

Task Scheduling in Grid Computing: A Review

Manjot Kaur Bhatia

*Professor, Department of Information Technology
Jagan Institute of Management Studies, Delhi, India.*

Abstract

Grid computing is a category of high performance computing environment facilitates to solve large scale computational demands. It involves various research issues such as resource management, task scheduling, security problems and information management. Task scheduling is an important element of parallel, distributed computing and grid computing. The main aim of task scheduling in grid computing is to increase system throughput, performance and also to satisfy resource requirements of the task. Grid scheduler in grid computing environment is an application responsible for allocating the suitable resources to independent tasks to maximize the system utilization.

Keywords: Grid Computing, Grid scheduling, Task scheduling, Grid issues, Scheduling algorithms

1. INTRODUCTION

Grid is an integration of heterogeneous resources, termed to be an ideal infrastructure if it is able to provide different types of resources such as: processing units, storage units and communication units (Foster, Kesselman, 1999). However, in reality grid implementations usually work on the collaboration of certain type of resources. Generally grid can be following types:

- i Computational grid: High performance computing needs hardware and software with high end computational capabilities to meet the processing demands of complex scientific problems. A computational grid shares processing power as the main computing resource among its nodes and provides dependable, consistent, pervasive and inexpensive computational facilities to tasks. It provides

computational power to process large scale jobs, to better utilize resources and to satisfy requirement for instant access to resources on demand.

- ii **Data grid:** Data grid can be viewed as the storage component of a grid environment assembled from portions of a large number of storage devices. It acts as a massive data storage system that deals with the storage, sharing and management of large amount of distributed data. Data grid helps scientists and engineers to access local or remote data to carry out extended calculations for applications that require large amount of distributed data.

2. GRID APPLICATIONS

This high end technology can be used to solve computationally complex scientific, mathematical and financial problems (Joseph, 2004):

- i In medical profession doctors need to analyze and compare large amount of images coming from various resources and work on problems like drug discovery.
- ii Team of meteorologists can utilize grid environment for analysis and security of petabytes of climate data and for weather forecasting.
- iii Problems like disaster response, urban planning and economic modeling are traditionally assigned to national governments.
- iv Engineers and accountants share resources to analyze terabytes of structural data.

Understanding the requirement of the above listed grid computing applications, some of the common needs of these applications can be summarized (Joseph, 2004):

- i Application partitioning that involves breaking the problem into discrete pieces
- ii Discovery and scheduling of tasks and workflow
- iii Data communications distributing the problem data where and when it is required
- iv Provisioning and distributing application codes to specific system nodes
- v Results management assisting in the decision processes of the environment
- vi Autonomic features such as self-configuration, self-optimization, self-recovery, and self-management

3. GRID SCHEDULERS

Grid scheduler is an application responsible for managing the selection and execution of jobs on suitable machine available in Grid system. The important functions performed by schedulers are: Managing large queue of global jobs, partitioning of jobs into tasks and to schedule parallel execution of tasks, providing best match to jobs from the available

resources, advance reservation of resources for jobs, storing jobs and data related to jobs, converting global jobs into local jobs, service-level agreement validation and enforcement, examining job executions and status, rescheduling and corrective actions of partial failover situations, maintain accounting records for all the jobs and transactions. These Grid schedulers then form the centralized, hierarchical and decentralized structure (Rodero, Guim & Corbalan, 2009)

- i In centralized scheduler , all tasks are sent to one entity in the system. This entity acts as a central server or central scheduler and need to have the knowledge of all the resources with more control on resources. The main drawback with this scheduler is its lack of scalability and centre of failure. Performance and availability of this type of scheduling system depends upon the accessibility and speed of the scheduler.
- ii The hierarchical scheduler, is organised on different levels having a tree structure (Kurowski, 2008). The meta scheduler at the top access larger set of resources through the local schedulers at lowest level in the hierarchy. Here, local schedulers have the data of the resources. Hierarchical schedulers can handle the problem of scalability and the single point of failure up to some extent as compare to centralized scheduling.
- iii The decentralized scheduler has multiple components that work independent and collaborate for obtaining the schedule (Lu, 2007). In decentralized or distributed scheduling each site in the grid has local scheduler that manages its computational resources. There is no central entity to control the resources. The job scheduling requests by the grid users or local users are managed and processed by the local schedulers or transferred to other local scheduler. Therefore, in comparison to centralized scheduler decentralized scheduler provides better reliability and fault tolerance but absence of global scheduler that maintains the records of all resources and jobs could results in low efficiency.

