la construcción del sistema utilice métodos formales que aseguren que se construye un software utilizando mejores prácticas.

Bibliografía

Burgess, A. (2012). Getting Good Wtih PHP. USA: Rockable.

Girones, J. T. (2012). El Gran Libro de Android. Barcelona, España: Marcombo.

Otto, M. (s.f.). Bootstrap. Recuperado el 16 de Dic de 2013, de http://getbootstrap.com/

Senn, J. A. (1992). Análisis y diseño de Sistemas de Información. México: McGraw-Hill.

Aplicación Web para la Gestion de Datos Obtenidos por la Torre Meteorologica y el Muestreador de Particula Pm10 del Itslv

Javier Colorado Pérez, Armando Pérez Aguilar
Instituto Tecnológico Superior de Villa La Venta
Calle Circuito Tecnológico No. 1, Col. El Cuatro,
Villa la Venta Huimanguillo Tabasco, Cp. 86410

Resumen:

Este proyecto surge de la necesidad de recopilar, procesar y dar a conocer datos obtenidos por la torre meteorológica del ITSLV y el muestreador de partículas PM10. Para lograr lo anterior se propone la realización de una plataforma web que esté intercomunicada con la torre meteorológica y que pueda procesar los datos captados por ella y por el muestreador de partículas PM10. Así como analizar la información histórica generada, que mediante técnicas avanzadas de análisis de datos permita hacer inferencias o predicciones que ayuden al sector productivo de la región y que proporcione conocimiento científico de interés para los diferentes sectores de la población cercana, como académicos, estudiantes, investigadores e instituciones gubernamentales.

Referencias

1. Cobo, Á. (2005). PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. Díaz de Santos.

2. Coronel, C.; Rob, P; Morris, S (2011): Bases de datos: Diseño, implementación y administración, Cencage Learning.

3. Forouzan, B.A (2007), Transmisión de datos y redes de comunicaciones, McGraw-Hill

4. Han, J.; Kamber, M.(2006) Data mining: concepts and techniques, Morgan Kaufmann.

5. Lopez, Q.J (2007). PHP y MySQL: Programación Dinámica en el lado del servidor

6. Orallo, J.H.; Quintana, M.J.R.; Ramírez, C.F.(2004), Introducción a la Minería de Datos, Pearson Prentice Hall

7. Tanenbaum, A.S(2003). Redes de Computadoras. Pearson Prentice-Hall.

Metodo de Intercambio de Información Encriptada Utilizando Mecanismo de Lockers Dinamicos

Miguel Jesús Torres-Ruiz, Marco Antonio Moreno Ibarra, Edwin Alan Moedano Cardiel
Centro de Investigación en Computación, Instituto Politécnico Nacional
E-mail: mtorres@cic.ipn.mx, marcomoreno@cic.ipn.mx, edwinmoedano@gmail.com

Resumen

Se propone una metodología para el intercambio de información cifrada en un esquema de comunicaciones Cliente – Servidor para establecer canales de comunicación de mensajes cortos que son encriptados con base en algún algoritmo de cifrado simétrico por medio de la selección de un grupo de llaves dinámicas pseudo-aleatoria. El cifrado de los mensajes cortos es llevado acabo de manera dinámica y pseudo-aleatoria por un conjunto de llaves denominadas lockers que son generadas una vez establecido el canal de comunicación con el servidor para poder proporcionar un intercambio de información dinámica con respecto a las respuestas del servidor en las peticiones solicitadas, dado a la naturaleza de los algoritmos de cifrado simétrico que son rápidos y utilizan menos recursos que otras formas de cifrado.

Desempeño de los momentos de Hu implementados en GPGPU

Netz Romero[1], Migdaled López-Juárez[2], Omar Sánchez-Rodríguez[2]
[1]Universidad Autónoma Metropolitana - Cuajimalpa
[2]Centro de Investigación en Computación - Instituto Politécnico Nacional
netzrod, migdaled.lopez, osanchezrod@gmail.com

Resumen

Un método utilizado para la extracción de características invariantes en imágenes, es el de los momentos de Hu o momentos invariantes. Lo anterior implica que las características obtenidas no han de cambiar respecto a la traslación, rotación y escala. El problema observado es que existen proyectos donde se presenta el reconocimiento masivo y rápido de objetos en imágenes como por ejemplo en la medicina y visión por computadora, por lo tanto se requiere de una velocidad de cálculo que cumpla con el desempeño requerido. En vías de solucionar este problema se plantea la implementación de un algoritmo en paralelo que extraiga las características invariantes necesarias para el reconocimiento de los objetos en las imágenes procesadas. Aprovechando el potencial de las tarjetas gráficas de propósito general (GPGPU) en el procesamiento en paralelo se hace uso de ellas. Finalmente se comparara el rendimiento del algoritmo secuencial y el paralelizado.

Palabras clave: *Algoritmo en paralelo, desempeño, GPGPU, momentos de Hu, reconocimiento de objetos.*

Evaluación del Desempeño de Algoritmos Genéticos Celulares Sobre Gpu

Migdaled López-Juárez, Ricardo Barrón Fernández, Salvador Godoy-Calderón
Centro de Investigación en Computación Instituto Politécnico Nacional C.P. 07738
migdaled.lopez@gmail.com, barron2131@gmail.com, sgodoyc@gmail.com

Resumen

Esta investigación está orientada a la implementación de algunos modelos de paralelismo en los algoritmos genéticos celulares (CGA), sobre una GPU. Los algoritmos genéticos celulares parten de los algoritmos genéticos, teniendo como particularidad en la estructura de sus individuos el paralelismo inherente. Los individuos sólo pueden obtener información de los individuos vecinos. El problema que se desea resolver, es el encontrar un número óptimo de particiones para un conjunto de datos, empleando como función objetivo un índice de validación (como Davies-Bouldin, índice – I.) Resolver este problema es sumamente útil en algoritmos como el k-means, ya que es el punto de partida para determinar el número de representantes en cada partición o clase. Además de aprovechar los recursos de la GPU resolviendo el problema en menor tiempo que una CPU. Con esta investigación se pretende obtener modelos de paralelismo fijando como parámetros de desempeño la rapidez y eficiencia.

