IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 1, January 2016

A New Approach for Dynamic Load Balancing using Simulation in Grid Computing

Mr. Kapil B. Morey[1], Prof. Sachin B. Jadhav[2]

PG Scholar, Computer Engineering Dept., Padm. Dr. V.B. Kolte College of Engineering, Malkapur, India[1]

Assistant Professor, Computer Engineering Dept., Padm. Dr. V.B. Kolte College of Engineering, Malkapur, India[2]

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 Scheduling and monitoring. Scheduling process directs the job to appropriate resource and monitoring process monitors the resources. The resources which are heavily loaded act as server of task and the resources which are lightly loaded act as receiver of task. Task will be migrated from heavily loaded node to lightly loaded node. 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: Grid, Resources, Simulation, Scheduling.

I. INTRODUCTION

The improvement of performance of computers and their cost reduction is due to development in computing resources. This availability of low cost powerful computers coupled with the popularity of the Internet and high-speed networks has led the computing environment to be mapped from distributed to Grid environments. The recent researches on computing architectures has resulted the emergence of a new computing paradigm known as Grid computing. Grid is a type of distributed system which supports the sharing and coordinated use of geographically distributed and multi owner resources, independently from their physical type and location, in dynamic virtual organizations that share the same goal of solving large-scale applications. In order to fulfil the user expectations in terms of performance and efficiency, the Grid system needs efficient load balancing algorithms for the distribution of tasks. A load balancing algorithm attempts to improve the response time of user's submitted applications by ensuring maximal utilization of available resources. 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, 6].

Resources are dynamic in nature so the load of resources varies with change in configuration of grid so the load balancing of the tasks in a grid environment can significantly influence grid's performance. In grid computing, individual users can retrieve computers and data, transparently, without taking into account the location, operating system, account administration, and other details. Grids functionally bring together globally distributed computers and information systems for creating a universal source of computing power and information. 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 is a technique to enhance resources, utilizing parallelism, exploiting throughput improvisation, and to reduce response time through an appropriate distribution of the application. Load balancing can be implemented using two techniques i.e. static and dynamic. Static load balancing algorithms allocate the tasks 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. The decisions related to load balance are made at compile time [7]. Dynamic load balancing algorithms make changes to the distribution of work among workstations at run-time. They use current or recent load information when making distribution decisions.

II. STRUCTURE OF SIMPLE GRID

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. The machines are usually in one department of an organization and their use as a grid may not require any special policies or security concerns.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 1, January 2016

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.

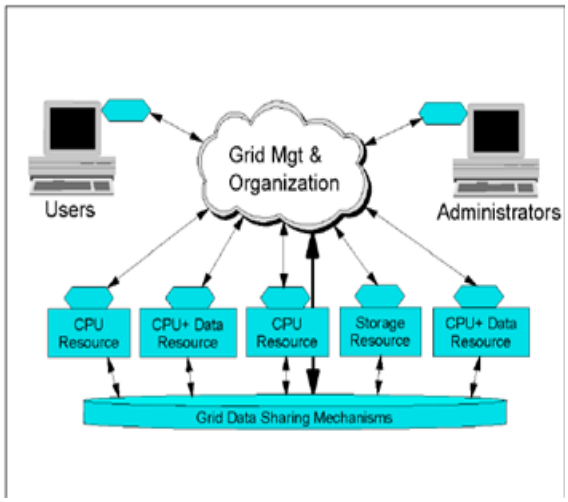

Fig1 A simple grid

When the machines are been include to the heterogeneous systems, more resources are available. The grid system is likely to include some scheduling components. File sharing may still be accomplished using network file systems. Such a grid is referred as an Intragrid [3].

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 last policy i.e. Selection policy defines the tasks that should be migrated from overloaded resources (source) to most idle resources (receiver).

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 [6]. 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, 6]. An unexpected peak can be routed to relatively idle machines in the Grid. And 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. Load balancing should take place when the scheduler schedules the task to all processors. There are some particular activities which change the load configuration in Grid environment. The activities can be categorized as follows:

•Arrival of any new job and queuing of that job to any particular node.
•Scheduler schedules the job to particular processor.
•Reschedule the jobs if load is not balanced
•Allocate the job to processor when it's free.
•Releases the processor after it complete the whole job.

IV. SCHEDULING AND LOAD BALANCING USING GRID SIMULATION

A. Scheduling

The scheduling algorithms do not adequately address congestion, and they do not take fairness considerations into account. Fairness is most essential for scheduling of task. In Fair Scheduling, the tasks are allocated to multiple processors so that the tasks with unsatisfied demand get equal shares of time. Tasks are queued for scheduling according to their fair completion times. The fair completion time of a task is estimated by its fair task rates using a max min fair sharing algorithm. The tasks are assigned to processor by increasing order of fair completion time. In this algorithm, tasks with a higher order 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. For this issue we propose a Load Balance (LB) Algorithm to give uniform load to the resources so that all tasks are fairly allocated to processor based on balanced fair rates. The main objective of this algorithm is to reduce the overall make span [5, 6, 8].

