International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 05 | May-2016 www.irjet.net p-ISSN: 2395-0072

Decision based Meta Scheduler in Grid

Savita Khurana[1], Dr. Tanupreet Singh[2]

[1]*Assistant Professor, Dept. of Information Technology, JMIT,Radaur,Haryana,India*
[2]*Professor, Dept. of Computer Science & Engineering, ACET, Amritsar,Punjab,India*

Abstract - Grid computing allows resource sharing between several entities [9], but to select the appropriate resource to run a specific job remains one of its major challenges. We try to solve this problem using Meta scheduler. It accepts the user request and identifies the efficient resource from the available list of resources in different scheduler. Meta-scheduler keeps the information about all resources. When the job is arrived for execution, meta-scheduler will arrange the jobs in a queue for scheduling in the local scheduler. Meta-scheduler will collect the current data available status from users and resource providers at periodically interval. The meta-scheduler schedule the jobs to different sites, instead of sending the jobs to single site to achieve the better resource utilization and load balancing. A collective effort of a local and meta-scheduler, a better scheduling decision can be taken.

Key Words: **Grid, Meta Scheduler, scheduling, Makespan**

1. INTRODUCTION

The sharing and co-ordination of heterogeneous and geographically distributed resources has become the fundamental capabilities of Grid Computing [9]. To perform application execution in the Grid, proper scheduling of Grid resources is necessary to achieve the best utilization of resources. To achieve this goal in an effective manner, An efficient Grid scheduling system is an essential part of the Grid. Grid Resource Scheduling is defined as the process of making appropriate scheduling decisions involving resources over same or different domains. The grid scheduling system is responsible to select appropriate resources in a grid for user jobs. The grid scheduling problem is multi-objective [10] in nature. Several performance and optimization metrics [11] can be considered to evaluate the performance of a given schedule and performance of overall grid system. Different criteria can be used for evaluating the efficiency of scheduling algorithms but the most important criteria are makespan and flowtime. Makespan is the time when a system successfully completed the task and flowtime is the sum of completion times of the entire task. A Grid scheduler (or broker) takes resource selection decisions but it has no control over the local resources, the resources are distributed, and information about the systems is not frequently updated. Here, schedulers are performing the tasks of job management i.e. allocating resources needed by particular job, divided the large job in to smaller jobs so that they can execute in parallel manner in parallel processing environment. Scheduler also responsible for management of data and service-level management capabilities. Meta schedulers in grid are different from local schedulers because a local scheduler only manages and control a single site or cluster and usually owns the resource. In the literature, a lot of scheduling algorithms were proposed each one has particular features and capabilities.

2. RELATED WORK

Thomas A. Henzinger et al. [1] proposed abstraction refinement approach. The static scheduling problem often arises as a fundamental problem in real-time systems and grid computing. They considered the problem of statically scheduling a large job expressed as a task graph on a large number of computing nodes, such as a data center. Proposed paper use abstraction refinement techniques to solve the large-scale static scheduling problem, abstraction refinement a technique commonly used in formal verification to efficiently solve computationally hard problems. Abstraction refinement based schedulers firstly solve the scheduling problem with abstract representations of the task and resources. Abstract representations are often small and the scheduling of resources can be done rapidly. If the obtained schedule does not meet specified quality conditions i.e. data center utilization or makespan. Then the scheduler refines the job and data center abstractions and, again solves the scheduling problem. Different schedulers on abstraction refinement technique was developed and implemented. These schedulers are used to schedule the task from different computing domains on simulated data centers with different topologies. After that a comparison is performed which compare the speed of scheduling and the quality of the produced schedules with their abstraction refinement schedulers against a baseline scheduler that didn't use any abstraction. At last it was concluded that abstraction refinement based scheduler give a considerable speed-up compared to traditional static scheduling, at a low cost. Proposed approach implements their static schedulers in system that they deployed on Amazon EC2 and comparison is done with Hadoop dynamic scheduler. Paper results indicated that there is great probability for static scheduling techniques. An important assumption behind static scheduling techniques is that the characteristics of the jobs like job duration, object sizes are known before job submission. At the same time, for certain classes of jobs, the duration of the job cannot be estimated before execution. For

example, computing the duration of software testing jobs is not predictable. Such jobs cannot directly use this technique, and thus call for dynamic scheduling and load balancing.

