A New QoS based Load Balancing Approach with Percentage Load Conversion in Grid Heterogeneous System

Smitha Jha
Assistant Professor
Department of Computer Science, Birla Institute of Technology, Mesra, Ranchi
INDIA

Ankit Gupta
Research Scholar
Department of Computer Science, Birla Institute of Technology, Mesra, Ranchi
INDIA

D.K. Mallick, PhD.
Associate Professor
. Department of Computer Science, Birla Institute of Technology, Mesra, Ranchi
INDIA

ABSTRACT

In grid computing, load balancing is a technique to distribute workload evenly across two or more computing nodes, in order to get optimal resource utilization, maximize throughput, minimize response time, and avoid overload. In Grid system, there are queues of jobs waiting for getting resources like storage space, CPU,I/O devices etc. The behaviour or state of the Grid system changes dynamically i.e. from time to time. The bandwidth of the n/w, the no. of jobs, the no. of resources etc. in the system changes dynamically. A new approach with load balancing algorithm with load conversion has been introduced here. This algorithm is applied on different scheduling algorithms using Grid Simulator (Alea 2). With different load conversion percentages in load balancing it has been found that existing scheduling algorithms can performs better if a specified percentage of Load is reallocated depending on the CPU speed of clusters.

General Terms

Grid Scheduling and balancing algorithm.

Keywords

Grid Scheduling, load balancing, load conversion, Grid Algorithm .

1. INTRODUCTION

Grid computing is an emerging distributed computing technology, network infrastructure, information society and is following the tradition known as the Internet, World Wide Web after the third generation of Internet applications [1]. Grid computing is a wide area network connecting large equipment to create a parallel computing environment, to share resources available on the network. Processing power is becomes much larger than the traditional node in distributed computing environment ,termed as computing nodes, each computing resources works according to the system's scheduling policy of the tasks assigned to their scheduler and implementation. Resource management and scheduling is the key grid services, but to achieve efficient grid resource management and scheduling, task scheduling and load balancing is one of the key issues that must be addressed at large. In grid computing, load balancing is a technique to distribute workload evenly across two or more computer nodes links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, maximize throughput, minimize response time, and avoid overload. The load balancing service is usually provided by a dedicated program or hardware device. Load balancing refers to grid computing system, by some scheduling strategy to ensure that the entire resource node computing the ratio of its own performance as an equal, thereby improving the utilization of resources based on nodes, reducing the overall task completion time [2]. To achieve these goals, the load balancing mechanism should be "fair" in distribution of load in the resource node, which indicates the maximum load nodes and the lightest load balance between nodes should be minimized. Load balanced is a collection of servers distributed in a symmetrical way, each server is equally important, providing service to the outside world by its own way without any aids of other servers. Through a certain kind of load shared technology, the outside requests can be allocated to a server with symmetrical structure equally, and then the received server will be responded to the clients independently.

In this paper, First time, in best of our knowledge, a new load balancing algorithm with load conversion based on the CPU speed of the clusters of grid is introduced to effectively address the problem of load balancing in grid computing. Load conversion is the percentage of jobs to be moved from one location to another. The rest of this paper is organized as ollows. Related works is presented in Section 2. In section 3, Experimental setup is explained,In section 4 experimental result is shown , in section 5 algorithm is described.Finally, section 6 concludes this paper.

2. RELATED WORK

Facing the load balancing algorithm is a NP complete problem. It attracted the attention of scholars home and abroad and a focus of grid computing research. One of

