Asian Journal of Information Technology 13 (2): 89-92, 2014
ISSN: 1682-3915

Exploring on Budget Constraint Scheduling Algorithms in Grid Computing

[1]I. Stephie Rachel and [2]S. Devi Priya
[1]Department of Computer Science and Engineering,
P.S.R. Rengasamy College of Engineering for Women, Sivakasi, Tamil Nadu, India
[2]Department of Computer Science and Engineering,
Tejaa Sakthi Institute of Technology, Coimbator, Tamil Nadu, India

Abstract: A grid is widely adopted to offer real solutions for industry and e-Science which must be interoperable and standardized. The economic based scheduling gives the opportunity to the users to prefer the deadline and budget which is improves the quality of service requirement needed for the applications.

Key words: Heterogeneous Distributed System, reliability, deadline, budget, quality

INTRODUCTION

Grid computing is the federation of heterogeneous computer resources used at the same time, to solve a single problem by sharing computer power and data storage capacity over the Internet. In 1998, the grid a blueprint for a new computing infrastructure defined "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities". Grid scheduling is used to make the scheduling decisions involving resources over multiple administrative domains. Grid scheduling involves three main phases such as resource discovery which generates a list of potential resources; information gathering about those resources and selection of a good set of them and job execution which executes the job on the resources. The scheduling of tasks to different processors in order to finish all the tasks as fast as possible while making sure the tasks is executed reliably. The tasks of an application may be dependent or independent. For maximum system reliability, failure rate of a system during executing an application should be minimized and the reliability cost should be minimized. Reliability of the grid system depends on the failure rate of the processors and the links between them. The failure rates can be derived from grid's resource's profiling, system log and statistical prediction techniques (Tang *et al*., 2012). Reliability of a system with expect to a task set as the probability that the system can run the task set without any failure (Srinivasan and Jha, 1999). Reliability of computational hardware, software and data resources that comprise the grid and provide the means to execute user applications and reliability of grid networks for messaging and data transport are important and should be met (Dabrowski, 2009). Ignoring grid reliability characteristics can lead to reduced application performance such as schedule length and speedup, due to wasted operations (Dogan and Ozguner, 2005).

Reliability cost, is used as one of the objective functions for task scheduling. Reliability cost is an indicator shows how the system is behaving if the tasks are assigned (Tang *et al*., 2010). The resources of computers owned by individuals or by organizations from several countries are connected to form a single, vast super computer. A grid gathers together resources and makes them accessible in a secure manner to users and applications (Grimshaw, 2009). Grid computing is needed when there is a necessity for huge computing of data and the data is stored in different institutions. Based on the needs, grid is classified into private vs. public, regional vs. global, all-purpose vs. particular scientific problem. The large applications from the users are divided into a set of subtasks and sent to the several resources connected to the main server which is freely available. There are four different classes of grid users such as end users of applications, application developers, system administrators and managers of organizations. After the computations are over at the particular resources, the results are sent back to the main global server. As soon as all the other computations are received by the global server, the result is then provided to the user. The reliability of the grid system is affected by several factors such as hardware failure, software failure.

Corresponding Author: I. Stephie Rachel, Department of Computer Science and Engineering,
P.S.R. Rengasamy College of Engineering for Women, Sivakasi, Tamil Nadu, India

KEY CONCEPTS

Heterogeneous Distributed System: The distributed and networked systems have become highly heterogeneous. This heterogeneity can result from many factors, ultimately impelled by enormous growth of the internet. Heterogeneous System is simply defined as "The Distributed System contains many different kinds of hardware and software working together in concerted fashion to solve problems". Task scheduling on a Heterogeneous System is very challenging.

Reliability: Reliability is a key issue in grid computing system. The reliability of a system depends on both the number of computing servers composing the system and their individual likelihood of failure. The hardware failure, software failure and other sources of failures such as network congestion, malicious attacks are the factors make the grid computing environment unreliable (Maheswaran *et al*., 1999). One way of taking failures into account is to employ a reliable scheduling algorithm in which tasks of an application are assigned to processors to minimize the probability of failures (Tang *et al*., 2010).

Deadline: It is defined as the latest time by which tasks or jobs should be completed. If the given tasks exceeds the deadline, either the performance will be worst or results in catastrophe.

Budget: The budget is given by the user to their application and it also refers to the cost of the resource in a network. Based on the budget value given by the user, the tasks will get priority to schedule it onto the processor.

