International Journal of Computer Communication and Information System(IJJCCIS)
Vol 7. No.1 2015 Pp. 1-7
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976–1349

QoS Guided Min-Mean Task Scheduling Algorithm for Scheduling Meta-tasks in Grid

Dr.G.K.Kamalam
Kongu Engineering College, India

Abstract

Grid computing is the collection of resources from multiple locations to reach a common goal. Grid environment comprises of heterogeneous and geographically distributed resources. A grid is a hardware and software infrastructure that provides a dependable, consistent, pervasive and inexpensive access to high performance computing resources. The computational power of the distributed grid resources is huge. To utilize the computational power of the grid efficiently, an effective grid scheduling algorithm is proposed in this paper. The proposed QoS Guided Min-Mean Task Scheduling Algorithm algorithm improves the performance of grid system by considering the QoS factors such as network bandwidth. The proposed algorithm provides load balancing, better resource utilization and minimum makespan at the time of scheduling.

Keywords: Qos, Min-Mean, Scheduling, Meta Task

I. INTRODUCTION

Grid achieves the same level of computing power as a supercomputer does, but at a much reduced cost. Grid is like a virtual supercomputer[1]. Distributed computing supports resource sharing. Parallel computing supports computing power. Grid computing aims to harness the power of both distributed computing and parallel computing [2,3].

.

The large-scale heterogeneous resources includes supercomputers, storage systems, workstations, data sources, networks, software, specialized devices are shared by creating a dynamic virtual organizations [4]. Grid computing environment includes the discovery of geographically distributed resources, selecting the resources appropriate for the particular problem, and scheduling of jobs to the resources to ensure overall high performance of the system. Applications may require enormous resources, which often are not available for the user, so a scheduling system is essential to allocate the resources to the input tasks. Managing various resources and task scheduling in highly dynamic grid environment is a challenging and indispensable task. How to effectively match grid tasks with available grid resources is a challenge for a grid computing system because of its dynamic, heterogeneous and autonomous nature.

Task scheduling is a vital and challenging work in heterogeneous computing environment. The problem of mapping meta-tasks to a machine is shown to be NP-complete[9]. The NP-complete problem can be solved only using heuristic approach.

The primary importance is to design an efficient scheduling algorithm for minimizing the total completion time of the tasks.

II. LITERATURE REVIEW

The Fastest Processor to Largest Task First Scheduling Algorithm (FPLTF) is a good representative for Bag-of-Tasks applications. The logic proposed in the FPLTF scheduling algorithm is to schedule jobs according to the workload of jobs and computing power of resources [14].

In Min-min scheduling algorithm, each job will be always assigned to the resource which can complete it earliest in order to spend less time completing all jobs. The Max-min scheduling algorithm is similar to Min-min scheduling algorithm. It gives the highest priority to the job with the maximum earliest completion time [12].

In the On-line mode heuristic scheduling algorithms, Jobs are scheduled as soon as it arrives. Because a grid environment is heterogeneous with different types of resources, on-line mode heuristic scheduling algorithms are more appropriate for grid environment. Dynamic FPLTF Scheduling Algorithm (DFPLTF) is based on FPLTF scheduling algorithm and is modified to make the FPLTF scheduling algorithm more adaptive for grid environment [12].

The simple grid simulation architecture and modified the basic ant algorithm for job scheduling in grid. The scheduling algorithm they proposed needs some information such as the number of CPUs, Million Instructions Per Second (MIPS) of every CPU for job scheduling. A resource must submit the information mentioned above to the resource monitor[15].

Opportunistic Load Balancing (OLB) algorithm assigns each job in random order to the next available machine without considering the job's expected execution time on the machine. Hence, it produces poor makespan. The main aim is to improve resource utilization [1]. The Minimum Execution Time (MET) algorithm assigns each job to the machine that has the minimum expected execution time. It does not consider the availability of the machine and the current load of the machine. The advantages of OLB and MET combined to design new algorithms called as Minimum Completion Time (MCT). The MCT algorithm calculates the completion time for a job on all machines by adding the machine's availability time and the expected execution time of the job on the machine. The algorithm selects the machine with the minimum completion time for executing the job. The MCT considers only one job at a time. The selected machine may have the expected best execution time for any other job [2].

