Volume 4, Issue 6, June 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Dynamic Load Balancing By Scheduling In Computational Grid System

Rajesh Kumar Gupta[#1]
Department of CSE, Al-Falah school of Engg. & Tech., MDU
India

Jawed Ahmed[#2]
Department of CSE, Jamia Hamdard University
India

Abstract - Grid computing systems are distributed systems that involve coordinate and involvement of heterogeneous resources with various characteristics where user jobs can be executed on either local or remote computer. These heterogeneous computing resources are used to run highly complex programs that require very high processing power and huge volume of input data. Recently the biggest issue in distributed system is to design of an appropriate and efficient dynamic load balancing algorithm that upgrade the overall performance of the distributed systems. In this research paper, we proposed a scheduling algorithm that manages the resources to improve the utilization of resource and minimize the job response time in computational grid system. So that no any resources will be heavily, low loaded or in some case will be in idle.

Keywords— Grid computation, Dynamic Load balancing, Load Migration, Schedule-DLB, Job Scheduling

I . INTRODUCTION

The term "Grid Computing" refers to a distributed computing infrastructure for advanced Science & Technology. Grid computing helps in the effective utilization of computing resources over the intranet/internet. Computational grid systems are distributed systems developed for heterogeneous resources, that heterogeneous computing resources can be personal computers, laptops, supercomputers, clusters, software's etc. [4]. As like Cluster is the collection of homogeneous elements, same as Grid is a collection of heterogeneous clusters interconnected to each other. Grid connected computers in the combination of computer resources applied to a common job. Grid is a type of a distributed system which supports the sharing and coordinates the resources used. This is due to the characteristics of Grid computing and to the complex nature of the problem itself. Grids have a lot of specific characteristics, like heterogeneity, autonomy and dynamicity, which remain obstacles for applications to harness conventional load balancing algorithms directly. To improve the performance of clusters in a grid, workload is balanced among the clusters. A workload is defined to be a job which can be an independent program or a partitioned module of a parallel program. In a grid some clusters may be heavily loaded while some may be idle. Therefore the main challenge of grid computing is the distribution of workload among physically dispersed clusters during run time. Hence, a load balancing algorithm attempts to improve the response time of user's submitted applications by ensuring maximal utilization of available resources. The main goal is to prevent, if possible, the condition where some processors are overloaded with a set of jobs while others are lightly loaded or even idle.

The computational Grid is the cooperation of distributed computer systems where user jobs can be executed on either local or remote computer systems [3]. With advanced use of heterogeneous resources, a effective scheduling and efficient load balancing approach is required for improving the performance of the system. In computational grid system the key performance metrics are Average Execution Time and the Average Response Time of jobs. Minimizing the average response time is often the goal of load balancing. The system load is a measure of the amount of work that a computer system performs [3]. If Grid system is unbalanced then some resources are heavily loaded than other or some is be in idle because of no proper scheduling of jobs. By scheduling of jobs in Grid System over all the resources available aimed to distribute the work load to all resources approximately equal.

At different environments the execution time of jobs will be vary. So, job scheduling in grid system is not an easy task where the streams of applications from different sources are arriving. Scheduling the jobs to set of available computing resources so that minimize the total completion time and improve the response time. This scheduling requires the matching of different jobs with the machines that satisfy their resource requirement. There are two different goals for job scheduling [5]:

i. Increasing computing performance, its aim is to minimize the execution time of each application that is considered in parallel processing [5].

ii. Increasing overall throughput, its purpose is to schedule a set of independent tasks in such a way that it increases the processing capacity of the systems for long period of time [5].

Our aim to propose an efficient job scheduling algorithm for grid system that provides efficient utilization of resources. This scheduling algorithm tries to minimize the total service time of the jobs and improve the response time for arriving jobs.

II. DYNAMIC LOAD BALANCING

Dynamic load balancing algorithms are always better than static as per as overload rejection, reliability, adaptability, cooperativeness, fault tolerant, resource utilization, response & waiting time and throughput is concert [10]. Dynamic load balancing in Grid computing is a technique to maximize resource utilization, utilizing parallelism, exploiting throughput, and to reducing the response time for arriving jobs so that an appropriate distribution of the application in system.

Load balancing methods in conventional parallel and distributed systems has been intensively studied; they do not work in Grid architectures because these two classes of environments are radically distinct [12]. Schedule of tasks on multiprocessors or multi computers supposes that processors are homogeneous and linked with homogeneous and fast networks. This assumption is not applied in Grid architecture because these following properties characterize them.

