A REVIEW ON RESOURCE SCHEDULING ALGORITHMS AND THEIR APPLICATIONS IN GRID COMPUTING

P. Pradeep Kumar[1], Dr. E. Chandra Blessie[2]

[1]Researcher, Department of Computer Applications, Nehru College of Management, Coimbatore, India.

[2]Professor, Department of Computer Applications, Nehru College of Management, Coimbatore, India.

Abstract

Grid computing is the technology of dividing computer networks with different and heterogeneous resources based on distributed computing which is used to solve problems in the grid system. The main aim of grid computing is to apply available computing resources for impenetrable calculations via sites that are distributed geographically without complexity. Resource scheduling and Resource management play a key role in achieving high utilization of resources in grid computing environments. The allocation of distributed computational resources to user applications is one of the most complicated and difficult tasks in the Grid system. The problem of allocating resources in Grid scheduling requires the definition of a model that allows local and external schedulers to communicate to achieve efficient management of the resources themselves. To rectify the disadvantages, this paper presents a review of some of the most widely known and recently proposed mechanisms in resource scheduling and resource management algorithms explained based on their advantages.

Keywords: *Grid Computing, Resource Scheduling, Resource allocation.*

I. INTRODUCTION

Resources of many computers in a network are used at the same time to solve a single problem with the help of Grid computing. Grid systems are designed for collaborative sharing of resources and it can also be thought of as distributed and large-scale cluster computing. Grid computing is making big contributions to scientific research, helping scientists around the world to analyze and store massive amounts of data by sharing computing resources. Grids tend to be more loosely coupled, heterogeneous, and geographically dispersed compared to conventional cluster computing systems. The goal of Grid Computing is to create the illusion of a simple yet large and powerful self-managing virtual computer out of a large collection of connected heterogeneous systems sharing various combinations of resources. In a basic grid computing system, every computer can access the resources of every other computer belonging to the network. At its most basic level, grid computing is a computer network in which each computer's resources are shared with every other computer in the system. Processing power, memory, and data storage are all community resources that allows authorized users to tap into and leverage for specific tasks.

A grid computing system can be as simple as a collection of similar computers running on the same operating system or as complex as inter-networked systems comprised of every computer platform in the form of distributed computing. In distributed computing, different computers within the same network share one or more resources. In the ideal grid computing system, every resource is shared, turning a computer network into a powerful supercomputer. With the right user interface, accessing a grid computing system would look no different than accessing a local machine's resources.

This paper is organized into four sections: Section 2 describes the applications of grid computing. Section 3 presents the related works of grid computing models especially in the area of resource scheduling. Finally, the conclusion and future work are discussed in section 4.

II. GRID SYSTEM ADVANTAGES

Following are some of the advantages of the grid system based on resource allocation.

Exploiting underutilized resources:

Grid computing collects the computational power from unutilized resources and used them for computational purpose. The resources that are in idle state, the computational power of that resource are used in Grid Computing. Parallel CPU capacity: In the Grid system, Jobs are executed parallel, so it dramatically reduces the total execution time of jobs. Jobs are first distributed into various available resources and then executed parallel.

Virtual resources and virtual organizations for Collaboration:

In the Grid computing environment, Virtual organization indicated the user of the grid. In Grid, resources can be shared virtually to the virtual organization. Access to additional resources: Whenever there is a need for additional resources for computing, the grid provides the additional one. It seems us Grid, as a large collection of resources. Resource balancing: Each resource should register in the grid for processing. The grid has information about the availability of resources. As per the capacity and availability of the resource, Jobs can be distributed to various resources.

Reliability:

Grid is highly reliable in terms of failure. Jobs in the grid can automatically resubmit to other resources if there is any kind of failure occurs at one location.

Management:

As the Grid computing environment is heterogeneous so better management is essential. The grid provides better management not only for jobs but also for resources used inside the environment.