Asef Al-Khateeb et al.[2] proposed Meta Scheduling in grid environment. Meta-scheduling systems play a crucial role in scheduling jobs that are submitted for processing and require special attention because large numbers of jobs are being executed on small number of resources. The primary problem of meta-scheduling is selecting the best resources to execute the tasks while achieving the following objectives: reducing the mean job turnaround time, load balancing for better resource utilization, and considering job priorities. They proposed an enhanced meta-scheduling system, called Job Nature Meta-scheduling (JNMgrid). JNMgrid consists of three components: (1) Job Analyzer, its function is to determine the types of jobs in specific ratios; (2) Job Matcher, it is used for matching the jobs with the appropriate resources and (3) Job Batch Allocator, its function is to determine the best number of jobs for execution. The performance of JNMgrid was compared with similar existing systems i.e. Queue Length, File Access Cost, and File Access Cost plus Job Queue Access Cost. The simulation results demonstrated that JNMgrid outperforms these systems and can thus be deployed in any grid middleware to improve sharing of limited exclusive resources among grid users. JNMgrid is responsible for categorizing the received jobs as Computational Jobs (CJ) and Data intensive jobs(DIJs). The CJ are those that require more execution time than data access time, which represents the time that is required for input and output (I/O) operations. DIJs are the jobs that demand more access time than execution time. Also, JNMgrid outperforms other similar meta-scheduling systems and produce better results for all three underlying objectives simultaneously. Therefore, JNMgrid could be deployed in real grid middleware, such as Globus. Moreover, JNMgrid includes a feature that was not present in existing systems: it considers users' priorities in the scheduling decisions to support SLAs in real grid systems. Hence, the proposed system can provide better resource sharing and resource optimization in a distributed grid environment. There is no doubt, this proposed meta-scheduler achieves the objectives like reducing job turnaround time, ensuring site load balance and considering user's priorities. But this meta-scheduler is not able to address other objectives like less execution time and more usage count which are also equally important objectives.

Salman Meraji et al., [3] have proposed a new algorithm which is called best-min algorithm in order to conquer the disadvantage of min-min algorithm that is schedule produced by min-min is not efficient in term of load balancing and max-min's relative time to finish assigning tasks is very high. It is a two stage algorithm. The best-min algorithm uses min-min heuristic to use makespan in first stage and reschedule the tasks in the second stage to reduce makespan time. Algorithm use best -min considered all the resources in grid environment and this is lead to maximize the utilization of resource in grid.

Sadegh Nejatzadeh et al.[4] presents a new scheduling algorithm for static mapping to achieve better performance. Task Scheduling is a vital design issue of distributed computing. A computational grid is a highly distributed environment. The goal of grid scheduling is to achieve high throughput and to allocate appropriate computing resources to applications. The Complexity of job scheduling[6] rapidly increases with the size of the grid and becomes challenge to solve it. Different algorithms have been proposed so far and some of them are based on heuristic techniques to provide an optimal or near optimal solution for large grids. The proposed heuristic approach execution time uses a simple mapping function which tries to show the machine state together. A new algorithm is proposed to schedule task using meta-scheduler in grid computing system. The proposes heuristics tries to consider the execution time and resource state simultaneously by a mapping function. However, the results of proposed method is not better than Min-Min method in terms of makespan, but its results are comparable with its running time that is three times faster than Min-Min.

Javad Akbari Torkestani[5] proposed a fully distributed, learning automata–based job scheduling algorithm is proposed for grid environments. Job scheduling is one of the major task in the design of grid environments. The performance of the grid system degrades if there is not any method to efficiently schedule the user jobs. The proposed method is composed of two types of functions: in the first, a function is run at the grid nodes and second function is run at the schedulers. The proposed algorithms synchronize the performance of the schedulers by the learning automata. The proposed algorithm selects their actions using the pseudo-random number generators with same data set. In this method, the grid computational capacity is allocated to each scheduler that is relative to its workload. Several simulation experiments were executed under different grid scenarios to show the superiority of the proposed method. The obtained results show that the proposed algorithm performance is good in terms of makespan, flow time, and load balancing.