Palabras clave: *algoritmos genéticos celulares, paralelismo, GPU.*

Historia del supercomputo en México

Omar Sánchez-Rodríguez, I. Huerta-Trujillo, J.C. Chimal-Eguia, J. Figueroa-Nazuno
Centro de Investigación en Computación, Instituto Politécnico Nacional,
UP Adolfo López Mateos, México D.F. osanchezrod@gmail.com

Resumen

En el presente trabajo se hace un recopilación de la historia del supercomputo en México. El término supercomputo fue acuñado a principios del siglo XX y a grandes rasgos describe la capacidad que tiene una máquina para computar algoritmos en múltiples procesadores que operan de forma paralela sobre una cantidad masiva de datos. El supercomputo a lo largo de la historia ha dotado de una gran capacidad de cálculo a instituciones académicas, industriales e incluso comerciales y de gobierno en gran parte del mundo. En el ámbito académico ha sido utilizado para realizar entre otras cosas simulaciones en las áreas de la física, biología y matemáticas, por mencionar algunas. Lo anterior pone de manifiesto la importancia del supercomputo para las instituciones antesmencionadas asi como para los paises. Es ante este hecho que se realiza una investigación histórica de la importancia que ha tenido este paradigma, lo cual es de vital importancia para entender el sitio de México en el panorama mundial de las supercomputadoras y planear el futuro de estas en el país.

Palabras clave: *Supercomputo, Historia, México.*

Simulación de crecimiento de gliomas en paralelo

Omar Sánchez-Rodríguez, J.C. Chimal-Eguia, Ricardo Barrón-Fernández
Centro de Investigación en Computación, Instituto Politécnico Nacional, UP Adolfo López Mateos, México D.F., osanchezrod@gmail.com

Resumen

En este trabajo se estudia el problema de difusión de células cancerosas en el cerebro. Comunmente este tipo de crecimiento de células malignas se denomina glioblastomas. El tratamiento tradicional del problema a nivel matemático consiste en la solución de una ecuación diferencial (Parcial u Ordinaria), la cual describe el movimiento de gliomas (células nerviosas en el cerebro) malignos en la matriz extracelular y, dependiendo de la zona del cerebro en donde se encuentren las células iniciadoras, se genera un proceso difusivo de éstas. Uno de los problemas del uso de esta técnica es el manejo de las condiciones de frontera para las ecuaciones diferenciales, pues si no se describen mediante funciones continuas, entonces el problema se puede volver computacionalmente intratable. Considerando que los autómatas celulares (AC) no tienen este tipo de restrición y que desde el punto de vista computacional son objetos mas ad-hoc para problemas de frontera libre, en este trabajo se presenta un modelo de autómata celular para describir la difusión de gliomas malignos en el cerebro. Tomando en consideración el paralelismo intrínseco de los autómatas celulares, se presenta un modelo paralelizado con GPUs y se hace un análisis del modelo comparandolo con los datos reales de difusión en pacientes con esta problemática.

Palabras clave: *Autómata celular, crecimiento tumoral cerebral, modelación matemática, difusión, difusión-reacción.*

Tecnologías verdes empleando Clientes-Ligeros en un laboratorio de cómputo de la ESCOM

Cuevas Mora Erick, Manzano Cruz Pedro, Nájera Valdez Edgar Rolando
Escuela Superior de Cómputo del Instituto Politécnico Nacional

Resumen

En el presente documento se describe el desarrollo de un proyecto basado en un prototipo de laboratorio de cómputo que utiliza una infraestructura de clientes-ligeros, servidores virtualizados y desarrollos tecnológicos con el objetivo de favorecer al medio ambiente reduciendo el consumo de energía eléctrica y haciendo un mejor uso de los recursos computacionales disponibles. Se realizaron las pruebas en tres escenarios diferentes de infraestructura de clientes-ligeros y se obtuvieron resultados de consumo de energía en cada uno. Basados en los resultados se realizó el análisis y comparación para determinar el nivel de eficiencia de estos.

Solución numérica en paralelo de la ecuación de difusión empleando MPI y OpenMP.

Luis Alberto Vázquez Maison
Departamento de Matemáticas, UNAM.
Julio César Clemente González
Dirección General de Cómputo y de Tecnologías de la Información y Computación, UNAM.
Leobardo Itehua Rico
Dirección General de Cómputo y de Tecnologías de la Información y Computación, UNAM.

Resumen

La ecuación de difusión se emplea para la descripción de fenómenos de fluctuación, como por ejemplo la conduc-ción de calor a través de un medio sólido. Debido a la complejidad que puede presentar al resolverla, es con-veniente el uso de métodos numéricos para obtener una solución; sin embargo estos métodos al implementarlos en una computadora pueden resultar muy costosos, ya sea en tiempo o en alcance. En el presente artículo se propone un modelo numérico para la resolución de la ecuación de difusión usando diferencias finitas, así como su implementación en secuencial. Como se menciona, la implementación serial, al correrla es muy costosa computacionalmente; es por ello que los tiempos de ejecución resultan muy elevados, por lo que se presenta un esquema de paralelización sobre el algoritmo se-cuencial para disminuir los tiempos, permitiendo au-mentar el número de pasos a considerar; o aumentar el alcance, al lograr involucrar más celdas en el proceso de simulación. Adicionalmente se presentan métricas de rendimiento mostrando las mejoras en cuanto a tiempo de la ejecución en paralelo en comparación con su ver-sión secuencial. Así mismo cabe mencionar la utilización de arquitecturas de última generación para la realización de las pruebas; Intel Xeon serie E5, Infiniband QDR y LUSTRE; MPI-2.1 y OpenMP 3.0.

