Grid Computing for Virtual Experimentation and Analysis

Nirupam Julka[1], Peter Lendermann[1], Wentong Cai[2], Nguyen Quoc Khanh[2], Terence Hung[3]

D-SIMLAB, Singapore Institute of Manufacturing Technology[1],
71 Nanyang Drive, Singapore 638075

School of Computer Engineering, Nanyang Technological University[2],
Singapore 639798

Institute of High Performance Computing[3], 1 Science Park Road, #01-01 The Capricorn,
Singapore 117528

nirupamj@SIMTech.a-star.edu.sg, peterl@SIMTech.a-star.edu.sg, aswtcai@ntu.edu.sg,
terence@ihpc.a-star.edu.sg

Abstract

Virtual experimentation essentially includes all studies that are performed on a computer representation of a real system. Any such study needs to evaluate the response of the system to a number of different parameters. This requires a large amount of computing power and resource sharing and thus become an ideal candidate for Grid Computing. This paper discusses the use of the Grid to perform studies using Discrete Event Simulation models. It also presents the challenges encountered during development of a prototype to enable the same and discusses some future work.

I. Introduction

Today, a corporation's operations and business processes are subject to constant changes. To stay competitive, adaptations and enhancements of manufacturing and service operations and the associated business processes need to take place continuously. This requires advanced virtual experimentation techniques, especially in highly capitalized industries where experimentation with the real system would be too disruptive and costly and often is not possible at all. Such techniques also have to take into account the specific characteristics of today's pull-environments, in which operational execution plans are the result of a complex translation from frequently-changing customer demand into material quantities to be released into and moved within the manufacturing and logistics systems at pre-specified times (Lendermann, Gan and McGinnis, 2001). Moreover, today's high-tech companies operate in global networks that often involve contract manufacturers and 3rd party logistics providers, driving the underlying systems towards mega-networks with large risk and reward, rapid responses and high-speed switching where relationships are formed and dissolved in context of particular jobs. In such fast changing business environments it thus becomes critical to perform virtual experimentation studies on various models representing the manufacturing and logistics systems and yield results in a short duration of time.

Discrete Event Simulation (DES) has been established as a very important tool for virtual experimentation in the manufacturing, logistics and service domains (Banks, 1998; Law and Kelton, 2000). Experiments in the above domains consist of scenarios having a distinct set of parameters. Applications supporting this form of experimentation are also known as parameter sweep applications or PSAs (Berman et. al, 2003). The aim is to achieve the right set of parameters through executing these scenarios using the same input data. Activities in the manufacturing, logistics and service domains have stochastic parameters associated with them, for example, the times to service a client in a bank teller queue form a normal distribution with a mean and standard deviation. It thus becomes necessary to execute replications of the scenarios to achieve statistically relevant results, effectively increasing the number of individual executions of the model by a large amount. The scenarios are not structured to be run in a sequential fashion and thus provide an ideal opportunity to use the Grid as the execution platform.

Development of a user-level middleware to seamlessly allow user the deployment of number of replications, scenarios and experiments on a computational grid is not a novel idea in itself. However, a middleware focused on the execution of DES models based on either Commercial-of-the-Shelf (COTS) packages or programming languages like C++ and Java is still lacking. There is a need of a middleware to which a user can define the application type it needs to run, scenario configurations and experiment structure and receive as a results the output generated by the simulation. This middleware can then be seamlessly integrated with optimization algorithms in effect making the entire process of simulation experimentation automated.

This paper describes a middleware to use the Grid as an execution platform for DES-based virtual experimentation in the manufacturing, logistics and service domain. The middleware takes into account the different implementations (COTS, C++, Java, etc.) and resource requirements of the simulations and defines interfaces for optimization modules. Section 2 presents two different DES-based virtual experimentation scenarios, one using COTS and the other using a C++ simulator, which motivated this work. Section 3 defines the middleware and provides the interfaces to optimization modules and future work planned under this project is discussed in Section 4.