The available Grid Resource Management system like: Condor(Raman et.al., 1998), Globus(Czajkowski,1998), NetSolve(Casanova et. al. ,1997), Nimrod/G(Buyya et. al., 2000), AppLeS(Casanova et. al., 2000) uses different task scheduling approaches. The Condor uses centralized scheduler and designed to improve overall throughput of the system in a controlled network environment. Its scheduling algorithm does not consider any QoS requirement of tasks. The AppLeS scheduling algorithm focuses on efficient co-location of data and experiments as well as adaptive scheduling. The Nimrod uses decentralized scheduler and its scheduling approach is based on predictive pricing model and Grid economy model. The Netsolve has decentralized scheduler and scheduling approach focuses on fixed application oriented policy considering soft QoS.

4. TASK SCHEDULING IN GRID COMPUTING

Task scheduling is an essential component of parallel, distributed computing and grid computing. Task scheduling in parallel computing aims to reduce the turnaround time of jobs while task scheduling in grid computing targets to maximize resource utilization. Many researchers have proposed scheduling algorithms for parallel system (Zhou et al., 1997, Feitelson et al., 1997, Krallmann et al., 1999) but scheduling in grid computing is more complicated. Therefore, various researchers (Hamscher et al., 2000, Li et al., 2005, Ernemann, 2002) have shown interest in it. Krauter et al. (2002), surveyed and analyzed grid resource management based on classification of scheduler organization, system status, scheduling and rescheduling policies. The aim of task scheduling in grid computing is not only to find an optimal resource to improve the overall system performance but also to utilize the existing resources more efficiently. Many researchers have proposed several task scheduling algorithms in grid environment. Queue based task scheduling like First Come First Served (FCFS), uses queues where tasks are stored until they are scheduled for execution. FCFS schedules tasks according to their submission order and checks the availability of the resources required by each task. The scheduling criteria in FCFS do not consider execution times of the submitted tasks that lead to low utilization of the system resources, because the waiting task cannot be executed if older tasks are in the queue, even if the resources it requires are all available (Techiouba et. al, 2007). Task scheduling systems such as Sun Grid Engine, (SGE) (Gentzsch, W, 2001), Condor (Thain, D., Tannenbaum, T. and Livny, M, 2002), gLite WMS (Burke, S., Campana, S., Peris, A., Donno, F., Lorenzo, P., Santinelli, and Sciaba, 2007), GridWay (Huedo, E., Montero, R. and Llorente, I., 2001), Grid Service Broker (Venugopal, S., Buyya, R. and Winton, L., 2004) etc., are all queueing systems. These systems use queues to store the tasks submitted to the scheduler until they are scheduled for execution.

Tasks scheduling algorithms based on heuristic approach can be divided into two categories online mode and the other batch mode. In online mode, whenever a job arrives to the scheduler it is allocated to the first free machine. In this method, the arrival order of the job is important. Each task is considered only once for matching and scheduling. In case of batch mode, the jobs are collected in a set and are examined for mapping at prescheduled times called mapping events. This independent event uses heuristic approach to make better decision. This mapping heuristics do better task/resource mapping because the heuristics have the resource requirement information for the meta-task, and know the actual execution time of a larger number of tasks. Several heuristics approaches like Min-min, Max-min, UDA and Suffrage are proposed for scheduling independent tasks.

- i **UDA (User Defined Assignment):** UDA (Fidanova, S., Durchova, M., 2006) assigns each task in an arbitrary order, to the machine, with the best expected

execution time for that task, regardless of the machine availability. It assigns too many tasks to a single grid node. This leads to overloading and the increases the response time of the tasks.

- ii **OLB(Oppportunistic Load Balancing):** OLB (Maheswaran, M., Ali, S., Siegel, H. J., Hensgen, D., Freund, R. F., 1999) assigns each task in an arbitrary order, to the next available machine, regardless of the task's expected execution time on that machine. This algorithm produces the poor results as it is not considering the expected execution time.
- iii **Min-Min Algorithm:** Min-Min (Wu, M., Shu, W., Zhang, H., 2000) begins with the batch of all unassigned tasks(UT) and is divided into two phases. In the first phase, the set of minimum expected completion time for each task in UT is found. In the second phase, the task with the overall minimum expected completion time from UT is chosen and assigned to the corresponding machine. This process is repeated until the machines are assigned to all the tasks. However, the Min- Min algorithm is unable to balance the load well as it usually does the scheduling of small tasks initially.
- iv **Max-Min Algorithm:** Max-Min (Maheswaran et. al., 1999) differs from Min-Min in second phase, where tasks with overall maximum expected completion time from UT is chosen and assigned to corresponding machine. In comparison to Min-Min, Max-Min gives priority to longer task and schedules it first.
- v **Suffrage Algorithm:** In Suffrage algorithm, difference between the minimum and second minimum completion time for each task is calculated and this value is defined as suffrage value. The task with maximum suffrage value is assigned to the machine with minimum completion time.