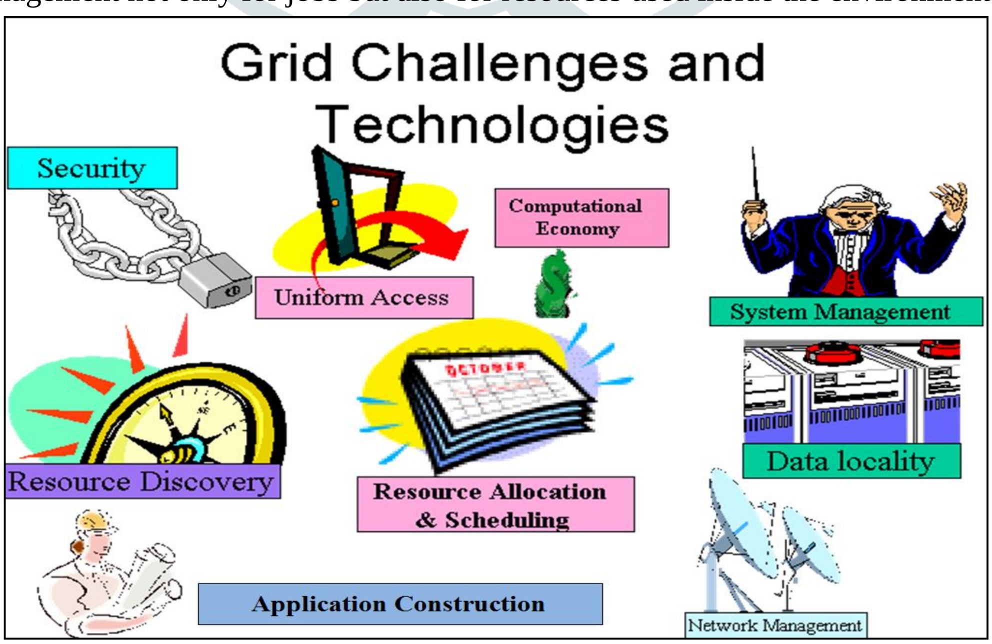

Figure 1: Challenges and Technologies of Grid Computing

III. LITERATURE REVIEW

Vahe Aghazarian *et al.* introduced an optimized real-time scheduling algorithm for tasks allocation in a grid environment by using modified RQSG algorithm. It is an optimized algorithm that is being improved using processing node and processing power parameters and this algorithm further allocates group jobs with prioritization rather than RQSG algorithm and it has dependent nodes in which the tasks are being done with less complexity as a set of output edges for each node. They have decreased the source loss time in their proposed algorithm using a new technique and increased the efficiency by choosing various weighting positions of nodes in comparison with similar algorithms. With sources prioritized allocation, they have evaluated the requested times and the answers and thus optimized final processing time for task allocation. In comparison with standard RQSG algorithm, the weight and communication cost parameters have been increased independent tasks in modified RQSG algorithm and this has caused decreased makespan in qualification functions. By using prioritization queues and weight setting in priority functions, they have optimized the loss time in multi-face graphs with higher time complexity.

Generally, they made strategic changes in fitness function using various simulations comparing common scheduling algorithms and optimized algorithm. Finally, a method by combining fitness function and scheduling queues for more optimization than RQSG algorithm was introduced. Comparing SGA and NG algorithms, the developed algorithm showed better performance regarding the going and returning tasks time and certifications according to results from various simulations. At last, the tasks prioritizing with a technical method introduced and that shows the performance improved by 1 % comparing standard RQSG in the worst case and 3% in the best case. Therefore, the loss of time has been an acceptable reduction comparing similar previous algorithms.

