Performance Evaluation of Grid using Random Scheduling

Saravana Kumar E
Research Scholar
Anna University of Technology
Coimbatore

A.Sumathi, PhD.
Professor & Head
Dept of ECE
ACE

ABSTRACT

The idea behind Grid is seamless, transparent supply of computing and data resources over the Internet when required to end users. The resource management system is the central component of a Grid system. Its basic responsibilities include accepting requests from users, matching user requests to available resources for which the user has permission to use and scheduling the matched resources. Various scheduling algorithms have been proposed in the literature. In this paper, it is proposed to evaluate the performance of one of the popular grid scheduling algorithms – Random scheduling for various scenarios.

Keywords

Grid computing, Scheduling algorithms, Random Scheduling, Performance Evaluation

1. INTRODUCTION

The last five decades saw the computational speed of standalone computers increasing drastically. But newer scientific problems require much higher computational power than available at present when run on supercomputers. This lead to the rise of a distributed computing infrastructure for advanced science/engineering, called Grid [1]. The idea behind Grid is seamless, transparent supply of computing and data resources over the Internet when required. As Grids are suitable for large scale computing and storage, it is extensively used for commercial computing problems too. Grid systems are classified as compute Grids or data Grids; where compute Grids are mainly used for computational purposes and data Grids manage distributed data. Condor, Globus toolkit, Legion are some of the systems used for Grid computing.

A typical Grid architecture is described in layers; the network layer, resource layer, middleware layer, and application and service layer. The purpose of Grid is coordinated resource sharing, thus, resource management system is vital to the Grid system. Resource management system responsibilities include acceptance of requests from users, matching such requests to available resources which the user has permission to use and finally scheduling matched resources [2]. Resources in the Grid are dynamic in nature and are typically heterogeneous. Resources are either hardware or software; processors, bandwidth, memory are hardware resources and applications are software resources.

Resource management and scheduling are key Grid services which are responsible for efficient performance of Grid systems. Issues such as task allocation and load balancing are major challenges for Grids [3]. In a computational Grid, the aim of resource management and scheduling is allocation of user defined jobs effectively by meeting deadlines and using all available resources [4]. Resource Management architecture and services are affected by the type of Grid system where it is deployed.

In conventional computing systems, resource management systems are based on the assumption of total resource control whereas the same is not applicable to the Grid systems. Grid systems face issues such as multiple administrative domains, dynamic resource capabilities and resource heterogeneity [1]. Grid resource management is implemented in multiple layers: Grid Resource: A specific Grid resource that could be accessed remotely and utilized for computation/storage purposes. Resource can be multiprocessors, disk storage or PCs. Grid resource managers handle execution of parallel tasks with maximum resource use [5].

Local Grid: Multiple resources from a single administrative domain form local grids. Its advantage is that resources are connected to high speed networks and fixed bandwidths [6].

Global Grid: Global grid includes all grid resources from various organizations. Agent based resource management with agents organized in a hierarchical manner are most commonly used [6].

Grid scheduling makes all scheduling decisions which involve resources spread over different domains of the Grid system. Schedulers are capable of searching various administrative domains to use a single resource. A job might be a query or computational work requiring a resource; it can range from a simple request for bandwidth to a complex set of applications. Depending on job size, schedulers could assign a single job to multiple resources in the same domain or to multiple sites.

There are three main phases of grid scheduling [7]:

Resource discovery: to generate a potential resources list.

Resource Scheduling: Resource Information being gathered with best resource choice for a job being paired to match application requirements [8].

Job Completion: The job including file staging and cleanup is executed.

The capability to achieve high performance computing and high throughput computing are the two goals of scheduling. The former is achieved by reduction of execution time for every job and is generally used for parallel processing. High throughput computing aims to increase system processing capacity.

In this paper, it is proposed to investigate the time performance for executing different number of tasks with different number of resource cluster for random scheduling algorithm. The rest of the paper is organized as follows: Section II reviews some of the related works in literature, Section III describes the experimental setup and section IV discusses the results and Section V concludes this paper.

