International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 4– No.3, September 2012 – www.ijais.org

A Novel Resource Scheduling Algorithm for Computational Grid

Gunjan Aggarwal[1] Meenakshi Kamboj[2] Charanjit Singh[3] Preeti Sharma[4]
Assistant Professor
Department of Computer Science & Engineering
Maharishi Markandeshwar University, Ambala, Haryana, India

ABSTRACT

Grid is an emerging technology for enabling resource sharing and coordinated problem solving in dynamic multi-institutional heterogeneous virtual organizations. However, the management of resources and computational tasks is a critical and complex undertaking as these resources and tasks are geographically distributed. A suitable and efficient scheduling algorithm is needed to schedule user tasks to heterogeneous resources distributed in the grid. We propose a genetic algorithm to solve the scheduling problems in computational grid. The proposed algorithm uses the optimal searching technique of genetic approach and takes different computing capabilities of nodes into consideration. The simulation result shows that proposed algorithm yields better performance when compared with other traditional heuristic approaches.

General Terms

Algorithm, Distributed Computing.

Keywords

Grid Computing, Genetic Algorithm, Resource Scheduling.

1. INTRODUCTION

Grid computing enables the sharing and aggregation of geographically distributed resources and support wide-area distributed computing [1][2]. Resource scheduling is a fundamental issue in achieving high performance in grid computing systems. However, it is a big challenge for efficient allocation and scheduling algorithm design and implementation. Unlike scheduling problems in conventional distributed systems, this problem is much more complex as new features of grid systems such as its dynamic nature and high degree of heterogeneity of tasks and resources must be tackled. In order to efficiently utilize available grid resources and promptly complete tasks assigned to the grid, providing a suitable resource scheduling strategy for the grid computing is necessary [3][4]. The grid systems include a scheduler also termed as “resource broker” that automatically finds the most appropriate resource on which to run any given task that is waiting to be executed. Scheduler also controls the order in which requests for a contended resource are processed, while ensuring certain performance, reliability or security criteria are met. Decision about the assigning of tasks to the resources and finding the best match between the tasks and resources is NP-complete problem [5]. Reduction in makespan is one of the fundamental objectives in scheduling problems in grid systems.

This paper proposes a new resource scheduling algorithm to minimize the makespan. The algorithm uses genetic heuristic and searches the possible couples of the tasks and resources to find the best matching between them. The rest of the paper is structured as follows: Related work is presented in Section 2. Section 3 describes the system model. A retrofitted genetic approach based algorithm for creating a resource plan of submitted tasks is described in Section 4. In Section 5, the performance of proposed algorithm is compared with other traditional heuristic approaches in a series of simulations. Finally, this paper is concluded in Section 6.

2. RELATED WORK

Min-min [6] gives the highest priority to the task which can be completed earliest. The idea behind Min-min is that assign tasks to processors that will execute them fastest. Max-min [7] gives the highest priority to the task with the maximum earliest completion time and is based on the concept of overlapping long-running tasks with short-running ones. For example, if there is only one long task, Max-min will execute many short tasks in parallel with the long task. In contrast, Min-min will execute short tasks in parallel and the long task will follow the short tasks. So, in such a case, Max-min is superior to Min-min. Min-min and Max-min need prediction information on processor speeds and task lengths. Priya et al. [8] presented a genetic algorithm for scheduling the tasks. Along with scheduling of task on best suitable node, their algorithm accomplishes the recovery in grid environment by check pointing the state of the system on a regular basis. This technique allows the recovery to a previous correct state and it moves failed tasks transparently to other resources, so that the task can continue its execution from the point of failure. Yang et al. [9] presented a model for resource scheduling where the system resources can be utilized up to its full extent and the total time of the parallel algorithm can be reduced. For obtaining optimum scheduling, genetic algorithm has been used. During mapping of task set to available resources, the scheduling system may find problem in overload balancing design. To overcome this, a revenue function is proposed to map different resource performance parameters over tasks by means of using weights. Because of dynamic property of grid computing, an intelligent genetic scheduling algorithm is adopted. It finds an optimal allocation of sub task running on each resource to decrease the total run time of parallel algorithm. The fitness function is needed to do proper evaluation. The proposed system resulted in high utilization ratio of system resources. Yu et al. [10] proposed an Evolution based Dynamic Scheduling Algorithm (EDSA) using genetic algorithm to schedule heterogeneous tasks to the appropriate computing nodes with different computing capabilities. In their research work, they investigated scheduling of batch mode. EDSA attempts to find a near optimal schedule in the heterogeneous grid computing environment. The genetic algorithm uses a random approach to generate the initial population. Each chromosome in the population is equal to the number of tasks in the batch. To determine the goodness of a schedule, the fitness value is evaluated by the fitness function. Then for each task in the batch, the computing time and the transmitting time is noted. And to optimize the population further, the heuristic algorithms Min-Min and Max-Min has been used. Furthermore, to control the speed of each mutation, incremental mutation is done. During the crossover phase, in the beginning the two point crossover is done, but due to variation in fitness value, mask crossover is performed. The results generated confirm the reduction of overall completion time of the tasks. In [12], author proposed Job Scheduling using Genetic Algorithm with Multiple QoS (Quality of Service) parameters, for scheduling independent tasks in grid environment. The main objective is to minimize the makespan of the system. The algorithm heuristically generates the initial population and improves the evolutionary process. It increases the search efficiency with limited number of iterations and based on the QoS Satisfaction, the algorithm meets the feasible results. An Enhanced Genetic Algorithm for load balancing in grid is proposed [13]. The algorithm achieved task scheduling with load balancing by reducing the makespan and maximizing the utilization of grid resources. The algorithm uses genetic heuristic and searches the possible couples of the tasks and resources to search the near optimal schedule.

