**More than Research**

2nd International Supercomputing Conference in México 2011 (ISUM)

Conference Proceedings

# ISUM

IPICYT

CNS CENTRO NACIONAL DE SUPERCÓMPUTO

CONACYT Consejo Nacional de Ciencia y Tecnología

CGTI

UNIVERSIDAD DE GUADALAJARA

**ISUM 2011 Conference Proceedings**
Editorial Board

**Editor:** Dr. Moisés Torres Martínez

**Primary Reviewers**
Dr. Andrei Tchernykh (CICESE)
Dr. Juan Carlos Chimal Enguia (IPN)
Dr. Manuel Aguilar Cornejo (UAM-Iztapalapa)
Dr. Moisés Torres Martínez (UdG)
Dr. René Luna García (IPN)

**Secondary Reviewers**
Luis Ángel Alarcón Ramos, UAM-Cuajimalpa
Jorge Matadamas, UAM-Iztapalapa
Juan Manuel Ramirez Alcaraz, UCOLIMA
Adán Hirales Carbajal, CICESE
Ariel Quezada Pina, CICESE

**Language reviewer**
Dr. Eduardo Mosqueda, U.C. Santa Cruz

**Editorial Board Coordination**
Fabiola Elizabeth Delgado Barragán
Verónica Lizette Robles Dueñas
Leticia Benitez Badillo
Carlos Vazquez Cholico

**Formattting**
Gloria Elizabeth Hernández Esparza
Cynthia Lynnette Lezama Canizales

# More than Research

"2nd International Supercomputing Conference in México 2011"

**Volume Editor:**

Dr. Moisés Torres Martínez

*University of Guadalajara*
*General Coordination of Information Technologies*
*México 2011*


**ISBN: 978-607-450-484-2**


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

**Volume II: 978-607-450-484-2**
1st Edition

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

Printing: 1,000 Issues/1,000 ejemplares

Printed in México

# DIRECTOR´S MESSAGE

The 2nd International Supercomputing Conference in México (ISUM) hosted by the *Instituto Potosino de Investigación en Ciencia y Tecnología-Centro Nacional de Supercomputo and the Universidad de Guadalajara* in collaboration with the national ISUM Committee proved to be a real success with the participation of the many experts in supercomputing who came from the European Union, Latin America, México and the United States. The participation of these supercomputing experts gave the conference attendees a new insight on the latest research being conducted throughout the globe, for example Dr. Mateo Valero from the Barcelona Supercomputing Center gave participants a new insight on Exaflop Supercomputers with his work on the MareIncognito research project, which aims at developing some of the technologies considered of key relevance on the way to Exascale. This project is the mere example on the innovative work not only presented by the illustrious keynote speakers, but also by the conference presenters who demonstrated the high quality work being done throughout the nation.

It is rather satisfying to see that the ISUM is continuing to grow nationally and internationally and serves as the space for researchers from and throughout the globe to share their work and be able to create collaborations with researchers in México and Latin America. It is through these collaborations that research with the uses of supercomputing is able to advance at higher levels and solve some of the most challenging problems our respective societies are facing today. This book presents some of this research and it wouldn´t have been possible, if it was not for the commitment from the ISUM National Committee in making this event a reality with the intent of fostering the uses of supercomputing in research and development. Thus, it is an honor to have had the University of Guadalajara and all the participating academic institutions in such an event that is able to gather the brightest minds nationally and internationally with the objective of advancing science and technology using supercomputing.

My deepest congratulations to all ISUM 2011 participants for advancing their research work through the uses of high performance computing and especially to the authors who contributed to this book. I invite you to read through the articles of your choice and to participate in future events by submitting your research work for publication in the ISUM conference proceedings.

Look forward to seeing the continuous success of ISUM events and publications to continue fostering the growth of supercomputing nationwide.

**Ing. Léon Felipe Rodríguez Jacinto**
General Director
Coordinating Office of Information Technologies
University of Guadalajara

VOLUME II CONTENTS 1st EDITION

## Architectures



## Grids

## Infrastructure



## Parallel Computing

## Scientific Visualization



## Appendix I



## Appendix II

# FOREWORD

The international scientific community is addressing the challenging problems that our individual societies are facing whether it is in education, climate change, health, economics or national security (to name a few), and the use of supercomputers allows this community to analyze data at incredible speeds achieving results and/or solutions to some of the most complex problems we face in our respective societies today. What once took days, months or years to analyze can now be accomplished in a matter of seconds, minutes or hours due to the power of supercomputing. It is known that supercomputing in México and Latin America continues to lag in comparison to Europe, Asia or the United States. However, during the past decade we´ve seen growth in the uses of supercomputing to conduct high quality research in México and Latin America; and we are beginning to notice this growth in México with the surge of supercomputer centers around the nation. For example the National Supercomputing Center in San Luis Potosí (IPICyT-CNS) is a national model for its innovativeness and commitment to work with industry to find faster and more efficient solutions to some of their most complex problems. Supercomputing centers of this caliber are now more accessible to the scientific community to conduct high quality research throughout the nation.

The burgeoning interest in supercomputing by the scientific community is a testament of the value they see in computing to advance their research. Needless to say that the participation of this community in the 2nd International Supercomputing Conference in México does in fact demonstrate that the nation´s growth in this area will continue to be steady as our scientists and engineers continue to participate and share their work on events of such magnitude that promote the uses of supercomputing in research. As supercomputing continues to evolve in México, we also expect to see growth in the research aspect of supercomputing so the country can contribute to the evolution of High Performance Computing (HPC) and not simply be a user of these systems. Some of the research presented in this book addresses issues dealing with architecture, grids, parallel processing, scientific visualization, infrastructure and applications, which in turn are advancing and impacting research in supercomputing. It is motivating to witness the country's significant contributions to these areas of supercomputing and we expect to see more of these works as the ISUM conference proceedings continue to be published in the coming years. This second volume presents the works of 46 authors focused on the aforementioned themes, which are some of the most contemporary issues in High Performance Computing.

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. The host of this year´s 2nd International Supercomputing Conference in México, *Instituto Potosíno de Investigación Cientifica y Tecnologíca- Centro Nacional de Supercomputo* (IPICyT-CNS) did an

excellent job hosting ISUM 2011 in the city of San Luis Potosi. Kudos to all who contributed to the success of this conference because without the commitment from all the individuals and academic institutions that participated, this event would not have had the success it did, and these conference proceedings are a 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 work and contributed to this second volume of the ISUM conference proceedings. The nationwide impact ISUM is making in fostering the use of supercomputing in research and development will continue to grow as this event continues to have presence throughout Mexico. Especially as the research community around the country continues to contribute their work in events and publications like this one focused on supercomputing. On behalf of the University of Guadalajara, congratulations to all the contributing authors of this publication for their work and commitment to advance research in science and technology. I cordially invite you to read through the articles of your interest.

**Dr. Miguel Angel Navarro Navarro**
University of Guadalajara
Executive Vice Chancellor

# PREFACE

The power of high performance computing (HPC) in research is transforming the way scientists work to solve some of the most challenging problems in Mexico today. These problems span the areas of weather forecasting, earthquake analysis, atmospheric science, materials science, among many others, and are helping to advance the impact of the scientific community today. The 2nd International Supercomputing Conference in México (ISUM 2011) focused in *MORE THAN RESEARCH presenting some of the most contemporary problems that are faced in HPC today*, thus helping to further enhance computing power to higher levels and to expand research nationally. During the past decade, Mexico has made significant strides in the application of supercomputing in science and technology research and in fostering its uses among young and seasoned scientists and technologists. The participation of this research community in ISUM 2011 is further testament to the value and interest in HPC to solve some of their most demanding research problems. It is rather motivating to see that the ISUM 2011 was successful in achieving the goals of fostering and expanding such uses of supercomputing in México.

The 2nd International Supercomputing Conference in México was held at *the Instituto Potosíno de Ciencia y Tecnología - Centro Nacional de Supercomputo (IPICyT-CNS)* where more than 400 researchers, technologists, graduate and undergraduate students from 43 universities participated, in addition to research institutes and centers from across the nation. The conference also welcomed 6 participating foreign institutions from the United States, Europe, and South America. There were 10 keynote speakers representing among the most established international supercomputing centers and institutes. Among the distinguished guests were the Supercomputing Directors from the Barcelona Supercomputing Center (Spain), Dr. Mateo Valero, and from the San Diego Supercomputing Center (U.S.A.), Dr. Michael Norman, both renowned scientists run two of the top centers in the world. The conference also had distinguished keynote speakers from the private sector including Altair, IQTech, Intel, IBM, EMC2, Grupo SSC, Cisco, and Silicon Graphics, who shared the latest technological tools in the industry. In addition to these distinguished speakers we had 43 research presentations, 3 round table discussions, 16 posters and 8 technical workshops that also contributed to the success of this conference. The conference was able to draw 48 research papers for review by the evaluation committee representing five academic institutions nationwide and representing the United States and France. In this second edition of the ISUM 2011 conference proceedings, the review committee selected 17 papers for publication using international criteria and covering a wide spectrum of topics related to HPC.

This second edition presents research conducted nationally in the areas related to HPC that include architecture, parallel processing, scientific visualization, grids, applications, and infrastructure. This set of studies was conducted by 46 authors representing 13 academic

institutions, research centers and the private sector in México and the United States, including notable research that undertakes performance analysis of a parallel genetic algorithm implementation on a cluster environment, and work that focused on the management and monitoring of large datasets on distributed computing systems for the IceCube Neutrino Observatory. These studies are representative of the type of research that you will find in this 2nd volume of the ISUM 2011 conference proceedings and our intent is to share the knowledge with the scientific and technological community interested in the latest supercomputing applications and/or research. In addition, this book includes an introductory article that focuses on the need to set a national agenda on supercomputing aligned to the national science and technology plan. As a national committee, we recognize the importance of making supercomputing an integral part of the growth of science and technology in the nation, and without its integration in the national science and technology plan, we foresee that growth in research in the country will be limited.

This publication includes among the most innovative research being conducted on supercomputing nationwide. The national committee sees these contributions as evolving and believes that as academia continues to foster the growth of HPC in research, this work will improve every year. Our intent is for ISUM to continue to provide a collegial space where researchers can share their work with colleagues nationally and internationally creating a scientific community centered around the uses of supercomputing to advance scientific work around the nation.

On behalf of the ISUM National Committee, I invite you to read about this pioneering work and I would also like to encourage you to participate in the upcoming International Supercomputing Computing Conference and share your research with the scientific community by presenting and submitting your work for publication. I wish to commend all of the authors who contributed to this publication and look forward to your contributions on future ISUM publications.

**Dr. Moisés Torres Martínez**
**ISUM National Committee, Chair**

# ACKNOWLEDGEMENTS

This publication could not be possible without the contributions of participants representing institutions from México, Latin America, United States, and European Union whom participated in this "2nd International Supercomputing Conference in México 2011" 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**

| | |
|---|---|
| Dr. Marco Antonio Cortés Guardado, | *Rector General* |
| Dr. Miguel Ángel Navarro Navarro, | *Vicerrector Ejecutivo* |
| Lic. José Alfredo Peña Ramos, | *Secretario General* |
| Mtra. Carmen Enedina Rodríguez Armenta, | *Coordinadora General de Planeación y Desarrollo Institucional* |
| Ing. León Felipe Rodríguez Jacinto, | *Coordinador General, Coordinación General de Tecnologías de Información* |
| Mtra. Alejandra M. Velasco González, | *Secretario, Coordinación General de Tecnologías de Información* |

**Centro Nacional de Supercomputo del Instituto Potosino de Investigación Científica y Tecnológica**

| | |
|---|---|
| Dr. David Ríos Jara, | *Director General del IPICyT* |
| Dr. César Carlos Diaz Torrejón, | *Director General, Centro Nacional de Supercomputo (CNS)* |
| MPS. Cynthia Lynnette Lezama Canizales, | *Sub-Directora, Desarrollo de Sistemas* |
| LDG. Sofía González Cabrera | *Responsable de Área de Eventos* |

**Special recognition to the ISUM National Committee 2011 because without their participation and dedication in the organization of this event, the ISUM 2011 couldn´t have been possible.**

| | |
|---|---|
| Andrei Tchernykh | (CICESE) |
| Carmen Heras | (UNISON) |
| César Carlos Diaz Torrejón | (IPICyT-CNS) |
| Cynthia Lynnette Lezama Canizales | (IPICyT-CNS) |
| Daniel Mendoza | (UNISON) |
| Fabiola Elizabeth Delgado Barragán | (UdG) |
| José Lozano | (CICESE) |
| Juan Carlos Chimal Enguía | (IPN) |
| Juan Carlos Rosas | (UAM) |
| Nicholas P. Cardo | (Berkeley Nat. Labs.) |
| Juan Manuel Ramirez Alcaraz | (UCOLIMA) |
| Lizbeth Heras Lara | (UNAM) |
| Verónica Lizette Dueñas | (UdG) |
| Manuel Aguilar Cornejo | (UAM) |
| Moisés Torres Martínez | (UdG) |
| René Luna García | (IPN) |
| Salma Jalife | (CUDI) |
| Salvador Castañeda | (CICESE) |
| Sofía González Cabrera | (IPICyT-CNS) |
| Yesica Vidal | (UNISON) |

# INTRODUCTION

## The Need to Develop a Strategic Supercomputing National Plan for the Advancement of Science and Technology Research in México

**Dr. Moisés Torres Martínez**
*University of Guadalajara*
*General Coordinating Office of Information Technologies*
*moises.torres@redudg.udg.mx*

### Abstract

*Investing in developing a strategic path for the growth of supercomputing in México is critical to the evolution of research and development in science and technology. This introduction presents the results of a round-table discussion with national and international leading experts in supercomputing whom identified the need to develop a strategic supercomputing national plan to advanced science and technology in México. It also provides a brief proposal to convene a national committee that is able to develop a plan aligned to the science and technology plan for the nation. The goal is to be more strategic about the future direction and growth of supercomputing to ensure a greater impact not only on the research conducted in the country but also on problems our researchers are able to solve for the betterment of our society.*

### Background

During the 2011 International Supercomputing Conference held in México (ISUM 2011), the national coordinating committee focused the discussion on the country´s need to develop a strategic supercomputing national plan to advance science and technology research that was also aligned with the science and technology national agenda. A roundtable discussion was convened to answer the question, why do we need a strategic supercomputing national plan in México? The roundtable included leading experts from across the nation and abroad. The participants included directors of supercomputer centers from México and the U.S., and researchers and technology experts. Among the roundtable participants were representatives

from the top leading academic institutions across México including: *Instituto Politécnico Nacional, Universidad de Guadalajara, Universidad Nacional Autónoma de México, Universidad Autónoma Metropolitana Iztapalapa, Universidad de Sonora, Universidad de Colima, Instituto Potosino de Investigación en Ciencia y Tecnología-Centro Nacional de Supercomputo, and Centro de Investigación Científica de Educación Superior de Ensenada*.

In addition, the panel included Dr. Marc Snir from the University of Chicago, Urbana Champaign, co-author of, *Getting up to Speed, the Future of Supercomputing,* a study *sponsored by the U.S. National Academy (2004,) and he is also Principal Investigator of the Blue Waters Project*. Dr. Snir shared his insights on the importance of being strategic about the growth of supercomputing in México. His contribution to the discussion suggested that the federal government should be rather careful in investing in Supercomputing Centers (SC) and ensure they are sustainable in the long-term and purposeful in addressing key problems that are of national interest. He also noted that "Mexico has problems to solve that require computing equipment at the TOP 500 level." The contributions from Dr. Snir's address along with those of the roundtable participants were enriching and identified the following critical needs: Provide access to High Performance Computing (HPC) to all researchers in the nation; Identify specific problems that can be addressed with HPC; The need to build capacity to maintain HPC equipment; The long term sustainability of Supercomputing Centers; And, the careful and strategic creation of these centers throughout the country. The discussion concluded with a commitment to continue the national dialogue to encourage the federal government through the *Consejo Nacional de Ciencia y Tecnología* (CONACYT) to sponsor the work of crafting a supercomputing national strategic plan aligned with the national science and technology research and development agenda.

## Why a Supercomputing National Plan?

Why is it important for México to have a strategic supercomputing national plan? We must consider that countries with the highest dominance in science and technology have a wide array of supercomputers to support research institutions, government agencies, and private industry. Among these countries are the United States, United Kingdom, France, Germany, China, and Japan (Top 500 List, 2010). This past decade China has grown significantly in this area. Fifteen years ago, China was not featured on the Top 500 list, yet at the end of 2010, they developed the fastest computer in the world (Tianhe 1A). They are the perfect example of a country that made considerable investment in science and technology and achieved remarkable results in their global competitiveness and high economic global standing in a short period of time. It is undeniable that their global competitiveness has risen dramatically due to their strategic investment in science and technology, and supercomputing is one of the key factors that has helped them advance research and development to become a global economic power. Similarly, Japan is another Asian country that has grown in competitiveness because of their investment in science and technology. In June of 2011, Japan produced the K Computer,

the fastest computer in the world, beating China´s Tianhe 1A supercomputer (Top 500 List, 2011). Such significant accomplishments by Eastern countries are demonstrating the West that they are willing to make the investments necessary to be global competitors in science and technology. Nevertheless, the U.S. continues to dominate with slightly more than 50% of the Top 500 List of supercomputers in the world.

In the spirit of recognizing and increasing México´s competitiveness in science and technology-related research and development (R&D), investments in the early 1990´s mark its presence in the supercomputing world with the inauguration of the first supercomputer (later named KanBalan) in the National Autonomous University of Mexico (UNAM). Since its activation, academic, governmental, and private institutions throughout the nation have increased their use of HPC. In 2005, México had five supercomputers ranked in the Top 500 List; however this recognition has steadily declined throughout the years. In 2011, México no longer appears in the Top 500 list, although it continues to make modest investment in supercomputer centers throughout the nation. For example, the most sophisticated centers today are the *Instituto Potosino de Investigación en Ciencia y Tecnología- Centro Nacional de Supercomputo (IPICYT-CNS)*, *Universidad Autónoma Metropolitana de Iztapalapa, Universidad Nacional Autónoma de México-KanBalan and the Instituto Politécnico Nacional-CINVESTAV.* A new National High Performance Computing Center is presently being proposed in the state of México, which is expected to be the fastest in the country and also rank in the top 500 list with approximately 100,000 cores. This center is supported by the state of México and CONACYT and received nearly 100 million pesos for its implementation, which is expected to open June of 2012. In addition, the Delta Project, a high capacity ring network between UNAM, IPN-CINVESTAV, and UAM-Iztapalapa will also connect the new center in México. This high capacity network is designed to create a powerful GRID between the aforementioned institutions to share and augment their computing power. This project, through the established and new centers, is the driving force in research in México. However, the challenge ahead for the Delta Project is to resolve how to provide access to researchers from outside institutions, since most centers except for the IPICyT-CNS are in the State of México. Even if computing centers were made available to researchers outside of the State of México, the country faces a connectivity challenge that prevents researchers from accessing these high performance computers remotely with great efficiency. While there has been significant growth in HPC, there are other challenges that we are facing for the strategic growth of supercomputing to advance science and technology research and development.

One of the main challenges the country faces that slows down the supercomputing growth is the overall national investment in science and technology, according to the Organization for Economic Cooperation and Development (OECD). México only invests 0.39% of their Gross National Product (GNP) in science and technology, placing the country in last place among OECD nations (OECD, 2009). This lack of investment significantly impacts the adequate growth of science and technology research and development, and consequently limits funding for HPC growth. The second challenge is the lack of broadband capacity throughout the country.

México has a long way to go to provide the appropriate bandwidth and network infrastructure that supercomputers require to build national and international GRIDS, or to provide access to such computing power to researchers across the country (Torres Martinez, 2010). The connectivity in the State of Mexico and federal district (DF) is among the most advanced in the nation, but the network infrastructure in the rest of the country is not as robust. This limitation in connectivity is unable to support GRID projects from states interested in connecting to the supercomputing centers in the state of México. The third challenge is the need to invest in and train a new generation of professionals and researchers in supercomputing. Universities throughout the nation seldom offer technical degrees that are focused in such supercomputing areas that include parallel programming, cluster computing, and GPU´s. As México continues to open new Supercomputing Centers, there will be a greater need to have a well-trained group of researchers and technologist to run, maintain, and conduct high quality research addressing the nation's wide array of problems. Such work should address issues of national security, climate, or transportation among others. This book presents other problems that are being resolved, which demonstrates the significant work being completed in supercomputing in addition to providing justification for more academic programs that can develop student´s skills to continue work in High Performance Computing.

A Supercomputing strategic national plan is critical to addressing present challenges and to ensure the country´s growth in supercomputing is well planned. A well-established plan can help save resources in the long-run, and this is especially important since the investment in science and technology is scarce. The strategic plan should include ways to increase collaborations among science and technology projects being conducted to save and maximize on the resources available. Such collaborations have the potential to accelerate research projects and obtain faster results. The country´s brightest minds in this discipline will have the opportunity to formulate a national plan that could have broad impact in the development of a long-term supercomputing research and development plan.

## A Proposal to Initiate a National Agenda for Supercomputing

México is presently facing a difficult phase in its history with the security issues that are of high priority to the federal government. This priority is obviously important to the well-being of the population; however, the country cannot ignore the importance of investing significantly in research and development for its economic growth. At the end of the day, this is important for the country´s economic growth. Although it has made significant strides over the past few years, México continues to fall short in comparison to other countries that belong to the OECD. Thus, the country needs more than ever to be strategic with its investments and ensure they are aligned with the national growth plan to advance research and development in the country.

This proposal calls for investment on a national strategic plan for supercomputing that is aligned with the R&D science and technology national plan. As mentioned earlier, the

following suggestions emerged from the roundtable discussion that took place in ISUM 2011, and reflect the thinking of experts in supercomputing from national and international academic institutions, and the private sector. The results of their thinking were as follows:

1) Create a national committee that can develop a Strategic Supercomputing National Plan to Advanced Science and Technology Research in México aligned to the science and technology national plan.

2) More graduate programs are needed nationally related to supercomputing to broaden the supply of researchers and technologists in this area.

3) A critical focus is necessary to examine sites where new supercomputing centers will emerge, to ensure researchers have full access to the computing power necessary to conduct their research.

4) Long term sustainability of all new and old supercomputing centers is important.

5) Seek strategies so that all researchers have access to high performance computing to conduct their research work.

These five points summarize the main round table discussion. Expanding on these discussion points, the following points are additional suggestions to consider during the process of creating a strategic supercomputing national plan to advance science and technology research in México:

1) Create a national committee that investigates and works on the strategic supercomputing national plan to advance science and technology research in México. This plan must present a global perspective on the future of supercomputing and its future in México. The plan should also incorporate the latest global innovations, and define the expected growth necessary to address the many challenges in our society. The committee must be composed of national experts in the various areas of supercomputing and have a broad vision, objective, and independent view from its respective institution.

2) The supercomputing national committee must work with national and international subgroups to build a plan that represents the new global tendencies in this arena.

3) Once the supercomputing strategic national plan is completed, it must be presented at three governmental levels: 1) Federal (relevant government agencies); 2) State (relevant government agencies); and, 3) National IT Associations (CANETI, AMIPCI,

AMITI). The objective for presenting the plan to these entities is to ensure the plan is well diffused nationally and to consider legislation and/or public policy as necessary to foster the growth of supercomputing in Mexico.

4) Financial support from CONACYT is necessary to allow the national supercomputing committee to work on a high quality and timely document that will advance High Performance Computing nationwide.

## Conclusion

The need to create a strategic supercomputing national plan emerges from the need to conduct high quality research in science and technology that is competitive at a global level. Since we know that science is not science unless you use some type of High Performance Computing to achieve faster results and quality research, it is recognized the importance of expanding its uses to continue to make significant leaps in science and technology (Torres Martinez, 2010). México's growth in the use of these tools to conduct research is steady, but we continue to lag in this area relative to developed countries that historically have made a significant investment to expand in their research in science and technology. This book presents work being conducted nationally in supercomputing that demonstrates the applications of HPC to address pressing issues. HPC is growing and new generations of researchers depend more and more on HPC to conduct their work. If as a country we do not foster the use of supercomputing to conduct high quality research, we will continue to move away from the global competitiveness in research and development. This proposal is neither the end nor the solution to solve the many challenges we face in science and technology research and development, rather it is a positive step toward reaching greater coherence on the many science and technology initiatives that will allow us to work better and smarter, and to using HPC.

This book presents seventeen articles in applications, architecture, grids, infrastructure, parallel processing, and scientific visualization that would not have been possible to complete and publish if not for the access researchers had to HPC. We know that as more researchers throughout the country continue to work on problems related to HPC, as a nation we will be more competitive globally. This can only be accomplished by investing intelligently and strategically in the growth of supercomputing. In the meantime, I invite you to read through the selection of papers that were presented at the 2nd International Supercomputing Conference in México (ISUM). We also cordially invite you to participate in future events and to contribute to future conferences and publications.

## References

Graham, S.L., Snir, M., Patterson, C. (2004). "Getting Up to Speed: The Future of Supercomputing", National Academies Press.

Organization for Economic Co-Operation and Development. (2009). http://www.oecd.org/home/0,2987,en_2649_201185_1_1_1_1_1,00.html

Report on the Top 500. (2008, 2010, 2011). http://www.top500.org/resources/reports Secretaría de Comunicaciones y transporte. (2009). Estrategia Nacional de Conectividad 2009-2010.

Torres Martínez, Moisés (2010). "ISUM Conference Proceedings: Transforming Research Through High Performance Computing." Universidad de Guadalajara, Volume 1, 1st Edition (ISBN: 978-607-450-348-7).

Torres Martínez, Moisés (2010). "The State, Challenges, and Future Directions of Supercomputing in México." Universidad de Guadalajara, Volume 1, 1st Edition (ISBN: 978-607-450-348-7), PP. 13-29.

ISUM 2011
2nd INTERNATIONAL SUPERCOMPUTING CONFERNCE IN MEXICO
CONGRESO DE SUPERCÓMPUTO

# APPLICATIONS

# Numerical Implementation and Analysis of an Encryption System

G. Flores-Eraña[1], J.S. Murguía[1, 2] and M. Mejía-Carlos[1]
[1] *Universidad Autónoma de San Luis Potosí,*
[2] *BioCircuits Institute, University of California, San Diego*
*gustavo.flores@cactus.iico.uaslp.mx, ondeleto@uaslp.mx, marcela.mejia@uaslp.mx*

## Abstract

*In this work, we present a numerical implementation of an encryption system based on a rule 90 cellular automaton [2,5]. We consider the encryption scheme used in Ref. 2, where the synchronization phenomenon of cellular automata has been applied to devise the two families of permutations and an asymptotically perfect pseudorandom number generator. A generating scheme consisting of three coupled transformations h is proposed to attain an asymptotically unpredictable generator under a random search attack [3]. This generator requires two initial keys of length of N and (N+1) bits and the new generated pseudo random sequences of N bits were analyzed using the NIST and DIEHARD statistical tests. It was found that the longer the length of the generated sequences the better is the quality of the random sequences we obtain.*

***Keywords***: *Cellular automata, basic unit cipher encryption system, pseudo-random generator.*

## 1. Introduction

The great advance in different fields has increased the interest to protect different kind of information. Thus, there is still a driving need to look for efficient algorithms to handle information in a secure way. Actually there are a large number of encryption systems, where its main objective is to protect information using an algorithm that makes use of one or more keys. In the implementation of many encryption systems have used different schemes with a chaotic approach in order to give extra strength and security information to be encrypted. For example, an encryption system which has proved reliable and easy to implement digitally is the encryption system based on the synchronization in cellular automata [2, 3, 5, 8]. In this paper it is presented its numerical implementation of this encryption system, where its main components are implemented with a similar matrix approach as was carried out in [1]. The organization of this paper is as follows. Section 2 gives a description of the encryption system considered. In Section 3, we describe the way to implement the main elements of the encryption scheme with help of the unit basic cipher. Whereas in Section 4 is discussed the numerical implementation of the pseudo-

random generator keys and the indexed families of permutations. We also include the analysis of the quality of the pseudo-random generator, where the generated sequences are evaluated statistically by the NIST and DIEHARD suites. Finally, the conclusions can be found in Section 5.

## 2. Encryption system

The encryption system considers the usage of cellular automata. A linear cellular automaton (LCA) can be considered as discrete nonlinear dynamical systems that evolve at discrete time steps. It consists of a chain of N lattice sites with each site is denoted by an index *i*. Associated to each site *i* is a dynamical variable $x_i$, which can take only *k* discrete values. Most of the studies have been done with $k = 2$ , where $x_i$ or 1.Hence, there are $2^N$ different states for these automata. The LCA considered evolves according to the local rule

$$x_i^{t+1} = \left( x_{i-1}^t + x_{i+1}^t \right) \bmod 2 \tag{1}$$

which corresponds to the rule 90. Figure 1 (a) illustrates the forward evolution of the LCA, where the symbol of a circled + represents a XOR gate and the connectivity of gates follows the automaton rule. One can see that the time, space, and states of this system take only discrete values. In addition, it is important to observe that the evolution rule of this LCA is determined by the initial conditions.

We consider the encryption scheme used in Reference [2], where the synchronization phenomenon of cellular automata has been applied to devise the two families of permutations and an asymptotically perfect pseudorandom number generator. The phenomenon of synchronization in coupled pairs of LCA is described in detail in Ref. [4], where it was found that a pair of coupled LCA with local rule 90 can synchronize if every pair of consecutive coordinates is separated by a block of $\left(2^k - 1\right)$ uncoupled sites.

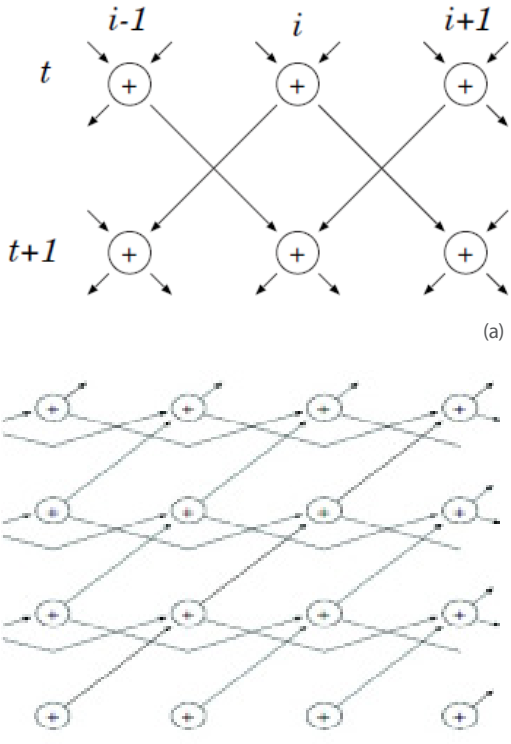



**Figure 1. (a) Forward and (b) backward evolution of the LCA**

Figure 2 shows the complete encryption system. Basically, the class of block cryptosystem considered transforms a plain text sequence **m** to a sequence **c**, called the cipher-text. The transformation $\mathbf{m} \mapsto \mathbf{c}$ is selected from an indexed family of permutations $\Psi = \{\psi_{\mathbf{k}} : M \to C | \mathbf{k} \in K\}$ by choosing an index **k** from the set of indices *K*. The sets *M, C* and *K* are all sets of binary words of length *N*, i.e. $Z_2^N$, where $Z_2 = \{0,1\}$ .The words in *M* and *C* are called the clear-blocks and cipher-blocks, respectively, whereas the words in the set of indices *K* are the encyphering keys. To disclose from the sequence of cipher-blocks, the cryptosystem also provides the family of inverse permutations $\Phi = \{\phi_{\mathbf{k}} : C \to M | \mathbf{k} \in K\}$ such that for every $\mathbf{k} \in K$ one has $\mathbf{m} = \phi_{\mathbf{k}}(\varphi_{\mathbf{k}}(\mathbf{m}))$. In this process, we demand to know the seed that was used to generate the pseudorandom sequence of keys, i.e., the complete encryption scheme is private where the encryption and decryption processes use the same deterministic generator that is initialized with a common seed. Notice that the plain text to be encrypted is generally much longer than the length *N* that is accepted by the family of permutations $\Psi$. In this case, we proceed to divide it into succession of blocks $\mathbf{m}^0$, $\mathbf{m}^1$, $\mathbf{m}^2$,… each of length *N*, and these blocks are then encrypted sequentially by using a different key $\mathbf{k}^i$ for each block $\mathbf{m}^i$. If the cipher-text is intercepted, this encryption system must avoid that the intruder be able to infer any information about the text. In this case is relevant to select as random as possible the succession of permutations, because the intruder must have to agree on a very long sequence of keys that determine the permutations [2]. This problem is solved by using a pseudorandom generator of keys.

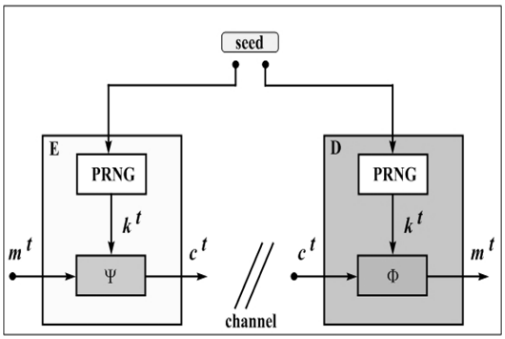



**Figure 2. The general encryption system with its main components: the indexed families of permutations and the pseudo-random generator keys**

## 3. The basic unit cipher

With the help of the synchronization phenomenon of LCA, was possible to implement in a flexible way the cryptography primitives, the pseudorandom generator of keys (function *h*) and the two indexed families of permutations $\Psi$ and $\Phi$. In the construction of these primitives we take into account an initial infinite sequence $\underline{x}^0$ , i.e.,

$$\underline{x}^0 = (\ldots, x_{-1}^0, x_0^0, x_1^0, \ldots, x_{N-1}^0, x_N^0, \ldots) \quad (2)$$

that evolves according the local rule (1), $\left(x_{i-1}^t + x_{i+1}^t\right) \bmod 2$ from $t = 0$ to $t = N = 2^\kappa - 1$ ,where $i \neq 0$ and $i \neq N+1$ , since and $x_{N+1}^t$ are externally assigned at each time $t$ .

Figure 3 (a) shows the space-time pattern from the infinite initial state (2), according to the evolution of the automaton rule. From the coupled coordinates, $x_0^0$ and $x_{N+1}^0$, we define the basic unit cipher (BUC) as the $N \times N$ square pattern in the lattice that consists of the $N$ time-running words $(x_1^0, x_1^1, \ldots, x_1^N), \ldots, (x_N^1, x_N^2, \ldots, x_N^N)$ . The first time-running word is distinguished by the name $\underline{k}' = (x_1^1, x_1^2, \ldots, x_1^{iv})$ . The words surrounding the square are $\mathbf{x} = (x_0^0, x_0^1, \ldots, x_0^{N-1})$ on the left side $\mathbf{c} = (x_{N+1}^1, x_{N+1}^2, \ldots, x_{N+1}^N)$ , on the right side $\mathbf{t} = (x_1^1, x_2^1, \ldots, x_N^1)$ , on the top, and $\mathbf{m} = (x_N^{N+1}, x_N^{N+1}, \ldots, x_N^{N+1})$ at the bottom.

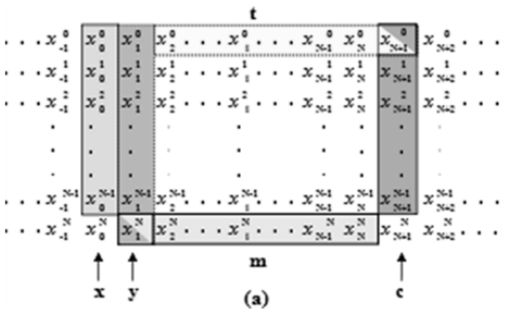
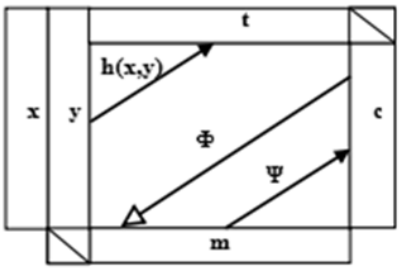



**Figure 3. (a) Space-time pattern from an infinite initial state according to the evolution of the rule 90.**

$$x_i^{t+1} = [x_{i-1}^t + x_i^t] \bmod 2$$




(b)

**Figure 3. (b) Primitives defined by the basic unit cypher. The functions and are determined by iterating the CA backward in time, whereas the function is computed by running the CA forward in time.**

Now, we can define and implement the family of permutations $\Psi$ and $\Phi$ , and the $h$ function, with the help of the BUC, where we identify the five main words x, y, m **c**, and **t** = *h* (x,y).

• Permutation Ψ

This permutation determines the cipher-blocks, when the LCA is iterated backward in time using the input words **x** and **m** , i.e., $x_{i+1}^t = \left(x_i^t + x_{i-1}^{t-1}\right) \bmod 2$ . The word located on the right side of the BUC, $\mathbf{c} = (x_{N+1}^0, x_{N+1}^1, \ldots, x_{N+1}^{N-1})$, is a cipher-block word, and it is obtained using the indexed family permutation $\Psi_{\mathbf{x}}$ , i.e., $\mathbf{c} = \Psi_{\mathbf{x}}(\mathbf{m})$ .

This is the inverse permutation and it is computed when the automaton is made to

run forward in time, but using the input words and . This permutation allows us to bring the word back to the plain text sequence **m** , i.e.,

$$\mathbf{m} = \Phi_{\mathbf{x}}(\mathbf{c})$$

• Function t = *h* (x-y)

In this case, the two words located on the left side of the BUC, **x** and **y** , are the input of the function **t** = *h* (**x,y**) . To generate this function, the automaton is also iterated backwards in time. The result of this function is on the top of the BUC and is identified as $\mathbf{t} = (x_2^0, x_3^0, \ldots, x_{N+1}^0)$ where $\mathbf{x} = (x_0^0, x_0^1, \ldots, x_0^{N-1})$ and $\mathbf{y} = (x_1^0, x_1^1, \ldots, x_1^N)$ .

The objects implemented in the BUC are shown in Figure 3 (b). Notice that the sequences of **y** and **m** share the symbol $x_1^N$ , whereas **t** and **c** the symbol $x_{N+1}^0$ . The backward evolution of the ECA is illustrated in Figure 1 (b), which is employed as an operation to devise the permutation $\Psi$ and function **t** = *h*.

## 4. Numerical implementation

The complete numerical implementation of the encryption system was carried out under the graphical programming language of LabVIEW, a trademark of National Instruments.

a. Pseudo-random number generator

The PRNG in its basic form was previously considered and implemented in Reference [3]. Its numerical implementation follows an algorithm that is shown in Figure 4(a). At first, the key generator requires two seeds, $\mathbf{x} = \mathbf{x}_0^{k+1}$ and $\mathbf{y} = \mathbf{x}_0^k$ of *N* and (*N*+1) bits, respectively. These seeds are the input of function **t** = *h* (**x,y**). Considering the seeds $\mathbf{x} = (x_0^0, x_0^1, \ldots, x_0^{N-1})$ , and $\mathbf{y} = (x_1^0, x_1^1, \ldots, x_1^N)$ , thus, the first number generated of *N* bits is the sequence output of function *h*, $\mathbf{t} = \{t_1, t_2, t_3, \ldots, t_N\}$ .

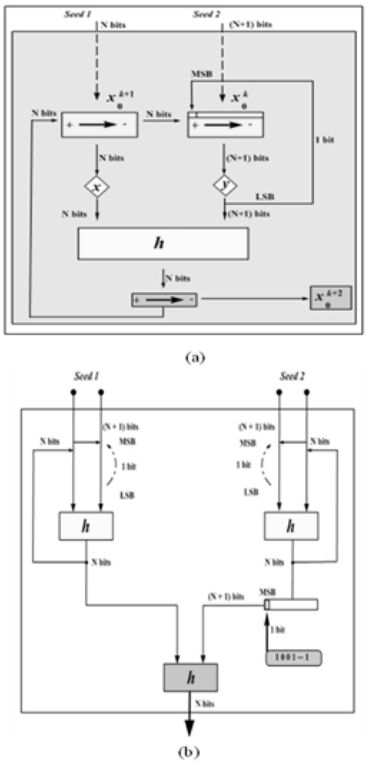



**Figure 4. (a) Basic form of the pseudo-random number generator. (b) Modified generator consisting of three coupled transformations. MSB and LSB correspond to the most significant bit and the least significant bit, respectively. Generation of a pseudo-random key with input $(\mathbf{x}, \mathbf{y})$ and output $\mathbf{t} = h(\mathbf{x}, \mathbf{y})$ .**

Now, this sequence is feeding back to the input, which becomes the next value of **x**, and the previous value of **x**, becomes the initial bits of the new **y**, where the missing bit is the least significant bit (LSB) of the previous , which becomes the most significant bit (MSB) of this sequence, and the same procedure is iterated repeatedly. As was said above, in order to compute the function **t** = *h*(**x,y**), it is required that the cellular automaton runs backward in time. Such situation is depicted in Figure 1(b). However, this way to compute the pseudo-random sequences is not efficient since it requires the application of the local rule of the automaton at all points in a lattice of the order of $N^2$, where *N* is the number of bits considered in the generation process.

To overcome this, Mejía and Urías [3] formulated an efficient algorithm that gets rid of the intermediate variables and produces Boolean expressions for the coordinates of the output sequence **t** = *h*(**x,y**) in terms of the input (**x,y**). This algorithm offers a Boolean representation of *h* , without intermediate steps, in terms of some "triangles" in the underlying lattice.

However, a generating scheme consisting of three coupled transformations *h* is considered to attain an asymptotically unpredictable generator under a random search attack. This proposal is shown in Figure 4(b), and it is explained briefly. Inside the new generator two copies of the basic transformation *h* are iterated autonomously from their initial words generating two sequences, $\{p_k\}_{k\geq 0}$ and $\{q_k\}_{k\geq 0}$ . The third copy, called the -map, is iterated in a slightly different manner, the function *h* in the *x*-map is driven by the autonomous *p*-map and *q*-map according to $x_k = h(p_k, q_k)$ .The three maps generate pseudo random sequences, but only the *x* sequence is released. In order to prevent predictability, the first two words are generated, used and destroyed inside this key generator, therefore they are not available externally. Since the sequences $p_k$ and $q_k$ have a length of *N* bits each and the required inputs of the *h* transformation must be one of bits and the other of *(N+1)*bits, the missing bit is obtained by applying an addition modulo 2 operation between the two respective LSB's that become the MSB's of their respective previous inputs of the maps. Despite that there exist different manners to generate this missing bit, we consider this way. In order to implement the above scheme we consider the matrix approach given in Reference [1], where the matrix form was introduced with the aim of computing recursively the pseudo random sequences $N=(2^K-1)$ of bits. In Figure 5 is depicted the numerical implementation of an example to generate a random sequence of 31 bits using three transformations.

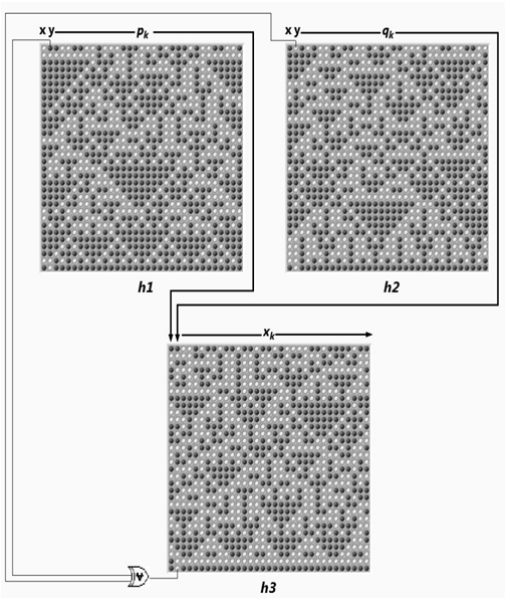



**Figure 5. Complete backward evolution of the LCA to generate a random sequence of *N*=31 bits with three transformations according to the modified generator.**

## b. Indexed families of permutations

In the same spirit as the pseudo-random number generator, we consider the matrix approach considered in [8] to implement the families of permutations. In order to compute the indexed family permutation $\mathbf{c} = \Psi_{\mathbf{x}}(\mathbf{m})$, we require two matrices, *PN* and *QN*, such that.

$$\mathbf{c} = \Psi_{\mathbf{x}}(\mathbf{m}) = \left[ \left( PN \times \mathbf{x} \right) + \left( QN \times \mathbf{m} \right) \right] \bmod 2 \quad (3)$$

These matrices have dimensions of $N \times N = (2^n - 1) \times (2^n - 1)$ for *n*= 1, 2. 3 The *PN* matrix is generated initially from the vector $\mathbf{p} = [p_1, p_2, \ldots, p_N]$, which constitutes the first row, and the components with position index , $j = (2^n + 1) - 2^{i+1}$ , have a value of 1 and 0 otherwise. The (*N*-1) rows are generated applying a right shift by one position of the previous row with a zero as its first value. In a similar way, the *QN* matrix is generated initially from two vectors with *N* elements, $\mathbf{w} = [w_1, 0, \ldots, 0]$ and $\mathbf{u} = [0, u_2, \ldots, 0]$, where the components $w_1$ and $u_2$ have a value of 1. The vectors **w** and **u** constitute the two first rows of the *QN* matrix and the other (*N*-2) rows are generated by applying the local rule 90 (an addition modulo 2 operation) of the two previous rows, with the elements of the previous row shifted to the right by one position, but again with a zero as its first value.

In Figure 6 is shown a numerical implementation of (3) for *n*= 5 , i.e., *N*= 31 bits. In this example are illustrated the two resulting matrices *PN* and *QN* .

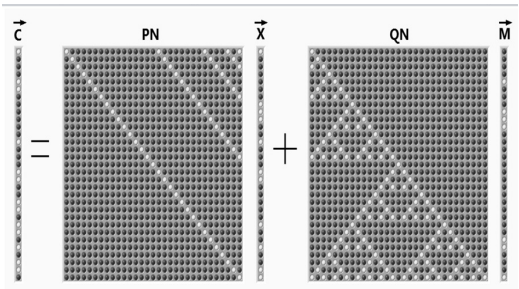



**Figure 6. A numerical implementation of the indexed family permutation, $\mathbf{c} = \Psi_{\mathbf{x}}(\mathbf{m})$ with $N = 31$ bits.**

On the other hand, the generation of the inverse permutation $\mathbf{m} = \Phi_{\mathbf{x}}(\mathbf{c})$ has a similar structure of (3), that is,

$$\mathbf{m} = \Phi_{\mathbf{x}}(\mathbf{c}) = \left[ \left( RN \times \mathbf{x} \right) + \left( TN \times \mathbf{c} \right) \right] \bmod 2 \ , \quad (4)$$

where the matrices have dimensions of $N \times N = (2^n\text{-}1) \times (2^n\text{-}1)$, for $n = 1,2,3,4\ldots$. In order to generate the *RN* matrix, we just rotate the *QN* matrix 90 degrees in the counterclockwise direction, i.e., the *RN* matrix is a transformed or rotated version of the *QN* matrix. In addition, the *TN* matrix can be generated from the *RN* matrix, since we just need to flip the rows of the second half of the *TN* matrix in the up-down direction.

Figure 7 illustrates a numerical implementation of (4) for $n = 5$, i.e., $N = 31$ bits, where the two resulting matrices *RN* and *TN* are depicted.

## 5. Statistical Test

Despite there exists several options for analyzing the randomness, in this work we consider the NIST suite and the DIEHARD suite to analyze the generated pseudo-random sequence keys. The main reason is that these suites have several appealing properties [5-7]. In addition, the source code of all tests in the suite is public available and is regularly updated [6]. In fact, in Ref. [6] is mentioned that the NIST suite may be useful as a first step in determining whether or not a generator is suitable for a particular cryptographic application.

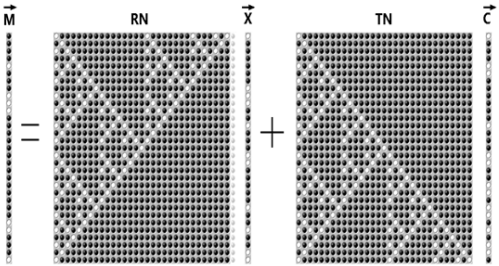


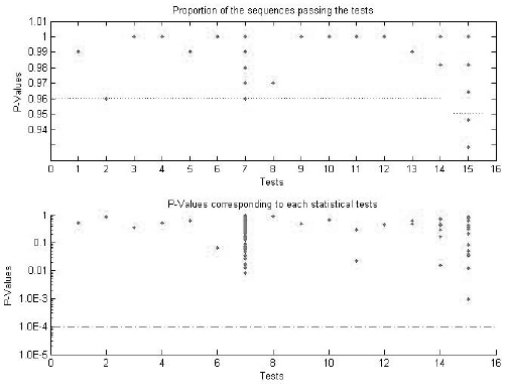

**Figure 7. A numerical implementation of the indexed family permutation $\mathbf{m} = \Phi_{\mathbf{x}}(\mathbf{c})$, with $N = 31$ bits.**

### a. NIST suite

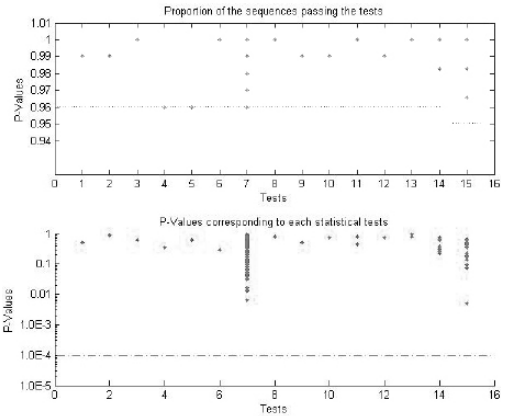
The NIST suite is a statistical package consisting of 15 tests that were developed to test the randomness of (arbitrarily long) binary sequences produced by either hardware or software based cryptographic random or pseudo-random number generators. These tests focus on a variety of different types of non-randomness that could exist in a sequence. Some tests are decomposable into a variety of subtests, and the 15 tests are listed in Table 1.

**Table 1. List of NIST statistical tests.**

| Number | Test name |
|---|---|
| 1. | The Frequency (Monobit) Test |
| 2. | Frequency Test within a Block |
| 3. | The Runs Test |
| 4. | Tests for the Longest-Run-of-Ones in a Block |
| 5. | The Binary Matrix Rank Test |
| 6. | The Discrete Fourier Transform (Spectral) Test |
| 7. | The Non-overlapping Template MatchingTest |
| 8. | The Overlapping Template Matching Test |
| 9. | Maurer's "Universal Statistical" Test |
| 10. | The Linear Complexity Test |
| 11. | The Serial Test |
| 12. | The Approximate Entropy Test |
| 13. | The Cumulative Sums (Cusums) Test |
| 14. | The Random Excursions Test |
| 15. | The Random Excursions Variant Test |




a




b

**Figure 8. Proportions and P-valuesT corresponding to (a) *N*=15 bits with one transformation, and (b) *N*=63 bits with three transformations. Dashed line separates the success and failure regions.**

For each statistical test, a set of P-values is produced. For a fixed significance level $\alpha$ , a certain percentage of P-values are expected to pass/fail the tests. For example, if the significance level is chosen to be 0.01 (i.e., $\alpha$= 0.01), then about 1% of the sequences are expected to fail. A sequence passes a statistical test whenever the P - value $\geq \alpha$ and fails otherwise. For each statistical test, the proportion of sequences that pass is computed and analyzed accordingly. It is not sufficient to look solely at the acceptance rates and declare that the generator be random if they seem fine. If the test sequences are truly random, the P-values calculated are expected to appear uniform in [0, 1]. For the interpretation of test results, NIST has adopted two approaches, (1) the examination of the proportion of sequences that pass a statistical test and (2) the distribution of P-values to check for uniformity.

To determine if the generated sequences are random or not, we have considered, for this statistical test, m = 100 samples of $10^6$ bit sequences, where each sequence has been generated from a randomly chosen seed, and the proportion must lie above 0.960150 ($\alpha$= 0.01) and P - $value_T \geq 0.0001$ . In order to investigate the performance of the generator, we analyze the generated pseudo-random sequences for *N*=15 and *N*=63 bits, considering one and three transformations, respectively. In Ref. [1] was carried out the evaluation for *N*=15, *N*=7, and *N*=31 bits, considering one and three transformations. In Figure 8 are shown the results from the NIST testing for (a) N=15, and (b) *N*=63 bits. We can observe that the generated pseudo random sequence with three transformations passes all tests, whereas fails in some with one transformation, but it is uniformly distributed.

### b. DIEHARD suite

This option comprises a battery of statistical tests for measuring the quality of a set of random numbers. These tests are exquisitely sensitive to subtle departures from randomness, and their results can all be expressed as the probability the results obtained would be observed in a genuinely random sequence [7]. Probability values close to zero or one indicate potential problems, while probabilities in the middle of the range are expected for random sequences.

Table 2 shows a summary of the results of this statistical suite for N=63 bits, with three transformations.

| Number | Test name | Result |
|---|---|---|
| 1. | Birthday spacing test | pass |
| 2. | Binary rank test | pass |
| 3. | The monkey test | pass |
| 4. | Count the 1's test | pass |
| 5. | Parking lot test | pass |
| 6. | The minimum distance test | pass |
| 7. | The 3D sphere test | pass |
| 8. | The SQUEEZE test | pass |
| 9. | The overlapping sums test | pass |
| 10. | The runs test | pass |
| 11. | The craps test | pass |

**Table 2. A summary of the DIEHARD test results.**

## 6. Conclusions

In this work we have described and analyzed the numerical implementation of an encryption system, which is based on a rule-90 cellular automaton. It was explained how the main components of the encryption scheme were implemented with a matrix approach. In addition, the performance of the generated pseudo random sequences is evaluated using two statistical suites, the NIST tests and the DIEHARD tests. We could also observe some statistical problems using one transformation, but as was discussed in [1], this PRNG can generate high-quality random numbers using one or three transformations as the size of keys is increased. In fact, in this work we analyze longer sequences than it was carried out previously. Since the complete numerical implementation is simple and fast, we consider that this encryption system could be embedded without problems in an existing communication system.

## 7. References

[1] J.S. Murguía, M. Mejía Carlos, H.C. Rosu and G.Flores- Eraña, International Journal of Modern Physics C, 21, 741 (2010).

[2] J. Urías, E. Ugalde and G. Salazar, Chaos, 8, 819 (1998).

[3] M. Mejía and J. Urías, Discrete and Continuos Dynamical Systems, 7, 115 (2001).

[4] J. Urías, G. Salazar and E. Ugalde, Chaos, 8, 814 (1998).

[5] M. Mejía Carlos,, Ph. D. thesis, Universidad Autónoma de San Luis Potosí, SLP (2001).

[6] Kenny, C., Random Number Generators: An Evaluation and Comparison of Random. org and Some Commonly Used Generators, Trinity College Dublin, Management Science and Information Systems Studies Project Report (2005).

[7] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, NIST Special Pub. 800-22 Rev. 1, http://csrc.nist.gov/rng/ (2008).

[8] Chang, W., Fang, B., Yun, X., Wang, S., and Yu, X., Randomness Testing of Compressed Data, Journal Of Computing, 2, 44-52 (2010).

[9] G. Flores-Eraña, J. S. Murguía, M. Mejía Carlos, and, H. C. Rosu, in preparation.

# Devising a Geographic Database (GDB) of the San Miguel River Basin, for Geoscience Applications

María del Carmen Heras Sánchez, Dora Guzmán Esquer, Christopher Watts Thorp
& Juan Saiz Hernández
Universidad de Sonora
Email: carmen@acarus.uson.mx

## Abstract

*A geographic database for the San Miguel river basin is being developed by integrating data from multiple sources for analysis and graphical representation of diverse physiographic features and hydroclimate phenomena such as rainfall, temperature, soil-evaporation, and topography among others. The projected database will allow us to combine digital maps and images along with thematic information and spatially-referenced vector data. Building a GDB with validated references further requires geographical referencing and validating processes in order to be able to accurately represent continuous data through discrete data structures that fit the mathematical models used in representing the physical phenomena at the study site.*

*Our methodology integrates systematically validated data, and as such, it may prove useful to scholars and geoscientists performing numerous modeling and numerical analysis such as tridimensional-temporal-thematic hydroclimate modeling, spatial-temporal rainfall analysis, and hydrological modeling of distributed parameters for basins similar to the San Miguel river basin.*

***Keywords:*** *GDB, GIS, digital elevation model, river basin, e-geoscience, modeling, mapping.*

## 1. Introduction

The analysis and representation of hydrological data in a GDB depends on the particular processes taking place inside the basin under study [1]. The latter predominantly include rainfall, evapotranspiration, ground-water recharge and runoff [2]. These in turn can be depicted through mathematical as well as computational models, with the aim of faithfully rendering the physics of each of the collected phenomena [3]. The significance of the data resulting from any ensuing simulations is directly correlated to the care given to the integration of the GDB; otherwise such data would be worthless [4].


**Figure 1. GDB build-up procedure diagram**

Devising a GDB for the San Miguel, Sonora, Mexico, river basin forms the basis of Maria del Carmen Heras Sanchez's ongoing master's degree thesis project: "Analyzing, processing and generating tridimensional-temporal-thematic models for the San Miguel, Sonora, river basin". The project's goal is to combine the available thematic information with digital elevation models (DEM[1]) in order to generate an assortment of models: luminous intensity, slope, topographical profiles, temperature, rainfall, and runoff among others, that would be impossible to realize without a properly validated and referenced GDB as the current methodology aims to achieve (See figure 1). The results yielded by the modeling, simulation and visualization processes arising from the integrated data from a GDB like this one are of vital importance in decision-making situations [5][6].

A series of computational tools have been designed for organizing, analyzing, processing, retrieving, displaying, modeling and exploring spatially referenced geodata [7] aimed at building a fundamental, complete and standardized information source as the one we are constructing for the San Miguel, river basin.

## 2. Study area: an overview

The San Miguel river drainage basin is located in the northern-central part of the state of Sonora, Mexico as depicted in figure 2, the main runoff contributors being the Cucurpe, Opodepe and Rayón municipalities. The polygon area enclosing it expands an area of 13,931 km$^2$. The Universal Transverse Mercator (UTM) geographic coordinate system's coordinates for the polygon vertices are as follow:

a (499836,3433694), b (573655,3433694), c (499836,3244973) and d (573655,3244973).

Figure 3 shows the physical makeup of the basin delimited by the watershed and the major streams that discharge onto San Miguel River.

1 A visual and mathematical representation of mean sea level-referenced earth's surface feature's elevation that allows us to characterize topographic relief and may even render a 3D image.

## 3. Data

We have integrated data from several sources, mainly from INEGI[2], hydrometereological stations (HS), and satellite images, among others. We have collected from INEGI both DEM and digital cartographic images (DCI) at 1:50,000 scale covering the span from -111°20'0'' W to -110°0'0'' W and from 29°15'0'' N to 31°15'0'' N.

### 3.1 Digital Elevation Models

In 2003 the *Continuo de Elevaciones Mexicanas* at 1:50,000 were created in order to integrate the different DEM based on the previous cartographic[3] maps created by INEGI at such scale. They are raster format models organized and identified by the topographic code number given to the particular area, and have a uniform vertical and horizontal coordinate system. The DEM files can be downloaded free of charge in formats such as .BIL, .HDR and .BLW (auxiliary files).

