IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Issue-1 E-ISSN: 2347-2693

Grid Computing Approach for Dynamic Load Balancing

Kapil B. Morey[1*], Sachin B. Jadhav[2]

[1*,2] *Dept. Of Computer Engineering,*
Sant Gadge Baba Amravati University, India

www.ijcseonline.org

Received: Dec /15/2015 Revised: Dec/28/2016 Accepted: Jan/11/2016 Published: Jan/30/ 2016

***Abstract*—** Grid computing technology can be seen as a positive alternative for implementing high-performance distributed computing. The goal of Grid computing is to create the illusion of virtual computer out of a large collection of connected heterogeneous nodes sharing various resources. The Grid system needs competent load balancing algorithms for the distribution of tasks in order to increase performance and efficiency. The process of identifying requirements, matching resources to applications, allocating those resources, and scheduling and monitoring grid resources over time in order to run grid applications efficiently is known as Grid Resource Management. The first phase of resource management is resource discovery. The next step is monitoring and scheduling. Monitoring keeps the details of the resources and scheduling guides the job to appropriate resource. The resources which are heavily loaded act as server of task and the resources which are lightly loaded act as receiver of task. Task relocation takes place from the node which has high load towards the node which has fewer loads. The main aim of load balancing is to provide a distributed, low cost scheme that balances the load across all the processors. In this paper a dynamic load balancing algorithm is proposed in a simulated grid environment which fulfils the objective to achieve high performance computing by optimal usage of geographically distributed and heterogeneous resources in the grid environment.

***Keywords*—**Load; Computing; Grid; Dynamic

I. INTRODUCTION

The improvement of performance of computers and their cost reduction is due to development in computing resources. The recent researches on computing architectures has resulted the emergence of a new computing paradigm known as Grid computing. Grid is a kind of distributed system which maintains the sharing and synchronized use of geographically scattered and multi owner resources autonomously from their physical type and location, in dynamic implicit organizations that share the same objective of solving large-scale applications. The main aim is to prevent the condition where some processors are overloaded with a set of tasks while others are lightly loaded or even idle [4, 5]. The load balancing of the jobs in a grid environment can considerably influence grid's performance as the resources are dynamic in nature. Hence the load of resources changes with variation in configuration of grid. A key characteristic of Grids is that resources like CPU cycles and network capacities are shared among various applications, and therefore, the amount of resources available to any given application highly fluctuates over time. Load balancing uses parallelism for boosting throughput in order to minimize the response time by intelligently distributing application resources. Load balancing can be implemented using two techniques i.e. static load balancing and dynamic load balancing. Static load balancing algorithms assign the jobs of a parallel program to workstations based on either the load at the time nodes are allocated to some task, or based on an average load of our workstation cluster. Dynamic load balancing algorithms alter distribution of work during runtime. Dynamic load balancing algorithms use the updated current information of the load for taking decisions about load distribution.

II. GRID STRUCTURE

The simple grid can be defined as an interconnected system for a distribution of non interactive workloads that involve a large number of files. Grid can be built in all sizes ranging from just a few machines to the group of machines organized as a hierarchy spanning the world. The Simplest Grid consist of just a few machines all the same hardware architecture and same operating system connected on a local network. Because the machines have the same architecture choosing application software for these machines is usually simple. This type of structure the grid is known as Intergrid which is homogeneous systems. When the machines are been include to the heterogeneous systems, more resources are available. File sharing may still be accomplished using network file systems. Such a grid is referred as an Intragrid [3].

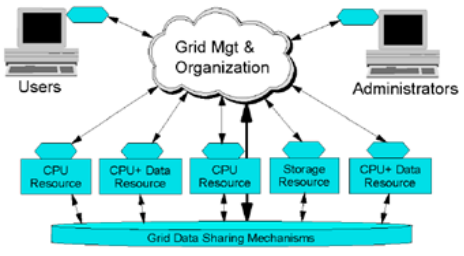

Fig.1: Grid Structure.

III. DYNAMIC LOAD BALANCING

Load balancing algorithms can be defined by their implementation of few policies [5]. The Information policy states the workload of task information to be collected, when it is to be collected and from where it is to be collected. The next policy, Triggering policy determines the appropriate period to start a load balancing operation. The Resource type policy orders a resource as server or receiver of tasks according to its availability status. The Location policy uses the results of the resource type policy to find a suitable partner for a server or receiver. The final policy i.e. Selection policy determines the tasks to be relocated from the node which has highly loaded resources towards the node which has fewer loaded resources. The load balancing algorithms are implemented to allocate the tasks of a parallel program to workstations. Multicomputer with dynamic load balancing allocates or reallocates the resources at runtime based on task information. In our approach Dynamic Load Balancing Algorithm is implemented to multicomputer based on resource type policy [5]. Load balancing feature can prove invaluable for handling occasional peak loads of activity in parts of a vast organization. There are important issues in Load Balancing [4, 5]. If the Grid is already fully utilized, the lowest priority work being performed on the Grid can be temporarily suspended or even cancelled and performed again later to make room for the higher priority work. The load arrangement in grid environment is altered by some actions. These actions can be categorized as follows:

- Arrival of any new job and queuing of that job to any particular node.
- Allocation of job to meticulous processor.
- Reallocating the job if load is unbalanced.
- Allocate the job to processor when it is free.
- Releasing the processor after it completes the entire job.