Palabras clave; *ecuación de difusión, diferencias finitas, MPI, OpenMP, programación paralela.*

Referencias:

Simulación numérica. Ecuación de difusión. Zulma Millán, Leonor de la Torre, Laura Oliva, María del Carmen Berenguer; Revista Iberoa-mericana de Ingeniería Mecánica. Vol. 15, N.º 2, pp. 29-38, 2011

Designing a GPU based Heat Conduction Algorithm by Leveraging CUDA. Xiaohua Meng, Dasheng Qin, Yuhui Deng, 2013 Fourth Inter-national Conference on Emerging Intelligent Data andWeb Technolo-gies. pp 519-526 2013

Wind River Systems, “Advanced compiler optimization techniques” April 2002. Online [December. 2012].

Implementación Paralela en GPUs de Algoritmos Evolutivos.

M.C. Hilda María Chablé Martínez, M.C. Oscar Alvarado Nava
Universidad Autónoma Metropolitana, Unidad Azcapotzalco

Resumen

Los Algoritmos Evolutivos (AE) son métodos estocásticos de búsqueda que son aplicados exitosamente en problemas de optimización, búsqueda y máquinas de aprendizaje [1]. Los AEs pueden tomar mucho tiempo en encontrar una solución, ya que requieren ejecutar muchas evaluaciones.

La filosofía de diseño de una Unidad de Procesamiento Gráfico (GPU) está determinada por el rápido crecimiento en la industria de los videojuegos. Esto ejerce una enorme presión económica sobre los vendedores de GPUs que buscan formas de maximizar la cantidad de cálculos de punto por fotograma de video [2].

Realizar cálculos de propósito general en una GPU es un nuevo concepto respecto al procesamiento en las Unidades Centrales de Proceso (CPUs) convencionales. Sin embargo, la idea de la computación en procesadores gráficos no es tan nueva como se podría pensar [3].

En este trabajo se presenta una implementación paralela en GPUs de un Algoritmo Evolutivo (AE) para la minimización de funciones de N variables (N menor o igual que 100). Haciendo uso de un analizador de ecuaciones matemáticas, se permite que las funciones a evaluar sean definidas dinámicamente.

Se realizaron experimentos de ejecución del algoritmo evolutivo en dos ambientes de prueba: una GPU Ge Force GT 630 (con 96 núcleos) y una GPU Tesla M2090 (con 512 núcleos). Se obtuvieron ganancias importantes en el factor de velocidad en los dos ambientes de prueba, con diferentes funciones objetivo.

Probando con poblaciones de diferente tamaño se alcanzó un factor de aceleración máximo de 43.54 en la GPU Ge Force, manteniendo una aceleración constante respecto al tamaño de la población. El factor de aceleración máximo en la GPU Tesla M2090 fue de 257.94, manteniendo un incremento lineal en la aceleración respecto al tamaño de la población.

References

[1] E. Alba and M. Tomassini. Parallelism and evolutionary algorithms. IEEE Trans. Evolutionary Computation.

[2] Wen-mei W. Hwu David B. Kirk. Programming Massively Parallel Processors A Hands-on Approach. Editorial Morgan Kaufmann, 2010.

[3] Edward Kandrot Jason Sanders. CUDA by example: an introduction to general purpose GPU programming. Editorial Addison Wesley, 2011.

Sistema de localización absoluta mediante odometría para un robot móvil (3,0)

Alvarado-Juárez Diego Armando[1], Villarreal-Cervantes Miguel Gabriel[1], Ponce de León Zarate Gerardo Arturo1, Sepúlveda-Cervantes Gabriel[1].
[1] Instituto Politécnico Nacional – CIDETEC. Av. Juan de Dios Bátiz s/n, Col. Nueva Industrial Vallejo, Deleg. Gustavo A. Madero, C.P. 07700, México D.F.

Resumen

La odometría depende en gran medida de la resolución de los decodificadores ópticos utilizados y del tiempo de muestreo. En este trabajo se desarrolla e implementa un sistema odométrico con base en un análisis cinemático para la estimación de la posición absoluta de un robot móvil omnidireccional (3,0). Resultados experimentales muestran el desempeño del sistema odométrico propuesto comparado con la estimación de la posición absoluta obtenida a través de un sistema de visión.

Palabras clave: *Robot móvil, omnidireccional, odometría, posición absoluta, estimación de la posición.*

Referencias

[1] J. Borenstein, H.R. Everett, L. Feng, D. Wehe. "Mobile Robot Positioning. Sensor and techniques." Invited paper for the journal of robotic systems, Special issue on mobile robots, Journal of Robotic Systems, Vol 14. Pp 231-249.

[2] Y. Kamer, S. Ikizoglu. "Effective accelerometer test beds for output enhancement of an inertial navigation system", Elsevier, Meassurement #46, 2013.

[3] J. Timothy, J. Peters. "Automobile Navigation Using a Magnetic Flux-Gate Compass", IEEE Transactions on vehicular technology, Vol. 35, Mayo 1986.

[4] T. Venet, T. Capitaine, M. Hamzaoui, F. Fazzino, "One Active beacon for an indoor absolute localization of a mobile vehicle", IEEE International Conference On Robotics & Automation, Washington, Mayo 2002.

[5] Y. Hada, K. Takase, "Multiple Mobile Robot Navigation Using The Indoor Global Positioning System (iGPS)", IEEE International Conference on Intelligent Robots and Systems, Hawaii, Octubre 2001.

[6] P. MacKenzie, G. Dudek, "Precise Positioning Using Model-Based Maps", IEEE International Conference on Robotics & Automation, 1994.

[7] K. Nagatani, S. Tachibana, M. Sofue, Y. Tanaka, "Improvement of Odometry for Omnidirectional Vehicle using Optical Flow Information", IEEE International Conference on Intelligent Robots and Systems, 2000.

[8] S. Kernabach, "Encoder-free odometric system for autonomous microrobots". Elsevier Science Direct Mechatronics #22. Pp. 870-880. 2012.

[9] J. Palacin, I. Valgañon, R. Pernia. "The optical mouse for indoor mobile robot odometry measurement". Elsevier Science Direct Sensors and Actuators A #126. Pp. 141-147. 2006

[10] J. Gutierrez, F. Villa, M. A. Porta. "Vertically aligned accelerometer for wheelded vehicle odometry". Elsevier Science Direct Mechatronics #20. Pp. 617-625. 2010.

