FIRE MODELING: BEST PRACTICES FOR CONSTRUCTING ACADEMIC HIGH PERFORMANCE COMPUTING CLUSTERS

David J. Icove, PhD, PE, CFEI
The University of Tennessee, Knoxville, Tennessee, USA

Gregory E. Gorbett, MS, CFEI
Eastern Kentucky University, Richmond, Kentucky, USA

J. Douglas Birdwell, PhD
The University of Tennessee, Knoxville, Tennessee, USA

and

Richard E. Merck, PE, CFEI
Montgomery County Fire and Rescue Service, Rockville, Maryland, USA

ABSTRACT

The prudent teaching of the forensic fire engineering analysis[1] [2]of complex building fires must include the subject of computer fire modeling, particularly the use of the Fire Dynamics Simulator (FDS), developed by the National Institute of Standards and Technology (NIST). However, the problems associated with FDS require that large buildings be divided into rooms or zones, with each assigned to an individual computational mesh. These mutli-mesh models often produce relatively long execution times of days or weeks and typically require parallel processing computing clusters, which require technically challenging set up and maintenance. These performance gains can only be accomplished using high performance parallel processing computer clusters designed specifically to use operating systems and hardware that exploit the parallel functions of the FDS code. This paper presents best practices from the latest research in constructing small, medium, and large-scale high performance parallel processing computing clusters for use in academic environments to support NIST's FDS fire model.

INTRODUCTION

This paper presents best practices from the latest research in constructing small, medium, and large-scale high performance parallel processing computing clusters for use in academic environments.[3] It reports on joint research by the University of Tennessee, Eastern Kentucky University, and the Montgomery County (Maryland) Fire and Rescue Service into the design and implementation of turn-key high-performance computing clusters used specifically to support NIST's FDS fire model.

Users of computer fire models would prefer hours, not weeks, to review results of their large scale fire models. These performance gains can only be accomplished using high performance parallel processing computer clusters designed specifically to use operating systems and hardware that exploits the parallel functions of the National Institute of Standards and Technology (NIST) Fire Dynamics Simulator (FDS) code.

In the Spring Semester 2010, Mr. Scott Hansen, Mr. Timothy Wentz, Mr. Devin Robinson, and Mr. Jonathan Peyton undertook a Senior Design Project at the University of Tennessee, Department of Electrical Engineering and Computer Science (EECS). The purpose of this Senior Design Project was to explore performance bottlenecks in the FDS code, document the installation procedures, and explore the use of the Sun Microsystems computer servers and the OpenSolaris[4] operating system in comparison to other hardware/software configurations.

PROJECT GOALS

The prudent teaching of the forensic fire engineering analysis of complex building fires must include the subject of computer fire modeling, particularly the use of various computer fire models. In particular, the Fire Dynamics Simulator (FDS), developed by the National Institute of Standards and Technology (NIST), presents a challenge to both students and instructors in dealing with long execution times. However, the problems associated with FDS require that large buildings are divided into rooms or zones, with each assigned to an individual computational mesh. These mutli-mesh models often produce relatively long execution times of days or weeks and typically require parallel processing computing clusters, which require technically challenging set up and maintenance.

Project goals for the Senior Design Team included: (1) accelerating and improving the execution times of the FDS fire modeling program code; (2) exploring alternative hardware and software configurations using OpenMP and OpenMPI, code optimization, and targeted chipsets compilations; and (3) the design of three turn-key high-performance computing clusters suitable for use in forensic fire engineering analysis.

HARDWARE CONFIGURATIONS

The Project Team was directed to use three distinct classes of hardware computing platform configurations as a recommended business model. Considerations included the number of processors, memory capacity, and cost.

Available equipment for the testing included:

Sun X2200, Dual Quad Core Opteron	2.312 GHz	16 GB RAM	$2,800
Sun X4100, Dual Dual Core Opteron	2.793 GHz	8GB	$2,199
HP P6420F, Intel Core 2 Quad	2.5 GHz	8GB	$650
Asus A8N-VNM, Athelon 64 XD	2.2 GHz	4 GB	$550
SuperMicro X6DHT-G, 4x Intel Xeon	3.0 GHz	8 GB	$700

Available operating systems used in testing:

OpenSolaris Build 133	64-bit	free
Ubuntu Server 9.10	64-bit	free
Windows Server 2008	64-bit	$1,029
Windows 7 Professional	64-bit	$299.99

BENCHMARKING

Results with Small-Scale Systems

The results of this testing showed that the four-meshed multifloor_hi.fds model using Windows 7, Linux, and OpenSolaris with OpenMPI had similar performance characteristics. The conclusion reached for small-scale systems is that Window 7 operating system platform is adequate for small problems involving both single and multi-mesh computation, particularly when executions are for 4 meshes or less.