2. Related Works

A large number of lightweight jobs are present for Grids but, these application take overall processing including high overhead time and cost in terms of 1) to and from Grid resources, the job transmission and 2) at the Grid resources, the job processing. Muthuvelu et al., [9] introduces a scheduling strategy to perform dynamic job grouping activity at runtime. Job processing granularity size is also presented, in order to allow the job grouping activity to identify the overall jobs that can be processed in a resource, within a specified time. The goal of job grouping strategy is to minimize the total processing time and cost. The small scaled user jobs are grouped into few job groups considering the processing capabilities of available Grid resources by the proposed strategy that decreases the communication overhead time and processing overhead time of each user job.

William M. Jonesy et al., [10] introduced a bandwidth-centric job communication model. Applied across multiple clusters, the proposed model captures the interaction and effect of co-allocating jobs occurred at the same time. This dynamic model is compared with earlier models that utilize a fixed execution time penalty for co-allocated jobs. Many bandwidth-aware co-allocating meta-schedulers are also presented in this paper. The performance of multi-cluster scheduling algorithms is estimated using a bandwidth-centric parallel job communication model that captures the time-varying utilization of shared inter-cluster network resources. Different algorithm were employed and the co-allocating jobs during the allocation a large fraction (85%) on a single cluster gives the greatest performance in mitigating the effect that co-allocated jobs encounter because of the slowdown caused by inter-cluster network saturation, was obtained as a result.

Volker Hamscher et al., [11] investigated typical scheduling structures that occur in computational grids. To these structures, scheduling algorithms and selection strategies are presented and classified. To estimate these features regarding combinations of various Job and Machine Models, discrete-event simulation was performed. The results obtained are discussed in quantitative and qualitative way. The importance of Backfill, for hierarchical scheduling is proven from the simulation results. Using a central job-pool, unexpected results were obtained as First come first serve (FCFS) confirms better performance than Backfill.

Junwei Cao et al., [12] presented “GridFlow” for a workflow management system in grid computing. The GridFlow comprises a user portal and services of both local grid sub-workflow scheduling and global grid workflow management. At the global grid level, the execution, simulation, and monitoring functionalities are allocated. To address new tribulations of workflow management present in a cross-domain and highly dynamic grid environment, a fuzzy timing method is implemented. The results obtained from a case study reveals that local and global grid workflow management are capable to coordinate with each other in order to optimize execution time of the workflow and resolve the other variance of concern.

Mainstream parallel and distributed platforms for high-throughput, high performance and high-availability computing is developed by clusters of computers. Many cluster management systems and schedulers are designed previously to facilitate effective resource management on clusters. The emphasis is mainly to increase the CPU performance and not enhancing the quality of services and the value of utility delivered to the user. Sherwani et al., [13] introduced a novel computational economy driven scheduling system named Libra. Libra is mainly designed based on the users’ quality of service requirements in order to assist allocation of resources. It works as an add-on to the already present resource management system and queuing. To the Portable Batch System, the first version is employed as plugin scheduler. Using the GridSim toolkit, the proposed Libra scheduler is stimulated to perform a complete performance analysis. Comparing with system-centric scheduling strategies, the results obtained increases the utility of the system and user satisfaction by implementing the deadline and budget based proportional resource allocation strategy.

In the grid computing environment, an essential infrastructure is resource management. The two challenges faced to employ these complicated systems are scalability and adaptability. For resource management in a metacomputing environment, Junwei Cao et al., [14] presented a new model implementing a hierarchy of homogeneous agents that is capable of service discovery. Employing various optimization strategies, the performance of the agent system can be enhanced. On the overall performance of the system, a simplified model of the resource management infrastructure is provided as a case study and the simulation results involved reveals the effect of the performance optimization strategies selected.

Different systems developed, allocate resources utilizing market mechanisms, but performance was not adequately studied. Gomoluch, et al., [15] examined the scenarios that outperform a traditional round-robin technique which is obtained by market-based resource allocation with continuous double auctions and with the proportional share procedure, equally. For servers, clients and the market, a model is developed and the simulation results are discussed. Limited to the allocation of independent tasks, the results are: 1) Continuous Double Auction Protocol (CDA) performs best, in a cluster containing homogeneous resources and slight communication delays, 2) Proportional Share Protocol (PSP) performs similar to CDA when the load is low and the difference among the three protocol is less, 3) Round-Robin performs worse than the two market-based protocols, with heterogeneous resources. PSP works better when there are high communication delays. Hence in most cases, CDA works as the best protocol.

