Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HYBRID ADAPTIVE JOB AND RESOURCE SCORING META- SCHEDULING SYSTEM FOR GRID COMPUTING

[1]N.KRISHNAMOORTHY, [2]R.ASOKAN
[1] Assistant Professor (Sr.Grade), Department of Computer Science,
Kongu Engineering College, Perundurai, Erode, Tamilnadu-638052- India
[2]Principal, Kongu Nadu College of Engineering and Technology, Tholurpatti,
Thottiam, Trichy, Tamilnadu-621215-India
E-mail: [1]krishnamoorthyer@gmail.com, [2]asokece@yahoo.com

ABSTRACT

Grid is a heterogeneous system that allows sharing of resources. Grid computing is a technology that works what super computer does. Scheduling independent jobs to the resources is not an easy task. Meta-scheduling schedules maximum number of jobs to the minimum amount of resources which is a very tedious task. Many scheduling algorithms exist to focus either on the job side or on the resource side. Current Grid meta-schedulers are either based on system-centric metrics or utility metrics provided by the users. In the proposed work, to utilize the power of grid completely, both the job and resources are taken into account. Jobs are prioritized based on the common location sum that considers both user and system priority. Resources are prioritized based on the computing power which is obtained as the normalized value of the CPU_available. So the high priority job can be submitted to the corresponding resources based on the computing power and it helps to improve the performance and reduce the completion time of the job.

Keywords: *Grid Computing, Meta-Scheduling, Grid Entity, Grid Role, Grid Management Environments*

1. INTRODUCTION

Grid computing is a technology that couples disparate and distributed heterogeneous software and hardware resources to provide uniform computing environment to solve data and computational-intensive problems. It enables the sharing of resources to solve a problem that work towards our common goal. The resources of many computers in a network can be applied to solve a single problem at the same time. The process that needs a large number of computer processing cycles can use this technology to solve their problem.

Grid computing can be a form of heterogeneous, distributed and large-scale cluster computing. It needs to schedule the process to the appropriate computation resources so that it can divide and form the pieces of a program to as many thousand computers as possible in the virtual organization. The different type of computer resources can be applied to a different type of problem.

Grid computing can experience the difference in resource utility because, they can be owned by an independent organization but they are utilized by another set of organization. It provides the good infrastructure for the user communities to access and to solve their specific problems with the theme of obtaining the objective.

Grid computing works towards the three common objectives: (1) more cost-effective with the given amount of computer resources, (2) as a way to solve problems with less amount of computing power, and (3) whole resources are working towards our common objective. Not only with these objective it can also focus towards the standards in certain areas like scheduling, resource discovery, load balancing, security, system management and so on. Because of the heterogeneity nature of the grid, the global grid forum has been structured as a working body for designing standards for the grid.

1.1. SCHEDULING PARADIGMS

1.1.1 Centralized Scheduling

In centralized scheduling, a central machine acting as a resource manager which schedule jobs to all the surrounding nodes that are a part of the grid environment. It is used in situations like a computing center where resources that have the similar characteristics and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

usage policies.

Jobs are first submitted to the central scheduler who then dispatches the jobs to the appropriate nodes. Jobs which are not assigned to a node are normally stored in a central job queue.

1.1.2 Distributed Scheduling

In distributed computing, multiple localized schedulers interact with each other to assign jobs to the participating nodes. The scheduler communicates with other schedulers either by direct or indirect communication. It overcomes the scalability problems which are incurred in the centralized paradigm and also offers better fault tolerance and reliability. However the lack of a global scheduler which has the necessary information on available resources, usually leads to suboptimal scheduling decisions.

1.1.3 Hierarchical Scheduling

In hierarchical scheduling, job submissions are managed by centralized scheduler interacting with local schedulers. Similar to the centralized scheduling, hierarchical scheduling can have scalability and communication bottlenecks. However, advantage of hierarchical scheduling compared with the centralized scheduling is that the global and local schedulers can have different strategies in scheduling jobs.

1.2 SCHEDULING STAGES

Scheduling can work with four main stages [2]. They are resource discovery, resource selection, Schedule generation, job execution.