Figure 1.- Execution benchmark times for small scale systems

Results with Mid-Scale Systems

The results of this testing showed that less complex input files are indistinguishable. For the multifloor_hi.fds testing, OpenSolaris outperformed Windows Server 2000, providing the best overall performance. OpenSolaris provided the best overall performance, with Windows next, and Ubuntu last. For financial considerations, both Ubuntu and OpenSolaris are free operating systems.

This type of system is acceptable computational platforms, particularly when executions are for 5 to 8 meshes. The team also suggested that a separate computer running Windows and PyroSim would be beneficial in assisting the generation of FDS datafiles.

Results with Large-Scale Systems

The test results for large-scale systems using 2 nodes revealed that both OpenSolaris and Windows executed files in comparable times. The project team identified several variables in distributed testing that impact execution.

These factors include Ethernet communications between nodes, suggesting that high speed networks may improve performance. This type of system provides an acceptable computational platform, particularly when executions are for greater than 8 meshes. However, more benchmarking work is needed to be pursued, noting that this research was performed over only an academic semester by the UT student Project Team.

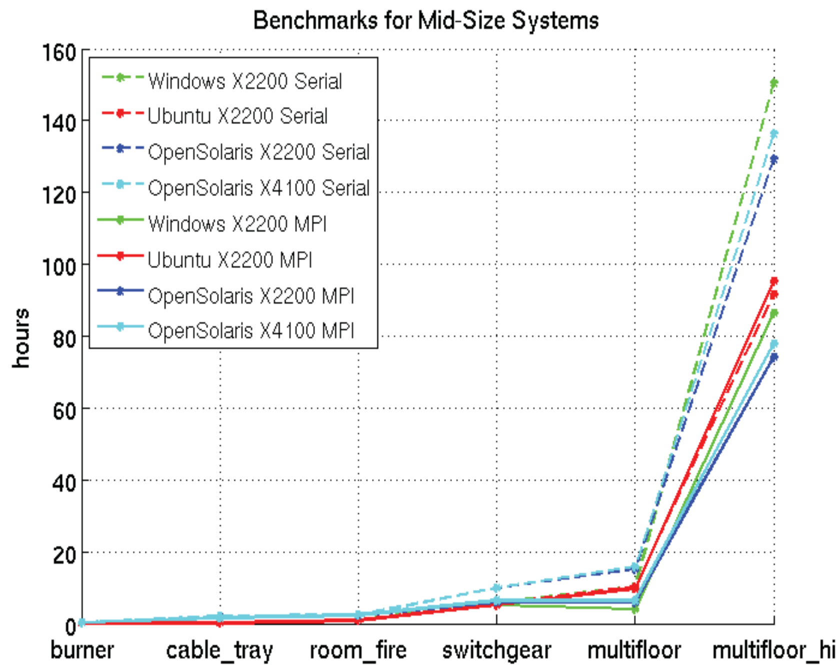

Figure 2- Execution benchmark times for mid-scale systems

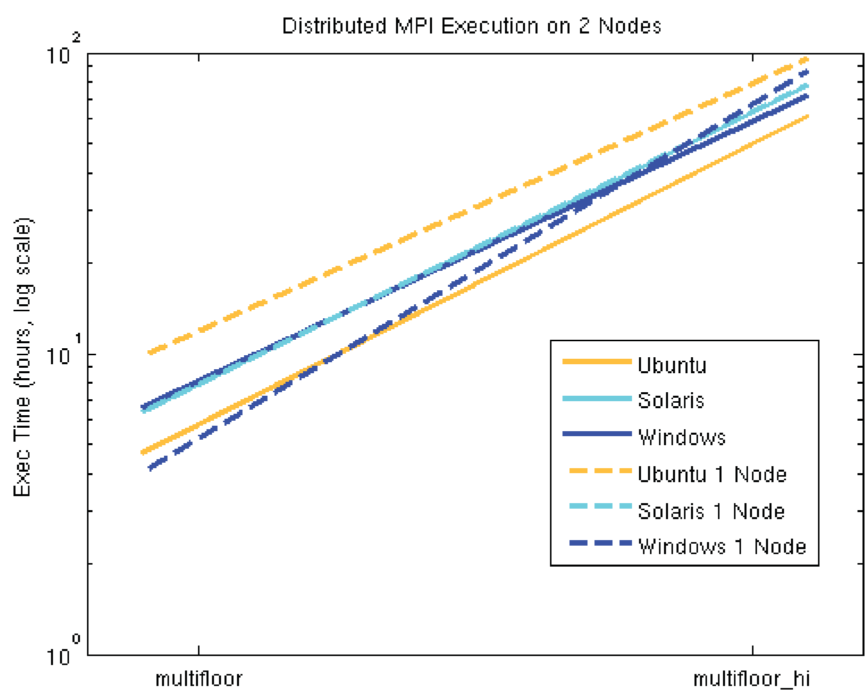

Figure 3- Execution benchmark times for large-scale systems

CONCLUSIONS

The UT Project Team reported that the benchmark analysis revealed unique characteristics for small, mid-, and large-scale computing platforms. Appreciable performance gains were not seen in small-scale systems, suggesting that larger scale systems be used, especially for complex input files.

The general recommendations for improved performance of FDS include using serial FDS binaries only for single mesh input files. A generally observed rule of thumb is to have at least as many processing cores available as meshes, and prevent oversubscription of cores which result in performance penalties.

The overall observation is that Windows 7 operating system should be reserved for small computer fire modeling problems of less than 4 meshes. However, medium- and large-scale computer fire modeling problems should use platforms running OpenSolaris or Linux operating systems.