3. DECISION BASED META SCHEDULER FRAMEWORK

We have implemented a decision based Meta scheduling framework based on the reliability. It accepts the user request and finds out the trust full nodes from the available resources in different scheduler. From the list of reliable node it selects the best node to execute the task.

Components:

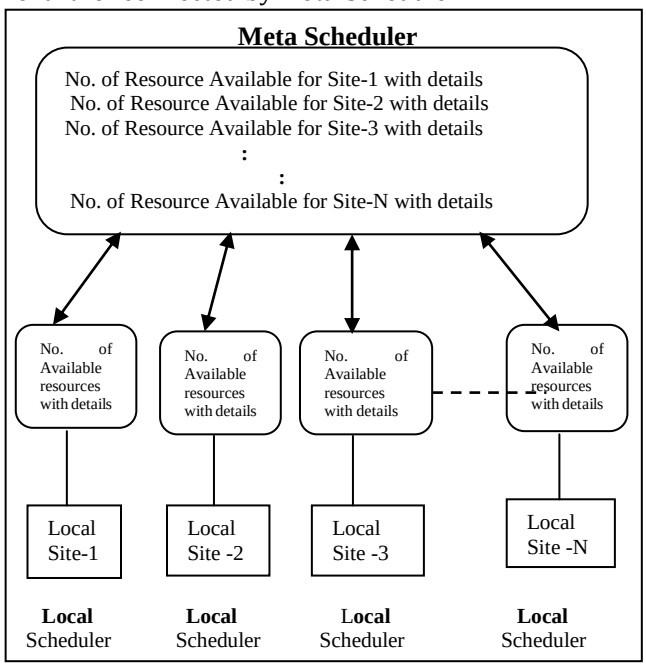
The proposed framework has three major components:-

1. Local Site
2. Local scheduler
3. Meta Scheduler

Local Site:- as we know grid is a distributed system having resources widespread distributed and belongs to different administrative domain. Nodes present at particular domain are known as local site.

3.1 System Architecture

The architecture of Meta Scheduling framework is shown in Fig. 1. The Architecture consists of sites at different locations which are connected by local scheduler. The local schedulers are further connected by Meta-Scheduler.



3.1.1 Local Scheduler:

Local level scheduler is collection of Computing Elements (CE) and Storage Elements (SE). This collection is visualized as cluster of resources [1]. Local scheduler only manages a single site or cluster and usually owns the resource but it has complete control of all the resources on that cluster.

Local scheduler in proposed approach rank the resources based upon self-protection capability[13]. Self-protection capability of an resource is calculated by given weightage to important security factors. The values of these factors differ in the range between 0 and 1. The security factors and their proportions are listed in the following table:

Table 1: Security Factors and their value

Security Factors	Proportion(P)
Anti-virus Capabilities (AV)	0.85
Firewall Capabilities (FC)	0.9
Authentication Mechanism(AM)	0.8
Secured File Storage Capabilities(SFC)	0.7

Based on these security factors SPC is calculated as:

$$SPC = \frac{\sum_{i=1}^{n} P(i)}{n}$$

Where,

n = the total number of factors.

P(i) = the proportion.

Based on all these factors we can calculate the rank of the resource as follows:

$$Rank(R) = \frac{AV+FC+AM+SFC}{4}$$

Local scheduler give ranking to the resources based upon rank value [12], so that most reliable node come upward in the list.

The Local level scheduler decomposes application received from Global level into set of jobs. These jobs are input to the Local level algorithm. Different algorithms executed on local and global scheduler level like Round Robin, First come first serve scheduling algorithms.