1.2.1 Resource Discovery

Resource discovery aims to identify a group of authenticated resources that are offered for job submission. A scheduler needs to have some way of incorporating dynamic state information about the available resources into decision making process to handle the dynamic nature of the grid. Grid environment mostly uses the pull, push or pull-push model for resource discovery.

1.2.2 Resource Selection

Once the targeted resource is known, the second phase is to select the resources that best suit the user constraints and conditions imposed by the user, such as CPU usage, disk storage or RAM availability. The resource selection process identifies a resource list in which all resources can meet the minimum requirements for a submitted job.

1.2.3 Schedule Generation

The generation of steps involves two steps, selecting jobs and producing resource selection strategies. The aim of job selection is selecting of a job for execution from a job queue First come first served, Random selection, Priority-based selection, Backfilling selections are the strategies that are used to select a job.

1.2.4 Job Execution

Once a job and resource are selected, the job is submitted to the resource for execution. Job execution can either be run as a single command or as a series of scripts.

1.3 META SCHEDULER

Grid computing supports execution under workload across various computing resources from various organizations, institution to form virtual organization. The core of the Grid system is the management entity commonly known as Meta schedulers or grid resource broker. Grid entity is an entity that can be defined by its roles and is to be shared. An entity can be computer, data, storage, software, etc and its entity role can be resource, account, job, policy, contract, user, etc. An entity can have one or more roles. A Grid entity can fulfill many roles. For example, it can have both a Job role and a Resource role; it can have both a Job role and a Workflow role, or indeed any other combination of roles.

A Grid entity should be managed within different management environments, depending on the various roles that it fulfills. A Grid entity role and Grid entity are defined by a set of role specific attributes and its roles. The definition of a Grid entity could be captured in a description document as sets of role attributes. The description document will then allow the Grid entity to be managed according to its various roles, within a Grid.

Different roles require different management environments. For example, resource management environment, job management environment, etc. A Grid management environment sets the rules for the management of a set of Grid entity role types. The types of Grid entity roles managed in a particular management environment cannot be managed by any other.

Different management environments should allow for custom management models, methods, protocols, languages, and semantics. However, due to the interdependencies between

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

these management environments, they must also share a common set of languages, semantics, and ontologies in an interoperability layer to allow them to interoperate.

Without this common set of interoperability features, it would not be possible to manage role and environment interdependencies. Scheduling should allow the management of roles within their respective management environments. Meta-scheduling should handle the interdependencies between roles and their environments, both within Grid entities, and between Grid entities.

The rest of the paper is organized as follows. Section 2 reviews work specifically related to priority of the job and resource selection based on the job. The proposed work is discussed in section 3. The Simulation results and analysis of the results are presented in section 4. Section 5 concludes and gives direction of the future research.

2 RELATED WORKS

Over the last few years, a lot of the scheduling algorithm was proposed in the past. The Grid which helps in the sharing of resources has gained a lot of attention from different researchers. Resources in the grid are distributed across multiple domains. The efficient way of scheduling the jobs to the resources can be performed in the number of ways. Some of them are listed below.

Srikumar Venugopal and Rajkumar Buyya [3] focused on scheduling the jobs to the resources by presenting a model and an algorithm. The model considers the times and costs for transferring the dataset to the required job from different data hosts to compute under resources and algorithm help in building the resource set for the job that minimize the time and cost depending on the user preferences. Without considering the possible combination of resources for complexity, they achieve the least cots and times from their evaluation test bed.

Dongsung et al [4] focused on the use of virtualization to the computing resources. Priority based scheduling scheme is used for the virtual machine monitors. The next task which is to be executed is selected based on the task priorities and the input-output stats. The algorithm was implemented on Xen virtual machine and the average response time to the input-output event is improved by 22% in the consolidated environment.

Ravin et al [5] focused on the Dynamic Priority Scheduler for Advanced Reservation. It is used to assign the resources based on job priorities and job length. The job allocation takes place only after the dynamic analysis of the job at run time.

Shortest Job First technique is used for sorting the user jobs and the scheduled jobs are reserved to the resources in a random manner. Starvation for low priority jobs has been reduced by limiting the number of jobs that each user has to be reserved. The number of rejection jobs which do not get any reservations has also been reduced.