Miguel Pinto and Orlando Belo integrated a performance prediction method in a data cube distribution strategy to define a more effective allocation of data cube views in a conventional Grid environment with the ability to receive them. Their proposed method allows for the distribution of data cube views on a Grid environment, capitalizing on the advantages of a performance prediction feature of a Grid, the usage user profile of a data cube, and the knowledge of the computational nodes used by each decision-maker with the help of allocation method. The allocation model aims to bring data cube views close to users, trying to improve the performance of an OLAP platform and avoid the fragmentation of the data cube views. Unfortunately, G-BOV is the worst model in terms of execution time, because it makes too many calculations to avoid unnecessary materialization and searching for the best node to store a specific data cube view. But, analyzing the space needed to store the data cube or to know what is the faster model answering to a user query, the G-BOV model reveals an allocation strategy with better levels of performance needing less space to store the data cube. The experimental results of the simulation show the superiority of G-BOV in the use of the storage space to materialize the data cube and in the allocation of the data cube views in better nodes. Although, G-BOV's run-time execution results show that it is slower when compared with RR and Random models.

Syed Nasir Mehmood Shah *et al.* proposed the Hybrid Resource Allocation method, based on the Least Cost Method (LCM) and Divisible Load Theory (DLT) method. The Hybrid Resource Allocation method is an improved form of the Divisible Load Theory method. We have developed a modified assignment strategy of DLT by integrating it with Least Cost Method. At first, formulated an LP model for Grid resource allocation and also proposed the Hybrid Resource allocation method for Grid resource allocation and performed a comparative analysis of it with DLT and other well-known resource allocation methods. The Hybrid Resource Allocation method favors the equal distribution of jobs and finally, the Hybrid Resource Allocation method produces encouraging results as compared to other well-known resource allocation strategies. The Hybrid Resource Allocation method is a potential candidate for resource allocation in a computational Grid environment.

Amudha and Dhivyaprabha proposed a new framework and QoS(Quality of Service) Priority-Based Scheduling Algorithm for effective task scheduling to the resources in the grid environment. Grid computing comprises of distributed computer systems, which are geographically dispersed to share the combination of resources in a heterogeneous environment. Grid computing is one of the latest technologies to solve very complex problems in scientific applications, using the resource-sharing concept in dynamic virtual organizations. The main problem occurs in most of the organization is that the resources have become idle and underutilized. In some organizations, even the server machines can often be relatively idle. Grid computing provides an approach for exploiting these underutilized resources and thus has the possibility of substantially increasing the efficiency of resource usage.

Effective and Efficient Task scheduling is an important aspect of Grid computing. The objective of their work is that a grid scheduler first allocates the high priority jobs to the resources and then it allocates the low prioritize job to achieve the maximum resource utilization rate, minimize the makespan and avoid the load balancing level problem. It assigns tasks to all the resources equally so that the load-balancing problem is solved in their framework and algorithm. Therefore, the proposed work gives the optimum solution for task scheduling in a grid environment. The performances of the grid environment are measured from various factors such as makespan, resource utilization rate and load balancing level, etc. The developed algorithm evaluates and measures the performance of the makespan and resource utilization rate of a grid system. The results show that developed QoS priority-based scheduling algorithm gives better results in makespan and resource utilization rate than other existing algorithms such as QoS guided weighted mean time min, Min-Min, and Max-Min heuristic algorithms.

Murugesan and Chellappan introduced a new resource allocation model with multiple loads of originating processors as an economic model and their objective is to minimize the total cost of the grid user those who are assigning the job to the grid system. They assumed that the grid system consists of five processors i.e. resources namely P1, P2, P3, P4, and P5 with three sources S1, S2, and S3 are trying to utilize the grid system to execute their workloads. After processing it is clear that the total workload of S1 is divided into three parts, the total workload of S2 is divided into three parts and the total workload of S3 is divided into four parts and allotted into processors. Also, it shows that the completion time of each source's workloads. It is found that the resource allocation model can efficiently and effectively allocate workloads to proper resources. Experimental results showed that their proposed model obtained a better solution in terms of cost and time and further they consider the problem of scheduling large-volume loads within a cluster system, which is part of grid infrastructure.

IV. COMPARITIVE ANALYSIS

In this section, two different approaches to task scheduling in the grid are going to analyze based on the algorithms and their properties. Vahe Aghazarian *et al.* introduced an optimized real-time scheduling algorithm for tasks allocation in a grid environment by using modified RQSG algorithm. Amudha and Dhivyaprabha proposed a priority-based scheduling algorithm for effective task scheduling to the resources in the grid environment. Both used makespan as an attribute to calculate their performance of their algorithms in the grid system.

