

Dynamic and Decentralized Load Balancing Approach in Small-Scale Distributed Systems

R.C. Dharmik

Department of Information Technology,
YCCE, Nagpur
raj_dharmik@yahoo.com

Dr. S.R. Sathe

Department of Computer Science & Engineering,
VNIT, Nagpur
srsathe@yahoo.com

Abstract:

To improve the throughput of small-scale distributed systems, workloads have to be evenly distributed among all the computing nodes of distributed systems. As any distributed application, can undergo critical performance issues due to workload imbalances on small scale distributed system and heterogeneity of resources. The main goal to prevent, if possible, the condition where some processors are overloaded with set of jobs while other processors are lightly loaded, moderately loaded, heavily loaded and some may be idle processors. This is due to the characteristics of distributed systems and complex nature of the users job.

In this paper we have proposed real-time prototype model for small-scale distributed systems. Dynamic and decentralized load balancing algorithm has been implemented to evenly distribution of jobs among all the computing nodes of small-scale distributed systems through which to minimize execution time, response time, communication latency and maximize throughput, CPU utilization. It is beneficial for all those researches who want to make more enhance and efficient load balancing techniques for real-time distributed systems in near future.

Keywords: Small-scale distributed systems, Load Balancing, workload, execution time, response time, CPU utilization, Throughput, Communication latency.

I. Introduction

Small-scale distributed Computing Infrastructure that supports the resource sharing and coordinated use of geographically distributed and multi- owner, multi-administrative domain resources independently from their physical type and location in dynamic virtual organizations that share the same goal of solving large-scale applications [12].

Distributed systems are collection of autonomous processing nodes connected by a communication network. Through the communication network, the resources of the system can be shared by users at different locations. However, a fundamental problem arises in making effective use of the

total computing power of a distributed computing system. It is often the case that a certain node has very few tasks to handle at a given time, while another node has many. It is desirable to spread the total workload of the distributed system over all of its computing nodes. This avoids under-utilization of power, and decreases response times for work introduced at more heavily loaded computing nodes. This form of computing power sharing for improving the performance of a distributed system by redistributing the workload among the available nodes is commonly called “load balancing”. The purpose of load balancing is to improve the performance of a system by redistributing the workload among nodes, thus improving the response time and resource utilization. Resource management and load balancing are key distributed systems services, where issues of local balancing represent a common concern for the most distributed systems infrastructure developers [11]. Load Balancing is based on the idea of migration of load from heavily loaded node to the lightly loaded node. The problem starts with to determine when to migrate a load or task [2].

In traditional research, load balancing algorithms can be classified as centralized and decentralized. In the centralized approach, there is only one node making load balancing decisions, and all the information have to go through this node. All the jobs in the system are allocated by this node to the other nodes to be processed. So there may be the single point of failure. In decentralized approach, all nodes involved in the load balancing decisions. Though, it is more robust than the centralized one, it is costly for many nodes to maintain load balancing information of whole system in decentralized approach. Most decentralized approaches have each node obtaining and maintaining only partial information locally to make suboptimal decisions.

II. Types of Load balancing algorithms

Load balancing mainly deals with distributing a set of independent jobs among all the computing nodes of the distributed system such that the jobs are uniformly distributed and none of the nodes are overloaded or under

loaded. There are two methods of performing load balancing – static method and dynamic method [11].

A. Static Load balancing method

Static load balancing methods assume that a priori information about all the characteristics of the jobs, the computing nodes and the communication network is known. Load balancing decisions are made deterministically or probabilistically at compile time and remain constant during runtime. The static method is attractive because of its simplicity and the minimized runtime overhead. However, it has two major disadvantages. Firstly, the workload distribution of many applications cannot be predicted before program execution. Secondly, it assumes that the characteristics of the computing resources and communication network are all known in advance and remain constant. Such an assumption may not apply to a distributed environment.