Min-min algorithm starts with a set of all unmapped tasks. The algorithm calculates the completion time of each job on the each machine. It selects the machine that has the minimum completion time for each job and then selects the job with the overall minimum completion time and allocates to the corresponding machine. Again, this process repeats with the remaining unmapped tasks. Compared to MCT, Min-min algorithm considers all unmapped tasks at a time. Max-min algorithm begins with a set of all unmapped tasks. The algorithm calculates the completion time of each job on the each machine. It selects the machine that has the minimum completion time for each job. The algorithm assigns the job with the overall maximum completion time within the set of minimum completion time to the machine. Again the above process repeats with the remaining unmapped tasks. Similar to Min-min, Max-min also considers all unmapped tasks at a time [2]. The Duplex heuristic is literally a combination of the Min-min and the Max-min heuristic algorithms [1,4].

Proposed QoS Sufferage heuristics algorithm schedules tasks to the resources with best sufferage value. The high QoS tasks is scheduled on resource with high QoS provision and the low QoS tasks is scheduled on resource with low QoS provision [13].

Min-mean heuristic scheduling algorithm works in two phases. In the first phase, Min-mean heuristic scheduling algorithm starts with a set of all unmapped tasks. The algorithm calculates the completion time for each task on each resource and finds the minimum completion time for each task. From that group, the algorithm selects the task with the overall minimum completion time and allocates to the appropriate resource. Removes the task from the task set. This process repeats until all the tasks get mapped. The algorithm calculates the total completion time of all the resources and the mean completion time. In phase 2, the mean of all resources completion time is taken. The resource whose completion time is greater than the mean value is selected. The tasks allocated to the selected resources are reallocated to the resources whose completion time is less than the mean value[5,6,7,8].

Resource Fitness Task Scheduling Algorithm (RFTSA) assigns tasks to the resources by identifying the fitness value of the resource [10].

Grid architecture is organized as a collection of clusters with multiple WN in a cluster. The algorithm Best Cluster Decentralized Job Scheduling Algorithm optimizes the completion time of the job by dividing the jobs into subjobs. The subjobs are scheduled to the WN of different clusters in a decentralized grid environment. The algorithm efficiently schedules both the computing-intensive jobs and data-intensive jobs based on the best cluster value [11].

The current research problem in task scheduling is to bring out an efficient QoS guided scheduling algorithm to improve the resource utilization and reduce makespan. The scope of this work is to propose an efficient task scheduling algorithm on the basis of the QoS parameters Network bandwidth of the resource, high QoS task, low QoS task, high QoS resource and low QoS resource. The QoS GMMTSA algorithm efficiently schedules the tasks based on the minimum completion time of the task and the minimum load of the resource.

III. MATERIALS AND METHODS

An application is consisting of ‘n’ independent tasks and a set of ‘m’ heterogeneous resources. The problem of mapping the ‘n’ tasks to the ‘m’ resources in a gird environment is an NP-Complete problem. The heterogeneous and the dynamic nature of the grid is making the scheduling a complicated problem.

A. Existing Research: The previous research work Min-mean and Improved Min-mean provides better scheduling results [5].

Algorithm Min-Mean:

The scheduling of the tasks to the available resources is done in two major phases.

In phase 1, the task is allocated to the resources based on the Min-Min algorithm.

In phase 2,

Compute

= / .

Select resources R_k whose CTk >

Order the resources R_k in the decreasing order of CT_k.

For each tasks scheduled to the selected resources R_k

Reallocate the job to the resource R_j whose CTj >

Calculate = +

Compute = max( ) , 1 ≤ ≤

B.Proposed Work: The mapping of meta-tasks to the resources is done based on the following assumptions.

