IOSR Journal of Computer Science (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727
PP 17-22
www.iosrjournals.org

Hierarchical Scheduling Algorithm for Grid Computing Environment

Abhishek Nachankar[1], Prof. Rajesh Dharmik[2]
[1](IT Department, Yeshwantrao Chavan College of Engineering, India)
[2](IT Department, Yeshwantrao Chavan College of Engineering,India)

ABSTRACT : *Grid computing is increasingly being viewed as the next phase of distributed computing. Grid aims to maximize the utilization of an organization's computing resources by making them shareable across applications. In grid computing, job scheduling is an important task. Load balancing and proper resource allocation are critical issues that must be considered in managing a grid computing environment. Load Balancing is the technique which distributes the workload across multiple computers to reduce the latency of process execution with proper resource utilization. To resolve these issues, we are proposing hierarchical scheduling algorithm to provide a superior load balancing mechanism at grid level. And improved hierarchical scheduling algorithm which reduces makes span of the job and improves overall performance of the grid environment.*

Keywords *-Grid Computing,Hierarchical Environment, Job Scheduling,Load Balancing, ResourceUtilization.*

I. INTRODUCTION

Due to day to day progress of science and engineering technologies the related problems get more complicated as that of ever before. To solve these problems we need some powerful computing facility. Utilizing the resources around the environment is good concept.Grid computing ties unexploited processing cycles of all the computers for solving problem those are too intensive for any stand alone machine [1].

Grid computing suggests a model for solving massive computational problems by making use of the idle resources of large numbers of disparate computers, often desktop computers as well. Grid computing focus on the ability to support computation across administrative domains sets it apart from traditional computer clusters. Grid computing has the design goal of solving problems too big for any single supercomputer, whilst retaining the flexibility to work on multiple smaller problems. Therefore Grid computing provides a multi-user environment. Its secondary aims are better exploitation of available computing power and catering for the intermittent demands of large computational exercises [2].

A grid is or homogeneous resources to deal with large scale scientific problems. There are many issues in using grid computing. Appropriate and efficient allocation of tasks to the resources is called as job scheduling. Main purpose of job scheduling is to provide proper load balancing at higher level of nodes and reduce the response time by proper allocation of jobs. In grid computing, resource providers and tasks are changing constantly. So, no traditional scheduling algorithm works properly for dynamic grid system. It is very important to allocate tasks to proper resources by using good scheduling algorithm to enhance throughput and avoid unnecessary delays. Although many proposed scheduling algorithms proved that they are suitable for a dynamic environment, only little work has been done on the aspect of job scheduling considering the real time characteristics of grid resources.

Load balancing is very much overlapping issue that is recently studied very much because of need of high performance computing. The computation grid involves high performance platforms to heterogeneous dynamic and shared environment. Thus how to migrate or adapt load balancing schemes to Grid becomes the focus of this research area. For parallel applications, load balancing attempts to distribute the computation load across multiple processors or machines as evenly as possible with objective to improve performance.

Generally, a load balancing scheme consists of three phases: information collection, decision making and data migration. During the information collection phase, load balancer gathers the

information of workload distribution and the state of computing environment and detects whether there is a load imbalance. The decision making phase focuses on calculating an optimal data distribution, while the data migration phase transfers the excess amount of workload from overloaded processors to under loaded ones[3].

The rest of paper is organized as follows: Section 2 focuses on related work; Section 3 discusses the proposed work and algorithm; Section 4 we are coming up with conclusion.

II. Related Work

Resource manager is central authority of the system and its main responsibility is to accept requests from users, match user requests to available resources for which the user has access and schedule the matched resources. Such resource requests are considered as jobs by the Grid. A proper scheduling and efficient load-balancing algorithm is required for improving the performance of the system.

It is difficult to achieve load balancing in grid systems than in traditional distributed computing environment because of the heterogeneous and dynamic nature of the grid. Most of the studies present only centralized schemes. All of them suffer from significant deficiencies, such as scalability problems when we talk about the centralized approaches. In [4], the authors consider a hierarchical tree structure for grid computing services similar to ours.

The grid environment collects, integrates, and uses heterogeneous or homogeneous resources which scattered around the globe. A grid environment can be classified into two types: computing grids and data grids. In computing grid, job scheduling is a very important task [5]. A good scheduling algorithm can assign jobs to resources efficiently and can balance the system load. A job scheduling algorithm called Hierarchical Load Balanced Algorithm for Grid environment is used to balance the system load in determining a balance threshold.

