A REVIEW ON DYNAMIC LOAD BALANCING BY GRID COMPUTING

[1]Afreen Jiwani, [2]Anagha Baraskar, [3]kajal Zodape, [4]Vaishali.R.Surjuse
[1]Student, [2]Student, [3]Student, [4]Assistant Professor
[1]Department of Computer Technology,
[1]K.D.K.C.E, Nagpur, India

***Abstract*— Grid computing technology can be seen as a positive alternative for implementing high-performance distributed computing. To carry out complex computational problems, grid computing is an emerging technology that involves coordinating and sharing of resources. Load balancing is the process of improving the performance of a parallel and distributed system through a redistribution of load among the processor. The main aim of load balancing is to provide a distributed, low cost scheme that balances the load across all the processors .Resource management and scheduling plays an important role in achieving high utilization of resources in grid computing environments. To achieve the performance in grid environment, many Job scheduling algorithms are implemented. To resolve these issues, we are proposing optimal hierarchical scheduling algorithm for grid environment and implementing it on a Grid simulation toolkit (GridSim) to analyze the performance of OHLBA algorithm with other algorithms in terms of efficiency.**

***IndexTerms*— Distributed System, Grid Computing, GridSim, Job Scheduling, Load Balancing, Resource Management.**

I. INTRODUCTION

Load Balancing is a process that allows jobs to move from one node to another within the distributed system. It is the process of roughly equalizing the work load among all computers of the distributed system. Global improvement in system has been produced due to load balancing. Load balancing goes one step further than load sharing, which avoids having some nodes idle in the distributed system when other nodes have too much work. Some of the main goals of a load balancing algorithm are:

1) To achieve a great performance of a system at a reasonable cost.
2) To have a fault tolerance
3) To treat all jobs in the system equally regardless of their origin.
4) To have the ability to modify itself in accordance with any changes or expand in the distributed system configuration.
5) To maintain system stability.[2]

Distributed Computing System is generated or developed for different types of distributed system and to improve the performance and there cost reduction. This system is a collection of nodes which are connected with each other through a network. This system is a collection of nodes which are connected with each other through a network. This type of system is mainly in heterogeneous nature. It means that the different networks are distributed over all the world. To increase the efficiency, the system needs to distribute the load in equal manner among the nodes according to their performance over the network. This is also the main purpose of distributed computing system. This system is also used in Grid Computing environment.

When one or more nodes are connected in the same network but it is observed in the network that some processor are lightly loaded or heavily loaded. The lightly loaded is the fast processor and the heavily loaded is the slower nodes.
Load balancing algorithm is important in balancing the work load in several network. This algorithm define the work load in the form of number of resources and also based or depend on threshold value.

Load balancing [3] is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded. A load balancing protocol is dynamic in nature doesn't contemplate the previous state or behavior of the system, that is, it depends on the current behavior of the system. It is common these days in redundant high-availability computer systems that incoming network traffic is distributed on network level by deploying one of the frequently used network load balancing algorithms like:-random-allocation, round-robin allocation, weighted round-robin allocation, etc). These algorithms use solely network parameters of incoming traffic to create selections wherever to forward traffic, with none data from different elements of database system, like current load of application or info servers. Since these days it is extremely common to possess internet servers acting as application servers, it is usual that load balancers use session-switching technique, which suggests that once a user opens website on one server, it will stay on it server whereas the session lasts. Depending on who initiated the process, load balancing algorithms can be of five categories:

- Sender Initiated:

If the load balancing algorithm is initialized by the sender

- Receiver Initiated:

If the load balancing algorithm is initiated by the receiver

- Symmetric:

It is the combination of both sender initiated and receiver initiated

- Static:

It doesn't depend on the current state of the system. Prior knowledge of the system is needed.

- Dynamic:

Decisions on load balancing are based on current state of the system. No prior knowledge is needed. So it is better than static approach.