3. SYSTEM MODEL

A grid architecture in figure 1 involves grid scheduler, grid information service (GIS), Job Launching and Monitoring, local resource manager [11].

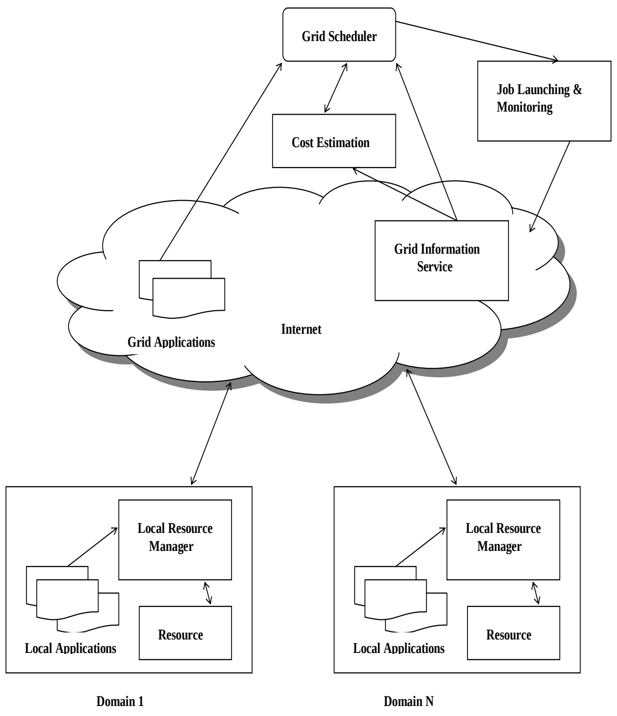

Fig 1: A Common Grid Architecture.

1. **Grid scheduler** works as a broker that would find a appropriate resource for an application on the basis of information given by GIS.
2. **Grid Information Service** maintains information about each resource in grid such as CPU capabilities, network bandwidth, memory size, software availabilities and load of a site in a particular period etc. On the basis of provided information, cost is evaluated and appropriate resources are selected for the jobs.
3. **Job Launching & Monitoring** will implement the final schedule by submitting jobs to the selected resources, provides necessary data and monitors execution of all the jobs.
4. **Local Resource Manager** will perform scheduling inside its domain. It will not only schedule the local jobs but also the jobs send by grid scheduler. If any new resource enters into its domain, it will send the information about that resource to GIS.

4. MODIFIED GENETIC ALGORITHM

The proposed algorithm (MGA) is based on the standard GAs. The method requires an encoding scheme which can represent all legal solutions to the optimization problem. Any particular solution is uniquely represented by a particular chromosome (or schedule). Chromosomes are manipulated in various ways by applying two genetic operators until the termination condition is met. In order for this manipulation to proceed in the right direction, a quality function called fitness function, is required. In proposed algorithm, we assume that there are sufficient independent arriving tasks. The notations used in the description of MGA are illustrated in Table 1.

