

Implementation of Computational Grid Services in Enterprise Grid Environments

¹R.J.A. Richard, ²Ajay Anant Joshi and ¹C. Eswaran

¹Faculty of Information Technology, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

²University of West Indies, St. Augustine Campus, St. Augustine, Trinidad, West Indies

Abstract: Grid Computing refers to the development of high performance computing environment or virtual super computing environment by utilizing available computing resources in a LAN, WAN and Internet. This new emerging research field offers enormous opportunities for e-Science applications such as astrophysics, bioinformatics, aerospace modeling, cancer research etc. Grid involves coordinating and sharing of computing power, application, data storage, network resources etc., across dynamically and geographically dispersed organizations. Most Grid environments are developed using Globus toolkit which is a UNIX/Linux based middleware to integrate computational resources over the network. The emergence of Global Grid concept provides an excellent opportunity for Grid based e-Science applications to use high performance super computing environments. Thus windows based enterprise grid environments can't be neglected in the development of Global Grids. This study discusses the basics of enterprise grids and the implementation of enterprise computational grids using Alchemi Tool Kit. This review study is organized into three parts. They are (i) Introduction of Grid Technologies, (ii) Design Concepts of Enterprise Grids and (iii) Implementation of Computational Grid Services.

Key words: Grid computing, enterprise grids, computational grids, alchemi, grid middleware

INTRODUCTION

The essence of Grid Computing in enterprises will be realized as a rule rather than as an exception in the near future. Most of the world-wide researches have aimed at developing a Grid framework that embraces the Linux/Unix clusters, but on the contrary there exists a large scale deployment of windows-based desktops in most of the organizations. Therefore the main objective of this study focuses on evolving a virtual supercomputer by aggregating the unused power of the desktop PCs in an enterprise, by developing Grid Services that use Windows-based Grid computing framework called Alchemi (A Grid Services Middleware Developed by GRIDS Lab, University of Melbourne, Australia) as the Grid Engine.

GRID COMPUTING TECHNOLOGY

Computational Grid can be defined as a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities^[1,2].

Grid computing applies the resources of many computers in a network to a single problem at a given

time-usually to a scientific or computational problem that requires a greater number of CPU cycles or access to large amounts of data. A well-known example of grid computing in the public domain is the ongoing SETI (Search for Extraterrestrial Intelligence) @Home project in which thousands of people are sharing the unused processor cycles of their PCs in the vast search for signs of rational signals from outer space^[3]. European Data Grid project provides huge data storage capacity by utilizing available data resources across geographical locations^[4]. In the United States, the National Technology Grid is prototyping a computational grid for infrastructure and an access grid for people^[5,6].

Grid Computing Infrastructure will support the sharing and coordinated use of resources in dynamic global heterogeneous distributed environments. This includes resources that can manage computers, data, telecommunication, network facilities and software applications, provided by different organizations^[7]. As grid computing provides several solutions that meet the requirements of e-Science challenges, grid technologies have been proposed as an enabling infrastructure for applications where large volume resources are required. There have been significant advances in grid

Corresponding Author: R.J.A. Richard, Faculty of Information Technology, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

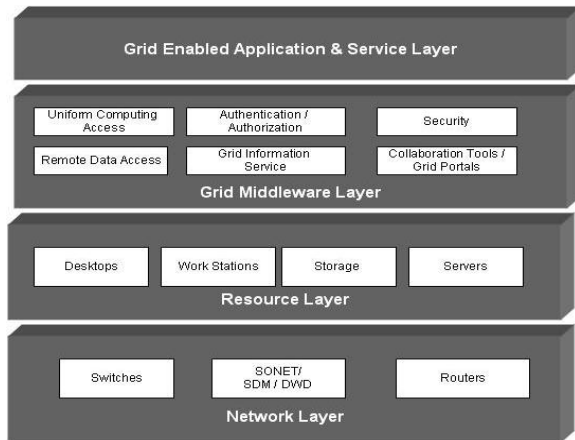


Fig. 1: Basic grid architecture

technologies on all levels, from the network infrastructure to user applications, which have led to the emergence of data grids^[8]. Thus the implementation of Grid infrastructure in an organization would provide a virtual super computer to run high performance e-Science applications and mass data storage provisions. Figure 1 shows the basic Grid architecture^[9].