3. Experimental Setup

Simulations were carried out in Simgrid framework. The simulations were conducted using 10 clusters, 20 clusters and 30 clusters of resources. The resources are located at different locations connected using high speed switches. The resources are scheduled using Random scheduling algorithm. The number of jobs varies from 50 to 300. The jobs are of uniform size.

In random scheduling, resources are selected randomly and the jobs are allocated. On completion of given task, the resource may be selected for the next task. The jobs are also selected randomly, thus the schedule is non-deterministic. Random scheduling is popular among public grid systems. SETI@home [16] and Condor [17] scheduling algorithms are based on random scheduling.

4. Results

Simulation is conducted for 10, 20 and 30 clusters with varying number of jobs. The time taken to execute 100 jobs of uniform size of 5M and communication size of 0.1M for various numbers of clusters using random scheduling is shown in Figure 1, Figure 2 and Figure 3. Table 1 lists the time taken to execute the task for all the scenarios.

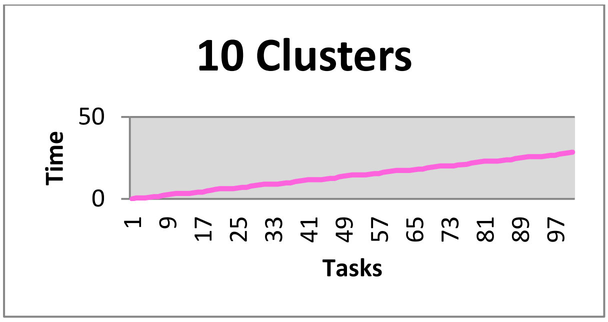

Figure 1: Job processing time for Scheduling 10 clusters to execute 100 jobs

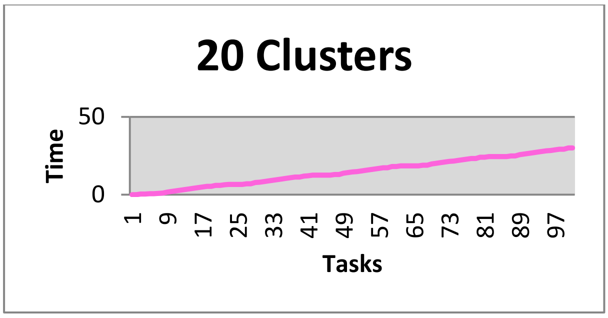

Figure 2: Job processing time for Scheduling 20 clusters to execute 100 jobs

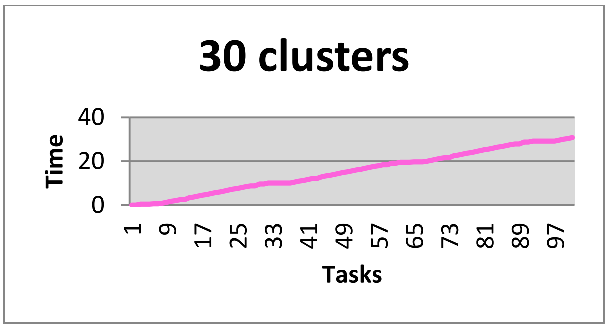

Figure 3: Job processing time for Scheduling 30 clusters to execute 100 jobs

Table 1: Time taken to execute the task

	10 clusters	20 clusters	30 clusters
Number of tasks	Time	Time	Time
50	14.5	14.52	15.64
100	28.5	30.17	30.79
150	42.51	44.45	47.81
200	56.51	60.17	63.34
250	70.52	74.39	78.49
300	84.52	90.3	95.71

.

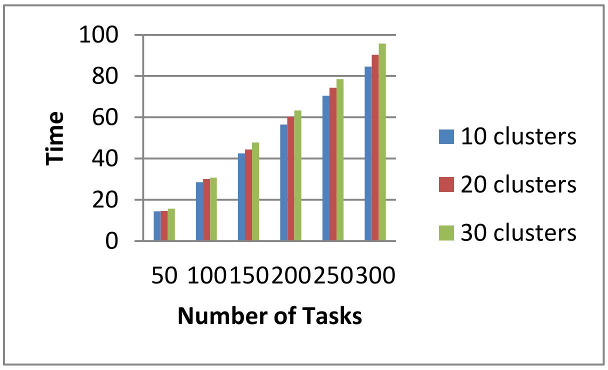

Figure 4: Comparison of time taken in three scenarios

5. Conclusion

In this paper, it is proposed to study the performance of Random scheduling for various scenarios. Simulation is conducted for 10, 20 and 30 clusters with varying number of jobs from 50 to 300. The tasks assigned to the grid were of uniform size. From Figure 4, it is seen that the job processing time for scheduling increases with the increase in number of tasks. Further work is required to be done in the area of improving the timing and proposing a scheduling algorithm.