When a job is to be assigned to a cluster with the highest Computing Power, the load of the selected cluster will be checked first whether it is already overloaded or not. If the cluster's average load is larger than the balance threshold, the cluster is marked as overloaded and the job will seek the cluster with the next highest ACP. This is called a local update. Local update is necessary because it reflects the situation changes in the local cluster. The average computing power and average load of other clusters are not affected. When a job is completed, the job had to release all resources and a global update will recalculate all parameters again.

Load balancing technique in distributed computing environment is used to optimize the scalability of the entire system. Numerous works have been proposed on the issues of process migration and load balancing [6]. An effort has been made in the present work to cite a reliable and comprehensive load balancing approach is based on the priority of the processes. This approach introduces a Process Migration Server that also acts as future cluster management server ensuring that the latency time in migrated process execution is reduced along with no starvation policy for any process.

Static and dynamic load balancing techniques are themostly used techniques for task allocation in grid environment [7]. Several works have been done on dynamicload balancing approach. Some are already implementedand some are still under research. No universal algorithmexists for load balancing. It can vary according to problemsize depending on different types of parameters. Little hasbeen done in case of multiple parameters [8]. In these areashierarchical load balancing algorithm works successfully.

III. PROPOSED SYSTEM

The proposed algorithm only works for hierarchical grid environment. The overall system and algorithm mainly focuses on assigning suitable resources to a job performing an adaptive load balancing algorithm between cluster nodes. Since jobs that we are selecting are computing intensive jobs, we consider the computing power of each resource as the standard for the selecting resources. The computing power is defined as the product of CPU speed and the idle CPU percentage. Besides

the computing power, the average load of each cluster is also considered in our algorithm. The average load of each resource is estimated by the weighted sum of squares method. We take three parameters into account, network utilization, memory utilization, and idle CPU percentage, in calculating the weighted sum of squares method. We use the status of each resource in the grid system to initialize the values which are needed for the scheduling algorithm. Those values will be adjusted by performing local update and global update. Therefore, the system can assign the job to the appropriate resource according to the updated information.

To improve the performance of the system, grid environment requires an efficient load balancing algorithm for task distribution. Main task for load balancing algorithm is to improve the response time of the user submitted applications by utilizing maximum amount of resources. Machines are arranged in both ways such as hierarchical or flat manner (non-hierarchical). In flat, systems are connected in such a way that, all of them able to communicate each other directly. Hierarchal organization machines in same level can directly communicate with the machines directly above them or below them, or peer to them in the hierarchy called second level workstation/ distributor or cluster node [7]. Most grid systems follow the hierarchical approach in order to improve the performance and efficiency. To show the efficiency of the algorithm here we are considering few parameters.

Avg. CPU Utilization: Percentage of CPU utilization involved at the time of job execution.
Available Heap Memory: Available heap memory used to execute the particular job.
Maximum Heap memory: Maximum heap memory availability in the main memory.
CPU Idle time: The current CPU idle time of during execution.
Available Processor: For particular job execution, how many processors are available at that time.
Node id: Indicates the unique identification of each local and remote node.
N: {n1, n2…} is the total number of local and remote nodes.
J: {j1, j2…..}
$\sum_{i=1}^{m} \boldsymbol{ExeTi}$: Total execution time of all jobs.

The Portal provides an interface for users to submit jobs.The Information Server discovers resource nodes registeredwith the system, and records the information of the resource,such as CPU speed, idle CPU percentage, memory utilization,average load of each cluster, etc. The job scheduler acceptsthe job from the portal and uses the HLBA with theinformation from Information Service to choose theappropriate cluster and compare its load with the system.Then, it selects the resource with the strongest computingpower in the cluster to execute the submitted job. After the jobis finished, the result and the new status of the resource willbe sent back to the Information Service for anotherscheduling.

A. System Framework:

In a computational grid, a number of machines are connected to each other as workstations as shown in fig1. It is a complete hierarchical structure which is grouped together with the help of middleware. From the topology this system is divided into three levels: L1 – Grid Node level, L2 – Master Node level, L3 – Workstations / Computing Node level. Each managerial node owns a set of worker nodes with the same LAN / WAN domain.