- Heuristics derive a mapping statically.
- Each resource executes a single independent task at a time.
- The sizes of the tasks and the number of resources are static and known a priori.
- The accurate estimate of the expected execution time of each job on each resource is represented within an ETC matrix of size n*m, where n-represents the number of tasks and m represents the number of resources.
- Task set is represented as T= {T_1, T_2, … ,T_n}.

- Subset of High QoS tasks is represented as T_h
- Subset of Low QoS tasks is represented as T_l
- Resource set is represented as R= {R_1, .. , R_m}.
- Subset of High QoS resources is represented as R_h

• Subset of Low QoS resources is represented as R_l

• ET_{ij}-expected execution time of task T_i on resource R_j.

• TCT_{ij}-expected completion time of task T_i on resource R_j.

• RT_j-ready time of resource R_j.

• = + .

• = max (,) .

- The ETC matrix ETC (Ti, Rj) is computed by the formula

(,) = h /

where Lengthi is the length of the task T_i in MI and Powerj is the processing power of the resource R_j in MIPS.

- The ready time of the resource R_j is the time at which the resource R_j can complete the execution of all the tasks that have been previously assigned to the resource. The ready time of the resource R_j is

$$RT(R_j) = \sum_{i=1}^{n} ETC(T_i, R_j)$$

- The communication time of each task represents the time taken to transfer the input file and the time taken to transfer the output file to the scheduler where the task is submitted. The communication time is calculated by the formula

(,) = / + /

IFS_i = Input file size of task T_i

OFS_i = Output file size of task R_j

BW_j = Bandwidth of the resource R_j.

- The completion time of each task T_i on each resource R_j is calculated by

(,) = (,) + ( )
+ (,)

- The maximum TCT (T_i, R_j) of all the tasks is the overall completion time of all the tasks and is called the makespan.
- Task heterogeneity: Variation in the execution time of the task for a given resource.
- Resource heterogeneity: Variation in the execution time for a particular task among the entire resource.
- The tasks

QoS Guided Min-Mean Task Scheduling Algorithm (QoS GMMTSA):

The proposed algorithm works in two phases.

Phase 1:

For all tasks in the task set T in meta-task, Group the tasks into high QoS tasks and low QoS tasks.

Group the resources into high QoS resources and low QoS resources.

Phase 2:

do until all tasks with high QoS request are mapped.

for each task in the task subset t_h with high QoS find a machine in the high QoS qualified resourse set R_h that obtains the earliest completion time.

The task t_i with overall minimum completion time is selected and scheduled to the selected resource r_j

The selected task t_i is removed from the task set t_h.

end do

Compute

= / .

Select resources R_k whose CTk >

Order the resources R_k in the decreasing order of CT_k.

For each tasks scheduled to the selected resources R_k

Reallocate the job to the resource R_j whose CTj >

Calculate = +

Compute = max( ) , 1 ≤ ≤

do until all tasks with low QoS request are mapped.

for each task in the task subset t_l with low QoS find a machine in the low QoS qualified resourse set R_l and also high QoS resource set R_h that obtains the earliest completion time.

The task t_i with overall minimum completion time is selected and scheduled to the selected resource r_j

The selected task t_i is removed from the task set t_h.

end do Compute = / . Select resources R_k whose CTk > Order the resources R_k in the decreasing order of CT_k. For each tasks scheduled to the selected resources R_k Reallocate the job to the resource R_j whose CTj > Calculate = + Compute = max( ) , 1 ≤ ≤

IV. RESULTS AND DISCUSSION

Every instance consists of 512 tasks and 16 resources. The experimental results are based on the two groups which comprises of Low QoS tasks and High QoS groups. The first group relates to the equal number of Low QoS tasks and High QoS groups. The second group relates to the 70% of Low QoS tasks and 30% High QoS groups and the third group relates to the 30% of Low QoS tasks and 70% High QoS groups. The simulation is carried out for four different set of 512 tasks and 16 resources.