6. REFERENCES

[1] Foster and C. Kesselman. The grid in a nutshell. pages 3–13, 2004.

[2] Ch. Kandagatla. Survey and taxonomy of grid resource management systems. University of Texas, Austin.

[3] R. Subrata, A. Y. Zomaya, and B. Landfeldt. Artificial life techniques for load balancing in computational grids. Journal of Computer and System Sciences, 73(8):1176–1190, 2007.

[4] C. Grosan, A. Abraham, and B. Helvik. Multiobjective evolutionary algorithms for scheduling jobs on computational grids. ADIS International Conference, Applied Computing, Salamanca, Spain, Nuno Guimares and Pedro Isaias (Eds.), 2007.

[5] D. P. Spooner, J. Cao, J. D. Turner, H. N. Lin Choi Keung, S. A. Jarvis, and G. R. Nudd, "Localised Workload Management Using Performance Prediction and QoS Contracts", in Proc. of 18th Annual UK Performance Engineering Workshop, Glasgow, UK, pp. 69-80, 2002.

[6] J. Cao, S. A. Jarvis, D. P. Spooner, J. D. Turner, D. J. Kerbyson, and G. R. Nudd, "Performance Prediction Technology for Agent-based Resource Management in Grid Environments", in Proc. of 11th IEEE Heterogeneous Computing Workshop, Fort Lauderdale, FL, USA, 2002.

[7] Jennifer M. Schopf, "A General Architecture for Scheduling on the Grid", special issue of JPDC on Grid Computing, April, 2002.

[8] David Fernández-Baca. "Allocating modules to processors in a distributed system", IEEE Transactions on Software Engineering, 15(11):1427-1436, November 1989.

[9] N. Muthuvelu, J. Liu, N. L. Soe, S.r Venugopal, A. Sulistio and R. Buyya, A Dynamic Job Grouping-Based

Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids, Proceedings of the 3rd Australasian Workshop on Grid Computing and e-Research (AusGrid 2005), Newcastle, Australia, January 30 – February 4, 2005.

[10] W. M. Jones, W. B. L. III, L. W. Pang, and D. C. S. Jr. Characterization of bandwidth-aware meta-schedulers for coallocating jobs across multiple clusters. The Journal of Supercomputing, 34(2):135–163, 2005.

[11] Hamscher, V., Schwiegelshohn, U., Streit, A. and Yahyapour, R., Evaluation of Job-Scheduling Strategies for Grid Computing. in 7th International Conference of High Performance Computing, (Bangalore, India, 2000)

[12] Junwei Cao, Stephen A. Jarvis, Subhash Saini and Graham R. Nudd, GridFlow: Workflow Management for Grid Computing In Proc. of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 198-205, 2003.

[13] J. Sherwani, N. Ali, N. Lotia, Z. Hayat, and R. Buyya, "Libra: A computational economy based job scheduling system for clusters," in Int. J. Software: Practice and Experience, May 2004, vol. 34, pp. 573–590

[14] J. Cao, D. J. Kerbyson, and G. R. Nudd, "Performance Evaluation of an Agent-Based Resource Management Infrastructure for Grid Computing", in Proc. of 1st IEEE/ACM Int. Symp. on Cluster Computing and the Grid, Brisbane, Australia, pp. 311-318, 2001.

[15] J. Gomoluch and M. Schroeder. Performance evaluation of market-based resource allocation for grid computing. Concurrency and Computation: Practice and Experience, 16(5):469–475, 2004.

[16] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, M.Lebofsky, "SETI@home-Massively distributed computing for SETI," Comp. in Sci. and Eng., v3n1, 81, 2001.

[17] M. Litzkow, M. Livny, M. Mutka, "Condor - A Hunter of Idle Workstations," Proc. of 8th Intl. Conf. of Distributed Computing Systems, 1988