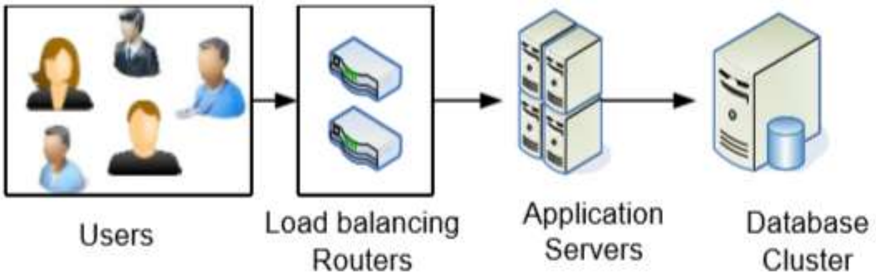

Figure 3 Schematics of typical high-availability computer system with hardware load balancers.

II. Literature survey

In the literature, many scheduling algorithms have been proposed. Most of them can be applied to the grid computing environment with suitable modifications. Though method of good scheduling algorithm, application can avoid unnecessary delays and the system perform better.

- Round Robin scheduling algorithm (RR):

Round Robin focus on the fairness problem .The RR algorithm defines a ring as its queen and also defines a fixed time quantum. Each job can be executed only within this quantum, and in turn. If the job cannot be completed in this quantum, it will return to the queue and wait for the next round. The major advantage of RR algorithm is that jobs are executed in turn and do not need to wait for the previous job completion. Therefore, it does not suffer from a starvation problem. However, if the job queue is fully loaded or workload is heavy, it will take a lot of time to complete all the jobs. Furthermore, a suitable time quantum is difficult to decide.[4]

- Min-min and max-min algorithm

The min-min scheduling sets the jobs that can be completed earliest with the highest priority. Each job will always be assigned to the resources that can complete it earliest. Similar to Min–min algorithm, Max–min algorithm sets the highest priority to the job with the maximum earliest completion time.[5] The main idea of Max–min algorithm is to overlap long running tasks with short-running tasks.Max–min can be used in cases where there are many shorter tasks than there are longer tasks. For example, if there is only one long task, Min–min will first execute many short jobs concurrently, and then execute the long task. Max–min will execute short jobsconcurrently with the long job.[6]

- Most Fit Task scheduling algorithm (MFTF)

The MFTF algorithm mainly attempts to discover the fitness between tasks and resources for user. It assigns resources to tasks according to a fitness value, and the value is calculated as follows: Fitness(i, j) = 10000/(1 + |Wi /Sj− Ei |) (1) where Wi is the workload of the ith task, Sj is the CPU speed ofthejth node, and Ei is the expected time of the ith task. Wi /Sjisthe expected execution time using this node. |Wi /Sj− Ei | is thedifference of the estimated execution time and the expected taskexecution time. Ei is determined by the user or estimated by themachine. How to set Ei is calculated by (2). Ei= A + n × S (2) where A is the average response time;n is a non-negative real number and S is the standard deviation oftask response time.When the estimated execution time is closer to Ei , it meansthat the node is more suitable for the task. However, the MFTF scheduling algorithm has some problems for estimating. It does not consider the resource utilization, and the estimated function is an ideal method. Therefore, incorrect scheduling may occur in the real environment.[7]

- ACO algorithms in job scheduling

Ant Colony Optimization (ACO) was used for solving the scheduling problem in grids in recent years. Xu et al. proposed a simple grid simulation architecture and modified the basic ant algorithm for job scheduling in grid. The scheduling algorithm proposed in the paper needs some information such as the number of CPUs, Million Instructions Per Second (MIPS) of every CPU for job scheduling. A resource must submit the information mentioned above to the resource monitor. However the load of the better resources will be more than others and it will decrease the performance of job scheduling.[8]

- First Come First Served scheduling algorithm (FCFS)

  In this algorithm, jobs are executed according to the order of job arriving time. The next job will be executed in turn. The FCFS algorithm may induce a “convoy effect”. The convoy effect happens when there is a job with a large amount of workload in the job queue. When this occurs, all the jobs queued behind it must wait a long time for the long job to finish.[9]

III. Proposed System and Result

The grid computing environment require standard scheduling algorithm for task distribution, to satisfy end user expectations with improved performance and throughput. To reduce the response time of end user’s submitted the jobs by maximizing the utilization of the available resources, a standard scheduling algorithm is used. In this, we prefer optimal hierarchical structure.The system consist of various module. They are Controller Module, Requestor Module, Request handling, Load calculation and assign task as shown in fig.2**.**Thecontroller add all the sub server with its configurations parameter. The requester,register and login to upload data on server, send a request of requester to data center i.e. controller. Controller, handles all the request from requester, calculate load of sub servers as per request, assign a task to the sub server which have minimum load using OHLBA.

