Analysis of Services, Challenges and Performance of a Grid

P.K.Suri[1] and Manpreet Singh[2]

[1] Department of Computer Science & Applications
Kurukshetra University, Kurukshetra, India.
[2] Department of Computer Engineering, M. M. Engineering College, Mullana, India.

Summary- Grid computing is a state-of-the-art parallel computing technology which enables the worldwide computers to dynamically share their computing powers and resources to each other. The driving forces for the grid and some key challenges of grid computing are discussed in this paper. This paper also presents a service oriented grid computing model which emphasizes that grid is a special computing system. The proposed model provides a framework for theoretical grid research. Performance analysis shows how grid improves service quality and utilization of the computing resources.

Keywords- Grid Computing, Resources, Queuing System, Service.

1. Introduction

Distributed network computing environments have become a cost-effective and popular choice to achieve high performance and to solve large scale computational problems. Unlike past supercomputers, a cluster or grid computing system can be used as multi-purpose computing platform to run diverse high performance parallel applications. Cluster computing [1][2] environment consist of PCs or workstations that are interconnected using high speed networks and are located at same location, whereas grid computing involves coupled and coordinated use of geographically distributed resources for purposes such as large-scale computation and distributed data analysis [3][4]. Grid computing provides an environment in which network resources are virtualized to enable a utility model of computing. Normally, we hope that grid can gather computing power of many supercomputing facilities together. But because of the limitation of network bandwidth and latency over the internet, the true mathematical computation over the Grid is still a rather elusive goal.

In grid computing, connection on application level among different machine platforms is essential in providing shared applications and distributed resources globally. In order to do this, grid requires the development and deployment of a number of services, including those for resource discovery, scheduling configuration management, security and payment mechanism in an open environment [6][9][13]. A service is a software component that can be accessed via a network to provide functionality to a service requester. This paper proposes a service-oriented grid computing model, which contains five layers and is based on the concept that everything is unified under the realm of service. Service-oriented grid computing offers the potential to provide a fine grained virtualization of the available resources to significantly increase the versatility of a grid. Existing grid packages such as Globus [5], Condor [10] and Nimrod/G [7] already provide essential grid services such as batch scheduling, assignment of processes to nodes, process migration, inter process communication and remote data access. In this paper we have also analyzed the performance of grid in terms of reduction in average waiting time of computing requests.

2. Driving Forces for Grid Computing

Grid applications couple resources that cannot be replicated even at a single site or may be globally located for other practical reasons. These are some of the driving forces behind the inception of grids. Grid computing supports a number of compute-intensive, data-intensive, sensor-intensive, knowledge-sensitive applications ranging from multiplayer video gaming, fault diagnosis in jet engines, biomedical imaging, astrophysics and earthquake engineering.

Industries that are early adopters of grid computing and web services would benefit from reductions in labor costs, outsourcing of operations to suppliers and reduction in other costs as depicted by Table 1. Figure1 shows the expected values of above parameters in some sectors of North Carolina, United States by the end of 2010 due to prospective adoption of grid computing and web services.

Manuscript received July 5, 2007
Manuscript revised July 25, 2007

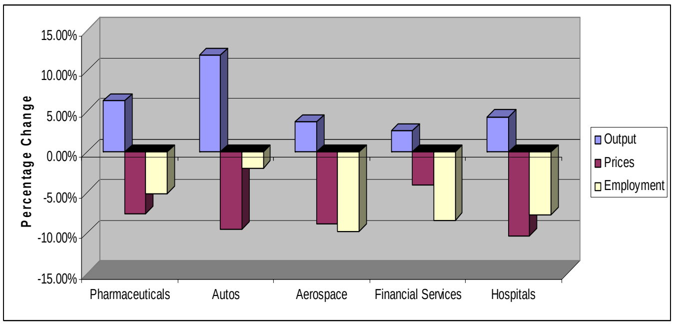

Figure 1: Net Output, Price and Employment Changes in Five North Carolina Industries in 2010

Table 1: Estimated Net Changes in North Carolina's Industries that are Prospective early Adopters of Grid Computing and Web Services by 2010 Compared to values in 2002.