Heterogeneity: A Grid involves too many numbers of resources that are heterogeneous in nature and might span numerous administrative domains across a potentially global expanse.

Scalability: In Grid there are no limits for the resources and that reaches from few resources to millions. And that arises problems of potential performance as the size of a Grid increases.

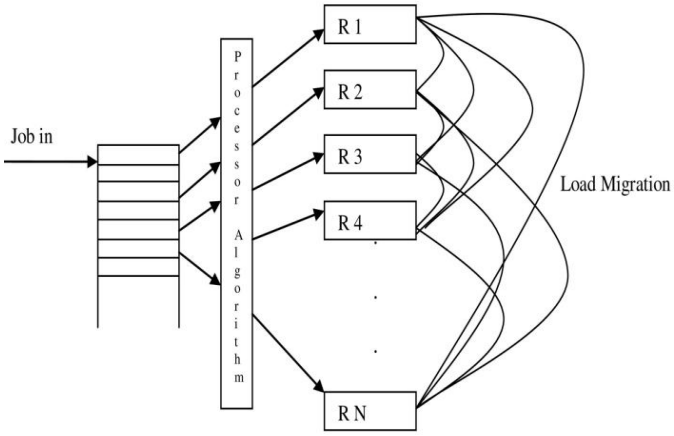

Figure:1 Load Balancing to avoid overload

Adaptability: In a Grid, if failure accrues for a resource, that is not exception. That means that the probability of some resources fail is naturally high. Resource managers must tailor their behavior dynamically so that they can extract the maximum performance from the available resources and services.

These properties make the load balancing problem more complex than in traditional parallel and distributed systems, which offer homogeneity and stability of their resources. Also interconnected networks on grids have very disparate performances and tasks submitted to the system can be very diversified and irregular. These various observations show that it is very difficult to define a load balancing system which can integrate all these factors.

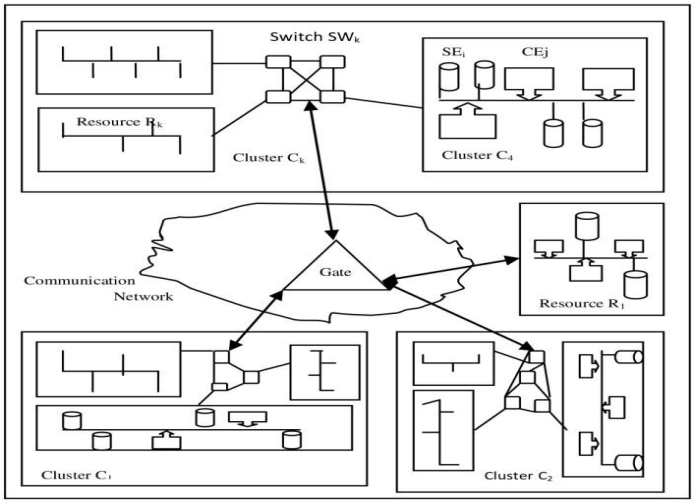

Figure:2 Grid Topology

In the grid computing (Fig. 2) there is a finite set of clusters Ck, which are interconnected by gates, where i,j,k ∈ {0, ...,G − 1}, where each cluster contains one or more resources S_j interconnected by switches SW_k and each resource contains some Computing Elements CE_j and some Storage Elements SE_i, interconnected by a local area network.

III. PREVIOUS WORK

In Grid System, there are two methods for balancing the load in Grid environment: static load balancing which is based on static information such as CPU capacity, memory space and predefine algorithm etc that will not rebalance further. Dynamic load balancing is based on current status of the system [8]. In some algorithms the combination both static and dynamic methods are used which are referred as combined algorithms.

Load balancing algorithms is fully dependent upon in which condition workload assigned to the different resources in the clusters, during compile time or execution time of jobs[11]. dynamic load balancing algorithms always gives better result than static load balancing as per as overload rejection, adaptability, fault tolerant, reliability, utilization of resource, parallelism, response time & waiting time and throughput is concert [10]. As the increase the number of jobs in the system due to large number of resources used by user the previous load balancing algorithms are not as sufficient to perform efficiently for distributing workload to provide greater performance [9].

There are some different algorithms for dynamic load balancing in Grid system.

Sender-Initiated Algorithm:

In the sender initiated algorithms let the heavily loaded resources take the initiative to request the lightly loaded resources to take the jobs .There are three basic decisions that need to be made before a transfer of a job can take place [12]:

1. Transfer policy: This policy decides when a resource become the sender and request to other resource.
2. Selection policy: This policy decides how any other resource chooses for transfer the job which is low loaded.
3. Location policy: This policy finds the location of the desired resource in the cluster.