Selvarani and Sudha sadhasivam [6] proposed an improved heuristic approach based on Particle Swarm Optimization (PSO) algorithm. It is used to solve the task scheduling problem in grid. Tasks are grouped and allocated to the resources in an un-uniform manner. The new method of grouping the tasks helps in achieving the minimum completion time of tasks, minimum cost and maximum utilization of resources by comparing to the uniform grouping method.

Abdullah et al [7] presented the concept of job type. Job can be classified as two types. They are Computational job and Data Intensive job. Computational jobs needs more processing power than data access cost and Data intensive jobs are those which needs more data access cost than processing power. Based on the two types of the jobs, job ratio is calculated to execute the jobs. Thus the turnaround time of the job is reduced and improved the overall performance.

Raksha et al [8] focused on achieving possible highest system throughput by matching the application to the needed computing resources. The main motivation of this survey is to understand the concept of grid computing and to develop a new efficient scheduling algorithm which will benefit the interested researcher to carry out their further work. This paper also concentrates on memory requirement of the job, when they are submitted to the computing resources.

Zafril et al [9] determined the best algorithm performance by comparing six initial scheduler algorithms. Since the initial scheduler algorithms can affect the total scheduling process, it is important to select the best scheduler

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm. Results have been used to determine the best initial scheduling algorithm based on five different inter arrival time that represents different level of contention in the grid system. So that researcher can use the bet initial schedule for their proposed algorithm.

Abdullah et al [10] proposed an enhanced meta- scheduling system that considers job type and priority. The job type can be classified as computational job and data intensive job. Job priority considers both the user and system priority. The resource selection based on their processing power, job queue access cost, data access cost. The submission of the jobs to the appropriate resources can helps in the reduction of makespan of the individual resources and hence the overall improvement of the minimum turnaround time had been achieved.

Bhaskaran and Parthasarathy [11] reduced the execution time of the jobs by using the Priority queuing model. Based on the priority of jobs, they are dispatched to the computing resources in the adhoc grid layer. As the resources in the ad hoc network can compute the jobs faster with less execution time and compared with the existing system job's execution time, the proposed work job's execution time is reduced three times approximately.

Chang et al [12] dealt with the scoring of the computing resources among the cluster. In the proposed system, an adaptive scoring method is used to schedule jobs in grid environment are designed. ASJS selects the fittest resource to execute a job according to the status of resources. High computing power cluster can selected among the different clusters and the appropriate resources in the selected cluster can be identified to submit the job using the average transmission power. Local and global update rules are applied to get the newest status of each resource. Local update rule updates the status of the resource and cluster which are selected to execute the job after assigning the job and the Job Scheduler uses the newest information to assign the next job. Global update rule updates the status of each resource and cluster after a job is completed by a resource.

The major one of the focus to the grid computing is scheduling. The way of scheduling the job to the resources plays a main role in their overall performance. Selection of appropriate resources to the particular job helps in improving their own makespan. Thus focusing on both the job and the resources side helps to improve their performance.

3 PROPOSED WORK

Grid computing performs resource sharing and problem solving across distributed environments. Scheduling plays a critical role in grid computing. The major concentration over scheduling is to improve the overall performance. Issues of scheduling in grid computing can be mainly tackled by assigning the job to the resources based on their computing power. Meta-scheduling plays a main role in scheduling the maximum number of jobs
with minimum available resources. It helps to manage the scheduling within the grid entities or between the grid entities based on their grid roles. Scheduling itself is not an easy task, rather than that scheduling of maximum number of jobs with minimum resources becomes very tedious. It is best to consider the priorities on both the job and resource side.

For example, consider the 5 jobs and the 3 resources, from that common location sum are calculated for job priority and the computing power is calculated for resources.

In earlier, the job id and their related ranked sites are based on the computational job, data intensive job, processing power, data access cost and job queue access cost.

For the job side, both the user and system priority is taken into account. By using those priorities, the common location sum is calculated for the jobs. Then submitted to the less computing power resources. Those jobs have been executed first and each and every time computing power of the resources have been calculated high priority jobs can be assigned to the first ranked site to achieve the minimum turnaround time for the completion of the job with the available resources.