[11] L. Kooktae, C. Woojin, Y. Kwanghyun. "Kinematic parameter calibration of a car-like mobile robot to improve odometry accuracy". Elsevier Science Direct Mechatronics #20. Pp. 582-595. 2010.

[12] J. Hu, Y. Chang, Y. Hsu. "Calibration and on-line data selection of multiple optical flow sensors for odometry applications". Elsevier Sensors and Actuators A #149. Pp. 74-80. 2009

[13] M. Muniandy, K. Muthusamy. "An innovative design to improve systematic odometry error in non-holonomic wheeled mobile robots". International Symposium on Robotics and Intelligent Sensors. 2012.

[14] N. Ushimi, Y. Aoba. "Development of a two-whheels caster type of odometry with a dual shaft for omnidireccional mobile robots". International Symposium on Robotics and Intelligent Sensors. 2012.

[15] D. Alvarado, M.G. Villarreal, G. Sepúlveda "Modelo cinemático y dinámico de un robot móvil omnidireccional". Congreso Nacional de Mecatrónica 2012.

Topology Optimization with Univariate Marginal Distribution Algorithm in Parallel

N. Faurrieta, S. Botello, S. I. Valdez
Centro de Investigación en Matemáticas, CIMAT, A.C. Jalisco SN, Col. Valenciana, Guanajuato, Gto. México, C.P. 36240

Abstract

Structural optimization refers to the problem of finding the optimum design of structure that minimizes an objective function, furthermore, the structure must meet a set of restrictions, such as does not exceed the maximum allowable stress for the material, does not exceed a maximum displacement, etc. There are different methods to optimize structures as analytical, numerical and empirical. Some types of optimization within these methods are sizing optimization, shape optimization and topology optimization. The topology optimization is one of the most used numerical method to find structures with optimal material distribution. In this type of optimization can introduce and remove material, generate holes in any part of the structure, and add or delete elements from this, and is the least restricted in the type of geometries that can be achieved. A proposal for structural optimization method combining local and global optimization is performed in the present work. The local optimization is based on the Von Mises stress information, which is calculated by performing a stress analysis by the Finite Element Method (FEM) parallelized with OpenMP. The global search using guided by the objective function populations. This process used an Univariate Marginal Distribution Algorithm, where the evaluation of each individual in the population means applying the local search method, due to the computational cost and the characteristics of the algorithm, it is very advantageous to parallelize these evaluations with MPI. Finally, a post-process is applied to the resulting structure, which aims to create structures with better aesthetics and ease to manufacture.

References

[1] Método de los Elementos Finitos para Análisis Estructural / Juan Tomás Celigüeta Lizarza. Campus tecnológico de la Universidad de Navarra.

[2] Una nueva formulación de elementos finitos para la optimización topológica de estructuras basadas en la minimización del peso con restricciones en tensión/ J. París,F. Navarrina, I. Colominas y M. Casteleiro/ Grupo de Métodos Numéricos en Ingeniería. Universidade da Curuña.

[3] Introduction to Shape Optimization: Theory, Approximation, and Computation / J. Haslinger, R. A. E. Mäkinen, p. cm.- (Advances in design and control)

[4] H. Muehlenbein1, J. Bendisch1, and H.-M. Voight2. From Recombination of Genes to the Estimation of Distributions II. Continuous Parameters, 1996.

[5] P. Larrañaga and J. A. Lozano. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

Shapetimization using Genetic Algorithms, Delaunay Condition Based Boundary Search and the Finite Element Method

Victor Cardoso-Nungaray, Ivvan Valdez-Pe~na, Salvador Botello, A. Hernandez.
Centro de Investigacion en Matematicas A.C.,
Jalisco S/N, Col. Valenciana, Zip 36240,
Guanajuato, Gto., Mexico,
email: victorc, ivvan, botello, artha@cimat.mx, web: www.cimat.mx

Abstract

The shapes design has been an important challenge throughout the history, from building bridges, aqueducts, temples and weaponry until the complicated mechanisms pieces of todays world, all this components must be optimum in some way, and it has been responsibility for the engineers to accomplish this goal. We propose the term \Shapetimization" as a contraction of \ Shape optimization", to encapsulate the concept in a single word. We dene \shapetimization" as the process of nding an optimal shape of a component that should give a determined service eciently, the eciency is measured by an objective function. In this way, \shapetimization" is not a specic methodology.

We use a Genetic Algorithm (GA) to shapetimize a structure automatically, Delaunay Condition Based Boundary Search (DCBBS) to dene the structure boundary and the Finite Element Method (FEM) to evaluate the eciency of each shape codied by a chromosome. We start the process with a raw and gross shape, then we lled up with N points distributed uniformly across this initial shape. Each chromosome is formed by one real positive gene and N binary genes, the real positive gene is used to dene the boundaries of the structure with DCBBS, and the N binary genes are used to dene the structure, if the gene is equal to one, then the corresponding point is used to create the shape, otherwise the corresponding point is useless. Balamurugan et al. presents a similar approach with a Two Stage Adaptative Genetic Algorithm (TSAGA) in structural topology optimization [1], Valdez et al. estimate the search distribution with EDAs also for structural topology optimization [2], and Talischi et al. uses polygonal elements to generate, analyze and evaluate the structures [3].

The evaluation of the adaptation function is very expensive due to the use of FEM, including meshing, inside the computation, this could lead to long waiting times if we don't use High Performance Computing (HPC) techniques, by such reason we propose a ring topology in a distributed memory paradigm, implemented with the Message Passing Interface (MPI).

Keywords: *Shapetimization, Genetic Algorithms, GA, Finite Element Method, FEM, Parallel Computing, MPI, shape, design, optimization.*

Resumen

El diseño de formas ha sido un reto importante durante toda la historia de la humanidad, desde la construccion de puentes, acueductos, templos y armamento hasta las piezas de los mecanismos complicados del mundo actual, todos estos componentes tienen que ser optimos de alguna forma, y ha sido responsabilidad de los ingenieros alcanzar este objetivo. Se propone el termino \Shapetimizacion" como una contraccion en ingles de \Optimizacion de forma", para encapsular el concepto en una sola palabra. Se dene \Shapetimizacion" como el proceso de busqueda de una forma optima para un componente que debe dar un servicio determinado ecientemente, midiendo la eciencia con una funcion objetivo. En este sentido, \ Shapetimizacion" no es una metodologa en especco.