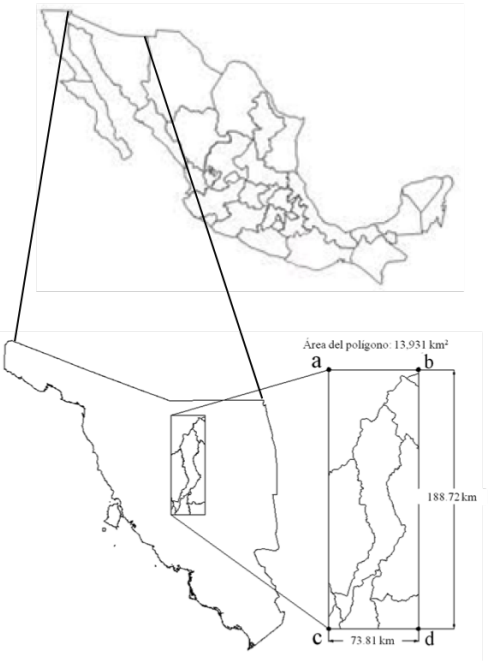



**Figure 2. Study area**

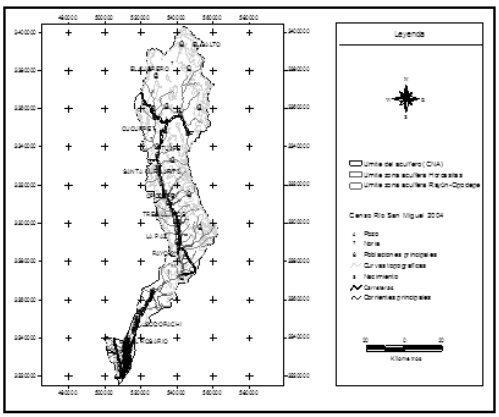



**Figure 3. San Miguel River's watershed**

2 Instituto Nacional de Estadística, Geografía e Informática, a Mexican federal government agency for the collection, classification and distribution of geographical information.

3 Cartography: Earth sciences discipline that formulates methods and techniques to convey geographic spatial information.

Elevation data are stored as 16 bits signed integer variables each, corresponding to a 1" by 1" cell and are arrayed in rows going from north to south and from west to east. The coordinates for the pivot cell, corresponding to the upper left corner of the grid[4], as well as the total number of rows and columns are stored in the accompanying auxiliary files.

Elevation information is reported in geographic coordinates; Z units are expressed in meters, datum corresponds to ITRF92, 1988.0 epoch or GRS80 ellipsoid.

In order to build the polygon corresponding to the study site, a total of 32 files were downloaded. The cells were laid in a mosaic-like fashion from where we could evaluate the resulting area of interest (see figure 4). The models that span this area are: H12A49, H12A59, H12A69, H12A79, H12A89, H12C19, H12C29, H12C39, H12B49, H12B59, H12B69, H12B79, H12B89, H12D11, H12D21, H12D31, H12B42, H12B52, H12B62, H12B72, H12B82, H12D12, H12D22, H12D32, H12B43, H12B53, H12B63, H12B73, H12B83, H12D13, H12D23 and H12D33.

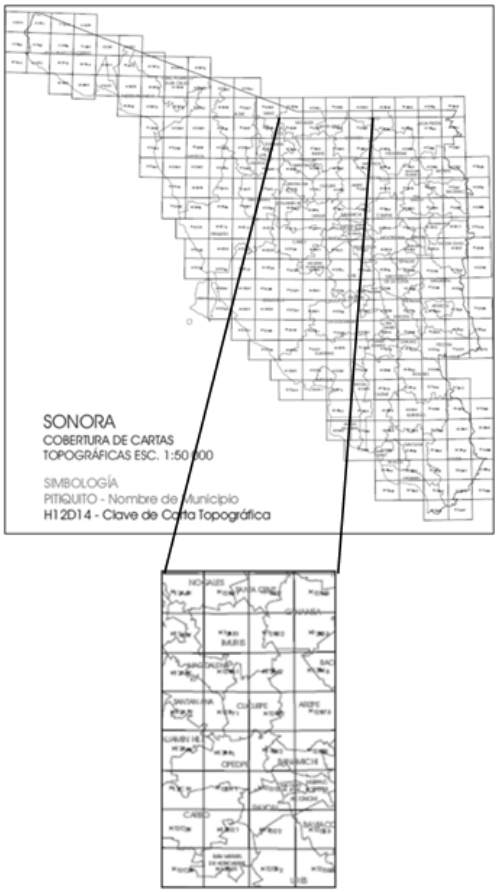



**Figure 4. Study area's mosaic**

## 3.2 Digital Cartographic Images

INEGI's 1:50,000 DCI's are compiled though photogrammetric methods from aerial photography snapshots, geodesic information and topographic surveys. Just as

4 Matrix-like bidimensional array commonly used in GIS systems.

any topography map, it utilizes a standard 20' longitude by 15' latitude format covering an approximate area of 960 kms$^2$, depicting the main land features along with conventional symbols and text highlighting points of interest. The DCI's data structure is raster-based and the files are stored in GIF and TIFF formats, which can be readily assembled in a Geographic Information System (GIS). DCI files share the same code name as DEM files.

### 3.3 Hydrometereological stations

An HS is an instrument designed to measure and register a series of meteorological variables such as rainfall, temperature, and soil humidity, among others, used in calculating the transference of water and energy between land surface and the lower atmosphere.

There are 45 HS installed in our study area collecting hourly data, for a total of 24X365X45 records per year. Data are stored as alphanumeric values in space-separated columns fashion, readily accessible for storage and inspection through spreadsheet applications. Nevertheless, those data still need to be preprocessed and georeferenced in order to be integrated to the GDB using a GIS. See section 4.2.

### 3.4 Data attributes specification

Once data were collected and analyzed, we devised the strategy for structuring the GDB information based on the study area's geographic span, spatial resolution, required level of precision, amount of hydroclimatic records required, data format for the collecting stations and coordinate system suitability, among others.

#### 3.4.1 Cartographic projection and coordinate system

The coordinate system's main purpose is to identify a given point's position in space with regard to a reference point. From an Earth's point of view, it is given by a pair of coordinates relative to the reference point that correspond to the junction between the Equator[5] and the Greenwich[6] meridian[7]. Coordinates are referred as latitude[8] and longitude[9] and are expressed in degrees, with minutes and seconds for finer resolution, or its decimal equivalents. Latitudes have a range of ± 90 degrees and ± 180 for longitudes.

Owing to the fact that gravitational forces warp the Earth's shape into a geoid (see figure 5), instead of the idealized geometrical figure of an ellipsoid, a series of cartographic projections have been devised to represent Earth's features on a plane. The most used projection for cartographic, geodesic[10] and navigational purposes is the Mercator

5 Equator: an imaginary circle perpendicular to the Earth's axis and equidistant from both geographic poles. It delimits the Northern hemisphere from the Southern hemisphere.

6 Greenwich meridian is by convention the starting meridian. See below.

7 Meridian: imaginary circles passing through the poles that helps to demarcate the time zones.

8 Latitude: angular distance of an Earth location from the equator.

9 Longitude: angular distance of an Earth location with respect to the Greenwich meridian.

10 Geodesy: scientific discipline that deals with the measurement and representation of the Earth.

projection. It is a cylindrical projection formulated by Gerardus Mercator in the 16th century, and although it exhibits some distortion toward the poles, the latter can be adjusted using an appropriate datum[11].

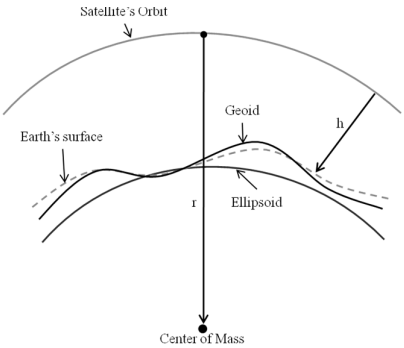



**Figure 5. Earth's shape's aproximation**

We opted to use the UTM geographic coordinate system and the World Geodetic System 84 (WGS84[12]) ellipsoid as basis for the coordinate system. References are given in metric units; therefore there is no need to convert from degrees to meters so distances and areas can be immediately calculated. We are confident our study area lies on the central strip of UTM projection Zone 12 (see figure 6), therefore the chance of these projections being out of phase are negligible.

11 Datum: is a set of reference points on the Earth's surface against which position measurements are made, and (often) an associated model of the shape of the earth (reference ellipsoid) to define a geographic coordinate system.

12 WGS84 is a reference ellipsoid for altitude data, and is the reference coordinate system used by the Global Positioning System.


**Figure 6. Mexico's UTM zones**

### 3.4.2 Spatial resolution

For any given physiographic-related undertaking it is necessary to consider the altitude for the reference area. In our case, we base our definition of spatial resolution with the aid of the MDE's at hand to render the projection of remaining data integrated to the GDB [8].

An image's spatial resolution is indicative of the pixel's size expressed in terms of dimensions on the terrain. Even though a pixel is a graphic element, it is commonplace to refer to pixel's size in regards to the grid's cell's size in a raster file. Pixel size and cell size are used interchangeably to refer to a grid's primary building block [9].

GIS systems do not assume pixels are square since spatial resolution, as a whole, is determined by X and Y coordinates scope and the number of rows and columns forming the grid [10]. Using datum UTM12N and the WGS84 ellipsoid, cell's dimensions for our study area were defined as follows:

Spatial resolution X:

Xmax – Xmin / number-of-columns =

595218.3 - 467611.7 / 4800 = 26.6

Spatial resolution Y:

Ymax – Ymin / number-of-rows =

3457739.9 - 3235730.0 / 7200 = 30.8

Cell's spatial resolution X and Y = 26.6X30.8

Therefore, spatial resolution means that an image's fine details are distinctively resolved: as smaller terrestrial areas are represented by a given pixel, finer detail can be picked-up and greater spatial resolution is achieved.

### 3.5 Hardware and software resources

We aimed to integrate vast amounts of tabular data for several hydroclimatic features, as well as cartographical thematic information, plus daily images, in addition to multivariate data that are combined by a complex set of algorithms to deliver the expected indicators.

#### 3.5.1 Geographic Information System (GIS)

A GIS system is a computer-based system designed to manage and present data with reference to geographic location data. Data can be either collected from remote sensors and satellites, or numerically calculated and analyzed. Most of the information stored in a typical GIS system is portrayed in two dimensions and only a few of them analyze and render three-dimensional data, which requires substantial computing power and storage capabilities.

There are a fair amount of GIS systems that provide the applications and soft tools required for creating interactive queries, analyzing spatial information, edit data and maps, and present the results of all these operations. What tells them apart is cost, required computing power and economic resources required for sustained and optimal operation. The application selected was Idrisi Taiga developed by Clark Labs at Clark University, Worcester, MA, that has among its many features porting ability among different geodata formats, contains fast algorithms, can be installed in diverse platforms, and it is reasonably priced.

#### 3.5.2. Geographic database data models

Having such a diverse set of multivariate data, we pondered how better organize them so their handling will not become a difficult task. We decided to use the raster, and vector data models, as well as vector-tagged alphanumeric text files to represent the characteristics of the San Miguel river basin.

A computational model is a numerical representation of a given natural phenomenon chosen according to its data structure and file type. Case in point: because of its grid-based organization, the raster model was selected to represent elevation.

Vector data render a digital representation of discretized phenomena using points, lines or polygons with well defined geographical limits that form overlapping information layers such as topographic contour lines

(isohypse) charts, hydrographic information, and localities information among others.

As for the vector-tagged alphanumeric text files, they show different context-related attributes relative to the vector models: geodesic reference points, geographic names (toponyms and localities), reference sampling points from natural resources maps, and thematic units descriptions among others.

### 3.5.3 Computer infrastructure

The GDB's design requires substantial computing power. It was implemented by means of a server with 16 GB in memory and 4TB for mass storage. It was acquired with the support of Telemetry and Geographic Information Systems Applications Development Centre at the Mines and Civil Engineering Department and the High-Performance Computing Area (ACARUS) at Universidad de Sonora.

## 4. Results

The GDB's design required a detailed analysis of the spatial domain to be charted as well as a comprehensive understanding of the available data. From both sources we devised the most suitable spatial resolution and coordinate system used in mapping out the data. Our methodology considered an iterative pre-processing phase of temporal input data that rendered standardized, co-registered, and georeferenced data files.

Each newly generated file contained a complete set of metadata explicitly stating its origin, data type and/or model it held, size, generating process and so forth. In order to identify the kind of data being handled with the geographic space under study a set of prefixes was chosen. See table 1.

**Table 1. Image data files ID prefix**

| Prefix | Vector or Image files |
|---|---|
| icd | Digital cartographic image |
| ofd | Digital orthophotography |
| mde | Digital elevation model |
| uss | Soil use |
| tvg | Vegetation type |
| ehm | Hydrometeorology stations |
| ipt | Thiessen polygon-based interpolation |

Since models use temporal-thematic data files that cover the same geographic space but represent completely different phenomena, and yet hold numeric values that are handled and displayed in much the same way, an identifying scheme was formulated to name such files using the first three consonants (in Spanish) of the phenomenon being represented. See table 2. A file's name would begin with such prefix, say **tmp**, followed by a dash and date in **ddmmyy** fashion. The files extension also plays an important role and is set straightforward by the Idrisi application depending on the kind of model being used and the process it was subjected to.

### 4.1 Mosaic generation and polygon extraction

The procedure began by processing the DEM's . BIL files along with its corresponding metadata files, .HDR and .BLW,

through Idrisi's GENERICRASTER command. It resulted in a set of 32 raster files containing the grid with the elevation data for a segment 20' longitude long by 15' latitude wide. The segment actually runs from -111°20´0´´ through -110°0´0´´ West longitude, and from 29°15´0´´ through 31°15´0´´ North latitude, spanning an area 2 2/3 degrees$^2$. Each file was labeled with the code for the area it covered.

**Table 2. Thematic data files ID prefix**

| Prefix | Represented phenomenon |
|---|---|
| tmp | Temperature |
| prc | Rainfall |
| hmd | Humidity |
| trn | Plant transpiration |
| vpr | Evaporation |

Next we used Idrisi's CONCAT command to form the mosaic with the previous 32 raster files, maintaining the original coordinate system. Later we used the PROJECT command to modify the coordinate system as well as the datum: from a lat/long convention to a UTM12N and set the X and Y minima and maxima. Afterwards, we used the WINDOW command to pull out the precise polygon spanning the study area by specifying the top-left and bottom-right corners in metric coordinates. The resulting file's name is MDERSM (Spanish acronym for digital elevation model for San Miguel River). See figure 7.

In much the same way the DCI images were subjected to import (using the GEOTIFF/TIFF command), concatenation, reference frame switching, and polygon extraction processes. The end file's name is ICDRSM.

### 4.2 Temporal-thematic hydrometereological information pre-processing

The GDB shall include the average daily values of temperature, rainfall and soil humidity from June 1st 2010 up to September 30th 2011. Those averages are calculated from the thematic tabular files generated at each of the 45 HS stations and stored as text files following the naming conventions previously outlined. There will be a total of 487x45 per theme files generated for the study period.

### 4.3 HS stations digitizing process and space-temporal image generation

Once the HS files were in place, using the MDERSM file for reference, we took advantage of the DIGITIZE command to situate the stations within the San Miguel river's DEM. It created a vector file.

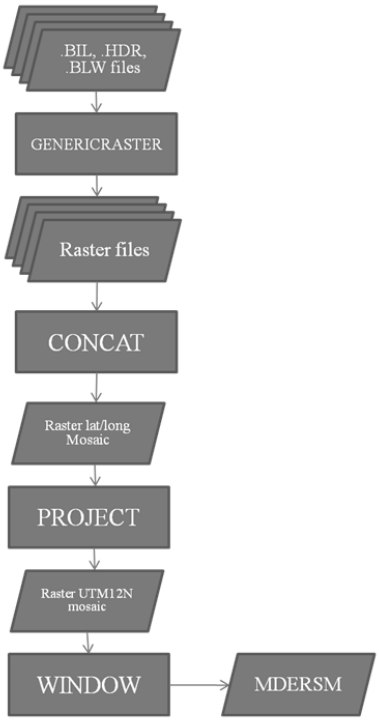



**Figure 7. Polygon's DEM construction diagram**

With the aid of the DCI images, this last file went through a georeferencing process to try to match its contents with either hydrographic features or geodesic points. If not, the digitizing and matching processes were repeated until we came up with a fully georeferenced HS vector file.

We then subjected the resulting HS vector file to a Thiessen polygon-based interpolation process.

The latter step produced images as well as vectors depicting areas, which can then be used in time series analyses and thematic-temporal model simulations for a given phenomenon under examination.

As is the case for the thematic information, we would be generating as well 487x45 georeferenced double-precision binary files during the study period. See figure 8.

### 4.4. Assessing the DEM's spatial resolution

Since much of the usefulness of the GDB rests on having precise altimetry data, we took special interest in determining the spatial resolution of the cells shaping up the MDERSM grid. In order to corroborate the resolution resulting from the general formula given in section 4.3.2, short experimental runs were made with sections of the polygon.

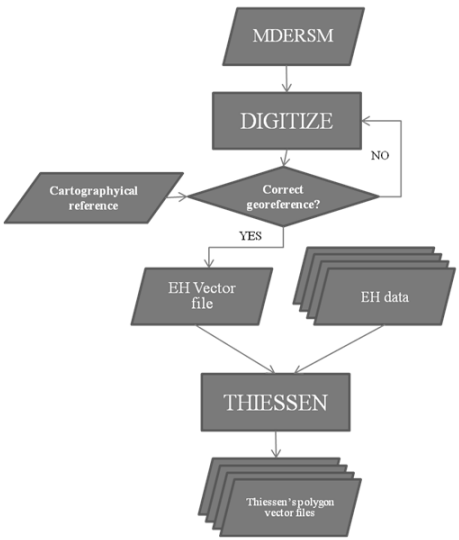



**Figure 8. Thiessen's polygon creation procedure diagram**

Aiming to validate the results of our methodology, we asked INEGI's cartography experts for feedback on our mosaic generation and polygon extraction procedures. Their opinion was that a given pixel must have square dimensions. In our estimation, their analysis is correct as long as we use the geographic coordinate system.

Several tests were run with different datums: in order to calculate the area expressed in meters squared of a surface whose sides were an arc of a second long, we converted two pairs of geographic coordinates to both North American Datum of 1927[13] (NAD27) and North American Datum Ellipsoid WGS84 from 2007 (NAD83.2007[14]). The results are as shown in table 3.

When converted, the metric resolution is 26.8 * 30.5 meters for NAD27 and 26.8 * 30.6 meters for NAD83.2007, although the pixel is still a square of 0.0002777777778 by 0.0002777777778 degrees.

**Table 3. 1" coordinate systems comparison**

| Geographic coordinates | NAD27 | NAD83.2007 |
|---|---|---|
| (-110°0'0",31°0'0") | (595408.7,3430042.9) | (595468.3,3430031.0) |
| (-110°0'1",31°0'1") | (595381.9,3430073.4) | (595441.5,3430061.6) |

13 NAD27 is the horizontal control datum for the United States that was defined by a location and azimuth on the Clarke ellipsoid of 1866, with origin at Meades Ranch (Kansas) survey station.

14 NAD83.2007 is the latest control datum for the United States, Canada, Mexico, and Central America, a refinement of the NAD83 datum using data from a network of very accurate GPS receivers at Continuously Operating Reference Stations (CORS).

Later we performed the conversions for the whole mosaic and obtained results shown in table 4.

**Table 4. Whole mosaic coordinate systems comparison**

| Geographic coordinates | NAD27 | NAD83.2007 |
|---|---|---|
| (-110°0'0",31°15'0") | (595280.4,3457543.9) | (595218.3,3457739.9) |
| (-111°20'0",29°15'0") | (467670.1,3235537.1) | (467611.7,3235730.0) |

When the later set was converted, the metric resolution for the cell is 26.59 * 30.83 meters for both datums, yet we observed a southeastern shift of the NAD83.2007 coordinate system's origin (see figure 9). In our estimation this corresponds to a 58.4 to 62.1 meters shift in longitude (X) and a -192.9 to -196 m shift in latitude (Y). We speculate this inconsistency owes to the fact we were performing the calculations based on dissimilar geometric figures.

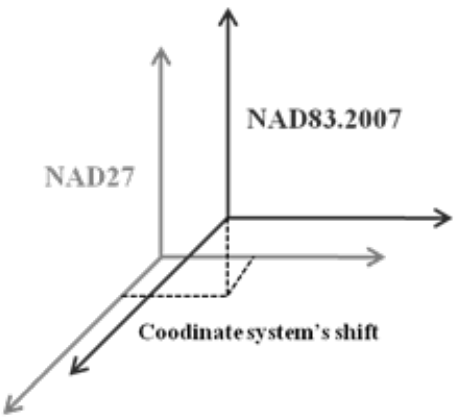



**Figure 9. Coordinate system's shift**

If a GIS takes an origin and from it tries to extend the coordinate system to the rest of the points making up a given image, we tried to manually fine-tune the resolution to a 30 * 30 meters cell. However, the maximum coordinates outcome form the calculations did not match the original coordinates, not even the geometric coordinates once the conversion was performed. See table 5.

**Table 5. Grid's final coordinates w/30*30 cell resolution**

| Grid's origin coordinates | |
|---|---|
| NAD83.2007 | Geographic |
| (467611.7, 3235730.0) | (-111°20'0",29°15'0") |
| Grid's maximum coordinates – 4800 * 7200 cells | |
| NAD83.2007 | Geographic |
| (611611.7,3451730) | (-109°49'42",31°11'39") |

This seems to suggest a given pixel from INEGI's DEMs is not truly square, and depending on the metric coordinate system used plus the specific area being sampled, there could be variations on the pair of resulting coordinates, as well as in the results for areas and distances calculations.

## 5. Conclusions

A validated GDB was integrated using data from multiple sources and graphical representation of diverse physiographic features and hydroclimate phenomena. A strategy for structuring, identifying, handling and increasing or improving the GDB information was also devised. An iterative digitizing and georeferencing process was formulated to match remotely collected hydroclimate data with DCI images. Spatial resolution calculations were performed that would allow us to identify discrepancies in the coordinate system being used, and adjust the latter to improve the geographical information representation.

Our methodology may prove useful to researchers performing various modeling and numerical analysis for similar river basins.

## 6. Acknowledgements

The authors would like to thank Raúl Hazas-Izquierdo, M. Sc., for his kind help in revising and correcting the manuscript's English version. They would also like to thank our colleagues at the following Universidad de Sonora departments for their technical and material assistance: Telemetry and Geographic Information Systems Applications Development Centre at the Mines and Civil Engineering Department, Geology Department, and the High-Performance Computing Area (ACARUS). Lastly, the authors would like to thank the anonymous reviewers whose perceptive recommendations were taken into account to improve this paper.

## 7. Bibliography

[1] Llamas, J. (1993). *Hidrología general: Principios y aplicaciones*. Servicio Editorial de la Universidad del País Vasco. Bilbao. 635 pp.

[2] Linsley, R., Kohler, M. y Paulhus, J. (1988). *Hidrología para Ingenieros*. Mc Graw-Hill Latinoamericana. D.F., México, 386 pp.

[3] Burrough, Peter A. y McDonnell, Rachael A. (1998). *Principles of Geographical Information Systems*. Oxford University Press. 333 pp.

[4] Bonham-Carter y Graeme F. (1994). *Geographic Information Systems for Geoscientists: Modelling with GIS*. Pergamon/ Elsevier Science Publications. 398 pp.

[5] Dercourt, Jean y Paquet, Jacques (1984). *Geología*. Editorial Reverté. Barcelona. 423 pp.
[6] Jones, Christopher B. (1997). *Geographical Information Systems and Computer Cartography*. Addison Wesley Longman Limited, Edinburgh Gate, England. 319 pp.

[7] Crawford, N. H. and Linsley, R. K. (1966). "*Digital simulation in hydrology: Stanford Watershed Model IV.*" Technical Report No. 39, Stanford University, Palo Alto, California.

[8] Hutchinson, M. and J. Gallant (2000). "*Digital elevation models and representation of terrain shape*", in *Terrain Analysis. Principles and Applications*, Wilson, J. and Gallant, J., editors, John Wiley & Sons, Inc. New York, USA.

[9] Thompson, J. A., J. C. Bell and C. A. Butler (2001). *Digital elevation model resolution: Effects on terrain attribute calculation and quantitative soil–landscape modeling*. Geoderma, 100. pp. 67–89.

[10] Hengl, T., S. Gruber and D. P. Shrestha (2004). *Reduction of errors in digital terrain parameters used in soil–landscape modeling*. International Journal of Applied Earth Observation and Geoinformation, 5. pp. 97–112.


# ARCHITECTURES

# Multi Agents System for Enterprise Resource Planning Selection Process Using Distributed Computing Architecture

Augusto Alberto Pacheco-Comer, Juan Carlos González-Castolo
Universidad de Guadalajara, Centro Universitario de Ciencias Económico Administrativas, Doctorado en Tecnologías de Información
apacheco@cucea.udg.mx, jcgcastolo@cucea.udg.mx

## Abstract

*This paper shows a review research literature regarding multi-agents systems related with enterprise resource planning systems and high performance computing. A multi agent system model proposal for enterprise resource planning selection process using distributed computing architecture is presented. The enterprise resource planning selection process is divided in six steps. The proposal model is an aid for evaluation process and has five different types of agents. A distributed computing architecture is proposed as a mean to improve the performance in the use of proposal model.*

***Keywords:*** *Enterprise Resource Planning, Multi agent system*

## 1. Introduction

The process of selecting an Enterprise Resource Planning (ERP) system is a complex problem which involves multiple actors and variables, since it is a decision-making process which is characterized as unstructured type [1, 2]. Multi Agent Systems (MAS) are designed to assist in the process of solving complex problems [3], so its use can be recommended for modeling and simulation of the selection process of ERP systems.

In this paper we present literature related to MAS and ERP systems. How this two information system types could be related. Following with a propose MAS architecture for ERP selection process using distributed computing architecture to leverage high computer power (HPC) to process a simulation prototype MAS proposed. Finishing with a couple suggestions for future research topics regarding HPC and MAS aligned to ERP implementation.

The paper is organized beginning in section two with a research literature review regarding MAS, Distributed Computing System and their relation with ERP systems. Section three presents a MAS proposed architecture for ERP selection process. Section four presents discussion and conclusion regarding the proposed model; how the model could be include it on Distributed System research area, ideas for future research work and next step for the model.

## 2. Research literature

Agent-based computing is one of the tools used by computers intelligences to find solutions to complex problems. [3]

MAS seek to address the trends in computer science such as ubiquity, interconnectedness and intelligence [4]. An agent is a computer system capable of undertaking activities independently to benefit its owner or user [4]. A multi agent system consists of a set of agents that interact within an environment, which act on behalf of the objectives and motivations of their owners. Agents have the ability to cooperate, coordinate and negotiate looking at all times comply with the purposes for which they were created [4].

Intelligent agents have the ability to make decisions, those decisions are defined by their inherent properties, which are: reactivity, pro-activity and sociability. These properties are intended to enable the agent to meet the objectives for which they were designed, following rules of behavior that enable them to communicate with their environment [4].

In the development of multi-agent systems, an important point is how the agents represent users, both in physical appearance, as in the activities. In a study by King [5], was identified in a preliminary way for people to better identify an agent when it has human physical characteristics because these characteristics are perceived as a sign of greater intelligence. And if the face of representation allows eyes blink, the perception of intelligence is higher [5].

Performance requirements embedded in agents must be clearly defined, as mentioned by Feber [6]. He indicates that the requirements of organizations can be represented by a set of agents. He identified four types of behavioral requirements, which are: Requirement of behavior for the individual role, for the interaction within groups, for successful communication within groups and for interaction between groups [6].

Moreover Zhang [3] mentioned that the resolution of complex problems include how the subsystems of the problem can be managed, at times, autonomy, intelligence, relevance and control of their own state and behavior. In his work, He used, as an example of the use of multi-agent system, hierarchical distribution of the coordination of industrial process control, where the control system is divided into four agents: Agent for scheduling system administration, control agent for work in process for each of the workstations, the agent for field monitoring and the environment agent. The management scheduling system agent has communication with ERP system end user (Figure 1) database to receive work orders [3].

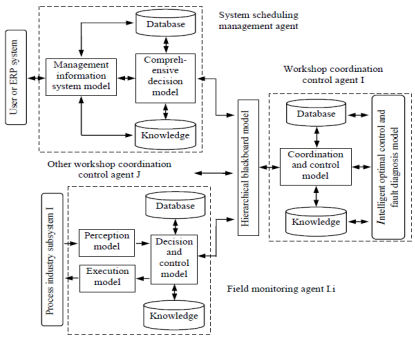



**Figure 1: Model architecture of MAS, Zhang [3]**

In this agent architecture [3], the lowest level agent is the field monitoring agent, which is responsible for following up the status of each existing machine and materials in work stations.

The next level of management correspond to the control of the operations agent which is performed by different workstations [3], this agent has interaction with field agents and is responsible for accepting jobs submitted by the senior agent. The senior agent is the management scheduling system agent. It is responsible for receive production orders from the ERP and send them to the work stations agents. Also is responsible to conduct negotiations between workstations agents and balance the work load when a station is unable to perform their assigned tasks [3]. In the case study presented by Zhang, the mechanisms of cooperation between agents are very important since most number of agents in the MAS is field agents, followed by station management and only one senior agent, It is this agent which control the whole industrial process and interface with other company's information systems. A clear definition and delineation of roles and behaviors of each agent corresponds to the complicated process of developing a MAS, which correspond to a model of the whole system [3].

Another example of multi-agent system related to ERP is presented by Kehagias [7]. He mentioned the connection with the ERP database designing a new functional layer. This layer corresponds to a MAS architecture which control sales orders and give a recommendation on how could be fulfilled. This system consists of five agents: the client orders agent, the recommender agent, the customer profile identifier agent, the inventory profile identifier agent and the supplier profile identifier agent [7]. The goal of the system is to obtain a recommendation of how the customer orders should be fulfilled, but also, each of the agents should recommend additional actions to perform, such as: the generation of purchase orders according to statistics inventories and time range, addiction to customer orders based on their purchase history and recommend suppliers according to products stock, statistical spend, lead times and historic operations with that suppliers. One aspect that seems important to stress is the proposed layer structure. If we take the network layers of the OSI model and we compare that model with this layer model, we could indicate which agents are integrated in which layers. This layered structure, as the author mentioned, allows the possibility to superimpose MAS layer over different types of systems, according to the characterization of systems proposed by Laudon [8].

Reviewing some of the case studies using agents, we can identify that the development and operational framework is not clearly defined, in this respect, the work of Flores-Mendez [9] allows us to learn more about this research area because He introduces us to the different terminologies used in this topic and provides us with information regarding the different groups that are working in defining what should a MAS architecture must have, the language with which the agents must communicate and how the ontologies used by agents could be defined

Kendall [10] mentioned that MASs should be designed based on principles of development of valid systems, rather than an architecture tailored to each project. This would allow the design and development of MASs based on a proper software architecture and design. He suggested a layer design of agents because a layered architecture allows: a) upper layers with more sophisticated behaviors, this layer should dependent on the capabilities that lower layers can provide. b) layer only depend on their own surrounding and neighbors elements and layer. And c) layer promotes bilateral communication between neighboring layers. The layer structure suggested: Sensory, Beliefs, Reasoning, Actions, Collaboration, Translation and Mobility. Sensory is the lowest layer and Mobility the highest layer [10]. The created agents may or may not contain all seven layers, but all should be designed with the ability to operate the seven layers. He also mentioned that the systems may contain four patterns agents, such patterns are: reactive, deliberative, the opportunistic and interface. Each agent acting according to the needs of the system, but with the behaviors needed to communicate with peer layers of other agents.

The behavior with which the agents are infused is important because this behavior can be benevolent or selfish [11]. The benevolent behavior is concerned to achieve agent common goal, while the selfish behavior only care to achievement of its own goals. This type of behavior is highly dependent on system requirements for which they were created, as there are functions in which selfishness is what will achieve the expected success of the project, where a benevolent behavior would not be appropriate [11]. For example: programing a machine agent with complete defined limits due to exceeding these limits could result in breaking down the machine requires a selfish agent to care for its own integrity. This does not prevent than nay agent could have both behaviors, selfish and benevolent, since a machine agent should communicate their status to other agents, this behavior is a collaborative one.

Chaturvedi [12] identifies that agents can be used to simulate administrative and economic problems. As an example he mentioned stock market where different agents can be defined: buyers, sellers and controlling environment. Where buying and selling agents must achieve a balance so they can achieve adequate margins, meanwhile controlling agent should be aware regarding that the stock market rules are fulfilled.

Jingrong [13] refers to the use of a MAS in order to identify the standards used by the flows of processes regarding an already implemented ERP system, since the use of agents allows that each functional unit are defined as an agent than could interacts with other agents in order to identify the exchange of knowledge and how they should handle coordination of functional units.

Ferber [14] in his work regarding analysis and design of organizations through the use of MAS, claims that its model can be used to describe any type of organization using only three elements: agents, roles and groups. Since agents can belong to any group and include several roles, which occur naturally in any organization. It is relevant to

the study of ERP systems and the selection process because the selection process takes place within an organization, which could be modeled as a MAS.

Kosoresow [15] claims for a use of agent in computer supported cooperative work as auction, negotiation and haggling agents, using cash as metric to reach agreements with other agents. The agents will contain behavioral rules to follow; the negotiation agent will be the most complex, since it needs to know the opponent intention model.

Shi et al [16] claim that agent could be used it to develop three classes of systems: open, complex and ubiquitous computing. That is because agents are capable of create dynamically changing structures, make modular systems and interaction become an equal partnership between agents. They propose a four layer model for agent-based grid computing. They indicate that MultiAGent Environment (MAGE) is a good tool to support development of agent-based grid computing system.

Peleg [17] claims that main characteristic of distributed systems is that there may be autonomous processors active in the system at any moment. And all these processor should be in communication between each other to reach a reasonable level of cooperation. They can be distributed in different geographic locations. The type and purpose of cooperation depends on the individual of each processor participant and user, as on MAS.

Peleg [17] also claims that issues and concerns regarding distributed computing could be categorized as communication, incomplete knowledge, coping with failures, timing and synchrony; and algorithmic and programming difficulties.

Marlowe [18] claims that distributed systems has the following defining properties: multiple computers (nodes), resource sharing, concurrency and message passing. Also the following characteristics properties: heterogeneity, handle of multiple protocols, openness, fault tolerance, persistence, security, insolation and decentralized control. All this properties allow a distributed system to perform a never-ending stream of diverse operations that should fulfill safety and liveness requirements.

Tanenbaum [19] define a distributed system as autonomous computers that work together to give the appearance of a single coherent system. A distributed system is transparent, scalable, open and failure tolerant. Its architecture is layer-based. The main goals of a distributed system are easily connect users to resources, hide the fact that resources are distributed across a network, allow the connection of different kind of computers at any single moment without stopping if a component crash.

Rotem-Gal-Oz [20] claims that software industry has been writing distributed system for several decades taking eight assumptions that is now known as the 8 fallacies of distributed computing: the network is reliable, latency is zero, bandwidth is infinite the network is secure, topology does not change, there is one administrator, transport cost is zero and the network is homogeneous. So, to create a distributed system designer need to address that network is unreliable, take latency in consideration to

make as few as possible connections between distributed components, try to simulate production environment considering bandwidth availability, security is not only beyond perimeter, topology at production environment is in the wild and out of control, there are many administrators and all of them need to know how to diagnose a problem, transport of data cost a lot of money and resources, can increment latency and reduce bandwidth; and finally, network are not homogeneous, there are a lot of proprietary technologies and protocols, be aware of that.

## 3. MAS proposed architecture for ERP selection process

ERP selection process consists of several stages, among them are identification of needs and requirements, identification of suppliers, analysis of vendor ERP solutions, conform evaluation criteria, evaluation process and making decision [21-46].

The intent of the prototype proposed model is exclusive for evaluation process stage, which is where the selection process and decision making is modeled [1, 2]. This unstructured problem requires establishing evaluation criteria that has to be followed [2, 28, 47].

Based on literature [26, 28, 47, 48], we could define that the categories to be evaluated include: functional, operational, technical and economic aspects. These categories are the easiest to represent in a model and simulation tool, since these categories could be translated to quantitative rules easily.

As important as evaluation categories, the actors who performance the evaluation process are important. Such actors correspond to leadership roles in the implementation project [49]. Some of those roles are: chief executive officer, chief financial officer, chief market officer, chief operational officer, chief information officer and functional department managers.

Finally any ERP selection process needs to identify the ERP Solutions to evaluate. The solutions have individual characteristics, functionalities and features that need to be qualify.

In this case, a multi-agent environment could be a good place to model the evaluation process in ERP selection stage. At Figure 2 it can be observed the proposal MAS architecture.

In this architecture there are five different kinds of agents. Those agents could be composed by two databases, one for knowledge and behaviors and one to archive characteristics.

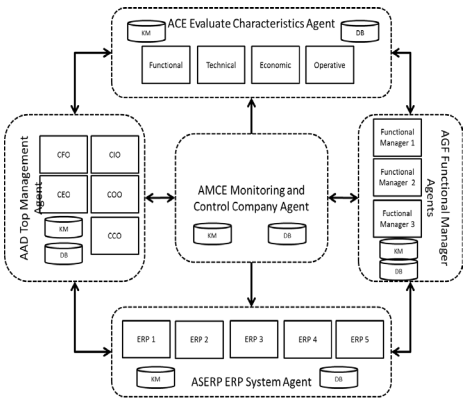



**Figure 2: Proposal MAS architecture to ERP selection process.**

The five agents modeled are: Evaluate Characteristics Agent (ACE), Top Management Agent (AAD), Functional Manager Agent (AGF), ERP System Agent (ASERP) and Monitoring and Control Company Agent (AMCE).

ACE agents will be responsible to contain criteria information that must be met for each ERP. Also, ACE agents databases will contain the characteristics to evaluate and the rules that permit an interaction between those characteristics, with different levels of uncertainty. It also attends requests filed by AAD and AGF agents. ACE agent is a set of agents whose amount could varies according to the amount of aspects need it to evaluate.

AAD agents will contain features and personal profiles of top management behaviors, presented in accordance with data obtained from the ACE and ASERP agents. Also contain decision rules based on profiles of AAD, ACE and ASERP agents. AAD agent is a set of agents whose amount varies according to the number of people at the top management.

AGF agents will contain features and personal profiles of functional management and behaviors, presented in accordance with data obtained from ACE and ASERP agents. Also contain decision rules based on profiles of AGF, ACE and ASERP agents. AGF agent is a set of agents whose amount varies according to the number of people at the functional management, are very similar to AAD agents, but their level of influence in decision making is less.

ASERP agents will contain the information and characteristics of ERP systems, which are linked to the criteria that must be met for each of the characteristics evaluated. ASERP agents attend the request filed by AAD and AGF agents, so that they can apply their own set of rules of decision. ASERP agent is a set of agents whose amount varies according to the number of ERP systems to evaluate.

AMCE agent will be responsible to initiate the evaluation process. AMCE agent will be responsible to register and release ACE, ASERP, AAD and AGF agents. With ACE and ASERP agents, AMCE agent is going to keep communication related to their creation. ACE and ASERP agents will be responsible by themselves to administer their databases. With AAD and AGF agents, AMCE agent will maintain communication in order to inform them regarding the creation of new ACE and ASERP agents, so they can renew their decision-making processes according to new agents involved. AMCE agent will monitor the decision making process performance by AAD and AGF agents. Receive the correspond results from them and perform the final scoring process. AMCE agent will contain the necessary knowledge to issue a suggested ERP system solution based on the information contained at ASERP agents, based on evaluations submitted by the AAD and AGF agents, combined with the levels of influence on decisions of the agents mentioned.

The proposed MAS could be implemented as a distributed computing system as the architecture of MAS meets the definition of a distributed system.

Each of the agents can be placed on different computers which will connect to the central agent that will allow agents to communicate between them in order to

achieve the purpose for which they were created.

The MAGE framework [16] contains all the features of a distributed system, just as the JADE framework. These frameworks allow the creation of a distributed system that enables communication between its components and the transparency necessary to identify the system as a single entity, with the possibility of scaling the system to different environments and equipment. Similarly, there is fault tolerance, because failures of one agent do not affect operation of the rest.

## 4. Conclusions

MAS are composed by one or more agents that behave as individuals. There is a set of control agents, which are responsible to sensor and monitor the environment where agents are, updating individual status and characteristics of agents. Controlling also fault tolerance, openness, transparency and security need it in a distributed computing system. All his activities should be done by the environment agent. It interacts with all the other agents and it is responsible to fulfill the goals for wish the MAS was development and also keep the coherence of a distributed computing system.

There are related literatures that allow the development of solid distributed computing system based in an agent, by example, Buyya [50] present a Network Enabled Parallel Simulator (NEPSi) which can be seen as a multi agent system since each component could be programmed as agents: The arbiter, the daemons and the Very High Speed Hardware Integrated Circuits Hardware Description Language simulator. Each one needs communication and coordination between each other to fulfill their particular goals.

JADE framework is a good platform on which MAS systems could be development. It contains most of the elements found in the infrastructure required for such types of systems, additionally, It meets some of existing standards in terms of agents and MAS. And it also fulfills requirements to develop distributed computing systems using message passing protocols that can be operated on multiprocessor or multicomputer systems.

MAS architecture form a solid distributed computational system that can be used to solve complex problems, since it permit multiple computers systems, resource sharing, concurrency, message passing heterogeneity, the use of multiple protocols, openness, fault tolerance, persistence, security, insolation and a partial decentralized control.

Multi agents systems can be used in the design of distributed decision support systems, where designers have the information needed to characterize the variables, interactions, behaviors, rules, standards, trends and scenarios of particular environments to be simulated.

Next step in this research include develop of a system prototype using JADE, including in its database all the selection variables found in literature, rules, algorithms, evaluation criteria and intelligence behavior. At the same time, perform an empirical study focus on Guadalajara, México organizations

with the purpose of identify additional variables involve in ERP selection process and related stakeholders behavior.

Even after the evaluation and implementation of any kind of system, distributed multi agent systems can be used to process a success assessment of those systems. This success assessment system will require different model architecture that the presented in this paper, since it should handle other variables, relations and criteria rules; but certainly distributed multi agents system can be used in the resolution and simulation of the problems involved in the selection, implementation and post implementation of information systems.

Our proposal to mix MAS, DS and ERP in a decision support system prototype that help organization in the selection process of complex information systems, it is because this kind of decisions have multiple variables involved. They also have a lot of interaction between variables, rules and stakeholders. All this elements require high power computers because the calculus and application of evaluation criteria algorithms could be very computing demanding, since decision support and intelligence systems usually demand high computational power.

The literature review done on the ERP selection process [51]. It did not show the existence of another model for this type of process, using multi agents.

The main focus of MAS systems, found in the literature, are to simulate and improve functional and operational management of business, regardless of the methodology of decision.

The proposed architecture is still under development, which requires proof of concept of architecture. It also needs the characterization of the variables involved in the selection process. As soon as the characterization has been done, a prototype could be implemented.

Future work on this issue could be oriented to the simulation of different decision scenarios and how those scenarios might affect the decision process and the intelligence has to evaluate the scenarios. As a secondary area of research, the graphic representation of the decision process in accordance with stakeholders characterize intelligence and program agents, to verify what kind of environment that the agents are more efficient.

## 5. References

[1] E. Turban and J. Aronson, *Decision Support Systems and Intelligent Systems*: Prentice Hall, 1998.

[2] H. A. Simon, *The New Science of Management Decision*: Prentice Hall PTR, 1977.

[3] L. Zhang and Y. Zhang, "Research on Hierarchical Distributed Coordination Control in Process Industry Based on Multi-agent System," in *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, 2010, pp. 96-100.

[4] M. Wooldridge, *An introduction to multi agents systems*. Sussez, UK: John Wiley & Sons Ltd, 2002.

[5] W. J. King and J. Ohya, "The representation of agents: a study of phenomena in virtual environments," in *Robot and Human Communication, 1995. RO-MAN'95 TOKYO, Proceedings., 4th IEEE International Workshop on*, 1995, pp. 199-204.

[6] J. Ferber, O. Gutknecht, C. M. Jonker, J. P. Muller, and J. Treur, "Organization models and behavioural requirements specification for multi-agent systems," in *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on*, 2000, pp. 387-388.

[7] D. Kehagias, A. L. Symeonidis, K. C. Chatzidimitriou, and P. A. Mitkas, "Information agents cooperating with heterogenous data sources for customer-order management," presented at the Proceedings of the 2004 ACM symposium on Applied computing, Nicosia, Cyprus, 2004.

[8] K. C. Laudon and J. P. Laudon, *Management Information Systems: New Approaches to Organization & Technology*. New Jersey, EUA: Prentice Hall, 1998.

[9] R.A. Flores-Mendez, "Towards a standardization of multi-agent system framework," *Crossroads,* vol. 5, pp. 18-24, 1999.

[10] E. A. Kendall, P. V. M. Krishna, C. V. Pathak, and C. B. Suresh, "Patterns of intelligent and mobile agents," presented at the Proceedings of the second international conference on Autonomous agents, Minneapolis, Minnesota, United States, 1998.

[11] V. R. Lesser, *Multi-agent systems*: John Wiley and Sons Ltd.

[12] A. R. Chaturvedi and S. R. Mehta, "Simulations in economics and management," *Commun. ACM,* vol. 42, pp. 60-61, 1999.

[13] Y. Jingrong, "Research on Reengineering of ERP System Based on Data Mining and MAS," 2008, pp. 180-184.

[14] J. Ferber, "A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems," 1998, pp. 128-128.

[15] A. P. Kosoresow and G. E. Kaiser, "Using agents to enable collaborative work," *Internet Computing, IEEE,* vol. 2, pp. 85-87, 1998.

[16] Z. Shi, H. Huang, J. Luo, F. Lin, and H. Zhang, "Agent-based grid computing," *Applied Mathematical Modelling,* vol. 30, pp. 629-640, 2006.

[17] D. Peleg, *Distributed Computing: A locality-Sensitive Approach*. Philadelphia, USA: Society for Industrial and Applied Mathematics, 2000.

[18] J. Marlowe, D. Lea, and M. Atkinson, *Distributed systems*: John Wiley and Sons Ltd.

[19] A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and paradigms*, First ed. Upper Saddler River, New Jersey: Prentice-Hall, Inc, 2002.

[20] A. Rotem-Gal-Oz, "Fallacies of Distributed Computing Explained," ed, 2004.

[21] (2008, Best Practices for Software Selection. *TEC Technology Evaluation Centers*. Available: http://www.technologyevaluation.com

[22] B. S. Ahn and S. H. Choi, "ERP system selection using a simulation-based AHP approach: a case of Korean homeshopping company," *Journal of the Operational Research Society,* vol. 59, pp. 322-330, Mar 2008.

[23] C. A. Asad, M. I. Ullah, and M. J. U. Rehman, "An approach for software reliability model selection," in *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, 2004, pp. 534-539 vol.1.

[24] Z. Ayag and R. G. Özdemir, "An intelligent approach to ERP software selection through fuzzy ANP," vol. 45, 2007.

[25] P. Blackwell, E. M. Shehab, and J. M. Kay, "An effective decision-support framework for implementing enterprise information systems within SMEs," *International Journal of Production Research,* vol. 44, pp. 3533-3552, 2006.

[26] P. Botella, X. Burgues, J. P. Carvallo, X. Franch, J. A. Pastor, and C. Quer, "Towards a quality model for the selection of ERP systems," in *Component-Based Software Quality: Methods and Techniques*. vol. 2693, ed, 2003, pp. 225-245.

[27] S. Bueno and J. L. Salmeron, "Fuzzy modeling Enterprise Resource Planning tool selection," *Computer Standards & Interfaces,* vol. 30, pp. 137-147, Mar 2008.

[28] X. Burqués, X. Franch, and J. A. Pastor, "Formalizing ERP Selection Criteria," Barcelona2000.

[29] J. P. Carvallo, X. Franch, and C. Quer, "Determining Criteria for Selecting Software Components: Lessons Learned," *Software, IEEE,* vol. 24, pp. 84-94, 2007.

[30] U. Cebeci, "Fuzzy AHP-based decision support system for selecting ERP systems in textile industry by using balanced scorecard," *Expert Systems with Applications,* vol. 36, pp. 8900-8909, Jul 2009.

[31] D. Das Neves, D. Fenn, and P. Sulcas, "Selection of enterprise resource planning (ERP) systems," *South African Journal of Business Management,* vol. 35, p. 45, 2004.

[32] L. Fan and C. Jinliang, "Influencing factors on ERP system selection," in *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on*, 2010, pp. 671-673.

[33] H. Haghighi and O. Mafi, "Towards a Systematic, Cost-Effective Approach for ERP Selection," *Proceedings of World Academy of Science: Engineering & Technology,* vol. 61, pp. 231-237, 2010.

[34] N. Hollander, *Guide to Software Package Evaluation and Selection*: American Management Association International, 2000.

[35] A. Jadhav and R. Sonar, "A Hybrid System for Selection of the Software Packages," in *Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on*, 2008, pp. 337-342.

[36] A. Jadhav and R. Sonar, "Analytic Hierarchy Process (AHP), Weighted Scoring Method (WSM), and Hybrid Knowledge Based System (HKBS) for Software Selection: A Comparative Study," in *Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on*, 2009, pp. 991-997.

[37] N. Karaarslan and E. Gundogar, "An application for modular capability-based ERP software selection using AHP method," *The International Journal of Advanced Manufacturing Technology,* vol. 42, pp. 1025-1033, 2009.

[38] E. E. Karsak and C. O. Ozogul, "An integrated decision making approach for ERP system selection," *Expert Systems with Applications,* vol. 36, pp. 660-667, Jan 2009.

[39] D. Kuiper, "Software selection: The key to a custom fit.," *Evolving enterprise, information technologies for manufacturing competitiveness,* vol. 1, 1998.

[40] P. Nikolaos, G. Sotiris, D. Harris, and V. Nikolaos, "An application of multicriteria analysis for ERP software selection in a Greek industrial company," *Operational Research,* vol. 5, pp. 435-458, 2005.

[41] J. Razmi and M. S. Sangari, "A hybrid multi-criteria decision making model for ERP system selection," in *Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on*, 2008, pp. 489-495.

[42] J. J. Shuai and C. Y. Kao, "Building an effective ERP selection system for the technology industry," in *Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on*, 2008, pp. 989-993.

[43] C. Stefanou, "The Selection Process of Enterprise Resource Planning (ERP) Systems," 2000.

[44] C. C. Wei, C. F. Chien, and M. J. J. Wang, "An AHP-based approach to ERP system selection," *International Journal of Production Economics,* vol. 96, pp. 47-62, Apr 2005.

[45] S. Ya-Yueh, "A Study of ERP Systems Selection via Fuzzy AHP Method," in *Information Engineering and Electronic Commerce (IEEC), 2010 2nd International Symposium on*, 2010, pp. 1-4.

[46] H. R. Yazgan, S. Boran, and K. Goztepe, "An ERP software selection process with using artificial neural network based on analytic network process approach," *Expert Systems with Applications,* vol. 36, pp. 9214-9222, Jul 2009.

[47] F. Chiesa, "Metodología para selección de sistemas ERP," *Reportes técnicos en ingenieria de software,* vol. 6, p. 17, 2004.

[48] J. Pastor and C. Estay, "Selección de ERP en Pequeñas y Medianas Empresas con un Proyecto de Investigación – Acción," Barcelona2000.

[49] N. Garcia-Sanchez, L.-S. Pedro, and M.-H.-S. J. Jose, "Definición y características de roles de liderazgo en la implementación de sistemas empresariales (ERP): Caso de estudio de una universidad," *Revista Latinoamericana y del Caribe de la Asociación de Sistemas de Información,* vol. 2, pp. 43-60, 2009.

[50] R. Buyya, *High Performance Cluster Computing: Programming and Applications, Vol. 2. Edited Book*, Chapter 19 ed.: School of Computing Science and Software Engineering Monash University, 2005.

[51] A. A. Pacheco-Comer and J. C. González-Castolo, "A review on Enterprise Resource Planning System Selection Process," *Research in Computing Science,* vol. 52, pp. 204-213, 2011.

# High Performance Computing Architecture for a Massive Multiplayer Online Serious Game

César García-García, Victor Larios-Rosillo, Hervé Luga
*Universidad de Guadalajara, Université Toulouse-1*
*cesar.garcia@cucea.udg.mx, vmlarios@cucea.udg.mx, herve.luga@univ-tlse1.fr*

## Abstract

*This work presents the current development of a serious game that requires massive amounts of data storage and processing. We will have to create massive simulation with tens of thousands of virtual characters acting in a congruent manner, which presents several challenges to standard computing platforms. Our project consists on a massively distributed training game and is part of the much bigger DVRMedia2 Framework for serious game development. The current focus of the project is to develop a distributed virtual environment where massive events can be simulated with the real-time interaction of multiple users across different geographical areas.*

***Keywords:*** *High Performance Computing, Behavior Modeling, Crowd Simulation, Serious Games, Artificial Life*

## 1. Introduction

The most appropriate definition of serious games in the context of this paper is that of an interactive contest played with a computer using specific rules and with a purpose beyond that of entertainment as its main function, but that uses entertainment as a means to advance domain-specific objectives in education, awareness, training, marketing, etc., based on what Michael Zyda [15] called *collateral learning*.

Section 2 describes the problem as a series of activities that must be executed as part of the functionality of the game. Section 3 illustrates our proposed solution architecture to solve the problem using high-performance computin. Section 4 reinforces the proposed solution by enumerating complexity concerns related to massive multi-user games. Section 5 reflects the actual status of the project, amongst some preliminary results obtained from applying High Performance Computing. Section 6 presents the next steps to be taken towards completion of the project, as well as expected results.

## 2. Problem Description

Real-world events with massive amounts of people present several challenges for modeling and simulation research.

The dynamics of evacuation behaviour can be understood as a functional system, where it is more important to understand the relationships between the parts than to describe the components in detail. A formal model is proposed to effectively study this system.

Our project aims to create artificial entities for distributed virtual environments where massive events can be simulated. When developing any kind of game, several factors must be taken into account to create an appealing experience that effectively immerses the user and further promotes the purpose of the game.

To maintain interactivity it is necessary to provide the user with input processing and display output fast enough that the illusion of real time is created for him. User input must be processed as fast as possible and then transmitted to the virtual environment for its correct application to the world state.

The virtual environment must be recalculated very fast, taking into account the actions of all players in a coherent manner, and updating the world state as well as all of the other users' view of the game.

But the users are not alone in the game: artificial actors must take in account the actions of users as well as other actor, and behave in such a manner that they realistically interact with their world. The generated behaviors must either support or oppose the player's objectives and actions by creating plans and taking actions of their own that affect the world as intended.

Once all the calculations are performed, changes must be delivered to concerned users in such a manner that the flooding of data does not overwhelm the available network resources. Finally, graphics for the virtual environment must be rendered for each user according to the changes in the world state.

However, most of those actions need not necessarily be performed in the user's computer. Some of those could be distributed to a high-performance computing cluster or even into a cloud, but taking into account that moving some of the tasks could severely impact the network architecture.

## 3. Proposed Architecture

Our approach in this case is to distribute some tasks of the problem presented earlier into a high-performance computing cluster. Such tasks include: offline generation of 3D models for the crowds [14], real-time generation of crowd behaviors [3, 4], real-time optimization and balancing of network communications [9] and implementing a data-distribution scheme to support all other activities [13].

The proposed architecture (shown in Figure 1) contains the whole development cycle for the game. Explained from right to left:

A 3D Warehouse process where artists create 3D humanoids using Poser [10] and MakeHuman [8] and 3D buildings using Google SketchUp [6]. Textures for

both are created using GIMP [5].

The 3D buildings and their textures are then stored into a repository in our Titan server that acts as a media server. The 3D humanoids, however, are not directly stored but rather they are used as seeds for crowd generation using a genetic algorithm [14] that produces additional 3D humanoids, meshes and textures similar to the seed that can then be stored into the server.

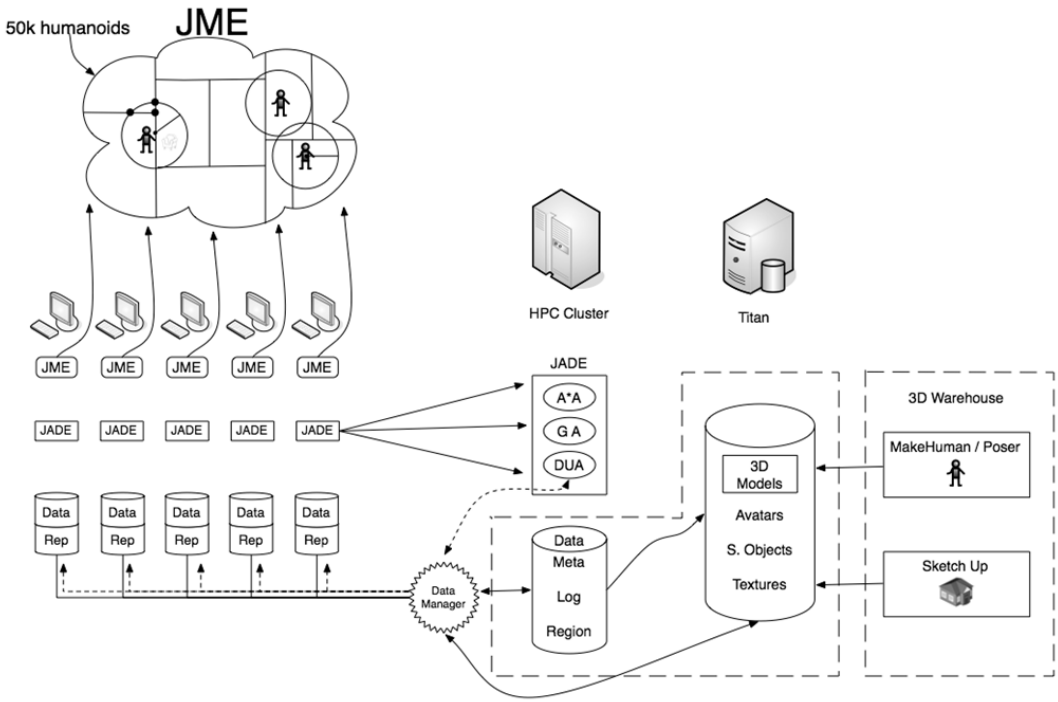



**Figure 1: Proposed HPC System Architecture**

Using the generated 3D models, a rich virtual environment (Figure 2) is created first as coordinates in a database and later rendered by the clients using JMonkey Engine [11].

During gameplay, only input processing, graphical rendering and output display are handled by the client, however, while JME is a good choice for our graphical needs, an additional communications layer, based in the JADE platform [2] is also required. This layer allows for the distribution of the goal management and planning aspects of the behavior generation.

The bulk of this behavior generation will be sent to the HPC Cluster, where a

dedicated JADE container will manage specific agents for navigation (A*Agent), goals and planning (GAgent) and data update (DUAgent). It also logs, applies and distributes all changes to the virtual world.

Each client must also handle a small repository, where 3D models that have already been downloaded from Titan can be stored for some time. The exact length of time depends on the size and distribution level of the region [13].


**Figure 2: A fire scenario created in JME**

The expected result is that, even if the complete virtual environment will not be rendered in any single machine, every client will render a partial view of the environment with coherence and interactivity preserved across all clients.

## 4. Complexity Issues

According to Ulicny [12], the main technical challenge in simulating crowds is that the increased demand of computational resources, be it processor or memory utilization, which grow in different patterns, ranging from the lineal to the exponential, with the number of simulated agents.

The framework provided by JADE allows several aspects of crowd simulation –like agent-agent, agent-environment communication, mobility, etc.– to be transparent to the application, allowing the developer to apply distributed programming techniques to increase the number of agents available for the simulations.

It is also possible to increase the number of simulated agents by managing the complexity of the 3D representation of each agent, reducing the complexity when the agent is far from view and increasing it when the agent is close to the simulation point-of-view. This technique, along with the use of 2D impostors, is known as Level of Detail [1]. At this time we are not contemplating integration of any kind of impostors, as all the rendering is performed by the client, but this step is a clear candidate for distribution to a HPC once sufficient graphic resources are allocated.