SURVEY

Venugopal and Buyya (2005) aims to minimizes either the cost or time depending upon the choices of the user based on deadline and in his study. They used Bag of Tasks application for scheduling consists of costs with requests. They built a detailed cost model called economic-based data grid environment on the models for system costs. The data grid environment composts of storage resources, compute resources, datasets, scheduler, data directory and information service. Depending upon the user given deadline, budget and scheduling choice, two objective functions such as cost minimization and time minimization should be met. A deadline and budget constrained scheduling algorithm proposed in this study can be applied to solve the two different objective functions. The algorithm consists of three sections. In the first section, at regular interval, updating the performance data of the computational resources and calculating the allocation. For each data resource, network conditions between data resource and the computational resources is updated. Finally, in this section, the computational resources are sorted either by the cost of the network link or the bandwidth between the compute resource and storage resource. Second section in the algorithm is mapping the jobs to the selected compute resources depending upon the selected objective. The third section in the algorithm is dispatching the jobs while enforcing the deadline and budget constraints.

Buyya *et al*. (2002) aims to allocate the jobs to the resources in such a way that the users requirement should met. They built the broker architecture called economic grid resources broker architecture composts of experiment interface, resource discovery and trading, scheduling flow manager, gridlet dispatcher and gridlet receptor, broker resource list and gridlets queue. The algorithm introduced in this model called DBC cost-time optimization is to optimize for time time, keeping the cost of computation at the minimum. It allocates jobs to the resource using the time-optimization strategy for entire deadline duration since there is no need of extra budget. The algorithm consists of four parts. In the first part, resource discovery will identify all the resources and its capability using GIS (Grid Information System). In the second part, resource trading will identify the resource costs. In the scheduling part, sort the resources by increasing order of cost. Based on the processing costs and budget availability, schedule the jobs onto the resource by calculating job completion time. Final step is to dispatch jobs. When a deadline is relaxed given by the user, the number of jobs processed increases as the budget value increases and all jobs can be completed using first few cheapest resources. When the deadline is tight, there will be high demand for all resources. When the budget value is high given by the user, the broker leases expensive resources to process more jobs within the deadline. When the budget value is low, the number of jobs processed increases as the deadline is relaxed.

Sakellariou *et al*. (2007) aims is to schedule the directed acyclic tasks onto the resources that satisfy the budget constraint and is still optimized for overall time. The algorithm proposed in this study is to satisfy budget constraint by finding the best affordable assignment as possible. The financial cost of the schedule does not exceed the certain budget is the aim of the algorithm. There are two types of algorithms used in this study are loss approach and gain approach. If the budget is less than the cost of schedule, the loss approach is invoked.

The aim of the Loss approach is to schedule with minimum loss in execution time for the largest money savings. If the budget is high than the cost of the schedule then it utilizes all budgets get over. The aim of the Gain approach is to making biggest makespan for the smallest money cost.

Yu and Buyya (2006) stated that to minimize the execution of the jobs while achieving the budget constraints using Genetic algorithm. This proposed algorithm in this study consists of six steps such as problem encoding, initial population, fitness function, genetic operators, time slot assignment and schedule refinement. In the first step, the following conditions should be followed. A task can only be started after all its predecessors have completed. Every task appears once and only once in the schedule. Each task must be allocated to one available time slot of a service capable of executing the task. In the second step, initial population is generated through a random heuristic. In the third step, a fitness function is used to measure the quality of the individuals in the population and it should minimize the execution time. The two functions are cost-fitness and time-fitness. The fourth step is called genetic operation consists of four step. The first step called selection is to compare the new individuals with the earlier generation. An individual with a small value of fitness is better than the one with a large value of fitness. The fittest individuals are retained in the population as successive generations evolve. In second step, crossovers are used to create new individuals on the current population by combining of rearranging parts of the existing individuals. In third step, mutations occasionally occur in order to allow a certain children to obtain features that are not possessed by either parent. In time slot assignment step, time slot assignment process should be developed on the offspring produced by crossover and mutations in order to transfer an offspring string to a feasible solution. The final step called schedule refinement; researchers develop a refinement method to refine the schedule generated by genetic operations. This step may help the genetic algorithm to converge faster, especially when the budget is very low.

Bsoul *et al.* (2006) proposed a framework that supports economic scheduling in grid computing. The users and resources employ Tender/Contract-Net Model to negotiate the prices and deadlines. The negotiated users and resources aim to maximize their performance which is measured by a number of metrics. The framework consists of a number of entities that use Tender/Contract-Net Model. The entities are classified into resources and users. For each negotiation, the job parameters are sent from the user to resources that the user wants to negotiate with. Then, the resources send their bids to the user that in turn accepts one of the sent bids or keeps negotiating. In order to maximize resources' utilities, the resources employ a number of strategies. The first strategy is price strategy employed by the resource to checks the user's price. The second strategy is deadline strategy, the user deadline is checked. The users employ a number of strategies for selecting the resources that will execute their jobs. The first strategy is price strategy, the user starts the negotiation with an initial price that is less than or equal to his maximum acceptable price for the submitted job and that is sent with other job's parameters to all resources. The second strategy is completion time strategy. The user employs this strategy continues the negotiation until he receives a bid with completion time and price that are less than or equal to the user's maximum acceptable completion time and price, respectively or until his maximum acceptable completion time and price can't be increased any further. The third strategy is price and completion time strategy which regards both price and completion time for deciding what resource to select.