En este trabajo se usa un Algoritmo Genetico (GA) para shapetimizar el componente automaticamente, utilizando la Busqueda de Fronteras Basada en la Condicion de Delaunay (DCBBS) para denir los lmites de la estructura y el Metodo de Elemento Finito (FEM) para evaluar la eciencia de cada forma codicada por un cromosoma. El proceso comienza con una forma burda e ineciente, despues se llena con N puntos distribuidos uniformemente dentro de la misma. Cada cromosoma esta formado por un gen de codicacion real (positivo) y N genes de codicacion binaria, el gen de codicacion real es usado para denir las fronteras de la estructura con DCBBS, y los N genes binarios son utilizados para denir la estructura misma, si y solo si el gen binario es igual a uno, el punto correspondiente formara parte de esta. Balamurugan et al. presentan un enfoque similar con un Algoritmo Genetico Adaptativo de Dos Etapas (TSAGA) en una optimizacion topologica [1], Valdez et al. estiman la distribucion de la busqueda utilizando EDAs [2], y Talischi et al. usan elementos poligonales para generar, analizar y evaluar las estructuras [3].

La evaluacion de la funcion de adaptacion es muy costosa debido al uso de FEM dentro del computo, incluyendo el mallado, lo que puede conducir a tiempos de espera muy largos si no se utilizan tecnicas de Computo de Alto Rendimiento (HPC), por esta razon se propone una topologa de anillo en un paradigma de memoria distribuida, implementado con MPI.

Palabras clave: *Shapetimizacion, Algoritmos Geneticos, GA, Metodo de Elemento Finito, FEM, Computo Paralelo, MPI, formas, diseño, optimizacion morfologica.*

References

[1] R. Balamurugan, C. Ramakrishnan, N. Singh, \ Performance evaluation of a two stage adaptative genetic algorithm (TSAGA) in structural topology optimization". Applied Soft Computing, 8(4): 1607-1624, 2008, 1586-4946.

[2] S.I. Valdez-Pe~na, A. Hernandez-Aguitte, S. Botello-Rionda, \Approximating the search distribution to the selection distribution in EDAs", in Proceedings of the 11th Annual conference on Genetic and evolutionary computation, GECCO '09, pages 461-468. ACM, New York, NY, USA, 2009, 978-1-60558-325-9.

[3] C. Talischi, G. Paulino, A. Pereira, I. Menezes, \ Polygonal nite elements for topology optimization: A unifying paradigm", International journal for numerical methods in engineering, 82(6): 671-698, 2010.

Algoritmo evolutivo en paralelo para problemas de optimización

Ericka García-Blanquel, René Luna-García
Centro de Investigación en Computación – IPN

Resumen

En los últimos años la solución de problemas utilizando cómputo paralelo ha sido de gran interés dentro de la comunidad científica en diversas áreas ya que permiten la solución de problemas que antes eran casi imposibles de resolver como los problemas de optimización, los cuales además de tener una complejidad NP computacionalmente, requerían de mucho tiempo para el procesamiento de los datos. Por tal motivo el presente trabajo se enmarca dentro de la línea de investigación del cómputo de alto rendimiento implementando un algoritmo evolutivo en paralelo sobre en una arquitectura CUDA para la solución de un problema de optimización en el área de dinámica molecular conocido como minimización energética.

Los algoritmos evolutivos son métodos efectivos para dar solución a diversos problemas de optimización, sin embargo tienen la desventaja que requieren de un costo computacional muy alto, debido al elevado número de posibles soluciones candidatas que tienen que evaluar, por lo que para superar esta desventaja se hace uso de las técnicas en paralelo, las cuales nos ayudan para aumentar la velocidad de procesamiento y al mismo tiempo nos permite encontrar una mejor solución ya que se tiene la opción de explorar con mayor precisión el espacio de búsqueda.

Referencias

[1] Md. Ashfaquzzaman Khan, Martin C. Herbordt. Parallel discrete molecular dynamics simulation with speculation and in-order commitment. Journal of Computational Physics 230 (2011) 6563–6582

[2] E. Proctor, F. Ding, N. Dokholyan, Discrete molecular dynamics, Wiley Interdisciplinary Reviews: Computational Molecular Science 1 (1) (2011) 80–92.

[3] S. Yun, H. Guy, Stability tests on known and misfolded structures with discrete and all atom molecular dynamics simulations, Journal of Molecular Graphics and Modelling 29 (5) (2011) 663–675.

[4] Silvia Ochoa, Günter Wozny, Jens-Uwe Repke. A new algorithm for global optimization: Molecular-Inspired Parallel Tempering. Computers and Chemical Engineering 34 (2010) 2072–2084.

[5] John E. Stone a, David J. Hardy a , Ivan S. Ufimtsev b , Klaus Schulten . GPU-accelerated molecular modeling coming of age . Journal of Molecular Graphics and Modelling 29 (2010) 116–125.

[6]Tsutsui, S.; Fujimoto, N. An Analytical Study of GPU Computation for Solving QAPs by Parallel Evolutionary Computation with Independent Run (2010) 889–896.

[7]Banzhaf, W.; Harding, S.; Langdon, W. B.; Wilson, G. Accelerating Genetic Programming Through Graphics Processing Units (2009) 1–19.

[8] Matsuoka, S.; Aoki, T.; Endo, T.; Nukada, A.; Kato, T.; Hasegawa, A. GPU accelerated computing from hype to mainstream, the rebirth of vector computing (2009)

[9] Calvo, F. Non-genetic global optimization methods in molecular science: An overview. Computational Materials Science(2009) 45, 8 .

