*The Monte Carlo Method: Versatility Unbounded In A Dynamic Computing World*
Chattanooga, Tennessee, April 17–21, 2005, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2005)

# FEASIBILITY STUDY OF MCNP IN A GRID COMPUTING ENVIRONMENT

**Nicholas DiFrancesco and Bernadette L. Kirk**

The Ohio State University - Nuclear Engineering Program
Suite 255
650 Ackerman Road
Columbus, OH 43202

Oak Ridge National Laboratory
Building 5700, Room H301, Mail Stop 6171
Oak Ridge, TN 37831

difrancesco.6@osu.edu; kirkbl@ornl.gov

## ABSTRACT

With the proliferation of varied hardware architectures from personal computers (PCs) to massively parallel systems, computing has become an intrinsic part of the daily work environment. In spite of the workload that is borne by computers in a typical office, many PCs exhibit a lot of idle time [3]. In this study, we investigate the feasibility of the Monte Carlo N-Particle (MCNP) [4] code in a grid computing environment.

The particular software we use is the Globus Toolkit (http://globus.org), an open source software toolkit used for building grids. The open source Globus Toolkit is a fundamental enabling technology for the "Grid," letting people share computing power, databases, and other tools securely online across corporate, institutional, and geographic boundaries without sacrificing local autonomy. The toolkit includes software services and libraries for resource monitoring, discovery, and management, plus security and file management.

*Key Words*: Globus, MCNP, Grid Computing, Open Source

## 1 INTRODUCTION

The original definition of the computing grid is given below [1].

"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."

Later the definition was refined [2] and stated that grid computing includes coordinated resource sharing and problem solving in virtual organizations. Individuals, institutions and resources form the virtual organization. Thus, grid computing is not impeded by physical boundaries.

Nuclear applications historically have demanded cutting edge computing technology to perform the best analysis. The next change in the use of computing resources will utilize grid computing. Grid computing uses heterogeneous computer systems from various locations from across the internet to pool computing resources. This technology promises to change the way computing resources are allocated to resemble a model similar to traditional utilities. With this new model, computing resources are requested and payment is exchanged for processing time and storage space.

Our exploration of this technology is being performed using an open source grid infrastructure called the Globus Toolkit. This code has been developed at Argonne National Laboratory. Our test application is Monte Carlo N-Particle (MCNP5) that must be grid enabled. MCNP5 has been written for some distributed computing environments. An interface call Message Passing Interface (MPI) can be implemented on clusters and other distributed networks. Globus has an implementation of the open source MPI called MPICH-G2, also supported by Argonne National Laboratory. This version was written originally for Globus version 2.4.

Other considerations with this project include the use of MCNP5. By reducing operating overhead, scalability of MCNP would improve as a grid application.

This project development required the installation, operation and development of applications within Globus.

# 2 PROCESS

## 2.1 Installation

The installation of Globus must occur on a machine-by-machine bases. Each installation involves numerous pieces of supporting software, including Java SDK, Apache Ant, and a SQL server. The installation of Globus is suited to the Linux environment; however, installations can occur on any Java supported platform. The installation process requires a minimum of a day to complete. This is because of the installation requirements of supporting software and several hours of time to compile the Globus source.

The Globus toolkit comes with several grid-supporting daemons that run primarily on the Linux operating system, such as MDS (Monitoring and Directory Service) and GRAM (Grid Resource Allocation and Management). These tools support the construction of more advanced grids. Platforms other than Linux such as Windows are limited to the basic Globus installation as a client machine for the grid.

After installation, authentication setup of a computer grid requires the creation of a common certification authority to sign both host and user certificates. This requires the systems administrator to distribute these certificates to each of the machines on the grid. Trusted certification authorities can be added to the local list of virtual organizations. This is used to set up partnerships between different facilities/organizations located elsewhere.

Another area of future study should address the area of export control. Globus has several types of system security to protect communication and local applications. Relationships need to be established between trusted institutions to avoid violation of these rules.

## 2.2 Operations/ Difficulties

The Globus service is started by Globus user, the owner of the Globus files. The container is typically started on port 8080, like most web services. The user then authenticates credentials that expire in 12 hours. After these two steps have been completed, there are two methods to test the container and web services. GT 3.0 includes a GUI service browser; this allows quick testing of deployed services. Many instances can be created by launching one of the many Globus sample factories. The other method uses command line calls to the Globus service. This requires a long string for an input.

Most of our time was invested in learning to debug the Globus installations, tracking error messages and changing Linux settings. Most errors in the final installation occurred because of problems with the certification files. In many of the tutorials, simplified certifications settings were suggested. This slowed the setup process because it created more configuration conflicts than it simplified.

The Globus Team has not effectively organized the technical supporting resources; support was more effectively sought using the internet search engine Google. Explanations of error messages were often incomplete, and created difficulty aligning error messages with proper remedies. Error messages generated by Globus are targeted towards developers, which complicated the understanding of the underlying system. Support for the application development by the Globus Team is even more limited requiring the reliance on outside organizations. Evaluation of grid-supporting daemons must occur once scaling of the grid begins.