Industry	Change in Output	Change in Price	Change in Employment
Pharmaceuticals	6.3%	-7.6%	-5.1%
Autos	11.9%	-9.5%	-2.0%
Aerospace	3.7%	-8.8%	-9.8%
Financial Services	2.6%	-4.0%	-8.4%
Hospitals	4.3%	-10.3%	-7.7%

3. Architecture of Service-Oriented Grid Computing Model

The proposed model consist of five layers: *Interface* layer, *Development* layer, *Managemen*t layer, *Service Pool* layer and *Resource* layer as shown in figure 2. The *Resource layer* contains all the resources available in a grid. They could be computers, clusters, storage devices, databases and special scientific instruments. All resources have to be visited through services located in the S*ervice Pool layer*. This layer offers the following core services:

- *Scheduling and brokering services* allows grid users to request the allocation of one or more resources for a specific purpose and the scheduling of tasks on the appropriate resources.
- *Data replication services* support the management of grid storage resources to maximize data access performance with respect to metrics such as response time, reliability and cost.
- *Information services* allow grid users to discover the existence and properties of resources. An information service may allow its users to query for resources by name and by attributes such as type, availability or load [11].
- *Software discovery services* discover and select the best software implementation and execution platform based on the parameters of the problem being solved.

The *interface layer* incorporates grid-enabled web portals [12] that give users a familiar, consistent way to interface with complex underlying grid services using a standard web browser. Condor is an environment for scheduling and executing applications on distributed network desktop computers. Condor Dagman portlet is a web enable portlet that interacts with remote web services. The *management layer* support the monitoring of grid resources for failure, intrusion detection, overload and so forth. The Network Weather Service (NWS) is a distributed system that periodically monitors and dynamically forecasts the performance of various network and computational resources over a given time interval [14]. Efficient discovery of services becomes essential for the success of grid applications with the increasing number of grid applications being developed in the grid. The Semantic Web is a vision of new architecture for World Wide Web, in which information is given well defined meaning enabling computers and people to work in cooperation. Semantic Web Services (SWS) promises to provide solution to the challenges associated with automated discovery, dynamic composition and other tasks associated with managing and using service-based systems. Three main approaches have been driving the development of Semantic Web Service frameworks: IRS-II, OWL-S and WSMF. IRS-II (Internet Reasoning Service) [16] is a knowledge-based approach to SWS which evolved from research on reusable knowledge components. OWL-S [17] is an agent -oriented approach to SWS providing fundamentally ontology for describing Web Service capabilities. WSMF (Web Service Modeling Framework)[18] is a business-oriented approach to SWS focusing on a set of e-commerce requirements for Web Services including trust and security.

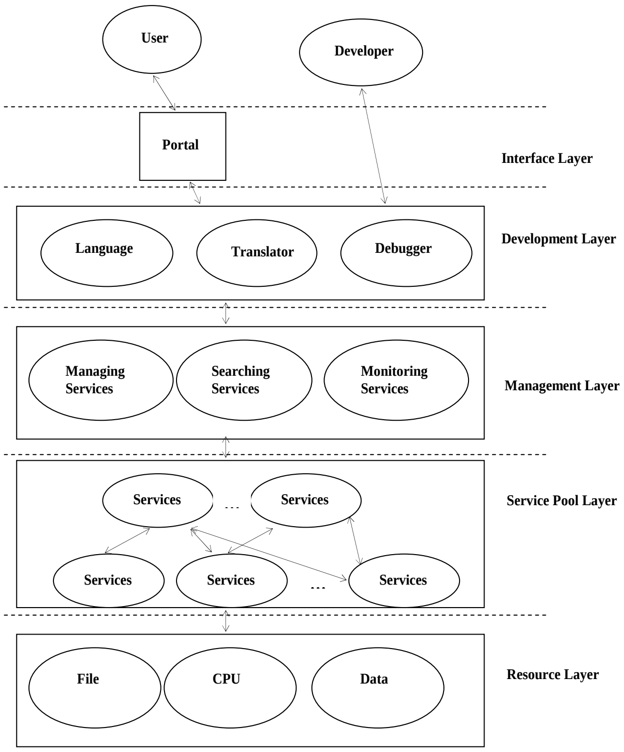

Figure 2: Service-Oriented Grid Computing model

The D*evelopment layer* provides developers with a pragmatic means of implementing a range of services to provide a wide-area application execution environment [5]. Legion [15] provides a software infrastructure so that a system of heterogeneous, geographically distributed, high performance machines can interact seamlessly.