## 5. Current Development

We are currently using the JMonkey Engine [11] to develop a serious game where we can simulate evacuations of massive structures such as stadiums or shopping malls.

Adding the JADE layer, we can have a game manager agent controlling the JME graphical interface while also providing other fundamental agent-related benefits such as communication and mobility. Our tests

in a single cluster node suggest that adding this game manager agent does not severely impact graphical performance, as shown in Figure 3.

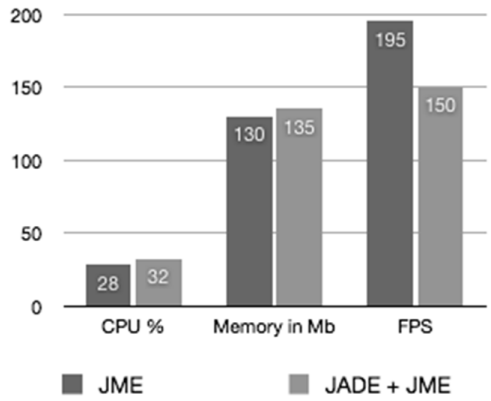



**Figure 3: Comparative performance when running JME only vs. JADE-managed JME**

Implementing the humanoid behaviour as a Finite State Machine seems like a natural decision since the states and the order in which they are executed will never change, this is also very convenient as there is already a Finite State Machine Behaviour pattern implemented in JADE.

As shown in Figure 4, this finite state machine has been divided in conceptual modules which differ in the exact implementation but that essentially perform the same activities in the same order:

- **Process Messages:** Send all previously prepared messages and wait a few milliseconds for messages from other Person agents and process them if present. Messages can contains notifications regarding new threats, smart objects, neighbors and agent intention.
- **Perceive World:** Query the environment agent for changes in the virtual world, if no reply comes in a few milliseconds, assume no change in world state and continue with the current behaviour.
- **Apply Personality Filters:** Apply personality filters to threats and intentions perceived from the environment.
- **Update Internal State:** Update the internal states according to the information being perceived.
- **Update World Model:** Incorporate perceived threats, smart objects and neighbors into the appropriate lists in the knowledge module.
- **Goal Selection:** Update all goal priorities and select the most relevant.
- **Action Planning:** If the current goal is not the same as the previous, discard the current plan and create a new one. If the goal did not change, ignore and continue with current plan.
- **Navigation:** Use the world model to calculate paths required for movement as part of the current plan.
- **Action Sequencing:** Create each action required to fulfill the plan, sequence actions and GOAP actions are different, as sequence actions must be understood by JME: i.e. the GOAP action goto(x, y) will translate in several sequence actions, with only small increments of the coordinates will occur at every step.
- **Action Execution:** Inform the environment agent of the next action to execute in the JME environment. Also prepare any message that needs to be sent.

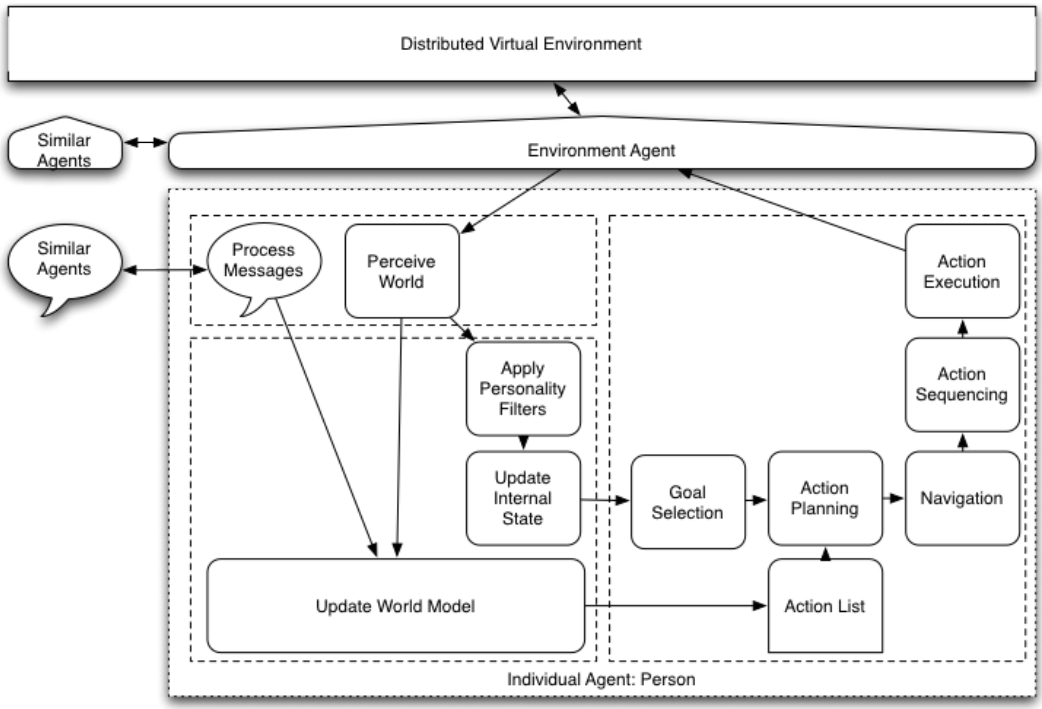



**Figure 4: Multi-Agent System Architecture**

As proof of concept, we generated different crowd sizes using other agents in the same container beyond the game manager or "peer" agent.

Implementing this architecture in the Muli-Agent System allows for easy escalation and replication, as we can create an agent container and environment agent for each HPC computer resource (node, computer, cluster, etc.). Each environment agent communicates with each other to effectively create a nearly unlimited distributed virtual environment for the MMO serious game.

Given that we are working with a serious game, we decided to test the following variables that we consider important for a game: CPU usage, Memory usage and Frames per second or FPS.

Our results show (Figure 5) that even when running 2500 agents, the graphical performance did not suffer from quality loss, and that the real limit was the Virtual Machine's limitation to launch additional threads.

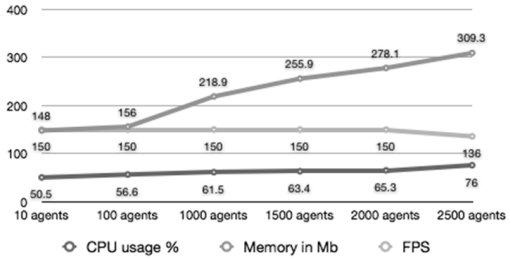



**Figure 5: CPU, Memory and FPS performance during simulation**

## 6. Conclusion and Future Work

We find the current results promising and will continue to develop in this direction, adding modules as part of the agent's behaviors as they are developed and using the obtained values to optimize the maximum number or agents that should be running per peer in a large scale simulation.

The DVRMedia2 framework, to simulate evacuation of massive structures such as stadiums, shopping malls or education centers such as CUCEA.

It is highly desirable to use a detailed environment that is representative and easily recognizable by the general population in this case, the Gimnasio de Usos Multiples in Unidad Revolución shown in Figure 6, where actors can behave in day-to-day situations but also simulate evacuations.

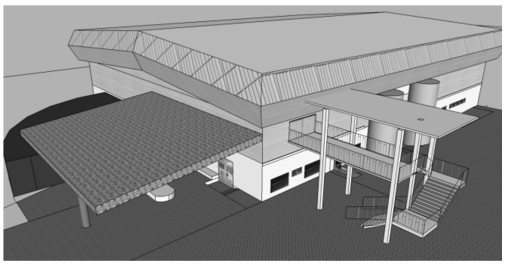


**Figure 6: The Gimnasio de Usos Multiples in Unidad Revolución will be one of the venues for the 2011 Panamerican Games**

As part of the bigger project, the first working demo will be the evacuation of a simulated stadium in the context of the 2011 Guadalajara Panamerican games. This will be a great opportunity to demonstrate the crowd generation, communication, behavior, distributed processing and database capabilities of this architecture.

Future work in this research includes escalating to a several nodes of the High-Performance Computing environment, such as the cluster being used to test network capabilities in other modules of the project [9], where current results suggest capabilities of 20,000 agents per node. Also, we will have to validate the data distribution architecture being used [13].

## 7. Acknowledgements

This project is funded by CONACYT and COECYTJAL grants. The DVRMedia2 framework is a cooperative effort between Universidad de Guadalajara and Institut de Recherche en Informatique de Toulouse (IRIT) trough Le Programme de Coopération Post-Gradués (PCP) Franco-Mexicain. Special

thanks to Intel Guadalajara for providing the required high-performance computing support and to Unidad Estatal de Protección Civil y Bomberos Jalisco for providing required training and access to facilities.

## 8. References

[1] Aubel, A., Boulic, R. and Thalmann, D. Real-time display of virtual humans: levels of details and impostors. IEEE Transactions on Circuits and Systems for Video Technology, 10(2):207–17, 2000.

[2] Bellifemine, F., Caire, G., Poggi, A. and Rimassa, G. Java Agent DEvelopment Framework: A White Paper, 2003.

[3] García, C., Larios, V. and Luga, H. Crowd behavior modeling using high performance computing. In ISUM 2010: 1st International Supercomputing Conference in Mexico, Guadalajara, Mexico, 2010.

[4] García, C. Torres, L., Larios, V. and Luga, H. A GOAP architecture for emergency evacuations in serious games. In GAME-ON'2010: 11th International Conference on Intelligent Games and Simulation, pages 10–12, Leicester, UK, November 2010.

[5] GIMP, http://www.gimp.org/, 2011.

[6] Google SketchUp, http://sketchup.google.com, 2010.

[7] Kuffner, J. Autonomous Agents for Real-time Animation. PhD thesis, Computer Science Dept., Stanford University, Stanford, CA, 1999.

[8] MakeHuman, http://www.makehuman.org/, 2011.

[9] Martinez, M., Larios, V. and Torguet, P. (2009). DVRMedia2 P2P networking strategies to support massive online multiuser virtual environments. Internal report, Universidad de Guadalajara.

[10] Poser, http://poser.smithmicro.com/, 2010.

[11] Powell, Mark. JMonkeyEngine, http://www.jmonkeyengine.com/home/, 2003.

[12] Ulicny, B. and Thalmann, D. Crowd simulation for interactive virtual environments and VR training systems. Europgraphics Workshop, 2001.

[13] Torres, L. and Larios, V. DVRMedia2 P2P Database System to manage coherence in massive multiuser online games and virtual environments. Internal report, Universidad de Guadalajara, 2009.

[14] Zavala, M., Larios, V. and Luga, H. DVRMedia2 Virtual Reality Advanced Editor for Crowded Worlds. Internal report, Universidad de Guadalajara, 2009.

[15] Zyda, M. (2005). From visual simulation to virtual reality to games. Computer, 38(9):25–32.

# Architecture for Virtual Laboratory GRID

Francisco Antonio Polanco Montelongo[1,2], Manuel Aguilar Cornejo[1]
*[1] UAM-Iztapalapa, [2] UPIITA-IPN*
*Department of Electrical Engineering, Department of Engineering and Advanced Technologies*
*México City, México*
*fpolancom@ipn.mx, mac@xanum.uam.mx*

## Abstract

*In the development of distributed and parallel applications, it is very important to verify and validate its functionality. This task is very hard due to the nature of the distributed systems. To validate these systems in real conditions it is necessary to allocate resources in amount and characteristics equivalent to the production environment. This situation is difficult to achieve.*

*In this work, we propose the development of a Virtual Laboratory for GRIDS, which is a system that allows the creation of a testbed using virtualization technologies. The proposed architecture is based on a multilevel scheme. At the lower level we build a virtual cluster through virtual machines and finally, the superior level constitutes a virtual grid. In order to maintain the testing environment as realistic as possible, we use commodity Middleware tools. A virtual component communicates through a virtual network that emulates a WAN network behavior.*

*In addition to the evaluation and validation of distributed applications, this system allows the training on configuration of clusters and Grids.*

***Keywords:*** *Testing Platform, Virtual Machine, Virtual Cluster, Virtual GRID*

## 1. Introduction

In order to evaluate the effectiveness of a distributed application it is necessary to execute it into test scenarios that represent the typical characteristics of the production environment for which it was designed. To reach this aim, literature has proposed the use of experimental methodologies through simulation and emulation. The experiments that use simulation require the development of the models of the application and of the execution environment. These models must abstract details with the purpose of maintaining their complexity within a reasonably manageable level. Therefore, the fidelity of the evaluations is diminished with respect to which it would be possible to be developed by means of a real execution environment and a real application. The obtained results of the simulation of an experiment are abstract and require of interpretation on the part of experts, increasing the complexity of the process of evaluation.

To increase the fidelity on the evaluation, the platforms "In-Situ" [1] is used. This option of experimentation executes a real application over a real environment of execution. This solution has the disadvantage of not maintaining control over the execution environment, since this occurs when it use a real environment, so it has a very low capacity for reproducible results.

There are alternatives such as the experimental methods that use emulation. This option allows the execution of a real implementation or model of it within the model of the execution environment. The model of the execution environment recreates the conditions within a production environment for which such application was created. The emulator keeps under control all the conditions of the execution environment, so it achieves high reproducibility of results. Running a real application without requiring any change in the evaluation platform, enables accurate results, in the sense that the results approximate to those obtained by a real production environment.

The emulators are highly complex systems of software, to represent systems on a large scale, it is necessary to multiplex the physical resources among the emulated ones. Virtualization technologies simplify the implementation of an emulator, as it manages the physical resources and multiplexes among different emulated resources, isolating each other [2]. Resources that can be emulated include the processor, memory, storage and connectivity.

Using an experimental platform for Distributed systems GRID that uses the emulation experimental methodology supported by virtualization technology represents a very promising solution. It represents a solution for the problem of allocating physical resources for the activities of validation and verification of distributed applications. To set a large scale scenario it only requires a fraction of the physical resources. In addition, it allows representing the heterogeneity in hardware, software and networking technology implicit within a distributed GRID.

The Virtual Laboratory GRID (VLG) creates an evaluation scenario generating a virtual infrastructure of a distributed system GRID that emulates a real production environment. The virtual infrastructure emulates the hardware (using Virtual Machines), software (operating systems, middleware, applications), and connectivity components. In order to maintaining a high fidelity reproduction of a production environment, commodity GRID Middleware tools are used. Due to the possibility to represent large scale scenarios, with hundreds of emulated nodes, it is necessary that experimental platform automatically builds the infrastructure and configure the services that provide.

To create the whole virtual infrastructure, the Virtual Laboratory GRID system uses the same way that a real GRID system. The computer resources are aggregated in a multi-level fashion. At the lower level there are the physical computer components, those are connected together usually through a local area network. Cluster Middleware Tools are used to coordinate the computational resources to obtain the behavior of only one resource with a high computational capacity. Then multiple

Clusters systems are interconnected within a wide area network integrating the resources of a distributed system GRID.

Typically, each computer Cluster is inside a different administrative domain. The Clusters aggregated in a GRID system can use resources heterogeneous in hardware, software (Middleware) and connectivity technology. To coordinate the resources the distributed systems GRID use Middleware tools to manage the security, job execution, resources scheduling, monitoring and accounting aspects. When the platform is ready, the distributed application can be executed using the tools provided by the Middleware. The multilevel structure of the resources generated by VLG is shown in Figure 1.

The Virtual Laboratory GRID system architecture considers the following design drivers: flexibility, heterogeneity, scalability and interoperability required by an experimental platform for distributed systems. Additionally, the Virtual Laboratory GRID system also serves as a platform for training of specialized personnel on the Middleware tools used in distributed systems GRID and computational Clusters.

This paper is organized as follow: In section II we present the Virtual Laboratory GRID architecture. We discuss the related work in section III. Finally in section IV, we summarize the result of this work.

## 2. Virtual Laboratory GRID Architecture

The Virtual Laboratory GRID aims to provide a platform for experimentation that recreates a distributed system GRID. These types of systems are composed by physical resources, usually grouped in Clusters, geographically dispersed and into different administrative domains.

The Middleware Cluster tools make that all the resources into the Cluster behave as a single resource with the accumulated capacity of all its components. Typically the resources of a Cluster are inside of a unique administrative domain and use a LAN network to communicate to each other. On the other hand, the GRID system, interconnects the resources through a WAN network, and has to resolve the different policies established on each administrative domain to be able to coordinate the resources to represent just an only resource with a large capacity.

The objective of a Virtual Laboratory GRID is to build an experimental platform that can execute distributed applications with evaluation aims.

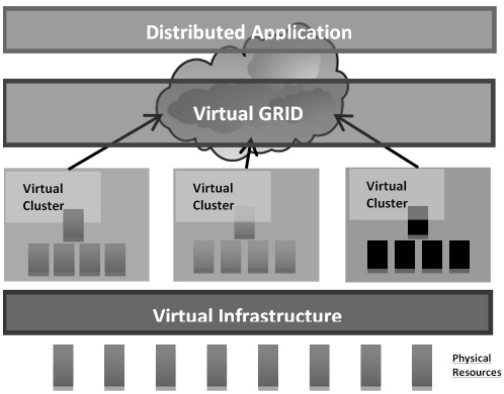



**Figure 1. The multilevel resources structure for the VLG**

Therefore, the architecture proposed for the Virtual Laboratory GRID must satisfy the following aspects: a) Heterogeneity, since it must allow to count on nodes of characteristic manifolds, such as, different operating systems, storage capacity, etc., b) Interoperability, to make possible the integration of multiple tools Middleware Cluster and GRID, c) Flexibility to allow the creation of multiple evaluation scenarios, d) Scalability to express infrastructures of diverse scale, e) Usability, so that the evaluator expresses the configuration of the infrastructure of a simple way.

The architecture of the Virtual Laboratory GRID that satisfies the expressed requirements is shown in figure 2.

The proposed system requires use a physical computer Cluster. Configured like a master node (Front-End) and multiple worker nodes (Back-End), with Cluster Middleware tools that provide an operational Cluster.

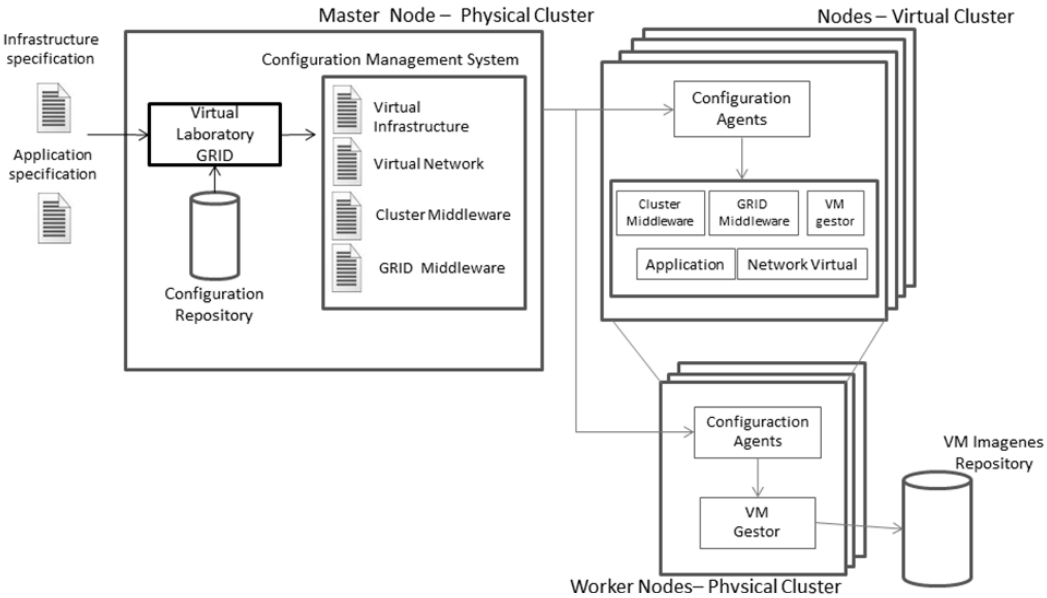



**Figure 2. Virtual Laboratory GRID – Architecture**

The different components that integrate the Virtual Laboratory GRID are described next:

A. Infrastructure and Application specification

The Virtual Laboratory GRID orchestra multiple middleware tools to build the required infrastructure. The proposed architecture uses a configuration-driven

design. The infrastructure specification defines the infrastructure and network model of the experiment. The specifications are translated to an intermediate representation, and then to specifics commands of the corresponding tools. We use automatic configuration administration system tools, like Cfengine[3], Puppet[4], or Bcfg2[5]. These can manage the configuration model representation and translate into the Middleware tools commands. These tools use a client-server schema to coordinate the reinforcement of the configuration model into a multiple-node system. Additionally, it has the capacity to express relations of dependency between applications and services, and defines the procedures to satisfy those dependencies.

Once the experimental platform is operating, the system uses the application specification to deploy de application across the virtual GRID.

### B. Virtual Infrastructure

To support the experimental platform it is necessary to create an infrastructure that satisfies the requirements defined by the tester. To provide such infrastructure we use virtual resources represented by a Virtual Machine (VM) – which represent a complete platform, hardware, software – Operating System, applications, libraries, Middleware tools, etc., and network connectivity. The virtual machines require special software to manage the multiplexing of physical resources, the Virtual Machine Monitor (VMM) or Hypervisior. There are multiple VMM products, the main are: Xen[10], KVM[11] y VMWare[12]. The life cycle of the virtual machines (VM) is administrated using a management system of virtualized resources. OpenNebula[6], OpenStack[7], Eucalyptus[8], XPC[8] which are management systems of VM. These systems allow using different hypervisors. Additionally, they have the capacity to configure each virtual machine with its own name, IP address and a particular configuration.

The VM images are in the VM image repository. The VLG uses predefined VM images, that are configured with the Middleware Cluster and/or GRID required on the virtual Cluster Master Node and the virtual Worker Node of each virtual Cluster into the experiment scenario.

To configure a Virtual Cluster a virtual machine must be designed like Cluster Master Node. Then they are configured with the appropriate Middleware tools on it and the services required to operate the entire virtual Cluster. In parallel, to configure the Worker Node of the virtual Cluster the VLG selects a VM in operation and deploys the Middleware tools on it. Then it takes a snapshot of this VM and it is used to clone the virtual nodes required by the experimental scenario to a particular virtual cluster.

### C. Virtual Network

The nodes that integrate a virtual cluster are connected logically configuring the virtual network interface of each node to a same subnetwork. The VMM on each node maintain a routing table with an entry for every virtual node that integrates the virtual cluster. The VMM use a virtual bridge to connect every VM inside the same physical

node and with the others VM at different physical nodes. To implement the capacity of interconnection the Hypervisors use a NAT (Network Address Translation) service.

The links between the multiple virtual Clusters represent a WAN links. These links have delays, losses and another characteristics specific of a WAN network links. To achieve such behavior we use a network emulator. DummyNet[13] and NISTnet[14] emulated WAN network links.

The network topology for each virtual Cluster is specified at the infrastructure specification.

### D. Virtual Cluster

The virtual machines created into the Virtual Laboratory GRID are grouped into a multiple virtual Clusters. Every virtual cluster can use different Cluster Middleware tools. The experimentation platform according the infrastructure specification deploys the Cluster Middleware into the VMs of each virtual cluster. Verify the dependency relations of each middleware tool and if it is necessary active the necessary procedures to satisfy these dependencies. To verify the correct operation of the virtual Cluster, a test case is executed. The results from the verification are sent to the VLG module.

### E. Virtual GRID

Once the virtual Clusters are in operation, we add them to a virtual GRID system. In the same way that the virtual Clusters, the virtual GRID is configured using GRID Middleware tools. To deploy the Middleware tools it is necessary at first to create users and groups accounts across the multiple virtual nodes to permit the operation of the distributed system. To finalize the deployment procedure; we verify the operation of the GRID system using a test case.

### F. Distributed Application

With the infrastructure in place, we already have a experimentation scenario configured according to the specifications defined by the tester. Then the distributed application can be deployed and executed into the experimentation platform. This procedure is accomplished using the GRID Middleware tools.

### G. Virtual Laboratory GRID

The VLG module function is to coordinate the orchestration of the creation of the experimentation platform, also to represent the interface with the tester to capture the specification of the configuration model.

Likewise, it coordinates the conversion from a high level abstraction infrastructure specification to a series of specific commands of the multiple Middleware tool used. According to the level within the resource structure, the VLG module generates the intermediate representation of the configuration model. There is one configuration repository that contains the necessary configuration specifications for each Middleware tool used by the experimentation platform. The configuration

specifications are modified according to the configuration model defined by the tester.

To keep the control of every resource into the experimentation platform, the VLG module maintains a relation of resources and configuration of these in a configuration table. It permits to the system trace a specific configuration of a resource, and can modify the configuration on-line.

To reinforce the configuration, the VLG module communicates with configuration agents located inside each physical and virtual resource. These agents are the responsible for enforcing the configuration model. To verify the correct operation of the Middleware tools the configuration agents execute a test case and report the result to the VLG module.

Once that VLG module confirms the correct operation of a level of resources, then the tester must continue with the next level.

To monitor the creation and configuration of the resources into the experimentation platform, the tools provided by the Middleware tools are used.

### H. Configuration Agents

The configuration agents reinforce the model of configuration convert the intermediate specification to a series of specific commands depending on each Middleware tools.

Also, verify the satisfaction of the dependency relations for each tool using the procedures settled by the experiment platform.

## 3. Related Work

Due the importance of the distributed systems in the scientific and engineering research, several experimentation platforms have been developed. Gustedt et al [1], define taxonomy of the experimentation methodologies. The experimentations allow to evaluate and to validate a solution by executing a real application (or a model of it) into a model of the execution environment.

There are several experimentation platforms "In-situ", Grid5000[15], Das-3[16] or PlanetLab[17]. This system uses the actual conditions which are presents in an operating environment. Very realistic results were obtained, but they have a very low capacity for reproducibility, necessary to develop assessment activities.

On the other hand emulators have been developed to create platforms for evaluation, such as the case of Emulab [18], which uses technology based on BSD Jails [19] to perform multiplexing of virtual resources into physical resources. However, there is no complete isolation between execution environments. Likewise, no network resources are virtualized in whole, so that the network protocol stack is shared between the physical node and virtual resources. Childs et al [20] present GridBuilder system, which proposes the creation of a Grid emulator hybrid system (combining physical and virtual nodes) for the Irish Grid system. This system has a specific purpose.

Calheiros et al [21] presented a framework for the automated creation of an evaluation platform for Grid applications, considering the creation of a virtual platform

using the Xen hypervisor, using the protocol WBEN as the management protocol to control the experiment. The system intends to use a limited set of virtualization technologies and distributed systems grid.

Wang et al [22], propose a system that produces a grid system by creating virtual resources within a physical Grid system for the creation of Virtual Distributed Environments in order to create the computing environment required by the user. The architecture of the proposed system requires physical and virtual network resources. The proposed solution deals with the effects of the constraints of using resources provided by the physical grid. While our proposal focuses on the provision of virtual grid platform inside a set of resource owners.

## 4. Conclusions

To develop distributed applications GRID assessment is necessary to allocate resources to this activity. Usually, such kind of systems remain in operation of all its resources. Therefore, it is difficult to have the necessary resources to represent a production environment.

The Virtual Laboratory GRID is an experimentation platform that solves the above situation, since it recreates a GRID distributed system with the characteristics and dimensions typically found in the operation of a real system. With the ability to generate the test scenario required by the tester in an automated manner.

To achieve high fidelity to real production environments, we consider the use of commodity Middleware tools used in creation of both cluster and the Grid systems. The VLG build a testing platform that allows the evaluation of distributed applications requiring only a fraction of physical resources. Also the infrastructure created can be used as a training platform for personnel in Middleware tools which both Cluster and GRID systems.

## 5. References

[1] Jens Gustedt, Emmanuel Jeannot, and Martin Quinson. Experimental methodologies for large-scale systems: a survey. Parallel Processing Letters, 19(3):399–418, Sep. 2009.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In Procs. of the 19th ACM Symposium on Operating Systems Principles, SOSP'03, pages 164–177. ACM Press, 2003.

[3] Cfengine. http://www.cfengine.org, 2011.

[4] Puppet. http://www.puppetlabs.com, 2011.

[5] Bcfg2. http://trac.mcs.anl.gov/projects/bcfg2 , 2011.

[6] OpenNEbula Project. http://www.opennebula.org, 2011.

[7] OpenStack. http://www.openstack.org/, 2011.

[8] Eucalyptus. http://open.eucalyptus.com, 2011.

[9] XCP. http://www.xen.org/products/cloudxen.html, 2011.

[10] Xen. http://www.xen.org/, 2011.

[11] KVM. http://www.linux-kvm.org/, 2011.

[12] VMWare. http://www.vmware.com/, 2011.

[13] DummyNet. http://info.iet.unipi.it/~luigi/dummynet 2011.

[14] NIST Net. http://snad.ncsl.nist.gov/nistnet/, 2011.

[15] R. Bolze, F. Cappello, E. Caron, M. Daydé, F. Desprez, E. Jeannot, Y. Jégou, S. Lanteri, J. Leduc,N. Melab, G. Mornet, R. Namyst, P. Primet, B. Quetier, O. Richard, E.-G. Talbi, and I. Touche. Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed. International Journal of High Performance Computing Applications, 20(4):481–494, November 2006.

[16] The DAS-3 project: http://www.starplane.org/das3/, 2011.

[17] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson,M. Wawrzoniak, and M. Bowman. PlanetLab: an overlay testbed for broad-coverage services. SIGCOMM Comput. Commun. Rev., 33(3):3–12, 2003.

[18] Hibler M, Ricci R, Stoller L, Duerig J, Guruprasad S, Stack T, Webb K, Lepreau J. Feedback-directed virtualization techniques for scalable network experimentation. Technical Note FTN-2004-02, University of Utah Flux Group, 2004.

[19] Kamp PH, Watson RNM. Jails: Confining the omnipotent root. Second International System Administration and Networking Conference. SANE: Maastricht, Netherlands, 2000.

[20] Childs S, Coghlan B, O'Callaghan D, Quigley G, Walsh J. A single-computer Grid gateway using virtual machines. In: Proceedings of the 19th international conference on advanced information networking and applications. Washington, DC (USA): IEEE Computer Society; 2005. p. 310–5.

[21] Rodrigo N. Calheiros, Rajkumar Buyya, and César A. F. De Rose. 2010. Building an automated and self-configurable emulation testbed for grid applications. Softw. Pract. Exper. 40, 5 (April 2010), 405-429.

[22] Lizhe Wang, Gregor von Laszewski, Marcel Kunze, Jie Tao, Jai Dayal, Provide Virtual Distributed Environments for Grid computing on demand, Advances in Engineering Software, Volume 41, Issue 2, February 2010, Pages 213-219, ISSN 0965-9978.

# Autonomous Decentralized Service Oriented Architecture for Mission Critical Systems

Luis Carlos Coronado-García, Pedro Josué Hernández-Torres and Carlos Pérez-Leguízamo
*Banco de México*
*email:{coronado, pjhernan, cperez}@banxico.org.mx*

## Abstract

*Service Oriented Architecture (SOA) represents a new model in the traditional way of designing systems, this is due to frequent change of requirements and high integration of the organizations' applications around the world. Although the use of SOA has increased, certain applications' features do not allow its use. This is the case of mission critical applications, whose characteristics are high availability, continuous operation, high flexibility, high performance, etc. On the other hand, these features have been covered in Autonomous Decentralized Systems (ADS). This article presents a modeling novel approach SOA-ADS called Autonomous Decentralized Service Oriented Architecture (ADSOA). Additionally it is presented the Loosely Coupling Synchronization and Transactional Delivery Technology that provides consistency and high availability. To show the viability of the proposal, a prototype is presented.*

***Keywords:*** *Decentralized Autonomous Systems, High Reliability, Fault Tolerance, Service Oriented Architecture, High Availability, Data Consistency.*

## 1. Introduction

Recently, the Service Oriented Architecture (SOA) has positioned as a key for integration and interoperability amongst different applications and systems in an organization.

In [1] SOA is defined as business architecture at logical level, in which application's functionality is available to users as shared reusable services on an IT network. SOA services are the functionality of some applications that are exposed through interfaces and are invoked by messages.

SOA represents a new model in the traditional way of systems design because it is built around the individual processes of the different areas of an organization. In SOA, applications are split into small units that offer services that can be shared among different applications in an organization. Therefore, SOA facilitates integration between different applications in an organization, allowing easy adaptation to the continuous changes and new needs.

In recent years, the critical application integration need has been increased with SOA, that is the case of Financial Applications. These applications require high availability, timeliness and

information consistency. According to Gartner [2], SOA was used between 50 % and 80% of the new critical applications developed in the last year. Although it has been accepted, the conventional SOA presents problems of availability, performance and flexibility [1]. Therefore, it is necessary to explore new models and service oriented architectures that meet the needs presented in this kind of implementations.

On the other hand, the Autonomous Decentralized Systems [3], [4] was proposed to provide high availability, on-line properties and fault tolerance. It has been used to successfully implement mission critical applications in industrial, financial and transport organizations [5], [6]. These properties suggest to ADS as the best option to meet the above requirements to integrate mission critical applications with a SOA. However, we must consider that the implementation of ADS generally requires specialized hardware and software, so the challenge is how to model and implement a SOA based on ADS, without specialized hardware.

This article is organized as follows: In Section II we give a view of the ADS concept and architecture, in Section III we describe SOA and indicate the mission critical application requirements, in the Section IV we present the architecture and terminology proposed and finally, in the Section V are the conclusions.

## 2. Autonomous Decentralized System (ADS)

### 2.1. ADS Concept

The processes in the nature are an unlimited source of inspiration. For example, the cells from human organisms are originally constituted by the same DNA code, but resulting in heterogeneous organs (kidney, heart, lung, etc.) with specialized functionalities. The basic concepts of ADS can be found on [3], [4].

In addition, each of these organisms also is constituted by a group of heterogeneous cells that altogether carry out a specific task. For example, a kidney has a subsystem that regulates the production of urine. In the analysis of this subsystem, we find that it is also formed by a group of heterogeneous organized cells with a certain level of expertise. On the other hand, following with the analysis in this last level, we would observe that many of these cells are identical in all parts and are able to exist independently, but they continue granting heterogeneous functionalities to the entity. All of this is the base on that ADS is composed.

In ADS, we design an application with the following attributes:

- The application is composed of subsystems and entities.
- A failure of any entity is a normal situation.
- The application continuously changes among operation, maintenance and expansion states.

These attributes allow the development of applications with the following properties:

***On-line Expansion:*** An application evolves constantly. New subsystem integration must be on-line. These actions assure high availability to the application. A service interruption of an application is not acceptable; the gradual improvement/ growth of the resources is preferable.

***Fault Tolerance:*** Neither hardware nor software is 100% free failure, for that reason it is required to guarantee the continuous availability of the service. Therefore, if a part of the application is prone to failure, the application must continue its operation.

***On-line Maintenance:*** Maintenance and testing must be done without halting the services, especially on real time applications.

To satisfy fault tolerance and on-line properties, it is necessary that the subsystems have the following basic features:

***Autonomous controllability:*** Each subsystem should be self-administrated and continue operating although the failure of other subsystems.

***Autonomous coordinability:*** If any subsystem is incorporated, modified or failed, other subsystems must be able to coordinate to each others for completing their individual goals.

Additionally, to reach autonomous Controllability and Coordinability each subsystem must satisfy the three following principles:

***Equality:*** All the subsystems are equal, it does not exist a master-slave relation among them.

***Locality:*** Each subsystem is self-managed and coordinates with others on the basis of the local information.

***Self-Containment:*** Each subsystem contains the functionality for self-management and operation.

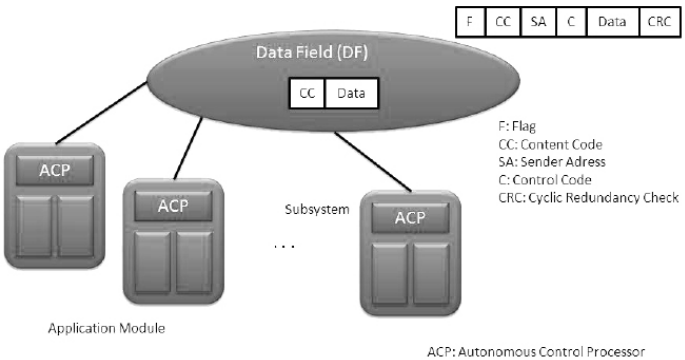



**Figure 1. ADS Architecture**

### 2.2. ADS Architecture

ADS architecture, showed in Figure 1, is composed by the following components:

***Entity***: It is an independent element that is part of an application. Each entity contains an Autonomous Control Processor (ACP) and their specific functionality. Each entity registers the Content Codes that can be processed in the ACP. The ACP filters the data received based on the Content Code that is in the message header.

***Data Field (DF)***: It is the media between application entities that allows exchange messages. These messages are sent and circulated through this DF and all connected entities can read them.

***Content Code (CC)***: It is the base of the communication protocol between entities. The CC identifies the message with its content. When an entity sends a message to another one, it is not necessary to set the id of the receiver, just set the CC. The entities connected to the DF will accept or reject the incoming message according to the CCs registered in their ACP. The communication protocol is based on the message data instead of the receiver address. In ADS, is not necessary to know the destination address. The communication by CC allows the replication of entities which increases reliability. An entity may receive duplicate messages and discard the previously received. In addition, a failure of an entity does not affect the others.

## 3. Service Oriented Architecture (SOA)

The organizations around the world require a high IT resources integration. CORBA and COM+ are some efforts to integrate heterogeneous technologies, but they have not been popular because they require further development [1]. Another related effort is SOA, which unlike the other proposals do not use a particular technology and is based on open standards. SOA is an architecture that changes the application development concept, its main principle is the service orientation [1], [7]. In this service architecture there are three roles as shown in Figure 2:

1. Service Provider: who offers the service.
2. Service Consumer: who invokes the service.
3. Services Registry: who offers and maintains the service registry.

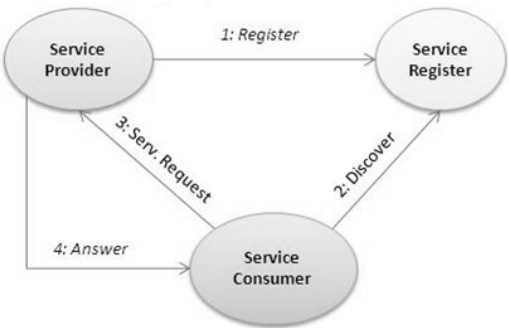



**Figure 2. SOA Architecture**

A SOA service life cycle could be described as follows: 1) A Service Provider registers its services in the Service Registry; 2) A Service Consumer discovers the services that interest it in the Service Registry; 3) The Service Consumer sends a request to the Service Provider that offers the service; and 4) The Service Provider receives and processes the request and sends the result to the Service Consumer.

To carry out this task, it requires the following: 1) A communication media that allows message flow; 2) A communication protocol; 3) A service description that indicates the rules; 4) A set of available services; 5) The business processes (sequence of rules that satisfy a business requirement); and 6) A service registry.

Additionally, quality of service is considered by:

1. Policies: conditions or rules for the consumers.
2. Security: rules for the identification, authorization and access control to the consumers.
3. Transactions: consistent results.
4. Management: services maintenance.

To develop solutions based on this architecture it is used standard protocols and conventional interfaces that allow access the business logic. The result is a simple application of IT, integrated and flexible that could be aligned with business objectives. SOA services are characterized by:

1. be reusable,
2. to provide a formal contract,
3. loosely coupled,
4. to allow composition,
5. autonomous,
6. stateless,
7. discoverable and
8. open.

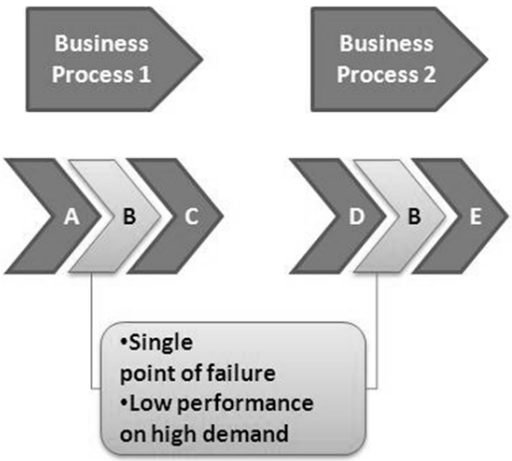



**Figure 3. Conventional SOA**

Reusing services across different business processes is a basic feature of SOA. A service shared by different processes may have the following problems, shown in Figure 3:

- Low performance and low availability when demand increases.
- The processes stop when a service fails.
- Low flexibility because there is no on-line maintenance.

SOA conventional technologies provide mechanisms for disaster recovery that provide acceptable response time for certain types of systems. Moreover, when a service has a high demand the performance is low, to solve this issue, it is required to increase the computing resources and it is necessary stopping the system operation. In this sense, this type of solutions is not acceptable for a mission critical system that requires high availability, continuous operation, high flexibility, high performance, etc. In order to meet such the requirements, in this paper is proposed a combination of SOA concepts and ADS features.

## 4. Autonomous Decentralized Service Oriented Architecture (ADSOA)

### 4.1. ADSOA Concept

In Figure 4 is depicted the conceptual model of the elements that result from the combination of the characteristics of SOA with ADS. ADSOA is composed of autonomous entities that offers or requests services via messages. Each entity is uniquely identified by a reference composed by three values called business id, subsystem id and entity id. For each entity may be several instances fully independent. Each instance has the same functionality that its entity represents.

A subsystem can be formed by a group of entities and in the same sense a business may be formed by a group of subsystems.

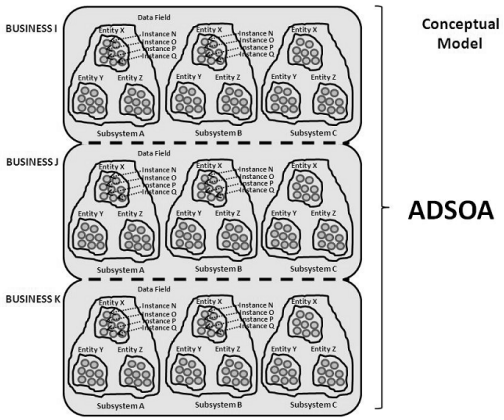



**Figure 4. ADSOA Conceptual Model**

The media in which the messages flow is called Data Field (DF). Many businesses can coexist in a DF. Each entity must discern which messages received or not. The multiple instances of the same entity should focus on carrying out the same task, regardless of the sequence that their messages have been received.

### 4.2. Proposed Architecture

The ADSOA is composed of entities connected to a DF. The entities can request and/or provide services. Each entity may have one or more instances simultaneously connected to the DF. The difference between instances and entities is that an entity has a unique identifier that can be distinguished from the others. However, instances of an entity share the same identifier. The logical architecture is shown in Figure 5.

The DF is composed of several interconnected processes between them. The entities are connected to these processes in order to exchange messages as it is shown in Figure 6.

The authentication protocol is as follows:

1.- An entity instance asks for permission to connect to the DF.

2.-The DF sends a challenge to the instance.

3.- The instance answers the challenge.

4.-If the challenge is correctly answered, the connection is accepted, otherwise is rejected.

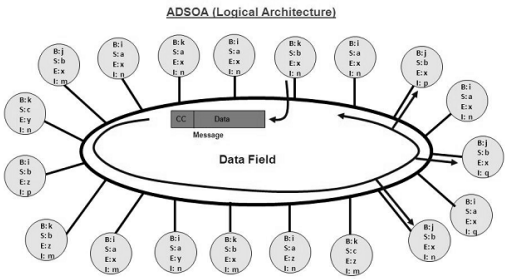



**Figure 5. ADSOA Logical Architecture**

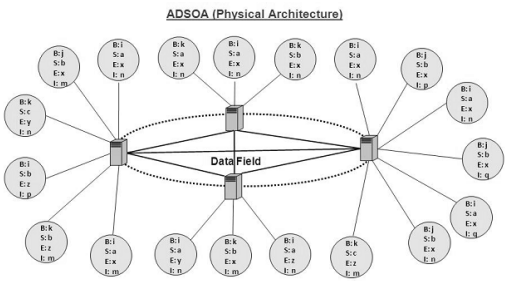



**Figure 6. ADSOA Physical Architecture**

When the instance of an entity is connected to the DF, it could request or provide services by sending messages accord to the Content Code specified.

In this architecture, may be multiple autonomous entity instances, they may receive several service requests. An instance of a particular entity can receive messages in a different order in compare to another one, or even fewer messages than other instances.

To solve this problem, we propose the Loosely Coupling Synchronization and Transactional Delivery Technology described in subsection C.

### 4.3. Loosely Coupling Synchronization and Transactional Delivery Technology

In this technology we define the concept of transaction in the scenario in which an entity requests a service to another and requires know if it has been received. The requesting entity must maintain this request in pending processing state until it receives an acknowledgement from receiving entity. Also we define sequential in the sense that the entity requester must receive a minimum number of acknowledgments from receiving entities to send the next request for service, for example, a Y request should not be sent until receives the minimum number of acknowledgments of the X request.

The service request information structure should include the following elements: Content Code, Foliated Structure and Request Information.

The Content Code specifies the content and defines the requested service.

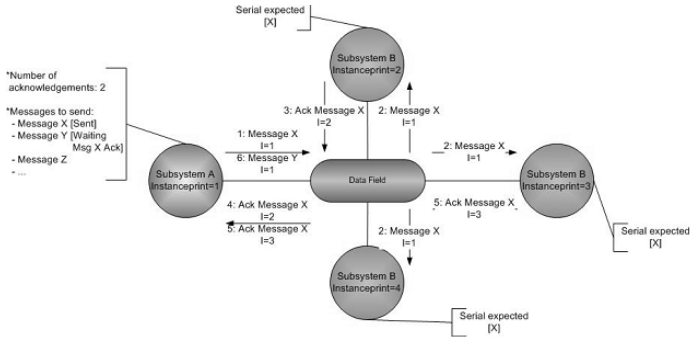



**Figure 7. Sequentiality and Transactionality**

The foliated structure identifies the transaction. This structure is based on:

1. requester id,
2. specialized task id for that request (Pivot),
3. a sequence number,
4. an generated id based on the original request information (event number) and
5. a dynamic id for the instance of the entity (instance print).

With these elements we can guarantee the identification of acknowledgments received by the entity. We can also ensure the sequence of multiple requests, as shown in Figure 7.

If an instance receives a service request with a sequence number greater than expected, then by the principle of sequentiality, knows that another instance of its entity will have the missing messages. In this case, the receiver instance asks to his entity the missed messages, that is, the other instances of the same entity. This idea is represented in Figure 8.

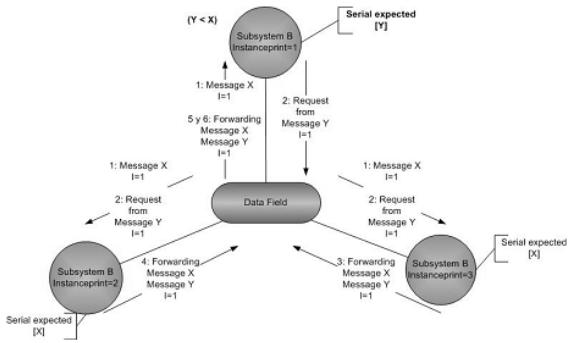



**Figure 8. Synchronization with other Instances**

On the other hand, if an entity receives several times the same service request, this can be distinguished by the instance-print if this request belongs to the same requester instance or from a different instance of the same entity. According to this, the receiver entity can determine whether requests received are in accordance with the minimum number of requests that the requester entity are required to send, as shown in Figure 9

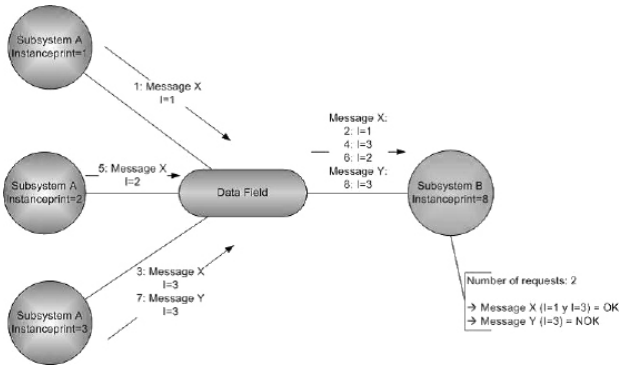



**Figure 9. Receiving Multiple Requests from an Entity**

## 4.3. ADSOA Feasibility Probe

The system used to test the feasibility of the technology and architecture proposed is a system that displays information very sensitive and very important through digital white board. For these features, this system does not allow business disruption and must operate 365 days / 24 hours. Therefore, this system is considered as a mission critical application. It was decided to distribute its functionality redundantly in autonomous elements. The functionality of the system was modeled as follows:

1.-Information generation subsystem.

• 2 Generator type entities. Element that generates information of numbers and letters. Each of these entities has only one instance.

2.-Information display subsystem.

• 2 Display type entities. Element that displays information generated from a generator entity. Each of these entities must have at least 2 instances running to make it fault tolerant.

3.-Monitoring subsystem.

• 2 Monitor type entities. Element that monitors if the information of a generator entity is properly displayed in de display entities. Each of these entities has only one instance.

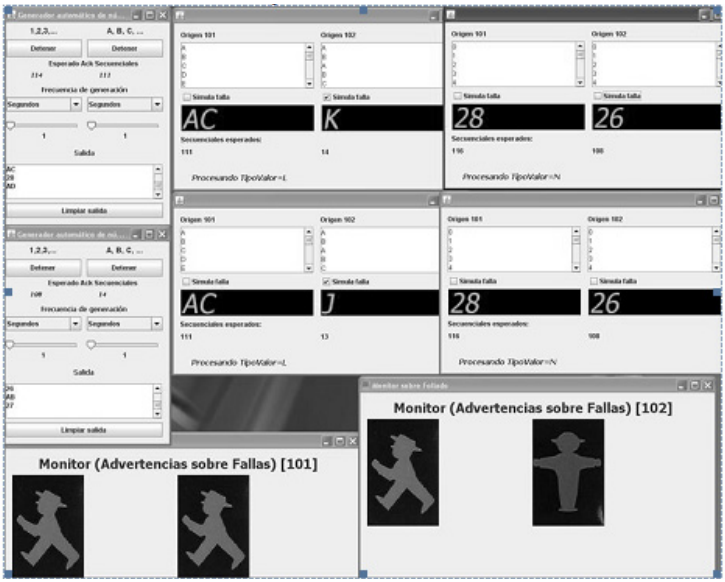



**Figure 10. Prototype Implementation**

All elements of the system are connected through a Data Field (DF). The DF is formed by four distributed and interconnected processes whose function is to allow the flow of messages between the elements connected to it. When system operation starts, all autonomous entities are connected to the DF. Generators entities begin to produce random numbers and letters information, which is injected into the DF. In turn, the Displayer will receive the messages flowing through the DF to take those who are responsible to process according to the Content Code and Pivot (number or letter), and will send receipts that flow through DF, according to the principles of transactionality and sequentiality. Generators take acknowledgments from the DF to send the next message according to the proposed technology.

Displayer can simulate fails, so the transactional principle will not be achieved, that is the minimum number of acknowledgments, Generators send alerts reaching Monitors through the DF. Monitors report the failure of any part of the system. When the element recovers, it starts the synchronization process with their peers and once it operates properly, to meet the principle of transactional, the Generator sends a message to stop the alert.

With this scheme we probe that the behavior of the proposed sequential and transaction technology operates as expected. On the other hand, it was possible to demonstrate that the features offered by

ADS like on-line expansion, fault tolerance and on-line maintenance are also satisfied with this proposal. Figure 10 shows the prototype.

## 5. Conclusion and Final Remarks

In this paper we analyze the characteristics of a conventional SOA and determine that it cannot meet some characteristics of mission critical applications such as high availability, continuous operation, high flexibility, high performance, etc. On the other hand, we explore the concept of ADS and its architecture, and we find that such requirements are satisfied by this system designing paradigm. We propose a novel approach of modeling an SOA with ADS (ADSOA). In order to ensure data consistency and high availability of the application, we propose the Low Coupling Synchronization and Transactional Delivery Technology. The effectiveness and feasibility of the ADSOA and proposed technology was tested by a prototype.

## 6. References

[1] M. Horst and T. Burg, "Business-critical soa-based services on nonstop servers." The Connection Magazine, Nov-Dec 2006.

[2] S. Hayward, "Positions 2005: Soa adds flexibility to business processes." Gartner, February 2005.

[3] K. Mori, "Autonomous decentralized software structure and its application." in Fall Joint Computer Conference (FJCC'86), pp. 1056{1063, November 1986. Dallas, Texas, USA.

[4] K. Mori and et al., "Proposition of autonomous decentralized concept," of IEEE, vol. 104, no. 12, pp. 303 - 310, 1984.

[5] IEEE Computer Society, 8th International Symposium on Autonomous Decentralized Systems (ISADS '07), March 2007. Sedona, USA.

[6] IEEE Computer Society, 9th International Symposium on Autonomous Decentralized Systems (ISADS '09), March 2009. Athens, Greece.

[7] M. E. A. and B. Michael, Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology. Wiley,

# Parallel Genetic Algorithms on Cluster Architecture: A Case Study

Sisnett Hernandez, Ricardo.
*Intel GDC [no longer there]*
*sisnett@tesisinteractive.com*

## Abstract

*Genetic Algorithms have represented a powerful tool to solve optimization problems, mainly those with multivariate or complex objective functions. However due to the random nature of the algorithms, some search spaces appear too big to be covered, and the utility of GA´s seems diminished. Parallel Genetic Algorithms provide a way of increasing the coverage of search space with almost none drawbacks on performance or implementation difficulty. This paper presents the parallelization of a genetic algorithm to solve a multivariate combinatorial problem using message passing paradigm and a 16-node cluster. We explore two different ways of taking advantage of HPC: Increase in the search space and reduction of time using multiple processors. We also explore and explain some of the drawbacks of the cluster architecture on this applications and how to tackle them.*

***Keywords:*** *Genetic Algorithms, EvolutionaryComputing,GeneticProgramming, HPC*

## 1. Introduction

### 1.1 Genetic Algorithms

Genetic Algorithms (GA) are a search technique based on Darwin's 'survival of the fittest', where a group of solutions are ranked depending on their fitness, a numeric value which calculation is problem dependant created by John Holland. A population is created and evaluated then its individuals are reproduced via crossover and mutated to create new individuals to replace the least-fit individuals [1]. GA's have been proved useful in a variety of experiments, mainly used in multi-objective optimization where traditional (mathematic) techniques fail to find an answer either because they get stuck into a local minimum/maximum or because the amount of time needed to compute makes them useless.

### 1.2 Parallel Genetic Algorithms

Parallel Genetic Algorithms (PGA) are the distributed version of GA's. Using the apparent independence of diverse populations genetic algorithms can be parallelized almost in a trivial manner, and with very good results, this parallelization yields either an improvement in the time

spent evaluating individuals or an increased coverage of the search space. There are two main versions of PGA's [2]:

*Fine Grain Parallelism or Cellular Model:* In this variant, each individual or a very small group of individuals is run in a different processing unit, and can only exchange genetic information (reproduce) with individuals in neighboring spaces.

*Coarse Grain Parallelism or Island Model:* This variant places more individuals per processing unit and exchanges information via a migration operator, this model appears more often when individual or generation evaluation can take a relatively long time.

### 1.3 Cluster Architecture

The concept of a cluster involves taking two or more computers and organizing them to work together to provide higher throughput, availability, reliability and scalability than can be obtained by using a single system [3].

Computers are connected using a high speed network interface (Gigabit Ethernet, Infiniband) and through a clustering software package they appear to the user as being just one entity with a lot of available resources.

For the experiments reported in this paper we used a 17 Node cluster, each node had 24 Gb Ram, 200 HDD @ 15000 RPM, Intel® Xenon® X7550 (8 Cores, 2Ghz), connected through a Gigabit Ethernet Switch. The clustering tool we used was Rocks+® Clustering Software Package, and we ran on RedHat Enterprise Linux.

## 2. Implementation

We implemented a traditional GA, using roulette for reproduction selection and 2 point cross for crossover; for parameters we used *mutation=0.15*, *reproduction=0.8* and *generation size=100* we ran for 100 generations, for all experiments the maximum possible fitness is 10,000.

For the parallel version, we chose coarse grain parallelism using each processor as an island and migrating after a fixed number of generations. For this we used the Message Passing Interface library (MPI) mainly because message passing paradigm models perfectly the idea of migration, adds very little overhead, is scalable and keeps blocking communication to a minimum. Migration occurred every 10 generations, and all populations migrated a random number of individuals in a random direction, this direction would be the same for this migration for all processors, i.e. all processors would migrate to the "EAST" between generation 10 and 11 and then all would migrate "NORTH" between generation 20 and 21.

At startup we would create a logical adjacency map based on the ids provided by MPI to each process. This grid-like map would then be turned into a torus to assure all processes have four neighboring processes with whom they could exchange individuals.

## 3. Experiments & Results

For this paper we designed two main experiments that show the benefits of using cluster architecture with PGA's we denominated them *Time Attack* and *Search*

*Space Increment* we present results and conclusions for each below.

### 3.1 Time Attack

For this experiment we increased the number of processors but kept the total amount of individuals, thus decreasing the population size as we added processors. This experiment showed a direct gain while adding processors, showing that the Amdalh's non-parallel section of GA's is considerably small when compared to the parallel section. Figure 1 shows the relation between linear speedup and our speed up.

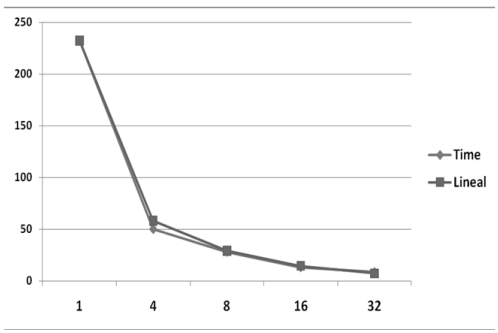



**Figure 1. Relation between time (in minutes) and the number of processes used to run the same population size.**

On Figure 2 showing fitness vs number of nodes, we realize that the last case, for the one we used 32 nodes, we have a considerable decrease of fitness on the best individual, this problem arises due to the small size of population that led to a lack of genetic variability. However in the same chart we show improvements if we either increment the probability of mutation (Point A, mutation 35%) or if we increase the number of migrations during the experiment (Point B, 100% increase of migration).

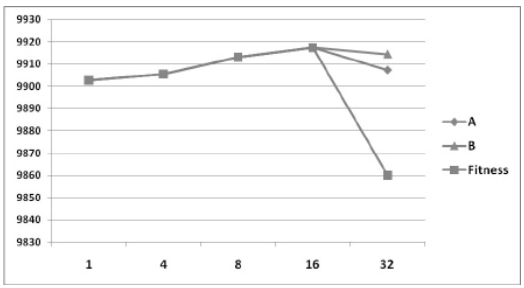



**Figure 2. Relation between fitness and number of nodes used in the 'Time Attack' experiment, A represent the same experiment on 32 nodes but increased mutation and B with twice as much migration**

### 3.2 Search Space Increment

For this experiment we increased the number of processors but kept the same amount of individuals for each one, experiment ran in the same time for all cases but overall fitness saw an increment while adding nodes. As shown on Figure 3 we can see that adding nodes increased the max fitness or the average fitness, which can be translated in a better understating of the search space.

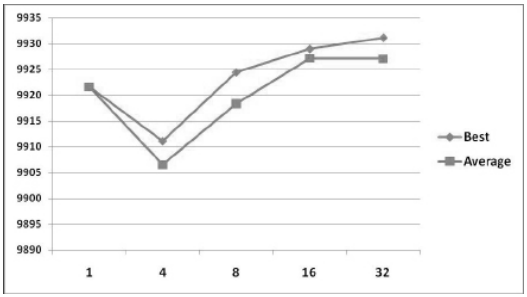



**Figure 3. Relation between number of nodes and average and best fitness among nodes**

### 3.2.1 An in-depth look into the 32 Node run

For the biggest run of the search space coverage we present the final layout of the best individuals for each processor. In Figure 4 you can observe that populations with the best individuals tend to be together, while a line of under fitted populations can be seen (Sixth Row). This is one of the main issues of PGA's as good populations tend to be together or close leaving other populations under developed, this also translate as a loss in processing time since under-developed populations are less likely to provide a good answer but still are spending CPU and other resources.