As an example in table 1, in user priority, Job J1 is in second location and in system priority; it occupies second and third position. Hence their location sum is seven, which ranked first in the job prioritized list. Based on the example, job J1 is executed in the first site, job J2 is executed in the second site, and then job J5 is executed in third site and so on.

For the achievement of the minimum completion time, consider the two types of the job and their corresponding cost needed. The achievement of the minimum turnaround time for the completion of the 100 randomly selected jobs with the 25 resources had been taken.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TABLE 1. JOB AND THEIR LOCATION SUM

Job ID	User priority	System priority SJF		Job location sum	Prioritized job list
		SJF	FIFO		
J1	J5	J2	J3	2+2+3=7	J1
J2	J1	J1	J2	5+1+2=8	J2
J3	J4	J5	J1	4+5+1=10	J5
J4	J3	J4	J4	3+4+4=11	J3
J5	J2	J3	J5	1+3+5=9	J4

In table 2, for the resource side, the CPU available is calculated by using the formula

CPU _ available=CPU speed * (1-load) (1)

TABLE 2. COMPUTING POWER FOR RESOURCES

Needed values	Resources	
	R1	R2
CPU_ speed(MHZ)	3100	3400
Load(%)	20	25
CPU_ available(MHZ)	2480	2550
CP	9.73	10

Then the CP value can be normalized from 1 to 10. Assign 10 to the highest available CPU and then highest value divided by the next highest value of CPU_available, multiplied with 10 gives the next CP value.

The highest priority job can be submitted to the resources which contain more computing power, if it has less data access cost. Otherwise the job has been submitted to the less computing power resources. Those jobs have been executed first and each and every time computing power of the resources have been calculated and updated globally for each and every job. Thus it minimizes the execution time of the job as it has high computing power and hence the performance can be increased by improving the makespan.

Figure .1 illustrates the overall architecture of the proposed work. The portal acts as an interface between the user and the grid resources. It also helps to monitor the status of the grid resources. The job scheduler maintains the information about the job and prioritized job list has also been maintained here
to submit the job to the appropriate resources. The regular update of the job has been noted here. Information server maintains the overall grid resource information. Appropriate can be submitted to the resource based on the information in the information server.

3.1 Grid Portal

Grid portal are similar to web portal which helps in the uniform access of the grid resources in a distributed environment. It acts as an interface between users and the grid resource environment. The complexity of the task management can be reduced by providing the capabilities such as remote access of resources, authentication of the resources, scheduling capabilities, and monitoring the status
information. There are number of software development toolkits and standards to develop the grid portal and it plays a major significant role in emerging web service portal development.

3.2 Job Analyzer

Job analyzer helps to analyze the ratio of the job. It can be specified in ratio of two numbers. The job can be classified as computational or data intensive job based on the history data file. If the history data file is not present for particular type of job, then the jab ratio is calculated as an average ratio of some existing jobs under execution. It also monitors the job execution time and job input output time.

3.3 Job Decider

Job decider helps to obtain the prioritized list of the job based on the information obtained from the job analyzer component and moves the list to the next component, Job batcher.

3.4 Job Batcher

Job batcher received the prioritized job list from the job decider and related queue information from the grid sites. It helps to batch the ranked prioritized job based on available queue length and submitted to the site instead one by one.

4 SYSTEM EVALUATIONS

Performance evaluation Gridsim is a simulation tool used to evaluate the performance of the scheduling strategy. The study of our scheduling strategy and selection algorithms was carried out. The performance analyze of the proposed work is compared with the other existing

Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ystem. The simulation assigns the values for the CPU_speed and load at each grid site to test the underlying systems.

Performance metrics There are three metrics used to evaluate the overall performance of the scheduling strategy. Local resources can be used to schedule the appropriate job where their needed data are stored. But it degrades the performance by without considering the global resources, even though it contains high performance. So the response time can be improved but it reduces the overall turnaround time. Thus the maximum turnaround time is the major parameter used here. It can be defined as the total time required for completion of all the jobs.

