Study on Task Scheduling in Grid Computing

Ms.N.RUBA MCA., M.Phil., Assistant Professor in Computer Applications
Bon Secours College for Women, Vilar Bypass, Thanjavur -613006
Email_id : rubaanand17@gmail.com

Abstract

Grid computing is focused on creating an efficient framework. Now a day's Grid computing is a high performance computing environment to facilitate solve large scale computational problems. This paper explains about various research issues such as task scheduling, security issues, resource management and information management. Task scheduling is an important element of parallel, distributed computing and grid computing. The scope of task scheduling in grid computing is to increase system performance and to satisfy resource requirements of the particular task. Grid scheduler in grid computing environment is an application responsible for allocating the suitable resources to independent tasks to maximize the system utilization and performance.

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

1. INTRODUCTION

The Grid is an integration of heterogeneous resources; it is an ideal infrastructure, if it is able to provide different types of resources such as: processing units, storage units and communication units. In a reality grid implementations usually work on the collaboration of various type of resources. Generally grid can be following two types:

Computational grid: In a high performance computing needs hardware and software with high end computational capabilities to meet the processing demands of complicated 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. Computational grid provides computational power to process large scale of jobs, to better utilize resources and to satisfy requirement for instant access to resources on demand.

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.

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

- 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.

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

3. GRID SCHEDULERS

In a Grid scheduler, application is responsible for managing the selection and execution of jobs on suitable machine available in grid computing system. The main 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 task scheduling in grid computing *Task Scheduling. A Review* 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 the job executions and status, rescheduling and corrective actions of partial failover situations, maintain accounting records for all the jobs and transactions. The Grid schedulers then form the centralized, hierarchical and decentralized structure.

- In centralized scheduler, all tasks are sent to one entity in the system. This entire entity acts as a central server or central scheduler and need to have the knowledge of all the resources with much control on resources. The drawback with this scheduler is its lack of scalability and centre of failure. The Performance and availability of this type of scheduling system depends upon the accessibility and most of the speed of the scheduler.
- In a hierarchical scheduler is organized on different levels, its having a tree structure. The meta scheduler at the top access larger set of resources through the local schedulers at lowest level in the hierarchy. Here, the local schedulers have the data of the resources. Hierarchical schedulers can handle the various problem of scalability and the single point of failure up to some extent as compare to centralized scheduling.
- The decentralized scheduler has multiple components that work independent and collaborate for obtaining the schedule. In a 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. In a job scheduling requests by the grid users or local users are managed and processed by the local schedulers or transferred to other local scheduler. In a 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.

In a available Grid Resource Management system, the Condor, Globus, NetSolve, Nimrod/G, AppLeS uses different task scheduling approaches. The uses centralized scheduler and designed to improve overall throughput of the system in a controlled network environment. The 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

In a Task scheduling is an essential component of parallel, distributed computing and grid computing. Task scheduling in parallel computing goals to reduce the turnaround time of jobs while task scheduling in grid computing targets to maximize resource utilization. Now a day many researchers have proposed scheduling algorithms for parallel system, but scheduling in grid computing is more complicated. Therefore, various researchers have shown interest in it, surveyed and analyzed grid resource management based on classification of scheduler organization, system status, scheduling and rescheduling policies. The main goal 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.

Most of the researchers have proposed several task scheduling algorithms in grid environment. The Queue based task scheduling like First Come First Served (FCFS), uses queues where tasks are stored until they are scheduled for execution. The FCFS schedules tasks according to their submission order and checks the availability of the resources required by each task. The 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. The previous tasks are in the queue, even if the resources it requires are all available. The following task scheduling systems are Sun Grid Engine, (SGE), Condor, gLite WMS, Grid Way, Grid Service Broker etc., are all queuing systems. Above systems use queues to store the tasks submitted to the scheduler until they are scheduled for execution.

The Tasks scheduling algorithms based on heuristic approach can be divided into two categories online mode and the other batch mode. In the online mode, whenever a job arrives to the scheduler it is allocated to the first free machine. The arrival order of the job is most important. In each task is considered only once for matching and scheduling. In the case of batch mode, the jobs are collected in a set and are examined for mapping at prescheduled times called mapping events. The independent event uses heuristic approach to make better decision. The mapping heuristics is do better task or 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. The following heuristics approaches like Min-min, Max-min, UDA and Suffrage are proposed for scheduling independent tasks.

- **UDA(User Defined Assignment):** UDA assigns each task in an arbitrary order, to the machine, with the best expected *Task Scheduling in Grid Computing: A Review* execution time for that task, regardless of the machine availability. The UDA assigns too many tasks to a single grid node. This leads to overloading and the increases the response time of the tasks.
- **OLB(Opportunistic Load Balancing):** OLB 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.
- **Min-Min Algorithm:** Min-Min begins with the batch of all unassigned tasks(UT) and is divided into two phases. In a first phase, the set of minimum expected completion time for each task in UT is found. In a second phase, the task with the overall minimum expected completion time from UT is chosen and assigned to the corresponding machine. This kind of process is repeated until the machines are assigned to all the tasks. This algorithm is unable to balance the load well as it usually does the scheduling of small tasks initially.
- **Max-Min Algorithm**: Max-Min differs from Min-Min in second phase, where tasks with overall maximum expected completion time from UT is chosen and assigned to corresponding machine. The comparison between Min-Min and Max-Min gives priority to longer task and schedules it first.
- **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. This task with maximum suffrage value is assigned to the machine with minimum completion time.

The Scheduling is a process to find the optimal solutions, the 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:

Genetic Algorithm(GA): GA is a technique used for large populations space. This technique works on a population of chromosomes for a given set of UT where each chromosome represents a possible solution. This initial population can be generated either randomly from a uniform distribution or by applying other heuristic algorithms, such as Min-Min. The generating initial population by Min-Min is called seeding the population with a min-min chromosome.

Simulated Annealing (SA): SA is an iterative technique which is based on the physical process of annealing. SA 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 total completion time and better solution is the optimal task machine mapping.

The Genetic Simulated Annealing(GSA): GSA heuristics is a combination of the GA and SA techniques. GSA follows the procedures similar to the GA. In a selection process, GSA uses the SA cooling schedule and system temperature.

Tabu search (Tabu): Tabu is a strategy that saves previous regions of the solution space and avoids repeating a search near the areas that have already been searched. The mapping of meta-tasks uses the same representation as a chromosome in the GA approach. It 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. They are GA, SA and GSA are very difficult to implement. In heuristic algorithms proposed for task scheduling on static environment of system load and the expected value of execution times. The scheduling algorithm describes that the task can search very large spaces of candidate and will be moved from one machine to another machine on the solutions, it increases the traffic in the grid system that leads to degradation in performance. The paper proposed is considers communication cost and different ant agents.

CONCLUSION

The Grid computing is useful in various ranges of applications such as medical, weather forecasting, engineering, research and others. The task scheduling and resource scheduling are the important areas of research in grid computing. This paper illustrated 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 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. Future work will include a 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 *Task Scheduling in Grid Computing: A Review* 1713 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. InEvolutionary 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., Puppin, D. and Pasquali, M.(2007). Back_lling 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. a