[10] Li, Y., Mascagni, M., & Gorin,. A decentralized parallel implementation for parallel tempering algorithm. Parallel Computing (2009) 35, 269.

[11] J.C. Phillips, J.E. Stone, Probing biomolecular machines with graphics processors, Commun. ACM 52 (2009) 34–41.

[12] NVIDIA Corp., CUDA Architecture Introduction & Overview, NVIDIA Cor., Santa Clara California. Versión 1.1. (2009) 3.

[13] J.D. Owens, M. Houston, D. Luebke, S. Green, J.E. Stone, J.C. Phillips, GPU computing, Proc. IEEE 96 (2008).

[14] G. Giupponi, M. Harvey, G.D. Fabritiis, The impact of accelerator processors for high-throughput molecular modeling and simulation, Drug Discov. Today 13 (2008) 1052–1058.

[15] Derek Juba, Amitabh Varshney. Parallel, stochastic measurement of molecular surface area . Journal of Molecular Graphics and Modelling 27 (2008) 82–87.

Algoritmo evolutivo para el desarrollo en cascada de un rayo cosmico

Andres Zamudio-Cortes, René Luna-García
Centro de Investigación en Computación Instituto Politécnico Nacional
C.P. 07738 E-mail: azamudio_b12@sagitario.cic.ipn.mx, lunar@cic.ipn.mx
Web: http://www.cic.ipn.mx

Resumen

El desarrollo de los rayos cósmicos esta gobernado por la ecuación de cascada y por la ecuación de Gaisser-Hillas, estas ecuaciones muestran en forma general la densidad de partículas en función del tiempo y de la posición, asi como de la energía y dirección inicial de la particula primaria.

En este trabajo realizamos simulaciones de rayos cósmicos con la ayuda de programas como CORSICA Y AIRES, y con el uso de diferentes modelos de colisiones hadronicas tales como Sibyll y QGSJET.

Se propone la construcción de una red neuronal que es entrenada con los datos simulados, esta tiene como finalidad la recontrucción de datos obtenidos experimentales por diversos observatorios de rayos cósmicos.

Algoritmo Evolutivo Paralelo para la Modelación de Tráfico Vehicular.

Jonatan Alvarez Méndez, René Luna García
Centro de Investigación en Computación del Instituto Politécnico Nacional

Resumen

En la actualidad se busca la mejora del flujo del tránsito vehicular en una red urbana de vialidades, siendo de gran importancia la optimización del modelo matemático que lo describe.

En este trabajo, presentamos un modelo de comportamiento del flujo del tránsito vehicular a partir de una representación de ecuaciones de flujo continuo dentro de un grafo direccionado. El algoritmo evolutivo propone una solución inicial con todos los parámetros dentro de valores preestablecidos y a partir de ésta se busca mejorar la solución anterior, mediante variaciones diferenciales implementadas en paralelo para cada ecuación de flujo; evaluando y sincronizando dentro del algoritmo las condiciones de frontera (semáforos, señales de alto, la frecuencia con la que los automóviles o peatones cruzan por una determinada vialidad, entre otras).

Por lo tanto para la optimización del algoritmo, la solución de las ecuaciones se encuentra numéricamente en forma paralela.

Esquema Adaptativo para Problemas Tridimensionales Utilizando Campos Vectoriales

René Luna-García, Claudia García-Blanquel, Ericka García-Blanquel
Centro de Investigación en Computación
Instituto Politécnico Nacional, C.P. 07738
E-mail: lunar@cic.ipn.mx, cgarciab@sagitario.cic.ipn.mx, egarciab10@sagitario.cic.ipn.mx
Web: http://www.cic.ipn.mx

Resumen

El análisis en tres dimensiones de datos se ha convertido en una herramienta importante en la investigación científica. En muchas aplicaciones, el análisis de estructuras, son de gran interés y puede ayudar a obtener una comprensión más profunda del problema subyacente. Por lo general este análisis se lleva a cabo con algoritmos numéricos. Uno de los problemas que se enfrentan a este tipo de algoritmos numéricos es la naturaleza discreta de las estructuras. Dependiendo de los datos de entrada, la estructura resultante puede depender en gran medida de los parámetros algorítmicos y procedimientos numéricos. Un problema es la presencia de ruido, por ejemplo debido al proceso de formación de imágenes, o artefactos de muestreo. Ambos pueden crear fluctuaciones en los valores que pueden crear estructuras adicionales, que son muy complejos y difíciles de analizar. Una distinción entre los elementos importantes y esenciales es por lo tanto crucial. Sin embargo, el marco es aplicable únicamente a pequeños conjuntos de datos de tres dimensiones desde la construcción de la secuencia requiere de varios recorridos. Esto se traduce en un tiempo de funcionamiento no factible para grandes conjuntos de datos. Enfoques alternativo para extraer los puntos esenciales al para separar los elementos no esenciales procedentes de las más importantes, se ven rebasados al encontrar información que se conecta entre sí, lo que causa en tiempo de procesamiento una sobrecarga de memoria ya que la información geométrica es almacenada por separado.

En este trabajo se presenta un esquema adaptivo para problemas tridimensionales mediante la selección del campo vectorial que mejor describa las características intrínsecas del problema. El esquema adaptativo incluye como estrategia la división del dominio en subdominios de tal manera que se puedan seleccionar los puntos de interés de la estructura y con ello se lleve a cabo la estimación del error. Con ello se va a ajustar la densidad de los puntos en los subdominios. Es posible encontrar un conjunto de puntos que nos proporciones un coste computacional menor y/o mejorar la precisión de los resultados.

Supervisión Metaheurística en Paralelo para Formación y Crecimiento de Cúmulos de Células Anómalas

Jiménez Espinosa Juan Carlos, René Luna-García
Centro de Investigación en Computación, Instituto Politécnico Nacional, C.P. 07738
E-mail: lunar@cic.ipn.mx, jjimenez_b12@sagitario.cic.ipn.mx, Web: http://www.cic.ipn.mx

Resumen

Se propone un modelo simple con base en la teoría del modelo CSC (Cancer Stem Cells) para simular el crecimiento de un cumulo de células anómalas, cuyas características principales a modelar son las siguientes: proliferación, movilidad y muerte, además de observarse la influencia reciproca entre células, su adaptación y adaptabilidad, así como la probabilidad de aparición del valor de simetría o asimetría de las células CSC.

