An Efficnet Job Scheduling Approach In Grid Environment using Biggest Hole Priority Algorithm

G.Suganya.M.sc.,B.Ed[1]

[1] Mphil.Scholar, Department of Computer Science, KG College of Arts and Science,Coimbatore. Tamil nadu, India.

Abstract— Advance scheduling performs predictions about future network status and about job duration of resources. This is known as meta-scheduling. This is done with the aim of not overlapping executions but without making any physical reservation. Grid resources may vary dynamically since they may fail, join or leave the Grid at any time. Algorithms for meta-scheduling in advance need to be efficient so that they can adapt themselves to dynamic changes in resource availability and user demand without affecting system and user performance. The constantly changing characteristic of the heterogeneous resources is known as dynamic resources. To make it more critical, the jobs that need to be process by these resources are arriving from different length of time and not knowing by the system until the particular jobs arrive to the system. In order to utilize these dynamic resources and jobs optimally, a scheduling strategy should be able to continually adapt to the changes and properly distribute the workload and data amounts scheduled to each node. Gap filling or backfilling is one of the techniques used to optimize First Come First Serve (FCFS) and have been used widely in scheduling area to achieve load balancing. The proposed system introduces Biggest Hole (BH) strategy. It determines which particular machine is suitable to execute the job. It places new job to the particular suitable machines schedule. It Check the current machines schedule whether a suitable gap for the new job exists. Experiemental result provides better result when compare with the exising methods

Keywords— Fragmentation Grid computing,Meta-scheduling in advance, Runtime predictions, Biggest Hole (BH) strategy

I. Introduction

Grids are highly variable systems in which resources may join/leave the system at any time. This variability makes Quality of Service (QoS) highly desirable though often very difficult to achieve in practice. Advance reservation of resources has been identified as a basic requirement to guarantee QoS in distributed systems like Grids. However reservations may not be always feasible. Since not all the Local Resource Management Systems (LRMS) permit them. Apart from that there are some types of resources such as bandwidth which lack a global management entity and make their reservation impossible.

In order to utilize these dynamic resources and jobs optimally a scheduling strategy should be able to continually adapt to the changes and properly distribute the workload and data amounts scheduled to each node. Unfortunately, several researchers have stated that serious difficulty in concurrent programming of a grid computing has occurred in terms of dealing with scheduling and load balancing of such a system, which may consist of heterogeneous computers. The current popular techniques in grid scheduling are schedule-based algorithms that adapting optimization technique such as Genetic Algorithm, Ant Colony Optimization etc. The corresponding authors of these techniques have proof that their techniques are much better compared to the priority rules algorithms. Priority rules scheduling has been widely used in the popular scheduling production systems such as Condor and PBS. Although schedule-based algorithms were proven to be more effective (based on objective the researchers want to achieve), priority rules techniques are still very important because most of the schedule-based algorithms will have to apply priority rules algorithms for initial schedule in their schedule-based algorithms.

It is based on meta-scheduling in advance in Grids which may be defined as the first step of the reservations in advance algorithm where the resources and the time periods to execute the jobs are selected and the system keeps track of the decisions already made and usage of resources but making no physical reservation. It proposed a network-aware Meta scheduling in advance Grid architecture as a possible solution. This architecture takes into account the status of the system in order to make efficient meta-scheduling decisions. However incorporating such mechanisms into current Grid environments is a challenging task because of the resulting resource fragmentation. So job execution request could be rejected due to the fragmentation even if the overall remaining capacity is sufficient to handle it. Therefore it is rather easy to identify fragmentation as an individual reason for the rejection of a single allocation.

The problem found could be several reasons for rejected allocation requests in Grid environments.

- **High utilization**: If there is no vacancy on the resources the meta-scheduler will reject the request.
- **Fragmentation**: Free parts of the resources are shattered in space and time is known as Fragmentation. It is a well known effect in resource allocation which decreases the resource utilization. Certain jobs were rejected due to the fragmentation problems related to the resource allocation policies.