Of note is the recent purchase of Sun Microsystems by Oracle and the uncertainty surrounding the future of OpenSolaris.

Finally, the Project Team noted that the convenience of PyroSim is invaluable in the building and inspection of data files. Although the computation takes place on an OpenSolaris server based system, it would be worth while to have a desktop computer running PyroSim.

ABOUT THE AUTHORS

David J. Icove, PhD, PE, CFEI, is a Research Professor in the Department of Electrical Engineering and Computer Science at the University of Tennessee, located in Knoxville, Tennessee. Dr. Icove is the co-author of several leading textbooks in the fire investigation field. He is a Registered Professional Engineer and a Certified Fire and Explosion Investigator (NAFI) and can be contacted at icove@utk.edu.

Gregory E. Gorbett, MSc, CFEI, is an Assistant Professor in the Fire and Safety Engineering Technology Program at Eastern Kentucky University. He is the co-author of a recently publish textbook on fire dynamics and a Certified Fire and Explosion Investigator (NAFI) and can be contacted at Greg.Gorbett@eku.edu

J. Douglas Birdwell, PhD, is a Professor in the Department of Electrical Engineering and Computer Science at the University of Tennessee, located in Knoxville, Tennessee. Dr. Birdwell is the Director of the Laboratory for Information Technologies. He can be reached at Birdwell@lit.net

Richard E. Merck, PE, CFEI, is the Senior Fire Protection Engineer for Montgomery County Fire and Rescue Services, Rockville, Maryland. He can be reached at rick.merck@montgomerycountymd.gov

REFERENCES

[1] A. M. Christensen and D. J. Icove, The Application of NIST's Fire Dynamics Simulator to the Investigation of Carbon Monoxide Exposure in the Deaths of Three Pittsburgh Fire Fighters, J. Forensic Sci., 49 (1), 2004.
[2] Y. Huang Shen, , and S. Chien, Using Fire Dynamic Simulation (FDS) to Reconstruct an Arson Fire Scene, Building and Environment, 42, 2008, 1036-1045.
[3] For a discussion of this field see: D. Eadline, High Performance Computing For Dummies®, Sun and AMD Special Edition Published by Wiley Publishing, Inc., Hoboken, NJ , Wiley Publishing, Inc., Indianapolis, Indiana, 2009.
[4] H. J. Foxwell and C. Tran, Pro OpenSolaris: A New Open Source OS for Linux Developers and Administrators, New York, NY: Springer-Verlag, 2009.