**Fitness Distribution**

| | | | |
|---|---|---|---|
| 9919.34 | 9919.27 | 9919.27 | 9919.34 |
| 9923.07 | 9923.07 | 9919.27 | 9919.27 |
| 9923.1 | 9923.1 | 9918.73 | 9923.08 |
| 9923.11 | 9923.08 | 9919.01 | 9923.11 |
| 9923.08 | 9923.1 | 9918.67 | 9923.08 |
| 9917.18 | 9917.18 | 9917.18 | 9917.18 |
| 9919.48 | 9919.27 | 9919.27 | 9919.48 |
| 9919.48 | 9919.27 | 9919.27 | 9919.48 |

**Figure 4. Distribution of fitness in the process geography. Processes on top row can migrate to the bottom row and vice-versa, this is also true for left-right most processes, generating a torus shape.**

## 4. Issues & Future Work

As mentioned before, the main problem with PGA's is the creation of clusters of underdeveloped populations, this ends up being ineffective as resources are spent in this populations which might not improve on the fitness of the most effective ones, this is especially true at the final stages of the GA.

Genetic Algorithms part from the premise that all genetic material is available at all times, and partitioning it yields different problems as we have seen. However, this is not necessarily true in the biological process in which GA's are supposed to be based; fittest individuals are not always present in every single population at the moment of mating. Our future work will center on solving this problem, either by current techniques or by proposing a new way.

## 5. Acknowledgements

I would like to thank Intel® GDC for the processing time of its HPC cluster in which all this experiments were run. I would also like to thank PhD Marco Antonio de Luna from ITESM Campus Guadalajara for lending his code for MEWMA Optimization problem which we attacked with this PGA.

## 6. References

[1] Holland J.H., *Adaptation in natural and artificial system,* Ann Arbor, The University of Michigan Press.

[2] Nowostawski, Mariusz. Poli, Riccardo. *Parallel Genetic Algorithm Taxonomy.*

[3] Guangzhon Sun. *A Framework for Parallel Genetic Algorithms on PC Cluster.* Proceedings of the 5th WSEAS Int. Conf. ( pp274-278)

Cantu-Paz, Erick. Topologies, Migration Rate and Multi-Population Parallel Genetic Algorithms.

Guan, Yu. Xu, Baowen. Leung, Karl. Parallel Genetic Algorithms with Schema Migration.

Kazunori, Ishigame, Chakraborty, Hatsuo, and Makin. Asynchronous Parallel Distributed Genetic Algorithm with Elite Migration. International Journal of Information and Mathematical Sciences 4;2 2008.

Koza, Jhon. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press.


ISUM
2011
2nd INTERNATIONAL SUPERCOMPUTING
CONFERNCE IN MEXICO
CONGRESO DE SUPERCÓMPUTO

# GRIDS

# Probabilistic Scheduler for General Purpose Clusters

Ismael Farfán Estrada
*Instituto Politécnico Nacional*
*Centro de Investigación en Computación*
*ifarfane0900@ipn.mx*

## Abstract

*This work presents a proposal for a scheduler which dynamically modifies the user run-time estimates, based in the statistical behavior, in a way that it's possible to make a load-balancing of the jobs such that there exists a possibility P(s) that the schedule s is respected and reduce the need to use back-filling algorithms to fill the gaps resulting from bad estimations; without negatively impacting the utilization of the cluster or wasting computing time in a schedule which won't be honored.*

***Keywords:*** *scheduling, statistics, resource manager, clusters*

## 1. Introduction

The issue of scheduling in any type of cluster is a very important one because depending on the scheduling policies used the utilization of the cluster as well as the performance of the scheduler varies.
There exists lots of scheduling algorithms among which we can highlight First Come First Served (FCFS), Shortest Job First (SJF) and Longest Job First (LJF), all of them improve greatly when used together with Backfilling politics, either aggressive or conservative.

All of this scheduling algorithms, including backfilling, requires the user estimate of the runtime[4] to make the schedule as good as possible, the problem is that most of the time this estimations aren't accurate, which make the schedule really dynamic, ie: the jobs finish earlier than expected and there is the need to repeatedly use backfilling to fill the gaps resulting from this poor prediction.

In this work we propose a workaround for the issue of the bad estimates from the perspective that the runtime of the jobs can be categorized in a finite amount of "similar" jobs with similar expected runtime[5] as suggested by the statistical run time distribution resulting from real workload logs that will be studied.

This paper is distributed as follows: in part 2, the selection of the middleware to be used will be discussed, in part 3 we'll talk a little about the way that FCFS with back-filling works, in part 4 we present how this proposal is expected to work compared to the usual way, in part 5 we present a little analysis of the workload of the ANL and the statistical models to be used, finally in part 6 the conclusions and further work.

## 2. Middleware for clusters

Due to the size of most clusters, it's unpractical to expect the users to verify the availability of the nodes by hand and then execute their scripts, or the administrator to manually search the logs to verify utilization, that's why a really complex piece of software is needed. This software is the middleware which will be described in this section.

The middleware is a sophisticate collection of software which sits in between the users, the administrators and the nodes of the cluster[10, 12], this software makes easier the administration and use of the cluster since it provides various tools with different levels of complexity for different purposes.

The administrator can retrieve the most relevant characteristics from the nodes with a simple command, can define job queues with certain priorities within a configuration file or equivalent, it's possible to define permissions per users or groups to each queue and also it's possible to define subsets of nodes accessible for each user or for use for each queue.

The software provided by the middleware saves time for the user as well, the user doesn't need to contact the administrator to verify the characteristics of the computing nodes, so he can more easily adapt his software to the characteristics of the cluster, also it's easier to request resources because a simple script is used in which the user specifies the amount of nodes, cores, RAM memory, disc or even GPUs he needs. Finally but not least important, the middleware provides an easy way to execute parallel programs with most MPI flavors, instead of having to specify the number of processes to starts in each allocated node by hand, the user issues the "run" command and the middleware does the job of actually starting the process in the appropriate nodes.

To emphasize the importance of the middleware, let's notice that some cluster providers and hardware manufacturers develop their own middleware for clusters like the one shown in figure 1 and many laboratories develop their own too, each one of them with their own characteristics, some may care about graphic monitor tools and other about flexibility.

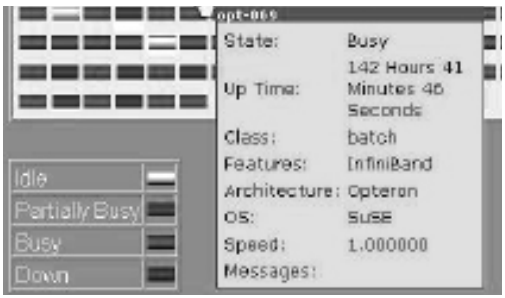



**Figure 1: Moab Access Portal showing the state of the nodes using colored squares, and some statistics of a node.**

Since one of the most important functions of the middleware is to administer the cluster, it's also referred to as resource manager; there is not a single rule to decide which one to use it's up to the needs of each individual and organization.

### 2.1. SLURM resource manager

For this work, various resource managers where considered, among them

PBS, TorquePBS, Oscar and SLURM[2]. No comparison among them will be provided, just some facts about the reason why we decided to use the Simple Linux Utility for Resource Management (SLURM).

SLURM is a modular middleware that provides a well defined set of demons for servers and computing nodes, and commands for users and administrators as can be seen in figure 2.

This modular construction of SLURM makes it a good choice for a developing project since it's not necessary to search through all the source code, one only needs to study the part in which one is interested and ignore the rest.

Since this proposal is based in the statistic behavior, it's especially important the fact that SLURM provides a medium sized database with lots of statistics and runtime information on per job, per user and per queue basis as is shown in figure 3.

Lastly, SLURM is extensible by means of plugins with a structure defined by the developers of this middleware, this fact reduces the work of software engineering needed to make the extension suggested, since it's only needed to implement the functions defined and documented in the developer manual.

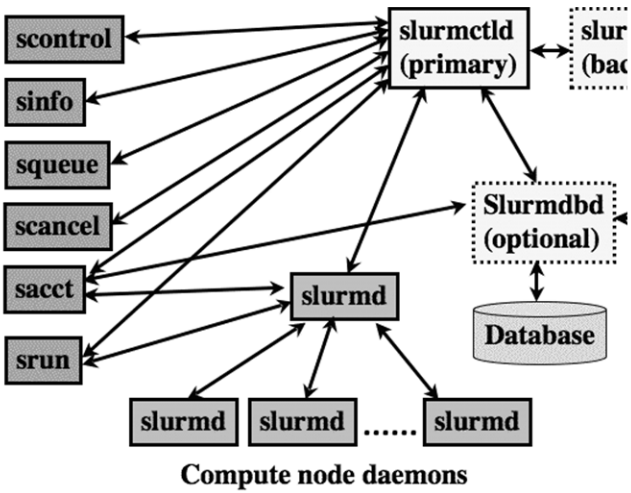



**Figure 2: Overview of the architecture of SLURM showing some commands (left) and the main daemons with connection to the external database system.**

An example of this extensibility can be seen in the definition of the following functions that must be implemented to make a new scheduler plugin:

```
int slurm_sched_plugin_schedule (void);
int slurm_sched_plugin_newalloc (void);
int slurm_sched_plugin_freealloc (void);
```

As the names suggest, each function must implement a specific behavior for the scheduler plugin, thus saving time of engineering.

## 3. Scheduling algorithms

Since the development of the assembly line plenty of scheduling algorithms have been created[1] to make as efficient as possible the use of the various machines that can be (or not) working at and during certain period of time.

Some of these algorithms have been ported to use in cluster environments, some of the most popular are First Come First Served (FCFS), Longest Job First (LJF) and Shortest Job First (SJF). This algorithms are very straight forward, FCFS states that the jobs must be taken care of in the same order in which they arrived; LJF give preference to those jobs that require the greatest amount of time to complete as opposed to SJF which give preference to the jobs that will be dispatched sooner, therefore finishing as many as possible as soon as possible.

As simple as this algorithms are, they tend to let many "holes" in the scheduling, this can be an issue when the objective is to have as many machines working most of the time, ie. Maximize the utilization of the cluster.

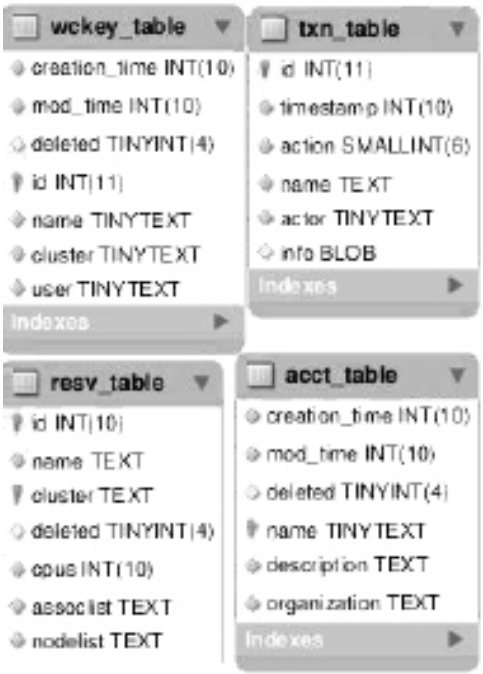



**Figure 3: The log database of SLURM consists of 27 tables with node/cluster events, statistics, user/job information, etc.**

To attend this problem, the backfilling algorithm is used. First whichever algorithm or policy was selected do the initial scheduling, upon finishing backfilling searches for holes in the schedule, for each hole found, backfilling selects one or more jobs that fit within these holes, so they actually starts ahead of the time they were supposed to start.

Backfilling comes in two flavors: aggressive and conservative[11]. In conservative backfilling every job gets a reservation and smaller jobs are allowed to start ahead as long as they don't delay

the reservations of other jobs, aggressive backfilling will make reservation only for the next job in line or a handful of them and allow any job to start ahead if it doesn't delay other jobs.

An example of the way FCFS with backfilling works can be seen in figure 4; lets suppose that those 6 jobs arrive in the same order they are numbered, the estimated time is the whole colored rectangle, the darkest side is the real runtime and the brightest one is the "inaccuracy" (=100-accuracy) of the estimations; accuracy =100·runtime/estimate.

As we can see in figure 4, first FCFS will schedule the jobs such that job 3 will start at time 12, however job 2 finishes before scheduled at time 4, the resulting hole is filled with job 4, then job 1 finisher earlier to at time 6 and the hole is filled with job 5 and so on. In the end job 3 actually starts at time 10, but lets notice that the earliest time that job 3 could have started is 6, so it was postponed 4 units of time in favor of the smallest jobs.

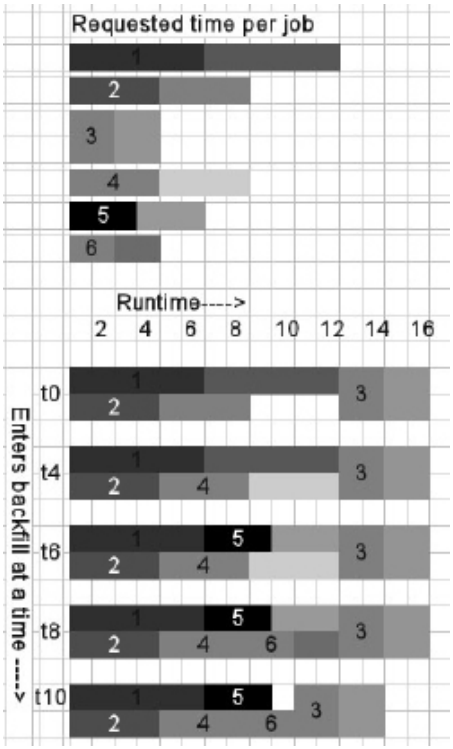



**Figure 4: 6 jobs running on 2 nodes scheduled with FCFS and backfilling, the darkest color represent the real runtime.**

The fact that every time a job finisher earlier than expected makes backfilling try to fill the holes in the schedule, has at least 2 implications: the schedule is really dynamic in the sense that most of the jobs finish earlier and therefore the gaps should be filled, and the second is that the scheduler may run out of small jobs for backfilling[6] since, as can be seen in figure 4 with the jobs 4 and 6, many small jobs may fit in the hole left by one big job.

## 4. Modifying runtime estimates

In reality, and in contrast with the conservative "inaccuracy" of the jobs in figure 4 (real runtime x 2), the inaccuracy of the estimates is usually greater, and not only that, but also the estimations are modal[5], which means that the requested execution time falls in a finite (and small) set of values like 10, 30, 60 minutes, 2, 4, 6 hours and maxim allowed runtime just to mention some.

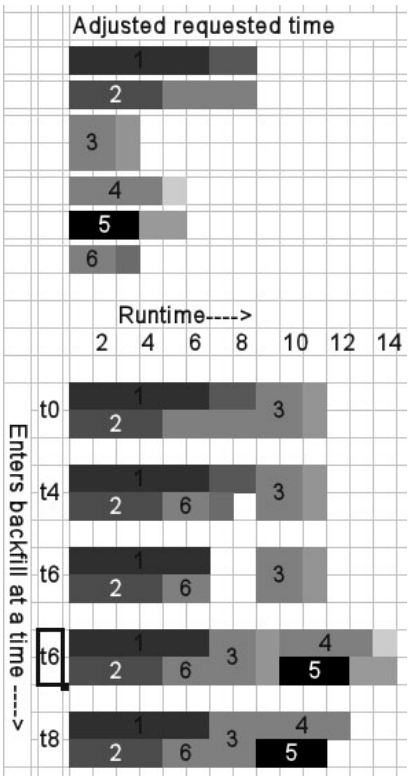



**Figure 5: Adjusting the requested time would make the jobs to leave smaller holes that are**

The modal requested time makes the initial scheduling easier, but the holes left are more irregular making them harder to fill; the fact that the inaccuracy is many times the real runtime generates other problems too.

In real workload logs we can see jobs with real runtimes of about 15 minutes that requested about 4 hours of execution[5], that means an inaccuracy of about 93.75%, this is a problem because it means that, at certain point, small jobs won't be selected to fill the gaps because request many times more the time they needed (in this case 10 times).

To solve this problem of (very) dynamic schedules and the extra work of keep backfilling the holes left by the jobs used to fill other holes, we propose to change the requested runtime for an expected run time based in the statistical behavior of the jobs.

Lets suppose that we start with the same jobs that where shown in figure 4, but this time before scheduling them we first adjust the runtime estimates leaving them as show in figure 5, then we do the usual FCFS scheduling letting job 3 start at time 8 (instead of at time 12 as in the former example), when job 2 finishes we use this time job 6 to fill the gap, then both job 1 and 6 finishes and it allows job 3 to start at time 6 which is the earliest time it could have started.

At this point the problem is to calculate what time can we expect a job to finish its execution, as was already proved, the runtime of various jobs of the same user tend to be the same[5], thereof once we have enough statistics we can calculate the expected execution time of a job with 1σ or 2σ of certainty.

By using the statistical behavior of the jobs of the user to calculate the expected finishing time we can create a more stable schedule which respects as much as possible the arrival order of the jobs in the case of FCFS without reducing the utilization of the cluster.

Another advantage of this method is the feasibility of being able to calculate the probability that exists of the final schedule being as similar as possible to the one that was calculated by the selected scheduling algorithm, since for each job we have 1σ or 2σ of certainty that it will take that much time to finish.

As suggested by Tsafrir [13], the predicted runtime of the job will be used only for scheduling purposes, allowing overlaps in the schedule, in the event that a job actually runs for more time than the predicted, a proper adjustment has to be made, in this case, falling back to the user estimate.

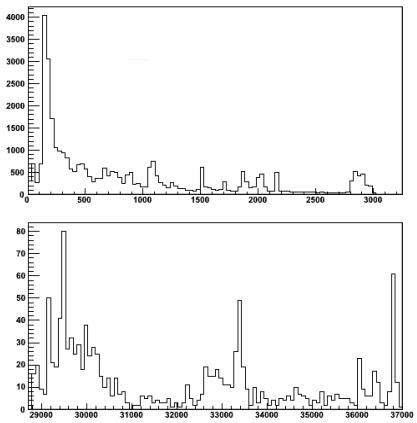



**Figure 6: In real workloads we can see a tendency for the real runtime to fall near one of many groups that can vary in width depending of the time.**

## 5. Behavior of real work load logs

The revision of the log of the ANL 2009 shows some interesting behaviors: from the users with tens consecutive of jobs of the same runtime to chaotic users which are hard to predict as noticed in previous works [5, 13].

For different ranges of time, we can think of a distribution to be used, for example in figure 6 we can see exponential and Gaussian in certain zones, we believe that once we can make a fit of this statistics we can use it to predict the runtime for a given job.

In order to detect this potential sets or categories we could use algorithms like k-means, but for this work we decided to use multimodal analysis[7], which is an statistical tool to detect zones around the which many samples fall.

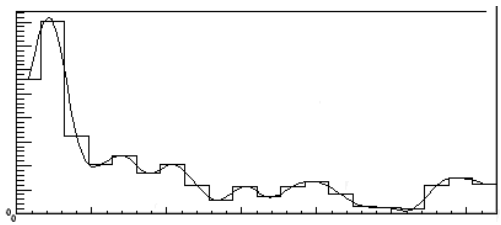



**Figure 7: Interpolation of a 20 bin curve using splines using the center of the bins as the known points.**

In case that it's not possible to fit the distribution to a known one, it is possible to do an interpolation of the data, for this various interpolation techniques where tried: Gaussian, Vandermonde polynomial, and splines. Of these techniques, splines [14] is the one that gives a smoother curve as shown in figure 7.

Having done the interpolation we convert it into a probability density function as follows: Let's call I(N) the (numerical) integration from the middle of bin 1 to the middle of bin N being N the total number of bins, and let's call I(i) the (numerical) interpolation from bin 1 to bin i for i<=N. The accumulate possibility P(i) up to bin i is P(i)=I(i)/I(N).

Another interesting possibility given by the analysis of real runtime logs is the dependency among different characteristics of the jobs, that is, among different random variables[8]. For the resource manager that we are to use at least 4 pieces of information with possible correlation can be detected, this are:

- User
- Real runtime
- Number of cores
- Queue

In order to verify the possible dependency among some of those parameters, the real time and the number of cores used by each job in the log of the ANL where multiplied giving a pseudo-gaussian distribution of times as can be seen in figure 8.

The classical statistical approach is not enough to detect efficiently the dependencies among parameters, for this reason a more appropriate tool is needed. This tool is the Copula model.

The Copula model[9] is a tool used to understand relationship between multivariate distributions. This relationships con be of various types since the model has defined a copula for everyone of the most common relationships among variables.

In the recent years, the copula model has been used extensively in the area of macroeconomics where a lot of random variables are analyzed and is important do detect possible correlation among them in order pro make accurate predictions of the movement of the markets.

For the present work we want to detect the possible relationship between up to 4 random variables, task that should be easy compared to the macroeconomics analysis and its tens of variables.

For the graph shown in figure 8 we can expect the kind of copula that describes a Gaussian correlation for the runtime and the number of cores used, in this case, having the number of cores requested by the user we can use the appropriate copula to calculate the expected real runtime based in this correlation.

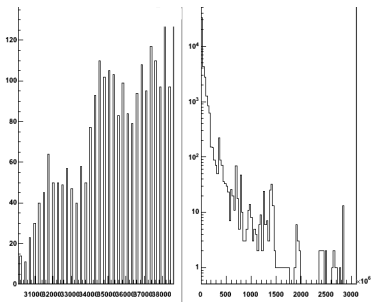



**Figure 8: There seems to exist some dependency between real runtime and number of cores.**

This same process can be extended to the other 2 random variables, the dependence doesn't need to be limited to pairs of variables, copula model allow the

detection of chain dependencies, which means that we can search for a correlation between real runtime and number of cores used, number of cores used and queue to the which the job was submitted, and the queue with the user in question. Also we can detect dependences in form of branches, for example the real runtime depends of the user, the number of cores requested and the queue in which it was submitted.

## 6. Conclusions and future work

The actual way of scheduling jobs in a general purpose cluster which uses backfilling gives preference to small jobs over logs jobs instead of respecting the schedule approach selected by the cluster administrator because of the inaccuracy of the runtime estimates which in turn make the schedule very dynamic by requiring constant backfilling of the holes left by jobs finishing earlier.

Our approach to make a more stable scheduling of jobs in a general purpose cluster looks very promising due to the fact that it has mathematical basis which in this case comes from the statistical approach with multivariate analysis and the copula model.

All the information needed to generate the statistics is already provided by SLURM resource manager in a relational database along with the mechanisms to access it.

The actual write of the scheduler plugin will be made easier due to the fact that there exist a well defined interface which needs to be programmed in order to make a personalized scheduler for SLURM saving lots of time of software engineering.

The remaining activities are the actual writing of the scheduler and implementation of the algorithms to detect the distributions of copulas of certain sets of data.

## 7. References

[1] Haupt, R., "A survey of priority rule-based schduling", *OR Spectrum*, Springer, 1989, pp. 3-16.

[2] SLURM: A highly scalable re-source manager. computing.llnl.gov/linux/slurm/.

[3] Brown, R. G., *Engineering a Beowulf-style compute cluster*, 2004.

[4] Tsafrir, D. and Feitelson, D. G., "The Dynamics of Backfilling: Solving the Mystery of Why Increased Inaccuracy May Help", *iiswc*, IEEE, 2006, pp. 131-141.

[5] Tsafrir, Dan, Etsion, Yoav and Feitelson, Dror G., "Modeling User Runtime Estimates", *Job Scheduling Strategies for Parallel Processing*, Springer Verlag, 2005, pp. 1-35.

[6] Tsafrir, Dan "Using Inaccurate Estimates Accurately", *Job Scheduling Strategies for Parallel Processing*, Springer Verlag, 2010, pp. 208-221.

[7] Manski, Charles F., *Partial Identification of Probability Distributions*, Springer, 2001.

[8] Mardia, K. V., Kent, J. T., Bibby, J. M., *Multivariate Analysis*, Academic Series, 1979.

[9] Nelsen, Roger B., *An Introduction to Copulas (2nd)*, Springer, 2006.

[10] Smith, Norris Parker, *In Search of Clusters (2nd)*, Prentice Hall, 1998.

[11] Srinivasan, S. R., Kettimithu, V. and Sadayappan, P. "Characterization of backfilling strategies for parallel job scheduling".

[12] Bishop, Tony A. and Karne, Ramesh K. "A Survey of Middleware", 18th International Conference on Computers and Their Applications, 2003.

[13] Tsafrir, Dan, Etsion, Yoav, Feitelson, Dror G. "Backfilling Using System Generated Predictions Rather Than User Runtime Estimates", *Transaction on Parallel and Distributed Systems*, IEEE, June 2007, vol. 18, num. 6.

[14] Ahlberg, Nilson, and Walsh "The Theory of Splines and Their Applications", 1967.

# Management and Monitoring of Large Datasets on Distributed Computing Systems for the IceCube Neutrino Observatory

J. C. Díaz-Vélez
*University of Wisconsin-Madison*
*juancarlos.diazvelez@icecube.wisc.edu*

## Abstract

*IceCube is a one-gigaton neutrino detector designed to detect high-energy cosmic neutrinos. It is currently in its final phase of construction at the geographic South Pole [1,2]. Simulation and data processing for IceCube require a significant amount of computational power. We describe the design and functionality of IceProd, a management system based on Python, XMLRPC and GridFTP. It is driven by a central database in order to coordinate and administer production of simulations and processing of data produced by the IceCube detector upon arrival in the northern hemisphere. IceProd runs as a separate layer on top of other middleware and can take advantage of a variety of computing resources including grids and batch systems such as GLite, Condor, NorduGrid, PBS and SGE. This is accomplished by a set of dedicated daemons which process job submission in a coordinated fashion through the use of middleware plug-ins that serve to abstract the details of job submission and job management. We describe several aspects of IceProd's design including security, data integrity, scalability and throughput as well as the various challenges in each of these topics.*

***Keywords:*** *Data Management, Grid Computing, Monitoring, Distributed Computing*

## 1. Introduction

Large experimental collaborations often need to produce large volumes of computationally intensive Monte Carlo simulations as well as data processing. For such large datasets, it is important to be able to document software versions and parameters including pseudo-random generator seeds used for each dataset produced. Individual members of such collaborations might have access to modest computational resources that need to be coordinated for production and could be pooled in order to provide a single, more powerful, and more productive system that can be used by the entire collaboration. For this purpose, we have designed a software package consisting of queuing daemons that communicate via a central database in order to coordinate production of large datasets by integrating small clusters and grids.

## 2. IceCube

The IceCube detector shown in figure 1 consists of 5160 optical modules buried between 1450 and 2450 meters below the surface of the South Polar ice sheet and is designed to detect neutrinos from astrophysical sources [3]. However, it is also sensitive to downward-going muons produced in cosmic ray air showers with energies in excess of several TeV. IceCube records ~10^10 cosmic-ray events per year. These cosmic-ray-induced muons represent a background for most IceCube analyses as they outnumber neutrino-induced events by about 500 000:1 and must be filtered prior to transfer to the North due to satellite bandwidth limitations [3]. In order to develop reconstructions and analyses, and in order to understand systematic uncertainties, physicists require a comparable amount of statistics from Monte Carlo simulations. This requires hundreds of years of CPU processing time.

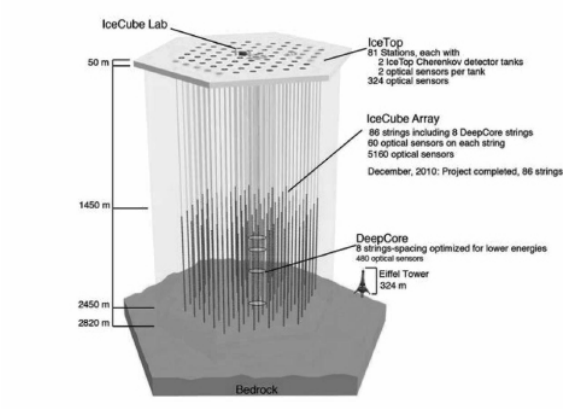



**Figure 1: The IceCube detector**

### 2.1. IceCube Computing Resources and Demands.

The IceCube collaboration is comprised of 36 research institutions from Europe, North America, Japan, and New Zealand. The collaboration has access to 24 different clusters and grids in Europe, Japan and the U.S. These range from small computer farms of 30 nodes to large grids such as the European *Enabling Grids for E-sciencE* (EGEE), *Louisiana Optical Network Initiative* (LONI), *Grid Laboratory of Wisconsin* (GLOW), SweGrid and the *Open Science Grid* (OSG) that may each have thousands of compute nodes. The total number of nodes available to IceCube member institutions is uncertain since much of our use is opportunistic and depends on the usage by other projects and experiments. In total, IceCube simulation has run on more than 11,000 distinct nodes and a number of CPU cores between 11,000 and 44,000. On average, IceCube simulation production has run concurrently on ~4,000 cores at a given time and we anticipate running on ~5,000 cores simultaneously during upcoming productions. In order to accomplish this, we have developed a dataset management software package called IceProd.

## 3. IceProd

IceProd is a software package written in Python and designed to manage, run, control and monitor the production of IceCube detector simulation data and related filtering and reconstruction analyses. It provides a graphical user interface (GUI) for configuring simulations and submitting

jobs through a production server. It provides a method for recording all the software versions, physics parameters, system settings and other steering parameters in a central production database. IceProd also includes an object-oriented web page written in PHP for visualization and live monitoring of datasets.

The package includes a set of libraries, executables and daemons that communicate with the central database and coordinate to share responsibility for the completion of tasks. Because of this, IceProd can thus be used to integrate an arbitrary number of sites including clusters and grids at the user level. It is not however, a replacement for Globus, GLite or any other middleware. Instead, it runs on top of these as a separate layer with additional functionality. The software package can be logically divided into the following components illustrated in Figure 2:

1. *Soaptray* - a server that received client requests for scheduling jobs and steering information.
2. *Soapqueue* - a daemon that queries the database for tasks to be submitted to a particular cluster or grid.
3. *Soapmon* - a monitoring server that receives updates from jobs during execution and performs status updates to the database.
4. *Soapdh* - a data handler/garbage collection daemon that takes care of cleaning up and performing any post processing tasks.
5. A database that stores configured parameters, libraries (including version information), job information and performance statistics.
6. A client (both graphical and text) that can download, edit and submit dataset steering files to be processed.
7. A PHP web application for monitoring and controlling dataset processing.

### 3.1. IceProd Server

The IceProd server is comprised of the four daemons mentioned in the list above (items 1-4) and their respective libraries. There are two basic modes of operation: the first is a non-production mode in which jobs are sent to the queue of a particular system, and the second stores all of the parameters in the database and also tracks the progress of each job. The *soapqueue* daemon running at each of the participating sites periodically queries the database to check if any tasks have been assigned to it. It then downloads the steering configuration and submits a given number of jobs. The size of the queue at each site is configured individually based on the size of the cluster and local queuing policies.

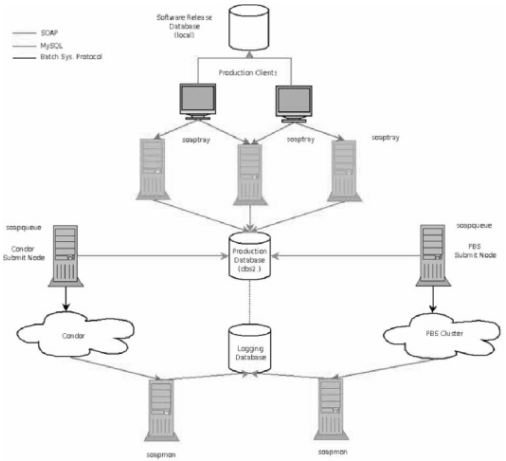



**Figure 2: Network diagram of IceProd system.**

### 3.2. Plug-ins

In order to abstract the process of job submission for the various types of systems, IceProd defines a base class *I3Grid* that provides an interface for queuing jobs. Other classes known as *plug-ins* then implement the functionality of each system and provide functions for queuing and removing jobs, status checks and include attributes such as job priority, max wall time and job requirements such as disk, memory, etc. IceProd has a growing library of plug-ins that are included with the software including Condor, PBS, SGE, Globus, GLite, Edg, SweGrid and other batch systems. One can additionally implement user-defined plug-ins for any new type of system that is not included in this list.

## 4. XML Job Description

In the context of this document, we define a dataset to be a collection of jobs which share a basic set of scripts and software but whose input parameters depend on the enumerated index of the job. A configuration or steering file describes the tasks to be executed for an entire dataset. IceProd steering files are XML documents with a defined schema. This document includes information about the specific software versions used for each of the sections known as *trays* (a term borrowed from the C++ software framework used by IceCube [4]), parameters passed to each of the configurable modules and input files needed for the job. In addition, there is a section for user-defined parameters and expressions to facilitate programming within the XML structure. This is discussed further in section 5.

### 4.1. IceProd expressions

A limited programming language was developed in order to allow more scripting flexibility that depends on runtime parameters such as job index, dataset ID, etc. This allows for a single XML job description to be applied to an entire dataset following a SPMD (Single Process, Multiple Data) operation mode. Examples of valid expressions include the following:

1. $args() – a command line argument passed to the job (such as job ID, or dataset ID)
2. $steering() – a user defined variable

3. $system() – a system-specific parameter defined by the server
4. $eval() – a mathematical expression (Python)
5. $sprintf() – string formatting

The evaluation of such expressions is recursive and allows for a fair amount of complexity. There are however limitations in place in order to prevent abuse of this feature. An example of this is that $eval() statements prohibit such things as loops and import statements that would allow the user to write an entire python program within an expression. There is also a limit on the number of recursions in order to prevent closed loops in recursive statements.

## 5. IceProd Modules

IceProd modules, like plug-ins, implement an interface defined by a base class *IPModule*. These modules represent the atomic tasks to be performed as part of the job. They have a standard interface that allows for an arbitrary set of parameters to be configured in the XML document and passed from the IceProd framework. In turn, the module returns a set of statistics in the form of a string to float dictionary back to the framework so that it can be automatically recorded in the database and displayed on the monitoring web page. By default, the base class will report the module's CPU usage but the user can define any set of values to be reported such as number of events that pass a given processing filter, etc. IceProd also includes a library of predefined modules for performing common tasks such as file transfers through GridFTP, tarball manipulation, etc.

### 5.1. External Modules

Included in the library of predefined modules is a special module *i3,* which has two parameters, *class* and *URL*. The first is a string that defines the name of an external IceProd module and the second specifies a Universal Resource Locator (URL) for a version-controlled repository where the external module's code can be found. Any other parameters passed to this module are assumed to belong to the referred external module and will be ignored by the i3 module. This allows for the use of user-defined modules without the need to install them at each IceProd site. External modules share the same interface as any other IceProd module.

### 5.2. Pilot Job

One of the complications of operating on heterogeneous systems is the diversity of architectures and operating systems and compilers. For this reason we make use of Condor's NMI-Metronome build and test system [5] for building the IceCube software for a variety of platforms. IceProd sends a Job Execution Pilot (JEP), a Python script that determines what platform it is running on and after contacting the monitoring server, determines which software package to download and execute. During runtime, this executable will perform status updates through the monitoring server via XMLRPC, a remote procedure call protocol that works over the Internet [6]. This

information is updated on the database and is displayed on the monitoring web page. Upon completion, the JEP will clean up its workspace but if configured to do so, will cache a copy of the software used and make it available for future runs. When caching is enabled, an md5sum check is performed on the cached software and compared to what is stored on the server in order to avoid using corrupted or outdated software. Jobs can fail under many circumstances. These can include submission failures due to transient system problems and execution failures due to problems with the execution host. At a higher level, errors specific to IceProd include communication problems with the monitoring daemon or the data repository. In order to account for possible transient errors, the design of IceProd includes a set of states through which a job will transition in order to guarantee a successful completion of a well-configured job. The state diagram for an IceProd job is depicted in Figure 3.

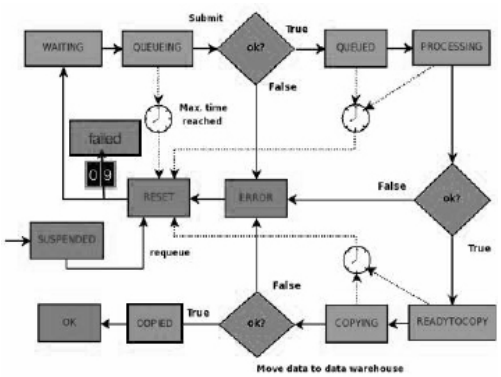



**Figure 3: State diagram for the JEP**

Each of the non-error states through which a job passes includes a configurable timeout. The purpose of this timeout is to account for any communication errors that may have prevented a job from setting its status correctly.

## 6. Client

The IceProd-Client contains two applications for interacting with the server and submitting datasets. One is a Gtk-based GUI and the other is a text-based application that can run as a command-line executable or as an interactive shell. Both of these applications allow the user to download, edit, submit and control datasets running on the IceProd-controlled grid. The graphical interface includes drag and drop features for moving modules around and provides the user with a list of valid parameter for know modules. Information about parameters for external modules is not included since these are not known a priori. The interactive shell also allows the user to perform grid management tasks such as starting and stopping a remote server, adding and removing participation of specific sites in the processing of a dataset as well as job-specific actions such as suspend and reset.

## 7. Monitoring

The status updates and statistics reported by the JEP via XMLRPC and stored in the database provide useful information for monitoring not only the progress of datasets but also for detecting errors. The monitoring daemon *soapmon* is an HTTP daemon that

listens to XMLRPC requests from the running processes (instances of JEP). The updates include status changes and information about the execution host as well as job statistics. This is a multi-threaded server that can run as a stand-alone daemon or as a cgi-bin script within a more robust Web server. The data collected from each job can be analyzed and patterns can be detected with the aid of visualization tools as described in the following section.

### 7.1. Web Interface

The web interface for IceProd works independently from the IceProd framework but utilizes the same database. It is written in PHP and makes use of the CodeIgniter object oriented framework [7]. The IceCube simulation and data processing web monitoring tools provide different views that include, from top level downwards;

1. Grid view: which displays everything that is running a particular site,
2. Dataset view: all jobs and statistics for a given dataset including every site that it is running on, and
3. job view: each individual job including status, job statistics, execution host and possible errors.

The web interface also uses XMLRPC in order to send commands to the *soaptray* daemon and provides authenticated users the ability to control jobs and datasets. Other features include graphs displaying completion rates, errors and number of jobs in various states.

## 8. Security and Data Integrity

IceProd integrates with an existing LDAP server for authentication. If one is not available, authentication can be done with database accounts though the former is preferred. Both *soaptray* and *soapmon* can be configured to use SSL certificates in order to encrypt all data communication between client and server. This is recommended for client-*soaptray* communication but is not necessary for monitoring information sent to *soapmon* as this just creates a higher CPU load on the system. In order to guarantee data integrity, an MD5sum or digest is generated for each file that is downloaded or uploaded. This information is stored in the database and is checked against the file after transfer. Data transfers support several protocols but we primarily rely on GridFTP which makes use of GSI authentication [8,9]. An additional security measure is the use of a temporary random-generated string that is assigned to each job at the time of submission. This *passkey* is used for authenticating communication between the job and the monitoring server and is only valid during the duration of the job. If the job is reset, this *passkey* will be changed before a new job is submitted. This prevents stale jobs that might be left running from making monitoring updates after the job has been reassigned. It also decreases the likelihood of a malicious attempt to inject data on the monitoring database.

## 9. Active Development

There is continued development of the IceProd framework. Current work includes support for directed acyclical graphs (DAGs) distributed across multiple sites and a distributed database for improved scalability and fault tolerance. In addition, we are removing current interdependencies with the IceCube software framework in order to make it available for more general use.

## 10. Conclusions

Simulation and data processing for large scientific collaborations requires a significant amount of computational power. IceProd was developed within the IceCube collaboration as a tool for managing productions across distributed systems. This Python-based distributed system consists of a central database and a set of daemons that are responsible for various roles on submission and management of grid jobs as well as data handling. IceProd makes use of existing grid technology and network protocols in order to coordinate and administer production of simulations and processing of data. The details of job submission and management in different grid environments is abstracted through the use of plug-ins. Security and data integrity are concerns in any software architecture that depends heavily on communication through the Internet. IceProd includes features aimed at minimizing security and data corruption risks. IceProd is undergoing active development with the aim of improving including security, data integrity, scalability and throughput with the intent to make it generally available for the scientific community in the near future.

## 11. Acknowledgements

We acknowledge the support from the following agencies: U.S. National Science Foundation-Office of Polar Programs, U.S. National Science Foundation-Physics Division, University of Wisconsin Alumni Research Foundation, the Grid Laboratory Of Wisconsin (GLOW) grid infrastructure at the University of Wisconsin - Madison, the Open Science Grid (OSG) grid infrastructure; U.S. Department of Energy, and National Energy Research Scientific Computing Center, the Louisiana Optical Network Initiative (LONI) grid computing resources; National Science and Engineering Research Council of Canada; Swedish Research Council, Swedish Polar Research Secretariat, Swedish National Infrastructure for Computing (SNIC), and Knut and Alice Wallenberg Foundation, Sweden; German Ministry for Education and Research (BMBF), Deutsche Forschungsgemeinschaft (DFG), Research Department of Plasmas with Complex Interactions (Bochum), Germany; Fund for Scientific Research (FNRS-FWO), FWO Odysseus programme, Flanders Institute to encourage scientific and technological research in industry (IWT), Belgian Federal Science Policy Office (Belspo); University of Oxford, United Kingdom; Marsden Fund, New Zealand; Japan Society for Promotion of Science (JSPS); the Swiss National Science Foundation (SNSF), Switzerland; A. Groß acknowledges support by the EU Marie Curie OIF Program; J. P. Rodrigues acknowledges support by the Capes Foundation, Ministry of

Education of Brazil.

## 12. References

[1] F. Halzen *IceCube: A Kilometer-Scale Neutrino Observatory at the South Pole* IAU XXV General Assembley, Sydney, Australia, 13-26 July 2003, ASP Conference Series, Vol. 13, 2003

[2] O. Schulz *The IceCube DeepCore*, 4th International Meeting on High Energy Gamma-Ray Astronomy, Heidelberg, Germany, 7-11 July 2008 p. 783-786

[3] Francis Halzen and Spencer R. Klein *Invited Review Article: IceCube: An instrument for neutrino astronomy.* Review of Scientific Instruments, AIP, August 2010, Vol 81, 081101

[4] De Young, T R *IceTray: a Software Framework for IceCube*, Computing in High Energy Physics and Nuclear Physics 2004, Interlaken, Switzerland, 27 Sep - 1 Oct 2004, p. 463

[5] A. Pavlo *et al. The NMI build & test laboratory: continuous integration framework for distributed computing software,* Proceedings of the 20th conference on Large Installation System Administration, Washington, DC 200 p. 21-21

[6] Dave Winer. (June 15, 1999) XML-RPC Specification UserLand Software, Inc.

[7] CodeIgniter User Guide, http://codeigniter.com/

[8] W Allcock, *et al*, *GridFTP: Protocol Extensions to FTP for the Grid*, April 2003, http://www.ggf.org/documents/GWD-R/GFD-R.020.pdf

[9] Globus Alliance, *Overview of the Grid Security Infrastructure,* http://www.globus.org/security/overview.html


# INFRASTRUCTURE

# Real-Time Communication Protocol for Supercomputing Ecosystems

Carlos Alberto Franco Reboreda, Luis Alberto Gutiérrez Díaz de León
*University of Guadalajara - CUCEA*
*carlos.franco@cucea.udg.mx, luis.gutierrez@redudg.udg.mx*

## Abstract

*Supercomputing ecosystems are typically integrated by end-user communities, high-tech development communities, high performance computing infrastructure and other support systems that perform different activities or tasks. These ecosystems can be found and interact in local or distributed environments.*

*For certain type of applications real-time communication within the ecosystem is a critical factor that must be guaranteed. One of the main challenges in the design of real-time communication systems is the definition of the required scheme to perform all system tasks in the ecosystem so all time constraints are met.*

*In this work is presented a real-time communications scheme for a distributed ecosystem. It is presented a communication protocol based on arbitrated message contention according to priority, given by a communications master plan.*

*The protocol considers periodic and aperiodic message delivery with static schedule and is based on common characteristics found in hard real-time systems (HRTS), includes a closed task set with time constraints, where critical tasks are defined as periodic tasks and it takes advantage of the broadcast nature of most networks found in real-time distributed systems. This work includes also a simulation scheme.*

***Keywords:*** *supercomputing ecosystems, distributed real-time communication systems, arbitrated message contention, message scheduling.*

## 1. Introduction

Supercomputers not only allow people to address the biggest and most complex problems, they also allow people to solve problems faster, even those that could fit on servers or clusters of PCs.

Supercomputing is not only about technologies, metrics and economics. It is also about the people, organizations, and institutions that are key to the further progress of these technologies. It is about the complex web that connects people, organizations, products, and technologies.

A supercomputing ecosystem (figure 1) is a continuum of computing platforms, system software, and the people who know how to exploit them to solve supercomputing applications:

stockpile stewardship,
* intelligence/defense,
* climate prediction,
* plasma physics,

* transportation,
* bioinformatics and
* computational biology,
* societal health and safety,
* earthquakes prediction and impact,
* geophysical exploration and geoscience,
* astrophysics,
* materials science and
* computational nanotechnology,
* human/organizational systems studies,
* among others.

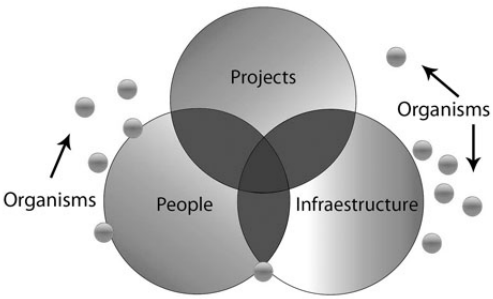



**Figure 1. Typical supercomputing ecosystem**

Organisms are the technologies that mutually reinforce one another and are mutually interdependent, such as: vector architectures, vectorizing compilers and applications tuned for the use of vector hardware; shared memory architectures, scalable operating systems, OpenMP-compliant compilers, runtime systems and applications that can take advantage of shared memory; and message passing architectures, parallel software tools and libraries, and applications that are designed to use this programming model.

The paper is organized as follows: In section 2 problems related to communication delays are presented. Section3 and 4 describe the proposed communication protocol; section 5 presents a formal description of the protocol; section 6 shows the results of two different approaches for the proposed protocol. Finally the results and conclusions are presented.

## 2. Problem Definition

The characteristics of the processors and the interconnection network (latency and bandwidth of access to memories, local and remote) within a supercomputing ecosystem are key features and determine to a large extent the algorithms and classes of applications that will execute efficiently on a given environment.

Broadcast networks are present in almost all today's network environments including supercomputing ecosystems, and bus topology [1] is widely used because of its low cost and ease of administration.

In the literature [7][9][11] are discussed several factors that contribute on the delay of message delivery in the communication process: queuing, packeting, switching and propagation. These factors are present in different stages of message transmission. There are in particular, six delay moments that are presented in the different OSI model layers. (See figure 2)

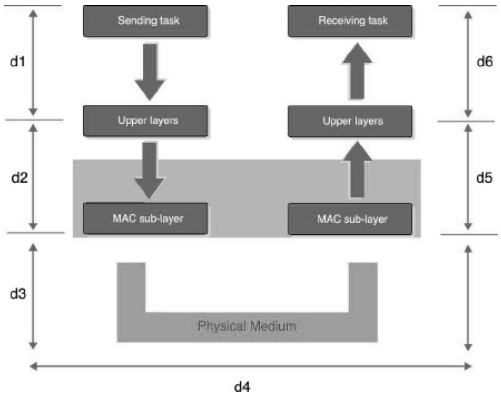



**Figure 2. Communication delays between applications in the OSI model**

Delays d1 and d6 are presented when communication process is taking place in the upper layers of OSI model (layers 3 to 7). This is when the sending task and the receiving tasks are executed. Delays d2 and d5 are generated within layer 2 and generally are due to physical medium access control.

However, d5 is significantly shorter than d2 because in the receiving task, there is no contention actually for physical medium.

Delay d3 emerges when data is put into the physical medium and it is considered a queuing delay. Delay d4 is due message propagating in physical medium. As we can see, d2 is the hardest delay to deal with (it occurs in the medium access MAC sub-layer) because it is required to develop admission control mechanisms and packet scheduling schemes.

Additionally, some techniques and communication models allow to shape traffic and to evaluate quality of service (QoS) requirements [5] for a particular application.

As it is common in several hard real-time systems, execution plan is known in advance. In centralized or single-processor environments, the execution depends on a single entity -a dispatcher or a network referee- that defines which task is executed next.

In other environments, such as Profibus [2], the execution control is distributed, where a token grants access to the network to its possessor. Both approaches have their own advantages but also disadvantages [6].

On one side, for the approach that uses a token, real-time execution can only be guaranteed to the node that holds token. On the other hand, the FIP [13] approach, based in a bus referee turns out to be a very rigid scheme. CAN [3] is also based in the node priority, not in the priority of the task, situation that can lead to a problem in the real-time execution of the system. In this paper, an execution scheme based on distribution of control in the whole network is presented.

## 3. General description of the communication scheme

It is assumed that there exist a closed number of participant nodes and there is already a master execution plan, where all tasks have been assigned to their corresponding processing entity. This plan is feasible and all deadlines are met. Each node has an instance of the global execution plan and it is assumed that each node has available all resources required to perform all the assigned tasks.

Assigned tasks to each node are not necessarily communication tasks in all cases, so it is possible to schedule tasks that do not require delivery of messages. It is very important to identify communication tasks from the others.

Through broadcast we can guarantee a minimum synchronization level between all participant nodes, because each sent message would be known and listened by the entire network. Global plan allows each node to know when a message is going to be sent, the order of sent messages and therefore, identify when and which message send each time.

Regarding to the transmitting node, when it sends a message, this is received by all network members (broadcast) but only the node that the message is directed to will process it. In that moment, as the received message has the address of the next node allowed for transmitting, it is assured an accurate synchronization in messages delivery.

The authorized node sends its message according to the described procedure, which is repeated until the execution plan is completed. Figure 2 presents a general view of the communications network.

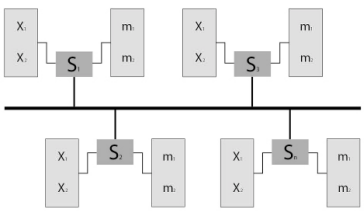


**Figure 3. General view of the communications network**

In the figure 3 can be seen: the instance of the global plan, represented by the set X, the set of participant nodes $S = \{s_1, s_2, ...s_n\}$ and the schedulable set of messages assigned to each node, denoted by $M_{(S_i)} = \{m_1, m_2, ...m_n\}$. The proposed communications scheme allows delivery of periodic and aperiodic messages [8] with static schedule [10].

## 4. Protocol Description

### 4.1 Network topology

It is considered a bus network topology, where medium access is controlled with the implementation of a "logical ring" within the network to avoid collisions. The ring will work with a circulating token that will become the transmitting permission, so each node will send all required messages. Only one node will have the token at a time, so only one node can transmit at the same time. With this strategy, collisions are avoided and it is guaranteed that messages are delivered according to the schedule previously established.

### 4.2 Medium access control

Access to medium is serialized because the message delivery is developed according to global plan. When network is initializing, one node will randomly be designated as the master node and will verify the following:

- All nodes must have been assigned with all necessary tasks for the operation of the system

• Identify the sequence of messages to be delivered among all nodes
• Circulate a start message for the network initialization, in such a way that all nodes have an instance of the schedule, the messages and tasks assignment within the system
• Create the initial token for the network

Once the initialization token is circulated, each and every node will know which tasks must perform, what messages is going to receive, what messages are required to be sent and at what time these actions are going to take place. Therefore, the network will have a pre-established token circulation, and there will be guarantees that system feasibility can be accomplished.At this point, all nodes have a copy of the schedule that is going to be performed and they must identify if there is aperiodic traffic to transmit.

Then, the master node sends the first token to initiate the normal execution of the system. This token will consider that in this moment, the highest priority for sending aperiodic traffic is for the master node. This is only for the initialization of the network operation.

### 4.3 Periodic and Aperiodic Traffic

There exist a set of periodic messages, a set of aperiodic messages (represented both by communication tasks) and other tasks that are not communication tasks. Periodic messages have very strict time constraints because they represent critical communication messages (real-time) whereas aperiodic messages have more relaxed time constraints because they do not represent critical communication tasks.

All messages have fixed length and the same structure: message ID, data field and next station to transmit node ID (token).

### 4.4 Message Scheduling

Periodic messages will be sent according to system's global plan and aperiodic messages will be locally sorted in every node according to its deadline. A message with closest deadline will have higher priority compared to a message with a later deadline.

With this, messages will be delivered in the right order and system requirements will be satisfied. Provided the fact that broadcast based technology [4] (Ethernet, for example) is well known in their data propagation times and that now it already has a medium access control mechanism that avoids collisions, it is possible to accurately calculate if the delivery of a set of messages is schedulable in the network according to its deadline.

## 5. Formalization and evaluation of the protocol

The proposed protocol requires certain initial conditions for its operation:

C1: It is assumed a closed set of sites or participant nodes $S$

C2: Each node is assigned a set $M$ of nodes and tasks that is schedulable

C3: Each node has a copy of the global message schedule, known as execution plan $X$.

C4: Messages have fixed length and a three-field structure: message identifier, data field, and next transmitting station or node identifier.

C5: Message transmission time $\delta$ is negligible.

C6: All nodes receive the same message at the same time, including the origin node.

Let $S = \{s_1, s_2, ...s_n\}$ be the set of sites or participant nodes, $M_{(S_i)} = \{m_1, m_2, ...m_n\}$ the set of schedulable messages assigned to node $S_i$, such as $\forall m_j \in M$, we have that $m_j = (i, data, token)$, where $i$ is the index or identifier of the message in the execution plan $X$, $data$ represents the information that is going to be transmitted and $token$ represents the permission for the next node can actually transmit a message.

$X = \{x_1, x_2, ...x_n\}$ is the set of messages that conform the execution plan, where $\forall x_i \in X$, we have that $x_i = (s_{origen}, m_j, s_{destino})$ where $s_{origen}$ is the node that sends the message $m_j$ to node $s_{destino}$.

We have that $\forall s_n \in S, \exists p_n$, such as $p_n$ is a local administrator that manages communication activities in each node and is responsible to send, receive and deliver a message $m_j$ from node $S_{origen}$ to node $S_{destino}$ by using the functions $create_i(m_j)$, $send_i(m_j)$, $receive_i(m_j)$ and $deliver_i(m_j)$.

Functions $create_i(m_j)$ and $deliver_i(m_j)$ are higher level functions than sending or receiving the message $m_j$ at the nodes $s_{origen}$ or $s_{destino}$. Reception of a message does not imply the immediate delivery of the message because delivery is conditioned by timing parameters.

$send_i(m_i)$ is the function that assures the sending of the message $m_j$ from node $s_{origen}$ to node $S_{destino}$.

$receive_i(m_j)$ is the function that assures the reception of the message $m_j$ from transmission medium to node $S_{destino}$.

$create_i(m_j)$ is the function that communicates to the application level to generate messages that eventually are going to be send to the communication medium.

$deliver_i(m_j)$ is the function that assures the delivery of the message $m_j$ to the node $S_{destino}$ This is shown in figure 4.

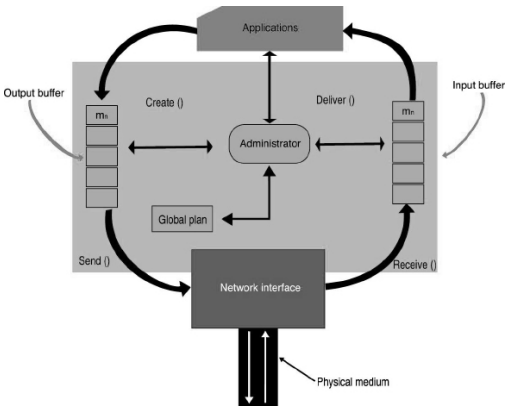



**Figure 4. Architecture of a communication node within the network**

It is necessary to perform a simulation process in order to verify the average arrival time of messages in several executions of the global plan and also to verify that there is no contention for the physical medium.

Next, are presented the considerations taken into account for the simulation:

1. It is assumed that there is a closed set of participant nodes

2. There exists a previous fixed schedule for periodic messages

3. It is assumed that the periodic messages set that is going to be send is feasible

4. Every node that comprises the network has a copy of the execution plan (to avoid medium access contention) and therefore, knows which are the messages to send.

5. Nodes do not require an explicit synchronization, because having a copy of the global schedule is an intrinsic mechanism of synchronization.

6. Messages are sent according to the sequence established in the execution plan.

7. In each message, a token is transmitted. If the token is released, the next site or node in the list can send its message.

8. The message has a three field structure: Message identifier, data, token.

The token acts as an identifier for the next node to transmit.

Based on the simulation previously described, it is expected:

1. Know the average arrival time of messages in every node

2. Verify that no exists contention for the medium.

For the simulation of aperiodic tasks there are included two proposals, the first (proposal 1) consists in applying the criteria that the node that has the token, is the node that send the aperiodic traffic and only in case this node has no aperiodic messages to send, the token goes to the next node according to global plan.

The second (proposal 2) applies the criteria of circulating the token (to send aperiodic traffic) in a predefined order (which is defined for example at the moment of the network initialization). It is convenient to remark that aperiodic traffic will be sent only after sending all periodic traffic (real-time traffic) trying to take advantage of network idle times.

## 6. Results

For the simulation, it was performed a work that consisted in the evaluation and the comparison of a couple of simulation scenarios described in the previous section. In order to perform those comparisons it was required to design a set of scenarios in a network simulation software tool that could allow creating the network environment to simulate. This implied the development of the following activities:

1. Determine the number of processors in the network
2. Generate a set of schedulable tasks for each node in the network, with the processing load in each task chosen for each node.

3. Determine the load of periodic messages present in each node.

4. Generate a global schedulable plan that includes the established communication tasks

5. Determine aperiodic messages load present in each node

6.Simulate the behavior of the communications network

7. Get the results

8. Interpret the results

9. Conclusions

### 6.1 Number of processors

Simulation environments were developed with 5 and 10 processors. In this work are only presented simulation results with 5 processors because results obtained with 10 processors are very similar in both cases.

### 6.2 Generation of a set of schedulable tasks.

There were created random sets of tasks for each of the network nodes, which are completely schedulable in each one of the processors. The simulation presents two cases, when the processor load is 60% and when it is 80%. This means that in each node we had a maximum load of tasks for each processor equivalent to 60% or 80% and that set of tasks was schedulable.

### 6.3 Determination of the periodic load in each participant node

From the total number of tasks assigned to each node, not all of them are communication tasks, that is, some tasks that do not require communicating with other nodes and do not require information exchange to perform correctly. This means that in the simulation scenarios it is required to define from the total number of tasks or messages assigned to each node, how many messages are periodic. For this simulation, it was considered that 10% of the tasks assigned to each node are communication tasks.

### 6.4 Generation of an execution plan in each node: Global plan

As each set of tasks is schedulable in each node, now it is required that all sets of tasks are schedulable when they combine. This is, if the sets of tasks are schedulable in the local environments, there is no guarantee that all local plans can be schedulable when it comes about a global schedule. The simulation scenario then creates local plans that are also schedulable in the global level. The simulation performed did not detect any problem regarding to collisions or missed deadlines in the system messages or tasks.

### 6.5 Aperiodic messages load

From the total of communication tasks present in each node, most are periodic tasks, but there were considered different levels of aperiodic tasks load, to analyze the

behavior of the network to these variations.