- **Unfavourable previous decisions**: It is possible that previously scheduled job uses the only blocks which fit the new incoming job. So this new job is rejected.

The main objective of this work is to provide QoS in Grids by means of Meta scheduling which makes advance reservations of resources. Then to develop efficient scheduling algorithms to predict the jobs completion time into the resources and to improve utilization of resources at the scheduling process.

II. RELATED WORK

This section describes various literature work regarding scheduling has been studied.

In [1] Dobber et.al surveyed some predictions techniques. Techniques for such predictions include applying statistical models to previous executions of the work proposed in [2] by Dinda and heuristics based on job and resource characteristics [3] by Jin.et.al.

Dinda in [2] shown that although load exhibits complex properties, it is still consistently predictable from past behaviour. Jin.et.al in [3] evaluated a various linear time series models for prediction of CPU loads in the future is presented. In our work, a technique based on historical data is used, since it has been demonstrated to provide better results compared to linear functions.

In [4] Smith.et.al d presented fragmentation which is a well known effect in resource allocation, which decreases the resource utilization. But how the fragmentation can be *quantified* in a system requiring continuous allocations has been hardly analysed.

In [5] Wilsonet.al presented the characteristics of dynamic memory allocators. However, the domain of memory management does not map onto the domain of Grid resources, since the main memory can be considered as homogeneous which is not true for Grid resources. Thus, further research is needed to address fragmentation issues in the Grid meta-scheduling domain.

In [6] Buyya and De Assuncao dealt the reallocation of jobs. This work investigates how the precision of availability information affects resource provisioning in multiple site environments and uses backfilling to perform that provision of resources. However, their scenario does not totally map into ours since we take into consideration not only deadline but also start time constraints for jobs .

III. PROPOSED WORK

Priority rules are referred as Queue-based. Instead of guaranteeing optimal solution, these techniques aim to find reasonable solutions in a relatively short time used for solving scheduling problem based on backfilling techniques. The purpose of backfilling is to improve system utilization of scheduler that used First Come First Serve (FCFS). Although FCFS is a simple policy and have been widely used, it suffers from the low system utilization. This happens because there is a gap between two jobs that make the resource idle. Backfilling improves resource utilization by allowing small job to fill in those gaps. Job which is lower in the queue is moved to the idle machines without delaying the execution of the job at the top of the queue.

Biggest Hole (BH) strategy is to make a large process faster. Large process is the process that usually involved many calculations and huge data that consumes a lot of time. While traditional backfilling intends to fill the holes with smaller jobs and Earliest Gap Earliest Deadline First (EG-EDF) fill the first holes with possible jobs that can fix in, Biggest Hole search for the biggest holes in the queue and match them with any jobs that can fix in. In the case of EG-EDF, if the hole or gap does not match the jobs or too small for the job, the system has to search again and the possibility of the candidate gap to be lost are higher because as time pass by, the candidate gap can be shrinks smaller or even lost from the system. This is not happening with BH technique. BH-PR carries out the same procedure as EG-PR algorithms. Instead of accepting the earliest gap in the queue, BH-PR search for the biggest gap in schedule at that particular time. This is carried out by sorting the gap decreasingly based on the gap size.

The system architecture of the proposed framework is shown in fig 1:

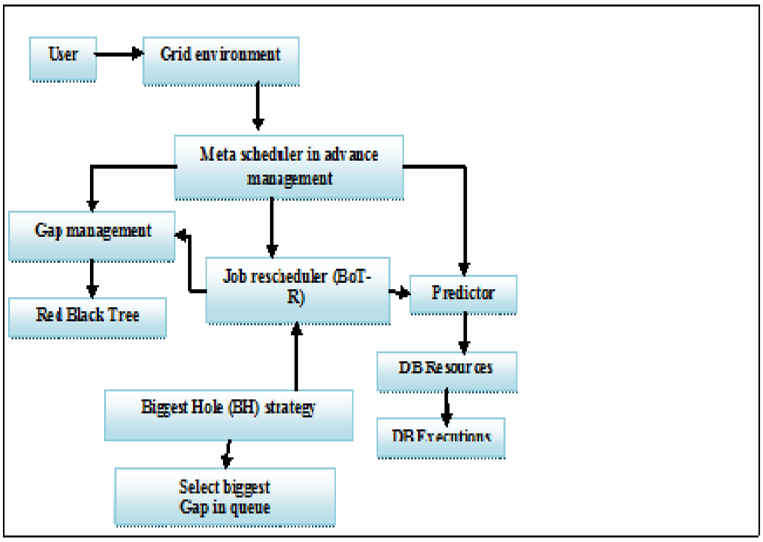


Figure 1: Proposed system achitecture

IV. METHODOLOGY

A. *SCHEDULER IN ADVANCE LAYER (SA-LAYER)*

- Meta-scheduling is done by scheduler in Advance Layer (SA-layer). This layer is an intermediate layer between the users and the on-demand Grid meta-scheduler.
- The SA-layer performs resource discovery and monitoring, job submission and execution monitoring in advance.
- The SA-layer includes predictor such as DB Executions and DB Resources

DB Executions: It stores information concerning previous executions of applications.

DB Resources: It stores information about the status of resources and network over time

B. *GAP MANAGEMENT MODULE*

The It represents the information of red-black trees to find the best slots for each job in an efficient manner. Red-black trees identify feasible idle periods for each arriving job request without having to examine all idle periods.

The usage of the resources is divided into time intervals termed as slots. These time slots divide the day into s intervals to schedule the future usage of resources by allocating the jobs into the resources at one specific time (taking one or more time slots). Thus data structure known as red-black trees is used to keep a trace of the usage of slots.

- If deadline expires for the scheduled job to be executed at a particular resource, that job will be dropped. Hence predictor is deployed.
- Predictor estimates job durations to know how many slots a job will need in a certain resource at a certain time in the future.
- Completion time of jobs in a given resource is estimated by characteristics of the jobs, the power and usage of the CPU of the resources and the network future status. By processing this information about applications and resources, accurate estimations on the completion time of the job in the different computational resources can be performed.

C. *PREDICTIONS OF DURATION OF JOBS*

Predictions for the duration of jobs are calculated by estimating the execution time of the job and the time needed to complete the transfers separately.

Prediction based on execution times:

- It is calculated as the average of previous executions. After that, the prediction on the future status of the CPU of each resource is calculated by means of an exponential smoothing function.
- Finally, the mean execution time is tuned by using the prediction about the future CPU status of each resource.

Prediction based on transfer times:

- Exponential smoothing function calculates the mean bandwidth in the time period between the job start time and its deadline.
- The time needed to complete the transfers is estimated by using the above information along with the total number of bytes needed to transfer

D. *IMPLEMENTATION OF THE BAG OF TASKS RESCHEDULING TECHNIQUE (BOT-R)*

This technique use Exponential Smoothing function to better estimate the time needed to complete the execution of a job in a specific resources along with network transfer rate.

The BoT-R technique is to avoid job allocation failures. This is done by executing jobs in certain interval for replanning process. The job which was executed before the beginning of the interval is excluded from replanning process. Thus there will not be pre-emption of jobs. BoT-R algorithm can be separated into two steps:

1) Estimate if there is any necessity of rescheduling tasks.

2) Perform the BoT rescheduling process for the selected resources, tasks and time intervals.

3) After BoT rescheduling, resource fragmentation must be measured which is used to predict future allocations.

Algorithm 1:

1. It checks several state variables to verify if rescheduling is needed.

If there are more jobs than resources and too few free time slots replanning is done.