### 2.3 Application Development

Primitive globus applications are programmed using Java. A sample Globus application, from B. Sotomayor was deployed onto our sample grid. To deploy a Globus application, it is required that a Java Implementation file - described as a Grid Web Service Description Language (GWSDL) file - be converted to a WSDL file. These files are used to generate the stubs files and then everything is packed into a GAR file (grid archive file). These files are deployed into Globus with Ant. Following this step, a client must be compiled with 'javac' to communicate with the newly deployed web service. Difficulties were encountered with the sample application, including the build.xml file and the server-deploy.wsdd and server-undeploy.wsdd files. Modification of such files is required for each version of a Globus release.

The web services and Globus application development are skewed towards building simple java applications from the ground up. Existing codes must satisfy numerous conditions, which are summarized well in 'Chapter 6: Project and Design of Grid Application' *Grid Services Programming and Application Enablement* [6].

One potential development tool that was worked with was the Eclipse IDE. This was created to streamline/automate the development process of coding grid applications. A Globus plug-in developed by the University of Deusto (Bilbao, Spain), which automated the generation of the GWSDL file, stubs and the creation of the GAR file.

Supporting components such as Java CoGs should be explored to reduce the administrative burden associated with running a grid application.

### 2.4 MPI Investigation

The required tool for development of a grid application from an existing code is MPICH-G2. This should facilitate using existing MPI applications in the grid environment without changes. Considering the complexity of most nuclear codes, it is unrealistic for portions of existing codes to be rewritten as a Java application.

In practice, MCNP is well suited for grid applications if the communication overhead can be minimized. The performance of MCNP on the grid with MPI will be dependent on how well it can operate in heterogeneous computing environments. Some of these concerns can be addressed within the Globus job files by explicitly specifying required computer capabilities. Study of the MCNP MPI implementation should examine where bottlenecks can occur. For example, processors may idle during statistical pooling periods, especially on heterogeneous grids.

MPICH-G2 requires that MCNP be compiled using its specific compiler. MPICH-G2 has support for both FORTRAN and C codes. The size and complexity of the MCNP source code may require debugging and modification of the MCNP code. At this same time, the MPI implementation could also be examined to determine how the code has been parallelized. Communication bottlenecks and barriers need to be reduced.

Compatibility issues may emerge as new Globus and MPICH-G2 releases become available. The last MPICH-G2 release was targeted for Globus 2.4. The next Globus 4.0 release is expected in February 2005. Globus 4.0 release would represent a more static platform for development because of the final adoption of the grid infrastructure in the WSDL 1.2 standard.

## 3 CONCLUSIONS

Now that the internals of Globus have been studied, and supporting grid technologies have been reviewed, development can proceed to grid-enable MCNP. Hurdles to overcome include examining whether Globus solves the issues of heterogeneity, MCNP requirements/limitations, and the complete rollout of a Globus testbed.

Benchmarking would involve performing various identical MCNP test cases on clusters and on the grid. With this exercise, the amount of communication overhead and linearity of scaling could be assessed. Specific benchmarking of the hardware should be performed to identify application bottlenecks.

A grid-enabled MCNP would allow the rapid completion of Monte Carlo calculations. This would provide users with the ability to increase solution detail or the speed of calculations in a time critical environment. A robust, dependable and secure solution is a goal that the Globus framework should allow.

## 4 ACKNOWLEDGMENTS

I would like to thank Bernadette Kirk for her mentoring and support during my internship at Oak Ridge National Laboratory. Funding for this project was also provided by the Department of Energy for this internship opportunity.

## 5 REFERENCES

1. I. Foster, C. Kesselman**,** "Computational Grids," *The Grid: Blueprint for a New Computing Infrastructure*, Morgan-Kaufman (1999).
2. I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International J. Supercomputer Applications*, 15(3) (2001).

3. M. Mutka and M. Livny. "The available capacity of a privately owned workstation environment," *Performance Evaluation*, 12(4):269{84, (1991).

4. X-5 Monte Carlo Team, "MCNP-A General Monte Carlo N-Particle Transport Code, Version 5 - Volume I: Overview and Theory," *MCNP5 Manual*,**Vol I** ( 2003).

5. B. Jacob, L. Ferreira, N. Bieberstein, C. Glizean, J. Girard, R. Strachowshi, S. Yu, *Enabling Applications for Grid Computing with Globus*, IBM: Redbooks (2003).

6. L. Ferreira, A. Thakore, M. Brown, F. Lucchese, (omitting 7 others) *Grid Service Programming and Application Enablement*, IBM: Redbooks, (2004).

7. "Oxford Physics, General GT3 Problem Debugging," http://www-pnp.physics.ox.ac.uk/~stokes/twiki/bin/view/DIRAC/DebuggingGT3 (2004).

8. B. Jacob, L. Ferreira, S. Slevin, M. Brown, S. Sundararajan, J. Lepesant, J. Bank, *Globus Toolkit 3.0 Quick Start*, IBM: Redbooks (2003).

9. X-5 Monte Carlo Team, "Installing and Running MCNP on Various Systems, Section: Installing Parallel Software – MPICH.NT or PVM," *MCNP Manual*, pp. C-24 – C-42, (2003).

10. "Globus: Training Resources," http://www.globus.org/training (2004).

11. "Globus Toolkit 3.2 Documentation" http://www-unix.globus.org/toolkit/docs/3.2/index.html (2004).

12. B. Sotomayor, "The Globus Toolkit 3 Programmer's Tutorial," http://www.casa-sotomayor.net/gt3-tutorial (2004).