the most common applications of load balancing is to provide a single Internet service from multiple servers, sometimes known as a server farm. Commonly, load balanced systems include popular web sites, large Internet Relay Chat networks, high-bandwidth File Transfer Protocol sites, NNTP servers and DNS servers. For Internet services, the load balancer is usually a software program that is listening on the port where external clients connect to access services. The load balancer forwards requests to one of the "backend" servers, which usually replies to the load balancer. This allows the load balancer to reply to the client without the client ever knowing about the internal separation of functions. It also prevents clients from contacting backend servers directly, which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports. In this section, we will introduce some research results at home and abroad. Min-min gives the highest priorities to the task which can be completed earliest [3]. Max-min gives the highest priority to the task with the maximum earliest completion time. The idea behind Max-min is that overlapping long-running tasks with short-running ones. Max-min will execute many short tasks in parallel with the long task [4]. Fast greedy assigns each task, in arbitrary order, to the machine with the minimum completion time for that task [5, 6]. Simulated annealing is an iterative technique based on Monte Carlo random search\ heuristic algorithm. Simulated annealing algorithm is applied to the grid task scheduling, which aims to make the total task execution time at least. Steepest descent algorithm is the iterative point along the negative gradient direction to search, start optimizing speed, close to the minimum point, the optimization of the extremely slow pace. Once the\ existence of local minima, it is generally difficult to break through local minimum, we can only obtain local optimal solution. Genetic algorithm uses crossover and mutation operators on the choice of two samples after the exchange, so that by selecting and breeding the next generation of code to be set [7]. One Kind of Improved Load Balancing Algorithm in Grid Computing [8] overcomes the shortcoming of that genetic algorithm drop into local optima easily. The global search is very capable, it can achieve resource load balancing effectively.

3. EXPERIMENTAL SETUP

Out of many workloads available , we choose “Metacentrum data set ” ,which is generated from the national grid of Czech republic using PBS-Pro consisting of 5000 jobs, 14 clusters and 806 CPUs. Out of 14 clusters , we choose 3 clusters namely cluster 4,cluster 7 and Cluster 10 based on their CPU Speed and Model. To continue the experiment, we manually reallocated load of all the clusters one by one to each of the above mentioned clusters to see and analyze the outcome of this newly reallocation on the overall performance of Grid. We did this procedure thrice for 3 different levels of allocation. First we allocated 5% of workload from all the clusters to Cluster 4 ,Cluster 7 and Cluster 10 one by one to create a new manually manipulate data set then we created 2 more new data set with 10% and 20% allocation. Please keep in mind that this complete reallocation is done on purely random basis.

In a nutshell, the basic concept was to check for the performance of different existing algorithm with some percentage of reallocation of workload.

4. EXPERIMENTAL RESULTS

Two algorithm FCFS and EDF were simulated using Alea 2 simulator and new manipulated Metacentrum data set.

Here graphs are displayed for different parameters of FCFS(FIRST COME FIRST SERVE). In appendix information about the machines and data set is given.

4.1 Using FCFS Algorithm

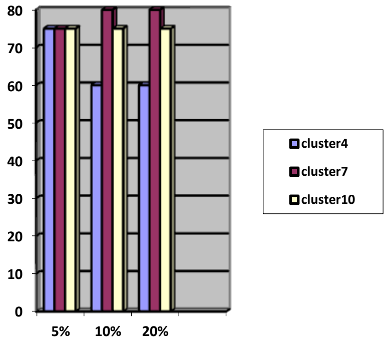

Fig 1:Maximum Machine Usage/Day

Here Graph shows that in most of the cases machine usage is increasing.

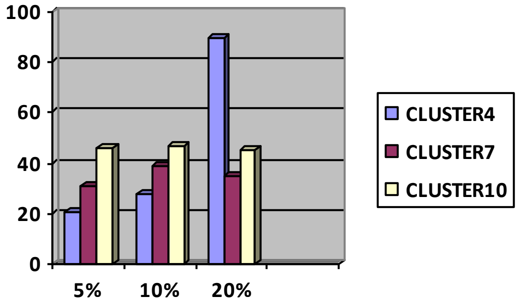

Fig 2: Used Maximum CPU/Day

Here Graph shows that in most cases the CPU usage is increasing.

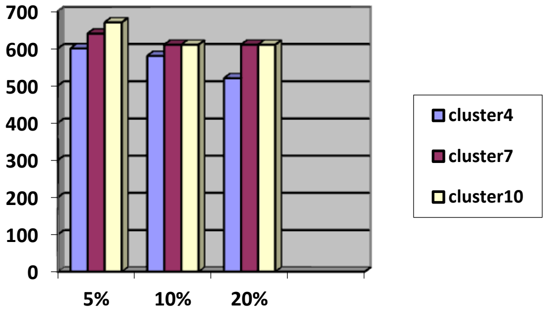

Fig 3:Maximum No. of Waiting jobs/Day

Here Graph shows that maximum no. of waiting time/day for jobs has been decreased.