- The period of time between the first and the last scheduled job is splitted into intervals of strip slots. Then these subintervals are checked separately so that replanning is only applied on the time period which presents fragmentation. Thus for each subinterval check the information related to load of the system and status of the Gap Management. Else no replanning is done due to insignificant load while resource usage is between the two thresholds (rulow and ruhigh) and lot of free time slots.

2. It calculates in which period of time and over which resources the system needs to apply Algorithm 2.

If [(number of gaps) > (number of resources)] and
[(average size) < maximum gap size (gapSize)]

Reschedule the tasks using Algorithm 2 for the selected time interval .

Algorithm 2:

After estimating which subintervals need task replanning to reduce the fragmentation Algorithm 2 is executed for each of them.
Input: Interval to perform replanning defined by start time and end time of the period.

1) Decide which resources to involve in the replanning process.
2) Adopt filtering process over the available resources so that consider resources which present fragmentation in their allocations within the defined period.
3) Select the jobs scheduled to the resources obtained.
4) Once the list of jobs to replan is obtained, it is sorted by job start time restriction.
5) Finally for each resource selected to be defragmented, the sorted list of jobs is scanned with the aim of allocating as many jobs as possible in each resource to reduce the number of free slots in between contiguous allocations.
6) When it is not possible to allocate any more jobs, the next resource is used for allocating the jobs which are not allocated yet and so on.

E. BIGGEST HOLE (BH) STRATEGY:

Large process is the process that usually involved many calculations and huge data that consumes a lot of time. While traditional backfilling intends to fill the holes with smaller jobs and Earliest Gap Earliest Deadline First (EG-EDF) fill the first holes with possible jobs that can fix in, Biggest Hole search for the biggest holes in the queue and match them with any jobs that can fix in. In the case of EG-EDF, if the hole or gap does not match the jobs or too small for the job, the system has to search again and the possibility of the candidate gap to be lost are higher because as time pass by, the candidate gap can be shrinks smaller or even lost from the system. This is not happening with BH technique. BH-PR carries out the same procedure as EG-PR algorithms. Instead of accepting the earliest gap in the queue, BH-PR search for the biggest gap in schedule at that particular time. This is carried out by sorting the gap decreasingly based on the gap size.

V. EXPERIMENTAL RESULTS

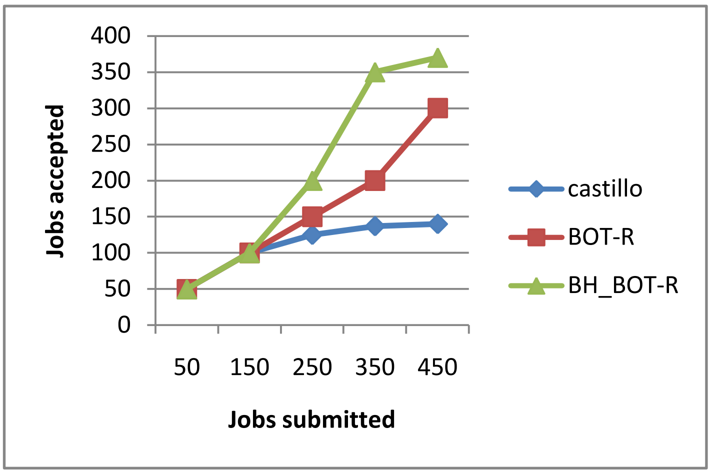
To evaluate the performance of those scheduling techniques several statistics are used. Scheduled job rate is the fraction of accepted jobs, i.e., those whose deadline can be met. QoS not fulfilled means the number of rejected jobs, plus the number of jobs that were initially accepted but their executions were eventually delayed. Thus, their QoS agreements were not fulfilled.

Scheduled job rate:

It is the fraction of accepted jobs, i.e., those whose deadline can be met.

QoS agreements:

If the number of rejected jobs and number of initially accepted jobs were executed without delay. Thus their QoS agreements were fulfilled.