A resource can activate the sender initiated algorithms when its queue size exceeds over some threshold value when job is arrived.

Receiver-Initiated Algorithm:

Receiver initiated algorithms is just like sender initiated algorithm in this algorithm low loaded resources search the heavily loaded resources and request to send their jobs to overcome the workload. It uses the similar transfer policy as the sender-initiated algorithm, which activates the pull operation when its queue length falls below a certain threshold, upon the departure of a job receiver [12]. Performance of the receiver initiated algorithm better than the sender-initiated algorithm for load balancing in grid system [12].

Symmetrically- Initiated Algorithm:

Symmetrically initiated algorithms basically work on both the above algorithm and combine the advantages of both sender and receiver initiated algorithms to improve the efficiency and response time for the jobs. This algorithm can perform the symmetrically and generally works better in almost all cases when the queue size exceeds or bellow some threshold value [6].

Central Scheduler Algorithm:

Central scheduler was an effective algorithm that can handle all load-balancing decisions with minimal inter processor communication [1]. The portability of the method for different kinds of applications with different kinds of meshes, and the number of parameters that must be set to obtain optimal performance from the method [2]. Each processor can send tasks to any of its neighboring processors without having any prior knowledge of the system [13]. There are several heuristics algorithms have been proposed to minimize the total completion time of the tasks in grid systems [7].

IV. PROBLEM STATEMENT

A typical distributed system will have a number of interconnected resources who can work independently or in cooperation with each other. Each resource has owner workload, which represents an amount of work to be performed and every one may have a different processing capability. To minimize the time needed to perform all tasks, the workload has to be evenly distributed over all resources based on their processing speed. The essential objective of a load balancing consists primarily in optimizing the average response time of applications, which often means maintaining the workload proportionally equivalent on the whole resources of a system.

The centralized approach is a simple approach and is beneficial when the communication cost is less significant. It is mainly used for a small size grid. Although the centralized approach is used currently, it limits the scalability of the grid by becoming a bottle neck.

Also failure of central controller can cause the entire system to fail. In the decentralized approach all nodes in the grid are involved in making the load balancing decision.

Sender initiated algorithms let the heavily loaded nodes take the initiative to request the lightly loaded nodes to receive the jobs. Receiver-initiated algorithm does not require an information process to collect the load information of other nodes, which incurs heavy communication traffic.

V. PROPOSED METHODOLOGY

Schedule-DLB is dynamic load balancing policy which works in Grid System to balance the work load on different clusters distributed in Grid. This Algorithm first search that clusters which average load of resources be minimum and then to that resource which will be Idle or low loaded resources. The workload at each resource can be determined as: CPU utilization, CPU speed, expected completion time and queue length. The workload index is determined dynamically [11].

Let there are number of clusters present in the Grid System. And in each clusters there are numbers of resources are present which may be different in each clusters.

Let there are M number of jobs that have to be scheduled and workload wl(i) of jobs are submitted to N number of resources. Schedule-DLB allocates number of born jobs to N number of resources based on their fair completion time of each jobs. Each cluster CR_i contains the information of their resources other clusters using the state object O_i. Each CR_i have workload of their resources and the average workload of other clusters. Average workload of clusters are calculated by

$$\text{Average workload of cluster (awlc)} = \frac{\text{Total workload on that cluster}}{\text{Number of resource (N)}}$$