3.1.2 Meta Scheduler:

Both local and Meta schedulers aim at resource allocation and management but Meta-schedulers are different from local scheduler. Because Local schedulers are used to perform the scheduling task at local site level, whereas meta-schedulers communicate to all the local scheduler to update the status of available nodes. There are three major goals for a meta-scheduler, first is to maximize the resource utilization, second is load balancing and third is to allocated user applications fairly to the resources for scheduling.

Meta-scheduler keeps the information about all resources. When the job is come for execution; meta-scheduler will arrange the jobs in a queue for scheduling in the local scheduler. Meta-scheduler will also collect the current data availability from both the users and resource providers' at periodically interval. The meta-scheduler distribute jobs to multiple sites, instead of sending the jobs to most lightly loaded site to achieve the resource utilization and load balancing. A collective effort of a local and meta-scheduler a better scheduling decision can be taken.

4. WORKING STEP of META SCHEDULER

For each local site there is a local scheduler which maintain the information of all the resources i.e. no. of resources are occupied by some tasks and no. of resources are available for execution.

Step 1:- Local schedulers are connected with Meta scheduler and rank the resources based upon self-protection capability of the resources i.e. Anti-virus, Firewall, Secured File Storage Capabilities, Authentication Mechanism.

Step 2:- The meta-scheduler on regular periodic intervals collects the information from the user and resource provider from local scheduler.

Step 3:- When the users submit his job request then Meta scheduler match jobs to the available resources.

Step4:- if the Resources required by
application < available resources then
Job is submitted to resources without any ranking mechanism.
Else

Rank the resources based upon reliability and status updation values and Submit the resources to top most resources.
Step 5: Apply Min Min algorithm
Step 6:- Meta-scheduler also maintain the Job queue for pending jobs and in the next scheduling cycles those pending jobs will be executed.

5. EXPERIMENTAL SETUP

We have setup a simulated Grid environment to evaluate the proposed Decision based Meta Scheduler scheduling algorithm. We used GridSim simulator for simulating Grid environment and the experimental results shown in Fig. 2. We used Pentium—i3 based system with CPU clock speed of 1.7 GHz, 4GB RAM running with Windows 7 operating system. The simulation is based on the grid simulation tool kit [9] Gridsim Toolkit 4.0 which allows modeling and simulation entries in grid system. The heterogeneous environment is build by various resource specifications. The resource in the grid environment differs in Operating system, CPU speed, memory and bandwidth. The performance of the algorithm with large no of jobs is analyzed and also compared with FCFS[7], round robin[8] approach. The Decision based Meta Scheduler scheduling algorithm and the conventional FCFS, round robin are compared by their makespans on the same set of tasks.

Table-2: System attributes and its values

Parametrs	Value
No. of Jobs	10-100
No. of Processing Elements	2-15
Resource requirements (MIPS)	1000-6000
Speed of resource nodes (MIPS)	1000-6000

The main aim for submitting a job is to minimize the makespan time. In our simulation test we have taken 10-100 jobs and considered 2-15 processing elements. In our simulation setup we have various nodes with different jobs for selecting the resources FCFS, Round Robin and our Decision based Meta Scheduler
In the first step we have consider totally 10 nodes are available in the grid, for executing all the 100 job it takes more time as shown in fig. 2, likewise if no. of nodes increases, makespan time of the job is decreased. The execution time for our decision based algorithm is minimal if it's compared with the other two methods.

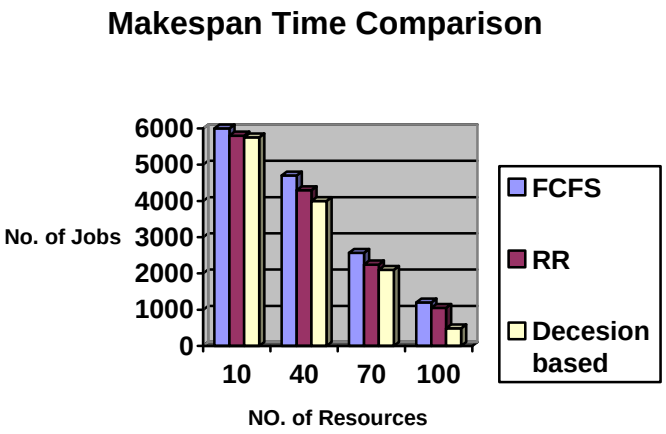


Fig2: Makespan Time Comparison

As clearly shown in Fig2 the decision based meta scheduler outperforms the conventional FCFS, Round Robin heuristic. The makespan of decision based meta scheduler is approximately 22% shorter than that of the conventional FCFS Scheduling on the same set of tasks and 13% shorter then Round Robin Scheduling.