For the task processing load of 60%, it was simulated a message load of 10% and an aperiodic messages load of 5%, 10%, 15% 20% and 25%.

For the task processing load of 80%, it was simulated a message load of 10% and an aperiodic messages load of 5%, 10%, 15% 20% and 25%.

### 6.6 Simulate behavior of the communications network

To simulate the behavior of the communications network, it is required to clear up that it was considered a 2 time units delay for the minimal transmission and propagation time (communication time between two nodes, the closest) and a 5 time units delay for the maximum transmission and propagation time (communication time between two nodes, the farthest). Tasks will have established periods between 100, 50 and 40 units. The execution units, the period and deadlines of each task or message are generated randomly. Precedence relationships are established randomly and after processes have been generated for each processor.

### 6.7 Obtained results

Obtained results have to do with the communications network behavior and consist in the following:

- Response time for periodic messages
- Response time for aperiodic tasks

Results are shown in tables 1 and 2.

| **Task load:** 60% of processor capacity | | | | | |
|---|---|---|---|---|---|
| **Periodic messages load:** 10% | | | | | |
| Aperiodic message load | **5%** | **10%** | **15%** | **20%** | **25%** |
| Proposal 1 | 2.96 | 3.1 | 3.52 | 3.67 | 3.98 |
| Proposal 2 | 2.8 | 2.96 | 3.41 | 3.56 | 3.78 |

**Table 1. Simulation results for 60% of tasks load**

| **Task load:** 80% of processor capacity | | | | | |
|---|---|---|---|---|---|
| **Periodic messages load:** 10% | | | | | |
| Aperiodic message load | **5%** | **10%** | **15%** | **20%** | **25%** |
| Proposal 1 | 2.56 | 3.01 | 3.38 | 3.71 | 3.95 |
| Proposal 2 | 2.52 | 2.91 | 3.31 | 3.66 | 3.92 |

**Table 2. Simulation results for 80% of tasks load**

The response time for periodic messages is constant, because its attention is not in risk because it is assigned in fixed intervals. Results were obtained in the format in what the simulation tool delivers them, however it was necessary to treat them so they could be interpreted and plotted.

### 6.8 Interpretation of results

As it can be seen in tables 1 and 2, there is a slight variation in the response times for the delivery of aperiodic traffic when the load percentage of processor capacity is increased in each node. This can also be noticed in figures 5 and 6.

In figure 5, we can appreciate the

behavior of the average propagation and transmission times when tasks load is 60% of the processors capacity. In figure 6, we can appreciate the behavior of the average propagation and transmission times when tasks load is 80% of the processors capacity. In both cases are considered the two proposals described previously for dealing with aperiodic traffic.

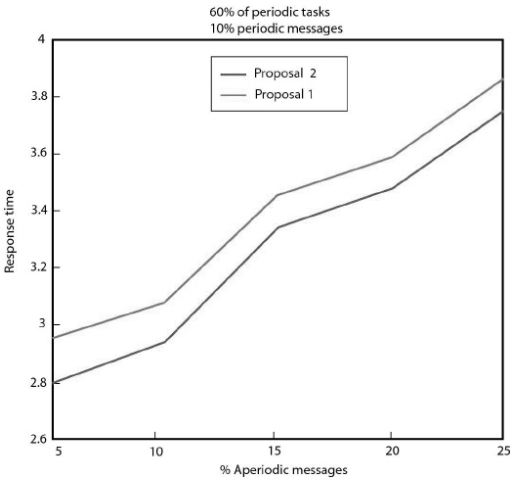



**Figure 5. Comparing response time for aperiodic messages with 60% of processor load**

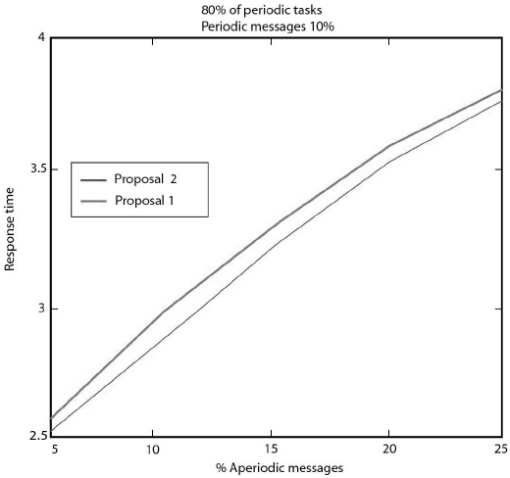



**Figure 6. Comparing response time for aperiodic messages with 80% of processor load**

## 6.9 Conclusions

According to behavior presented in graphics and tables, we can conclude that the first proposal offers better response times. However, as the processor load is increasing, the second proposal tends to a more stable behavior and to offer similar response times, whereas the firs proposal as the processor load is increasing tends to increase also its response times. This behavior is consistent with the idea that the first scheme is unfair under the perspective of how many nodes can send aperiodic traffic. That is the reason why it has better performance when there is not much load in the processor. Nevertheless, the second scheme, as it is more balanced (fair) it tends to improve the performance

when processor load increases.

The first proposal is more efficient for cases when processor load is not so demanding and in cases where aperiodic messages is kept within the 2%, referred in the literature by Stankovic [12]. The second proposal is better for cases where aperiodic messages load is above average. Even this scenario is hard to find in practical situations, it is interesting for analysis. According to these results, it could be evaluated a third proposal that have a medium between both presented proposals regarding the amount of aperiodic messages it can handle and possibly it can be obtained a better scenario between response times and stability.

## 7. References

[1] Weaver and C. Summers, "The IEEE Token Bus-A Performance Bound on GM MAP" IEEE transactions on Industrial Electronics, Volume 35, Issue 1, feb 1988.

[2] E. Tovar and F. Vasques. "Real-time fieldbus communications using Profibus networks". IEEE Transactions on Industrial Electronics. Volume 46, Issue 6, Dec 1999 Page(s): 1241 – 1251.

[3] H. Kaschel, E. Pinto. "Análisis protocolar del bus de campo CAN" Reporte de trabajo. Facultad de Ingeniería, Depto. de Ingeniería Eléctrica. Universidad de Santiago de Chile. Chile, 2002.

[4] G. LeLann and N. Rivierre, "Real-Time communications over broadcast networks: The CSMA-DCR and the DOD-CSMA-CD Protocols". INRIA Report RR1863, 1993.

[5] P.Ferguson and G. Huston. "Quality of Service: delivering QoS on the Internet and in corporate networks". John Wiley & Sons press. USA, 2000.

[6] C. Aras, J. Kurose, D. Reeves and H. Schulzrinne, "Real-Time Communication in Packet-Switched Networks", Proceedings of the IEEE, Vol. 82, No. 1, pp. 122-139. Enero, 1994.

[7] Cottet, Francis, Joêlle Delacroix, Claude Kaiser, & Mammeri, Zoubir, "Scheduling in real-time systems". England: John Wiley and sons press. USA, 2002.

[8] Y. Atif and B. Hamidzadeh, "A Scalable Scheduling Algorithm for Real-Time Distributed Systems", Proceedings of the 18th International Conference on Distributed Computing Systems, May 26-29 1998, pp. 352-359

[9] D. Verma, H. Zhang and D. Ferrari. "Delay jitter control for Real-Time communication in a packet switching network". In proceedings of TriComm. 1991.

[10] Buttazzo, Giorgio C. "Hard real-time computing systems: predictable scheduling algorithms and applications". Kluwer Academic Publisher. Boston, 1997.

[11] Cheng, Albert M.K. "Real-time systems: Scheduling, analysis, and verification". John Wiley and sons press. New Jersey, 2002.

[12] Stankovic, John A., Spuri, Marco, Ramamritham, Krithi, & Buttazo, Giorgio. "Deadline scheduling for real-time systems: EDF and related algorithms". Kluwer Academic Publisher Boston, 1998.

[13] M. León. "Calidad de Servicio en la Red Industrial WorldFIP". Reporte de Investigación. Facultad de Ciencias de la Computación. Benemérita Universidad Autónoma de Puebla. México, 2002.

# Construction and Design of a Virtualized HPC Cluster Type and Comparison With a Physical Mosix Cluster

Juan Alberto Antonio Velázquez, Juan Carlos Herrera Lozada, Leopoldo Gil Antonio,
Blanca Estela Núñez Hernández, Erika López González.
Tecnológico de Estudios Superiores de Jocotitlán, Centro de Innovación y
Desarrollo Tecnológico en Cómputo IPN, Depto. Sistemas.
*{jantoniov0801, jlozada}@ipn.mx, lgilant72@yahoo.com.mx,*
*blancahdez17@yahoo.com.mx,e_nada@yahoo.com.*

## Abstract

*Virtualization has an important role in computer security, and most companies use it to protect the information using virtual systems that communicate with physical systems. Companies also see advantages as saving space and lower energy use which in turn is included in saving a lot of money. This paper defines the parallelization technique for load balancing and process migration, which can be done by kernel-level software such as MOSIX.*

*You can define virtualization techniques used to handle different operating systems. Virtualizer software used was Virtualbox can be installed on different platforms and accept the installation of several operating systems including to Opensuse.*

***Keywords:*** *Virtualization, Mosix, clusters, Virtualbox.*

## 1. Introduction

Cluster is defined as a set of networked computers, generally used for the solution of tasks on a single computer can not do. The cluster type HPC (High performance cluster) used to solve problems of scientific nature. An alternative to building an HPC cluster type [2], using free software as in the case of the Linux operating system and kernel-level middleware such as Mosix. Mosix takes over certain functions at the level of operating system kernel, but its main function is to balance the load of the connected nodes, and this makes migrating micro-level processes. On the other hand offers virtualization as a tool that can simulate physical teams in a virtual cluster, causing it to reduce cost, space, energy expenditure, and can be added security. In this paper an analysis of work performance of a cluster of physical and virtual operation of a cluster [4].

## 2. Basics

HPC clusters have been implemented with kernel-level software, such as Mosix, at the University of Jerusalem have conducted research in bioinformatics clusters [1], this cluster occupies a large amount of performing calculations and rendering images of DNA and chemical bonds. Today's computer systems have used virtualization to solve problems. And they are using these techniques for implementation in the clusters. There is little information of virtual clusters, especially working with Mosix. At the University of Barcelona has been working with Cloud Grid using Xen virtualized systems and system for creating and Open Cloud Nebula [3]. In regard to Mosix and work in virtual clusters, the creators of Mosix demonstrate that through virtual machines to create a grid interconnection virtual clusters, each working on one computer and managed by two Workstation [4].

## 3. Cluster Virtual

Like the physical cluster virtual cluster creation involves having 4 nodes connected via virtual hardware, including virtual network cards, processors, hard drives, amount of memory, USB, etc..

In Virtual Box you can choose the brand of network card and also the kind of inner connection to communicate with each other hosts.

Some brands handled by the Virtual Box virtual machine [5] are: Intel PRO/1000 MT Desktop PC net-Fast III, among others.

Here there is a central device called a switch, but internally the virtualized connection takes the place of the physical network, but no devices or cables involved.

In turn, hard disks and virtual memory play an important part before installing an operating system, because these devices can take the size of storage required for the operating system functions. For example, with regard to the RAM can take different memory sizes that range from 4MB up to 2048 MB hard drives can be configured in the virtual machine and its capacity can be modified by the Administrator of the virtual system, from 4 Mb to 2 terabytes of space.

Already having nodes configured with the network type, choose the operating system is chosen openSUSE 11.1 and the virtual hard disk, which had previously been set size. The installation and configuration of openSUSE operating system, you can make a disk mounted on the physical drive of the computer or using a disk image with extension *. nrg or *. iso. The expected result is to make virtualization a considerable saving of space and temperature.

## 4. Differences between virtual and physical cluster

The differences between the virtual cluster and the cluster are remarkable physical eye, but in terms of processor speeds, as the hard disk capacity and RAM depend on the physical machine on your platform, which can be 32 and 64 bits. The virtual machine depends on these platforms to operate. The advantage of the platforms is that each virtual machine can be attached

to their physical characteristics, making a 32-bit virtual machine can operate on a 64-bit physical machine.

The clusters have physical devices that are physically installed on a motherboard and its features impose a processing speed and RAM memory access, in addition to the hard disk can store the required information at a speed read / writing depends on the motherboard. All of these hardware devices work together with the microprocessor, thus working at the speed of processing; all devices are compatible with the processor and the motherboard if it is a 64-bit.

## 5. Methodology

This work was based on documentary research, establishing the theoretical basis of the problem, researching and documenting the virtualization techniques and different types of clusters with parallel programming to solve problems.

The virtualizer used in the project is Virtual Box and created 4 virtual machines called nodo0, node1, node2 and Node3; configuring the internal network for communication between them, plus storage and remote access. Similarly Mosix was installed for the connection between nodes in the cluster.

We also performed the installation and configuration of a physical cluster for testing performance between it and the virtual cluster. The cluster has four physical computing devices connected to physical media via a switch sharing resources. These computer equipment made available, have a Pentium 4 processor at 1.5 GHz with a 80 GB hard disk storage. Each computer equipment with network cards that are installed on the PCI bus. One of the cards at a speed of 10/100 Mbps and the other card works at a speed of 1 Gbps and is the one in charge for the transmission of information.

One of the four nodes (see Figure 1) was chosen to be the master node. This works for the master node communication with other nodes with faster network cards. The network card uses, one is 10/100 Mbps and connects to the Internet in the institution's network and other network card is 10/100/1000 Mbps which is connected to the slave nodes; slave nodes are connected via 10/100/1000 Mbps card, which makes the network between nodes is efficient and fast.

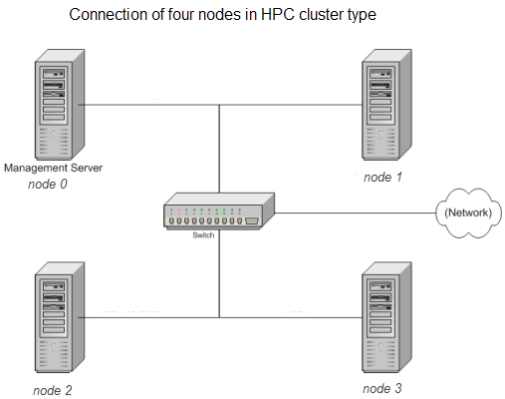



**Figure 1. Schematic of the cluster with 3 nodes and a master node**

The media is Category 5e UTP cable, which connects the nodes to switch. It has chosen the Linux operating system [6], their distribution openSUSE version 11.1 Mosix

because the patch is compatible with this distribution. This installation was done for each of the nodes, which are configured within the network, assigning network addresses. The virtual cluster is configured the same way that the physical cluster, but the difference between it and the physical system, which depends on the hardware is virtual in that account within the virtual machine, which is Virtual Box version 3.0 .2 of Sun Microsystems. As in the physical nodes in the virtual machine is installed and configured four virtual nodes with the Linux operating system on your openSUSE 11.1 [7], set to an internal network in which nodes are capable of interfacing via a network to share resources software and hardware. The physical cluster 4 computers have been implemented with similar characteristics and these are:

1. - Four interconnected nodes with a link to an Enterasys switch at 1000Mbps.

2. - Celeron Core Duo, 2 GB of RAM, each with a 160 GB hard drive

3. - Linux Operating System version 11.1 of openSUSE Novel.

4. - The master node (nodo0), has 2 NICs eth0 and eth1 where eth1 has an exit to the internet via the IP number 148.204.67.192. This node includes the Apache2 web server services and OpenSSH to access the cluster from the Internet.

5. - The middleware connection MOSIX kernel level [8].

The IP address configuration is done in each of the nodes also have to configure the computer name to recognize the network, this is done from the / etc / hosts, this file is edited, you put your IP address and the names of the nodes as will be recognized on the network.

Example:

192.168.10.1 nodo0
192.168.10.2 nodo1
192.168.10.3 nodo2
192.168.10.4 nodo3

Mosix To install, you must have the installation files and settings, which are obtained from the page creators Mosix [8], the files required for installation must be sought depending on the OS and the version of the system operating in this we find a section for installing the kernel for all Linux distributions that are known. In the case of the distribution can be obtained OpenSUSE install RPM files. In the case of the cluster was chosen version of OpenSUSE 11.1 and files that were installed were the latest Mosix, the 2.28.0.

Recommended files to install the kernel-Mosix-2.6.27.33_2.27.1-6.4.i586.rpm is necessary for the operation of the kernel and kernel Mosix Linux operating system and a file is required for operation file-utils-2.27.1.0 Mosix-1.i386.rpm, which contains other performance characteristics of migration Mosix [8].

The installation of these files is done automatically:

```
pc0@linux-ipvw:~/Escritorio/
kernelmosix> rpm -ihv kernel-mosix-
latest.i586.rpm
pc0@linux-ipvw:~/Escritorio/
```

kernelmosix> rpm -ihv mosix-utils-latest. i386.rpm.

When you install this installation Mosix basic properties connecting Mosix kernels, but not entirely appropriate for the Linux kernel.For Mosix operation must be restarted for the Grub bootloader, the kernel Mosix recognize as the first boot system.

## 6. Results

To develop the evidence was necessary to test and load balancing in each of the networked nodes and would be executed by Mosix. For this, algorithms were sought by their mathematical complexity respond to a time delay, these algorithms are algorithms factorial algorithm with 3 nested loops (Stress) and the algorithm that calculates Pi by the method of Leibniz. The tests were conducted in each of the algorithms were based on samples in numerical average of ten runs per sample yielded different times in each of the teams who were executed as a computer (sequential), the cluster physical and virtual cluster. In addition to its analysis of speedup to see the gain in time compared between sequential time and physical and virtual clusters.

The speedup is known as acceleration is a measure of performance between a multiprocessor and single processor system. In other words, the speedup factor measures how effectively proves the parallelization of a program, compared to its sequential version. This requires measuring the execution time of each version and they estimate the relationship between time and speedup, defined as:

Tsec= execution time of sequential program.
Tpar= execution time of parallel program.

Evidence Stress algorithm with 3 nested loops.

Stress for the algorithm, the samples were taken into account (Table 1) to develop the algorithm were according to the scope of the types of data that the C language can achieve. The computational complexity of the factorial algorithm is O (n3), and because of this complexity and execution is a delay to resolve the various samples.

```
#include <stdio.h>
int main( )
{
int i,j,k;
int n=100;
printf("Calculando el tiempo en el algoritmo de stress de un numero %d:",n);
for (i=0; i<=n; i--)
{
for (j=0; j<=i; j++)
{
 for(k=0; k<=j; k++)
{
printf("%i",k);
}
}
return 0;
}
```

3 for nested, resulting in a computational complexity of n3, causing a slowdown

| MUESTRAS, Y COMPARACION SECUENCIAL, CLUSTER FISICO, VIRTUAL Y SPEEDUP EN ALGORITMO STRESS | | | | | |
|---|---|---|---|---|---|
| MUESTRAS | SECUENCIAL | CLUSTER FISICO | CLUSTER VIRTUAL | SPEEDUP FISICO | SPEEDUP VIRTUAL |
| 25 | 0.004 | 0.001 | 0.001 | 4.000 | 4.000 |
| 31 | 0.015 | 0.001 | 0.002 | 15.000 | 7.500 |
| 39 | 0.038 | 0.002 | 0.002 | 19.000 | 19.000 |
| 46 | 0.078 | 0.002 | 0.003 | 39.000 | 26.000 |
| 60 | 0.082 | 0.003 | 0.003 | 27.333 | 27.333 |
| 52 | 0.099 | 0.005 | 0.007 | 19.800 | 14.143 |
| 68 | 0.186 | 0.007 | 0.009 | 26.571 | 20.667 |
| 80 | 0.245 | 0.009 | 0.010 | 27.222 | 24.500 |
| 91 | 0.337 | 0.168 | 0.174 | 2.006 | 1.937 |
| 100 | 0.398 | 0.198 | 0.234 | 2.010 | 1.701 |

**Table 1 Samples of calculated times sequentially, in the physical cluster, virtual cluster algorithm speedup with stress.**

In the sequential execution of the algorithm Stress samples were used from 25 to 100, the times shown in the chart below (Figure2).

The same samples were used to run in the physical cluster and the result is the following graph, Figure 3.

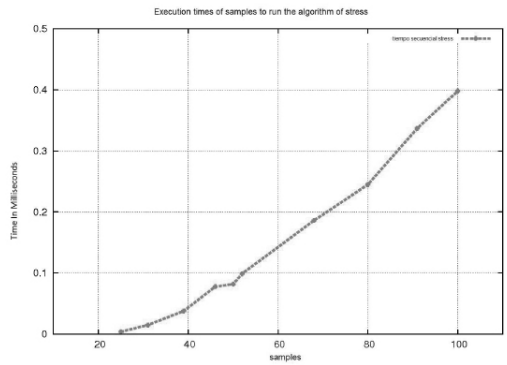



**Figure 2. Estimated time of stress algorithm sequentially**

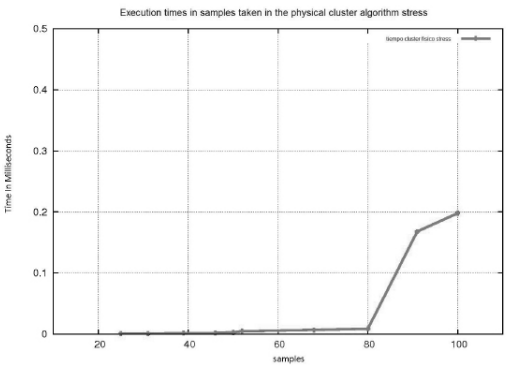



**Figure 3. Plot of time in the physical cluster.**

The behavior of the graph in virtualized cluster, Figure 4.

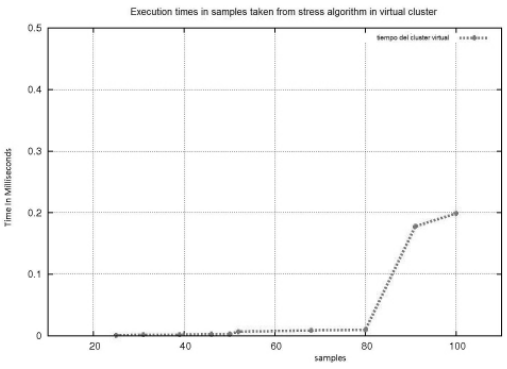



**Figure 4. Plot of behavior in the virtual cluster algorithm Stress.**

We used the speedup formula to see the difference between the cluster and a single physical computer running the algorithm sequentially stress Figure 5.

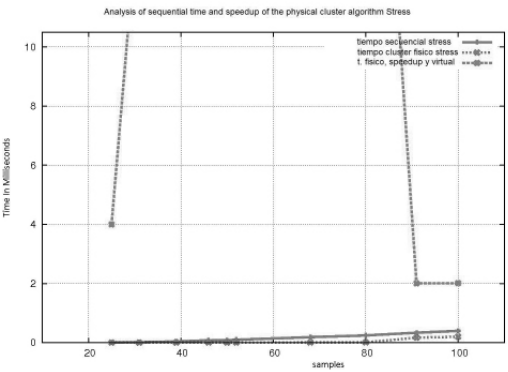



**Figure 5. Graphical comparison between sequential time, time speedup of the algorithm and physical stress.**

The stress algorithm execution behavior has a fairly high compared to the physical cluster (Fig. 6).

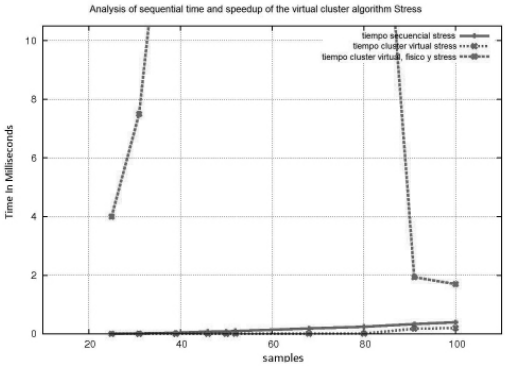



**Figure 6. Graphical comparison between sequential time and the virtual cluster speedup algorithm applied to Stress.**

*Testing Algorithm PI by the method of Gregory-Leibniz.* The PI algorithm mathematically calculated to the nearest approximation to the value of PI, which is 3.1415927, Pi algorithm developed in a programming language like C, for a data type capable enough to support a floating number that great as it is the PI value and this data type is long double. Their analysis yields a result computational complexity logarithmic O (log n) and this complexity shows that running this algorithm on a single node can have an execution delay of several minutes, so the delay is less parallelize.

```
#include <stdio.h>
#include <stdlib.h>
main( )
{
long double suma =0.0,suma1=0.0,
suma2=0.0, k, k1,k2, n=1000000;
//aquí se modifica la numeración
k=2*n;
k1=-3;
k2=-1;
while (k1<(k-3)) //primera sumatoria
{
k1=k1+4;
suma1= suma1+1/k1;
}
printf("La suma1 es %f \n",suma1);
while(k2<(k-3)) //segunda sumatoria
{
k2=k2+4;
suma2= suma2+1/k2;
}
printf("La suma2 es %f \n",suma2);
suma=4*(suma1-suma2); //sumar las 2 primera
sumatorias y multiplicarlos por 4 ya que es el
valor de ¼ deseado
}
printf("El valor de PI es de %.90Lf \n", suma);
}
```

Loops where the evolution of the control variable is downward nonlinear.

```
int c = n, O (1)
while (c> 1) O (log n)
{
    vector [c] = c; O (1)
    c = c / 2, O (1)
}
```

For this example, initially the value of c is equal to n, after iterations will be n * x 2-x Þ the number of iterations is such that n * 2-x £ n, a similar reasoning leads to log2 n iterations, Þ in this case is:
O (1) * O (log n) * O (1) = O (log n), logarithmic complexity.

The samples were chosen from a value of 1500 to 98,000,000 (Table 2), because in the algorithm while the value of n is higher, more overhead to the computer but the result is closer to the value of Pi.

| Sequential comparison of cluster samples physical, virtual and speedup of the algorithm of stress | | | | | |
|---|---|---|---|---|---|
| Samples | Sec time | Time physical | virtual time | speedup phys | speedup virtual |
| 1500 | 0.002 | 0.000 | 0.000 | 20.000 | 0.500 |
| 5800 | 0.002 | 0.001 | 0.001 | 2.000 | 1.000 |
| 7000 | 0.002 | 0.001 | 0.001 | 2.000 | 1.000 |
| 8500 | 0.004 | 0.001 | 0.003 | 4.000 | 0.333 |
| 22000 | 0.004 | 0.002 | 0.003 | 2.000 | 0.667 |
| 91000 | 0.004 | 0.003 | 0.003 | 1.333 | 1.000 |
| 147000 | 0.006 | 0.003 | 0.003 | 2.000 | 1.000 |
| 789000 | 0.013 | 0.010 | 0.011 | 1.300 | 0.909 |
| 1000000 | 0.023 | 0.012 | 0.019 | 1.917 | 0.632 |
| 5000000 | 0.100 | 0.087 | 0.923 | 1.156 | 0.094 |
| 10000000 | 0.192 | 0.161 | 0.187 | 1.191 | 0.862 |
| 98000000 | 1.921 | 1.234 | 1.673 | 1.557 | 0.737 |

**Table 2 shows, cluster calculated times sequentially in physical, virtual and clustered with the algorithm speedup IP Leibniz.**

The calculation algorithm sequentially Pi times throws us that were implemented in each of the samples, which can be seen in the graph of Figure 6.

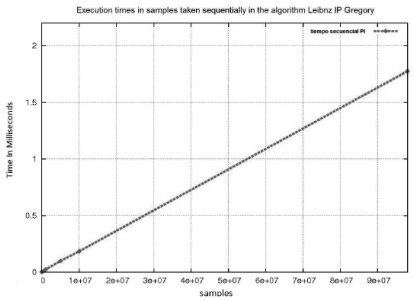



**Figure 7. Graph showing the sequential execution time of the algorithm of Pi by Leibniz.**

The execution algorithm in the physical cluster Pi denotes the improvement on the execution in a single node, this is shown in the graph of Figure 8.

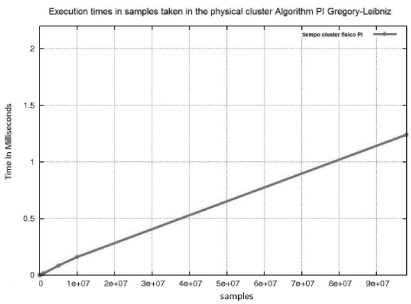



**Figure 8. Execution time on physical cluster algorithm for Pi by Leibniz.**

In the virtual cluster on the same samples were run and results in a graph similar to the physical cluster, Figure 9.

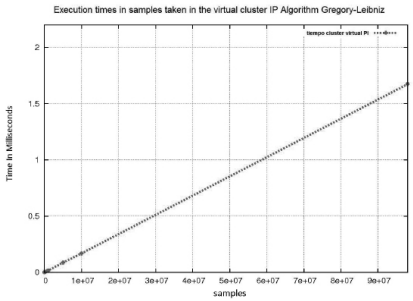



**Figure 9. Graph of time to execute the algorithm Pi by Leibniz in virtual cluster.**

The speedup formula applicable to compare the times between the physical cluster and single node denotes the acceleration and improvement obtained between the performance of the algorithm on a single node pi and physical cluster (Figure 10).

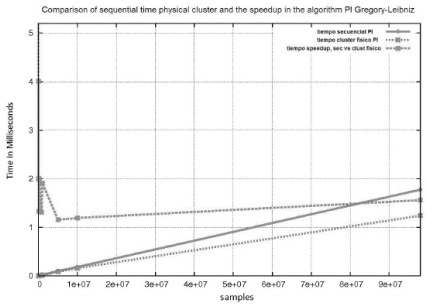



**Figure 10. Graph of time to execute the algorithm Pi by Leibniz in virtual cluster.**

Then we analyzed the acceleration between the sequential execution and enforcement of the samples in the virtual cluster, the speedup formula shows a similar result to the speedup obtained from the physical cluster and one node (Figure 11).

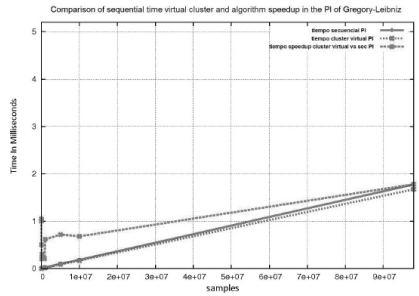



**Figure 11. Comparison of times obtained from virtual cluster and sequential execution and the result of acceleration with the PI algorithm.**

The command used to view the behavior as migrating processes between nodes and how memory management was using the command Mon (Mosix Load Monitor), which contains the following functions:

- Mosix Load Monitor Versión 1.6
- Selection keys that show.
- L shows the load
- S shows the speed
- M sample memory between nodes, and the amount of free memory between them.
- U shows the use of resources
- F shows the processes fallen.

In Fig. 12 shows as nodes 1 and 2 of the cluster shared memory according to the processes that are migrating from node to node in the cluster Mosix. In this case the command Mosix mon, used to see processors and processes that work across the cluster, but when you press the m key is shown when the memory load and the exchange of node to node at the time making process, no matter how small.


**Figure 12. Mon The command shows the exchange of memory load in the migration process.**

Mosix processes can also be viewed using the mon command actually comes in automatic writing to the command line terminal and automatically appears Mosix plotter resources and processes according to the load being undertaken Fig 13.

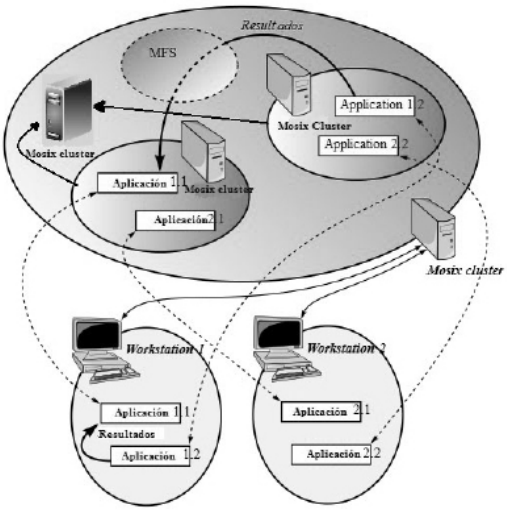



**Figure 13. The figure shows how to migrate processes between the four nodes of the cluster and how they are accessed from workstations.**

The work of the virtual cluster when running programs stress and pi Leibniz program, the execution time in the virtual cluster depends exclusively on the virtual nodes when processes migrate from node to node can be seen work in the graphic that replicates the mon command, which is listed as the primary node where you compile and run load remains the node implementation, however other nodes showing loads

processes in the algorithm execution time of stress, Figure 14.

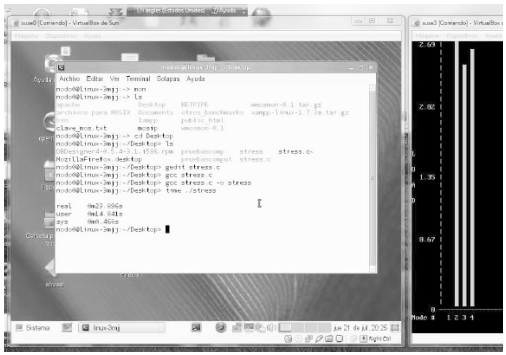


**Figure 14 The figure below shows how it affects the migration of processes to nodes in the cluster when running the algorithm of stress.**

## 7. Conclusions

You can see both the virtual cluster as in the physical cluster that need a good connection at high speed network and a good ability of RAM to work optimally. Performance tests were conducted in each cluster we favorable results as it relates to runtime. The physical cluster compared to virtual cluster, the performance obtained is slightly higher to that used in the virtual cluster because the cluster virtual machines work with a single virtual processor, which for a good performance using a memory large enough RAM for the use of virtual machines. The advantage of using virtual machines lies in saving power, cost and space reduction.

On the other hand, test performance was observed the behavior of each of the algorithms both physically executed in parallel. The algorithm will run slower algorithm was stress, which in the study of computational complexity is the result given O (n3), which shows the delay in implementation. In all cases in which performance tests were both physically and virtually, the better performing algorithm was the algorithm of PI by the method of Gregory-Leibniz.

The virtual performance in some of the executions was a time not higher than expected, because the execution was a little slower than it is in the physical cluster.

Using Mosix tells us that there is only one program that is part of an operating system, is also a program that works silently and when you are running a parallel program workload migrates from node to node looking for an available node for execution. Both the physical cluster in the virtual cluster can add more nodes, this can be done most powerful work of the cluster, but the disadvantage is that it works Mosix licenses. In this work we used a free license to connect up to 6 nodes, which makes the performance sometimes not as expected. If you want to expand to more than 6 nodes (which is recommended), you have to buy a license based on the number of nodes that you want to attach to the cluster. With this the performance of work executed in parallel is expected to be efficient and less time. working with licenses.

The development of the project where you created the virtual cluster, we conclude that meets the expectations expected, as is its construction and when compared to the physical cluster run times were optimal. Compared with other clusters that have been made with Mosix fitness, we can say that the virtual cluster performs

balancing operations and high performance in an optimal way. In regard to virtualization, we found no papers about virtual clusters, because they are still developing and the benefits applications [9].

## 8. References

[1] Lahoz-Beltrá Rafael., Bioinformática: simulación, vida artificial e inteligencia artificial., Edit. Ediciones Díaz de Santos., año 2004., 574 pp.

[2] D. Sloan, Joseph,. High performance Linux clusters with OSCAR, Rocks, openMosix, and MPI., 2a. ed. Edit. O'Reilly Media, Inc., 2005 - 350 pp.

[3] Pages Montanera Enric, Gestion sostenible de clústers de recursos virtuales,memoria de proyecto de ing, 83 págs, Universidad Autónoma de Barcelona,2009.

[4] Barak Ammnon, The Mosix Organizational Grid A white Papper,Deparment of Computer Science The Hebrew University of Jerusalem, Israel,August 2005.

[5] Becker Dirk., VirtualBox: Installation, Anwendung, Praxis., Edit. Galileo Press., año 2009., 321 pp.

[6] Bookman Charles., Building And Maintaining Linux Clusters., Edit. Sams Publishing, año 2003, 265 pp.

[7] M. Surhone Lambert, T. Timpledon Miriam, F. Marseken Susan., Opensuse: Operating System, Linux Kernel, O penSUSE Project, Novell, SUSE Linux Distributions., Edit. Betascript Publishers., año 2009., 108 pp.

[8] Barak A., The evolution of the MOSIX Multi-computer UNIX System., Edit. Hebrew University of Jerusalem. Dept. of Computer Science., Univ., 1989.

[9] S. Weygant, Peter., Clusters for high availability, 2a. ed., Edit . Hewlett-packard Profesional., 2003,345pp.

ISUM
2011
2nd INTERNATIONAL SUPERCOMPUTING CONFERNCE IN MEXICO
CONGRESO DE SUPERCÓMPUTO

# PARALLEL COMPUTING

# Three Dimensional Parallel FDTD Method Based On Multicore Processors

Abimael Rodríguez Sánchez[1], Mauro Alberto Enciso Aguilar[1], Jesús Antonio Alvarez Cedillo[2]
*Sección de Estudios de Posgrado e Investigación en la Escuela Superior de Ingeniería Mecánica y Eléctrica, [1] Maestría en Ingeniería de Telecomunicaciones IPN.*
*[2] Instituto Politécnico Nacional , Centro de Innovación y Desarrollo Tecnológico en Cómputo , Dpto. de Posgrado, Edif. del CIDETEC*
*E-mail: abima7@hotmail.com; mencisoa@ipn.mx; jaalvarez@ipn.mx*

## Abstract

*The FDTD method is an algorithm is widely used today to solve electromagnetic problems of complicated structures complex, based on Maxwell's equation and Yee algorithm. FDTD is an algorithm has been applied to analyze a variety of problems in relation with electromagnetic reflection and refraction. However this procedure demands huge computational and time resources as the problem to solve becomes more complex. Parallel computing tries this problem by means of distribute blocks of data on different cores. The purpose of this paper is shown an implementation of parallel three dimensional FDTD algorithm in order to solve indoor electromagnetic propagation with boundaries conditions. The technique used to develop parallel computation is based on OpenMP through of shared memory which is built on multi-core CPUs System constituted by high-performance PC based on linux Operative System. Meanwhile, an improved algorithm to increase the parallel efficiency is presented. Parallel three dimensional FDTD simulations are performed by this technique and the validity of the method is verified.*

***Keywords:*** FDTD, Maxwell´s equation, multicore, OPENMP, shared memory.

## 1. Introduction

The FDTD (Finite Difference Time Domain) method is one of most common techniques for electromagnetic field analysis based on Maxwell's equations and it has been applied to analyze a variety of problems relation with reflaction, reflection and other electromagnetic phenomenon. With the recent progress of computer resources, the possibility to improve this method increase. In the FDTD method, the space including the object to be analyzed is divided into a large number of meshes and the electric fields and magnetic fields are updated with time progress by using small time steps [1]. However this procedure demands huge computational and time resources as the problem to solve becomes more complex. The adoption of new techniques like the parallel computing which is one of the most

advantageous methods can alleviate to this problem.

## 2. FDTD Method

The FDTD method consists on the discretization of Maxwell's equations in time and space using central differences. The result of the finite difference equations are numerical solved: the electric field vector in a given space is solved in a given time, and the magnetic field vector in the same space is solved in the next instant of time, then the process is repeated several times depending of conditions of the problem.

In his initial work S. Kane Yee [2] replaced Maxwell's equations in differential form by a set of finite difference equations, and by proper selection of the points where the field components (E and H) are evaluated. This Set of equation is solved considering the boundary conditions, through selecting a geometric relation to the spatial sampling of the vector components of electric and magnetic fields.

### 2.1 Maxwell's equations.

Considering a region of space that has no electrical or magnetic sources, but may have materials that absorb the energy of electric or magnetic field, Maxwell's equations dependent of time in its differential form are:

Faraday's Law

$$\frac{\partial \overline{B}}{\partial t} = -\nabla \times \overline{E} - \overline{J}_m \tag{1}$$

Ampere's Law:

$$\frac{\partial \overline{D}}{\partial t} = \nabla \times \overline{H} - \overline{J}_e \tag{2}$$

Rotational Maxwell equations can be expanded into a system of three-dimensional rectangular coordinates (x, y, z):

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - \rho' H_x\right) \tag{3}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - \rho' H_y\right) \tag{4}$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - \rho' H_z\right) \tag{5}$$

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma E_x\right) \tag{6}$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma E_y\right) \tag{7}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z\right) \tag{8}$$

where

$E_x$, $E_y$, $E_z$ Cartesian components of the electric field, volts / m.

$H_x$, $H_y$, $H_z$ = Cartesian components of the magnetic field amps / m.

ε= Electric permittivity, farads / m.

σ = Electrical conductivity, siemens / m.
μ = Magnetic permeability, henrys / m
ρ= Magnetic Resistivity , ohms/m.

The system of six partial differential equations (3) to (8) form the basis of the FDTD numerical algorithm for the interaction of electromagnetic waves with general three-dimensional objects.

## 2.2 Yee algorithm

In 1966, Kane Yee introduced a set of finite-difference equations for the rotational system of Maxwell equations, in the case of a medium without loss r = 0 s = 0. The essential part of Yee's algorithm are:

1. Solve the electric and magnetic fields in time and space using the coupled Maxwell's equations instead of solving only the electric field (or just the magnetic field) with a wave equation.

2. Center components $\overline{E}$ and $\overline{H}$ of a three-dimensional space so that each component of $\overline{E}$ is surrounded by four circulating components $\overline{H}$ and each component $\overline{H}$ is surrounded by four $\overline{E}$ circulating components.

3. Also center $\overline{E}$ and $\overline{H}$ components in time (leapfrog scheme). All calculations in three-dimensional space of interest are completed and stored in memory for a particular time using the data previously stored in computer memory. Then all the calculations of $\overline{H}$ in the space is completed and stored in memory using data calculated of $\overline{E}$ in the previous step.

According to Yee notation, a point in space in a uniform rectangular mesh is defined as:

$$(i, j, k) = (i\Delta x, j\Delta y, k\Delta z) \quad (9)$$

and any function F of space and time evaluated at a discrete point on the grid and at a discrete point in time as:

$$F^n_{i,j,k} = F(i\Delta x, j\Delta y, k\Delta z, n\Delta t) \quad (10)$$

where Δx, Δy, and Δz are the space increments in the directions of x,y,z, and i,j,k,n are integers, Δt time, which is assumed uniform over the interval observation. According to Fig. 1, Yee get the expressions of finite difference (central difference) for the derivatives in space and time. For the particular case of the partial derivative of first order of F in the direction x and evaluated at the fixed time $t_n = n\Delta t$ is:

$$\frac{\partial F(i\Delta x, j\Delta y, k\Delta z, n\Delta t)}{\partial x} = \frac{F^n_{i+\frac{1}{2},j,k} - F^n_{i-\frac{1}{2},j,k}}{\Delta x} + O\left[(\Delta x^2)\right] \quad (11)$$

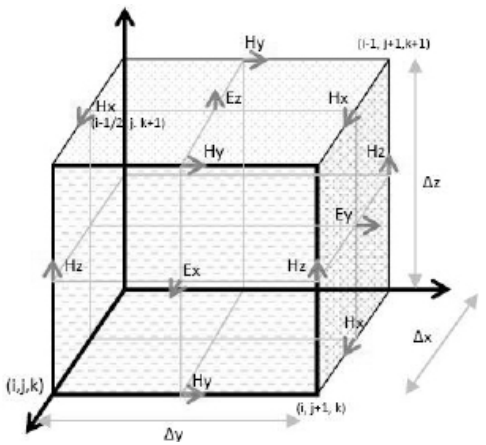



**Figure 1.Yee cell. Position of the vector components of electric $\vec{E}$ and magnetic fiel $\vec{H}$**

Also for temporary partial derivative of the first order of F, evaluated at a point in space (i, j, k) fixed, is:

$$\frac{\partial F(i\Delta x, j\Delta y, k\Delta z, n\Delta t)}{\partial t} = \frac{F_{i,j,k}^{n+1/2} - F_{i,j,k}^{n-1/2}}{\Delta t} + O\left[(\Delta t^2)\right] \quad (12)$$

Assessment $\overline{E}$ and $\overline{H}$ is done at intervals of ½-time alternate.It should be noted that expressions 11 and 12 are obtained by expanding the derivatives in a Taylor series in which the functions O represents the residual higher-order terms [3].

Using expressions 11 and 12 to the equation 8 gives:

$$E_{z(i,j,k)}^{n+1} = \left( \frac{1 - \frac{\sigma_{(i,j,k)}\Delta t}{2\varepsilon_{(i,j,k)}}}{1 + \frac{\sigma_{(i,j,k)}\Delta t}{2\varepsilon_{(i,j,k)}}} \right) E_{z(i,j,k)}^{n} + \left( \frac{\frac{\Delta t}{\varepsilon_{(i,j,k)}}}{1 + \frac{\sigma_{(i,j,k)}\Delta t}{2\varepsilon_{(i,j,k)}}} \right) \cdot \left( \frac{H_{y(i+1/2,j,k)}^{n+1/2} - H_{y(i-1/2,j,k)}^{n+1/2}}{\Delta x} - \frac{H_{x(i,j+1/2,k)}^{n+1/2} - H_{x(i,j-1/2,k)}^{n+1/2}}{\Delta y} \right) \quad (13)$$

## 3. Parallel Computing

Nowadays there are different techniques to achieve parallelism and no doubt this will have better information processing. The technique used with shared memory technology due to multi-core processors is the use of multi-threading.

### 3.1 Openmp

OpenMP is an application program interface(API)thatallowsustoaddconcurrency to applications through parallelism with shared memory. It is based on the creation of threads which are executed in parallel form. Threads share variables of the parent process that creates them. It is available on multiple platforms and languages, from those arising from Unix to windows platform. There are extensions for known languages like C, C + +, Fortran[4].

OpenMP is based upon the existence of multiple threads in the shared memory programming paradigm. A shared memory process consists of multiple threads. OpenMP is an explicit (not automatic) programming model which offers full control over parallelization. OpenMP uses the fork-join model of parallel execution[5]:

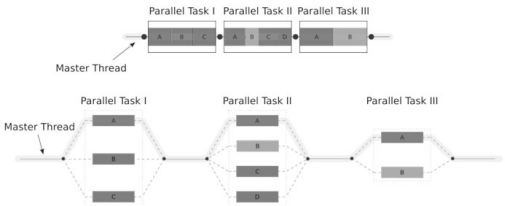



**Figure 2. Fork join model**

All OpenMP programs begin as a single process: the master thread. The master thread executes sequentially until the first parallel region construct is encountered. The master thread then creates a *team* of parallel threads.

The statements in the program that are enclosed by the parallel region construct are then executed in parallel among the various team threads. When the team threads complete the statements in the parallel region construct, they synchronize and terminate, leaving only the master thread.

## 4. Optimization of serial algorithm

FDTD, is an iterative algorithm which requires large processing resources for running. With the increment of multiprocesors, communication generate an additional time, known as overhead [6]. To minimize the overhead, the number of communication instructions must be reduced [7]. The optimization of serial algorithm was realized in order to minimize the overhead. For computational efficiency, loops are changed by matrix multiplication which matrix were divided in small blocks and calculating magnetic and electric fiels in several blocks of data in different instant time. However FDTD has a very large runtime because of iterations.

## 5. Modeling and simulation of parallel algorithm

The optimized FDTD algorithm is implemented by means of Ubuntu using OPENMP directives based on the programming language Fortran 90.

First we just parallelized main algorithm by using directives in order to assign every task in processor and setting some environment variables in several values[8]. Every Fiel is calculated by a single thread and the add is got throught several threads.

Then, care must be paid in handling the type of variables. Some of them are private and another are shared. Taking over threads is necessary because the possibility of loss data if threads collide between them. This requires you should examine variables type to avoid which are shared and private[9]. The parallel algorithm is shown in Table1:

**Do** one time initialization work;
Initialize fields, apply initial conditions;
**to** t = 1 **to** *tmax* **do**
**To every thread**
**to** i, ***to*** *imax* **do**
**to** j, ***to*** *jmax,* **do**
**to** k, ***to*** *kmax* **do**
Update electric fields (Exy,Exz,Eyx,Eyz,Ezx,Ezy) using magnetic fields;
Update magnetic fields (Hxy,Hxz,Hyx,Hyz,Hzx,Hzy) using updated electric fields;
**To a single thread**
Initialization of the source
**To every thread**
Update fields throught of r**eduction of fields** (Ex,Ey,Ez,Hx,Hy y Hz)
Update fields at boundaries, apply boundary conditions;
**Synchronization barrier**

**Table 1: Algorithm: Parallel FDTD Method in Three dimension**

The same directives to Hyx, Hyz,Hzx,Hzy with his respective punctual wave source.

## 6. Results

The example used for this simulation is applied to the free space electromagnetic propagation. The main data that are used for the numeric implementation are presented in Table 2.

| |
|---|
| IMAX=4500 |
| JMAX=4500 |
| KMAX=4500 |
| OperatingFrecuency=2.4E9 |
| Layer PML=12 |
| Grade: 4.0 |
| Number of iterations=6000 |

**Table 2: Characteristics and dimensions of the grid**

The source is a uniform pulse supplied at the position of (IMAX/2,JMAX/2,KMAX/2). It is expressed like:

$$H=Ho \sin(2\pi fon\Delta t) \quad (14)$$

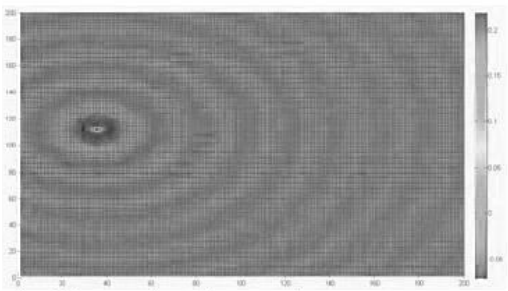


**Figure 3. Puntual Wave Source**

### 6.1 Simulation on the dual processor CPU

In general, parallel computing with multi-core CPU has the function of verify the performance. Runtime can be improved because the directives of OPENMP use the fork join model of parallel execution. Figure 4 shows a comparison between the speed up on serial mode and parallel mode on core duo processor based on Linux.

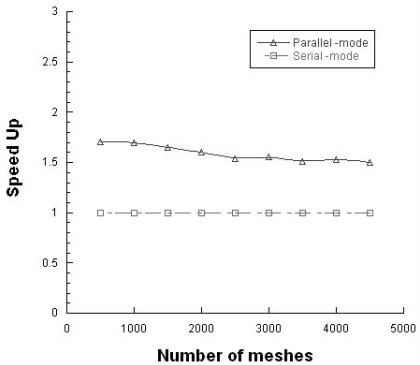



**Figure 4. Speed up with core duo processor**

In serial mode, only core0 is used, for parallel model both core 0 and core 1 are used. According to Figure 4,there is better runtime in parallel mode than serial mode.

### 6.2 Simulation on the dual processor CPU

Figure 5 shows the same comparison between serial and parallel mode to quad core processor. The performance reveals a better speed up when the process is executed on parallel and offers improvements. However after of certain number of meshes Linux also exhibit limitations in 24 matrix of a size

of 7000*7000*7000, which are calculated in every iteration.

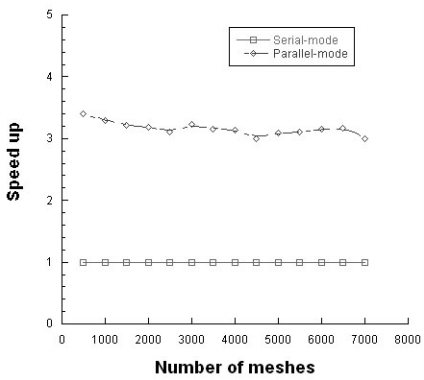



**Figure 5. Speed up with quad core processor**

Figure 6 shows an improvement efficiency in time with respect to number of meshes. Quad core processor reduce the use of resources computing up to 28% on average of a total of 100%.

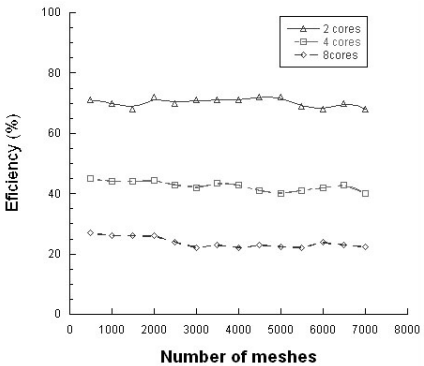



**Figure 6.Efficiency with core duo processor**

However some limitations can be observed with size of grid to parallel mode and serial mode. The biggest size of grid is around 7000*7000*7000.

This introduces a considerable problem that is managment of memory. The management of memory includes mainly two aspects: management memory to data and memory to threads.

## 7. Conclusion

In this paper optimized through multiplication of matrices FDTD algorithm. This technique can avoid using loops which demand computing resources. It improves communication time that is time that processor do not spend in process of user. Then parallelization was implemented in order to have the best performance through of shared memory. It has been confirmed that the performance of multicore CPU is very high for small scale simulation. Results show improvements on runtime. However, It can been looking for the increment in number of cores not always is convenient, because of cores share cache memory. In addition to this the main inconvenient is the manage of the memory and nested loops complicate the parallelization..

Three dimentional parallel FDTD based on multicore has been implemented with quad CPU's and it has been confirmed that this type of parallel computing is sufficiently useful for the electromagnetic simulation and getting better results in less time.

## 8. References

[1] Allen Taflove, Computational Electrodynamics Ed. Artech House, 2000.

[2] Kane S. Yee, "Numerical solution of initial boundary value problem involving Maxwell's equations in isotropic media" IEEE Transactions on antennas and Propagation, vol. 14 1966, pp. 302-307.

[3] Jorge Sosa Pedroza, Manuel Benavides, "Simulaciones de Frontera de Mur y Taflove en el Método de Diferencias Finitas en el Dominio del Tiempo" CIECE 2000, Aguascalientes Ags. México, Marzo 2000.

[4] http://www.openmp.org

[5] http://www.fesb.hr/~psarajce/openmp.html

[6] Parallel Finite-Difference Time-Domain Method, Wenhua Yu, Raj Mittra, Tao Su, Artech House Electromagnetic Analysis Ser.

[7] A parallel FDTD algorithm using the MPI library, C. Guiffaut and K. Mahdjoubi, IEEE Antenas and propagation Magazine, Vol 43, No2, April 2001.

[8] Parallel programming in C with MPI and OPENMP, Michael J. Quinn, McGraw-Hill, ©2004.

[9] Performance Analysis of a Parallel Genetic Algorithm Implementation on a Cluster Environment.

# Performance Analysis of a Parallel Genetic Algorithm Implementation on a Cluster Environment

Irma R. Andalon-Garcia and Arturo Chavoya
*Universidad de Guadalajara, Periférico Norte 799 - L308*
*Zapopan, Jal., México CP 45000*
*agi10073@cucea.udg.mx, achavoya@cucea.udg.mx*

## Abstract

*Genetic algorithms (GAs) are efficient heuristic-guided techniques inspired by natural evolution [7] that are used to solve search and optimization problems. Their major impact has been in producing satisfactory solutions to problems in which the application of standard optimization techniques is not feasible or recommended. GAs are appropriate to be parallelized due to their inherently parallel nature, as they can simultaneously evaluate subpopulations of potential solutions in their search for the fittest chromosome. The aim of this paper is to present results from the performance analysis of our implementation of a parallel genetic algorithm (PGA) by applying several variants to compare. The island and global PGA models were implemented. Two different topologies of the island model —the unidirectional ring and the star topologies— were applied for the migration of the fittest solutions among the chromosome islands. The implementation was developed and tested on a cluster platform.*

***Keywords:*** *Parallel Genetic Algorithm, Parallel Processing, Global Parallel Genetic Algorithm Model, Island Parallel Genetic Algorithm Model.*

## 1. Introduction

At present, genetic algorithms (GAs) are a nature-inspired approach widely used to solve complex problems due to their proved efficiency. Ever since 1975, when the fundamental principles of genetic algorithms were established by Holland [9], GAs have become a robust search and optimization method used to solve problems in several disciplines. One of the first applications of GAs was in dynamic system control [6]. This adaptive approach has been used in science to study gene expression, social systems, ecology, and in the automatic programming of industrial and financial systems with competitive results. For instance, at recent times GAs were applied to the problem of generating a 3D French flag pattern [4]. This model evolved artificial regulatory networks to control cell reproduction based on a 3D cellular automata (CA), until the target cellular structures were obtained.

In this work we present results from the performance analysis of a parallel genetic algorithm (PGA) implemented with different variants (the global population model, the

island model using a ring topology and the island model using a star topology) in order to compare their efficiency. This approach entails the evolution of the population (the set of possible solutions, encoded in a binary representation in a string) through the use of genetic operators similar to those present in the process of natural selection [5] such as inheritance, selection, mutation and crossover, and progressively improving the fitness of the population. The individuals were evaluated using benchmark functions (four different functions were used for the simulations) until the optimal fitness was obtained or a fixed number of generations was reached. We applied parallel programming techniques to take advantage of multiple processors to obtain better response times.

The remainder of the paper is structured as follows: it starts by presenting related work on parallel genetic algorithms (PGAs), followed by an overview of the models of PGAs that were implemented. Our implementation is described next, followed by a simulation section where well-known benchmark functions were used to test the algorithm performance. The next section describes the results obtained in this implementation. The final section presents the conclusions and the future work.

## 2. Related work

Nowadays, PGAs have become an efficient approach to generate solutions for optimization problems. Pit developed and implemented a PGA in order to determine its potential when running simulations [12]. He concluded that with a certain combination of parameters, PGAs might obtain appropriate results for a specific problem, but that they could not be applied effortlessly to other problems.

Alba presents a detailed work on distributed PGAs [1]. Alba et al. also implemented a PGA with the aim of analyzing the technical and practical issues of a distributed PGA model in a multiplatform environment [2]. They concluded that the super linear performance of PGAs could also be achieved in heterogeneous clusters.

Another work on heterogeneous PGAs analyzed their advantages in comparison to homogeneous PGAs [3]. The results were that traditional PGAs obtained better results for multimodal problems than serial GAs; however heterogeneous PGAs only showed improvement when applied to unimodal problems. In the next section, we present a brief description of the PGA models that were implemented in this work.

## 3. Parallel genetic algorithms

GAs are appropriate to be parallelized because their operators can be independently applied to the individuals of the chromosome subpopulations. PGAs are based on the idea that several subpopulations can evolve simultaneously on different processors in order to take advantage of the current technologies to obtain faster results. After a certain number of generations, each subpopulation migrate its best individuals to other subpopulations. In general, there exist three different models of PGA: the global population PGA model, the coarse-grained PGA model and the fine-grained PGA model.

This work concentrates on the first two models, the global population model and the island model. In the island model, the unidirectional ring and the star topologies were applied for the migration of the fittest solutions among the subpopulations, as described below.

### 3.1. Global population model

As the most time-consuming process of GAs is frequently the evaluation function, the global population model, also named master-slave model, parallelizes only the evaluation function. The master or coordinator process stores the global population, executes GA operations, and distributes individuals to the slave processes (Figure 1). The slave processes only evaluate the fitness of the individuals that are received from the coordinator process, and return the results of the evaluations to the coordinator.

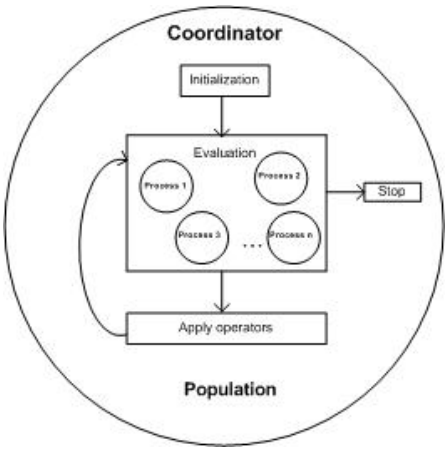



**Figure 1. Global population model**

### 3.2. Island model using the star topology

The second model that we implemented was the island model using the star topology. The island model is a class of coarse-grained PGA models, in which the global population is divided into subpopulations. Each computer node evolves its subpopulation and after a fixed number of generations, each process migrate its individual with the best fitness to the coordinator. The individual to be replaced by the coordinator is selected randomly, and if its fitness is worse than the fitness of the chromosome received, the replacement is made. Figure 2 shows a representation of the island model using the star topology.