C.Evaluation Parameters:Makespan

Makespan is the important optimization criteria for grid scheduling. Makespan is calculated as

= max (,)

Fig.1 shows that the the graphical representation of the makespan values obtained for the three different groups for QoS Guided Min-Mean Task Scheduling Algorithm and Min-Mean Algorithm. It is evident from the figure that the proposed algorithm QoS GMMTSA provides better makespan than the Min-Mean algorithm.

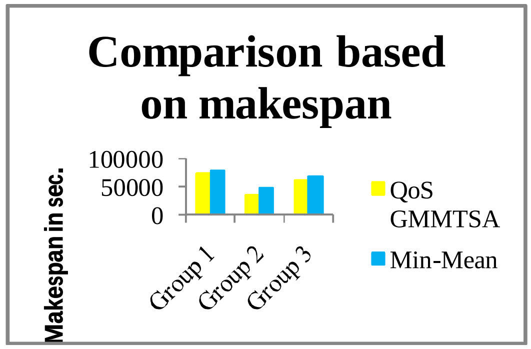

Fig.1. Comparison based on makespan for Case1

Fig. 2, 3 and 4 shows that the the graphical representation of the makespan values obtained for the three different groups and three different cases for QoS Guided Min-Mean Task Scheduling Algorithm and Min-Mean Algorithm. It is evident from the figures 2,3 and 4 that the proposed algorithm QoS GMMTSA provides better makespan than the Min-Mean algorithm.