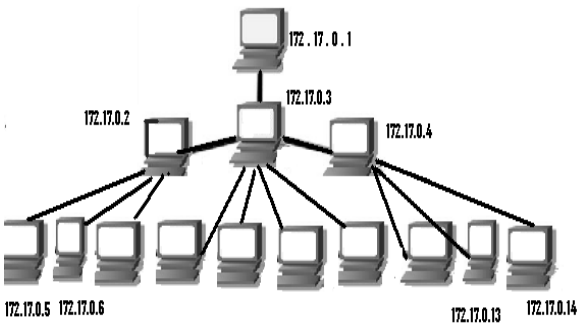

Fig1: hierarchical grid computing environment

The Hierarchical structure shown in the fig1 is completely independent of LAN they are connected. A Hierarchy of 3-level or more can also be formed by joining many number of worker nodes.This can be implemented in two possible ways. One is, to create the hierarchy logically on one machine by using Grid Simulator or any middleware [11]. Second is, to implement it physically by using as many nodes as possible on which dynamic load balancing can be applied. In this paper, we prefer the second option to implement ouralgorithm on physical real time hierarchy. In the above discussed model server node called primary load distributor node is only static for entire hierarchical structure and rest of level coordinator nodes and worker nodes are formed dynamically available at the moment. Second level distributor cluster node are formed by the primary distributor based on the need of distribution on a property of most available resource form the available list and set best nodes as second-level-cluster node. The second role of primary node is to distribute the available grid-nodes among the second-level-cluster node on the basis of total resources available on at that moment and total number of second-level cluster node. The number of second-level-cluster node depends on the total number of nodes available in grid factory and total number of jobs available to distribute; this is a simple approach to reduce the overhead communication cost on primary as well as second-level-cluster node.

Job Execution life Cycle:

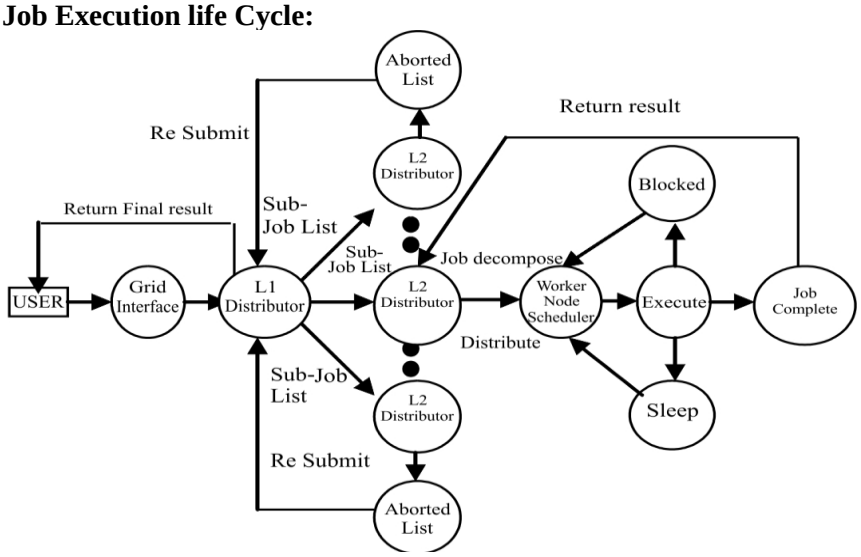

Fig2: job life cycle in hierarchical computational grid

Figure 2 depicts the life cycle of job execution process in a hierarchical grid environment and the sequence of operations shows the life span of hierarchical grid. In the figure L1, L2 and L3 depicts the operations that take place in levels 1, 2 or 3 respectively.

Role of Cluster Level Nodes:

During job distribution process, Grid node distributes the job on the basis of average computing power to cluster level nodes. Once job is allotted to the cluster with highest average computing power then the job isget split at the same level and mapping a group of jobs to the next level is done. Function of second-level cluster node is to distribute the jobs into their workers nodes on the basis of multi-parameter of grid worker nodes. Effective load distribution achieved on the basis of multiple parameter load distribution approach. The key feature of our distribution approach is on the basis of amount of resource available like CPU load, heap memory and CPU idle time on dynamic approach [4].

Job scheduling Algorithm for Grid Environment:

When the scheduler receives a job submitted by a user, it will transfer a request to the Information service in order to obtain the necessary information such as the idle CPU percentage of

each resource, average load of each cluster and average load of the system. And will sort clusters by their average loads.