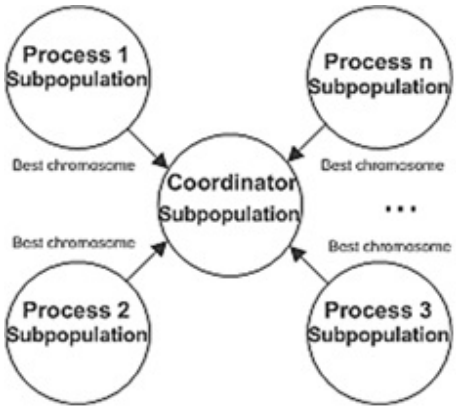



**Figure 2. Island model using the star topology**

### 3.3. Island model using the ring topology

The third model of PGA that we implemented was the island model using the unidirectional ring topology (Figure 3). In general, the model works similarly to the island model using the star topology, described in section 3.2. However, in this topology, the individual with the best fitness migrates to the next process, i.e. process *n* sends its best individual to process *n +1*, and not to the coordinator.

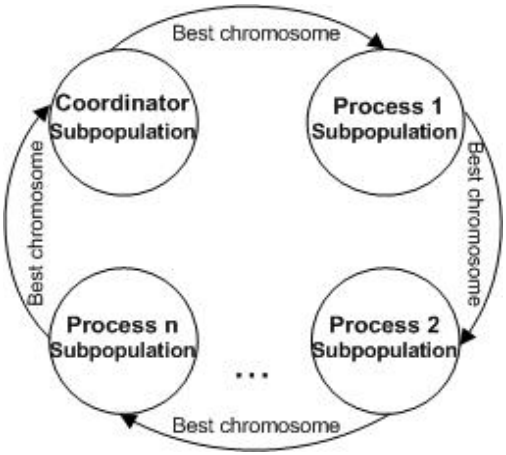



**Figure 3. Island model using the ring topology**

## 4. Implementation

As in a conventional GA (Figure 4), also named canonical or simple GA, in our implementation the alphabet used to represent the chromosomes was {0, 1}, i.e. a binary string. Thus, each individual was coded as a binary string. Chromosomes can contain one or multiple genes, depending on the number of parameters encoded. A group of individuals constitute the population. The initial population was randomly created and the number of individuals in the population was varied according to the complexity of the benchmark function in use. For instance, in the simplest tested function that was implemented, the size of the population we used was 1500 or 2000. However with more complex functions, the size was larger (8000) in order to have a greater variety of potential solutions

The individuals were evaluated by one of the tested functions to calculate their fitness. The standard genetic operators such as selection, crossover and mutation, were applied to the individuals in order to create the next generation. The process was repeated iteratively, so that better solutions for the specific optimization function were obtained from generation to generation until the stopping criterion was reached.

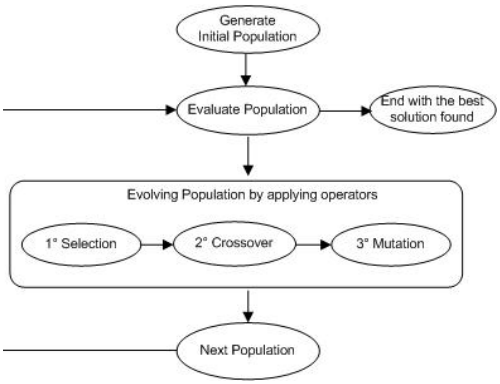



**Figure 4. General structure of a simple GA**

The IEEE-754 coding format [10] —the standard to represent floating point numbers in computer memory— was applied to code the chromosomes for functions that required the use of floating point numbers.

### 4.1. Operators

Basic GA operators were applied to the individuals, with the aim of evolving the population. In the selection operator, individuals with better fitness have better probabilities to survive or to be chosen to produce offspring in the next generation. There are different selection strategies. For instance, the tournament selection [8], the roulette selection [7], the rank-based selection [13] and the elitism selection, the last consists in passing the individual with the best fitness to the next generation. The tournament selection with sets of three individuals was applied in this implementation.

The crossover operator exchanges chromosome segments from two parents previously selected to create two new individuals by combining the segments from the parents. Thus, the offspring chromosomes have information from both parents. Single-point crossover was used where the crossover point was chosen randomly. The crossover rate used in this work was 70% in all the simulations.

The mutation operator flips the value of a randomly selected bit in order to create a slightly different individual. The mutation rate is usually small; we used a value of 0.5% in our implementation. The objective of this operator is to introduce new possible solutions into the next population.

The generational evolution model was used to make the replacement, instead of the steady-state replacement [13]. The number of generations was established depending on the benchmark function used and the tested PGA model. An additional parameter was introduced for the island models, the migration frequency, which is a fixed number that indicates the regularity at which the migration of the best individual occurs. We used a value of 10 or 20, depending on the population size.

The implementation was developed using the C++ programming language on a cluster platform through the use of the standard Message Passing Interface (MPI) library. The implemented benchmark functions and PGA models were run on the cluster provided by Intel, which contained 10 nodes, each with two Intel Quad-Core 1.6 Teraflop CPUs, 24-GB of 1066 MHz, DDR3 RAM, and a 146-GB, 15K RPM hard disk. The operating system used was Red Hat Linux Enterprise Server 5.3 running Intel MPI 3.2. The next section describes the benchmark functions that were used for testing and comparing the models of PGAs implemented.

## 5. Simulation

In order to test the performance of the PGA models, we used four well-known benchmark functions [11]. The functions are the all-ones function, the First De Jong's function, the axis parallel hyper-ellipsoid function and the Rosenbrock's valley function.

The first benchmark function used was the all-ones function, one of the simplest optimization functions. The optimal solution

for this function is an individual in which all the bits in the string are ones. Thus, the fitness is calculated by counting the number of ones in the string. Table 1 shows the parameters used in the simulation for the all-ones function.

| Variables | Global | Island (Star) | Island (Ring) |
|---|---|---|---|
| Chromosome size | 100 | 150 | 150 |
| Population size | 2000 | 2000 | 2000 |
| Number of generations | 150 | 500 | 500 |
| Crossover rate | 70 | 70 | 70 |
| Mutation rate | 0.5 | 0.5 | 0.5 |
| Migration Frequency | N/A | 10 | 10 |

**Table 1. Parameters for the simulations using the all-ones function**

The second function implemented in order to test and compare the models was the First De Jong's function, which is a continuous, convex and unimodal function. Figure 5 shows the function in 3D, i.e. for *n=2*. The implementation was made with *n=20*. Its formula is

$$f(x) = \sum_{i=1}^{n} x_i^2.$$

The objective is to find the global minimum for the function, *f(x) =0*, which is obtained for $x_i = 0$, $i = 1,\ldots,n$.

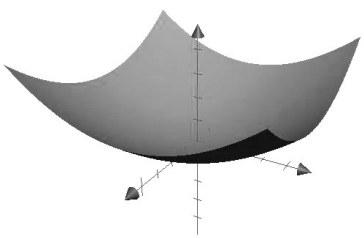


**Figure 5. The First De Jong's function in 3D**

Table 2 shows the parameters used in the First De Jong's function for simulations with *n=20*.

| Variables | Island (Star) | Island (Ring) |
|---|---|---|
| Chromosome size | 640 | 640 |
| Population size | 1500 | 1500 |
| Number of generations | 200 | 500 |
| Crossover rate | 70 | 70 |
| Mutation rate | 0.5 | 0.5 |
| Migration Frequency | 10 | 10 |

**Table 2. Parameters for the simulations using the First De Jong's function**

The third benchmark function used was the axis parallel hyper-ellipsoid function, which is similar to the previous one, but with added complexity. This function is also continuous, convex and unimodal. Figure 6 shows this function in 3D, i.e. for *n=2*. The implementation was made using *n=20*. Its general formula is

$$f(x) = \sum_{i=1} (i \cdot x_i^2) \quad .$$

The aim is to find the global minimum for the function, *f(x) =0*, which is obtained for $x_i = 0$, $i = 1, \ldots, n$.

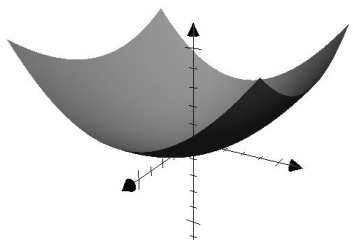


**Figure 6. The axis parallel hyper-ellipsoid function in 3D**

Table 3 shows the parameters used in the axis parallel hyper-ellipsoid function for simulations with *n=20*.

| Variables | Island (Star) | Island (Ring) |
|---|---|---|
| Chromosome size | 640 | 640 |
| Population size | 2000 | 2000 |
| Number of generations | 500 | 500 |
| Crossover rate | 70 | 70 |
| Mutation rate | 0.5 | 0.5 |
| Migration Frequency | 10 | 10 |

**Table 3. Parameters for the simulations using the axis parallel hyper-ellipsoid function**

The last function used for simulations was the Rosenbrock's valley function. This function is continuous and unimodal, but it has some local minima. Figure 7 shows this function in 3D, that is for *n=2*. The formula for this function is

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \quad .$$

The aim is to find the global minimum for the function, *f(x) =0*, which is obtained for $x_i = 1$, $i = 1, \ldots, n$.

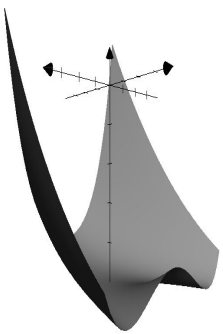


**Figure 7. The Rosenbrock's valley function in 3D**

Table 4 shows the parameters used in simulations using the Rosenbrock's valley function for *n=2*.

| Variables | Island (Star) | Island (Ring) |
|---|---|---|
| Chromosome size | 64 | 64 |
| Population size | 8000 | 8000 |
| Number of generations | 500 | 500 |
| Crossover rate | 70 | 70 |
| Mutation rate | 0.5 | 0.5 |
| Migration Frequency | 20 | 20 |

**Table 4. Parameters for the simulations using the Rosenbrock's valley function**

## 6. Results

The variables used for evaluating the PGA models were the speedup and the best response time. The speedup factor is calculated by the formula

$$S_p = \frac{T_1}{T_p} \quad ,$$

where $T_1$ is the execution time required to perform some task using one processor, and $T_p$ is the execution time necessary to perform the same task but using $p$ processors. Table 5 presents the measured execution times obtained with the all-ones function in the implemented PGA models.

| All-ones | | | | Number of | Processors | | | |
|---|---|---|---|---|---|---|---|---|
| Execution time (seconds) | 1 | 5 | 10 | 20 | 30 | 40 | 50 | 60 |
| Global | 0.788198 | 0.864547 | 0.878792 | 0.896290 | 0.908367 | 0.968589 | 1.210649 | 1.631580 |
| Island (Star) | 1.723460 | 0.418725 | 0.230595 | 0.128474 | 0.104295 | 0.087713 | 0.114141 | 0.301126 |
| Island (Ring) | 2.287748 | 0.579393 | 0.407640 | 0.237657 | 0.147344 | 0.134668 | 0.119686 | 0.126169 |

**Table 5. Execution times for the all-ones function**

The highest speedup reached with the island model using the star topology was 19.64 and it was obtained using 40 processors. The highest speedup for the island model using the ring topology was 19.11. As we can observe the speedup obtained with both island models was almost equal, but the best response time (0.087713) was obtained on the star topology.

Figure 8 shows that with the global population model using the all-ones function, as the number of processors increases, performance worsens, probably due to message passing overhead. Furthermore, with larger chromosomes and a greater number of individuals, memory problems occurred. Thus, this model was omitted for the following functions. In comparison, with the island models, execution times consistently drop up to a certain point as the number of processors increases.

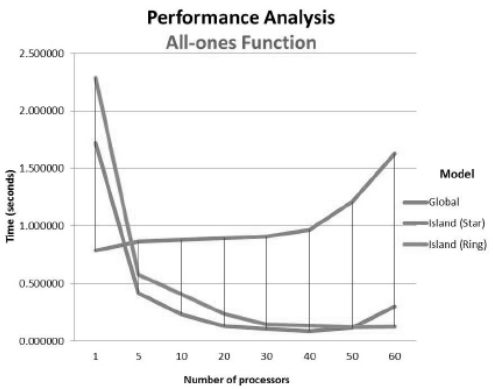



**Figure 8. Performance analysis for the all-ones function**

Table 6 presents the measured execution times obtained for the First De Jong's function with n=20 and using the two variations of the island model.

| First De Jong's Execution time (seconds) | Number of Processors | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 5 | 10 | 20 | 30 | 40 | 50 | 60 |
| Island (Star) | 11.336360 | 2.925864 | 1.798276 | 1.177134 | 0.964805 | 0.902725 | 0.814621 | 0.891774 |
| Island (Ring) | 13.722200 | 5.352370 | 2.993058 | 2.264422 | 1.313778 | 1.071159 | 1.188178 | 2.006842 |

**Table 6. Execution times for the First De Jong's function**

The results obtained were similar to the results for the all-ones function. The best speedup was reached with the island model using the star topology (12.81) and it was obtained using 50 processors, against the value of 13.91 yielded using the ring topology. The best response time (0.814621) was again obtained using the star topology. Figure 9 shows the response times achieved with the island model using both topologies evaluated with the First De Jong's function.

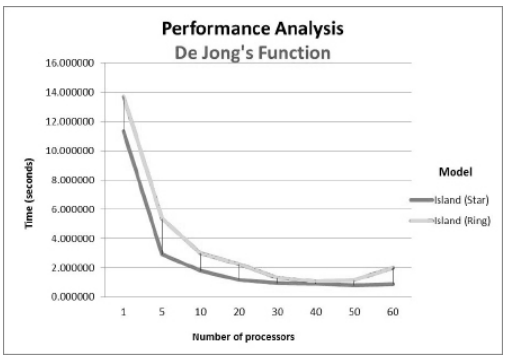



**Figure 9. Performance analysis for the First De Jong's function**

Table 7 presents the measured execution times obtained for the axis parallel hyper-ellipsoid function, with n=20 and using the island model with the star and the ring topologies.

| Axis parallel hyper-ellipsoid | | | | Number of Processors | | | |
|---|---|---|---|---|---|---|---|
| Execution time (seconds) | 1 | 5 | 10 | 20 | 30 | 40 | 50 |
| Island (Star) | 14.782060 | 4.291386 | 2.050976 | 1.370244 | 0.886442 | 1.335376 | 2.296958 |
| Island (Ring) | 14.491580 | 4.528468 | 2.411100 | 1.517672 | 1.212030 | 1.226442 | 1.257518 |

**Table 7. Execution times for the axis parallel hyper-ellipsoid function**

Once again, the results obtained were similar to the results from the simulations using the previous functions. The best speedup was reached with the island model using the star topology (11.95) and it was obtained using 30 processors, against a value of 16.67 reached using the ring topology. The best response time (0.886442) was also obtained using the star topology. Figure 10 shows the response times achieved with the island model using both topologies for the axis parallel hyper-ellipsoid function.

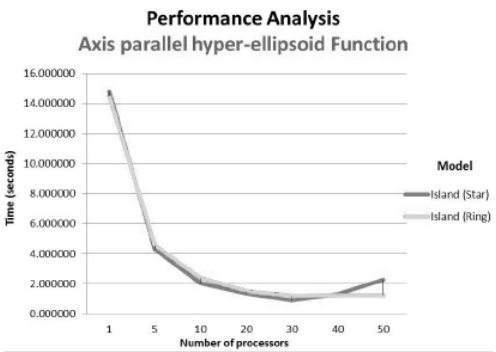



**Figure 10. Performance analysis for the axis parallel hyper-ellipsoid function**

Table 8 presents the measured execution times obtained for the Rosenbrock's function, with n=2 and using the island model with the star and the ring topologies.

| Rosenbrock's valley | | | Number of Processors | | |
|---|---|---|---|---|---|
| Executing time (seconds) | 1 | 3 | 5 | 10 | 15 |
| Island (Star) | 6.615720 | 2.038375 | 1.424185 | 1.028435 | 0.865871 |
| Island (Ring) | 4.626593 | 3.059400 | 1.793145 | 1.032853 | 1.804150 |

**Table 8. Execution times for the Rosenbrock's valley function**

As previously, the results obtained were similar to the results from the simulations using the first three functions. The best speedup was reached with the island model using the star topology (4.47) and it was obtained using 15 processors, against a value of 7.64 reached using the ring topology. The best response time (0.865871) was also obtained using the star topology. Figure 11 shows the response times achieved with the island model using both topologies for the Rosenbrock's valley function.

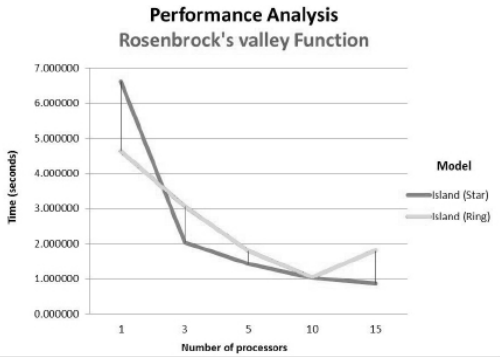



**Figure 11. Performance analysis for the Rosenbrock's valley function**

According to the results obtained from these simulations, the best of the three PGA models evaluated was the island model using the star topology. It is important to note that the results are the average of five simulation runs and only the cases when the optimal solution was found were considered

## 7. Conclusions and Further work

The results obtained with the global population model were disappointing. With the simplest evaluation function —the all-ones function—, execution time increased as the number of processors rose, probably due to message passing overhead. In addition, with larger chromosomes and a greater number of individuals, memory problems occurred.

The island model using the star topology yielded the best response times and speedup for all the tested functions in our implementation, probably because subpopulations evolve independently and all of them send their best solutions to the coordinator, which evolves rapidly.

The island model with the ring topology appears to be a feasible solution; however, the population evolves slowly, as compared to the star topology, probably because the chromosomes with the best fitness migrate through different processes in the ring, taking more time for the best solution to reach the master process.

Future work includes applying the best PGA model to the sequence alignment problem in the area of Bioinformatics, where the large amount of data held in biological databases requires faster methods for processing those data, and parallel programming techniques are needed to obtain better response times.

## 8. Acknowledgments

We would like to thank Intel Corporation for allowing us access to the cluster hosted at their Guadalajara facility. We would also like to thank the Coordination of Design of the CGTI of the Universidad de Guadalajara. Their support has been most helpful in developing this project.

## 9. References

[1] E. Alba, *Análisis y Diseño de Algoritmos Genéticos Paralelos Distribuidos*, PhD tesis Universidad de Málaga, Departamento de Lenguajes y Ciencias de la Computación, Málaga, 1999.

[2] E. Alba, A. J. Nebro, and J. M. Troya, "Heterogeneous Computing and Parallel Genetic Algorithms", *Journal of Parallel and Distributed Computing*, vol. 62, 2002, pp. 1362-1385.

[3] J. Cerný, "Heterogeneous Parallel Genetic Algorithms", *B.S. project Czech Technical University in Prague,* Department of Cybernectics, Prague, 2010.

[4] A. Chavoya, I. R. Andalon-Garcia, C. Lopez-Martin, and M. E. Meda-Campaña, "Use of Evolved Artificial Regulatory Networks to Simulate 3D Cell Differentiation", *BioSystems, Elsevier* vol. 102 , no. 1, 2010, pp. 41-48.

[5] C. Darwin, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, London: John Murray, 1859.

[6] D. E. Goldberg, "Genetic Algorithms and Rules Learning in Dynamic System Control", *ICGA'1985*, 1985, pp. 8-15.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Boston, MA, USA: Addison-Wesley, 1989.

[8] D.E. Goldberg and K. Deb, *A comparative analysis of selection schemes used in genetic algorithms*. Foundations of genetic algorithms, G. Rawlins, Morgan Kaufmann, 1991.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology*, Control and Artificial Intelligence, Cambridge, MA, USA: MIT Press, 1992.

[10] IEEE Task P754. ANSI/IEEE 754-1985, "*Standard for Binary Floating-Point Arithmetic*". IEEE, New York, NY, USA, 1985.

[11] M. Molga and C. Smutnici. *Test functions for optimization needs*. www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf, 2005.

[12] L. J. Pit, *Parallel Genetic Algorithms*, M.S. thesis Leiden University, Leiden Institute of Advanced Computer Science, Netherlands, 1995.

[13] D. Whitley; "The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best", *Proceedings of the 3rd International Conference on Genetic Algorithms and their application (ICGA)*, J.D. Schaffer (Ed.), Morgan Kaufmann, San Mateo CA, 1989, pp. 116-121.

# Comparison of Solution Strategies for Structure Deformation Using Hybrid OpenMP-MPI Methods

J. M. Vargas-Felix, S. Botello-Rionda
*Center of Mathematical Research (CIMAT)*
*miguelvargas@cimat.mx, botello@cimat.mx*

## Abstract

*Finite element analysis of elastic deformation of three-dimensional structures using a fine mesh could require to solve systems of equations with several mil-lion variables.*

*Domain decomposition is used to separate work-load, instead of solving a huge system of equations, the domain is partitioned and for each partition a smaller system of equations is solved, all partitions are solved in parallel. Each partition is solved in a single MPI (Message Passing Interface) process. Updates of boundary conditions among processes are done through MPI message routines.*

*Iterative and direct algorithms for solving local systems of equations are programmed using OpenMP to run in multi-core processors.*

*Different configurations for domain decompositions are tested, for instance, using many small partitions each one using a single core, or fewer partitions using several cores that share memory. Numerical experiments were done using a Beowulf cluster, looking for an adequate compromise between solution time and memory requirements*

***Keywords:*** *Parallel computing, domain decomposition, finite element method, sparse systems, linear algebra.*

## 1. Introduction

This is a high performance/large scale application case study of the finite element method for solid mechanics. Our goal is to calculate displacements, strain and stress of solids discretized with large meshes (mil-lions of elements) using parallel computing.

We want to calculate linear inner displacements of a solid resulting from forces or displacements imposed on its boundaries (i.e. Figure 1).

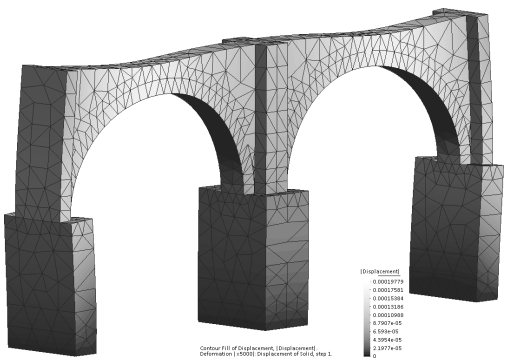


**Figure 1. Example of deformation problem**

The displacement vector inside the domain is defined as

$$\mathbf{u}(x\ y\ z)=\begin{pmatrix} u(x\ y\ z) \\ v(x\ y\ z) \\ w(x\ y\ z) \end{pmatrix}$$

the strain vector $\boldsymbol{\varepsilon}$ is

$$\boldsymbol{\varepsilon}=\begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_y \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{pmatrix}=\begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y}+\frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial z}+\frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x}+\frac{\partial u}{\partial z} \end{pmatrix}=\begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{pmatrix}\begin{pmatrix} u \\ v \\ w \end{pmatrix}=\mathbf{E}\mathbf{u} \qquad (1)$$

where $\varepsilon_x$, $\varepsilon_y$ and $\varepsilon_z$ are normal strains; $\gamma_{xy}$, $\gamma_{yz}$ and $\gamma_{zx}$ are shear strains. We define a differential operator **E**.

Stress vector is defined as,

$$\boldsymbol{\sigma}=\left(\sigma_x,\sigma_y,\sigma_z,\tau_{xy},\tau_{yz},\tau_{zx}\right)^{\mathrm{T}}$$

where $\sigma_x,\sigma_y$ and $\sigma_z$ are normal stresses; $\tau_{xy},\tau_{yz}$ and $\tau_{zx}$ are tangential stresses.

Stress an strain are related by

$$\boldsymbol{\sigma}=\mathbf{D}\boldsymbol{\varepsilon} \qquad (2)$$

$\mathbf{D}$ is called the constitutive matrix, it depends on Young's modulus and Poisson's ratio characteristic of media.

Solution is found using the finite element method with the Galerkin weighted residuals. This means that we solve the integral problem in each element using a weak formulation. The integral expression of equilibrium in elasticity problems can be obtained using the principle of virtual work [1 pp65-71],

$$\int_V \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}dV=\int_V \delta\mathbf{u}^{\mathrm{T}}\mathbf{b}\,dV+\oint_A \delta\mathbf{u}^{\mathrm{T}}\mathbf{t}dA+\sum_i \delta\mathbf{u}_i^{\mathrm{T}}\mathbf{q}_i \ , \qquad (3)$$

here $\mathbf{b}$, $\mathbf{t}$ and $\mathbf{q}$ are the vectors of mass, boundary and punctual forces respectively. The weight functions for weak formulation are chosen to be the interpolation functions of the element, these are $N_i$ ,i=1,..., $M$ . $M$ is the number of nodes of the element, $\mathbf{u}_i$ is the coordinate of the *i*-th node, we have that

$$\mathbf{u}=\sum_{i=1}^{M} N_i\mathbf{u}_i \ . \qquad (4)$$

Using (4), we can rewrite (1) as:

$$\boldsymbol{\varepsilon}=\sum_{i=1}^{M}\mathbf{E}N_i\mathbf{u}_i$$

or in a more compact form

$$\boldsymbol{\varepsilon}=\begin{pmatrix}\mathbf{E}N_1 & \mathbf{E}N_2 & \cdots & \mathbf{E}N_M\end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{pmatrix}=\mathbf{B}\mathbf{u}$$

Now we can express (2) as $\boldsymbol{\sigma}=\mathbf{D}\mathbf{B}\mathbf{u}$ , and then (3) by

$$\underbrace{\int_{V^e}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}dV^e\mathbf{u}}_{\mathbf{K}^e}=\underbrace{\int_{V^e}\mathbf{b}dV^e}_{\mathbf{f}_\mathbf{b}^e}=\underbrace{\oint_{A^e}\mathbf{t}dA^e}_{\mathbf{f}_\mathbf{t}^e}+\mathbf{q}^e \qquad (5)$$

By integrating (5) we obtain a system of equations for each element,

$$\mathbf{K}^e\mathbf{u}^e=\mathbf{f}_\mathbf{b}^e+\mathbf{f}_\mathbf{t}^e+\mathbf{q}^e$$

All systems of equations are assembled into a global system of equations,

$$\mathbf{Ku = f}.$$

$\mathbf{K}$ is called the stiffness matrix, if enough boundary conditions are applied, it will be symmetric positive definite (SPD). By construction it is sparse with storage requirements of order $O(n)$, where $n$ is the total number of nodes in the domain. By solving this system we will obtain the displacements of all nodes in the domain. The solution of this system of equations is the task that we want do using parallel computing.

## 2. Domain Decomposition

### 2.1. Schwarz alternating method

We can separate a huge finite element problem into smaller problems by partitioning the mesh to create sub domains. In this case we will use domain decomposition with overlapping, these methods were first studied by Schwarz [2] (figure 2). For each sub-domain a system of equations with a SPD matrix is assembled thus we can solve it using solver algorithms that are implemented to run in parallel, such as Cholesky factorization or preconditioned conjugate gradient.

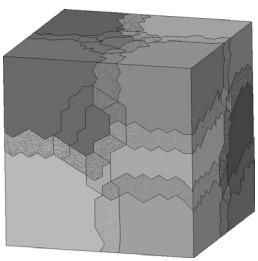


**Figure 2. Overlapped domain decomposition**

The domain decomposition algorithm we used is parallel Schwarz alternating method [3], this is an iterative algorithm. We start with a domain $\mathbf{\Omega}$ with boundary $\partial\Omega$ (figure 3).

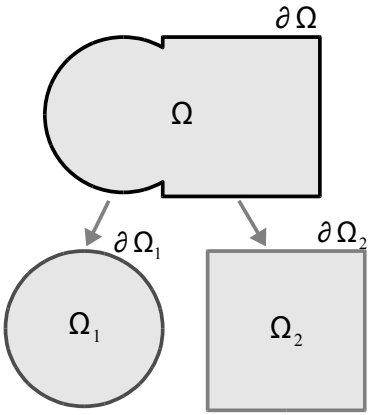



**Figure 3. Domain partitioning**

Let $\mathbf{L}$ be a differential operator such that $\mathbf{Lx= y}$ in $\mathbf{\Omega}$. Dirichlet conditions $\mathbf{x = b}$ are applied on $\partial\Omega$. The domain is divided in two partitions $\mathbf{\Omega_1}$ and $\mathbf{\Omega_2}$ with boundaries $\partial\Omega_1$ and $\partial\Omega_2$ respectively.

Partitions are overlapped, now $\Omega = \Omega_1 \cup \Omega_2$ and $\Omega_1 \cap \Omega_2 \neq \varnothing$. We define artificial boundaries $\Gamma_1$ and $\Gamma_2$, these are part of $\mathbf{\Omega_1}$ and $\mathbf{\Omega_2}$, and are inside $\mathbf{\Omega}$ (figure 4).

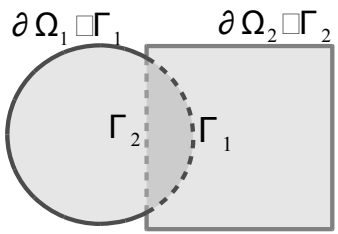



**Figure 4. Artificial boundaries**

Schwarz alternating method consists on solving each partition independently, fixing Dirichlet conditions in artificial boundaries with values from adjacent partition resulting from previous iteration.

$\mathbf{x}_1^0$ , $\mathbf{x}_2^0$ , initial approximations
$\varepsilon$ tolerance
$i \leftarrow 0$
while $\left\|\mathbf{x}_1^i - \mathbf{x}_1^{i-1}\right\| > \varepsilon$ or $\left\|\mathbf{x}_2^i - \mathbf{x}_2^{i-1}\right\| > \varepsilon$

| solve: | solve: |
|---|---|
| $\mathbf{L}\mathbf{x}_1^i = \mathbf{y}$ in $\Omega_1$ | $\mathbf{L}\mathbf{x}_2^i = \mathbf{y}$ in $\Omega_2$ |
| with $\mathbf{x}_1^i = \mathbf{b}$ on $\partial\Omega_1 \setminus \Gamma_1$ | with $\mathbf{x}_2^i = \mathbf{b}$ on $\partial\Omega_2 \setminus \Gamma_2$ |
| $\mathbf{x}_1^i \leftarrow \mathbf{x}_2^{i-1}\big\|_{\Gamma_1}$ on $\Gamma_1$ | $\mathbf{x}_2^i \leftarrow \mathbf{x}_1^{i-1}\big\|_{\Gamma_2}$ on $\Gamma_2$ |

$i \leftarrow i+1$

When the $\mathbf{L}$ operator has a matrix representation, alternating Schwarz algorithm corresponds (due to overlapping) to the iterative Gauss-Seidel by blocks [3 p13].

In finite element problems, overlapping is adding to each partition one or several layers of elements adjacent to the boundary between partitions (figure 5).

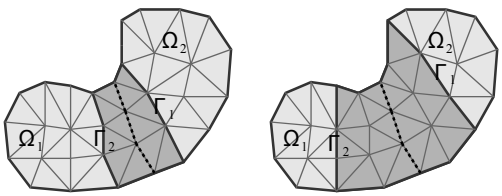



**Figure 5. One and two layers of overlapping**

## 2.2. Convergence speed

There is a degradation on the convergence speed when the number of partitions raise [3 p53]. Next image shows a pathological case of domain decomposition (figure 6).

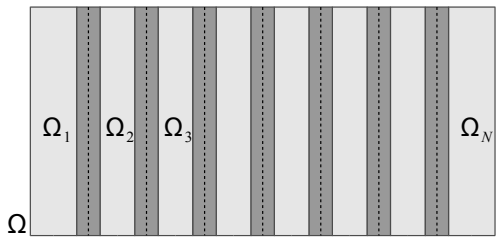



**Figure 6. Pathological case**

In each iteration of the alternating Schwarz method information is only transmitted to adjacent partitions. Therefore, if we have a boundary condition different to zero in the boundary of $\mathbf{\Omega_1}$, and we start in iteration 0, it will take $N$ iterations for the local solution of partition $\mathbf{\Omega}_N$ to be different to 0.

The Schwarz algorithm typically converge at a speed that is independent (or slightly independent) of mesh density when the overlapping is large enough [3 p74].

A deeper study of theory of Schwarz algorithms can be found in [2].

## 3. Computer clusters and MPI

We developed a software program that runs in parallel in a Beowulf cluster [4]. A Beowulf cluster (figure 7) consists of several multi-core computers (nodes) connected with a high speed network.

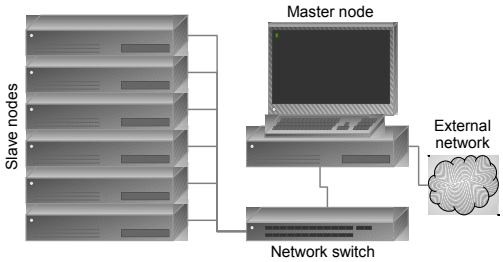



**Figure 7. Beowulf cluster**

In our software implementation each partition is assigned to one process. To parallelize the program and move data among nodes we used the Message Passing Interface (MPI) schema [5], it contains set of tools that makes easy to start several instances of a program (processes) and run them in parallel. Also, MPI has several libraries with a rich set of routines to send and receive data messages among processes in an efficient way. MPI can be configured to execute one or several processes per node.

For partitioning the mesh we used the METIS library [6].

## 4. OpenMP

Using domain decomposition with MPI we could have a partition assigned to each node of a cluster, we can solve all partitions concurrently. If each node is a multi-core computer we can also parallelize the solution of the system of equations of each partition. To implement this parallelization we use the OpenMP model.

This parallelization model consists in compiler directives inserted in the source code to parallelize sections of code. All cores have access to the same memory, this model is known as shared memory schema.

In modern computers with shared memory architecture the processor is a lot faster than the memory [7].

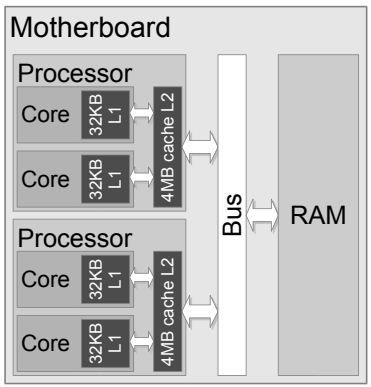



**Figure 8. Dual multi-core computer**

To overcome this, a high speed memory called cache exists between the processor and RAM. This cache reads blocks of data from RAM meanwhile the processor is busy, using an heuristic to predict what the program will require to read next. Modern processor have several caches that are organized by levels (L1, L2, etc), L1 cache is next to the core. It is important to considerate the cache when programming high performance applications, table 1 indicates the number of clock cycles needed to access each kind of memory by a Pentium M processor.

**Table 1. Data access speed**

| Access to | CPU cycles |
|---|---|
| CPU registers | <=1 |
| L1 cache | 3 |
| L2 cache | 14 |
| RAM | 240 |

A big bottleneck in multi-core systems with shared memory is that only one core can access the RAM at the same time.

Another bottleneck is the cache consistency. If two or more cores are accessing the same RAM data then different copies of this data could exists in each core's cache, if a core modifies its cache copy then the system will need to update all caches and RAM, to keep consistency is complex and expensive [8]. Also, it is necessary to consider that cache circuits are designed to be more efficient when reading continuous memory data in an ascending sequence [8 p15].

To avoid lose of performance due to wait for RAM access and synchronization times due to cache inconsistency several strategies can be use:

- Work with continuous memory blocks.
- Access memory in sequence.
- Each core should work in an independent memory area.

Algorithms to solve our system of equations should take care of these strategies.

## 5. Matrix storage

An efficient method to store and operate matrices of this kind of problems is the Compressed Row Storage (CRS) [9 p362]. This method is suitable when we want to access entries of each row of a matrix $\mathbf{A}$ sequentially.

For each row $i$ of $\mathbf{A}$ we will have two vectors, a vector $\mathbf{v}_i^{\mathbf{A}}$ that will contain the non-zero values of the row, and a vector $\mathbf{j}_i^{\mathbf{A}}$ with their respective column indexes. Figure 9 shows a matrix $\mathbf{A}$ and its CRS representation.

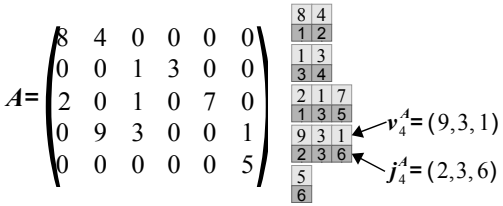



**Figure 9. Compress row storage**

The size of the row will be denoted by $\left|\mathbf{v}_i^{\mathbf{A}}\right|$ or by $\left|\mathbf{j}_i^{\mathbf{A}}\right|$. Therefore the $q$-th non zero value of the row $i$ of $\mathbf{A}$ will be denoted by $\left(\mathbf{v}_i^{\mathbf{A}}\right)_q$ and the index of this value as $\left(\mathbf{j}_i^{\mathbf{A}}\right)_q$, with $q = 1, \ldots, \left|\mathbf{v}_i^{\mathbf{A}}\right|$.

If we do not order entries of each row, then to search an entry with certain column index will have a cost of $O\left(\left|\mathbf{v}_i^{\mathbf{A}}\right|\right)$ in the worst case. To improve it we will keep $\mathbf{v}_i^{\mathbf{A}}$ and $\mathbf{j}_i^{\mathbf{A}}$ ordered by the indexes $\mathbf{j}_i^{\mathbf{A}}$. Then we could perform a binary algorithm to have an search cost of $O\left(\log_2\left|\mathbf{v}_i^{\mathbf{A}}\right|\right)$.

The main advantage of using Compressed Row Storage is when data in each row is stored continuously and accessed in a sequential way, this is important because we will have and efficient processor cache usage [8].

## 6. Parallel Cholesky for Sparse Matrices

The cost of using Cholesky factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathbf{T}}$ is expensive if we want to solve systems of equations with full matrices, but for sparse matrices we could reduce this cost significantly if we use reordering strategies and we store factor matrices using CRS identifying non zero entries using symbolic factorization. With

this strategies we could maintain memory and time requirements near to $O(n)$. Also Cholesky factorization could be implemented in parallel.

Formulae to calculate $\mathbf{L}$ entries are

$$L_{ij} = \frac{1}{L_{jj}}\left( A_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk} \right), \text{for } i > j \; ; \quad (6)$$

$$L_{jj} = \sqrt{A_{jj} - \sum_{k=1}^{j-1} L_{jk}^2} \; . \quad (7)$$

## 6.1. Reordering rows and columns

By reordering the rows and columns of a SPD matrix $\mathbf{A}$ we could reduce the fill-in (the number of non-zero entries) of $\mathbf{L}$. Figure 10 shows the non zero entries of $\mathbf{A} \in \Re^{556\times 556}$ and the non zero entries of its Cholesky factorization $\mathbf{L}$.

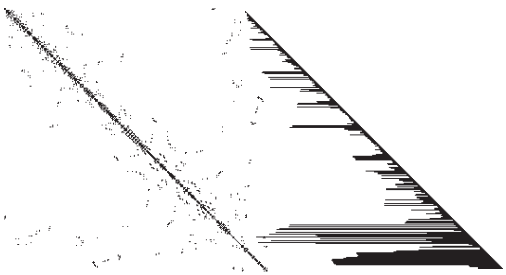


**Figure 10. Unordered Cholesky factorization**

The number of non zero entries of $\mathbf{A}$ is $\eta(\mathbf{A}) = 1810$, and for $\mathbf{L}$ is $\eta(\mathbf{L}) = 8729$. Figure 11 shows $\mathbf{A}$ with reordering by rows and columns.

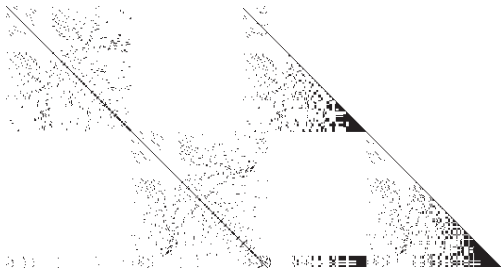


**Figure 11. Reordered Cholesky factorization**

By reordering we have a new factorization with $\eta(\mathbf{L}) = 3215$, reducing the fill-in to 0.368 of the size of the not reordered version. Both factorizations allow us to solve the same system of equations.

The most common reordering heuristic to reduce fill-in is the minimum degree algorithm, the basic version is presented in [10 p116]:

Let be a matrix $\mathbf{A}$ and its corresponding graph $G_0$
$i \leftarrow 1$
repeat
  Let node $x_i$ in $G_{i-1}(X_{i-1}, E_{i-1})$ have minimum degree
  Form a new elimination graph $G_i(X_i, E_i)$ as follow:
    Eliminate $x_i$ and its edges from $G_{i-1}$
    Add edges make $\text{adj}(x_1)$ adjacent pairs in $G_i$
  $i \leftarrow i+1$
while $i < |X|$

More advanced versions of this algorithm can be consulted in [11].

There are more complex algorithms that perform better in terms of time and memory requirements, the nested dissection

algorithm developed by Karypis and Kumar [6] included in METIS library gives very good results.

### 6.2. Symbolic Cholesky factorization

This algorithm identifies non zero entries of $\mathbf{L}$, a deep explanation could be found in [12 p86-88].

For an sparse matrix $\mathbf{A}$, we define $\mathbf{a}_j = \{k > j \,|\, A_{jk} \neq 0\}$, , as the set of non zero entries of column $j$ of the strictly lower triangular part of $\mathbf{A}$.

In similar way, for matrix $\mathbf{L}$ we define the set $\mathbf{l}_j = \{k > j \,|\, L_{jk} \neq 0\}$, $j = 1 \ldots n$.

We also use sets define sets $\mathbf{r}_j$ that will contain columns of $\mathbf{L}$ which structure will affect the column $j$ of $\mathbf{L}$. The algorithm is:

for $j \leftarrow 1 \ldots n$
$\mathbf{r}_j \leftarrow \varnothing$
$\mathbf{l}_j \leftarrow \mathbf{a}_j$
for $i \in \mathbf{r}_j$
$\mathbf{l}_j \leftarrow \mathbf{l}_j \cup \mathbf{l}_i \setminus \{j\}$
$p \leftarrow \begin{cases} \min\{i \in \mathbf{l}_j\} & \text{if} \quad \mathbf{l}_j \neq \varnothing \\ j & \text{other case} \end{cases}$
$\mathbf{r}_p \leftarrow \mathbf{r}_p \cup \{j\}$

This algorithm is very efficient, complexity in time and memory usage has an order of $O(\eta(\mathbf{L}))$. Symbolic factorization could be seen as a sequence of elimination graphs [10 pp92-100].

### 6.3. Filling entries in parallel

Once non zero entries are determined we can rewrite (6) and (7) as

$$L_{ij} = \frac{A_{ij}}{L_{jj}} - \frac{1}{L_{jj}} \sum_{\substack{k \in \mathbf{l}_i \cap \mathbf{l}_j \\ k<j}} L_{ik} L_{jk} \text{ , for } i > j \text{ ;}$$

$$L_{jj} = \sqrt{A_{jj} - \sum_{\substack{k \in \mathbf{l}_j \\ k<j}} L_{jk}^2} \text{ .}$$

Using notation from section 5, the resulting algorithm to fill non zero entries is:

for $j \leftarrow 1 \ldots n$
  $L_{jj} \leftarrow A_{jj}$
  for $q \leftarrow 1 \ldots \left|\mathbf{v}_j^{\mathbf{L}}\right|$
    $L_{jj} \leftarrow L_{jj} - \left(\mathbf{v}_j^{\mathbf{L}}\right)_q \left(\mathbf{v}_j^{\mathbf{L}}\right)_q$
  $L_{jj} \leftarrow \sqrt{L_{jj}}$
  $L_{jj}^{\mathbf{L}^T} \leftarrow L_{jj}$
  parallel for $q \leftarrow 1 \ldots \left|\mathbf{j}_j^{\mathbf{L}^T}\right|$
    $i \leftarrow \left(\mathbf{j}_j^{\mathbf{L}^T}\right)_q$
    $L_{ij} \leftarrow A_{ij}$
    $r \leftarrow 1$ ; $\rho \leftarrow \left(\mathbf{j}_i^{\mathbf{L}}\right)_r$
    $s \leftarrow 1$ ; $\sigma \leftarrow \left(\mathbf{j}_i^{\mathbf{L}}\right)_s$
    repeat
      while $\rho < \sigma$
        $r \leftarrow r+1$ ; $\rho \leftarrow \left(\mathbf{j}_i^{\mathbf{L}}\right)_r$
      while $\rho > \sigma$
        $s \leftarrow s+1$ ; $\sigma \leftarrow \left(\mathbf{j}_i^{\mathbf{L}}\right)_s$
      while $\rho = \sigma$
        if $\rho = j$
          exit repeat
        $L_{ij} \leftarrow L_{ij} - \left(\mathbf{v}_i^{\mathbf{L}}\right)_r \left(\mathbf{v}_j^{\mathbf{L}}\right)_s$
        $r \leftarrow r+1$ ; $\rho \leftarrow \left(\mathbf{j}_i^{\mathbf{L}}\right)_r$
        $s \leftarrow s+1$ ; $\sigma \leftarrow \left(\mathbf{j}_i^{\mathbf{L}}\right)_s$
    $L_{ij} \leftarrow \frac{L_{ij}}{L_{jj}}$
    $L_{ji}^{\mathbf{L}^T} \leftarrow L_{ij}$

This algorithm could be parallelized if we fill column by column. Entries of each column can be calculated in parallel with OpenMP, because there are no dependence among them [13 pp442-445]. Calculus of each column is divided among cores (figure 12).

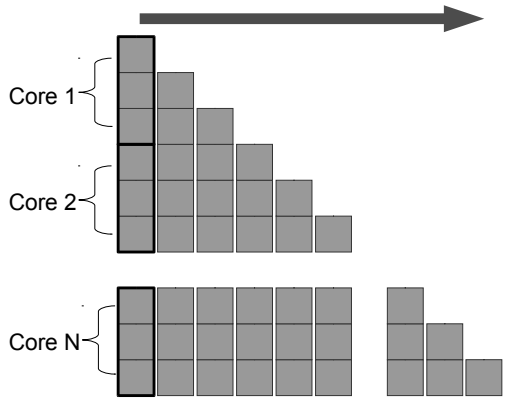



**Figure 12. Cholesky parallelized filling**

Cholesky solver is particularly efficient because the stiffness matrix is factorized once. The domain is partitioned in many small sub-domains to have small and fast Cholesky factorizations.

## 7. Parallel preconditioned conjugate gradient

Conjugate gradient (CG) is a natural choice to solve systems of equations with SPD matrices, we will discuss some strategies to improve convergence rate and make it suitable to solve large sparse systems using parallelization.

The condition number $\kappa$ of a non singular matrix $\mathbf{A} \in \Re^{m \times m}$, given a norm $\|\cdot\|$ is defined as

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| .$$

For the norm $\|\cdot\|_2$,

$$\kappa_2(\mathbf{A}) = \|\mathbf{A}\|_2 \cdot \|\mathbf{A}^{-1}\|_2 = \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A})},$$

where $\sigma$ is a singular value of $\mathbf{A}$.

For a SPD matrix,

$$\kappa(A) = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})},$$

where $\lambda$ is an eigenvalue of $\mathbf{A}$.

A system of equations **Ax= b** is bad conditioned if a small change in the values of $\mathbf{A}$ or $\mathbf{b}$ results in a large change in $\mathbf{x}$. In well conditioned systems a small change of $\mathbf{A}$ or $\mathbf{b}$ produces an small change in $\mathbf{x}$. Matrices with a condition number near to 1 are well conditioned.

A preconditioner for a matrix $\mathbf{A}$ is another matrix $\mathbf{M}$ such that $\mathbf{M}\,\mathbf{A}$ has a lower condition number

$$\kappa(\mathbf{MA}) < \kappa(\mathbf{A})$$

In iterative stationary methods (like Gauss-Seidel) and more general methods of Krylov subspace (like conjugate gradient) a preconditioner reduces the condition number and also the amount of steps necessary for the algorithm to converge.

Instead of solving

$$\mathbf{Ax} - \mathbf{b} = \mathbf{0},$$

with preconditioning we solve

$$\mathbf{M}(\mathbf{Ax} - \mathbf{b}) = \mathbf{0}$$

The preconditioned conjugate gradient algorithm is:

$\mathbf{x}_0$ , initial approximation
$\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$ , initial gradient
$\mathbf{q}_0 \leftarrow \mathbf{M}\mathbf{r}_0$
$\mathbf{p}_0 \leftarrow \mathbf{q}_0$ , initial descent direction
$k \leftarrow 0$
while $\|\mathbf{r}_k\| > \varepsilon$
$\alpha_k \leftarrow -\frac{\mathbf{r}_k^{\mathrm{T}}\mathbf{q}_k}{\mathbf{p}_k^{\mathrm{T}}\mathbf{A}\mathbf{p}_k}$
$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$
$\mathbf{r}_{k+1} \leftarrow \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k$
$\mathbf{q}_{k+1} \leftarrow \mathbf{M}\mathbf{r}_{k+1}$
$\beta_k \leftarrow \frac{\mathbf{r}_{k+1}^{\mathrm{T}}\mathbf{q}_{k+1}}{\mathbf{r}_k^{\mathrm{T}}\mathbf{q}_k}$
$\mathbf{p}_{k+1} \leftarrow \mathbf{q}_{k+1} + \beta_k \mathbf{p}_k$
$k \leftarrow k+1$

For large and sparse systems of equations it is necessary to choose preconditioners that are also sparse.

We will talk about three kinds of preconditioners suitable for sparse systems with SPD matrices:

- Jacobi $\mathbf{M}^{-1} = (\mathrm{diag}(\mathbf{A}))^{-1}$ .
- Incomplete Cholesky factorization $\mathbf{M}^{-1} = \mathbf{G}_l \mathbf{G}_l^{\mathrm{T}}$ , $\mathbf{G}_l \approx \mathbf{L}$ .
- Factorized sparse approximate inverse $\mathbf{M} = \mathbf{H}_l^{\mathrm{T}} \mathbf{H}_l$ , $\mathbf{H}_l \approx \mathbf{L}^{-1}$ .

For the first two preconditioners, $\mathbf{M}$ is not constructed, instead $\mathbf{M}^{-1}$ is defined and we have to solve a system of equations in each step to obtain $\mathbf{q}_k$

$$\mathbf{M}^{-1}\mathbf{q}_k = \mathbf{r}_k \ .$$

Parallelization of the preconditioned CG is done using OpenMP, operations parallelized are matrix-vector, dot products and vector sums. To synchronize threads has a computational cost, it is possible to modify to CG to reduce this costs maintaining numerical stability [14].

### 7.1. Jacobi preconditioner

The diagonal part of $\mathbf{M}^{-1}$ is stored as a vector, $\mathbf{M}^{-1} = (\mathrm{diag}(\mathbf{A}))^{-1}$ .

Parallelization of this algorithm is straightforward, because the calculus of each entry of $\mathbf{q}_k$ is independent.

### 7.2. Incomplete Cholesky factorization preconditioner

This preconditioner has the form

$$\mathbf{M}^{-1} = \mathbf{G}_l \mathbf{G}_l^{\mathrm{T}}$$

where $\mathbf{G}_l$ is a lower triangular sparse matrix that have structure similar to the Cholesky factorization of $\mathbf{A}$ .

The structure of $\mathbf{G}_0$ is equal to the structure of the lower triangular form of $\mathbf{A}$ .

The structure of $\mathbf{G}_m$ is equal to the structure of $\mathbf{L}$ (complete Cholesky factorization of $\mathbf{A}$ ).

For $0 < l < m$ the structure of $\mathbf{G}_l$ is creating having a number of entries between $\mathbf{L}$ and the lower triangular form of $\mathbf{A}$ , making easy to control the sparsity of the preconditioner.

Values of $\mathbf{G}_l$ are filled using (6) and (7). This preconditioner is not always stable [15 p535].

The use of this preconditioner implies to solve a system of equations in each CG step using a backward and a forward substitution algorithm, this operations are fast given the sparsity of $\mathbf{G}_l$ . Unfortunately

the dependency of values makes these substitutions very hard to parallelize.

### 7.3. Factorized sparse approximate inverse preconditioner

The aim of this preconditioner is to construct $\mathbf{M}$ to be an approximation of the inverse of $\mathbf{A}$ with the property of being sparse. The inverse of a sparse matrix is not necessary sparse.

A way to create an approximate inverse is to minimize the Frobenius norm of the residual **I - AM** ,

$$F(\mathbf{M}) = \|\mathbf{I} - \mathbf{AM}\|_{\mathrm{F}}^2 \quad (8)$$

The Frobenius norm is defined as $\|\mathbf{A}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\mathrm{tr}(\mathbf{A}^{\mathrm{T}}\mathbf{A})}$.

It is possible to separate (8) into decoupled sums of 2-norms for each column [16],

$$F(\mathbf{M}) = \|\mathbf{I} - \mathbf{AM}\|_{\mathrm{F}}^2 = \sum_{j=1}^{n}\|\mathbf{e}_j - \mathbf{Am}_j\|_2^2 ,$$

where $\mathbf{e}_j$ is the j-th column of $\mathbf{I}$ and $\mathbf{m}_j$ is the j-th column of $\mathbf{M}$. With this separation we can parallelize the construction of the preconditioner.

The factorized sparse approximate inverse preconditioner [17] creates a preconditioner

$$\mathbf{M} = \mathbf{G}_l^{\mathrm{T}}\mathbf{G}_l$$

where $\mathbf{G}$ is a lower triangular matrix such that

$$\mathbf{G}_l \approx \mathbf{L}^{-1}$$

where $\mathbf{L}$ is the Cholesky factor of $\mathbf{A}$. $l$ is a positive integer that indicates a level of sparsity of the matrix.

Instead of minimizing (8), we minimize $\|\mathbf{I} - \mathbf{G}_l\mathbf{L}\|_{\mathrm{F}}^2$, it is noticeable that this can be done without knowing $\mathbf{L}$, solving the equations $(\mathbf{G}_l\mathbf{L}^{\mathrm{T}})_j = (\mathbf{L}^{\mathrm{T}})_j$ , $(i,j) \in \mathbf{S}_l$ ,

this is equivalent to

$$(\mathbf{G}_l\mathbf{A})_j = (\mathbf{I})_j \ , \ (i,j) \in \mathbf{S}_l ,$$

$\mathbf{S}_l$ contains the structure of $\mathbf{G}_l$.

This preconditioner has these features:

- $\mathbf{M}$ is SPD if there are no zeroes in the diagonal of $\mathbf{G}_l$.
- The algorithm to construct the preconditioner is parallelizable.
- This algorithm is stable if $\mathbf{A}$ is SPD.

The algorithm to calculate the entries of $\mathbf{G}_l$ is:

Let $\mathbf{S_L}$ be the structure of $\mathbf{G}_l$
for $j \leftarrow 1 \ldots n$
  for $\forall (i,j) \in \mathbf{S}_l$
    solve $(\mathbf{AG}_l)_{ij} = \delta_{ij}$

Entries of $\mathbf{G}_l$ are calculated by rows. To solve $(\mathbf{AG}_l)_{ij} = \delta_{ij}$ means that, if $m = \eta((\mathbf{G}_l)_i)$ is the number of non zero entries of the column $j$ of $\mathbf{G}_l$, then we have to solve a small SPD system of size $m \times m$.

A simple way to define a structure $\mathbf{S}_l$ for $\mathbf{G}_l$ is to simply take the lower triangular part of $\mathbf{A}$.

Another way is to construct $\mathbf{S}_l$ from the structure taken from $\tilde{\mathbf{A}}$ , $\tilde{\mathbf{A}}^l$ , $\tilde{\mathbf{A}}^2$ ,...,

where $\tilde{\mathbf{A}}$ is a truncated version of $\mathbf{A}$ ,

$$\tilde{\mathbf{A}}_{ij} = \begin{cases} 1 & \text{if } i = j \text{ or } \left|\left(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\right)_{ij}\right| > \text{threshold} \\ 0 & \text{other case} \end{cases}$$

the threshold is a non negative number and

the diagonal matrix $\mathbf{D}$ is

$$\widetilde{D}_{ii} = \begin{cases} |A_{ii}| & \text{if } |A_{ii}| > 0 \\ 1 & \text{other case} \end{cases}.$$

Powers $\widetilde{\mathbf{A}}^l$ can be calculated combining rows of $\widetilde{\mathbf{A}}$. Lets denote the *k*-th row of $\widetilde{\mathbf{A}}^l$ as $\widetilde{\mathbf{A}}^l_{k:}$,

$$\widetilde{\mathbf{A}}^l_{k:} = \widetilde{\mathbf{A}}^{l-1}_{k:}\widetilde{\mathbf{A}}.$$

The structure $\mathbf{S}_l$ will be the lower triangular part of $\widetilde{\mathbf{A}}^l$. With this truncated $\widetilde{\mathbf{A}}^l$, a $\widetilde{\mathbf{G}}^l$ is calculated using the previous algorithm to create a preconditioner $\mathbf{M} = \widetilde{\mathbf{G}}_l^{\mathrm{T}}\widetilde{\mathbf{G}}_l$.

The vector $\mathbf{q}_k \leftarrow \mathbf{M}\mathbf{r}_k$ is calculated with two matrix-vector products, $\mathbf{M}\mathbf{r}_k = \widetilde{\mathbf{G}}_l^{\mathrm{T}}\left(\widetilde{\mathbf{G}}_l\mathbf{r}_k\right)$.

## 8. Numerical experiments

### 8.1. Solutions with OpenMP

First we will show results for the parallelization of solvers with OpenMP. The next example is a 2D solid deformation with 501,264 elements, 502,681 nodes. A system of equations with 1'005.362 variables is formed, the number of non zero entries are $\eta(\mathbf{K}) = 18'062,500$. $\eta(\mathbf{L}) = 111'873,237$. Tolerance used in CG methods is $\|\mathbf{r}_k\| \geq 1\times10^{-5}$.

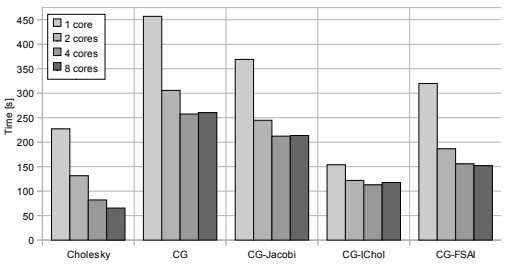



**Figure 13. 2D solid, solution times**

| Solver | core | core | core | core | Steps | Memory bytes |
|---|---|---|---|---|---|---|
| Cholesky | 227 | 131 | 82 | 65 | | 3,051,144,550 |
| CG | 457 | 306 | 258 | 260 | 9,251 | 317,929,450 |
| CG-Jacobi | 369 | 245 | 212 | 214 | 6,895 | 325,972,366 |
| CG-IChol | 154 | 122 | 113 | 118 | 1,384 | 586,380,322 |

The next example is a 3D solid model of a building that sustain deformation due to self-weight. Basement has fixed displacements.