4. Challenges for Making Grid a Reality

The goal of grid computing is to combine resources spanning many organizations into virtual organization that can more effectively solve important scientific, engineering, business and government problems. To achieve this goal, a number of new challenges must be taken into account, for example:

- Currently, a handful of administrators with special knowledge manage their grid infrastructure, configure the separate nodes and preinstall all grid services which are required. When a large number of nodes are added to the grid on a dynamic basis, central administration is no longer feasible.
- The heterogeneity of the grid is increased and reliability of the nodes is decreased due to reboots or crashes caused by the regular users of those nodes. The system itself must be capable of coping with the dynamic topology changes of the underlying network and heterogeneity of nodes.
- A grid might grow from few resources to millions. This raises the problem of potential performance degradation as a grid size increases. Consequently, applications that require a large number of geographically located resources must be designed to be extremely latency tolerant
- Security is also of vital importance to such an extended grid system. Since the number of users within a system is increased, new security mechanisms are needed to ensure that malicious code can not legitimate services running on the grid [8][13].
- In a grid, part of the challenge while using "desktop" or user controlled resources comes from their relative volatility as compared to their shared and managed counterparts. The owner of a desktop machine typically exercises ultimate control over the processes that run on it, its connectivity to the network and its reboot cycle. While system administrator may go to great lengths to ensure that shared resources are highly available, they rarely can exercise the same degree of control over resources that are assigned to individual users.

5. Performance Analysis

Grid computing integrates the computing equipments of various organizations to provide computing services for all correlative users over the web. Grid computing also improves the service quality or reduces the response time for each computing request. We are making the use of queuing theory to analyze the performance of grid. We assume that a grid consist of m computing facilities which have the same computing power. The average arrival rate λ of computing requests and average service rate μ of computing facilities are negative exponentially distributed. As a result of it, grid becomes a $M/M/m$ queuing system.

From queuing theory, we know that the average queue length of $M/M/m$ queuing system is

$$L_m = \frac{(\rho)^m \rho}{m!\,(m-\rho)^2} p_0$$

where ρ (traffic intensity) = $\frac{m\lambda}{\mu}$

and p_0 (probability that there is no request in the system)

$$= \left[\sum_{k=0}^{m-1} \frac{(\rho)^k}{k!} + \frac{(\rho)^m}{m!} \frac{1}{\left(1-\frac{\rho}{m}\right)} \right]^{-1}$$

Thus, the average waiting time W_{m^*} of computing requests in $M/M/m$ queuing system is

$$W_{m^*} = \frac{L_m}{\lambda}$$

and average waiting time W_m of m independent $M/M/1$ queuing systems is

$$W_m = \frac{W_1}{m}$$, where W_1 is average waiting time of $M/M/1$ system.

Assuming λ = 4 and μ = 5, Table 2 shows the comparison of average waiting time W_m of m independent $M/M/1$ systems and W_{m^*} of $M/M/m$ system.

Table 2: Average Waiting Time Comparison

Number (m)	W_m **in** m $M/M/1$ **system**	W_{m^*} **in** $M/M/m$ **system**
1	0.8	0.8
2	0.4	0.356
4	0.2	0.149
8	0.1	0.057
16	0.05	0.019

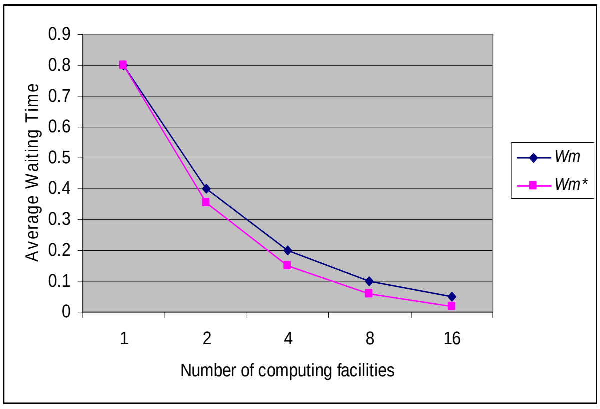

Figure 3: Comparison between W_m and W_{m^*} with respect to m

Figure 3 shows the comparison between W_{m^*} and W_m with respect to number of computing facilities. When the number of these facilities reaches 8, W_{m^*} is almost half of the W_m. From the figure 4 it is evident that more the number of computing facilities in the grid, the quicker is the accretion of speedup.

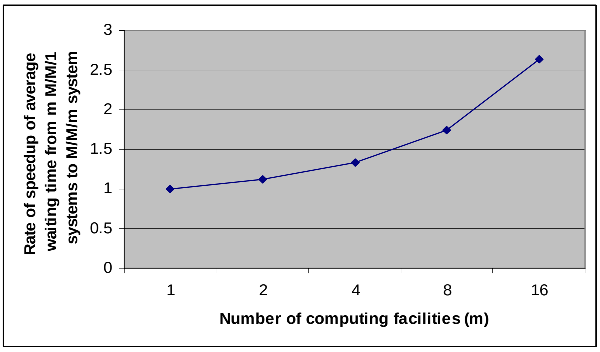

Figure 4: Rate of speedup of average waiting time from m $M/M/1$ systems to $M/M/m$ system

6. Conclusion

Computational Grids provide an emerging distributed computing platform for scientific computing. Grid computing provides highly available resources with a high degree of transparency to users. We have presented a grid computing model where a grid service is the basic unit for modeling resources. This model offers the potential to provide a fine grained virtualization of the available resources to significantly increase the versatility of a grid. We then illustrated the challenges that must be overcome to build grid computing environment. The results show that the collaboration between computing facilities can significantly improve the average waiting time of computing requests.