4.2 Using EDF Algorithm

Here graphs are displayed for different parameters of EDF (Earliest Deadline First). Here cluster 4, cluster 7 and cluster 10 are the clusters where loads are transferred from all other clusters.

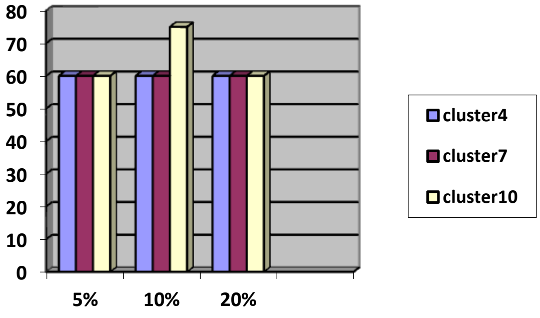

Fig 4:Maximum Machine Usage/Day

Here Graph shows that for 10% load conversion, in cluster 10 machine usage has increased.

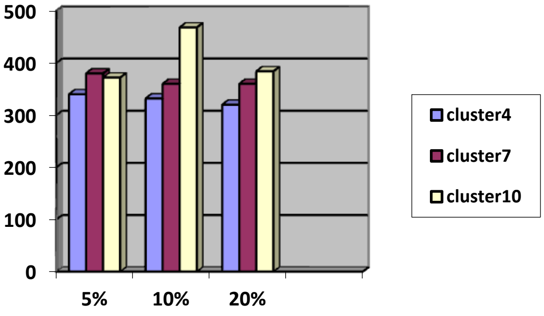

Fig 5: Used Maximum CPU/Day

Here Graph shows that for 10% load conversion, in cluster 10 CPU usage has increased.

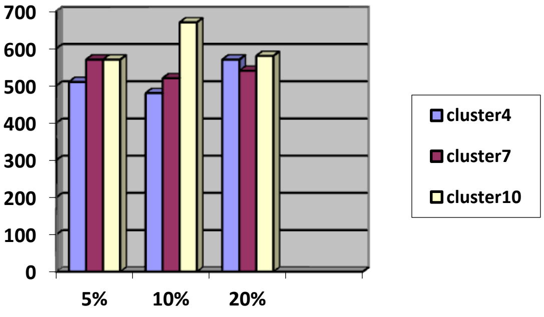

Fig 6: Maximum No. of Waiting jobs/Day

Here Graph shows that for 10% load conversion, In cluster 10 waiting time has increased.

5. LOAD BALANCING ALGORITHM

With the help of above results we can propose a new algorithm which definitely is not a stand alone algorithm but works in addition to existing algorithm.

Step1 For each of the 5%,10% and 20% load conversion do

Step1.1 Choose clusters according to their CPU strength(higher) say m,n,o.

Step 1.2 Assign loads(jobs) from all clusters to these(m,n,o) clusters.

Step 2 Simulator is run and the parameters like Machine Usage/day, waiting/running jobs, requested CPU/day ,used CPU/day, available CPU/day etc. for some scheduling algorithms are measured.

6. CONCLUSION

From the results it is concluded that load balancing with load conversion works well with FCFS algorithm. Machine usage/day and Used CPU/day is increased, Waiting Jobs/day is decreased with this algorithm for the FCFS algorithm. For EDF algorithm for 10% conversion in cluster 10 this algorithm is showing different behaviour. So for FCFS algorithms this proposed algorithm is showing good result, while for other it is not.

7. FUTURE WORK

We tend to extend this algorithm for more such grid scheduling algorithm to find some more correlation between performance and different constraints related to grid environment.

8. APPENDIX

Appendix is at the last of this paper as the data is very huge.