Table1. Notations Used

Notations	**Meaning**
T	Set of tasks to be scheduled
CN	Set of computing nodes
S	Number of schedules representing chromosomes
ETC	Expected time to compute matrix representing expected execution time of every task on each computing node.
NG	Number of generation

The working of proposed algorithm is as follow:

Step 1. Initialize the basic parameters of algorithm i.e. T, CN, S and Generation=1.
Step 2. Repeat Step 3 for i=1 to S-1
Step 3. Randomly assign every task $t \in T$ to any node $n \in CN$.
Step 4. Initialize ETC matrix.
Step 5. Create the last schedule using Max-min algorithm.
Step 6. Repeat Steps 7 to 9 for i=1 to S
Step 7. Repeat Step 8 for j=1 to CN
Step 8. Compute Completion Time of last task at each node.
Step 9. Compute fitness value of each schedule

$$Makespan = max\{Completion\ Time_k \mid k \in CN\}$$

Step 10. Select S/2 schedules with optimal fitness value i.e. schedules with a smaller fitness value to produce next generation.
Step 11. Repeat Steps 12 to 14 for i=S/2 to S
Step 12. Apply crossover by selecting two nodes from each schedule and interchanging their tasks.
Step 13. Find node with largest number of tasks scheduled (CN_L) and node with smallest number of tasks scheduled(CN_S). If $CN_L = CN_S$ then go to Step 15.
Step 14. Move the last task from CN_L to CN_S to implement mutation operation.
Step 15. Combine these S/2 schedules with schedules selected in Step 10 to produce a new population.
Step 16. Generation =Generation +1
Step 17. If Generation < NG then go to Step 6.
Step 18. Print Fitness value of all schedules.

5. SIMULATION RESULTS AND DISCUSSION

Experiments are carried out to prove MGA can efficiently assign a batch of task to fittest resources and reduce the makespan. Table 2 specifies the simulation environment. The functional code is implemented using simulator built in C language on an Intel core 2 duo, 2 GHz window based laptop. The following assumptions are devised for simulation model: Tasks are mutually independent i.e. there is no precedence constraint between tasks. Tasks are computationally intensive and communications overhead are negligible. Each resource has different computational capability i.e. heterogeneous environment.

Table 2. Simulation Parameter

Number of Generations	10-50
Number of Nodes	10
Number of Tasks	40-100
Number of Schedules	10 – 50
Expected Completion Time (mins.)	50-160

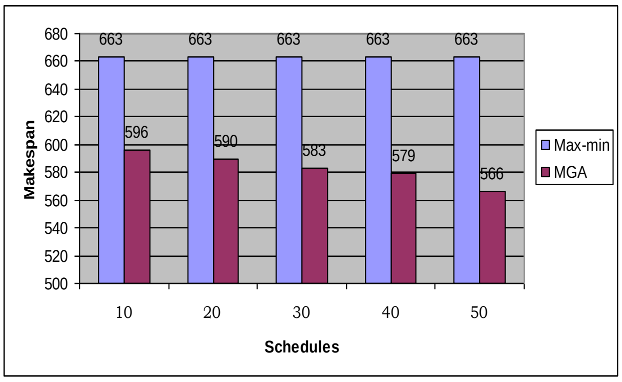

Fig 2: Performance comparison in terms of schedules with T=60 and NG=50

The effect of increase in the number of schedules under Max-min and MGA is illustrated in Fig 2. In Fig 3 the result show that when number of task increases, the time for finishing tasks increased too. And, in all situations MGA perform much better than Max- min.

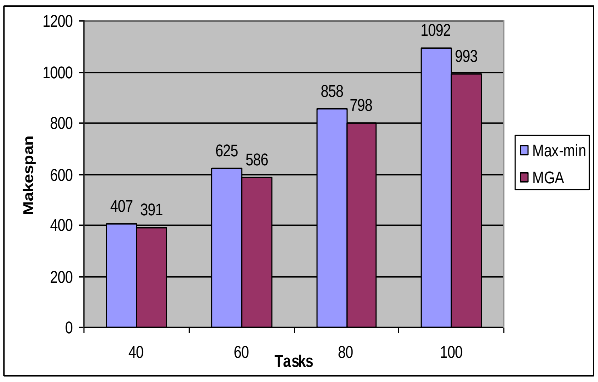

Fig 3: Performance comparison in terms of tasks with S=50 and NG=50

The increase in number of generations improves the performance of MGA to some extent as illustrated in Fig 4. The number of resources is varied from 10 to 30 while keeping other simulation parameters fixed. The average results of execution of the algorithms are demonstrated in Fig 5.