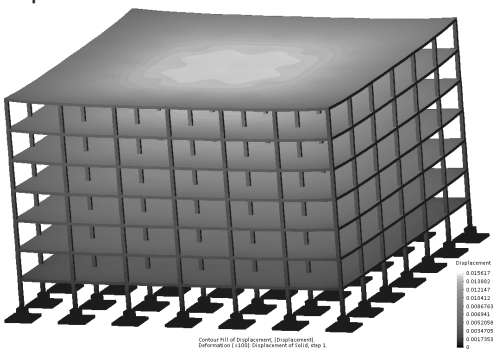


**Figure 14. 3D solid example**

The domain was discretized in 264,250 elements, 326,228 nodes, 978,684 variables, $\eta(\mathbf{K}) = 69'255,522$.

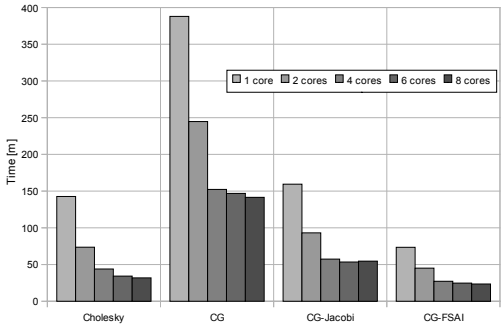



**Figure 15. 3D solid, solution times**

| Solver | 1 core [min] | 2 cores [min] | 4 cores [min] | 6 cores [min] | 8 cores [min] | Memory [bytes] |
|---|---|---|---|---|---|---|
| Cholesky | 143 | 74 | 44 | 34 | 32 | 19,864,132,056 |
| CG | 388 | 245 | 152 | 147 | 142 | 922,437,575 |
| CG-Jacobi | 160 | 93 | 57 | 54 | 55 | 923,360,936 |
| CG-FSAI | 74 | 45 | 27 | 25 | 24 | 1,440,239,572 |

In this model, conjugate gradient with incomplete Cholesky factorization failed to converge.

## 8.2. Solutions with MPI+OpenMP

Test were executed in a cluster with 14 nodes, each one with two dual core Intel Xeon E5502 (1.87GHz) processors, a total of 56 cores.

The problem tested is the same 3D solid model of a building. Using domain decomposition we tested the this problem using the following configurations:

- •14 partitions in 14 computers, using 4 cores per solver.
- •28 partitions in 14 computers, using 2 cores per solver.
- •56 partitions in 14 computers, using 1 core per solver.

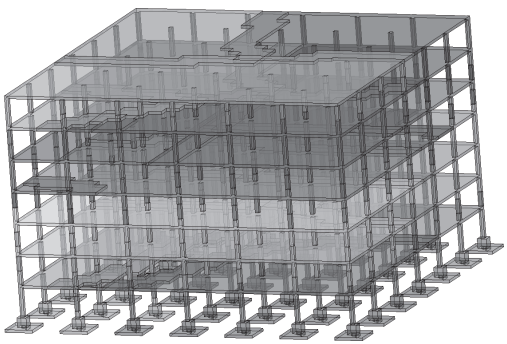


**Figure 16. 3D solid, partitioning**

Parallel alternating Schwarz method is set to iterate until a global tolerance of is $\|\mathbf{u}_i\| \leq 1 \times 10^{-4}$ reached for all partitions.

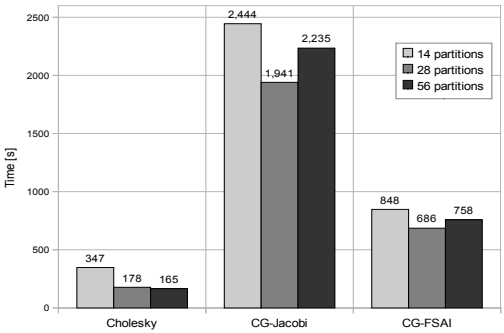



**Figure 17. Schwarz method, solution times**

| Partitions | Cholesky [sec | CG-Jacobi [s | CG-FSAI [sec |
|---|---|---|---|
| 14 | 347.3 | 2,444.2 | 847.5 |
| 28 | 177.5 | 1,940.5 | 685.9 |
| 56 | 165.3 | 2,234.8 | 757.9 |

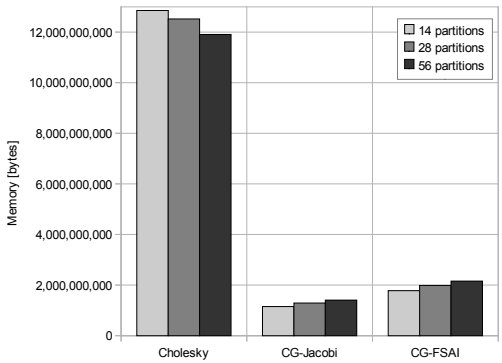



**Figure 18. Schwarz method, memory usage**

| Partitions | Cholesky [bytes] | CG-Jacobi [bytes] | CG-FSAI [bytes] |
|---|---|---|---|
| 14 | 12,853,865,804 | 1,149,968,796 | 1,779,394,516 |
| 28 | 12,520,198,517 | 1,290,499,837 | 1,985,459,829 |
| 56 | 11,906,979,912 | 1,405,361,320 | 2,156,224,760 |

## 9. Conclusions

We found that incomplete Cholesky factorization is unstable for some matrices, it is possible to stabilize the solver making the preconditioner diagonal-dominant, but we have to use a heuristic to do so.

The big issue for domain decomposition with iterative solvers is load balancing. Even thought partitioned meshes had almost the same number of nodes, the condition number of each matrix could vary a lot, making difficult to efficiently balance workload in each Schwarz iteration. The following images show this effect in several iterations. Left image correspond to the workload of the fastest solved partition (less used core), right image shows the workload of the slower solved partition (core used intensively).

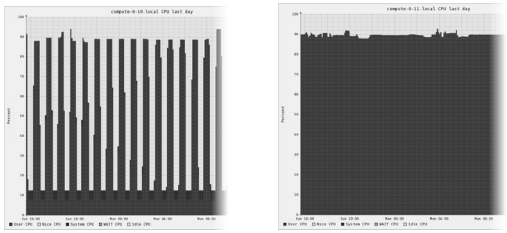


**Figure 19. Partitions, workload comparison**

It is complex to partition domains in such way that each partition take the same time to be solved. This issue is less noticeable when Cholesky solver is used.

To split the problem using domain decomposition with Cholesky works well, the fastest configuration was using one thread per solver. The obvious drawback is the memory consumption. We still can solve larger systems of equations using CG with FSAI but it will take more time.

For future work, some strategies can be taken to improve convergence:

Create from the problem mesh a coarse mesh to solve this first and have a two level solution, the coarse solution is used in the Schwarz algorithm and have a better approximation

It is possible to create the preconditioners from the overlapping of partitions to improve convergence.

Original alternating Schwarz algorithm does not solve both partitions at the same time, it alternates. Partition coloring could be used to solve in parallel all non adjacent partitions with color 1, and use these solutions as boundary conditions for all partitions with color 2, etc. Several colors could be used.

## 10. References

[1] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*. Sixth edition, 2005.

[2] A. Toselli, O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Springer, 2005.

[3] B. F. Smith, P. E. Bjorstad, W. D. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[4] T. Sterling, D. J. Becker, D. Savarese, J. E. Dorband, U. A. Ranawake, C. V. Packer. "BEOWULF: A Parallel Workstation For Scientific Computation". *Proceedings of the 24th International Conference on Parallel Processing*, 1995.

[5] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard, Version 2.1*. University of Tennessee, 2008.

[6] G. Karypis, V. Kumar. "A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs". *SIAM Journal on Scientific Computing*, Vol. 20-1, pp. 359-392, 1999.

[7] W. A. Wulf , S. A. Mckee. "Hitting the Memory Wall: Implications of the Obvious". *Computer Architecture News*, 23(1):20-24, March 1995.

[8] U. Drepper. *What Every Programmer Should Know About Memory*. Red Hat, Inc. 2007.

[9] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[10] A. George, J. W. H. Liu. *Computer solution of large sparse positive definite systems*. Prentice-Hall, 1981.

[11] A. George, J. W. H. Liu. "The evolution of the minimum degree ordering algorithm". *SIAM Review*, Vol. 31-1, pp 1-19, 1989.

[12] K. A. Gallivan, M. T. Heath, E. Ng, J. M. Ortega, B. W. Peyton, R. J. Plemmons, C. H. Romine, A. H. Sameh, R. G. Voigt. *Parallel Algorithms for Matrix Computations*. SIAM, 1990.

[13] M T. Heath, E. Ng, B. W. Peyton. "Parallel Algorithms for Sparse Linear Systems". *SIAM Review*, Vol. 33, No. 3, pp. 420-460, 1991.

[14] E. F. D'Azevedo, V. L. Eijkhout, C. H. Romine. "Conjugate Gradient Algorithms with Reduced Synchronization Overhead on Distributed Memory Multiprocessors". Lapack Working Note 56. 1993.

[15] G. H. Golub, C. F. Van Loan. *Matrix Computations*. Third edition. The Johns Hopkins University Press, 1996.

[16] E. Chow, Y. Saad. "Approximate Inverse Preconditioners via Sparse-Sparse Iterations". *SIAM Journal on Scientific Computing*. Vol. 19-3, pp. 995-1023. 1998.

[17] E. Chow. "Parallel implementation and practical use of sparse approximate inverse preconditioners with a priori sparsity patterns". *International Journal of High Performance Computing*, Vol. 15. pp 56-74, 2001.

# Parallel Processing in Networking Massively Multiuser Virtual Environments

Martha Patricia Martínez-Vargas[1],
*Víctor Manuel Larios Rosillo[2], Patrice Torguet*
*CUCEA Guadalajara University*
*University of Toulouse*
*{mmartinez[1], vmlarios[2]}@cucea.udg.mx*
*torguet@irit.fr*

## Abstract

*In this paper we proposed parallel processing in the simulation of Massively Multiuser Virtual Environments, in order to predict the interaction of 15,000 users. Parallel processing is required with the aim of simulate thousand of users, where each user is simulated by a process. Our way to better support user interactions is based on peer to peer networking technologies. Each peer or user contributes to the system with computer resources, reducing the cost to support Massively Multiuser Virtual Environments. The main original contributions of this work is the integration of a peer-to-peer VAST with a 3D interface in order to simulate connectivity among users with potential interactions. The social impact of this project is reflected in virtual environment and social network applications focusing on educational institutions or businesses, offering learning or even simulation experiences for large communities.*

***Keywords:*** *Parallel Processing, Massive Online Multiuser Virtual Environments, Mobile Social Networking, P2P Networks.*

## 1. Introduction

Massively Multiuser Virtual Environments (MMVEs) are growing almost exponentially. That is the case of some virtual environments that support millions of users like Second Life [1]. In order to support these growing applications, but with the main idea to bring advances in this research area, one question emerged. If there is a reduction in the bandwidth consumption, the cost to support these environments is also going to reduce. The current applications work with client-server architectures that bring some advantages as a central point of control, but they have to support the current user demand and have enough resources. With this constraint came the idea to work with peer to peer (P2P) architectures. In this kind of architecture any peer contributes its own resources and therefore the architecture supports MMVEs scalability.

The content of this paper is organized as follows: the state of the art of MMVEs gives a brief introduction of the current MMVEs like Second Life and World of Warcraft. The same section deals with scalability problems

in MMVEs. In the methodology section are shown the strategies we have chosen for our MMVE system. The current main contributions are the integration of a P2P framework called VAST with a 3D library called jME, implementing VAST over UDP sockets and adding dead-reckoning to VAST. Future contributions will deal with persistency either with servers or super-peers and implementing multicasting techniques in the P2P network in order to deal with crowding situations. Finally in section 4 are presented our experimental test-bed as well as some conclusions and future work.

## 2. Related work

In recent years, MMVEs have been more popular among communities of users over the Internet, and reported the support of thousands of concurrent users. Some of such applications are related to electronic commerce, the video-games industry, and virtual training for governmental or private organizations. Some examples of such applications can involve expert users of different disciplines in a collaborative environment in order to optimize human and material resources to reach organizational goals [6].

As relevant examples of MMVEs, it is important to talk about Eve Online with 60,453 concurrent users on the 6th June 2010 [2] and Second Life with 22,200,000 of concurrent users in Feb. 2011 [1].

Another relevant application in this context is World of Warcraft; which during the year 2008 reported a peak of 11.5 million active users [3] (even though each server cluster currently manages only about 5,000 peak concurrent players in the same world). In order to support MMVEs, current client-server applications have to increase their resources reaching limits that are not cost effective as well as not feasible. Under this context our proposal is based on the P2P architecture and filtering strategies as an approach to contribute in the development of MMVEs.

## 3. Proposal

In this section is presented the plan to resolve the problems defined in the previous section. The figure 1 displays the set of strategies suggested to solve the defined problem. This application can be executed on any platform, because it is developed in Java and the Virtual Machine offers heterogeneous support.

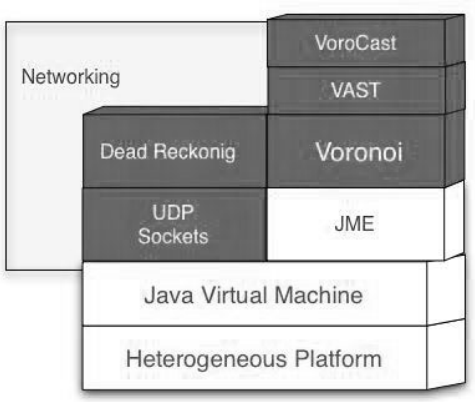



**Figure. 1 Methodology integration**

The main contribution of this work is the integration of the communication

support with 3D environments, to allow user interaction. The 3D interface is based on the jMonkey (jME) 3D graphics engine [5]. All the communication support is adapted to the 3D interface, which demands special care in the threads and security considerations.

The networking module is based on UDP[1] sockets to update the data of the 3D environment. With the goal to reduce the network traffic, the strategy is to divide the virtual world in regions to generate network traffic only in the region where the event happened and not in the whole environment. The Voronoi diagram strategy is implemented to apply filtering in the communication scheme by dividing the virtual world in cells. VAST[2] is a network library to support MMVEs with P2P architectures, and it is based on Voronoi diagrams. In order to further reduce communications we have implemented a dead-reckoning algorithm. Future strategies will take care of persistency and crowding situations. Which are common problems of P2P MMVEs. In the next subsections are more details about the currently implemented and future strategies.

### 3.1 JMonkey Engine Integration

The 3D interface was developed with the JMonkey Engine (JME). This engine enables features found in 3D video games such as scene visualization, graphic representations, and the manipulation of cameras, illumination and specific peripherals among others. In the figure 2 is shown a 3D interface with an avatar interacting in the virtual world. The avatar interactions must be transmitted through the network support module.

**Fig. 2 Virtual interface with JME[3]**

### 3.2 UDP Sockets

An UDP Socket manages all the communications. The Socket is used to update information without guaranty of arrival as in the TCP protocol. This protocol is useful because when sending a stream of new positions in a 3D environment, any lost or damaged packet is replaced by the following. If a TCP protocol were used, the communication would not be optimal on account of the retransmission of packets containing old positions. Those retransmissions, which happen in case of communication errors, drive to a bigger consumption of network bandwidth. On the contrary, the UDP transport protocol has no guaranty of packet delivery and is

1 User Datagram Protocol

2 Voronoi-based Adaptive Scalable Transfer, http://vast.sourceforge.net/

3 jME (jMonkey Engine) is an open source framework developed in Java and created to deal with high performance Open GL graphics and providing all the necessary tools for 3D games development.

more adapted to updating streams [6]. When needed some reliability is added through retransmissions and acknowledgements but this is used for a small percentage of the messages.

### 3.3 Voronoi diagrams and VAST

A Voronoi diagram is based on a mathematical algorithm that dynamically divides a 2D space into a number of cells. VAST [4] uses Voronoi diagrams in order to create neighborhoods in a set of peers. Using this library it is possible to create an Area of Interest (AOI) around a peer and manage a set of neighboring peers. This strategy theoretically reduces the communications, but it was not implemented in a real networked virtual environment. Our challenge was to implement VAST in this kind of application.

VAST provides simple functions: a peer may join the P2P network, and define the radius of its AOI in the virtual environment. VAST will then report new peers in this AOI (using mutual notification i.e. peers notify each other about new peers to connect to), leaving peers and peers updates. More details about VAST can be found in [4].

### 3.4 Dead Reckoning

Dead reckoning is a technique that allows predicting and estimating the current state of a virtual object in order to reduce the data to send. With this technique we are able to make a prediction of the trajectory of the object. The messages are only sent when there is a change in the trajectory that cannot be predicted [6]. Adding dead reckoning to VAST wasn't straightforward because the library uses the positions of avatars in order to manage dynamically the connections in the P2P network. Only the peers that do not participate in mutual notifications (i.e. those that are well inside the AOI and not near its perimeter) are updated using dead reckoning. Mutual notifications peers (called boundary neighbors by VAST authors) receive accurate positions in order to be able to correctly compute the P2P network updates. In the figure 3 is shown a simulation of dead reckoning technique, in the left side is illustrated the path of one avatar and in the right side is shown the update positions in a remote node. But the send data is only sending when the object has a change in the path. With dead reckoning is possible to reduce the number of messages to send.

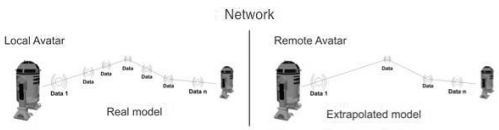



**Fig. 3 Dead Reckoning Techniques**

### 3.5 Hybrid Peer to Peer Persistency Support

VAST is an interesting way of managing distribution of updates in a P2P Virtual Environment. However it lacks a lot of features in order to manage a truly persistent Virtual World. Our current design (figure 4) involves a set of servers managing persistency for a set of meta-regions. Each server will host several databases (using a MySQL

DBMS – database management system). One database will be replicated on every server. It will contain user data (login, password, rights, last known coordinates…) as well as the coordinates and extents of every meta-regions. Another database will only be local to each server and will contain data for the virtual objects that exist in the meta-region (coordinates, URL of the 3D object file, owner ID…).

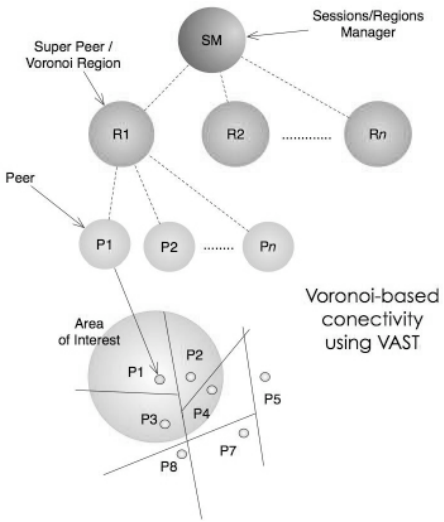



**Fig. 4 Communication Architecture with Voronoi diagrams**

When a connecting peer will want to connect to the system it will send a join request to any one of the meta-region servers (their IP address will either be user supplied or known from a previous run). This server will manage authentication and forward the join request to the server, which manages the meta-region containing the last known coordinates of the user's avatar. Then, using VAST, the join request will be propagated to the nearest peer, which will manage the insertion of the new peer in the P2P network.

### 3.6 VoroCast and FiboCast diffusion strategies

There are two forwarding AOI-cast (i.e. multicasting in AOI) designs known as VoroCast and FiboCast. Their objective is to further reduce the bandwidth consumption of VAST based P2P networks. VoroCast constructs a spanning tree over the neighbors across the AOI. VoroCast then propagates messages to all the neighbors in the same AOI [8].

FiboCast is an extension of VoroCast, and it focuses in further reducing bandwidth consumption. The messages are sent to the nearest peers using a Fibonacci sequence to control the message forwarding range [7].

## 4. Evaluation

To simulate virtual environments with thousands of users, it was required to make simulations with such quantities of peers. Intel sponsored a cluster in order to support the academy and its research. The cluster has 17 nodes, 272 cores and 1.6 terabytes of memory. With this test-bed it is possible to simulate massively multi user interactions. The figure 5 shows a simulation in the cluster with 6,000 thousand nodes. At the beginning of the interaction nodes send and received almost 9,000 thousand messages in order to know each other, but later the interactions were in a P2P way only

with the nearest users. Reducing drastically the numbers of messages.

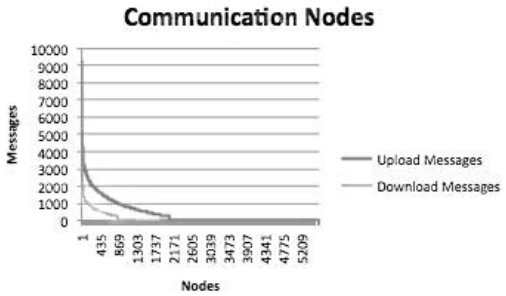



**Fig. 5 Communication cluster tests**

## 5. Conclusions and Perspectives

As we have mentioned before, this paper presented strategies for MMVEs applications, with hybrid P2P architecture and filtering techniques, with the aim to solve the problem of scalability. This brings many challenges to our research. The main contribution of this work is the simulation of thousands of users in the cluster with the purpose to predict resource consumption with grand quantities of users.

## 6. Acknowledgments

CONACYT

## 7. References

[1] M. Varvello, F. Picconi, C. Diot, and E. Biersack, “Is There Life in Second Life?”, in Context, Madrid, Spain, December 2008.

[2] CCP, “60,453 pilots: the new eve pcu record”, http://www.eveonline.com/news.asp?a=single&nid=3934&tid=1, March 2011.

[3] Blizzard Entertainment, “World of Warcraft® subscriber base reaches 11.5 million worldwide”, http://us.blizzard.com/en-us/company/press/pressreleases.html?081121, March 2011.

[4] S.Y. Hu, S.C. Chang and J.R. Jiang. “Voronoi State Management for Peer-to-Peer Massively Multiplayer Online Games”, In Proc. 4th IEEE Intl. Workshop on Networking Issues in Multimedia Entertainment (NIME 2008), March 2011.

[5] jME, “JMonkey Engine”, http://jmonkeyengine.com/, accessed in March 2011.

[6] S. Singhal and M. Zyda, “Networked Virtual Environments: Design and Implementation.”, ISBN: 0-201-32557-8, Addison-Wesley, 1999.

[7] J.R. Jiang, Y.L. Huang and S.Y. Hu, "Scalable AOI-Cast for Peer-to-Peer Networked Virtual Environments", in Proc. 28th International Conference on Distributed Computing Systems Workshops (ICDCSW) Cooperative Distributed Systems (CDS), June 2008.

[8] J. Jehn-Ruey et all.: Scalable AOI-cast for Peer-to-Peer Networked Virtual Environments, 2008.

[9] H. Shun-Yun.: Spatial Publish Subscribe, IEEE Virtual Reality workshop MMVE'09, March, 2009.

[10] D. Frey et all.: Solipsis: A Decentralized Architecture for Virtual Environments, MMVE 2008, The 1st International Workshop on Massively Multiuser Virtual Environments at IEEE Virtual Reality.

[11] Flora S. Tsai et all.; Design and development of a mobile peer-to-peer social networking application, School of Electrical and Electronic Engineering, Nanyang, Technological University, 2009.


# SCIENTIFIC VISUALIZATION

# The CinvesWall

Amilcar Meneses Viveros, Sergio V. Chapa Vergara
*Departamento de Computación, CINVESTAV-IPN*
*ameneses@computacion.cs.cinvesav.mx, schapa@cs.cinvestav.mx*

## Abstract

*The main goal of scientific visualization is to represent graphically information obtained from simulation or scientific databases. The graphic representation of data helps users to understand the phenomenon under studied. One of the problems in scientific visualization is to represent graphically large volumes of information. One way to attack this problem is to use video walls o tiled displays. The CINVESWALL is a tiled display of 12 screens controlled by a visualization cluster of low-cost based on Apple technology. Most applications running on this display clusters are created as Cocoa distributed applications. We present the projects associated with CINVESWALL, such as developing applications in Cocoa, visualization techniques, development frameworks, the management of graphical user interfaces and open problems that exist in the technological developments relating to such devices graphics.*

***Keywords:*** *Tiled Displays, visualization cluster, scientific visualization, distributed systems.*

## 1. Introduction

The video wall o tiled displays are devices for display large images or video. This devise a set of displays conformed in a two dimensional array. The idea of this construction is obtain a display device with high resolution and big size. Because the tiled display allow the visual analysis for big data volumes or show the data in human scale physical size, the video walls are useful in scientific visualization. However several problems appears when we like sending this information to displays and present this information appropriately to the user.

Display large high-resolution information is possible with computer printing devices, such as plotters and printers –this graphics device can draw 800 or 600 points per inch-. However its difficult to present this information at full resolution in outputs devices such as CRT monitors, Plasma panels, LCD panels or LED panels. The main problem is because the evolution of this kind of devices are oriented to power consumption, brightness or color definition, but not oriented to high-resolution image display –the biggest high resolution commercial displays is IBM T221, 22 inch monitor that can display 9 megapixels -.

The image resolution concept has change in time. Originally resolution means the number of points per inch. But now, it's common refer the resolution as the number of pixel on a display. The common measurement is the megapixel, that is $2^{20}$ pixels in an image. This concept is not only due to the popularity of digital images, we must also consider that the human eye has constraints to distinguish a lot of points on a surface small deployment. This yields the need for devices video with large high-resolution displays.

The popular solution to obtain devices with large high- resolution display is build tiled LCD Panels or video walls. These devices are 2D arrays of LCD displays. This solution has the advantage that the resolution increases linearly with the number of LCD panels to be added in to the array. Monitors on the market with higher resolution and large area of deployment, which is determined by the diagonal length, are 30-inch screen size with 4 megapixels resolution. There are several solutions available in the market for video walls based on flat TV screens 40 or 60 inches, however the resolution of each of these devices is just over one megapixel. This is mainly because they are designed to deliver content in a large display area without high resolution. What is required is to form tiled displays with high-resolution monitors.

Development in graphics cards allow that modern operating systems have dual-head video output, even multiple head are possible. Graphics cards can have different capacities such as: the number of devices that can handle simultaneously, and display attributes for each of its heads -resolution, video memory, graphics acceleration and optimization of video stream-.

The control of video walls can be of two ways: With multi-monitor desktop or a visualization cluster. In both cases, control of each monitor is made with graphics cards.

When using a multi-monitor desktop, there is a configuration of multiple cards in one computer and each card can control multiple heads. In this scheme, the graphics card drivers are responsible for indicating the type of information that is displayed on each monitor. Thus, the monitors can work together in an extended desktop format, or delegate some output information to a group of monitors. The end user works seamlessly with the system and should not make specific application programs to use the video wall, just the drivers provided by the manufacturer or are included in the operating system. This configuration reveals that the burden of deployment focuses on a single machine, i.e. we have a centralized system. The main constraint to this approach is scalability, since the number of monitors is dependent on the number of graphics cards that support the server. Currently the maximum is eight or sixteen monitors per server.

The visualization clusters solve the scalability problem, since the main idea is to add nodes of computers to handle a group of monitors. The main problem with this approach is the way it distributes and processes the information it displays. It requires interconnection networks with good bandwidth and low latency times. In addition, graphical applications are often built explicitly for these architectures.

The use of video walls in the laboratories of scientific computing has

become popular due to the need to display large volumes of information. The main support of these devices is in the process of data interpretation.

Distributed applications have been developed for the use of video walls in the scientific community, possibly the most significant include SAGE, ParaView and XDMX. SAGE is a solution that allows multiple applications to share the video wall as an output device. The SAGE applications using OptIPuter networks and distributed shared memory layer LambdaRAM. ParaView is a cross-platform whose function is to render and display of information. XDXM, multi-head distributed system, can enable an extended desktop between displays of different nodes in a computer cluster.

## 2. CinvesWall architecture

The CINVESWALL is part of the infrastructure of the Laboratory for Scientific Computing and Database of Computer Science Department of CINCESTAV-IPN. This laboratory is responsible for supporting several projects as GIS, problem solvers, scientific databases, cellular automata and scientific visualization, to name a few. In the search for enabled device to enable visual management of large volumes of information for scientific visualization problems of lab projects, we chose to address the technological development of this technology.

The objectives of CinvesWall are:

- Have a visualization cluster of low cost and low energy consumption.
- Take advantage of facilities and development of graphics applications distributed control of the operating system OSX.

The CinvesWall is an array of 12 Apple Cinema Display 24-inch, in a 3x4 configuration, controlled by a cluster of 12 Mac mini and one MacPro server as the master of visualization cluster. The network visualization cluster interconnect is Gigabit Ethernet. The total resolution of the display device is 27 megapixels (7680x3600).

The nodes in the cluster work with OSX version 10.6.5 (Snow Leopard). The administration of the nodes is done with the application Apple Remote Desktop.

## 3. Application software

A variety of software that has been developed to work in this type of device, whether in the form of distributed applications or development frameworks, examples are XDMX, SAGE, Aurea, Chromius, WireGL, and GCX, to name a few. However, only some of these can be adapted to the OS X platform, one of them is ParaView.

ParaView is an OpenSource program that allows you to render on a visualization cluster, but supports different modes of execution. This software use several frameworks and middleware such as VTK, MPI, Python and OpenGL. ParaView is very versatile and data management that supports several input files formats.

The research group of the Laboratory of Computational Mathematics and Data Base od Computer Department of CINVESTAV-IPN, has been given the task

of developing distributed applications for CinvesWall. However, we noticed that there is no standard architecture for developing such applications. As is evident in the following sections.

### 3.1. Java based applications

An application has been developed in CINVESTAV, particularly by students of the Masters degree program in Interactive Design and Manufacturing/Innov@prod for the Database course, was the system of "Monitoring and Refining Plant Simulation". This application was based on Java for handling user events, communication in the client process of each node and for connecting to a database. This system consists of 12 clients (one for each cluster node) who are responsible for displaying the part of the general diagram of the oil refining plant. Applications between nodes are communicating with their neighbors and they all link to a server process that is having a communication with the database that describes the processes and stores the history of the ground states of each of its components.

### 3.2. Cocoa based distributed applications

Another way to make applications for CinvesWall is to use the Cocoa technology that offers OS X for application development. This technology can easily develop object-oriented applications with good graphic interface support and opportunities to take control of distributed objects transparently.

Because most development environments with visual interfaces have the concept of event cycle, this proves to be a common element in the graphic information is displayed on the wall video.

There is a special type of objects in the Cocoa framework, are the View objects. View objects can respond to user events (mouse and keyboard events) and graphic display information in its own coordinate system, independent of the coordinate system or other View objects.

Cocoa applications use GUI View hierarchy to organize visual information. This organization is grouping objects into a hierarchy based on an inverted tree structure. Where each node is a View and nodes of their branches are subviews. This organization can handle the user events.

The Coca application architectures are based on the Model View Controller (VCM). This model handles different layers of the application. This is very attractive for the design of our distributed applications.

This architecture model is used in distributed applications for CinvesWall as follows:

- MODEL: The information being displayed on the CinvesWall is handled in the abstract, using an object model or process the data we want. For example, if you want to display video or image, then a NSImage object, an object QuickTime or Document object, can be objects of this layer.

- CONTROL: The control layer in the applications of CinvesWall, is presented as the communication between the various components of the distributed

application. Here distributed objects are used to control Cocoa application.

•VIEW: Finally, the view objects are responsible for displaying the information region in each of the nodes. The management of coordinates is done by an extrapolation of the coordinate system of the system where the origin is in the lower left corner of the monitor. Thus we can consider a meta-coordinate of the video wall that originates in the lower left point of the monitor in the lower left of the array of monitors. So the coordinate system of each of the nodes can work as if they were coordinates of View objects.

The data distribution is very sensitive in this type of architecture and depends on the nature of the issue of how data is distributed among network nodes. In some cases you can think about opening files and leave this as the responsibility of the network file system. In other cases you can think of the use of remote messages to pass data, however, in this case should be aware that remote objects are due to a TCP / IP, which can generate a communication overhead. There will be cases where you want to use a UDP protocol, as in the case of distributed video management, to name a few cases.

If the layer is delegated to control the management of events and actions to take the view enough to spend that little information to take actions.

To take advantage of the meta-coordinates in distributed applications, we use the following strategy: Each node has a copy of all information to be used. Then each node displays the information where it belongs. This task is easy to do if one thinks, for example, that an image should be displayed through the video wall, then the master node sends a copy to each slave node, this is a broadcast operation. After each node to rescale the image to the meta-coordinates and performs a translation of the coordinate system that corresponds to the arrangement of the video wall. Thus each node performs a scaling and a translation of the general information you want to deploy.

In other cases, when the image is being generated by each of the nodes should not be this technique, data is sent to those who work each node and the relevant information of neighboring nodes with which they must interact for a successful render.

Applications have been developed using this approach are:

1.-An Image Browser.
2.-Distributed graphics display.
3.-Application for visual data mining.
4.-Distributed video display (WALL-ITO).

Applications under development

1.-Distributed internet browser (BrowWall)
2.-Display linear cellular automata evolution.
3.-E-R Editor collaborative for database.

## 4. Research and development

The development of these applications has permeated various aspects noted. One of them is the way in which distributed data and the type of protocol to use.

Desirable behavior in applications that use the CinvesWall is the property of concurrency, ie the device can handle several applications running at the same time. This has led to the need for a distributed window manager to keep the hierarchy view in the style of window manager in OS X.

The visualization cluster has resulted in an Master degree thesis at the Department of Computer Science in CINVESTAV-IPN, and a Bachelor degree tesis in the Computing School of IPN.

The research on this device is in the following areas:
Software development technology based on distributed objects and view objects.

- Collaborative applications for database design.

  Visual data minig.

- Dynamic visualization for cellular automata.
- Displays various data content: video, large-scale images and HTML content. The original purpose of these kind of data are GIS systems.
- Monitoring of industrial processes.
- CAD/CAM/CAE Applications.

## 5. Conclusions

Tiled display devices are becoming popular in wide use in the scientific community. New technologies to use schemes with distributed displays allow you to have a variety of scenarios where you must run graphical applications, creating applications must be centrally run as a distributed way.

We tested a graphical interface that appears feasible to manage these devices. This interface is based on having a replica of the different contents in a window on the master node of the cluster display. Making the display of the content is completely intuitive.

To implement the content of the slave nodes is considered a meta-coordinate system whose origin is at the corner lower left node with this position, respecting the working philosophy of the system of windows OS X. Thus, when viewed the contents of the object view of the master node, view the contents of the object at each node makes a slave is a rescaling and a translation of the entire contents of the object view of the master node. Thus, changes in the contents of the view objects must be dispersed to all slave nodes, each user event on the master node, spreads to all nodes and handled following the philosophy of ls view of Cocoa objects.

## 6. References

[1] N. Tao, G.S. Schmidt, O.G. Staadt, M.A. Livingston, R. Ball, and R. May, "A Survey of Large High-Resolution Display Technologies, Techniques, and Applications"; *Proceedings of the IEEE conference on Virtual Reality*, IEEE Computer Society, Washington, DC, USA, 2006, pp. 223 – 236.

[2] Larry L. Smarr, Andrew A. Chien, Tom DeFanti, Jason Leigh, Philip M. Papadopoulos, "The OptIPuter"; *Communications of the ACM*, V. 14, no 11, Nov. 2003, pp. 58-66.

[3] Krishnaprasad Naveen, Vishwanath Venkatram, Chandrasekhar Vaidya, Schwarz Nicholas, Spale Allan, Zhang Charles, Goldman Gideon, Leigh Jason, Johnson Andrew,

"SAGE: the Scalable Adaptive Graphics Environment"; *In WACE*, 2004.

[4] Yuqun Chen, Han Chen, Douglas W. Clark, Zhiyan Liu, Grant Wallace, Kai Li, "Software Environments for Cluster-based Display Systems", *First IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2001.

[5] Laura Ramírez, Sergio V. Chapa Vergara, Amilcar Meneses Viveros; "DVO:Model for Make a Handler for a Tiled Display"; to appear in *World Congress on Engineering (WCE 2011)*, London U.K., 6-8 July, 2011.

[6] Laura Patricia Ramirez Rivera, "*Minería de datos visual sobre una pared de video*", Master Degree Thesis, Departamento de Computación, CINVESTAV-IPN, 2008.

[7] G. Alejandro Barrera Granados, Amellali L. López García, *"MANEJO DE VIDEO DISTRIBUIDO SOBRE UN VIDEO WALL"*, Trabajo Terminal, Ecuela Superior de Cómputo del IPN, Junio de 2009.

[8] Arno Puder, Kay Romer, Frank Pilhofer, "*Distributed Systems Architecture, A Middleware Approach*", Elsevier Inc., 2006.

[9] Beth Yost, Chris North, "The perceptual scalability of visualization", *IEEE Transactions On Visualization And Computer Graphics*, 12(5), September/October 2006.

[10] Y. Ng G. Humphreys, M. Houston and et al.; "Chromium: A stream processing framework for interactive rendering on clusters"; ACM TOG, 21(3), 2002.

[11] D. Germans T. van der Schaaf, L. Renambot and et al.; "Retained mode parallel rendering for scalable tiled displays"; In IPT, 2002.

[12] Ian Foster, "Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engieniering", Addison-Wesley, 1995.

# The Virtual Observatory at the University of Guanajuato: Identifying and Understanding the Physical Processes behind the Evolution and Environment of the Galaxies

Juan Pablo Torres-Papaqui1, René Alberto Ortega-Minakata1, Juan Manuel Islas-Islas1, Ilse Plauchu-Frayn2, Daniel Marcos Neri-Larios1, and Roger Coziol1
*1.- Departamento de Astronomía, Universidad de Guanajuato, 2.- Instituto de Astrofísica de Andalucía (CSIC)*
*papaqui@astro.ugto.mx*

## Abstract

*Astronomy is today one of the discipline in science which is richer in data, with an annual production of the order of tera-bytes, and with a few peta-bytes already archived. These data are now regulated by a global network under the new paradigm of the Virtual Observatory (VO). The goal of the VO is to develop and offer new tools that will facilitate the analysis of complex and heterogeneous astronomical data in order to produce new valuable information about the universe.*

*As one of the project of the VO, we are presently involve in a new study which have for main purpose identifying and understanding the physical processes behind the cosmic evolution of galaxies. Using the Sloan Digital Sky Survey, which has already collected the spectra for more than a million objects, we are creating an homogeneous catalog of 926246 galaxies, for which we have determined their nuclear activity type and identified their morphology and environment. This catalog will be available through a web server, and will be open to data exchange using a browser and a server-to-server talking pipeline using HyperText Transfer Protocol. This project involves developing new protocols and scripts, including Common Gateway Interface, Secure Socket Layer, and Active Server Pages, to increase the capacity of server to deliver their information codified in HyperText Markup Language.*

***Keywords:*** *Scientific Visualization*

## 1. Introduction

During the last two decades, the international astronomical community has witnessed an exponential grow in its capacity to produce and accumulate astronomical data. In astronomy today, information is gathered from large surveys from the ground and from space, covering virtually the entire electromagnetic spectrum, from gamma-rays to X-rays, and from the ultraviolet to optical and infrared, from the millimeter to the radio. Much of the data are made available to the community through public servers distributed among different institutions, using severally disparate formats. The quality of the data, their form and accessibility are all extremely heterogeneous, because each

particular project is also the curator of their data, which implies different choices as for the database structure, and different choices for the data formats that are usually instrument dependent. Putted bluntly, there are little efforts made to unify the panchromatic data gathering and their distribution.

As an underlying leitmotif for the Virtual Observatory (VO) project, it is believed that we can and must increase our capacity to produce meaningful scientific knowledge by improving and homogenizing the data access process, and combined the fluidity of data with powerful tools to manipulate and exploit them efficiently. The VO project is not an enterprise driven by a single institute or one particular country. It is the proper astronomical community's response to the technological challenge posed by the production of massive and complex data sets.

The effective and exhaustive scientific exploration and exploitation of large and complex data space is a highly non-trivial task, requiring a new generation of software (optimal database structures, with scalable data mining tools and interfaces), a new generation of hardware (increasing computing power, storage capacity and improving network infrastructures), and new generation of expertise (scientists with the capacity of interpreting, guiding and judging the validity of the information). The absence of these resources is a key bottleneck in data-rich astronomy: the data are there, but the means of producing valuable and new scientific knowledge from them are not. It is in this context that the department of astronomy at the University of Guanajuato is developing a new VO database that will be useful to the whole scientific community to study the relations between nuclear activity, galaxy morphology, and environments for more than 926 000 galaxies.

## 2. Sloan Digital Sky Survey

The spectroscopic data used in this study come from the seventh data release (DR7) catalog of the Sloan Digital Sky Survey (SDSS) [2]. Based on photometric images taken with the 2.5m Sloan telescope, holes are drilled on an aluminum plate (plug plate) where the lights of many different targets can be carried at once by 640 optical fibers to two slit-heads attached to a cartridge that can be quickly mounted and dismounted from the telescope. Each slit-head is mated with a spectrograph coupled with a 2 SITe/ Tektronix 2048 by 2048 pixel CCDs. This equipment produces two spectra covering the wavelengths in the blue from $\lambda$ = 3800Å to 6100Å, and in the red from 5900Å to 9100Å. Each spectrum is reduced to one dimension, calibrated in wavelengths and fluxes, and combined. All the objects in the catalog receive a classification as star or galaxy and a qualification as to the reliableness of the data.

For our study we have downloaded the spectra of 926246 SDSS objects classified as galaxies. The spectra were subsequently corrected for Galactic extinction, shifted to their rest frame (correcting for their redshifts) and re-sampled to $\Delta\lambda$ = 1Å between 3400 Å and 8900Å. Then, they were then processed using the spectral synthesis code STARLIGHT [6] which subtracts automatically a stellar population template by fitting the continuum,

leaving only the emission lines. Using IDL (Interactive Data Language) routines, we have measured automatically different important spectral attributes: emission line fluxes ($f_\lambda$), equivalent width (EW), signal-to-noise (S/N), and Full Width at Half Maximum (FWHM). From the stellar population template produced by STARLIGHT are retrieved two extra and important parameters: the stellar velocity dispersion (σ), from which we deduced the mass of the bulge of the galaxies, and the star formation history (SFH), which shows how the star formation rate of a galaxy varies over a time period covering log(t) =5.7 yrs to log(t) = 10.6 yrs.

## 3. Implementation

### 3.1. Nuclear Activity Type

To study the nuclear activity of narrow emission line galaxies (NELGs), that is star formation or accretion of matter onto a super massive black holes—the Active Galactic Nuclei (AGN) phenomenon—several spectral diagnostic diagrams were devised over the years that compare the intensity ratios of two adjacent emission lines ([4];[21]). The classical diagrams use the two Balmer lines Hα and Hβ, in combination with different forbidden lines, the nitrogen doublet [NII] at 6548 and 6584Å, the oxygen doublet [OII] at 3726 and 3729Å, the oxygen pair [OIII] at 4959 and 5007Å, and the sulfur doublet [SII] at 6717 and 6731Å. Applying different empirical separation sequences in these diagrams ([14]; [15]) allows distinguishing between two main ionization mechanisms: thermal photo-ionization by massive OB stars in star forming galaxies (SFG), and non thermal photo-ionization in AGNs.

However, for 20% of the SDSS NELGs some of these lines seem to be missing. Three cases are presenting themselves with a high frequency: either Hβ, [OIII], or both lines are missing. For these objects we have tested two new "diagnostic diagrams". The first one compares the equivalent EW of [NII]λ6584 Å with the line ratio [NII]/Hα (EW-NII diagram; [7]). The second one compares the ratio [SII]/Hα with the ratio [NII] /Hα (SII-NII diagram [8]). Like the classical diagnostic diagrams, these two new diagnostic diagrams allows to classify the galaxies as SFGs, AGNs or something in between, the Transition type Objects or TOs.

In the VO four activity types are recognized: Passive (non-emission line galaxies), SFGs, TOs and AGNs. The power of the VO resides in its dual capacity in describing precisely what the data are, and in identifying the computational facilities that can be used to transform them in valuable information. In VO, each data are associated with metadata—that is, data describing the data itself, their organization (data collections) and uses (data services). These metadata are required to manage and distribute VO user queries efficiently, so they can easily find the information of interest. In our project, the information was codified using the Metadata/UCD (Unified Content Descriptors) which follow the protocol IVOA (Standard for Unified Content Descriptors, Version 1.10 [9]). The following are all examples of legal UCD syntax for our implementation in VO of the nuclear activity study:

- **diagnostic_diagram.NIIHa_OIIIHb;SFR**
- **diagnostic_diagram.NIIHa_OIIIHb;T0**
- **diagnostic_diagram.NIIHa_OIIIHb;AGN**
- **diagnostic_diagram.OIHa_OIIIHb;SFR**
- **diagnostic_diagram.OIHa_OIIIHb;T0**
- **diagnostic_diagram.OIHa_OIIIHb;AGN**
- **diagnostic_diagram.SIIHa_OIIIHb;SFR**
- **diagnostic_diagram.SIIHa_OIIIHb;T0**
- **diagnostic_diagram.SIIHa_OIIIHb;AGN**
- **diagnostic_diagram.NIIHa_EWNII;SFR**
- **diagnostic_diagram.NIIHa_EWNII;T0**
- **diagnostic_diagram.NIIHa_EWNII;AGN**
- **diagnostic_diagram.NIIHa_SIIHa;SFR**
- **diagnostic_diagram.NIIHa_SIIHa;TO**
- **diagnostic_diagram.NIIHa_SIIHa;AGN**
- **diagnostic_diagram;Passive**

When one galaxy has more than one activity classification, due to different diagnostic diagrams, we added information about the S/N of the lines used and put more weight on the result with the highest S/N.

### 3.2. Morphology

Determining the morphology of galaxies is a difficult and time consuming task. Over the years, different methods were devised to simplify the process of retrieving the morphologies for a large sample of objects. For this study we have adopted the automatic classification method that relates the photometric colors and inverse concentration indices (how compact is a galaxy) with the classical Hubble morphological types [11][19]. The photometric colors used in this method are *u-g*, *g-r*, *r-i*, and *i-z*, as defined in the *ugriz* photometric system of SDSS (http://casjobs.sdss.org). The concentration index is defined as the ratio of the Petrosian radii [18], $R_{50}(r)/R_{90}(r)$, which compares the radius containing 50% and 90% of the total flux. A K-correction—equivalent to the redshift correction for the photometry—was applied to the magnitudes using the code developed by Blanton and Roweis [5]. The galaxies in our sample were classified along the morphological index scale T. The correspondence between T and the Hubble type is presented in Table~1.

| E | E/S0 | S0 | S0/Sa | Sa | Sa/Sb | Sb |
|---|---|---|---|---|---|---|
| 0 | 0.5 | 1 | 1.5 | 2 | 2.5 | 3 |

| Sb/Sc | Sc | Sc/Sd | Sd | Sd/Sm | Im |
|---|---|---|---|---|---|
| 3.5 | 4 | 4.5 | 5 | 5.5 | 6 |

**Table 1. Correspondence between Hubble type and morphological index T**

As an independent test we have also determined by eye the morphologies on a smallest sub-sample of NELGs: the 4240 galaxies where both emission lines are missing.

This was done using the ***chart tool*** service offered by SDSS DR7. Each galaxy observed by eye was given a morphological Hubble type which was then adapted to correspond to one of the 6 integer morphological classes used in our automatic classification. Galaxies for which no identification by eye was possible were placed in an extra class 7.

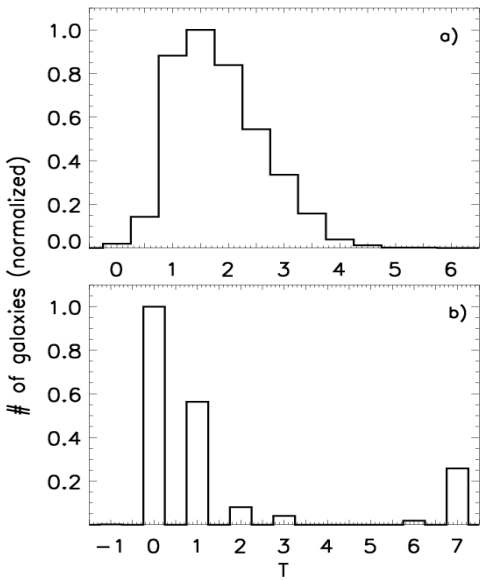



**Figure 1. Distributions of the morphologies of NELGs with both lines missing as determined a) based on their colors and compactness, and b) based on eye inspection**

In Figure 1, comparing the morphology distributions obtained using the two different methods; we observe that the visual classification favors slightly earlier types than the automatic classification. However, in both cases the total distributions are generally consistent, suggesting that about 80% of the galaxies are early type (T < 2).

The following are all examples of legal UCD syntax for our implementation in VO of the morphology study:

- **morphology_parametric;E**
- **morphology_parametric;E/S0**
- **morphology_parametric;S0**
- **morphology_parametric;S0/Sa**
- **morphology_parametric;Sa**
- **morphology_parametric;Sa/Sb**
- **morphology_parametric;Sb**
- **morphology_parametric;Sb/Sc**
- **morphology_parametric;Sc**
- **morphology_parametric;Sc/Sd**
- **morphology_parametric;Sd**
- **morphology_parametric;Sd/Sm**
- **morphology_parametric;Im**

### 3.3. Environment

Determining the density environment of galaxies, from rich to poor clusters, from compact to loose groups, from pairs to isolated, is even a much harder task than determining their morphologies. To simplify this process, we have cross identified the galaxies by names using two different catalogs of galaxy associations: one for rich clusters [3] and one more general catalog for clusters and groups [20]. For comparison, we have also determined the density environment of the 4240 galaxies for which both emission lines are missing by doing a visual inspection of the SDSS photometric images, each covering a region of 1 Mega-parsec (1parsec = $3.0857 \times 10^{16}$m) in radius

around the galaxy.

Our visual inspection reveals that the majority (65%) of the galaxies are obviously located in some kind of larger scale structure, similar either to poor clusters or groups. They are usually one of the most luminous members (BCM). Only about 1% could be a cD galaxy. A 25% are part of a large structure, not necessarily a BCM, but more like a member at the periphery. The remaining 9% are relatively isolated, although 50% of these galaxies could be in loose groups.

Comparison with Abell catalog [3] suggests that only 14% of the galaxies with both lines missing are in rich clusters. Comparison with a more general group catalog [20] suggests that 11% are in rich clusters and 35% more are in poor groups.

Our comparison shows how difficult it is in general to determine the environment of galaxies using automatic algorithms with SDSS data. For example, the more general catalog extracted groups of galaxies from the SDSS DR7, limiting the search by flux by choosing the Petrosian magnitudes between $r = 12.5$ and $r = 17.77$, and adopting an upper redshift limit of $z = 0.2$. The extraction algorithm used was a modification of the popular friends-of-friends method. Our comparison suggests that such technique fails in 20% of the cases, due to the lack of spectral information of nearby companions (in SDSS the fibers must be separated by 55''), and possibly could reach an uncertainty as high as 50% when the density of the large scale structures is low.

The environment of galaxies is one important factor affecting their formation and evolution. Unfortunately, the present automatic methods available to retrieve this information from large samples like SDSS are badly failing. Recognizing this fact, our group is currently working on developing new algorithms that would be applied directly on images to automatically retrieve the galaxy density.

The following are all examples of legal UCD syntax for our implementation in VO of the environment study:

- **environment_parameter;cluster.rich**
- **environment_parameter;cluster**
- **environment_parameter;compact.group**
- **environment_parameter;hickson.group**
- **environment_parameter;isolated**

## 4. VOTable

VOTable is an XML (extensible Markup Language) format defined for the exchange of tabular data in the context of the Virtual Observatory [17]. Within this context, a table is an unordered set of rows, with a uniform structure specified by the table description (the metadata table). Each row in a table is a sequence of cells. Each cell contains either a primitive data type, or an array of them.

VOTable is designed as a flexible storage and exchange format for tabular data, with particular emphasis on astronomical data. It derived from the Astrores format [1], which is modeled on the FITS Table format [10]. VOTable is designed to be close to the FITS Binary Table format, which makes it highly machine transferable.

The interoperability is encouraged through the use of standards in XML, which

allow applications to easily validate an input document, and facilitate their transformations through XSLT (eXtensible Style Language Transformation).

XML was derived from SGML (Standard Generalized Markup Language), a standard used in general for many years in the publishing industry and for technical documentation. Basically, XML consists of elements with payloads—an element consists of a start tag (the part in angle brackets), the payload, and an end tag (with angle brackets and a slash). Each element can contain other elements, or can bear attributes (keyword-value combinations).

### 4.1. Data Model

In this section we define the data model of a VOTable, and in the next sections its syntax when expressed as XML.

The data model of VOTable is described in Table 2:

| | | |
|---|---|---|
| **VOTable** | = | hierarchy of **Metadata** + associated **TableData**, arranged as a set of **Tables** |
| **Metadata** | = | **Parameters** + **Infos** + **Descriptions** + **Links** + **Fields** + **Groups** |
| **Table** | = | list of **Fields** + **TableData** |
| **TableData** | = | stream of **Rows** |
| **Row** | = | list of **Cells** |
| **Cell** | = | or variable-length list of **Primitives**<br><br>or multidimensional array of **Primitives** |
| **Primitive** | = | integer, character, float, floatComplex, etc. |

**Table 2. VOTable data model**

### 4.2. Our Data Model

Here we show four examples of VOTables for one galaxy in our project: Example 1: for parameters taken from the STARLIGHT code; Example 2: activity type; Example 3: morphology type; Example 4: galactic environment.

**Example 1:**

```
<?xml version="1.0"?>
<VOTABLE version="1.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
  xmlns:stc="http://www.ivoa.net/xml/STC/v1.30" >
<RESOURCE name = "NarrowEmissionLinesGalaxies">
  <TABLE name="results">
    <DESCRIPTION> Stellar Velocities Dispersions </DESCRIPTION>
                    <GROUP ID="J2000" utype="stc:AstroCoords">
 <PARAM datatype = "char" arraysize = "*" ucd = "pos.frame" name = "cooframe" utype = "stc:AstroCoords.coord_system_id" value = "UTC-ICRS-TOPO" />
      <FIELDref ref="col1"/>
      <FIELDref ref="col2"/>
    </GROUP>
<PARAM name = "Telescope SDSS" datatype = "float" ucd = "phys.size;instr.tel" unit = "m" value="3.6"/>
<FIELD name = "RA" ID = "col1" ucd = "pos.eq.ra;meta.main" ref="J2000" utype = "stc:AstroCoords.Position2D.Value2.C1"
```

```
datatype = “float” width = “6” precision = “2” unit = “deg”/>
    <FIELD name = “Dec” ID = “col2” ucd = “pos.eq.dec;meta.main” ref = “J2000” utype = “stc:AstroCoords.Position2D.Value2.C2” datatype = “float” width = “6” precision = “2” unit = “deg”/>
    <FIELD name = “Name” ID = “col3” ucd = a”meta.id;meta.main” datatype = “char” arraysize = “8*”/>
    <FIELD name = “Stellar Velocity Dispersion” ID = “col4” ucd = “stellar.Velocity” datatype = “int” width = “5” unit = “km/s”/>
    <FIELD name = “error_stellar.Velocity” ID = “col5” ucd = “stat.error;stellar.Velocity” datatype = “int” width = “3” unit = “km/s”/>
    <FIELD name = “R” ID = “col6” ucd = “pos.distance;pos.heliocentric” datatype = “float” width = “4” precision = “1” uni t= “Mpc”>
    <DESCRIPTION>Stellar Velocity Dispersion of Galaxy, assuming H=75km/s/Mpc</DESCRIPTION>
     </FIELD>
     <DATA>
      <TABLEDATA>
      <TR>
      <TD>010.68</TD><TD>+41.27</TD>
    <TD>224</TD><TD>97</TD><TD>5</ TD>
      <TD>0.7</TD>
      </TR>
      <TR>
      <TD>287.43</TD><TD>-63.85</TD>
          <TD>344</TD><TD>89</TD><TD>6</TD>
      <TD>10.4</TD>
      </TR>
      <TR>
      <TD>023.48</TD><TD>+30.66</TD>
          <TD>398</TD><TD>82</TD><TD>3</TD>
          <TD>0.7</TD>
         </TR>
         </TABLEDATA>
       </DATA>
      </TABLE>
     </RESOURCE>
    </VOTABLE>
```

**Example 2:**

```
    <?xml version=”1.0”?>
    <VOTABLE version=”1.2”
    xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
     xmlns=”http://www.ivoa.net/xml/VOTable/v1.2”
      xmlns:stc=”http://www.ivoa.net/xml/STC/v1.30” >
    <RESOURCE name = “NarrowEmissionLinesGalaxies”>
      <TABLE name=”results”>
       <DESCRIPTION> Activity Classification </DESCRIPTION>
                    <GROUP ID=”J2000” utype=”stc:AstroCoords”>
     <PARAM datatype = “char” arraysize = “*” ucd = “pos.frame” name = “cooframe” utype = “stc:AstroCoords.coord_system_id” value = “UTC-ICRS-TOPO” />
         <FIELDref ref=”col1”/>
         <FIELDref ref=”col2”/>
       </GROUP>
    <PARAM name = “Telescope SDSS” datatype = “float” ucd = “phys.size;instr.tel” unit = “m” value=”3.6”/>
    <FIELD name = “RA” ID = “col1” ucd = “pos.eq.ra;meta.main” ref=”J2000” utype = “stc:AstroCoords.Position2D.Value2.C1”
```

```
datatype = "float" width = "6" precision = "2"
unit = "deg"/>
    <FIELD name = "Dec"  ID = "col2" ucd =
"pos.eq.dec;meta.main" ref = "J2000" utype
=        "stc:AstroCoords.Position2D.Value2.C2"
datatype = "float" width = "6" precision = "2"
unit = "deg"/>
    <FIELD name = "Name" ID = "col3" ucd
= a"meta.id;meta.main" datatype = "char"
arraysize = "8*"/>
    <FIELD name = "Activity Classification" ID
= "col4" ucd = "diagnostic_diagram.NIIHa_
OIIIHb" datatype = "string" width = "5"/>
    <DESCRIPTION>Activity Classification, using
[NII]/Ha versus [OIII]/Hb </DESCRIPTION>
        </FIELD>
        <DATA>
         <TABLEDATA>
         <TR>
         <TD>010.68</TD><TD>+41.27</TD>
         <TD>SFG</TD>
         </TR>
         <TR>
         <TD>287.43</TD><TD>-63.85</TD>
         <TD>AGN</TD>
         </TR>
         <TR>
          <TD>023.48</TD><TD>+30.66</TD>
          <TD>TO</TD>
         </TR>
         </TABLEDATA>
        </DATA>
       </TABLE>
     </RESOURCE>
    </VOTABLE>
```

**Example 3:**

```
    <?xml version="1.0"?>
    <VOTABLE version="1.2"
    xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
     xmlns="http://www.ivoa.net/xml/VOTable/
v1.2"
      xmlns:stc="http://www.ivoa.net/xml/STC/
v1.30" >
    <RESOURCE              name              =
"NarrowEmissionLinesGalaxies">
       <TABLE name = "results">
        <DESCRIPTION> Parametric Morphology
</DESCRIPTION>
                        <GROUP      ID="J2000"
utype="stc:AstroCoords">
     <PARAM datatype = "char" arraysize = "*"
ucd = "pos.frame" name = "cooframe" utype
= "stc:AstroCoords.coord_system_id" value =
"UTC-ICRS-TOPO" />
          <FIELDref ref="col1"/>
          <FIELDref ref="col2"/>
         </GROUP>
    <PARAM name = "Telescope SDSS" datatype
= "float" ucd = "phys.size;instr.tel" unit = "m"
value="3.6"/>
    <FIELD name = "RA"  ID = "col1" ucd =
"pos.eq.ra;meta.main"  ref="J2000"  utype
=        "stc:AstroCoords.Position2D.Value2.C1"
datatype = "float" width = "6" precision = "2"
unit = "deg"/>
    <FIELD name = "Dec"  ID = "col2" ucd =
"pos.eq.dec;meta.main" ref = "J2000" utype
=        "stc:AstroCoords.Position2D.Value2.C2"
datatype = "float" width = "6" precision = "2"
unit = "deg"/>
    <FIELD name = "Name" ID = "col3" ucd
= a"meta.id;meta.main" datatype = "char"
```

```
arraysize = "8*"/>
    <FIELD name = "Parametric Morphology" ID = "col4" ucd = "morphology_parametric" datatype = "string" width = "5"/>
    <DESCRIPTION> Parametric Morphology </DESCRIPTION>
        </FIELD>
        <DATA>
         <TABLEDATA>
         <TR>
         <TD>010.68</TD><TD>+41.27</TD>
         <TD>E</TD>
         </TR>
         <TR>
         <TD>287.43</TD><TD>-63.85</TD>
         <TD>E/S0</TD>
         </TR>
         <TR>
         <TD>023.48</TD><TD>+30.66</TD>
         <TD>S0</TD>
         </TR>
         </TABLEDATA>
        </DATA>
      </TABLE>
     </RESOURCE>
    </VOTABLE>
```

**Example 4:**

```
    <?xml version="1.0"?>
    <VOTABLE version="1.2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
      xmlns:stc="http://www.ivoa.net/xml/STC/v1.30" >
    <RESOURCE name = "NarrowEmissionLinesGalaxies">
      <TABLE name="results">
            <DESCRIPTION> Environment </DESCRIPTION>
                  <GROUP ID="J2000" utype="stc:AstroCoords">
     <PARAM datatype = "char" arraysize = "*" ucd = "pos.frame" name = "cooframe" utype = "stc:AstroCoords.coord_system_id" value = "UTC-ICRS-TOPO" />
         <FIELDref ref="col1"/>
         <FIELDref ref="col2"/>
        </GROUP>
    <PARAM name = "Telescope SDSS" datatype = "float" ucd = "phys.size;instr.tel" unit = "m" value="3.6"/>
    <FIELD name = "RA" ID = "col1" ucd = "pos.eq.ra;meta.main" ref="J2000" utype = "stc:AstroCoords.Position2D.Value2.C1" datatype = "float" width = "6" precision = "2" unit = "deg"/>
    <FIELD name = "Dec" ID = "col2" ucd = "pos.eq.dec;meta.main" ref = "J2000" utype = "stc:AstroCoords.Position2D.Value2.C2" datatype = "float" width = "6" precision = "2" unit = "deg"/>
    <FIELD name = "Name" ID = "col3" ucd = a"meta.id;meta.main" datatype = "char" arraysize = "8*"/>
    <FIELD name = "Environment" ID = "col4" ucd = "environment_parameter" datatype = "string" width = "5"/>
    <DESCRIPTION> Environment Parameter </DESCRIPTION>
        </FIELD>
        <DATA>
         <TABLEDATA>
         <TR>
         <TD>010.68</TD><TD>+41.27</TD>
         <TD>Cluster</TD>
```

```
        </TR>
        <TR>
        <TD>287.43</TD><TD>-63.85</TD>
        <TD>Compact Group</TD>
        </TR>
        <TR>
        <TD>023.48</TD><TD>+30.66</TD>
        <TD>Isolated</TD>
        </TR>
        </TABLEDATA>
       </DATA>
      </TABLE>
     </RESOURCE>
</VOTABLE>
```

## 5. The Virtual Observatory at the University of Guanajuato

For this project we use a Server PowerEdge R510 from DELL, which is equipped with a processor Intel® Xeon® X5660, 2.8Ghz, with a 12M Cache, 1333MHz Max Mem. The server has an additional processor Intel® Xeon® X5660, 2.8Ghz, 12M Cache, Turbo, HT, 1333MHz Max Mem. 32GB Memory (8x4GB), 1333MHz. SAS 6/iR integrated with PERC,SAS 6/iR Wire, 4x1TB 7.2K RPM Near-Line SAS 6Gbps 3.5in. and running on Linux operative system Open SuSE Enterprise 11.