En cada simulación se considera el número de células en el cúmulo y su periferia, un gradiente espacial preferencial en su desarrollo, y las dimensiones iniciales del cumulo, además de tomar en cuenta valores de probabilidad de simetría, calculados a partir de encontrar el punto de inflexión que determina donde el sistema esta equilibrado.

El algoritmo propuesto genera paralelamente un numero n de generaciones de células hasta encontrar reciprocidad en la interacción de células en el cumulo y aplicación de las condiciones de frontera.

Semi-automatic Historical Climate Data Recovering Using a Distributed Volunteer Grid Infrastructure

Sergio Nesmachnow, Miguel Da Silva
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay {sergion,mdasilva}@fing.edu.uy

Abstract

This article presents the Digi-Clima project, whose main objective is to design a semi-automatic tool for digitalizing and recovering historical climate data using distributed computing techniques on grid and cloud infrastructures. The deploy of the processing tool for historical climate records on the Ourgrid middleware for grid and cloud is described, and the experimental evaluation using a volunteer-based Ourgrid infrastructure is reported. Accurate speedup values are reported for the distributed application.

Sergio Nesmachnow, Miguel Da Silva
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
{sergion,mdasilva}@fing.edu.uy

Prototipo Web de Ubicación Segura
Tema de investigación: Aplicación Científica y Tecnológica

Angélica Vianey Cruz Monroy, Leonor Elva Santa Martínez, Mayte Ballesteros Santana
Sergio Peña Orozco, Erik Guerrero Bravo, José Angel Pérez Hernández

Resumen

La seguridad pública se ha convertido en un tema crucial para el desarrollo social del país, porque esta busca brindar tranquilidad y seguridad humana, las cifras de delincuencia e inseguridad se han incrementado en los últimos años, razón por la cual es preciso identificar zonas de alta delincuencia e inseguridad a fin de tomar acciones preventivas gubernamentales.

El trabajo pretende mostrar a través de datos las situaciones críticas en localidades, municipios, regiones o estados. Existen en el país entidades dedicadas a presentar información en este tema, tal es el caso del Observatorio Ciudadano Nacional (OCN) y el Secretariado Ejecutivo del Sistema Nacional de Seguridad Pública, que a pesar de ser dos organizaciones de ámbitos diferentes, tienen como objetivo brindar información oportuna de los hechos o situaciones delictivas que se generan en el país; sin embargo, los cálculos de esta información no son de fácil acceso para los usuarios finales.

Por lo antes expuesto la minería de datos se convierte en una herramienta útil para el análisis estadístico de datos e identificación de patrones o tendencias, generando información de vital importancia para predecir el comportamiento delictivo de una zona determinada o tomar la mejor decisión, asociado a una Interfaz de Programación de Aplicaciones (API) de google Maps, la cual permite obtener una mejor perspectiva mediante la segmentación y ocurrencia de los delitos dentro de cualquier estado concibiendo de esta manera una referencia basada en el principio y simulación de un indicador similar al de un semáforo donde se visualiza información dinámica basada en estadísticas y técnicas de minería de datos bajo la siguiente nomenclatura: El color rojo representa para cada uno de los municipios o localidades una alta incidencia delictiva y el color verde representa un municipio con un grado de incidencia menor.

El desarrollo de este prototipo web busca poder tomar precauciones y acciones de gobierno para el bien social.

Referencias

José Hernández Orallo, M.José Ramírez Quintana, Cèsar Ferri, Ramírez, Introducción a la minería de datos Editorial: Pearson Navidi, Estadística para ingenieros y científicos, Editorial: McGraw Hill

Ramez Elmasari, Shamkant B. Navathe, Fundamento de sistemas de base de datos, Editorial: Pearson

Nicolás Arrioja Landa Cosio, Curso de programación C#, ISBN: 978-987-1347-76-6

Análisis de la Oferta Actual de Servicios de Cloud Computing ¿Es viable migrar a la nube?

Moisés Gil Solorio[1], Juan Manuel Ramírez-Alcaraz[1], Carlos Alberto Flores-Cortés[1]
[1]*Universidad de Colima, Facultad de Telemática*
Av. Universidad 333
Colima, Colima, México.
{moises_gil, jmramir, cfcortes}@ucol.mx

Resumen

En el mundo actual, dentro de las diferentes empresas y centros educativos es común encontrar infraestructura de cómputo que brinda los servicios que satisfacen las necesidades internas de cada uno de ellos (Bankinter Fundación de la innovación, 2010), sin embargo, la administración de los recursos de tal infraestructura computacional puede llegar a presentar problemas de diferentes índoles, como puede ser el mantenimiento preventivo/correctivo, actualizaciones de software, etc. En este trabajo se hace un análisis de la oferta actual de servicios de Cloud Computing con el objetivo de proporcionar una referencia para apoyo a la toma de decisiones respecto a la migración de los sistemas de cómputo a la nube, entendiendo esta como una serie de servicios que un proveedor brindará a nuestra empresa o centro educativo (Buyya, Broberg, & Andrzej, 2011), por el tiempo y la cantidad justa que los necesitemos.

Se analizan tres de los proveedores de servicios más conocidos: Google (Google, 2012), Amazon EC2 (Amazon Web Services, 2013) y Microsoft Windows Azure (Microsoft, 2013). Aun cuando no todos los proveedores ofrecen los mismos tipos de servicios y los costos pueden diferir significativamente, se observa en general que a largo plazo el beneficio de migrar a la nube se hace cada vez más visible.

Palabras clave: *Cloud Computing, Computación en nube, infraestructura, costos de servicios, migración a la nube.*

Referencias

Amazon Web Services. (2013). Amazon Web Services. Obtenido de Amazon Elastic Compute Cloud: http://aws.amazon.com/es/ec2/

Bankinter Fundación de la innovación. (2010). La apuesta de las empresas tecnológicas por la nube. En Bankinter, Cloud Computing: La tercera ola de las Tecnologías de la Información. (pág. 22). España: Accenture.