B. Dynamic Load balancing method

The dynamic load balancing methods attempt to use the runtime state information to make more informative decisions in sharing the system load. Despite the higher runtime complexity, dynamic algorithms can potentially provide better performance than static algorithms. Dynamic load balancing algorithm is preferred for heterogeneous networks where network elements may vary in capacity or number at runtime. Dynamic methods increase the communication overhead due to status exchange protocols. Dynamic load balancing method can be further classified into a centralized approach and a decentralized approach. In the centralized approach only one node in the grid acts as the central controller. It allocates jobs to each of the slave nodes. The slave nodes execute the jobs assigned by the controller. The centralized approach is a simple approach and is beneficial when the communication cost is less significant. It is mainly used for a small size grid. Although the centralized approach is used currently, it limits the scalability of the grid by becoming a bottle neck. Also failure of central controller can cause the entire system to fail.

In the decentralized approach all nodes in the grid are involved in making the load balancing decision. The decentralized algorithms are scalable and have better fault tolerance. The decentralized approach is preferred because elements of the network may vary in capacity or number during run time. Although the decentralized approach is suitable for dynamic heterogeneous resources it increases the communication overhead to a large extent.

III. Related Work

Most of the Load Balancing algorithms in Grid computing environment are implemented using GridSim Simulator.

Neeraj Rathore, Inderveer Chana has presented a hierarchical load balancing technique, which is based on variable threshold value. The attempt has been made to solve the problem of load balancing while maintaining the resource utilization and response time with the help of sender initiated policy. The proposed technique is suitable for dynamic and decentralized grid environment. The author has also done comparison of his result with the results of reference algorithms. The proposed load balancing technique is implemented by using GridSim Simulator 4.0. The author has also proposed migration steps for migrating job from heavily loaded computing node to lightly loaded computing node

B.Priya, Dr. T. Gnanasekaran has proposed Grid architecture for Load balancing-an assessment. The author has intended to propose a scheduling algorithm to improve the performance in an e-governance application for effective scheduling of the various tasks.

N.Malarvizhi,Dr.V.Rhymend Uthariaraj has proposed hierarchical Load Balancing algorithm for Grid Computing Environment and also Compare the results with MELISA, LBA, PIA algorithms

Ruchir Shah, Bhardwaj Veeravalli has proposed Adaptive and Decentralized Load –Balancing Algorithms for Computational Grid Environments.

Azzedine Boukerche et al, has proposed global dynamic load balancing algorithm for HLA-Based Simulation on large scale distributed system. The authors has design and implemented the scheme to support the redistribution of load for HLA- based simulations running on large scale distributed systems.

IV. Proposed Real-time set-up of distributed Systems

The Proposed Real-time Prototype model of distributed systems consisting of seven computing nodes of heterogeneous type. It is a hierarchical structure of computing nodes which is group together with the help of LAN and implemented on Ubuntu 14.04 LTS. LAMP (Linux Operating System, Apache HTTP Web Server, MySQL and PHP) software is installed on all computing nodes of grid computing environment. Fig. 1 shows the hierarchical structure of real-time distributed systems