Also installed is the Apache like HTTP Server (http://httpd.apache.org/). The goal of our project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. The advantage of Apache Software Foundation is that it takes a very active stance in eliminating security problems and denial of service attacks.

We employ MySQL, the open source database, because of its high performance, high reliability, and ease of use. It is also the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, PHP / Perl / Python).

We also use a PHP like connection between MySQL and the HTTP server, the Simple Cone Search Version 1.03. This is a simple query protocol for retrieving records from a catalog of astronomical sources. The query furnishes a sky position and an angular distance, defining a cone on the sky. The response returns a formatted VoTable, which contains a list of astronomical sources from the catalog whose positions lie within the cone. Our version is essentially a transcription of the original Cone Search specification into the IVOA standardization process. Cone Search represents the first and arguably the most successful "standard protocol" developed within the Virtual Observatory movement.

## 6. Conclusions

The information produced by our study is contained in database catalogues using MySQL. This information will be made available through a web server, and will be open to data exchange using a browser on a server-to-server basis using HTTP. Our project involves developing new protocols and scripts, including CGI, SSL, and ASP, to increase the capacity of our server to deliver its information codified in HTML into what we call the Virtual Observatory at the University of Guanajuato.

The VO is rapidly becoming a reality. The rapid increase in data volumes and complexity, in parallel with the growth

in computational power and algorithmic knowledge; have made the VO a necessity and possibility. We have described some of the ongoing projects to implement databases and develop general-purpose computational algorithms and other VO-enabling technologies. A common theme among all of these developments is the dire need for powerful computational resources (CPUs, storage and network), optimal software, and greater expertise in design and implementation. The international nature of astronomy (also the use of the WEB) implies that everyone can contribute and benefit to this enterprise. We have provided a basic, but certainly not exhaustive, outline of the components of the VO, and described the specific contributions our group at the University of Guanajuato has made. Our growing partnerships in large telescopes and unfettered access to public datasets demand that we develop our own tools and expertise to leverage these investments and strengthen our scientific output.

## 7. Acknowledgments

T-P acknowledges PROMEP for support grants 103.5-10-4684. Funding for the SDSS and SDSS-II has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, the U.S. Department of Energy, the National Aeronautics and Space Administration, the Japanese Monbukagakusho, the Max Planck Society and the Higher Education Funding Council for England. The SDSS web site is http://www.sdss.org/.

The SDSS is managed by the Astrophysical Research Consortium for the following Participating Institutions: the American Museum of Natural History, the Astrophysical Institute Potsdam, the University of Basel, the University of Cambridge, the Case Western Reserve University, the University of Chicago, the Drexel University, the Fermilab, the Institute for Advanced Study, the Japan Participation Group, the Johns Hopkins University, the Joint Institute for Nuclear Astrophysics, the Kavli Institute for Particle Astrophysics and Cosmology, the Korean Scientist Group, the Chinese Academy of Sciences (LAMOST), Los Alamos National Laboratory, the Max-Planck-Institute for Astronomy (MPIA), the Max Planck-Institute for Astrophysics (MPA), New Mexico State University, Ohio State University, University of Pittsburgh, University of Portsmouth, Princeton University, the United States Naval Observatory and the University of Washington.

## 8. References

[1] Accomazzi, et al, *Describing Astronomical Catalogues and Query Results with XML*. http://cds.u-strasbg.fr/doc/astrores.htx

[2] Abazajian, K.N., et al., *The Seventh Data Release of the Sloan Digital Sky Survey*, 2009, Astrophysical Journal Supplement, 182, 543

[3] Andernach, H., Tago, E., Einasto, M., Einasto, J., & Jaaniste, J., *Redshifts and Distribution of ACO Clusters of Galaxies*, 2005, Astronomical Society of the Pacific Conference Series, 329, 283.

[4] Baldwin, J.A., Phillips, M.M., & Terlevich, R., *Classification parameters for the emission-line spectra of extragalactic objects*, 1981, Astronomical Society of the Pacific, 93, 5

[5] Blanton, M. R. & Roweis, S., *K-Corrections and Filter Transformations in the Ultraviolet, Optical, and Near-Infrared*, 2007, Astronomical Journal, 133, 734

[6] Cid Fernandes, R., et al., *Semi-empirical analysis of Sloan Digital Sky Survey galaxies - I. Spectral synthesis method*, 2005, Monthly Notice of the Royal Astronomical Society, 358, 363

[7] Coziol, R., Ribeiro, A. L. B., de Carvalho, R. R., & Capelato, H. V., *The Nature of the Activity in Hickson Compact Groups of Galaxies*, 1998, Astrophysical Journal, 493, 563

[8] Coziol, R., Carlos Reyes, R.E., Considère, S., Davoust, E. & Contini, T., *The abundance of nitrogen in starburst nucleus galaxies*, 1999, Astronomy & Astrophysics, 345, 733

[9] Derriere, S., et al. 2005 (http://www.ivoa.net/twiki/bin/view/IVOA/IvoaUCD).

[10] FITS: *Flexible Image Transport Specification, specifically the Binary Tables Extension*. http://fits.gsfc.nasa.gov/

[11] Fukugita, M., Nakamura, O., Okamura, S., Yasuda, N., Berentine, J.C., Brinkmann, J., Gunn, J.E., Harvanek, M., Ichikawa, T., Lupton, R.H., Schneider, D.P., Strauss M.A., & York, D.G., *A Catalog of Morphologically Classified Galaxies from the Sloan Digital Sky Survey: North Equatorial Region*, 2007, Astronomical Journal, 134, 597

[12] Hickson, P., *Systematic properties of compact groups of galaxies*, 1982, Astrophysical Journal, 255, 382

[13] Karachentseva, V. E., Mitronova, S. N., Melnyk, O. V. & Karachentsev, I. D., *Catalog of isolated galaxies selected from the 2MASS survey*, 2010, Astrophysical Bulletin, 65, 1

[14] Kauffmann, G., Heckman, T.M., Tremonti, C., Brinchmann, J., Charlot, S., White, S.D.M., Ridgway, S.E., Brinkmann, J., Fukugita, M., Hall, P.B., Ivezic, Z., Richards, G.T., & Schneider, D.P., *The host galaxies of active galactic nuclei*, 2003, Monthly Notice of the Royal Astronomical Society, 346, 1055

[15] Kewley, L.J., Dopita, M.A., Sutherland, R.S., Heisler, C.A., & Trevena, J., *Theoretical Modeling of Starburst Galaxies*, 2001, Astrophysical Journal, 556, 121

[16] McConnachie, A. W., Patton, D. R., Ellison, S. L. & Simard, L., *Compact groups in theory and practice - III. Compact groups of galaxies in the Sixth Data Release of the Sloan Digital Sky Survey*, 2009, Monthly Notice of the Royal Astronomical Society, 395, 255

[17] Ochsenbein Francois & Williams Roy, 2009 (http://www.ivoa.net/Documents/VOTable/).

[18] Petrosian, V., *Surface brightness and evolution of galaxies*, 1976, Astrophysical Journal, 209, 1

[19] Shimasaku, K., Fukugita, M., Doi, M., Hamabe, M., Ichikawa, T., Okamura, S., Sekiguchi, M., Yasuda, N., Brinkmann, J., Csabai, I., Ichikawa, S-I., Ivezic, Z., Kunszt, P.Z., Schneider, D.P., Szokoly, G.P., Watanabe, M., & York, D.G., *Statistical Properties of Bright Galaxies in the Sloan Digital Sky Survey Photometric System*, 2001, Astronomical Journal, 122, 1238

[20] Tago, E., Saar, E., Tempel, E., Einasto, J., Einasto, M., Nurmi, P. & Heinämäki, P., *Groups of galaxies in the SDSS Data Release 7. Flux- and volume-limited samples*, 2010, Astronomy & Astrophysics, 514, 102

[21] Veilleux, S., & Osterbrock, D.E., *Spectral classification of emission-line galaxies*, 1987, Astrophysical Journal Supplement, 63, 295


# APPENDIX 1
# Conference Keynote Speakears

# Transforming Research through High Performance Computing

**Dr. Moisés Torres Martínez**

*Coordinator of the design of systems and technologies unit under the General Coordination of Information Technologies (CGTI) at the University of Guadalajara, México*

## Abstract

Supercomputing in México has had a steady growth in the past two decades and continues to be of importance to the advancement of science and technology in the nation. The objective of founding the "International supercomputing Conference in México" (ISUM) was to foster the uses and research of HPC in the country through a coordinated effort from Universities across the nation working in this area of specialty. Through this coordinated effort we expect that the country grows even more in this area than it has had in the last two decades. This book presentation will provide a brief pespective of the state, challenges, and future directions of supercomputing in Mexico and suggest eight recommendations identified as important to the growth of supercomputing in the country and foster its global competitiveness in research and development. In addition,it will provide a brief overview of the work that is published in the first book ISUM Conference Proceedings: Transforming Research through High Performance Computing.

# Enabling Exascale Computing through the ParalleX Execution Model

**Thomas Sterling, Ph. D.**

*Arnaud and Edwards Professor of Compute Science at the Louisiana State University Department of Computer Science.*

## Abstract

HPC is entering a new phase in system structure and operation dri-ven by a combination of technology and architecture trends as early research to achieve Exascale capability is initiated. Perhaps foremost are the constraints of power and complexity that as a result of the flat-lining of clock rates relies on multicore as the primary means by which performance gain is being achieved with Moore's Law. Indeed, for all intense and purposes, "multicore" is the new "Moore's Law" with steady increases in the number of cores per socket. Added to this is the highly multithreaded GPU technology moving HPC into the heterogeneous modality for additional performance gain. These dramatic changes in system architecture are forcing new methods of use including programming and system management.

Historically HPC has experienced five previous phase changes involvingtechnology, architecture, and programming models. The current phase of two decades is exemplified by the communicating sequential model of computation replacing previous vector and SIMD models. HPC is now faced with the need for new effective means of sustaining performance growth with technology through rapid expansion of multicore with anticipated structures of hundreds of millions of cores by the end of this decade delivering Exaflops performance. This pre-sentation will discuss the driving trends and issues of the new phase change in HPC and will discuss the ParalleX execution model that is serving as a pathfinding framework for exploring an innovative synthesis of semantic constructs and mechanisms that may serve as a foundation for computational systems and techniques in the Exascale era. This talk will use a kernel application code for numerical relati-vity via adaptive mesh refinement to demonstrate the effectiveness of the ParalleX model through the use of the HPX runtime software system library.

# Supercomputing Centers in the Era of Big Data

**Michael Norman, Ph. D.**
*Director of San Diego Supercomputing Center, Distinguish Professor of Physics*
*University of California, San Diego*

## Abstract

Supercomputers are prodigious producers of numerical data that need to be stored, archived, analyzed, visualized, and published before its full scientific potential is realized. Historically, the data-handling infrastructure at a supercomputer center was fairly primitive and of secondary impotance to the supercomputer itself. The sheer volume of data being produced by high-end machines today changes this paradigm, and requires a re-thinking of what kinds of resources a balanced, "data-intensive" supercomputer center should look like. For the past several years the San Diego Supercomputer Center (SDSC) has been struggling with this new reality.

In this talk, I will discuss how scientific computing is becoming more data-intensive, and what that means for users and administrators of supercomputing centers. I will describe several initiatives at SDSC to cope with the data deluge including our data-intensive supercomputer architecture called Gordon, and an integrated high performance storage environment called Data Oasis.

# Toward Exaflop Supercomputers

**Mateo Valero, Ph. D.**
*Director of the Barcelona Supercomputing Center*

## Abstract

Supercomputers are prodigious producers of numerical data that need to be stored, archived, analyzed, visualized, and published before its full scientific potential is realized. Historically, the data-handling infrastructure at a supercomputer center was fairly primitive and of secondary impotance to the supercomputer itself. The sheer volume of data being produced by high-end machines today changes this paradigm, and requires a re-thinking of what kinds of resources a balanced, "data-intensive" supercomputer center should look like. For the past several years the San Diego Supercomputer Center (SDSC) has been struggling with this new reality.

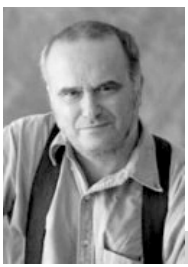
In this talk, I will discuss how scientific computing is becoming more data-intensive, and what that means for users and administrators of supercomputing centers. I will describe several initiatives at SDSC to cope with the data deluge including our data-intensive supercomputer architecture called Gordon, and an integrated high performance storage environment called Data Oasis.

## Speeds, Feeds and Needs: an Inside Look at Architecting I/O Subsystems

**Nicholas P. Cardo, M.S.**
*Lawrence Berkeley National Laboratory, Berkeley, CA USA*

### Abstract

When designing an HPC solution, the performance of the I/O subsystem is often overlooked. The end result is often an under performing, under specified file system with high expectations. Sacrifices in I/O performance are often made inadvertently, and comes as a shock, as the reality of the choices come to life. A careful understanding of the data rates of all the devices for the storage subsystem will shed light onto expectations. This paper will show how to understand the components and apply them in the design of the I/O subsystem. This feeds into the expectations and needs of the applications to be run on the system.



## Toward Exascale

**Marc Snir, Ph. D.**
*Michael Faiman and Saburo Muroga Professor in the Department of Computer Science at the University of Illinois at Urbana-Champaign*

### Abstract

The talk will position exascale research in the context of the pending slow-down in the exponential increase in chip densities and discuss some fundamental research problems that need to be addressed in order to reach exascale performance at reasonable expense.

# Green Grid Computing: Eco friendly power-aware

**Andrei Tchernykh, Ph. D.**
*Researcher in the Computer Science Department, CICESE Research Center, Ensenada.*

## Abstract

The increasing heterogeneity and dynamism of emerging distributed computing environment systems such as Grids and Clouds imply that resource management must be able to adapt to changes in their state and requirements to meet their desired QoS constraints. As system scales and energy consumption increase, such new technologies have the power to do significant damage to our ecosystems. Traditional heuristic-based approaches to resource optimization become insufficient. Efficient eco-friendly poweraware computing resources optimization should be considered, both in terms of redu-cing the environmental impact and reducing costs. Ecologically sustainable computing is a rapidly expanding research area spanning the fields of computer science and engineering, resource optimization as well as other disciplines. It relates to power-aware and thermal-aware management of computing resource, applications of computing that can have ecological impacts, areas of power, energy, temperature, and environment related research, etc.

The eco-friendly management service minimizes energy consumption, and adapts to dynamic load characteristics to meet desired QoS constraints. It is an interesting cost-effective and more environmentally friendly alternative to other technologies.

This talk deals with new scheduling system for eco-friendly parallel job management and optimization strategies. It will briefly discuss issues that have been raised regarding these strategies. We address eco-friendly management by offering job allocation strategies, with lower energy consumption and QoS. A central goal is the development of energy-efficient scheduling solutions for low-power computing that satisfy OoS constraints. This talk will present a new scheduling technique for supporting the design and evaluation of power-aware systems in Grids.

# Aplicación de Herramientas Computacionales Eficientes de la Minería de Datos a Biosenales

**Pablo Guillén Rondón, Ph. D.**
*University of Texas at El Paso*
*Program in Computational Science*

## Resumen

Actualmente la automatización y monitorización de sistemas físicos y biomédicos genera una gran cantidad de datos los cuales deben ser almacenados y analizados mediante el uso de herramientas computacionales de alto redimiento (hardware y software). En vista de la gran cantidad de datos que pueden ser generados y que solo una parte de estos contienen información útil y valiosa, surge la necesidad del desarrollo e implantación de técnicas novedosas de la matemática -estadística- computación para la obtención de dicha información y que se enmarcan dentro de lo que hoy en día se conoce como la Minería de Datos. La Minería de Datos es un área de investigación y desarrollo que permite la obtención de información contenida en grandes conjuntos de datos y la cual es utilizada para una mejor toma de decisión, supervisión, control, diagnostico y visualización del sistema en estudio. Esta charla comienza con una introducción a la Minería de Datos y la necesidad de implantar técnicas novedosas del análisis lineal y no lineal para el procesado de series temporales, seguidamente, se presentan diferentes índices estadísticos o características (reconocimiento de patrones) para la obtención de la información inherente en los datos, y finalmente se muestran un conjunto de aplicaciones basadas en máquinas de aprendizaje las cuales permiten clasificar registros electroencefalográficos de pacientes epilépticos y sujetos sanos, así como también, la clasificación y localización funcional de la actividad neuronal en estructuras subcorticales de pacientes parkinsonianos bajo estimulación profunda cerebral.

# Grid and Cloud computing in the context of supercomputers

**Uwe Schwiegelshohn, Ph.D.**
*Robotics Research Institute, TU Dortmund University, Germany.*

## Abstract

Cloud computing provides highly scalable resources on demand and has a steadily growing user community. Grid computing enables researchers to cooperate within virtual research environments. Both approaches often use virtualization to abstract from any particular hardware while supercomputer applications are often carefully tuned to extract the best performance from the available hardware. Is it possible to combine these approaches to extend the user community of highly parallel computing?

ISUM
2011
2nd INTERNATIONAL SUPERCOMPUTING CONFERNCE IN MEXICO
CONGRESO DE SUPERCÓMPUTO

# APPENDIX II
# Abstracts

## High-Performance Computing for Scientific and Technological Applications

Ponente:
**Gerardo Zavala G**
*Universidad de Guanajuato, Departamento de Estudios Organizacionales*

Abstract:

Triggered by the necessity of rapidly executing lengthy, complex tasks unavoidable in the current research activities performed by a group of scientists at the University of Guanajuato, Mexico; in the areas of science and technology, we see ourselves compelled to design, build and operate a small supercomputer. Genomic Sequencing, Energy Building Simulation, Molecular Simulation, Structure Formation in Astrophysics, Data Processing from High Energy Physics Experiments are some of the applications we are working on. In this talk, we report the current status, expected operating performance and characteristics of the envisioned final cluster. We make public as well the desire and need of integrating our cluster to a larger grid that could allow us to perform those tasks, which given our limited capabilities, lie currently out of reach.

## Present and Future Development of Mobile Applications

Ponente:
**Ma. del Rocío Maciel Arellano**

Luis Alberto Maciel Arellano / *Universidad de Guadalajara*
Ma. del Rocío Maciel Arellano / *Centro Univ. de Ciencias Eco. Administrativas*
Giovanna E. Parra Hipólito / *Departamento de Sistema de Información*

Resumen:

El presente documento es un análisis y estudio sobre el futuro de las aplicaciones móviles en México y en el mundo, donde se presenta las experiencias obtenidas para construir aplicaciones móviles en los diversos sectores productivos y de gobierno de nuestro país. Las áreas de oportunidad son inmensas, la tecnología que provee la movilidad es madura e innovadora, y por otro lado existe un campo totalmente nuevo con diversas oportunidades para todas aquellas empresas que se dediquen a dar soluciones en este rubro dentro del mundo de las tecnologías de información en los próximos años. Está situación debe ser aprovechada por nuestras instituciones educativas para formar futuros profesionistas en el sector de la movilidad y de las aplicaciones móviles en nuestro país.

Palabras clave:
Soluciones Móviles, Aplicaciones Móviles, Movilidad.

## Devising Geographic Database (GDB) of the San Miguel River Basin for Geoscience Applications.

Ponente:
**María del Carmen Heras Sánchez**

María del Carmen Heras Sánchez, Dora Guzmán Esquer, Christopher Watts Thorp & Juan Saiz Hernández / ***Universidad de Sonora***

Abstract:

A geographic database for the San Miguel river basin is being developed by integrating data from multiple sources for analysis and graphical representation of diverse physiographic features and hydroclimate phenomena such as rainfall, temperature, soil-evaporation, and topography among others. The projected database will allow us to combine digital maps and images along with thematic information and spatially-referenced vector data. On the other hand, building a GDB with validated references further requires geographical referencing and validating processes in order to be able to accurately represent continuous data through discrete data structures that fit the mathematical models used in representing the physical phenomena at the study site. Once our georeferencing and validating methodology is thoroughly tested, it may prove useful to geoscientists performing numerous modeling and numerical analysis such as tridimensional-temporal-thematic hydroclimate modeling, spatial-temporal rainfall analysis, and hydrological modeling of distributed parameters for basins similar to the San Miguel, Sonora, river basin.

Keywords:
GDB, GIS, river basin, e-geoscience, modeling, mapping.

## Implementación de Integridad en el Smbd Data Dictionary de Progress Version 9.1d

Ponente:
**Rodrigo Villegas Téllez**

Abstract:

Databases are essentials in any and all information systems, since this is the correct structure where data resides to generate the information needed by system users. Having in mind the importance of databases, is necessary to know that databases must have certain levels of integrity which ensures that data entered into the database is accurate, valid, and consistent. Those levels of integrity are Entity Integrity, Domain Integrity, Referential Integrity and Integrity defined by Users. This article will describe the correct development of the Entity Integrity, Domain Integrity and Referential Integrity for a database built on Progress Relational DataBase Management System that contains data of the Project Management System of Irapuato's Institute of Technology. I hope this article will be useful for DBA's who develops database integrity, considering there isn´t enough information to assist the implementation of integrity on Progress RDMS.

## Towards a Social Networks-Based Architecture to Improve Internal Communication Inside Big-Size Companies and Institutions

Ponente:
**Carlos Vázquez Castañeda**

Carlos Vázquez-Castañeda and Francisco-Edgar Castillo-Barrera
*Department of Information Technologies / Universidad de Guadalajara*

Abstract:

One of the most important social phenomena of the first decade of the XXI century are the on-line social networks [1], from which Facebook has the biggest increase of users [2][3] Facebook offers an open platform for everyone that could be interested on the development of web and desktop applications [4], becoming a potential tool for technology development and scientific research [5], especially in the artificial intelligence area. In this paper we propose a Social Network Architecture based on a Facebook platform that improves the communication inside big-size companies or institutions. This system takes advantage of the Facebook's supercomputational properties to reduce the use of local servers sources during internal email sending, while email replay information is processed using Semantic techniques to analyze and resend the information to the related people involved in a specific issue.

Keywords: on-line social networks, supercomputational, semantic techniques.

## High Performance Computing Architecture for a Massive Multiplayer Online Serious Game

Ponente:
**César Alejandro García García, UdG**

César García-García, Victor Larios-Rosillo, Hervé Luga, *Universidad de Guadalajara, Centro Universitario de Ciencias Económico Administrativas / Université Toulouse-1*

Abstract:

This work presents the current development of a serious game that for its very scale requires massive amounts of data storage and processing. We will have to create massive simulations with tens of thousands of virtual characters acting in a congruent manner, which presents several challenges to standard computing platforms. Our project consists on a massively distributed training game and is part of the much bigger DVRMedia2 Framework for serious game development. The current focus of the project is to develop a distributed virtual environment where massive events can be simulated with the real-time interaction of multiple users across different geographical areas. The Current Development section reflects the actual status of the project, amongst some preliminary results obtained from applying High Performance Computing. The Future Work section presents the next steps to be taken towards completion of the project, as well as the expected results.

Keywords: High Performance Computing, Behavior Modelling, Crowd Simulation, Serious Games, Artificial Life

## Multi Agents System for Enterprise Resource Planning Selection Process Using Distributed Computing Architecture

Ponente:
**Augusto Alberto Pacheco Comer**
*Universidad de Guadalajara, Centro Universitario de Ciencias Económico Administrativas, Doctorado en Tecnologías de Información.*

Abstract:

Enterprise resource planning system is one of the informationsystems most implemented by businesses organizations. Their use can be seen at all kind of enterprises. It is one of the most important projects on business optimization than an enterprise could attempt. We review research literature regarding multi-agents systems related with enterprise resource planning systems and high performance computing. A multi agent system model proposal for enterprise resource planning selection process using distributed computing architecture is presented. The enterprise resource planning selection process is divided in six steps: Identification of needs, identification of solutions and providers, analysis and evaluation of solutions, creation of evaluation criteria and methodology, evaluation process and make the decision. The proposal model is an aid for evaluation process and has five different types of agents. A distributed computing architecture is proposed as a mean to improve the performance in the use of proposal model.

Keywords: Enterprise Resource Planning, Multi agent system.

## Architecture for Virtual Laboratory for GRID

Ponente:
**Francisco Antonio Polanco Montelongo,** *UAM*

Francisco Antonio Polanco Montelongo 1;2, Manuel Aguilar Cornejo1
*1. UAM-Iztapalapa Department of Electrical Engineering Mexico City, Mexico.*
*2. UPIITA-IPN Department of Engineering and Advanced Technologies Mexico City, Mexico*

Abstract:

In the development of distributed and parallel applications, is very important to verify and validate its funtionality. This task is very hard due to the nature of the systems. To validate this systems in real conditions its necessary to allocate resources in amount andcharacteristics equivalent to the production environment. These situation is difficult to achieve. In this work, we propose the development of a Virtual Laboratory for GRIDS, which is a system that allows the creation of a test platform using virtualization technologies. The proposed architecture is based on a multilevel scheme. At the lower level we build a virtual cluster through virtual machines and finally, the superior level constitutes a virtual grid. In order to maintain the testing environment as realistic as possible, we are using existing middleware tools (e.g. globus, gLite, OSCAR). Virtual clusters communicates using a virtual network. In addition to the evaluation and validation of distributed applications, this system allows training on configuration of clusters and Grids.

## Arquitectura Orientada a Servicios Autónomos y Descentralizados para Sistemas de Misión Crítica

Ponente:
**Pedro Josué Hernández Torres**

Luis Carlos Coronado-García, Jesús Alejandro González-Fuentes, Pedro Josué Hernández, Torres and Carlos Pérez-Leguízamo
*Banco de México*

Resumen:

La Arquitectura Orientada a Servicios (SOA) representa un nuevo modelo en la forma tradicional de diseñar sistemas, esto se debe a la dinámica con la que cambian los requerimientos y al alto grado de integración de las aplicaciones de las organizaciones alrededor del mundo. No obstante que el uso de SOA se ha incrementado, existen algunos tipos de aplicaciones que no permiten su uso. Tal es el caso de aplicaciones de misión crítica, cuyas características son máxima disponibilidad, funcionamiento ininterrumpido, alta flexibilidad, alto rendimiento, etc. Por otro lado, estas exigencias se han cubierto por los Sistemas Autónomos Descentralizados (ADS). En este artículo se presenta un novedoso enfoque de modelado de una SOA con ADS, que hemos denominado Arquitectura Orientada a Servicios Autónomos y Descentralizados. Adicionalmente se presenta la Tecnología de Sincronización y Entrega Transaccional de Bajo Acoplamiento para asegurar la consistencia de la información y la alta disponibilidad de la aplicación. Para probar la viabilidad de la propuesta se presenta un prototipo.

## Beneficios del Supercómputo en la Industria Petrólera

Ponente:
**Gabriel Ventura Suárez**

Gabriel Ventura Suárez, Rafael Gómez González, *CIDESI*

Resumen:

Se ofrece una perspectiva del uso del súper cómputo en la industria petrolera. Los esfuerzos para buscar nuevos yacimientos de petróleo, del profesional mexicano, se harán más llanos con la implementación de un centro de súper cómputo, dirigido a esa industria.

Se ejemplifica con experimentación numérica de descomposición de dominios en equipo característico de las plantas de proceso petroleras, como lo son las turbinas de vapor.

## Programación Paralela Masiva en GPUs aplicada a dinámica molecular con herramientas de desarrollo PGI

Ponentes:
**Alejandra Maqueda; Karina Cruz**
*Global Computing*

Resumen:

Las simulaciones de dinámica molecular son aptas para las arquitecturas de procesamiento paralelo masivo de los GPUs. En esta platica presentaremos el algoritmo y los resultados obtenidos de dinámica molecular usando un fluido de Lennard-Jones. Los resultados indican que el código que se ejecuta en el GPU es 30 veces más rápido que el código en serial sobre un CPU. Para obtener estos resultados utilizamos las herramientas de desarrollo PGI (CUDA C / C++).

## Gpu-Based Polygonizer Algorithm for Two Tomographic Slices

Ponente:
**Jesús Alvarez Cedillo** 1,2

Klaus Lindig-Bos 2, Juan Carlos Herrera Lozada2
*1. Instituto Politécnico Nacional,Centro de Investigación e Innovación Tecnológica,*
*2. Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en*

Abstract:

We present a polygonizer algorithm for real-time 3d for volume rendering applications using two graphics processing units with SLI. Our GPU-based method provides real-time frame rates and outperforms the CPU-based implementation. This parallel algorithm build polygons and build a full 3d model using a GPU Polygonizer is a method whereby a polygonal (i.e., parametric) approximation to the implicit surface is created from the implicit surface function. This allows the surface to be rendered with conventional triangles; it also permits non-imaging operations, such as positioning an object on the surface. Polygonization consists of two principal steps. First, space is partitioned into adjacent cells at whose corners the implicit surface function is evaluated; negative values are considered inside the surface, positive values outside. Second, within each cell, the intersections of cell edges with the implicit surface are connected to form one or more polygons. In this paper we showed a Gpu-based polygonizer algorithm for general purposes. The implementation is relatively simple but some of this simplicity derives from the use of geometric properties of 3d model.

## Stochastic Scheduler for General Purpose Clusters

Ponente:
**Ismael Farfán Estrada**

Farfán Estrada Ismael, Dr. Luna García René
*Centro de Investigación en Computación*
*Instituto Politécnico Nacional*

Abstract:

This work presents a proposal for a scheduler which dynamically modifies the user run-time estimates, based in the user's statistical behavior, in a way that it's possible to make a load-balancing of the jobs such that there exists a possibility P(s) that the schedule s is respected so that no need to use back-filling algorithms to fill the gaps resulting due to bad estimations is needed; thereof maximizing the utilization of the cluster without wasting computing time in a schedule which won't be honored.

## Management and Monitoring of Large Datasets on Distributed Computing Systems for the IceCube Neutrino Observatory

Ponente:
**Juan Carlos Díaz Velez Berghouse**

Juan Carlos Díaz, Velez Berghouse
*University of Wisconsin / Madison for the IceCube Collaboration*

Abstract:

IceCube is a one gigaton neutrino detector designed to detect high energy cosmic neutrinos. It is currently in its final phase of construction at the geographic South Pole. Simulation and data processing for IceCube require a significant amount of computational power. We will describe the design and functionality of IceProd, a middleware system based on Python, XMLRPC and GridFTP driven by a central database in order to coordinate, administer and drive production of simulations and processing of data produced by the IceCube detector upon arrival in the northern hemisphere. IceProd run as a separate layer on top of other middleware and can take advantage of a variety of computing resources including grids and batch systems such as GLite, Condor, NorduGrid, PBS and SGE. This is accomplished by a set of dedicated daemons which process job submission in a coordinated fashion through the use of middleware plugins that serve to abstract the details of job submission. I will describe several aspects of IceProd's design including security, data integrity, scalability and throughput as well as the various challenges in each of these topics.

## Implementación de Herramientas de Seguridad Utilizando Técnicas de Virtualización en Grids

Ponente:
**Chadwick Carreto Arellano**

Ciro D. León Hernández, Cirilo G. León Vega, Chadwick Carreto
*CENAC – ESIME – ESCOM - Instituto Politécnico Nacional.*

Resumen:

Actualmente los ataques a redes de comunicaciones son más frecuentes, los intrusos utilizan técnicas que burlan rápidamente la seguridad de los sistemas informáticos. Por esta razón es importante pensar en soluciones que ayuden a prevenir intrusiones en redes y principalmente en redes de supercómputo o GRIDS. Con algunas herramienta de seguridad para sistemas informáticos pero principalmente con técnicas de virtualización se puede atraer a los intrusos pero también confundir y despistar simulando ser un equipo vulnerable, realiza un análisis y poniendo alerta a los involucrados en la red. La virtualización, es una herramienta de cómputo que puede trabajar conjuntamente con una red de supercómputo y en general con GRIDS, esta tecnología permite crear más de un Sistema Operativo sobre un equipo físico, en donde se obtienen grandes ventajas como lo es el ahorro económico, la consolidación de servidores, fácil manejo de la información etc. En este trabajo se propone el uso de herramientas de seguridad utilizando sistemas virtuales implementados en redes de supercómputo y GRIDS, en donde se tratan las diferentes tipos y técnicas de virtualización y los diferentes tipos de ataques y defensas.

Palabras clave:
Seguridad, GRID, máquina virtual.

## Random Algorithms for Scheduling Workloads on Grid

Ponente:
**Héctor Julián Selley**

M. en C. Héctor Julián Selley Rojas, Dr. Rolando Menchaca Méndez, M. en C. Manuel Alejandro Soto Ramos
*Computation Research Center / National Polytechnic Institute*

Abstract:

Grid computing has become a main paradigm for system development and computational analysis for complex systems because the advantages they offer to high performance computing.

Supercomputers give great computing capacity however they are too expensive for educational institutes and business companies. Because of this, Grid computing has been used and they can operate on a large number of conventional computers giving high performance computing. This computing system like any other high performance system requires continuous monitoring for performance and quality of service. Since Grid computing is built on machines that are on many different places monitoring all of them is very complex.

This work presents architecture of scheduling algorithm based on random algorithms. An analysis is made of some Grid scheduling algorithms trough experiments made on a Grid simulator GridSim. Such algorithms would be since random scheduling, round robin to some others algorithms like scheduling based on learning techniques.

## Simulation Environment for a Scheduling Algorithm with Guarantee Using Virtual Machines for Grid Systems

Ponente:
**Manuel Alejandro Soto Ramos**

M. en C. Héctor Julián Selley Rojas, Dr. Rolando Menchaca Méndez, M. en C. Manuel Alejandro Soto Ramos
Computation Research Center / National Polytechnic Institute

Abstract:

Grid computing came up like an infrastructure for large scale data distributed processing, this kind of technology scheduling systems needs to be design in a efficient way, considering fundamental parameters like resource utilization and processed jobs delivering time.

This article documents a simulation environment, implemented to characterized the performance of a scheduling algorithm in a grid system based in the concept of "virtual machines". The algorithm uses the jobs execution requirements and the services priorities execution to provide QoS to the grid users, the goal is the optimization of the grid resources and the environment implemented makes it possible to analyze the global system performance. In this sense, the simulation infrastructure allows to represent the grid behavior. It is possible to characterize the performance of the scheduling algorithms using a quantitative analysis of fundamental parameters like assignment times, execution, job's conclusion and the percentage of the resources used.

## Cluster Construction for Rendering of 3d Virtual Tours Models

Ponente:
**José Luis Cendejas Valdéz**

M.C.T.C. José Luis Cendejas Valdéz 1; L.I. Omar Ordoñez Toledo 1; M.T.I. Heberto Ferreira Medina 2 ; M.C. Víctor Hugo Zalapa Medina 1; I.S.C. Gerardo Chávez Hernández
*1. Universidad Tecnológica de Morelia*
*2. Ecosystem Research Center, UNAM, Campus Morelia, México*

Abstract:

In recent years the development of tours in 3D has been requested and has permitted creating animation, at the same time allows simulation of processes and its animation has helped organizations to reduce implementation costs on expensive tests. It is possible to observe these models in areas such as architecture, medicine, education, etc. For this reason the Multimedia and e-commerce. Faculty in conjunction with their networking seek to address the need in academic and business fields. This process is to expedite the rendering time on travels through a cluster based on free software and to provide the service on a portal that allows delivering a final file (video) with an optimal cost an in less time compared with very specific hardware features.

Keywords:
cluster, rendering, 3D tours, e-commerce, parallelism.

## Un enfoque para la Solución de TSP Utilizando Clústerización y Algoritmo de Genéticos.

Ponente:
**Daniel Jorge Montiel García**

ISC. Daniel Jorge Montiel García
Dr. Juan Martin Carpio Valadez
Dr. Rosario Baltazar Flores.
*Instituto Tecnológico de León.*

Resumen:

En el presente trabajo se propone una manera de combinar técnicas de clasificación y algoritmos genéticos dentro del marco de la hipótesis de bloques constructores, para reducir el tiempo de convergencia de los algoritmos genéticos en el problema TSP. Los resultados obtenidos de esta propuesta permitieron reducir el tiempo de cálculo y un mínimo cercano al óptimo.

## Proyecto del Centro de Supercómputo del Edo. México

Ponente:
**Jaime Klapp Escribano**
*Instituto Nacional de Investigaciónes Nucleares*

## Delta Metropolitana

Ponente:
**Juan Carlos Rosas / Manuel Aguilar Cornejo**
*Universidad Autónoma Metropolitana*

## Real-Time Communication Protocol for Supercomputing Ecosystems

Ponente:
**Carlos Alberto Franco Reboreda / Luis Alberto Gutierrez Díaz de León**
*Universidad de Guadalajara / CUCEA*

Abstract:
Supercomputing ecosystems are typically integrated by end-user communities, high-tech development communities, high performance computing infrastructure, and other support systems that interact to perform different activities or tasks. These ecosystems can be found and interact in local or distributed environments. For certain type of applications real-time communication within the ecosystem is a critical factor that must be guaranteed. One of the main challenges in the design of real-time communication systems is the definition of the required scheme to perform all system tasks in the ecosystem, so all time constraints are met.

In this work is presented a real-time communications scheme for a distributed ecosystem. This proposal is the result of the study of other local and distributed schemes and is extended to the supercomputing ecosystem approach. It is presented a communication protocol based on arbitrated message contention according to priority, which is given by a communications master plan. The proposed protocol considers periodic and aperiodic message delivery with static schedule and is based on common characteristics found in hard real-time systems (HRTS), includes a closed task set with time constraints, where critical tasks are defined as periodic tasks and it takes advantage of the broadcast nature of most networks found in real-time distributed systems. This work includes also a simulation scheme of the proposal.

Keywords:
supercomputing ecosystems, distributed real-time communication systems, arbitrated message contention, message scheduling.

## A Tier 1 Centre for ALICE in the UNAM

Ponente:
**Lukas Nellen Filla**

Ignacio Ania 1, Alejandro Ayala 2, Federico Carminati 3, Oscar Fernández 1, Lukas Nellen 2, Guy Paic 2, Fabián Romo 1
*1) DGTIC-UNAM, 2) ICN-UNAM, 3) CERN*

Abstract:

We present the project for the creation of a Tier 1 computing and data centre for ALICE at the UNAM. This centre will also support other projects and provide a platform for the development of new data-intensive scientific projects and support the development of a Mexican grid infrastructure, interconnected with existing infrastructures worldwide. We will present the existing grid node at the ICN-UNAM, the computing model of ALICE and the requirements of other experiments, in particular of the HAWC observatory. From this information, we derive the specifications of the data centre: at least 1000 cores and at least 1 PB of data. The large volume of data is new for HPC applications in Mexico, which makes this project an important tool, enabling scientific communities in Mexico to take on data intensive projects. Last, but not least, we will discuss how this projects challenges academic networking in Mexico.

## HPC Investigando la Isla de Calor mediante Modelos Geoestadísticos

Ponente:
**Elizabeth Brito Muñoz**

Rubén Sánchez-Gómez, Elizabeth Brito-Muñoz
*CU Valles, Universidad de Guadalajara*

Resumen:

La transformación antropogénica del medio ambiente en zonas urbanas, logra su máxima expresión en ciudades grandes, observándose por elementos como calles, banquetas y edificios, entre otros. Este cambio de uso de suelos afecta de modo especial las condiciones climáticas, sobre todo por variaciones locales en los flujos energéticos (atmósfera - superficie), que experimenta la naturaleza por este efecto poblacional, que se expresa con una diferencia de hasta 6oC de temperatura entre la periferia y el centro de la ciudad. Este fenómeno se conoce como Isla de Calor, su estimación implica cálculos intensos de registros en espacio y tiempo, lo que requiere el uso de cómputo paralelo para generar una mejor aproximación. En este trabajo se presenta un caso de aplicación de HPC mostrando en los resultados de interpolación espacial Kriging, implementado en paralelo y aprovechando la funcionalidad de los paquetes multicore y GridR del proyecto R para cómputo estadístico.

## Programación de Autómatas Programables con Lenguaje BASIC

Ponente:
**Ernesto Castellanos Velasco**

Ernesto Castellanos-Velasco , J.C. Chávez-Galván, I. Campos-Cantón
*Facultad de Ciencias, Universidad Autónoma de San Luis Potosí.*

Resumen:

La teoría de autómatas permite generar la descripción gráfica de una máquina de estados finitos. A partir de la descripción gráfica es posible la generación de varios lenguajes o códigos de programación. La descripción gráfica (en inglés directed graph), es un grafo dirigido que indica el conjunto de transiciones por realizar en un proceso. En ésta parte se puede contar con el auxilio de algún autómata programable o PLC. Para ciclar en forma continua la ejecución del proceso es preciso conocer las condiciones del estado inicial o "INICIO" y de aquellos estados intermedios que hagan bifurcación o unión de trayectorias en el grafo, lo cual se puede especificar a través de bloques de código que se conocen como rutinas, módulos, funciones, saltos/brincos, interrupciones, etc. Se ilustra en gran medida la similitud entre la técnica empleada en los años de 1970-1980 con la técnica GRAFCET y la programación modular a partir de la programación con lenguaje BASIC. EL objetivo del presente trabajo es mostrar un ejemplo práctico entre la alternativa de programación de los Autómatas Programables con sus entornos de programación enlatados y la programación modular configurada por el propio diseñador y programador del sistema.

## Grid Computing and Grid Initiatives for e-Science Virtual Communities in Europe and Latin America

Ponente:
**Ramón Diacovo**
*GISELA Project NGI/LGI Infraestructure Services Manager*

Abstract:

Grid Computing is a very well disseminated approach for solving computational and/or data intensive problems. With a relatively low investment, it can empower researchers with infrastructures that only the most resourceful would have access to otherwise.

GISELA is a project that aims at making this technology as widely adopted in Latin America as possible, by providing a broad set of production quality services, such as core grid services, Virtual Research Communities support and middleware interoperability platforms.

## Encrypted Information Transmission Through Chaotic Signals Over a Client-Server Communication Scheme.

Ponente:
**Maricela Jiménez Rodríguez**

Maricela Jiménez-Rodríguez, 1
Rider Jaimes Reátegui, 2
Octavio Flores Siordia, 1
*1. Centro Universitario de la Ciénega, Universidad de Guadalajara*
*2. Centro Universitario de Los Lagos, Universidad de Guadalajara.*

Abstract:

A numeric algorithm was designed for encryption and transmission of information. In this algorithm, two systems of nonlinear equations are solved, discreet and continuous, both of them within their chaotic regime. A synchronization method is used to transmit in the continuous system where the information is previously encrypted by confusion and diffusion techniques through the discreet system. The suggested algorithm guarantees high security for the information, encryption and decryption rapidity and strength against external attacks. In order to send information, a transmission scheme is created where two computers establish communication by means of client-server sockets implementation with the TCP protocol to guarantee data delivery.

Keywords:
Chaos, encryption.

## Construcción y Diseño de un Clúster Tipo HPC Virtualizado

Ponente:
**Juan Alberto Antonio**

Juan Alberto Antonio Velázquez / Jesús Antonio Álvarez Cedillo / Juan Carlos Herrera Lozada / Leopoldo Gil Antonio / Blanca Estela Núñez Hernández / Bruno Emyr Carreto Cid de León / Erika López González.
*Tecnológico de Estudios Superiores de Jocotitlán*
*Centro de Innovación y Desarrollo Tecnológico en Cómputo IPN, Depto. Sistemas.*

Abstract:

Virtualization has an important role in computer security, and most companies use it to protect the information using virtual systems that communicate with physical systems. Companies also see advantages as saving space and lower energy use which in turn is included in saving a lot of money. This paper defines the parallelization technique for load balancing and process migration, which can be done by kernel-level software such as MOSIX. You can define virtualization techniques used to handle different operating systems. Virtualizer software used was Virtual-box can be installed on different platforms and accept the installation of several operating systems including to Opensuse.

Keywords:
Virtualization, Mosix, clusters, Virtualbox

## Concurrent Rendering of 3D Interactive Simulations for Developing Job Skills in Mechanical Vibrations Measurement

Ponente:
**Alfredo Cristóbal Salas**

Alfredo Cristóbal-Salas(1), E. Morales-Mendoza(1), S. Pérez-Cáceres(1), T. Zárate-Martínez(1), E. Rodríguez- Alcantar(2)

*(1) Facultad de Ingeniería en Electrónica y Comunicaciones, Universidad Veracruzana, Poza Rica, Veracruz.*
*(2) Departamento de Matemáticas, Universidad de Sonora, Hermosillo, Sonora.*

Abstract:

This paper presents the experience in developing an interactive computer system for teaching mechanical vibrations based on skills development and case-based learning strategy. This system uses 3D high definition animations to illustrate mechanical vibrations concepts. We use Blender 2.49 to develop mechanical equipment and Multiblend which runs on a 70-node cluster to accelerate the render process.

Keywords:
Interactive Simulation, Mechanical Vibrations, Job Skills, Cluster computing, 3D animations

## The Cinveswall

Ponente:
**Amilcar Meneses Viveros**

Amilcar Meneses Viveros, Sergio V. Chapa Vergara
*Departamento de Computación, CINVESTAV-IPN*

Abstract:

The main goal of scientific visualization is to represent graphically information obtained from simulation or scientific databases. The graphic representation of data helps users to understand the phenomenon under studied. One of the problems in scientific visualization is to represent graphically large volumes of information. One way to attack this problem is to use video walls o tiled displays. The CINVESWALL is a video wall of 12 screens controlled by a visualization cluster of low-cost bassed in Apple technology. Most applications running on this display clusters are created as Cocoa distributed applications. We present the projects associated with CINVESWALL, such as developing applications in Cocoa, visualization techniques, development frameworks, the management of graphical user interfaces and open problems that exist in the technological developments relating to such devices graphics.

## Electronic Dissemination System of the Urban Observatory with Geographic Information System

Ponente:
**José Luis Jiménez Márquez / Romel Hernández Rosales**
*Research Department of the Instituto Tecnológico Superior de Puerto Vallarta.*

Abstract:

The dissemination of information between the scientific and technological community has become popular through web pages and not only that, virtually all sectors of the population have the ability to access Web pages and retrieve information.

This project aims to disseminate research results through a web system that allows users to place comments to the published
information, in order to enrich the published material.

Researchers can capture information and decide when and how long have it published, as well as if you have access to certain documents only through electronic payment.

Keywords:
Content Management System, Urban Observatory, Research dissemination.

## Monte Carlo Simulation Study Investigating High Threshold Method

Ponente:
**Rubén Sánchez Gómez**
*Universidad de Guadalajara*

Abstract:

The analysis of magnitudes that exceed a maximum allowable level or threshold has increased sharply in recent years, both in the statistics theory as in its application to natural processes, like environmental, climatological and hydrological processes among others. Threshold methods are applied when it has analyzed magnitudes that exceed over high threshold for an observed sample over a time period. Practical implementation of these methods requires methods for estimating their parameters. There are tree "general purpose" methods for estimating parameters of arbitrary distribution: Moment, Maximum likelihood and Bayesian methods. All of them require numerical computation, and one disadvantage is that the computations are not easily performed with standard statistical packages. In this work it presents a case of application on HPC, it shows the results of Monte Carlo simulation study and its implementation in parallel computing taking advantage of free software, running it with multicore and GridR packages.

Keywords:
Threshold methods, simulation study, parallel computing.

## Heterogeneous Humanoid Editor to Simulate Virtual Crowds in Parallel Processing's

Ponente:
**Martha Elena Zavala Villa**

Martha Elena Zavala Villa, Victor Manuel Larios Rosillo, Hervé Luga.
*CUCEA Guadalajara University, Toulouse France.*

Abstract:

This research work proposes a Heterogeneous Crowd Generation system for develop and manage virtual human crowds in parallel processing to populate different virtual spaces. The main research objective is to determine the needed mechanisms to generate crowds of hundreds of heterogeneous humanoids communicating and navigating in parallel. The humanoid crowd generation is achieved integrating in the DVRMedia2 project the parameterization principles that manages the MakeHuman project software, and integrating Genetic Algorithms to replicate humanoids in the crowd, and calculating a Distance Functions to obtain heterogeneous humanoids in the population. The parallel processing allows to speed the humanoid mesh generation and them representation in different three-dimensional worlds in less time and with higher quality.

Keywords:
heterogeneous crowd generations, humanoid crowd simulations, parameterization, parallel processing.

## Rediseño de Códigos Científicos para su Uso en Entornos de Trabajo

Ponente:
**Tania García Sanchez**

Tania García Sánchez, José Luis Villarreal Benítez, Ramón Ramírez Guzmán
*Universidad Nacional Autónoma de México*

Resumen:

El software científico se refiere a programas desarrollados para procesar datos científicos o cálculos numéricos a partir de un modelo científico. Su diferencia con software comercial o para el entretenimiento, desde el punto de vista de la computación no es mucha; inclusive en su pobreza en documentación, datos mal formateados, poco probados y con algunas rutinas poco entendidas. Este panorama no es exclusive del cómputo científico o del comercial, es un problema de malas prácticas en programación y es una de las principales razones por las que los científicos no publican sus códigos; pero si un código realiza su tarea, es suficiente razón para publicarlo y someterlo al escrutinio a través del peer review. La documentación y diseño de los códigos científicos también permite su más eficaz y eficiente depuración, actualización, y reuso. Rutinas probadas también pueden ser incorporadas en simulaciones más complejas. Se presenta un procedimiento y un diseño basado en componentes modificar las interfaces de las rutinas, probarlos, documentarlos e integrarlos en plataformas de software flexibles.

## The Virtual Observatory at the University of Guanajuato

Ponente:
**Juan Pablo Torres Papaqui**

Juan Pablo Torres-Papaqui 1, René Alberto Ortega-Minakata 1, Juan Manuel Islas-Islas 1, Ilse Plauchuf-Frayn 2, Daniel Marcos Neri-Larios 1, Roger Coziol.

*1.- Departamento de Astronomía, Universidad de Guanajuato.*

*2.- Instituto de Astrofísica de Andalucía (CSIC), E-18008, Granada, España.*

Abstract:

Astronomy is today one of the discipline in science which is richer in data, with an annual production of the order of tera-bytes, and with a few peta-bytes already archived. These data are now regulated by a global network under the new paradigm of the Virtual Observatory (VO). The goal of the VO is to develop and offer new tools that will facilitate the analysis of complex and heterogeneous astronomical data in order to produce valuable information about the universe.

As one of the project of the VO, we are presently involve in a new study which have for main purpose identifying and understanding the physical processes behind the cosmic evolution of galaxies. Using the Sloan Digital Sky Survey[1], which has already collected the spectra for more than a million objects, we are creating an homogeneous catalog of 926000 galaxies, for which we have determined their nuclear activity type, identified their morphology and environment. This catalog will be available through a web server, and will be open to data exchange using a browser and a server-to-server talking pipeline using HyperText Transfer Protocol. This project involves developing new protocols and scripts, including Common Gateway Interface, Secure Socket Layer, and Active Server Pages, to increase the capacity of server to deliver their information codified in HyperText Markup Language.

## Visualización Científica Orientada a Procesos de Tareas

Ponente:
**José Luis Villarreal**
*Universidad Nacional Autónoma de México*

Resumen:

La visualización científica es una herramienta muy poderosa para explorar y ganar intuición sobre datos complejos provenientes de instrumentos y simulaciones. El flujo de tareas para ir de la colección de datos, su control de calidad y preparación para simulaciones, hasta alcanzar la exploración visual y su presentación adecuada en diferentes foros, es un proceso que generalmente está desconectado; ya que requiere de muchas herramientas que no son construidas para el problema o proyecto, o fueron desarrolladas por grupos de investigadores para resolver tareas particulares, pero desconectadas para una nueva tarea. Por otro lado, el poder de la visualización radica en la generación de las imágenes a través de un proceso constructivo de enunciados visuales, argumentos y comunicación; es este proceso el que permite transformar los datos en información y ganar entendimiento y es requisito que quien estudia el fenómeno, participe en el proceso y que este proceso tenga una unidad de impresión (que el proceso tenga las mínimas interrupciones por los cambios de plataforma).

## VolcWorks: Suite de Simulación y Visualización para el Análisis de Riesgo Volcánico

Ponente:
**Ramón Ramírez Gúzman**

Hugo Delgado Granados, José Luis Villarreal Benítez
*Universidad Nacional Autónoma de México*

Abstract:

VolkWorks es una plataforma de software para la simulación y visualización de temas volcanológicos: procesos de nubes de cenizas, caída y depósitos de cenizas, proyectiles balísticos, flujos piroclásticos, flujos de lava, flujos de lajares y simulación de trayectorias; la cual permite el análisis de fenómenos volcanológicos de manera integral y orientada a la generación de mapas de riesgos volcánicos. Esta plataforma tiene como principio una arquitectura de implementación ágil que permite el desarrollo rápido de software a la medida, especificado por grupos de investigación concreta, líder en su campo. Se presentan los procesos para la implementación de los flujos de trabajo de una aplicación concreta, así como el conjunto de herramientas generales que acompañan a la plataforma y el GUI que las organiza. El diseño de la plataforma está orientado a componentes, con un diseño que permite integrar código nuevo o ya existente – de los investigadores - en cualquier paradigma de programación, incluyendo estructurado, manteniendo el desempeño a ese grano; lo cual permite incorporar soluciones particulares (rutinas, simulaciones, etc.) que trabajen acopladas.

## La realidad Virtual en los Nuevos Paradigmas de la Ciencia de Datos

Ponente:
**Lizbeth Heras**
*Universidad Nacional Autónoma de México*

Resumen:

La comunidad científica ha adoptado y enfatizado diferentes paradigmas, en respuesta al costo o esfuerzo de la tarea y herramientas que se requieren. De esta forma hemos pasado de la ciencia empírica a la teórica y luego a las aproximaciones con simulación. Actualmente el reto y mayor esfuerzo está en la exploración de grandes cantidades de datos – los generados por los observatorios astronómicos, los secuenciadores de DNA, los tomógrafos y muchos otros instrumentos y simulaciones. Estos paradigmas permiten avanzar rápidamente en la consolidación de nuevas teorías, pero pueden dificultar la generación de las alternativas junto con la resistencia por los paradigmas establecidos. Es importante innovar en herramientas que apoyen estos dos aspectos. La e-Science o el paradigma de la ciencia de datos, está aprovechando los nuevos medios y uno en particular es la Realidad Virtual Inmersiva. Esta herramienta permite el manejo de grandes cantidades de datos y su despliegue en muchas vistas enlazadas dinámicamente, pero sobre todo permite nuevas representaciones visuales en nuevos mundos artificiales.

# COMITÉ ORGANIZADOR

**CNS-IPICYT**
César Diaz Torrejón, Cynthia Lynnette Lezama Canizales, Sofía González Cabrera, Alberto Hernández Garcia, Pedro Gutierrez García, Adolfo Martínez Amador, Jose Antonio Solis Correa, Rómulo Guzmán Bocanegra, Roberto Alonso Hidalgo, Gabriela Lizbeth Meléndez Govea, Lázaro Delgado Hernández, Mónica Guerrero Montalvo

**UASLP**
Gerardo Padilla Lomelí, Liliana Félix Ávila, Mario Alejandro Chávez Zapata

**UPSLP**
Francisco Ordaz Salazar, Jorge Simón Rodríguez, Dr. Juan Antonio Cabrera, Rafael Llamas Contreras

**UdeG**
Moisés Torres Martínez, Veronica Lizette Robles Dueñas, Fabiola Elizabeth Delgado Barragán

**CUDI**
Salma Jalife

**UNAM**
Lizbeth Heras Lara, José Luis Villarreal Benítez

**UNISON**
María Del Carmen Heras Sánchez

**IPN**
Juan Carlos Chimal, René Luna

# COMITÉ ORGANIZADOR

**UAM**
Juan Carlos Rosas Cabrera, Jorge Elizalde Pérez
Luis Arturo Nuñez Pablo, René Pacheco López
Rodrigo Bermejo Martínez

**CICESE**
Andrei Tehernykh, José Lozano, Salvador Castañeda

**UCOL**
Juan Manuel Ramírez Alcaraz, Juan Antonio Guerrero Ibáñez

# AUTHOR INDEX

**Printed In:**

Los Talleres Gráficos de Transición

Mezquitan 465
Col. Artesanos
Guadalajara, Jalisco, México
November 2011

Printing 1,000 Issues/ 1,000 Ejemplares