Amudha and Dhivyaprabha QoS Priority Based Scheduling [QSPBS] is designed and implemented for batch mode independent tasks scheduling and here, the Priority Value of a job is taken as QoS parameter. Once all the jobs are generated and put into a set, the task partition divides the tasks into two groups such as high priority and low priority tasks based on their priority values. Grid Scheduler classified the tasks into four categories such as

- High complexity and high priority job (1A),
- Low complexity and high priority job (1B),
- High complexity and low priority job (2A),
- Low complexity and low priority job (2B).

The high priority tasks are allocated first to resources for scheduling. Grid computer systems are partitioned into two groups named as Cluster 1 and Cluster 2. Cluster 1 is a group of high processing speed systems and Cluster 2 is a hybrid system that consists of both high and low processing speed systems. Grid Scheduler allocates the high complexity jobs to Cluster 1 and low complexity jobs to Cluster 2 for scheduling. The term Resource Broker is used in the place of scheduler; it calculates the total number of tasks that are already holding by the resources before job scheduling. A Resource Broker assigns a task to a resource that has a minimum number of jobs.

Once the high priority tasks are allocated, then it assigns low priority tasks to resources for scheduling. This process will continue until all the tasks are mapped. In this approach, tasks are allocated equally to all the machines; therefore, developed work minimizes the makespan, maximizes resource utilization and reduces load balancing level effectively. For performance measurement, authors used Makespan and Average resource utilization rate to measure the performance of the grid environment. Table 1 shows the values for makespan comparison for a hundred seconds of four different algorithms and figure 2 represent the table value as a graph.

Table 1: Makespan comparison for Hundred Seconds

Task	**QSPBS**	**QWMTM**	**MINMIN**	**MAXMIN**
1	176	254	392	404
2	171	205	399	387
3	171	320	422	430

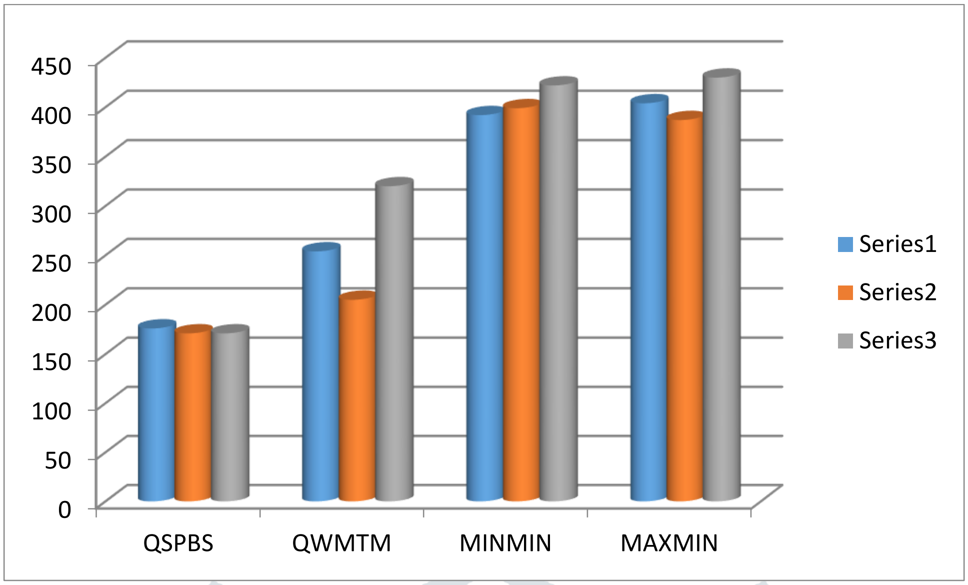

Figure 2: Makespan Values for Hundred Seconds

Vahe Aghazarian *et al.* introduced a suitable method in graph leveling for accurate using with task leveling. This leveling of graphs is being done in three levels: In zero level, the graphs should create scheduled order using a new algorithm for prioritizing of each node. This process leads to a more optimized scheduling order rather than previous methods. The node-level another index parameter is the number of graph leveling. For creating the graph, an optimized level is being used from mixed-function for covering all its states. For node level of L, it will create 2L states for priority function and this parameter causes the second level of nodes to not start working until the first level qualification function improvement in output result and the result from this function to reach to real value are final. This priority process will repeat for the next levels.