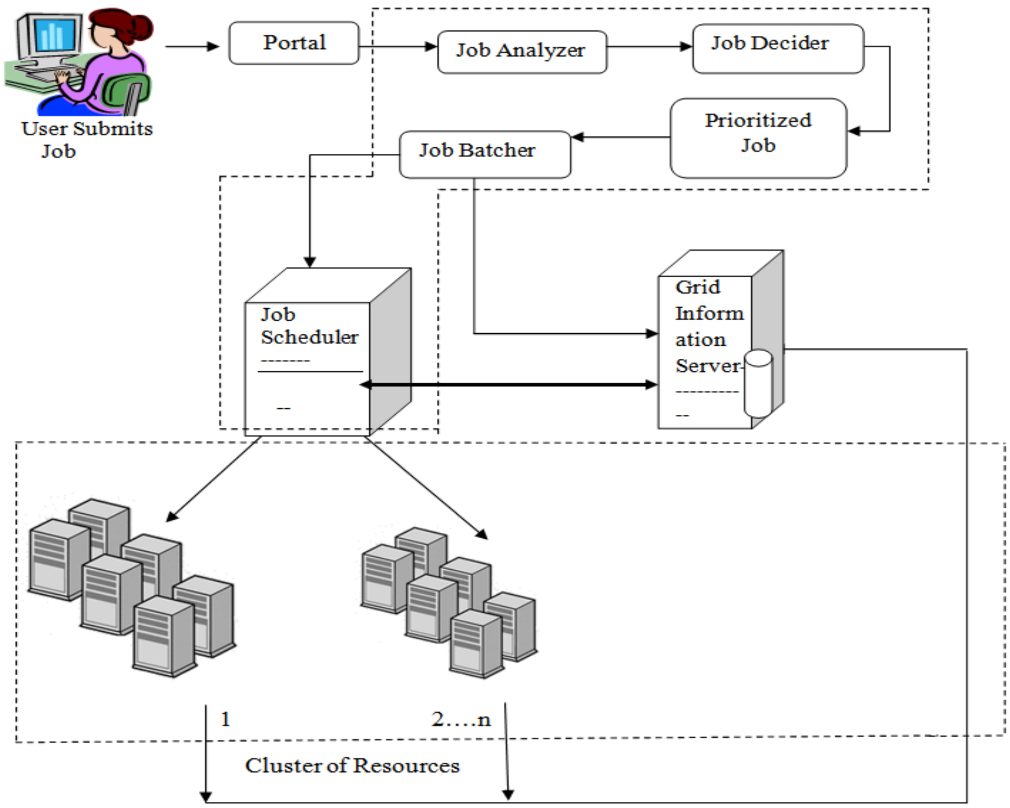


FIGURE 1 Overall Architecture Diagram

It can be calculated by

MJTT=

$$\frac{\sum_{i=1}^{n} T_{arrive(i)} - T_{departure(i)}}{n} \quad (2)$$

Where,

TArrive (i) =the time the job began execution.
TDeparture (i) =the time the job finished execution.
n =the total number of jobs processed through the system.

Another parameter is load balancing. When a site contains a more number of jobs to execute, it will become overloaded and hence the performance had been reduced. So the load should be balanced at each site, thus standard deviation is calculated to prove the effectiveness.

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(MS_i - MS_s)^2}{N}} \quad (3)$$

Where,
SD = Standard Deviation, MS_i = Mean per site
MS_s = Mean sites, N = Number of sites

At each site, the MJTT is calculated as MS. The average MS can be calculated as

MS=

Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\frac{\sum_{i=1}^{m} job_i turnaroundtime_{(site)}}{M} \quad (4)$$

Where,
MS = the mean per site, M = the number of jobs per site

The turnaround time for all the sites are calculated using MS_S and the MS_S can be calculated as

$$MS_S = \frac{\sum_{i=1}^{n} job_i turnaroundtime_{(site)}}{N} \quad (5)$$

Where,
MSs = the mean for all sites
N = the number of jobs processed at all sites

Since both the user and the system is taken, priority is also considered as one of the parameter and it can be calculated as follows