Network layer: The network layer provides the basic network infrastructure to connect the computational resources by utilizing switches, routers, SONET, SDM etc. Network layer carries the communication protocols for resources and data sharing in a Grid environment.

Resource layer: The resource layer defines the interface to local resources that may be shared. This includes computational resources, data storage, networks, catalogs, software modules and other system resources. This layer uses the communication and security protocols (defined by the connectivity layer) to control secure negotiation, initiation, monitoring, accounting and payment for the sharing of functions of individual resources. The resource layer calls the resource layer functions to access and control local resources.

Middleware layer: The middleware layer consists of intelligent software tools to deploy Grid Services in LAN/WAN/Internet environment. The middleware layer provides uniform access to the resources, schedules the applications and organizes the data distributions. It also provides security framework, authentication and authorization protocols for the Grid enabled applications to secure the data transmission. The recent grid toolkits enable the Web Services for data distribution through open standards such as OGSA

(Open Grid Service Architecture) or WSRF (Web Services Resource Framework) or OGSI (Open Grid Service Infrastructure).

Application layer: The application layer enables the use of resources in a grid environment through various collaboration and resource access protocols^[9,10].

Types of grids: The Grids can be classified as Compute Grid, Data Grid, Application Service Provisioning (ASP) Grid, Interaction Grid, Knowledge Grid and Utility Grid^[11]. Compute Grid provides a platform to connect distributed computing resources consisting of desktop, server and high performance computing systems to achieve high performance computational power. Application Service Provisioning (ASP) Grid focuses on providing access to remote applications, modules and libraries hosted on data centers or Computational Grids^[11]. Interaction Grid will provide architecture to connect distributed audio-visual equipments to provide high performance virtual meeting room. Knowledge Grid aims towards knowledge acquisition, processing, management and provide business analytics services driven by integrated data mining services. Data Grids provide the access to distributed data resources and their management. Data Grids, therefore primarily deal with providing services and infrastructure for distributed data intensive applications. The fundamental features of Data grids are provision of secure, high performance transfer protocol for transferring large datasets and a scalable replication mechanism for ensuring distribution of data. Utility Grid focuses on providing all the Grid services including compute power, data and services to end-users as IT utilities on a subscription basis and the infrastructure necessary for negotiation of required Quality of Service (QoS), establishment and management of contracts and allocation of resources to meet competing demands from multiple users and applications^[11,12].

Today, we have many open sourced Grid Toolkits available over the internet. These toolkits are providing resources to construct Grids painlessly in an organization. The most popular Grid Middleware among the researcher is Globus^[13] by Globus Alliance, a group of like minded researchers from various universities in US. Many Grids have been implemented with Globus Toolkit in the past 10 years. Globus was the first grid toolkit available in the internet to deploy grid services. Other popular grid toolkits are Alchemi^[14], COSM P2P Toolkit^[15], Gridbus^[16], Avaki, Unicore^[17], Vishwa^[18], etc.

Levels of grids: Grids can be implemented using open source toolkits such as Globus, Alchemi, NetSolve, etc at different levels namely Cluster Grid, Campus Grid and Global Grid. Cluster Grid is the simplest grid, which connects the LAN computers through a network to provide a high performance computing environment. Usually Cluster Grid will be accessed only by a team or a project. Campus Grids are the interconnected Cluster Grids which could provide access between multiple teams or projects in an organization. Global grids are the grids which are linked with cluster grids and campus grids across multiple organizations.

DESIGN CONCEPTS OF ENTERPRISE GRIDS

Enterprise or desktop grids are the grids implemented in an organization by connecting windows based desktop machines behind a single firewall without sacrificing security issues^[19]. The desktop grids can provide virtual super computers through synchronized un-utilized and idle desktop resources with the aim of constructing distributed Virtual Organizations (VOs).

In an enterprise grid computing scenario, the participating desktop machines are divided into two classes. They are worker nodes or executor nodes and a Master node or Manager node. While a Manager node is considerably a high end PC or a server class machine, executor nodes are the normal desktop PCs which provide computational resources in-terms of CPU life cycles, RAM and data storage etc. Executor nodes are attached to the Manager with a set of pre-defined resource sharing rules^[19]. Manager node acts as (i) a coordinator that schedules the grid services and jobs to the executor nodes (ii) a resource broker to find the computational resources for the grid jobs (iii) a collector to get back the executed results and (iv) an agent to channelize the results to the grid users.