II. Motivation

In April 2004, with support from Agency for Science Technology and Research (A*STAR), an initiative to develop a single software infrastructure with the capability to handle different forms of virtual experimentation in the manufacturing, logistics and service sectors was launched. This initiative is part of a national pilot for Integrated Manufacturing and Service Systems (IMSS). Two different experimentation approaches are studied under this pilot. The first approach uses COTS simulation packages in conjunction with agent technology to develop a symbiotic simulation system (Ayani et al., 2002). The agents update the simulation model based on state changes in the real system e.g. a machine breakdown in the factory needs to be reflected in the simulation model. The simulation model is thus kept up-to-date and simulation time and real time are effectively synchronized. In case a study needs to be performed to choose an operational decision from a set of options, another agent modifies the simulation model to reflect the different options. The agent then executes all the simulation runs at simulation speed to ascertain the different outcomes and reports it back to the user. The execution of all the simulation runs, which include replications of the different options, is very resource intensive. This was the motivation behind the development of a module which could accept input parameters from the agent, deploy all simulations on a grid, execute them in parallel and report back the output.

The second approach involves an optimization scenario where a set of parameters are tested using simulation and based on the results a modified set of parameters are simulated. The optimization module thus creates a set of new parameters based on the results of the earlier set. The replications of the same set of parameters can be run in parallel but the different sets are essentially in sequence. The optimization module here too needs a grid execution module similar to the one above.

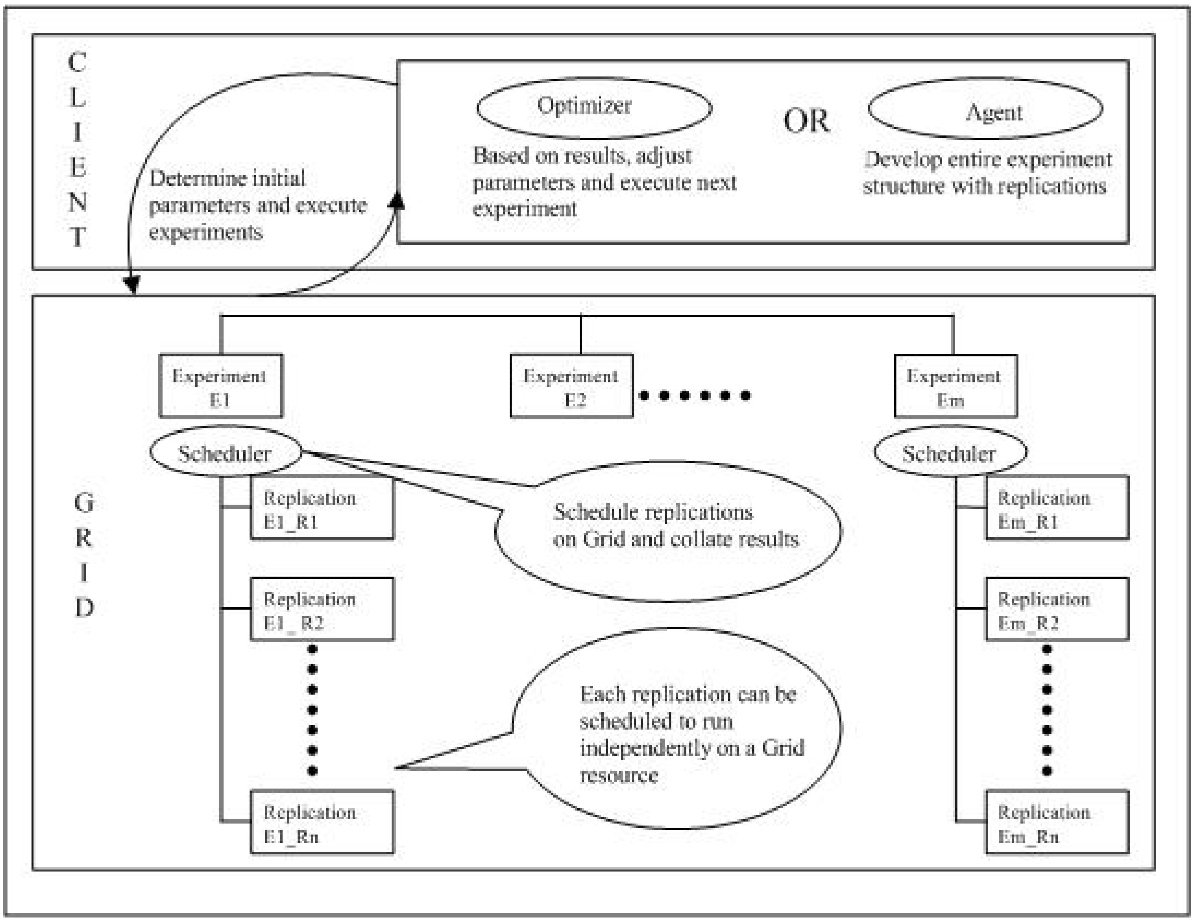

Figure 1 Generic Architecture

III. Overall Framework

Figure 1 shows the generic architecture of the solution needed for both the above approaches. The inputs for the client side software include the following.

a. Common parameters file (CP file): This file is essentially an archive of the input files for the replications of an experiment. These input files are needed by all replications of an experiment. The number and types of input files essentially depends on the client application.

b. Master Replication Parameters file (MRP file): This file includes information on number of replications, directory information for results, prioritization information and executable application.