Priority=

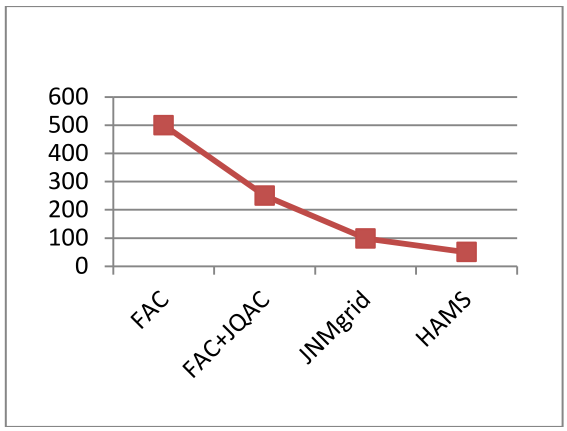
$$\sum_{i=1}^{n} ABS(RP - AP) \quad (6)$$

Each job that is submitted to the system has a priority (the user priority) called it as the Actual Priority (AP). The priority obtained from the scheduling system is called the Recorded Priority (RP). Thus the sum of the differences between the AP and the RP for each submitted job is computed.

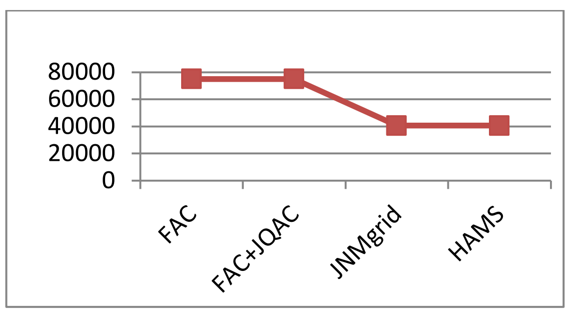
5 RESULTS AND DISCUSSION

Simulation is carried out using the well known simulator named Gridsim. In order to carry out the testing process, the datasets are generated. The simulation assigns values for the speed of the CPU and their corresponding load for the resources. The priority values for gridlets are also assigned. The scheduling strategies are carried out under the concept of Meta – scheduling, more number of jobs with minimum amount of resources are considered. 100 numbers of jobs are used to schedule among the 25 resources. The experimental results are drawn among the execution of the jobs with available resources. The experimental results are investigated on the effect of the turnaround time and standard deviation. The random generation of the jobs based on both the user and system priority can also lead to the factor priority.

First metric, Mean job turnaround time (MJTT) is defined as the total time required to complete all grid jobs. It has been measured in second and it also shows the slight improvement by 3%.Seconds in Y axis and different method in X axis



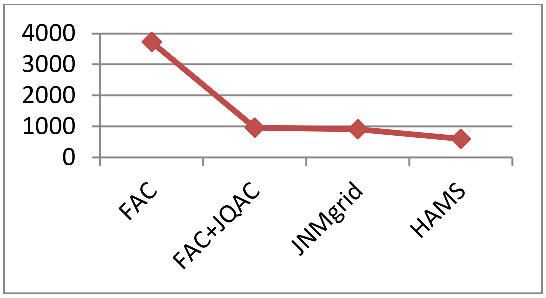
Second metric is the priority and it is calculated since both the user and system priority is used to generate the random job location sum. Based on the job common location sum, prioritized list of job is generated and it remains the same as the existing system. Priority in Y axis and different method in
X axis.



Third metric is the standard deviation and it has been measured for balancing the load and it also exhibits that the overall improvement has been occurred and the type of job submitted to the resources based on high and low computing power helps to improve the load balancing. The improvement has been shown in the below graph.