The average load of the cluster is defined by the average load of each resource in cluster i. We use the weighted sum of squares method to measure the load of each resource, as

$$Load = \sqrt{a1\ L1}$$

Load – Load of each resources in the each cluster.

a1 - Load attribute.

L1 - Load attribute and the corresponding weighted value.

The average load of each cluster i(ALC)is defined as –

$$ALC = -\sum_{1}^{n} Load$$

n – No. of resources in each cluster.

Although the time complexity of sort algorithm is O (nlgn), the time complexity of HLBA is still O(mn). If the average load of cluster (ALC) exceeds the balance threshold (Ψ), it means that the cluster is overloaded. We sort the clusters which are under loaded and select the cluster with the highest ACP within those clusters. After selecting the suitable cluster, we select the resource on the basis of multiple parameters in this cluster and assign the job. Local update and global update are also performed in HLBA to ensure that we can get the newest status of resources. Local update changes the equation of load for that cluster only and makes an impact on the global updated values. However, it doesn't change the equation of the other clusters [9].

A soon as jobs are allotted to the cluster; jobs are divided on the basis of the multiple parameter on cluster level such as, CPU Utilization, Heap Memory and No. of jobs, etc. And then the jobs are allotted to the worker / computing nodes where job execution is done [10].When a job is completed, the job had to release all resources and a global update will recalculate al parameters again. Overloaded clusters can become available again after a global update.

IV. CONCLUSION

In this paper, we are proposing a hierarchical scheduling algorithm which allocates the job to the cluster level nodes on the basis of ACP (Average Computing Power) with less time complexity. And job allocation from cluster level nodes to the worker / computing nodes is based on the multiple parameters. This is somewhat a hybrid approach that we are proposing in this paper. Such an approach would provide quality scheduling of the jobs with less latency of execution time. Thus our aim of minimizing process execution time would get satisfied. As there are many scheduling algorithms exists and our implementation is on physical real time environment which will give better results. Thus by proposing a hybrid approach we are hereby tried to standardize the scheduling algorithm.

REFERENCES

[1] http://www.wolfram.com/gridmathematica

IEEE Papers:

[2] N. Malarvizhi and V.R.Uthariaraj, Hierarchical Load Balancing Scheme for Computational Intensive Jobs in Grid Computing Environment, *ICAC, IEEE,* 2009

[3] J. Watts, S. Taylor, A Practical Approach to Dynamic Load Balancing,*IEEE transactions on parallel and distributed systems,* 1998, *vol. 9, no. 3, pp. 235-248.*

[4] B.Pradhan, A. Nayak and D.S. Roy, An Elegant Load Balancing Scheme in Grid Computing Using Grid Gain,*International Journal of Computer Science and Its Applications,* 2011, *Vol. 1, Issue 1, pp.254-257.*

[5] Yun-han Lee, Seiven Len and Ruay-Shiung Chang, Improving job scheduling algorithms in a grid environment, *Future Generation computer systems,* 2011, *vol.27, pp.991-998*

[6] Leyli Mohammad Khanil, Shiva Razzaghzadehb and Sadegh vahabzadeh Zargaric, A new step towards load balancing based on competency rank and transitional phases in grid networks, *Future Generation Computer systems,* 2012, *vol.28.pp.682-688.*

[7] E. Ajaltouni, A. Boukerche and M. Zhang, An Efficient Dynamic Load Balancing Scheme for Distributed Simulations on a Grid Infrastructure,*12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications,* 2008, *pp. 61-68.*

[8] A.Touzene, S. Al-Yahai, Hussien, AlMuqbali, A. Bouabdallah and Y. Challal., Performance Evaluation of Load Balancing in Hierarchical Architecture for Grid Computing Service Middleware, *International Journal of Computer Sciences,* 2011, *pp. 1694-1702.*

[9] R. Manimala and P.Suresh, Load Balanced Job Scheduling Approach for Grid Environment, IEEE, 2012.

[10] K.Hemant Kumar Reddy and Diptendu Shina Roy, A Hierarchical Load Balancing Algorithm for Efficient Job Scheduling in a Computational Grid Test-bed,*1st Int'l Conf. on Recent Advances in Information Technology,*RAIT, 2012.

[11] Grid Gain: www.gridgain.com