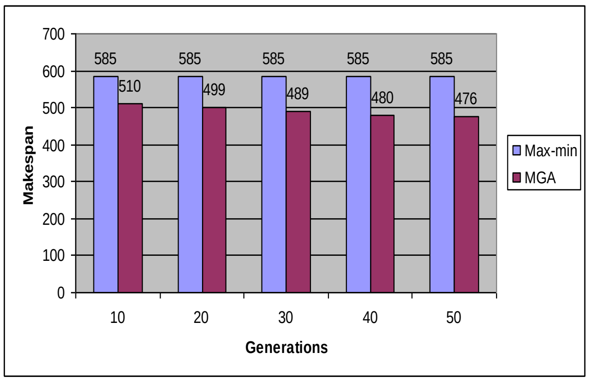

Fig 4: Performance comparison in terms of generation with T=50 and S=40

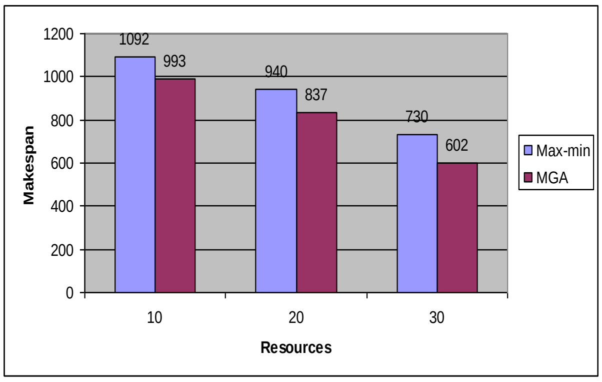

Fig. 5: Performance comparison in terms of resources with T=100

6. CONCLUSION

In this paper, we tried to find a method which optimizes scheduling algorithms for grid environment. Because of the NP-hardness of this problem heuristic methods attract great interests. Genetic Algorithm is used to solve optimization problem by imitating the genetic process of biological organisms. We proposed a scheduling algorithm using genetic approach to search a near-optimal schedule in grid. In order to verify the performance of proposed algorithm, the simulation is performed. The results when compared with traditional approach show a significant reduction in makespan.

7. REFERENCES

[1] Foster and Kesselman, C. 1998. The Grid. Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, Los Altos, CA.

[2] Foster, Kesselman, C. and Tuecke, S. 2001. The anatomy of the grid enabling scalable virtual organizations. International J.Supercomputer Applications, 15(3), 200-222.

[3] Zhang, H., Wu, C., Xiong, Q., Wu, L. and Ye, G. 2006. Research on an Effective Mechanism of Task-scheduling in Grid Environment. In Proceeding of 5th International Conference on Grid and Cooperative Computing, 86-92.

[4] Baghban, H. and Rahmani, M. 2008. A Heuristic on job scheduling in Grid Computing Environment. In Proceeding of 7th International Conference on Grid and Cooperative Computing (GCC'08), 141- 146.

[5] Singh, Manpreet. and Suri, P. K. 2008. QPS Max-Min<>Min-Min: A QoS Based Predictive Max-Min, Min-Min Switcher Algorithm for Job Scheduling in a Grid. Information Technology Journal, ANSInet (Asian Network for Scientific Information), 7(8), 1176 -1181.

[6] Ibarra, O. H. and Kim, C. E. 1977.Heuristic algorithms for scheduling independent tasks on non identical processors. Journal of the Association for Computing Machinery, 24(2), 280–289.

[7] Maleki, E. R. and Movaghar, A. 2011.A Genetic Algorithm to Increase the Throughput of the Computational Grids. International Journal of Grid and Distributed Computing, 4(2), 11-24.

[8] Priya, S. Baghavathi, Prakash, M. and Dhawan, K. K. 2007.Fault Tolerance- Genetic Algorithm for Grid Task Scheduling using Check Point. In Proceeding of 6th IEEE International Conference on Grid and Cooperative Computing (GCC 2007), 676 - 680.

[9] Fan, Yu. tao., Yu, Sheng. chen. and Yang, Xue. 2008. System for Performing Resource Management and Control and Task Scheduling in Grid Computing. In Proceeding of International Symposium on Computer Science and Computational Technology, 648-651.

[10] Yu Kun Ming and Chen Cheng Kwan. 2008. An Evolution-based Dynamic Scheduling Algorithm in Grid Computing Environment. In Proceeding of 8th IEEE International Conference on Intelligent Systems Design and Applications, 450-455.

[11] Y. Zhu, 2003. A Survey on Grid Scheduling System. Department of Computer Science, Hong Kong University of Science & Technology, Technical Report.

[12] Dewaki P. and Valarmathi M.L., 2012. Job Scheduling using Genetic Algorithm with QoS Satisfaction in Grid. European Journal of Scientific Research, 74(2), 272-285.

[13] Goyal Sandip Kumar and Singh Manpreet, 2012. Enhanced Genetic Algorithm Based Load Balancing in Grid. International Journal of Computer Science .9(3), 260-266.