While deploying a desktop grid the following design concepts such as centralized resource management, secured data communications, system efficiency, fault tolerance, system performance and inter-operability should be given a careful thought^[19].

Centralized resource management: The computational resources in a grid environment should be centralized into the manager node. The manager node should act as an intelligent system agent to control and coordinate the grid resources.

Secured data communication: The grid environment should protect the data integrity of the executor nodes. Thus well defined security rules should be implemented

upon the data transfer among the executor nodes. Other than the grid jobs, data should not be revealed and transferred to the grid users and the participating nodes.

System efficiency: The manager node must be efficient to collect and pool the under utilized computational resources such CPU cycles and data storage to convert the grid infrastructure into a virtual super computer, opaque to the grid users.

Fault tolerance: System failures in a network are expected and un-avoidable. In such incidents, the grid system should act intelligently to manage the system failures. The grid services in the failed machines should be transferred to other nodes to achieve predefined results.

System performance: The system performance should not be sacrificed to run user applications.

Inter-operability through web-services: Inter-operability is an important feature for the grid environments as many grid jobs may need other programming environments and cross platform grids. Recently e-Science applications have started using global grids for better performance and cost effectiveness. Web-services based grid standards like OGSi, WSRF are helpful to deploy inter-operable grid and e-Science applications in the global grids^[19,20]. Microsoft Net framework proves a simplified deployment for XML based web-services. This simplifies the efforts needed to construct Web Services in windows based Grid environment.

IMPLEMENTATION OF COMPUTATIONAL GRID SERVICES USING ALCHEMI

Alchemi (www.alchemi.net)^[19] is a Microsoft.Net based open-source Grid tool kit which can be deployed in windows based enterprise grid environments. It has been developed under Gridbus (www.gridbus.org) project at Melbourne University, Australia. The following are the features of Alchemi, defined by the Alchemi team.

- Internet-based clustering of desktop computers without a shared file system
- Integration of clusters to create hierarchical, cooperative grids
- Dedicated or non-dedicated (voluntary) execution by clusters and individual nodes
- Object-oriented grid thread programming model (fine-grained abstraction)

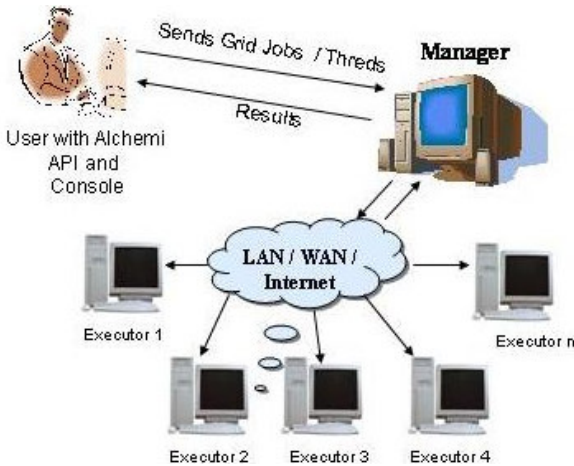


Fig. 2: Alchemi block diagram

- Web-services interface supporting a grid job model (coarse-grained abstraction) for cross-platform interoperability e.g. for creating a global and cross-platform grid environment via a custom resource broker component^[19]

Requirements for alchemi toolkit: Alchemi middleware provides plug and play software packages including Alchemi manager, Alchemi Executor and Alchemi Cross Platform manager. All three categories are built on Microsoft Net framework. They can be easily installed through a step-by-step procedure described by the Alchemi research group at the Melbourne University. All we need is a reasonable machine to support Net framework and backend database tools such as MySQL/MS SQL Server 2000/MSDE2000. The latest Alchemi version 1.0.6 works with Net Framework v.2.0. Figure 2 shows the block diagram of Alchemi^[20].

Alchemi API and console: Alchemi package comes with Alchemi API and Alchemi console^[14]. Alchemi API is an object oriented grid application development tool to deploy Grid threads and grid jobs through XML services. Experience in .Net technology is mandatory to develop Alchemi based grid applications. Alchemi console is a GUI that provides secured access to Alchemi resources from any user PC. It supports three

levels such as Administrator (Super User), Executor (Mid-Level user) and normal user.