Scheduling is a process to find the optimal solutions, some characteristics of the nature can be used to find the optimal solution for Grid scheduling from large solution set. Some of the nature inspired tasks scheduling algorithms are:

- i **Genetic Algorithm(GA):** GA (Braun et. al., 2001) is a technique used for large solution space. It works on a population of chromosomes for a given set of UT where each chromosome represents a possible solution. The initial population can be generated either randomly from a uniform distribution or by applying other heuristic algorithms, such as Min-Min. Generating initial population by Min-Min is called seeding the population with a min-min chromosome.
- ii **Simulated Annealing (SA):** SA (Zomaya, A. Y. and Kazman, R, 1999 and Liu, Y., 2004), is an iterative technique which is based on the physical process of annealing. It considers only one possible mapping for each unassigned task at a time. Simulated annealing uses a procedure similar to thermal process of obtaining low-energy crystalline states of a solid that probabilistically accept poor solutions

in an attempt to obtain a better search of the solution space based on a system temperature. The temperature is the total completion time and better solution is the optimal task machine mapping.

- iii **The Genetic Simulated Annealing(GSA):** GSA(P. Shroff, D. Watson, N. Flann, R. Freund, 1996) heuristics is a combination of the GA and SA techniques. GSA follows the procedures similar to the GA. For the selection process, GSA uses the SA cooling schedule and system temperature.
- iv **Tabu search (Tabu):** Tabu search(Falco, I. Balio, R., Tarantino, E. and Vaccaro, R., 1994 and Glover, F. and Laguna, M., 1997) is a strategy that saves previous regions of the solution space and avoid repeating a search near the areas that have already been searched. A mapping of meta-tasks uses the same representation as a chromosome in the GA approach. Tabu search is implemented beginning with a random mapping as the initial solution, generated from a uniform solution.

These algorithms have several advantages and have some drawbacks also. The algorithms GA, SA and GSA are difficult to implement. The heuristic algorithms proposed for task scheduling in (Braun et al., 2001) and (Ritchie, 2003) depend on static environment of system load and the expected value of execution times. The scheduling algorithm proposed by Li, Yang in 2006 describes that the task can search very large spaces of candidate and will be moved from one machine to another machine based on the solutions, it increases the traffic in the grid system that leads to degradation in performance. The paper proposed by Yan et al. in 2005 considers communication cost and different ant agents.

CONCLUSION

Grid computing is useful in various ranges of applications such as medical, weather forecasting, engineering, research and others. Task scheduling and resource scheduling are the important areas of research in grid computing. This paper discusses the task scheduling algorithms proposed by various researchers. The algorithms proposed by researchers can be categorized into heuristic approach algorithms and nature inspired algorithms. The main goal of task scheduling algorithms is to minimize the execution time of each job or to improve the processing capacity of the available resources. The advantages and disadvantages of the algorithms are discussed. The future work will include a new scheduling algorithm to optimize the waiting time of the jobs.

REFERENCES

- [1] Buyya, R., Abramson, D., & Giddy, J. (2000, May). Nimrod/G: An architecture for a resource management and scheduling system in a global