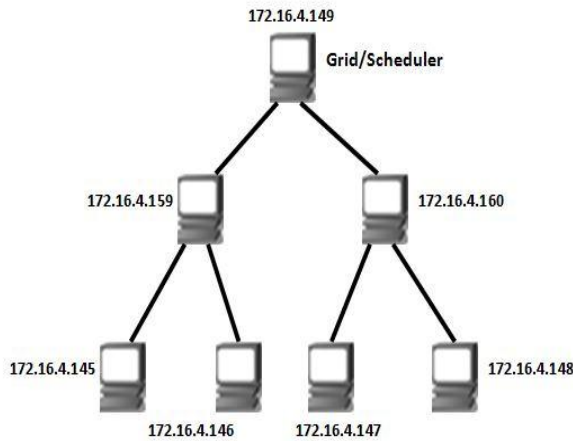


Fig. 1

V. Grid Architecture for Load Balancing

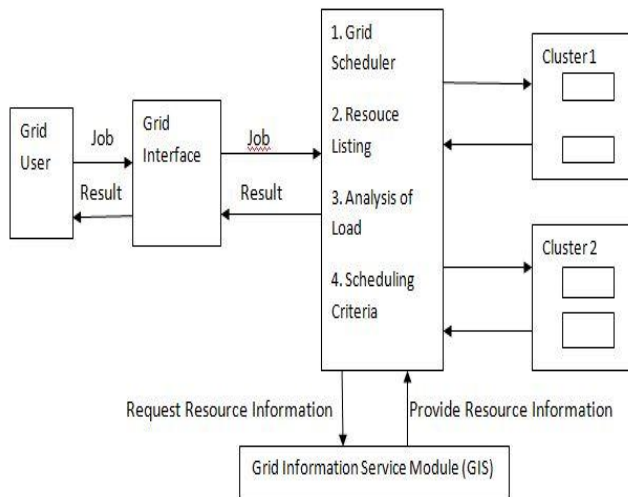
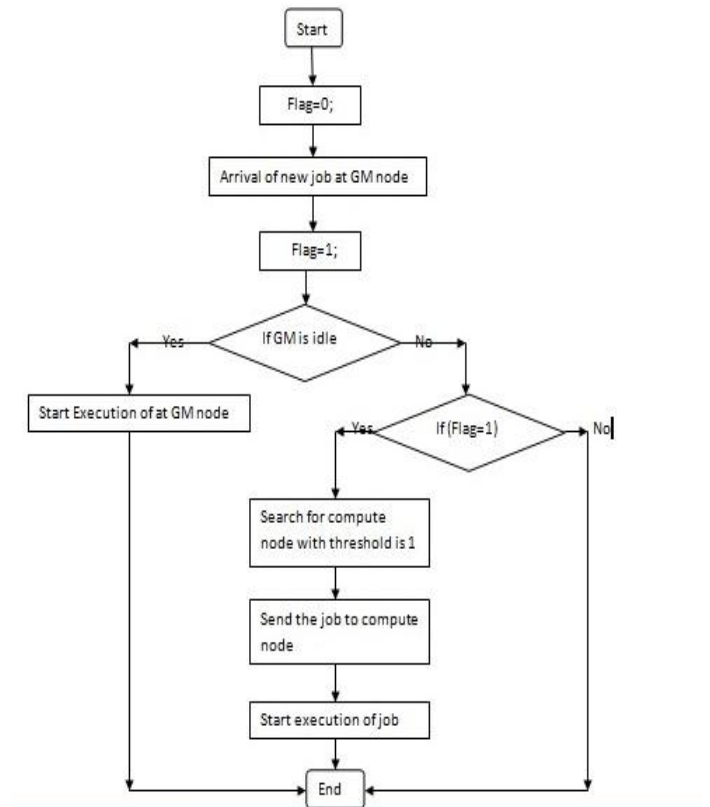


Fig. 2

In Fig. 2 the grid scheduler receives the jobs from the grid users and stored in First-Cum-First-Served (FCFS) manner with hostname, filename and arrival time of the job. The job of a requested grid user executed by the allocated computing node and send the result with finish time of the job to the grid scheduler and grid scheduler send result to the requested grid user. Grid Scheduler is running on the Grid node (Master node), select feasible resources for these jobs according to acquired information from the Grid Information Service Module (GIS). The Grid Information Service module is keep track of IP Address, % CPU Usages of Compute node and status of computing nodes (whether the compute node is idle, heavily loaded or lightly loaded)

and finally job allocated to the repetitive compute node based on the information from GIS[5].

VI. Proposed Dynamic and decentralized Load Balancing Algorithm (DLBA)



GM: Grid Manager node

Experiments and Results

Experiments have been performed to evaluate the performance of dynamic and decentralized load balancing approach for small-scale distributed systems. Such experiments have exposed how the load balancing approaches has detected and reacted to the load imbalances in the distributed systems. We have been performed experiments through which evenly distribution of load on each computing nodes of distributed systems based on CPU utilization of compute nodes.