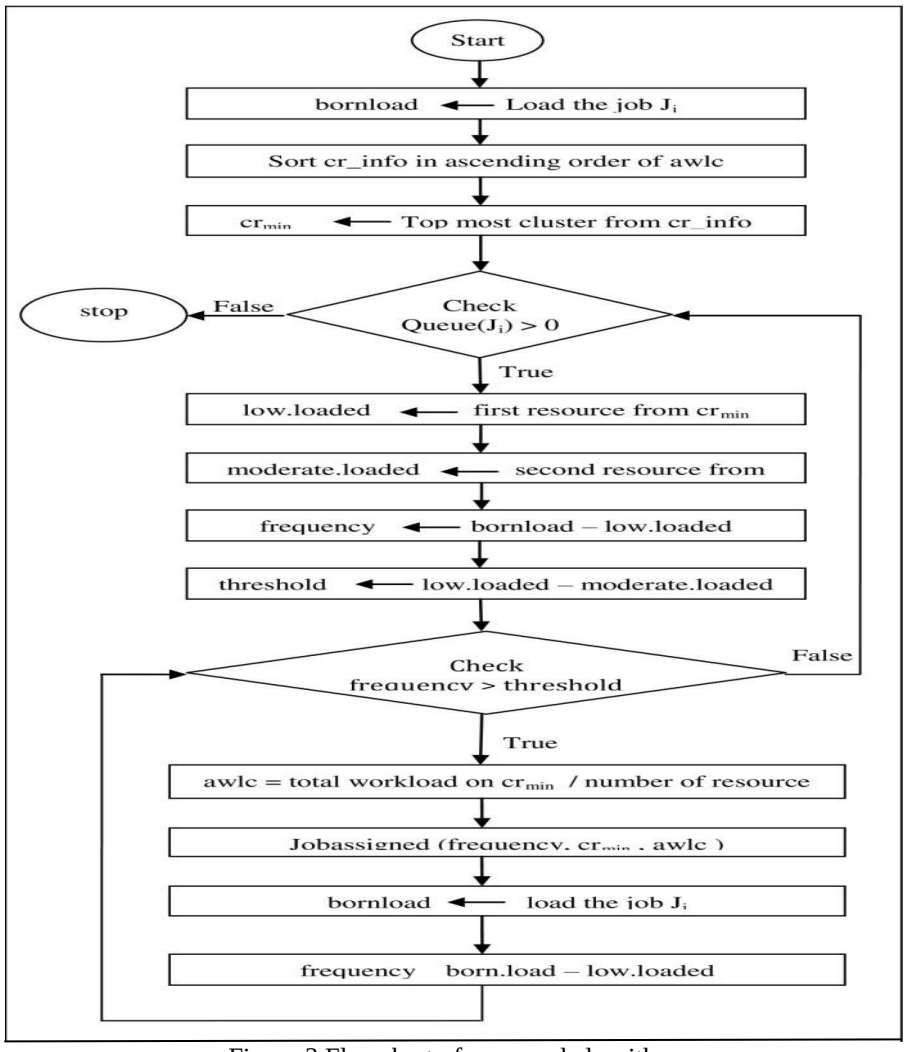

Figure:3 Flowchart of proposed algorithm

The state object O_i. helps a cluster to estimate the average workload of other clusters at any time. Each object in O_i have a property list O_i(awlc, N, T). O_i.awlc return the average workload of cluster CR_i, O_i.N return the number of resource present at that cluster CR_i and O_i.T return the current local time when the information is retrieved. Each cluster CR_i maintains the information about only its neighbor's cluster.

Schedule-DLB algorithm try to continuously reduce the average workload of each cluster CR_i and its neighboring clusters in the communication network by referring the jobs heavily loaded cluster to low loaded clusters. Flow chart for the proposed algorithm shown in figure3. This algorithm works in two phases. In first phase it works within the cluster to balance the

workload between the resources present in the cluster. And in second phase it works between all the clusters present in communication network. Schedule-DLB algorithm trigger when a new job born or any load information from their neighbors cluster. The entire clusters present in communication network exchange their average workload in dynamic nature.

VI. EXPERIMENT AND RESULT

GridSim toolkit*:*

The GridSim toolkit used as the simulation for the java-based discrete-event grid simulation toolkit. It allows simulation and modeling in distributed heterogeneous and parallel computing system for the user application and resource for evaluation for scheduling algorithm. It can also be used for the modeling and simulation of application scheduling on various classes of parallel and distributed computing systems such as clusters, grids and P2P networks [14].

There are some reasons why the GridSim toolkit was chosen over the other toolkit to simulate and evaluate our algorithm.

- It allows modeling of heterogeneous types of resources.
- Resource capability can be defined in the form if the Million Instructions Per Second (MIPS) and Standard Performance Evaluation Corporation (SPEC) benchmark.
- There is no limit on the number of application jobs that can be submitted to a resource.
- Network speed between resources can be specified.
- It supports simulation of both static and dynamic schedulers.
- Statistics of all or selected operations can be recorded.