- computational grid. In *High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International Conference/Exhibition on* (Vol. 1, pp. 283-289). IEEE.
- [2] Braun, T. D., Siegel, H. J., Beck, N., Bölöni, L. L., Maheswaran, M., Reuther, A. I., ... & Freund, R. F. (2001). A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed computing*, 61(6), 810-837.
 - [3] Burke, S., Campana, S., Peris, A. D., Donno, F., Lorenzo, P., Santinelli, R. and Sciaba, A. (2007). *gLite 3 user guide*. Worldwide LHC Computing Grid.
 - [4] Fidanova, S., & Durchova, M. (2006). Ant algorithm for grid scheduling problem. In *Large-Scale Scientific Computing* (pp. 405-412). Springer Berlin Heidelberg.
 - [5] Casanova, H. and Dongarra, J., (1997). Netsolve: A Network Server for Solving Computational Science Problems, *The International Journal of Supercomputing Applications and High Performance Computing*, Vol 11, Number 3, pp 212-223, 1997.
 - [6] Czajkowski, K., Foster, I., Karonis, N., Kesselman, C., Martin, S., Smith, W., & Tuecke, S. (1998, January). A resource management architecture for metacomputing systems. In *Job Scheduling Strategies for Parallel Processing* (pp. 62-82). Springer Berlin Heidelberg.
 - [7] Foster, I., & Kesselman, C. (1999). The Globus project: A status report. *Future Generation Computer Systems*, 15(5), 607-621.
 - [8] Falco, I., Del Balio, R., Tarantino, E., & Vaccaro, R. (1994, June). Improving search by incorporating evolution principles in parallel tabu search. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence.*, Proceedings of the First IEEE Conference on (pp. 823-828).
 - [9] Lu, K., Subrata, R. and Zomaya, A.Y.(2007). On the performance driven load distribution for heterogeneous computational grids. *Journal of Computer and System Sciences*, Vol. 73, Issue 8, pp. 1191-1206.
 - [10] Rodero, I., Guim, F., and Corbalan, J., (2009). Evaluation of coordinated grid scheduling strategies. In proceedings of the 11th IEEE international conference on High Performance Computing and Communications(HPCC) (pp. 1-10). Washington,DC,USA.
 - [11] Kurowski, K., Nabrzyski, J. , Oleksiak, A., Węglarz, J.(2008). Multicriteria approach to two-level hierarchy scheduling in Grids. *Journal of Scheduling* 11(5), 371-379.
 - [12] Techiouba, A., Capannini, G., Baraglia, R., Puppini, D. and Pasquali, M.(2007). Back_illing strategies for scheduling streams of jobs on computational farms. In *CoreGRID Workshop on Grid Programming Model, Grid and P2P Systems Architecture,Grid Systems, Tools and Environments*. Springer.
 - [13] Zhou, X., Dai. Y.S. and Rana, X. (July,2007) 'Dual-Level Key Management for secure grid Communication in dynamic and hierarchical groups', *Future*

- Generation Computer Systems, Science Direct, Vol. 23 No.6, pp.776-786.
- [14] Hamscher, V., Schwiegelshohn, U., Streit, A., & Yahyapour, R. (2000). Evaluation of job-scheduling strategies for grid computing. In *Grid Computing—GRID 2000* (pp. 191-202). Springer Berlin Heidelberg.
 - [15] Gentzsch, W. (2001). Sun grid engine: Towards creating a compute power grid. In *Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on* (pp. 35-36). IEEE.
 - [16] Thain, D., Tannenbaum, T. and Livny, M.(2002). Condor and the Grid. In Fran Berman, Geo_rey Fox, and Tony Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons Inc.
 - [17] Huedo, E., Montero, R., & Llorente, I. M. (2005). The GridWay framework for adaptive scheduling and execution on grids. *Scalable Computing: Practice and Experience*, 6(3).
 - [18] Venugopal, S., Buyya, R., & Winton, L. (2004, October). A grid service broker for scheduling distributed data-oriented applications on global grids. In *Proceedings of the 2nd workshop on Middleware for grid computing* (pp. 75-80). ACM.
 - [19] Wu, M. Y., Shu, W., & Zhang, H. (2000, May). Segmented min-min: A static mapping algorithm for meta-tasks on heterogeneous computing systems. In *Heterogeneous Computing Workshop* (pp. 375-375). IEEE Computer Society.
 - [20] Maheswaran, M., Ali, S., Siegal, H. J., Hensgen, D., & Freund, R. F. (1999). Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. In *Heterogeneous Computing Workshop, 1999.(HCW'99) Proceedings. Eighth* (pp. 30-44). IEEE.
 - [21] Zomaya, A. Y., & Kazman, R. (1999). Simulated annealing techniques. In *Algorithms and theory of computation handbook* (pp. 33-33). (M. J. Atallah, Ed.), pp. 37-1_37-19, CRC Press, Boca Raton, FL.
 - [22] Shroff, P., Watson, D. W., Flann, N. S., & Freund, R. F. (1996, April). Genetic simulated annealing for scheduling data-dependent tasks in heterogeneous environments. In *5th Heterogeneous Computing Workshop (HCW'96)* (pp. 98-117).