The above two files are processed by the client side software and jobs are prepared based on the information in the MRP file. The client side software then schedules the jobs on various remote machines, transfers the needed files to the remote machines, executes the jobs and stores the retrieved results on the local directory. The client side software consists of the following modules (See Figure 2).

a. Client interface program: This module is accessed by the optimizer or the agent. The above two files are passed as parameters to this module.
b. GRAM job module: This module provides a mechanism to run the jobs remotely and parallel on other machines. It uses the standard resource management functionality provided by Globus.
c. Resource allocation and scheduling module: This module monitors the available resources on the grid using MDS. It then performs parameter sweep job scheduling for the experiment.

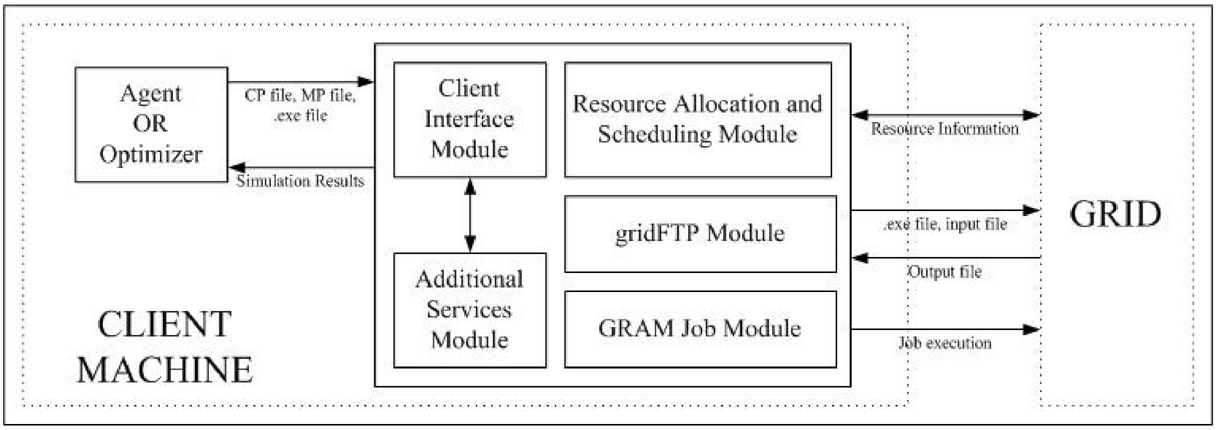

Figure 2 Interactions of Software Modules

a. gridFTP module: This module transfers executable and input files to the remote machines and retrieves output files from the remote machines to the predefined directory on the client machine.
b. Additional services module: This module provides additional services like processing of CP and MRP files to individual input files and creation of directory structures and mapping of information of the experiments and replications to the right directories.

The above modules mainly use Globus 2.2 API (Globus, 2004). Present implementation uses only work queue based scheduling and the client software is written in C++ and works on a Linux platform.

IV. Challenges Ahead

The present implementation of the virtual experimentation software can only work on the Linux platform. Since the optimizer and associated programs are written in C++ and can be executed on Linux, the present implementation of the virtual experimentation software can only be used with the optimizer. Cross platform execution of programs which is essentially required in the case of the symbiotic simulation system (symbiotic system is developed on windows and uses COTS packages available for windows only) is still unavailable. This will be overcome by using the Web Services based Globus Toolkit 4 with a stable release scheduled by May 2005 (Globus, 2005). Immediate future work will involve development of a new client application using version 3.9.4, an alpha-quality development release.

Acknowledgements

The work mentioned in this paper was supported by an A*STAR grant under the IMSS pilot. The authors would also like to thank the Parallel and Distributed Computing Centre for providing access to many computational resources.

REFERENCES

[1] Ayani, R., J. Fowler, R. Fujimoto, P. Mosterman, G. Riley, O. Rose, S. Strassburger, H. Szczerbicka, S. Turner, L. Uhrmacher, B. Unger and F. Zambonelli (2002). Parallel/Distributed Simulation Working Group Report. Dagstuhl
Banks, J. (1998). Handbook of Simulation. New York, John Wiley & Sons, Inc.