Initiative of alchemi based grid services in multimedia university: Multimedia University has a grid environment (<http://grid.mmu.edu.my>) to support the Malaysian National Grid initiative as a Grid Node, constructed based on Globus toolkit based on Linux machines. But the Grid could provide only a minimal computational power as most of the desktops located in the university are windows based. Therefore our study aims to develop a windows based enterprise grid computing environment by utilizing available idle computing power in the Local Area Network of the University. As a pilot run, we decided to develop a simple computational grid environment with a few worker nodes and a master node. Alchemi Manager was installed in a desktop (considerably a high end machine) as a windows desktop application and the Alchemi executors were installed in the dispersed locations in the intranet within the different gateways. This provides the environment to test the efficiency of the grid jobs and the grid threads transferred among the executor nodes. It may be noted that Alchemi executors can be installed either as a dedicated node or as a non-dedicated node^[19]. The nodes in the same domain usually act as dedicated nodes whereas the nodes away from the domain act as non-dedicated nodes, as they run behind the firewall.

Performance evaluation of computational grid services using alchemi (dedicated vs non dedicated nodes): The specifications for Manager and Executor Nodes are shown in Table 1:

We have tested the cost effectiveness of Alchemi by running a test Grid Application in Alchemi API^[14]. The Grid console in the user's machine will send the grid applications to the Manager and the Manager will convert the application into Grid Threads and assign the threads to the available executor nodes. The Alchemi Grid Environment utilizes the idle computing power of the executor nodes and thus the system performance of the executor nodes will not be affected. We chose Pi Calculation program to be tested in our Grid Environment^[14]. This simple Grid application calculates the decimal places of Pi value according to a given range. The results are given in Table 2 and 3.

Table 1: Specifications of grid nodes

Nodes	Machine specifications	Operating system	Same gateway?	Dedicated node?
Manager	P IV, 80GB HDD, 512MB RAM, 3.0GHz	Windows XP Professional	-	-
Executor1	P IV, 80GB HDD, 512MB RAM, 3.0GHz	Windows XP Professional	Yes	Yes
Executor 2	P IV, 60GB HDD, 512MB RAM, 3.0GHz	Windows XP Professional	Yes	Yes
Executor 3	P IV, 60GB HDD, 256MB RAM, 3.0GHz	Windows XP Professional	No	No
Executor 4	P IV, 80GB HDD, 512MB RAM, 3.0GHz	Windows XP Professional	No	No

Table 2: Duration of task completion of pi calculation in dedicated and non dedicated mode

No of digits calculated	Time taken (non-dedicated mode)		Time taken (dedicated mode)	
	1 Executor	2 Executors	1 Executor	2 Executors
500	00:01:10.9688	00:00:40.4531	00:00:25.3751	00:00:10.7866
1000	00:02:57.5469	00:01:30.5156	00:02:01.7969	00:01:46.1467
1500	00:05:58.1563	00:02:55.5156	00:03:20.2656	00:02:10.7866
1750	00:07:55.8438	00:03:58.4063	00:04:30.6554	00:03:08.4585

Table 3: Duration of task completion of pi calculation in combined mode

No of digits to be calculated	Time taken (combined mode)	
	3 Executor (1 dedicated +2non dedicated)	4 Executors (2 dedicated +2non dedicated)
500	00:00:20.2656	00:00:15.5654
1000	00:01:01.0938	00:00:50.5156
1500	00:01:59.2344	00:01:25.2535
1750	00:02:42.9063	00:02:01.9788

During our review study, we tested the same grid application using three different methods. Firstly we executed the grid application with the two non-dedicated nodes which are located at the different gateways. Secondly we tested the application with the two dedicated nodes in same gateway, which the Alchemi Manager located. Finally the grid application was executed with combined executor nodes. The results obtained are shown in Table 1 and 2. The following conclusions were arrived at our preliminary study:

- Alchemi manages the resources effectively across the distributed desktops and workstations
- It performs remote allocation, reservation, monitoring and controlling services among the computing resources
- Dedicated executor nodes perform the grid jobs faster than non-dedicated nodes
- The cost effectiveness in-terms of execution time are greatly achieved by the Alchemi environment during combined mode (dedicated+non-dedicated). Hence, High Performance Computational power could be achieved, if an organization deploys Alchemi based grid computing in the intranet
- The set of security rules protect the privacy of executor nodes
- The Alchemi manager is intelligent enough to divide the Grid Applications into individual units of grid threads and distribute the threads to executors based on the executors computing power

CONCLUSION

This study shows that implementation of enterprise grid environments can be easily deployed using

Alchemi. Hence the research laboratories which have computation intensive applications can be benefited by deploying Alchemi powered grid environments with available resources at a minimal cost. Although Alchemi supports the establishment of Computational Grids, it limits the conception of Data Grids. Further enhancements in Alchemi could lead to overcoming such limitations in the near future.