9. ACKNOWLEDGMENT

We are thankful to Dr. R.K. Suri for his valuable suggestions in research work.

10. REFERENCES

[1] Resource Management, Scheduling, and Computational Economy [A].WGCC 2000[C].Japan, March 15-17, 2000.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol.2. Oxford: Clarendon, 1892, pp.68–73.

[2] Vincenzo Di Martino, Marco Mililoti ,"Scheduling in Grid Computing enviroment using genetic algorithm." the 16th Int'1 Parallel and Distributed Processing Symp(IPDPS2002),USA.2002

[3] Vincenzo Di Martino, M Mililotti.,"Sub-optimal scheduling in a grid using genetic algorithm". Parallel Computing,2004,30(5/6):553~565.

[4] R. Wolski, N.T. Spring, J. Hayes, "The network weather service: a distributed resource performance forecasting service for metacomputing, Future Gen." Computer. System. 1999,15 (5) : 757–768.

[5] J. Cao, S.A. Jarvis, S. Saini, D.J. Kerbyson, G.R. Nudd,"ARMS:an agent-based resource management system for grid computing, Scientific Programming" ,2002,10 (2) :135–148.

[6] Ajith Abraham, Rajkumar Buyya. ,"Nature's heuristics for scheduling jobs on computational grids". The 8th

Int'1 Conf on Advanced Computing and Communications (ADCOM 2000),Cochin,India,2000.

[7] Shijue Zheng,Wanneng Shu and Guangdong Chen," A Load Balanced Method Based on Campus Grid", 2005 International Symposium on Communications and Information Technologies (ISCIT 2005). October 12- 14, Beijing.

[8] Hai-yun Peng, Qian Li ,"One Kind of Improved Load Balancing Algorithm in Grid Computing", International Conference on Network Computing and Information Security, May 2011 pp. 347-351

[9] Dalibor Klusáček and Hana Rudová," *Alea 2 - Job Scheduling Simulator*" In proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010), ICST, 2010

11. AUTHOR'S PROFILE

Mrs. Smitha Jha- I am working as Assistant Professor in Deptt. Of Computer Science at Noida Campus of Birla Institute of Technology,Mesra Ranchi. I did my B. Tech.(CS) From BIT,Ranchi and M.Tech(CS) from BIT,Ranchi. I am pursuing PhD in the field of Grid Scheduling

Mr. Ankit Gupta-I am a full time research scholar at Birla Institute of Technology, Mesra, Ranchi(Noida Campus).I am M.Tech.(CS) from BIT, Mesra and B.Tech.(CS) from UPTU,Lucknow. I have more than 8 years of teaching experience in an engineering college in UP .

Dr D. K. Mallick-Working as Associate Professor in Department of Computer Scinece at BIT,Mesra Ranchi. His current research area is Parallel and distributed computing.

APPENDIX

Table1. Machine Details of Metacentrum Data Set

Cluster	Cluster name	CPU Speed	RAM	Processor	OS	Nodes	PEs	Machine ID
0	cluster_0	1500	48000000	Itanium2	linux	1	8	16-23
1	cluster_1	2200	32000000	Opteron	linux	1	16	24-39
2	cluster_2	3200	1009000	Xeon	linux	10	10	42-51
3	cluster_3	2600	131182840	Opteron	linux	5	80	76-155
4	cluster_4	1600	1005000	AthlonMP	linux	16	32	158-189
5	cluster_5	2400	1048576	Xeon	linux	32	64	453-516
6	cluster_6	2659	15565060	Xeon	linux	36	148	517-664
7	cluster_7	3056	2021000	Xeon	linux	35	70	665-734
8	cluster_8	1600	1024000	Opteron	linux	10	20	807-826
9	cluster_9	2400	4000000	Opteron	linux	3	6	827-832
10	cluster_10	2000	4000000	Opteron	linux	23	92	833-924
11	cluster_11	3000	4556800	Xeon	linux	19	152	1023,-1174
12	cluster_12	2660	27343000	Xeon	linux	8	64	1175,-1238
13	cluster_13	2330	15200000	Xeon	linux	11	44	1239,-1282