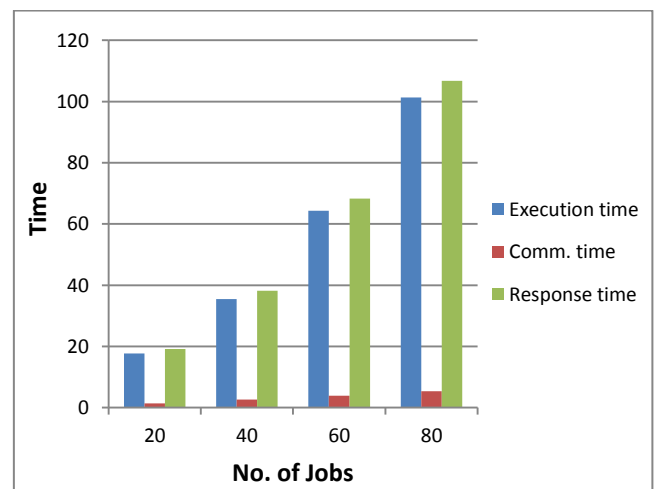
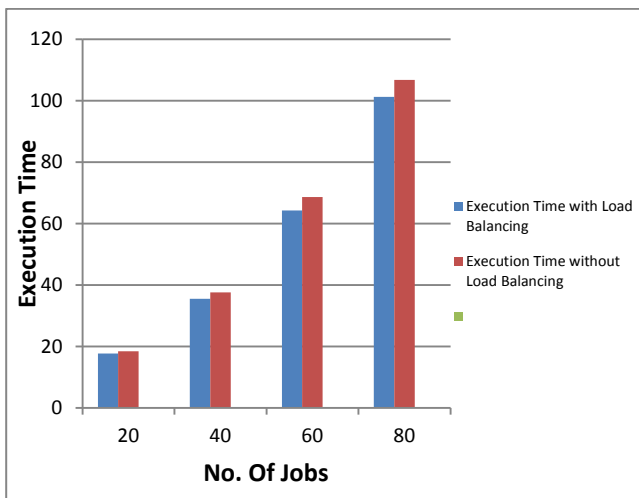
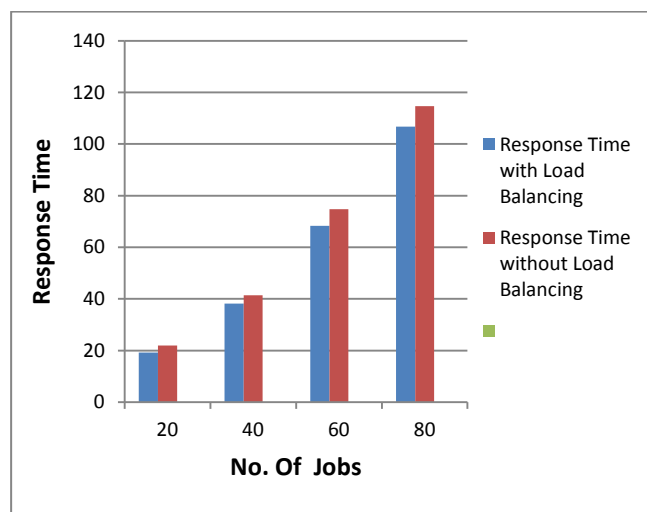
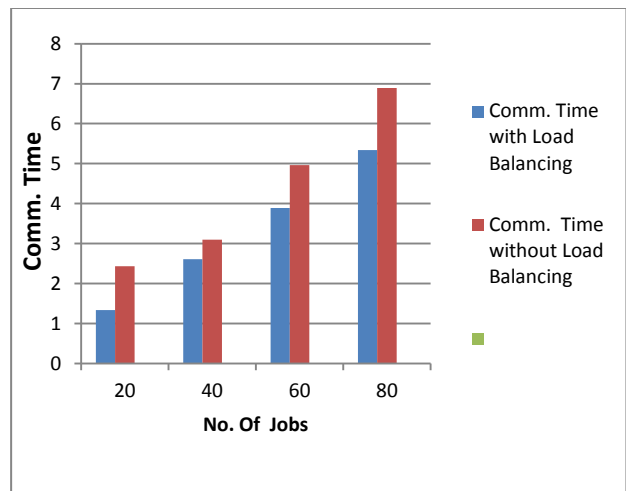
VII. Experimental setup