2.1 Process Execution Cycle

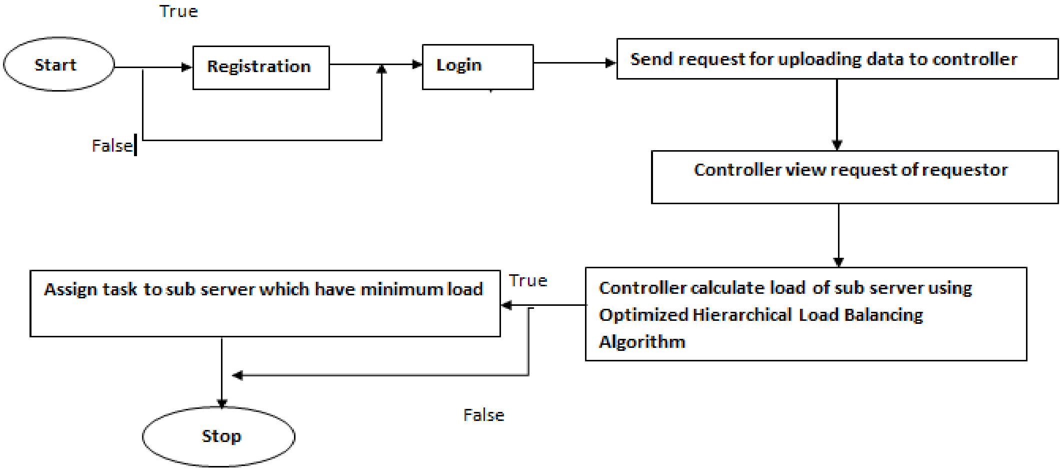
True
Start
Registration
Login
Send request for uploading data to controller
False
Controller view request of requestor
Assign task to sub server which have minimum load
True
Controller calculate load of sub server using Optimized Hierarchical Load Balancing Algorithm
False
Stop

Fig2: Process Execution in Hierarchical Grid

III. Conclusion

Load balancing is one of the main issues in the grid environment. Recent researches have proved that loadbalancing on computational grids is best solved heuristic approach. The various scheduling and the load balancing criteria have been reviewed along with the various tools associated with the grid scheduling .The paper provides an study on the scheduling of various task in different application areas .It is intended to propose a scheduling algorithm to improve performance in an E-Governance application for effective scheduling of the various tasks.

IV. References

[1] Sandeep Sharma Sarrabjit Singh, and Meenakshi Sharma, “Performance Analysis of Load Balancing Algorithms”, World Academy of Science,Engineering and Technology 38 2008.

[2] Ali M. Alakeel, “A Guide to Dynamic Load Balancing in Distributed Computer Systems”, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No,6,june 2010

[3] K. Ramana, A. Subramanyam and A. AnandaRao, Comparative Analysis of Distributed Web Server System Load Balancing Algorithms Using Qualitative Parameters, VSRD-IJCSIT, Vol. 1 (8), 2011, 592-600 IJCSI.

[4] Sheng-De Wang, I-Tar Hsu, Zheng-Yi Huang, “Dynamic scheduling methods forcomputational grid environments”, International Conference on Parallel andDistributed Systems 1 (2005) 22–28.

[5] Ruay-Shiung Chang, Chih-Yuan Lin, Chun-Fu Lin, “An Adaptive Scoring Job Scheduling algorithm for grid computing”, Future Generation Computer Systems 207 (2012) 79–89.

[6] Ramya R and Shalini Thomas “An Optimal Job Scheduling Algorithm in Computational Grids”, ICCMIT 2012.

[7] S. Sharma, S. Singh and M. Sharma, “Performance Analysis of Load Balancing Algorithms”, WorldAcademy of Science, Engineering and Technology, vol. 38, (2008), pp. 269-272.

[8] R. S. Chang, C. F. lin and J. J. Chen, “Selecting the most fitting resource for task execution”, Future Generation Computer Systems, vol. 27, (2011), pp. 227-231.

[9] Ungurean, “Job Scheduling Algorithm based on Dynamic Management of Resources Provided by Grid Computing Systems”, Electronics and Electrical engineering, vol. 103, no. 7, (2010).