Buyya, R., Broberg, J., & Andrzej, G. (2011). Cloud computing : principles and paradigms (Primera ed.). Hoboken: John Wiley & Sons, Inc.

Google. (2012). Google Compute Engine - computation in the cloud. Google.

Microsoft. (2013). Windows Azure. Recuperado el 16 de Abril de 2013, de Windows Azure Platform: http://msdn.microsoft.com/es-es/windowsazure/gg318632.aspx

Software educativo y procesos en paralelo como una herramienta en la enseñanza–aprendizaje en programación.

Maribel Acevedo Camacho, Nelly Beatriz Santoyo Rivera, Daniel Jorge Montiel García.
maribel.acevedo.camacho@hotmail.com, nesantoyo@itesi.edu.mx, dante02d@hotmail.com
Instituto Tecnológico Superior de Irapuato

Resumen

En la actualidad el uso de las tecnologías de Información han incursionado en casi todos los aspectos de la vida cotidiana, y la educación no es la excepción. Se pensaría que la educación seria el referente en la incursión de las nuevas tecnologías, pero la realidad es otra, existe un numero relativamente pequeño de proyectos que se enfocan en temas educativos. Esto debido a la complejidad que guardan dichos software, puesto que requieren de procesar múltiples parámetros de manera paralela. En el presente articulo se presentan las metodología empleada para la producción de un software educativo para facilitar la transferencia de conocimiento en el proceso de enseñanza de la programación.

Este procesamiento es en paralelo de las interfaces, inteligencia en las formas de mostrar la interacción con el usuario, el procesamiento de esta información, entre otros parámetros dio como resultado la aplicación de un software con alto rendimiento.

Referencias:

Arendes, R. (2004). Learning to teach (6ª. Edición). Nueva York:McGraw Hill.

Carolina Elizabeth Sánchez Zapata, "Tecnologías de la Información aplicadas a la educación.

Hannafin, M. Land S. Y Oliver, K. (2000). Entornos de aprendizaje abiertos: fundamentos, métodos y modelos, en Reigeluth, Ch. (Ed.). Diseño de la instrucción. Teorías y modelos. Madrid. Aula XXI Santillana, parte I, pp. 125-152.

Energy expenditure optimization on an on-demand operated cluster

María del Carmen Heras Sánchez, Daniel Mendoza Camacho, Raúl G. Hazas-Izquierdo, Aracely B.M. Dzul Campos
Universidad de Sonora

A novel electrical power-saving methodology is devised to optimize the power usage of an entry-level high-performance computing cluster. Born as an HPC developmental project by the Área de Cómputo de Alto Rendimiento de la Universidad de Sonora (ACARUS), this cluster has undergone several re-designs and re-deployments as newer hardware and software technologies have been available. It currently encompasses visualization, mathematical libraries, parallel computing, load-balance, scheduler, and monitoring tools among others. Since the cluster is aimed for training purposes and to support low-end academic workloads, it goes through periods of low resource demand therefore it is important to avoid energy waste without compromising availability, performance, and efficiency.

Efficient analysis of molecular data in the field of structural virology

Daniel Jorge Montiel-García[1], Vijay Reddy[2] and Mauricio Carrillo-Tripp[1]
dante02d@hotmail.com, reddyv@scripps.edu, trippm@gmail.com
[1]Biomolecular Diversity Laboratory, Unidad de Genómica Avanzada (Langebio), CINVESTAV, Sede Irapuato 36800, México
[2]Integrative Structural and Computational Biology, The Scripps Research Institute, La Jolla, CA 92037, USA

Abstract

Generation of knowledge derived from the analysis of `big data' has become a priority in the computational sciences [1]. This is particularly relevant in the study of biological data, given the high amount of information produced by the technological advances seen in the last decades in this area. Nonetheless, the analysis of big volumes of information with complex internal relationships still represents a great challenge from the computational point of view. There is an urging need to develop new methods that optimizes this process and facilitates the discovery of hidden patterns relevant in diverse biological processes [2]. In this work we present the development of strategies for the fast and efficient analysis of molecular information in the field of structural virology [3]. These include algorithms and techniques that allow to process data in a parallel fashion, taking advantage of the current high performance architectures available. Furthermore, it is worth noticing that the results obtained by the use of this methodology are the foundation which will allow for the rational design of novel specific antivirals.

References

[1] Piateski G., Frawley W., Knowledge, (1991), Discovery in Databases,MIT Press Cambridge.

[2] Wee-Chung A., Yan H., Yang M., (2005) Pattern recognition techniques for the emerging field of bioinformatics: A review, Pattern Recognition, 38, 2055-2073.

[3] Carrillo-Tripp ,M.,C.Shepherd,I.Borelli,S. Venkataraman, G. Lander, P. Natarajan, J. Johnson, C. Brooks, V. Reddy (2009) VIPERdb2: an enhanced and web API enabled relational database for structural virology Nucleic Acids Research 37 (Database issue), D436-D442.

Author Index

Edited In:
Los Talleres Gráficos de Transición
Mezquitan 465
Col. Artesanos
University of Guadalajara
March 2015

WHERE **HIGH PERFORMANCE COMPUTING** FLIES

www.isum.mx

This edition Supercomputing in Mexico: Where High Performance Computing Flies presents 14 original peer reviewed research papers written by 44 authors from México, Europe, United States, and Latin America. These works cover five thematic fields in the area of supercomputing. The four thematic fields include:

Applications
Cloud Computing
Grids/GPU´s
Infrastructure
Parallel Computing
Scheduling

The contents of this 5th volume, first edition can be of interest to computer science and engineering researchers, undergraduate and graduate students, professionals, and researchers working on HPC, as well as the general public interested in current research in Supercomputing.

Dr. Moisés Torres Martínez

ISBN: 978-607-742-205-1

CNS CENTRO NACIONAL DE SUPERCÓMPUTO

cudi

CONACYT

IPICYT

CICESE