FUTURE WORK

Since the windows based grid implementation is carried out in a small scale level, it was not possible to run large scale computation intensive applications. Further, we would like to extend the windows based grid into multi-cluster level grid to achieve high computing power. Future work also includes developing an organizational level cross-platform grid implementation, by integrating Globus based cluster and Alchemi based cluster through gridbus^[16] resource broker. We also intend to study the opportunities to run data intensive applications and grid based web-services in Alchemi based grid environments.

ACKNOWLEDGEMENT

We would like to thank the Alchemi team of GRIDS Lab, Melbourne University for providing Alchemi toolkit as an open source grid development toolkit. We thank Dr. Rajkumar Buyya for his comments and suggestions to bring this preliminary research and the study into final shape. The support of Dr. Rajkumar Buyya and other Alchemi team members through online grid users' forum is appreciated.

REFERENCES

1. Foster, I. and C. Kesselman, 1999. Computational Grids, The Grid: Blueprint for a New Computing Infrastructure, Morgan-Kaufman.
2. Foster, I., C. Kesselman and I. Tuecke, 2001. The Anatomy of the Grid: Enabling Scalable Virtual Organizations, Int. J. Supercomput. Appl., 15 (3).
3. SETI @ Home, <http://setiathome.berkeley.edu/>.
4. European Data Grid, <http://eu-datagrid.web.cern.ch/eu-datagrid/>.

5. Welch, V., F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, Meder, L. Pearlman and S. Tuecke, 2003. Security for Grid Services In Twelfth International Symposium on High Performance Distributed Computing (HPDC-12), IEEE Press.
6. Grid Computing, http://searchcio.techtarget.com/sDefinition/0,,sid19_gci773157,00.html.
7. Yang, C.T. and H.C. Ho, 2003. An e-Learning Platform Based on Grid Architecture, *J. Inform. Sci. and Eng.*, 21: 911-928.
8. Stockinger, H., 2001. Distributed Database Management Systems and the Data Grid. 18th IEEE Symposium on Mass Storage Systems and 9th NASA Goddard Conference on Mass Storage Systems and Technologies, San Diego.
9. Grid Architecture, <http://gridcafe.web.cern.ch/gridcafe/gridatwork/architecture.html>.
10. Fran, B., G.C. Fox and J.G.H. Anthony, 2003. Grid Computing: Making the Global Infrastructure a Reality, John Wiley Publishers 119.
11. Venugopal, s., R. Buyya and K. Ramamohanarao, A Taxonomy of Data Grids for Distributed Data sharing, Management and Processing, <<http://www.gridbus.org/reports/DataGridTaxonomy.pdf>>/.
12. Yeo, C.S., R. Buyya, M.D. de Assunção, J. Yu, A. Sulistio, S. Venugopal and M. Placek, 2006. Utility Computing on Global Grids, Technical Report, GRIDS-TR-2006-7, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia.
13. The Globus Alliance: <http://www.globus.org> (Globus Alliance)
14. Alchemi [Net Grid Computing Framework]: <http://www.alchemi.net> (GRIDS Lab, University of Melbourne)
15. The COSM Project: <http://www.mithral.com/projects/cosm/>
16. The Gridbus Project: <http://www.gridbus.org> (Melbourne University)
17. Unicore Forum: <http://www.unicore.org/unicore.htm>
18. Vishwa: Peer-to-Peer Middleware for Grid Computing, <http://dos.iitm.ac.in/Vishwa/> (IIT, Chennai)
19. Luther, A., R. Buyya, R. Ranjan and S. Venugopal, 2005. Alchemi: A NET-Based Enterprise Grid Computing System, Proceedings of the 6th International Conference on Internet Computing (ICOMP'05), Las Vegas, USA.
20. Luther, A., R. Buyya, R. Ranjan and S. Venugopal, 2005. Alchemi: A NET-based Grid Computing Framework and its Integration into Global Grids, Technical Report, GRIDS-TR-2003-8, Grid, Computing and Distributed Systems Laboratory, University of Melbourne, Australia.