References:

[1] T. Sterling, “An introduction to PC clusters for high performance computing,” The International Journal of High Performance Computing Applications, 15: 92-101, 2001.

[2] Manpreet Singh and R.B. Patel, “Cluster Computing: A Mobile Code Approach,” Journal of Computer Science, Science Publication, USA, 2: 798-806, 2006.

[3] K. Fujisawa, M. Kojima, A. Takeda and M. Yamashita, “High Performance Grid and Cluster Computing for some Optimization Problems,” Research Report B-400, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo, Japan,2003.

[4] N.Yamamoto,O. Tatebe and S. Sekiguche, “Parallel and Distributed Astronomical Analysis on Grid Datafarm. In Proceeding of 5th IEEE/ACM

International Workshop on Grid Computing: 461-466, 2004.

[5] Ian Foster, “Globus Toolkit Version 4: Software for Service-Oriented Systems,” Journal of Computer Science and Technology, 21: 513-520, 2006.

[6] S. Basu , S. Banerjee, P. Sharma and S. Lee, 2005, “Peer to Peer resource discovery for grids,” In proceeding of 5th International Workshop on Global and Peer-to-Peer Computing, IEEE Computer Society Press: 213-220, 2005.

[7]R. Buyya, D. Abramson, and J. Giddy, “Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid,” HPC ASIA'2000, China, IEEE CS Press, USA, 2000.

[8] Manpreet Singh, Rajneesh Gujral and Sanjeev Rana, “ Securing Grid Using Intrusion Detection System,” In Proceeding of National Conference on Challenges and Opportunities in Information Technology (COIT-2007), RIMT, Mandi Gobindgarh, Punjab: 88-91, 2007.

[9]D. Abramson, R. Buyya, and J. Giddy, “Economic Driven Resource Management Architecture for Computational Power Grids,” In Proceeding of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’2000), Las Vegas, USA, 2000.

[10] Douglas Thain, Todd Tannenbaum, and Miron Livny, “Distributed Computing in Practice: The Condor Experience,” Concurrency and Computation: Practice and Experience, 17: 323-356, 2005.

[11] Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, “Grid Information Services for Distributed Resource Sharing,” In Proceedings of 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10): 181-194, 2001.

[12]A. Natrajan, A. Nguyen-Tuong, M. Humphrey, M. Herrick, B. Clarke and A. Grimshaw, “The Legion Grid Portal,” Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, 2002.

[13] Xiangli Qu, Xueiun Yang, Jingwei Zhong and Xuefeng Lv, “Integration Patterens of Grid Security Service,” In Proceeding of Sixth International Conference on Parallel and Distributed Computing, Application and Technologies (PDCAT 2005): 524-528, 2005.

[14] R. Wolski, N. Spring, J. Hayes, “The network weather service: A distributed resource performing forecasting service for metacomputing ,” Future Generation Computer Systems, 1999.

[15] M.J.Lewis, A.J.Ferrari, M.A.Humphery, J.F.Karpovich, M.M.Morgan, A.Natrajan, A.Nguyen-Tuong, G.S.Wasson and A.S.Grimshaw, “Support for Extensibility and Site Autonomy in the Legion Grid System Object Model,” Journal of Parallel and Distributed Computing, 63: 525-538, 2003.

[16] E. Motta, J. Domingue, L. Cabral, M. Gaspari, “IRS-II: A Framework and infrastructure for Semantic Web Services,” The Semantic Web- ISWC 2003. Lecture Notes in Computer Science, Springer-Verlag, Heidelberg, 2870: 306-318, 2003.

[17] B. Norton, S. Foster and Andrew Hughes, “A Compositional Operational Semantics for OWL-S,” In proceeding of 2nd International Workshop on Web Services and Formal Methods (WS-FM 2005), Versailles, France, 2005.

[18] M. Paolucci, K. Sycara and T. Kawanmura, “Delivering Semantic Web Services,” Technical Report CMU-RI-TR-02-32, Robotics Institute, Carneige Mellon University, 2003.