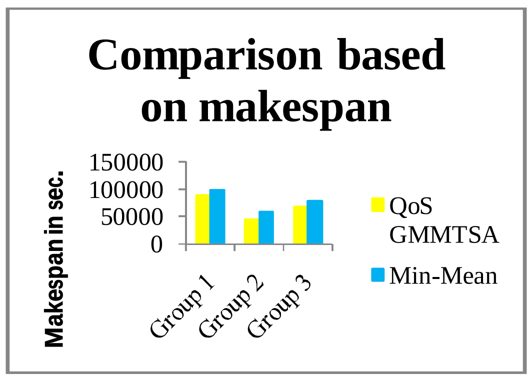

Fig. 2. Comparison based on makespan for Case2

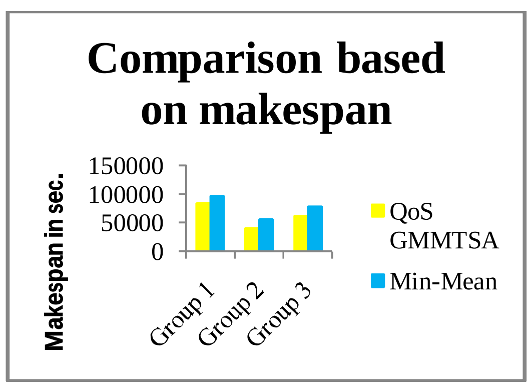

Fig. 3. Comparison based on makespan for Case3

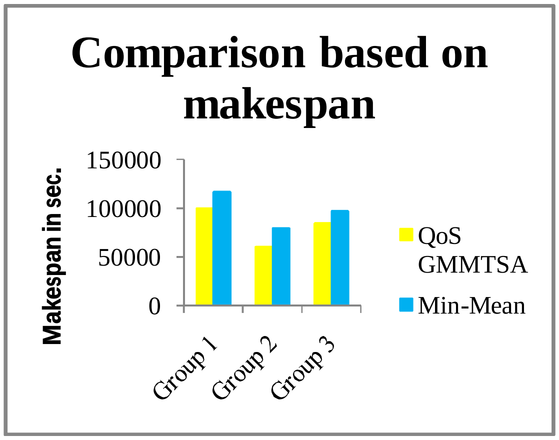

Fig. 4. Comparison based on makespan for Case4

V. CONCLUSION AND FUTURE WORK

In this paper, an efficient QoS Guided task scheduling algorithm is proposed. The proposed algorithm QoS GMMTSA and the Min-mean heuristic scheduling algorithm are tested for different cases for three different groups. The proposed algorithm QoS GMMTSA considers the heterogeneous nature of the tasks and resources, resource heterogeneity, and task heterogeneity while scheduling the jobs. By considering the QoS parameter Network bandwidth of the resource, high QoS task, low QoS task, high QoS resource and low QoS resource, the proposed scheduling algorithm effectively schedules the tasks to the best resources. From section Results and Discussion it is observed that the proposed algorithm QoS GMMTSA achieves better makespan than the existing Min-Mean heuristic scheduling algorithm. The results show that the proposed QoS GMMTSA algorithm reduces the makespan, improves the resource utilization and balances the load across the grid environment.To achieve higher efficiency in scheduling, the proposed algorithm can be improved along with the other scheduling features such as availability of the resources, fault tolerant techniques. The proposed algorithm can be extended to handle data-intensive jobs and to schedule dependent jobs in the future.

VI. REFERENCES

[1] Armstrong, R., D.Hensgen, and T.Kidd 1998. The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-time Predictions. In 7th IEEE Heterogeneous Computing Workshop(HCW'98), pp. 79-87.

[2] Braun, T.D., H.J.Siegel, and N.Beck 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, pp.810-837.

[3] Chang, R-S., C-Y. Lin, and C-F. Lin 2012. An Adaptive Scoring Job Scheduling algorithm for grid computing. Journal of Information Sciences, pp. 79–89.

[4] Foster, I., and C.Kesselman 1999. The GRID: Blueprint for a New Computing Infrastructure. Morgan, Kaufmann.

Freund, R.F., and H.J.Siegel 1993. Heterogeneous Processing. IEEE Computer , 26(6), pp. 13-17.

[5] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2010 a. A New Heuristic Approach:Min-Mean Algorithm For Scheduling Meta-Tasks On Heterogeneous Computing Systems. IJCSNS International Journal of Computer Science and Network Security, Vol.10 No.1, pp. 24-31.

[6] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2010 b. An Improved Min-Mean Heuristic Scheduling Algorithm for Mapping Independent Tasks on Heterogeneous Computing Environment. International Journal of Computational Cognition, Vol. 8, N0. 4, pp. 85-91.

[7] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2011. An Efficient Hybrid Job Scheduling for Computational Grids. International Journal of Computer Applications.

[8] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2012a. New Enhanced Heuristic Min-Mean Scheduling Algorithm for Scheduling Meta-Tasks on Heterogeneous Grid Environment. European Journal of Scientific Research, Vol.70 No.3, pp. 423-430.

[9] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2012b. Novel Adaptive Job Scheduling algorithm on Heterogeneous Grid Resources. American Journal of applied Sciences, pp. 1294-1299.

[10] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2014a. Resource Fitness Task Scheduling Algorithm for Scheduling Tasks on Heterogeneous Grid Environment. Australian Journal of Basic and Applied Sciences, Vol.8 No.18, pp. 128-135.

[11] Kamalam, G.K., and Dr.V.Murali Bhaskaran 2014b. Best Cluster Decentralized Job Scheduling Algorithm for Scheduling Tasks on Heterogeneous Grid Environment. Australian Journal of Basic and Applied Sciences, Vol.8 No.18, pp. 171-177.

[12] Maheswaran.M., S.Ali, H.J.Siegel, D.Hensgen, and R.Freund 1999. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing system. Journal of Parallel and Distributed Computing, pp. 107–131.

[13] Munir. E.U., Li.J, and Shi.S 2007. QoS Sufferage Heuristic for Independent Task Scheduling in Grid. Information Technology Journal, Vol.6 No.8, pp. 1166-1170.

[14] Saha.D., D.Menasce, and S. Porto 1995. Static and dynamic processor scheduling disciplines in heterogeneous parallel architectures. Journal of Parallel and Distributed Computing, pp. 1–18.

[15] Zhang. X., and L.Txang 2005. CT-ACO – hybridizing ant colony optimization with cycle transfer search for the vehicle routing problem. Congress on Computational Intelligence Methods and Applications,pp.6.