Bsoul *et al.* (2012) aims is to design and implement the MICOSim, an event-driven simulator. The four components of a MICOSim are TheSystem, Entity, Entitystrategy and Scenario. The system is the class that is responsible for handling the interaction (communications) between different entities. Entity is an object that can send and receive both jobs and bids. Users, brokers and resources are the three classes of entities are created from Entity. User is the entity that sends jobs and receives bids. The parameters are number of jobs, Job IDs, Job lengths, Jobs' classes, Job costs. The broker sends and receives jobs and bids. Each entity has an Entitystrategy that is an abstract class that contains definitions of methods. Scenario is the class that indicates the specifications of the performed simulation. Rachel *et al.* (2013) considered both deadline and budget are used as a main factor to schedule the tasks on reliable processors. Economic based Reliable Grid Scheduling Architecture proposed in this study consists of Global Server, Grid Information System, Grid Scheduler, Resource Manager, Resource Queue, Dispatcher, Grid Lookup table. After all the information is collected resource manager matches the parameters with the lookup table criteria. Based on the user requirement, the algorithm is executed to achieve the reliability and it satisfies the user requirement.

CONCLUSION

In this study, researchers give a survey on various scheduling algorithms which focuses on deadline and budget.

REFERENCES

Bsoul, M., I. Phillips and C. Hinde, 2006. A framework for economic scheduling in grid computing using tender/contract-net model. Proceedings of the 7th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, June 26-27, 2006, Liverpool John Moores University, Liverpool, UK., pp: 1-6.

Bsoul, M., I. Phillips and C. Hinde, 2012. MICOSim: A imulator for modelling economic scheduling in grid computing. World Acad. Sci. Engin. Technol., 68: 1298-1301.

Buyya, R., M. Murshed and D. Abramson, 2002. A deadline and budget constrained cost-time optimization algorithm for scheduling task farming applications on global grids. Proceedings of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications, June 2002, Las Vegas, Nevada, USA., pp: 1-7.

Dabrowski, C., 2009. Reliability in grid computing systems. Concurrency Computat.: Pract. Exper., 21: 927-959.

Dogan, A. and F. Ozguner, 2005. Biobjective scheduling algorithms for execution time-reliability trade-off in heterogeneous computing systems. Comput. J., 48: 300-314.

Grimshaw, A.S., 2009. What is a grid? Grid Today, Vol. 1, No. 26, December 9, 2002. http://grid.desy.de/docs/GRIDtoday/100840.html.

Maheswaran, M., S. Ali, H.J. Siegel, D. Hensgen and R.F. Freund, 1999. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems. J. Parallel Distrib. Comput., 59: 107-131.

Rachel, I.S., J.S. Raj and V. Vasudevan, 2013. A reliable schedule with budget constraints in grid computing. Int. J. Comput. Applic., 64: 23-28.

Sakellariou, R., H. Zhao, E. Tsiakkouri and M.D. Dikaiakos, 2007. Scheduling Workflows with Budget Constraints. In: Integrated Research in GRID Computing, Gorlatch, S. and M. Danelutto (Eds.). Springer-Verlag, USA., pp: 189-202.

Srinivasan, S. and N.K. Jha, 1999. Safety and reliability driven task allocation in distributed systems. IEEE Trans. Parallel Distrib. Syst., 10: 238-251.

Tang, X., K. Li, M. Qiu and E.H.M. Sha, 2012. A hierarchical reliability-driven scheduling algorithm in grid systems. J. Parallel Distrib. Comput., 72: 525-535.

Tang, X., K. Li, R. Li and B. Veeravalli, 2010. Reliability-aware scheduling strategy for heterogeneous distributed computing systems. J. Parallel Distrib. Comput., 70: 941-952.

Venugopal, S. and R. Buyya, 2005. A deadline and budget constrained scheduling algorithm for science applications on data grids. In: Distributed and Parallel Computing, Hobbs, M., A.M. Goscinski and W. Zhou (Eds.). Springer-Verlag, Berlin, Heidelberg, pp: 60-72.

Yu, J. and R. Buyya, 2006. A budget constrained scheduling of workflow applications on utility grids using genetic algorithms. Proceedings of the Workshop on Workflows in Support of Large-Scale Science, June 19-23, 2006, Paris, pp: 1-10.