B. Dynamic Load Balancing Algorithm code segment

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

1. Create a set of queues
2. qsize < R/N
3. For each queue qi in Q
4. While there are tasks in the queue do,
5. Assign demand rate of the task Xi
6. k=P/R
7. If Xi <k
8. Assign Xi to ith task as fair rate
9. Else
10. Assign k to ith task as fair rate
11. Calculate fair completion time ti(x)
12. End while
13. End loop
14. Calculate mean waiting time each scheduled task
17. If Zxy >0
18. Each processor which has least capacity is selected for migration
20. End If
21. End While.

C. Grid Simulation

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. Additionally, Grid testbeds are limited, and creating an adequately-sized testbed is expensive and time consuming. Moreover, resources are autonomous and owned by different organizations, it requires the handling of different administration policies at each resource. Due to these reasons, it is easier to use simulation as a means of studying complex scenarios. GridSim is a software platform that enables users to model and simulate the characteristics of Grid resources and networks with different configurations. GridSim is of great value to both students and experienced researchers who want to study Grids, or test new algorithms and strategies in a controlled environment. By using GridSim, they are able to perform repeatable experiments and studies that are not possible in a real dynamic Grid environment.

The GridSim toolkit provides a comprehensive facility for simulation of different classes of heterogeneous resources, users, applications, resource brokers, and schedulers. It can be used to simulate application schedulers for single or multiple administrative domains distributed computing systems such as clusters and Grids. Application schedulers in the Grid environment, called resource brokers, perform resource discovery, selection, and aggregation of a diverse set of distributed resources for an individual user. This means that each user has his or her own private resource broker and hence it can be targeted to optimize for the requirements and objectives of its owner. In contrast, schedulers, managing resources such as clusters in a single administrative domain, have complete control over the policy used for allocation of resources. This means that all users need to submit their jobs to the central scheduler, which can be targeted to perform global optimization such as higher system utilization and overall user satisfaction 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. This is useful for simulating data-intensive jobs over a public network where the network is congested.
•It provides clear and well-defined interfaces for implementing different resource allocation algorithms; and it allows modeling of several regional Grid Information Service (GIS) components.

V. CONCLUSION

This algorithm is able to provide sound results in terms of make span and execution cost. In particular the algorithm allocates the task to the available processors so that all requesting task get equal amount of time that satisfied their demand. Through this proposed algorithm, we have described multiple aspects of load balancing algorithm and introduced numerous concepts which illustrate its broad capabilities. Proposed algorithm is definitely a promising tendency to solve high demanding applications and all kinds of problems. Objective of the grid environment is to achieve high performance computing by optimal usage of geographically distributed and But grid application performance remains a challenge in dynamic grid environment. Resources can be submitted to Grid and can be withdrawn from Grid at any moment.

ACKNOWLEDGMENT

I am highly grateful to **Prof. Sachin B. Jadhav,** for his sincere advice, encouragement and continuous guidance in my work. I warmly acknowledge and express my special thanks for him inspiring discussions and infallible suggestions.

REFERENCES

[1] [Goswami& Das, 2015] Goswami S, Das A, "Deadline stringency based job scheduling in computational grid environment", Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference, ISBN: 978-9-3805-4415-1, pp-531-536, IEEE March 2015.

[2] [Kinhekar& Gupta, 2014] 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, ISSN:2321-7782Vol. 2, Issue 8, pp-102-108, August 2014.

[3] [Goswami& De Sarkar, 2013] Goswami S, De Sarkar A, "A Comparative Study of Load Balancing Algorithms in Computational Grid Environment", Computational Intelligence, Modelling and Simulation (CIMSim), 2013 Fifth International Conference, ISBN: 978-1-4799-2308-3, pp- 99-104, IEEE September 2013.

[4] [Kaur& Singh, 2012] PawandeepKaur, Harshpreet Singh, "Adaptive dynamic load balancing in grid computing an approach," International journal of engineering science & advanced technology , ISSN: 2250–3676 Volume-2, Issue-3, 625 – 632, May-Jun 2012.

[5] [Gogulan& Kavitha,2012] R.Gogulan, A.Kavitha, U.Karthick Kumar, "Max Min fair scheduling algorithm using in grid scheduling with load balancing", International Journal of Research in Computer Science, ISSN 2249-8265 Volume 2 Issue 3 (2012) pp. 41-49.

[6] [Kumar,2011] U. Karthick Kumar "A Dynamic Load Balancing Algorithm in Computational Grid Using Fair Scheduling ", International Journal Of Computer Science and Informatics, Vol. 8, Issue 5, No 1, September 2011 ISSN (Online): 1694-0814.

[7] [Vivekanandan&Ramyachitra, 2011] Dr. K. Vivekanandan, D.Ramyachitra, "A Study on Scheduling in Grid Environment", International Journal on Computer Science and Engineering (IJCSE), ISSN : 0975-3397 Vol. 3 No. 2 Feb 2011.

[8] [Srivastava, 2011] Prabhat Kr. Srivastava, "Improving Performance in Load Balancing Problem on the Grid Computing System", International Journal of Computer Applications (0975 – 8887), Volume 16– No.1, February 2011.

BIOGRAPHY

Mr. Kapil B. Morey PG Scholar, Computer Engineering Dept, Padm. Dr. V.B. Kolte College of Engg.,Malkapur, India.