6. CONCLUSIONS

Resource Scheduling is one of the well-known problems in distributed computing systems such as Grid environments. In this paper we propose an approach for resource scheduling based upon meta scheduling architecture. In proposed method meta-scheduler on regular periodic intervals collects the information from the user and resource provider from local scheduler. In our proposed approach we effectively utilized all the resource because the jobs are submitted to appropriate resources. The experimental result shows that proposed algorithm outperforms the traditional FCFS, Round Robin heuristic on the same set of task.

7. REFERENCES

[1] Thomas A. Henzinger, Vasu Singh, Thomas Wies, Damien Zufferey: *Scheduling large jobs by abstraction refinement*. EuroSys 2011: 329-342
[2] Asef Al-Khateeb ,Nur'Aini Abdul Rashid, Rosni Abdullah, "An Enhanced Meta-scheduling system for grid computing that considers the job type and priority", Computing (Springer Journal), Vol. 94, issue: 5 ,pp 389-410,2012.
[3] Salman Meraji, M. Reza Salehnamadi ,A Batch Mode Scheduling Algorithm for Grid Computing ,"Journal of Basic and Applied Scientific Research ,Volume:3 , Issue: 4, pp. 173-181, 2013.
[4] Sadegh Nejatzadeh et al., A New Heuristic Approach for Scheduling Independent Tasks on Grid Computing Systems", International Journal of Grid and Distributed Computing, Vol: 6, Issue:4 , 2013
[5] Javad Akbari Torkestani, "A New Distributed Job Scheduling Algorithm For Grid Systems", Cybernetics and

Systems: An International Journal, Volume 44 Issue 1,pp 77-93 2013.

[6] D. Thilagavathi & Dr. Antony Selvadoss Thanamani "A Survey on Dynamic Job Scheduling in Grid Environment Based on Heuristic Algorithms " published in*International Journal of Computer Trends and Technology- volume3Issue4-2012*

[7] U. Schwiegelshohn, R. Yahyapour, Analysis of First-Come-First-Serve parallel job scheduling, in: Proceedings of the 9th SIAM Symposium on Discrete Algorithms, 1998, pp. 629_638.

[8] Izakian Hesam, Abraham Ajith and Snasel Vaclav,Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments, *The 2009 IEEE International Workshop on HPC and Grid Applications(IWHGA2009), China, IEEE Press, USA*, ISBN 978-0-7695-3605-7, pp. 8-12, 2009a.

[9] I. Foster, C. Kesselman, The Grid_Blueprint for a New Computing Infrastructure, Morgan Kaufmann Pub., 1998.

[10] Xuan Wang And Ling-Fu Kong "Study On Grid Resource Selection Method Based On Multi-Goals" Proceedings Of The Sixth International Conference On Machine Learning And Cybernetics, Hong Kong, 19-22 August 2007.

[11] YU Huashan And XU Zhuoqun "Standardizing Resource Selection and Access On Computational Grids" Proceedings Of The Fifth International Conference On Grid and Cooperative Computing (GCC'06).

[12] Lilian Noronha Nassif, José Marcos Nogueira And Flávio Vinícius De Andrade "Distributed Resource Selection In Grid Using Decision Theory", Seventh IEEE International Symposium On Cluster Computing And The Grid(Ccgrid'07).

[13] Dr. Rajesh K. Bawa and Gaurav Sharma, "Reliability and Performance Based Resource Selection in Grid Environment", International Conference on High Performance Architecture and Grid Computing, Published in Springer Proceeding Series: Communications in Computer and Information Science, Volume 169, 19-20, July 2011, pp. 449-454.