Journal of Theoretical and Applied Information Technology
31st August 2013. Vol. 54 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



The overall performance can be compared with the existing system for the turnaround time and standard deviation metrics. The efficiency can be calculated by using the formula

Efficiency=

$$\frac{Metricvalue(Existingsystem) - Metricvalue(HAMS)}{Metricvalue(Existingsystem)} \quad (7)$$

A pair wise comparison of the MJTT efficiencies with the existing systems are presented below

MJTT	FAC	FAC+JQAC	JNMGrid	HAMS
HAMS	83	67	49	-
JNMGrid	85	60	-	-
FAC+JQAC	61	-	-	-
FAC	-	-	-	-

A pair wise comparison of the standard deviation efficiencies with the existing systems are presented below

SD	FAC	FAC+JQAC	JNMGrid	HAMS
HAMS	86	48	45	-
JNMGrid	72	46	-	-
FAC+JQAC	47	-	-	-
FAC	-	-	-	-

6. CONCLUSION

Grid computing is one of the major key concerns to develop the computing system that have the major ability to self configuration and optimization. Numerous algorithms in the literature has been proposed and focused on this serious scheduling problem. However there are several drawbacks in the existing system, the proposed system can able to overcome those problems. The tedious process of assigning the site ranking can be major problem and time consuming. The dynamic nature of computing the CPU available and the computing power is used by the resources. Thus the job with the highest priority which is obtained from the common job location sum can use those computing power resources to achieve the minimum turnaround time and hence the performance has been improved.

For the future directions, starvation may occur due to the long time waiting of the jobs in queue. So some other aging concepts can be used to avoid the long time waiting of job in queue. Thus it helps to achieve the performance improvement .

REFRENCES:

[1] Hamscher, V., Schwiegelshohn, U., Streit, A. and Yahyapour, R., : Evaluation of Job-Scheduling strategies for Grid computing GRID 2000, 191-202, 17-20 December 2000 Bangalore, India. Lecture Notes in Computer Science ,Springer-Verlag

[2] Jennifer M. Schopf, : Mathematics and Computer Science Division. Argonne National Laboratory

[3] Srikumar Venugopal and Rajkumar Buyya, : An Economy-based Algorithm for Scheduling Data-Intensive Applications on Global Grids. (2005)Dongsung Kim, Hwanju Kim, Myeongjae Jeon, Euiseong Seo, and Joonwon Lee, : Guest-Aware Priority-Based Virtual Machine Scheduling for Highly Consolidated Server, (2008)

[4] Ravin Ahuja, Gabrani, G., and Asok De, : A Dynamic Priority Scheduler for Advance Reservation for Grid computing, International Journal of Soft Computing, 4(2), 60-67, (2009)

[5] Selvarani, S., Sudha Sadhasivam, G., : Improved Job Grouping Based PSO algorithm for task scheduling in grid computing,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

International Journal of Engineering Science and Technology, 2(9), 4687-4695, (2010)

[6] Raksha Sharma, Vishnu Kant Soni, Manoj Kumar Mishra, Prachet Bhuyan, : World Academy of Science, Engineering and Technology 40 , (2010)

[7] Zafril Rizal M Azmi, Kamalrulnizam Abu Bakar, Abdul Hanan Abdullah, Mohd Shahir Shamsir, Wan Nurul safawati Wan Manan, : Performance Comparison of Priority Rule Scheduling Algorithms Using Different Inter Arrival Time Jobs in Grid Environment, International Journal of Grid and Distributed Computing , Vol. 4, No. 3, (2011), September

[8] haskaran, R., Parthasarathy,V., : An Improved Performance analysis of Priority Scheduling algorithm in modified Ad hoc Grid layer, International Journal of Distributed and Parallel Systems (IJDPS) , Vol.3, No.1,(2012), January

[9] Asef AL-Khateeb, Rosni Abdullah , Nur'Aini Abdul Rashid, : Job Type Approach for Deciding Job Scheduling in Grid Computing Systems, Journal of Computer Science , 5 (10), 745-750, (2009)

[10] Ruay-Shiung Chang , Chih-Yuan Lin, Chun-Fu Lin, : An Adaptive Scoring Job Scheduling algorithm for grid computing, Information Sciences 207 , 79–89, (2012)

[11] Asef Al-Khateeb, Nur'Aini Abdul Rashid , Rosni Abdullah, : An enhanced meta-scheduling system for grid computing that considers the job type and priority, Springer-Verlag , (2011)

[12] N. Krishnamoorthy and R.Asokan. Certain Improvements in Optimization Techniques for Grid Scheduling.*Life Sci J* 2013;10(7s):568-572]. (ISSN: 1097-8135). http://www.lifesciencesite.com. 89.