The needed processing power value is one of the index parameters in RQSG algorithm that introduces the needed processing power value for running the processing source. This parameter is being used in the certification function because of its positive effect on running the tasks in the scheduling process.

Table 2: Allocated tasks for various Algorithms

Algorithms	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
NG	934	875	819	756	679	612
SGA	835	789	736	699	645	590
RQSG	800	760	720	685	632	582
RQSG 1	790	754	712	678	625	575

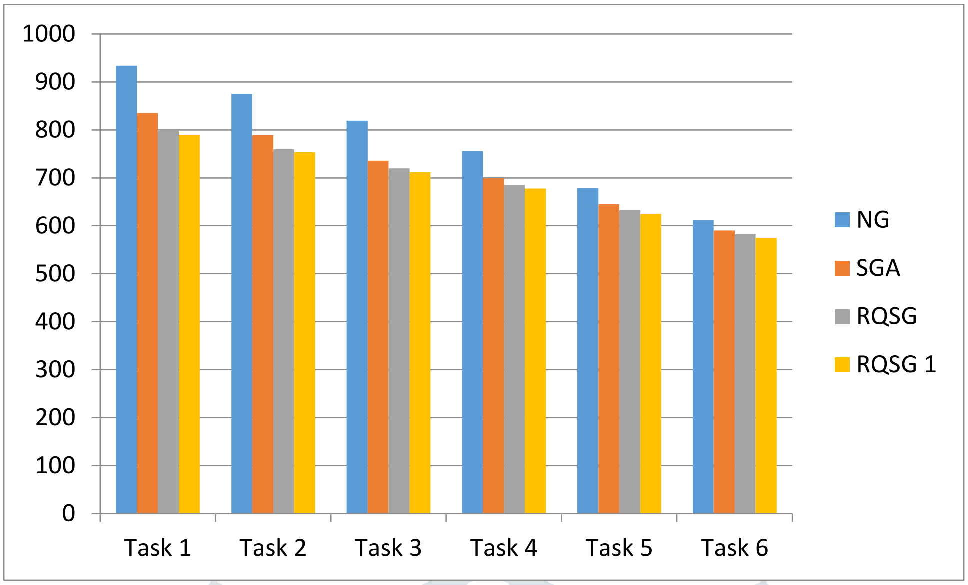

Figure 3: Available nodes for independent task

Amount of sent information between nodes: Amount of sent information between nodes is an index parameter that is being added in the proposed algorithm comparing standard RQSG algorithm for exploring of using an amount of bandwidth between processing elements. Another index parameter that is used in RQSG algorithm is the number of output vertex from each node in the studied graph and this parameter determines the release amount of nodes, which have finished their tasks and now are being allocated a selected task in processing source.

V. CONCLUSION

In this paper, major techniques that are used for resource allocation in grid computing are explained briefly. A comparison of various parameters like makespan, resource utilization, response time, monitoring, load balancing was prepared to get feedback on different types of resource scheduling. The concepts of resource allocation based on certain points of view in the literature are illustrated briefly and two important techniques are reviewed with different parameters. Results of the proposed method are compared with a different set of existing algorithms for confirmation. The roles of the algorithms are differentiated clearly with a neat graph. Both techniques used different parameters to measure the performance of the grid environment adequately. Further, future work will be based on the above findings and it will able to develop a more efficient algorithm for resource scheduling that will reduce the preprocessing time of jobs and considering memory constraint for resource selection with frequent parameters of measuring.