[2] Berman, F., R. Wolski, H. Casanova, W. Cirne, H. Dail, M. Faerman, S. Figueira, J. Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen, N. Spring, A. Su and D. Zagorodnov (2003). Adaptive Computing on the Grid Using AppLeS. IEEE Transactions on Parallel and Distributed Systems (TPDS), 14(4), 369--382.

[3] Globus (2004). Globus Toolkit API Reference. Available at www.globus.org/developer/api-reference.html

[4] Globus (2005). Status and Plans for the Globus Toolkit 4.0 (GT4) as of February 1 2005. Available at www-unix.globus.org/toolkit/docs/development/4.0-drafts/GT4Facts/index.html

[5] Law, A. M. and W. D. Kelton (2000). Simulation Modeling and Analysis, McGraw-Hill.

[6] Lendermann, P., B. P. Gan and L. F. McGinnis (2001). Distributed Simulation with Incorporated APS Procedures for High-fidelity Supply Chain Optimization. Winter Simulation Conference, pp 1138-1145. Arlington, U.S.

NIRUPAM JULKA is a Research Engineer with the D-SIMLAB Programme at Singapore Institute of Manufacturing Technology. He received a Bachelor of Technology (Honors) from Indian Institute of Technology (IIT) Kharagpur in 1999 and a Master of Engineering from the National University of Singapore in 2002. His dissertation work included investigation of chemical supply chains and development of proper frameworks for decision support systems to handle them. He is presently pursuing a Singapore-based Engineering Doctorate with Cranfield University in the field of Manufacturing Systems. His research interests are supply chain management and optimization, interoperability of decision support systems with simulation components and capacity expansion processes in global manufacturing networks. His email address is nirupamj@SIMTech.a-star.edu.sg.

PETER LENDERMANN is a Senior Scientist and Head of the D-SIMLAB Programme at Singapore Institute of Manufacturing Technology (SIMTech). Previously he was a Managing Consultant with agiplan in Germany where his focus was on the areas of supply chain management and production planning. He also worked as a Research Associate at the European Laboratory for Particle Physics CERN in Geneva (Switzerland) and Nagoya University (Japan). He obtained a Diploma in Physics from the University of Munich (Germany), a Doctorate in Applied Physics from Humboldt-University in Berlin (Germany) and a Master in International Economics and Management from Bocconi-University in Milan (Italy). His research interests include modelling and analysis of manufacturing and logistics systems as well as distributed simulation. His email address is peterl@SIMTech.a-star.edu.sg.

CAI WENTONG is an Associate Professor with School of Computer Engineering at Nanyang Technological University (NTU), Singapore, and the head of Computer Science Division. He received his B.Sc. in Computer Science from Nankai University (P. R. China) and Ph.D., also in Computer Science, from University of Exeter (UK) in 1985 and 1990 respectively. He was a Post-doctoral Research Fellow at Queen's University (Canada) before joining NTU in Feb 1993. Dr. Cai's research interests include *Grid & Cluster Computing, Parallel & Distributed Simulation,* and *Parallel & Distributed Programming Environments.* He has authored or co-authored over 150 journal and conference papers in the above areas. His recent research has focused on the development of Middleware for Grid/Cluster computing, particularly in the context of the development and execution of distributed simulations. His email address is ASWTCAI@ntu.edu.sg.

TERENCE HUNG is the Senior Programme Manager of the Software & Computing programme at the Institute of High Performance Computing where he is responsible for R&D in the areas of grid computing, analytics, optimisation, geometry processing and visualization. He was formerly VP Technology of Commerce Exchange Pte Ltd and has consultancy experience in HPC, e-commerce and e-financial. Terence's current research focus is on grid middleware to address needs in computational science and engineering problems and remote collaboration. In addition to leading the national Physical Science Virtual Grid Community efforts, Terence is also a member of the national Manufacturing Virtual Grid Community and Middleware workgroup. He obtained his Ph.D. in Electrical Engineering from the University of Illinois at Urbana-Champaign in 1993. He currently holds an adjunct associate professor appointment at the School of Computer Engineering of the Nanyang Technological University. His email address is terence@IHPC.a-star.edu.sg.

NGUYEN QUOC KHANH is pursuing a Bachelor of Engineering degree in Computer Engineering at the Nanyang Technological University. His email address is khanht5@pmail.ntu.edu.sg.