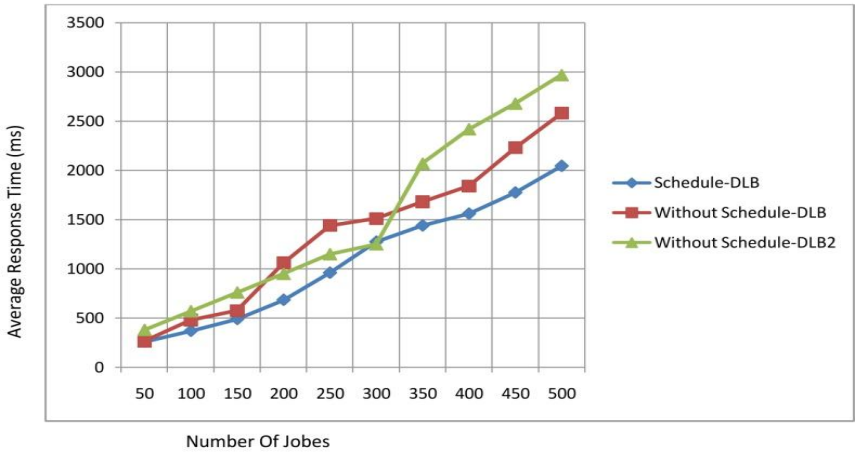

Figure4: Analysis with other load balancing algorithm

Proposed algorithm along with other previous algorithm is tasted accordance with different number of resource in different environment, we experiment with all attributes and resources are taken random, all algorithms were tested in same environment as simulator, hardware and operating system. Graph between number of jobs and average response time will be shown in figure4. The performance measurement of Schedule-DLB algorithm is based on four metrics: average response time, average workload of cluster, communication delay, number of jobs. As the number of jobs increases the average response time also increases but the increase of average response time is less in Schedule-DLB as compared to the increase in average response time to the other policy.

VII. CONCLUSION

In this paper, an efficient and improved dynamic load balancing algorithm namely Schedule-DLB for grid has been proposed where the availability of jobs and resources is dynamic. Experimental result, shown that the Schedule-DLB algorithm provides better response time for jobs, improve the resource utilization and balances the load in an effective manner. This experimental result also shows that the proposed algorithm enhanced the load balancing and for future work It can introduced more effectively with resource utilization using better scheduling approach.

REFERENCES

[1] Albert Y. Zomaya, Yee-Hwei Teh, "Observation on Using Genetic Algorithms for Dynamic Load Balancing", IEEE, Volume 12, Issue 9, September 2001.

[2]Roy D. Williams, "Performance of Dynamic Load Balancing Algorithms for Unstructured Mesh Calculations", Concurrent Supercomputing facilities, June 1990

[3] Saravanakumar E. and Gomathy Prathima, "A novel load balancing algorithm for computational grid", International Journal of Computational Intelligence Techniques, ISSN: 0976–0466 & E-ISSN: 0976–0474 Volume 1, Issue 1, 2010, PP-20-26.

[4] J. Dongarra, B. Tourancheau, "Special section: Cluster and computational grids for scientific computing", Future Generation Computer Systems, 21-30, 2008.

[5] Deepti Malhotra "A Minimum Execution Time Job Scheduling Algorithms in Simulated Grid Environment", Volume 3, Issue 12, December 2013 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering.

[6] Abhijit A. Rajguru, S.S. Apte, "A Comparative Performance Analysi of Load Balancing Algorithms in Distributed System using Qualitative Parameters", International Journal of recent Technology and Engineering, Volume 1, Issue 3, August 2012.

[7] Fahd Alharbi, "Simple Scheduling Algorithm with Load Balancing for Grid Computing", Asian Transactions on Computers, Volume 2, Issue 2,May 2012

[8] K. Koyama, K. Shimizu, H. Ashihara, Y. Zhang, H. Kameda, "Performance evaluation of adaptive load balancing policies in distributed systems", Proceedings of the Singapore International Conference on Networks/International Conference on Information Engineering, 606-611, 1993.

[9] Urjashree Patil, Rajashree Shedge, "Improved Hybrid Dynamic Load Balancing Algorithm for Distributed Environment", International Journal of Scientific and Research Publications, Volume 3, Issue 3,March 2013.

[10] Abhijit A. Rajguru, S.S. Apte, "A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters", International Journal of recent Technology and Engineering, Volume 1, Issue 3, August 2012.

[11] An Effective Dynamic Load Balancing Algorithm for Grid System" International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 8- August 2013.

[12] Toufik Taibi, Abdelouahab Abid, Engku Fariez Engku Azahan ,"A Comparison of Dynamic load Balancing Algorithms", J.J Appl. Science, Volume 9, Issue 2, 2007.

[13] C. Barmon, M.N. Faruqui, G.P. Battacharjee, "Dynamic Load Balancing Algorithm in a Distributed System", Elsevier, Volume 29,Issue 5,March 1991.

[14] Jasma Balasangameshwara, Nedunchezhian Raju "A hybrid policy for fault tolerant load balancing in grid computing environments" Journal of Network and Computer Applications 35 (2012) 412–422.