The obtained result for Scheduled job rate and QoS agreements are shown in following graph**.**

Figure 2: Scheduling job rate

Thus the proposed method in figure 2 clearly shows BH_BOT-R accepts a greater number of jobs since there are a greater number of free slots together. On the other hand, it is also highlighted that the Castillo technique presents the worst results, clearly outperformed by the other two techniques.

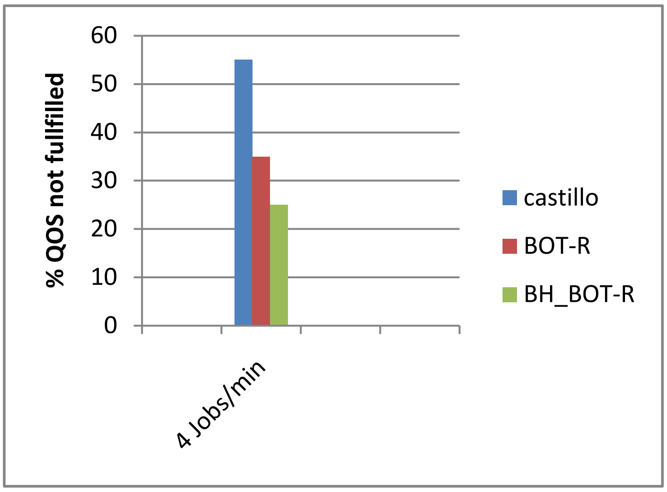


Figure 3.QOS Agreement

The above graph in figure 3 shows that there were submitted 4 jobs per minute until a total of 500 jobs. The graph represents that the QoS not fulfilled statistic (including rejected jobs plus accepted jobs not executed within their deadlines). As this figure depicts, using BH_BoT-R the total number of jobs accepted is increased since the fragmentation is reduced by rescheduling tasks when needed. BH_BoT-R obtains a reduction in the QoS not fulfilled of 40% against Castillo and 20 % against BOT-R.

VI. CONCLUSION

The present work proposes an extension to the framework presented to address the fragmentation and poor resource utilization issues. This new feature lets the system make rescheduling of task already scheduled with the aim of being able to allocate a greater number of jobs by using the resources more efficiently. To do that, the jobs are rescheduled by its start time instead of by its arrival time. Hence the reallocation of those tasks will create less fragmentation into resources. Moreover this reallocation entails several jobs. So it is made in the same way as a BoT. Apart from presenting the improved framework a comparison between whether using the rescheduling strategy is outlined. This comparison highlights the importance of performing this rescheduling process so that a greater number of jobs could be allocated into resources. Thus the QoS perceived by users is improved.

REFERENCES

[1] M. Dobber, R. van der Mei, and G. Koole, "A prediction method for job runtimes on shared processors: Survey, statistical analysis and new avenues," Performance Evaluation, vol. 64, no. 7-8, pp. 755–781, 2007

[2] P. A. Dinda, "The statistical properties of host load," Scientific Programming, vol. 7, no. 3-4, pp. 211–229, 1999.

[3] H. Jin, X. Shi, W. Qiang, and D. Zou, "An adaptive meta-scheduler for data-intensive applications," Intl. Journal of Grid and Utility Computing, vol. 1, no. 1, pp. 32–37, 2005.

[4] W. Smith, I. Foster, and V. Taylor, "Scheduling with advanced reservations," in Proc. of the 14th Intl. Parallel and Distributed Processing Symposium (IPDPS), Washington, USA, 2000

[5] P. R. Wilson, M. S. Johnstone, M. Neely, and D. Boles, "Dynamic storage allocation: A survey and critical review," in Proc. of the Intl. Workshop on Memory Managment (IWMM), Kinross, UK, 1995

[6] M. D. De Assuncao and R. Buyya, "Performance analysis of multiple site resource provisioning: effects of the precision of availability information," in Proc. of the 15th Intl. Conference on High Performance Computing (HiPC), Bangalore, India, 2008.