IV. SCHEDULING, GRID COMPITING AND LOAD BALANCING ALGORITHM

A. *Scheduling*

The scheduling algorithms do not adequately address congestion, and they do not take fairness considerations into account. Fairness is needed for proper scheduling of jobs. In Fair Scheduling, the jobs are allocated to multiple processors so that the tasks with unsatisfied demand get equal shares of time. The completion time of the jobs is used to determine scheduling queue of the jobs. The evaluation of completion time of the job is done by task rate using a max min fair sharing algorithm. The job is alloted to processor in accordance with growing degree of completion time. In scheduling algorithm, higher order tasks are completed first which means that tasks are taken a higher priority than the others which leads to starvation that increases the completion time of tasks and load balance is not guaranteed. To overcome this we put forward a load balancing algorithm to provide unvarying load to the resources so that all jobs are uniformly assigned to processor depending on balanced fair rates. The main aim of this algorithm is to reduce the overall time required to complete the processing.

B. *Grid Computing*

In order to evaluate the performance of Grid resource management and application scheduling algorithms, we need to conduct repeatable and controlled experiments, which are difficult due to Grid's inherent heterogeneity and its dynamic nature. This indicates that all users need to submit their tasks to the central scheduler that can be targeted to perform comprehensive optimization such as higher system utilization and overall satisfaction of user depending on resource allocation policy or optimize for high priority users. Some of the GridSim features are outlined below:

- It allows modeling of different resource characteristics and their failure properties; It enables simulation of workload traces taken from real supercomputers;
- It supports reservation-based or auction mechanisms for resource allocation; It allocates incoming jobs based on space-or time-shared mode;
- It has the ability to schedule compute- and/or data-intensive jobs;
- It has a background network traffic functionality based on a probabilistic distribution.

C. *Algorithm for Load Balancing*

Input: A set of R task and N number of processor with computational capacity Pj

Output: A Schedule of R tasks

1. Construct a set of queues
2. q_size < R/N
3. While tasks present in queue do,
4. Assign demand rate of the task Xi
5. k=P/R
6. If Di < v
7. Assign Di to ith task as fair rate
8. Else
9. Assign v to ith task as fair rate
10. Calculate fair completion time t(D)
11. End while
12. End loop
13. Arrange the task in increasing order based on their t(D) and submitted to processor
14. Calculate mean waiting time each scheduled task
15. If Zxy >0
16. Every processor having small amount capacity is selected for relocation.
17. End If
18. End While.

V. CONCLUSION

This algorithm is able to provide sound results in terms of make span and execution cost. Thus the algorithm assigns the job to the available processors so that all requesting jobs get identical time span which satisfies their demand. We have described several aspects of load balancing algorithm and established abundant notions which illustrate its extensive potential through this projected algorithm. This projected algorithm is definitely a capable tendency to solve all types of load influenced problems and high demanding applications. Objective of the grid environment is to achieve high performance computing. But grid application performance remains a challenge in dynamic grid environment. At any moment resources can be inhibited from grid and can be submitted to the grid. This paper presents efficient results of the balancing of resource load through a dynamic approach.

ACKNOWLEDGMENT

I warmly acknowledge and express my special gratitude to Prof. Sachin B. Jadhav for his earnest guidance, encouragement and infallible suggestions in my research work. I am also thankful to my family and colleagues for their incessant support.

REFERENCES

[1] Goswami S, Das A, "Deadline stringency based job scheduling in computational grid environment", Computing for Sustainable Global Development (INDIACom), 2nd International Conference, IEEE, ISBN: 978-9-3805-4415-1, Page No(**531-536),** March **2015,**.

[2] Abhishek M. Kinhekar, Prof. Hitesh Gupta, "A Review of Load Balancing in Grid Computing", International Journal of Advance Research in Computer Science and Management Studies, Vol. **2**, Issue **8**, Page No (**102-108),** August **2014**.

[3] Goswami S, De Sarkar A, "A Comparative Study of Load Balancing Algorithms in Computational Grid Environment", Computational Intelligence, Modelling and Simulation (CIMSim), Fifth International Conference, IEEE, Page No(**99-104),** September **2013**.

[4] PawandeepKaur, Harshpreet Singh, "Adaptive dynamic load balancing in grid computing an approach," International journal of engineering science & advanced technology, Volume-**2**, Issue-**3**, Page No(**625 – 632,** May-Jun **2012**.

[5] Prabhat Srivastava, "Improving Performance in Load Balancing Problem on the Grid Computing System", International Journal of Computer Applications, Volume **16**– No.**1**, Page No(**975– 8887),** February **2011**.