REFERENCES

1. Amudha T, T Dhivyaprabha, "QoS Priority-Based Scheduling Algorithm and Proposed Framework for Task Scheduling in a Grid Environment", International Conference on Recent Trends in Information Technology, IEEE, 2011.
2. Murugesan, C. Chellappan, "An Economic-based Resource Management and Scheduling for Grid Computing Applications", IJCSI International Journal of Computer Science Issues, Volume 7, Issue 2, 2010, pp. 20-25.
3. Boopathi Kumar E, Thiagarasu V, "Big Data and Its Applications: A Review", International Journal of Intelligent Computing and Technology (IJICT), Volume 2, Issue 2, 2019, Pages: 1 – 10.
4. Farhad Soleimanian Gharehchopogh, Majid Ahadi, Isa Maleki, Ramin Habibpour, and Amin Kamalinia, "Analysis of Scheduling Algorithms in Grid Computing Environment", International Journal of Innovation and Applied Studies, Volume 4, Issue 3, 2013, pp. 560-567.

5. Harshadkumar B. Prajapati, Vipul A. Shah, “Scheduling in Grid Computing Environment”, Fourth International Conference on Advanced Computing & Communication Technologies, IEEE, 2014, pp. 315-324.
6. Massimiliano Caramia, Stefano Giordani, “Resource Allocation in Grid Computing: An Economic Model”, WSEAS Transactions on Computer Research, Issue 1, Volume 3, 2008, pp. 19-27.
7. Mateusz Andrychowicz, “Optimization of distribution systems by using RES allocation and grid development”, IEEE, 2018.
8. Miguel Pinto, Orlando Belo, “Allocating Data Cubes on a Grid Environment”, International Conference on Systems, Man, and Cybernetics, IEEE, 2013.
9. Ritu Garg, Awadhesh Kumar Singh, “Adaptive workflow scheduling in grid computing based on dynamic resource availability”, Engineering Science and Technology, an International Journal, 2015, pp. 1-14.
10. S. Vaaheedha Kfatheen, A. Marimuthu, “ETS: An Efficient Task Scheduling Algorithm for Grid Computing”, Advances in Computational Sciences and Technology, Volume 10, Number 10, 2017, pp. 2911-2925.
11. Seyyed Mojtaba Ahmadi, Behrooz Masoumi, Reza Pourshaikh, "Resource Allocation in Grid Systems Using Collective Case-Based Reasoning Methods And Learning Automata”, IEEE, 2013.
12. Shashi Bhushan Semwal, Amit Das, “Effective Time and Cost Based Task Scheduling In Grid Computing”, International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 5, Sep-Oct 2015, pp. 118-122.
13. Shiv Prakash, Deo Prakash Vidyarthi, “A Hybrid GABFO Scheduling for Optimal Makespan in Computational Grid”, International Journal of Applied Evolutionary Computation, Volume 5, Issue 3, 2014, pp. 57-83.
14. Syed Nasir Mehmood Shah, Ahmad Kamil Bin Mahmood, and Alan Oxley, "Hybrid Resource Allocation Method for Grid Computing", Second International Conference on Computer Research and Development, IEEE, 2010.
15. Vahe Aghazarian, Arash Ghorbannia, Nima Ghazanfari Motlagh, Mohsen Khajeh Naeini, "RQSG-I: An Optimized Real-time Scheduling Algorithm for Tasks Allocation in Grid Environments”, IEEE, 2011.
16. Victor Toporkov, Dmitry Yemelyanov, Alexander Bobchenkov, Petr Potekhin, “Fair Resource Allocation and Metascheduling in Grid with VO Stakeholders Preferences”, International Conference on Parallel Processing Workshops, IEEE, 2016.
17. Zahra Pooranian, Mohammad Shojafar, Jemal H. Abawajy, Ajith Abraham, “An efficient meta-heuristic algorithm for grid computing”, Journal of Comb Optim, Springer, 2013, pp.1-22.