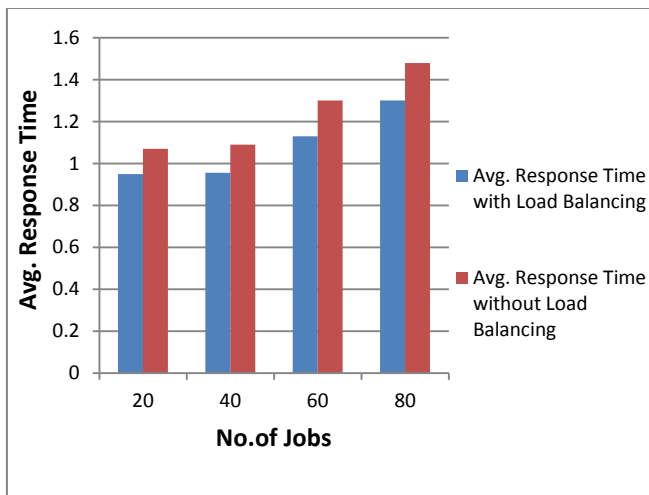
All experiments have been performed on small-scale distributed systems composed of 7 nodes. Two Dell clusters of three nodes each connected with HP Z1 G2 Workstation of 3.3 GHz, 16 GB DDR3 RAM, 1TB HDD with Quad Core Xeon Processor act as master node of distributed systems.

Each compute node of Dell clusters with Quad-Core Processor, 3.1 GHz, 4 GB DDR3 RAM, and one compute node with Dual-Core Processor with 2 GB DDR RAM. All the compute nodes are inter-connected through high speed network of ethernet cable that allows data transmission upto 1 gigabits per second. Moreover, Ubuntu 14.04 LTS Operating systems were installed in all the compute nodes, LAMP (Linux, Apache HTTP Web Server, PHP) were installed on each compute node for Processing of user jobs, Each Compute node is used as FTP Server by installing proftpd on all the compute node for uploading user jobs to other compute node also install FILEZILLA for establishing connectivity among all the compute nodes of distributed systems. We were installed CURL (Client Uniform Resource Locator) for executing user jobs on compute node browser. We have provision of addition of new resources or deletion of existing resources by maintaining scalability and heterogeneity of compute resources.

VIII. Implementation

The experimentation has been performed on real time distributed system. We also considered resource heterogeneity. Following figures shows the result of proposed load balancing approach and compare with without load balancing algorithm.





IX. Conclusion

In this paper different load balancing algorithms and load balancing policies have been discussed. The prototype model of small-scale distributed system and Dynamic Load Balancing approach has been proposed. The real-time set up of distributed systems is established with the help of installation of different softwares like LAMP, FTP SERVER, FILEZILLA, and CURL for the synchronization of different computing nodes, uploading of jobs and execution of user jobs. The proposed dynamic and decentralized load balancing approach is effectively implemented on real-time small-scale distributed system and we have been minimize the execution time, response time, communication latency and maximize throughput, resource utilization. The further work is to execute more number of users job to minimize average response time of a real-time distributed system and proposed real-time distributed systems is used as grid computing environment.

X. References

1. Neeraj Rathore, Inderveer Chana, “A sender Initiated Based Hierarchical Load Balancing Technique for Grid Using Variable Threshold Value”, 978-1-4673-6190-3/13, 2013IEEE
2. Neeraj Rathore, Inderveer Chana, “ Load Balancing and Job Migration Techniques in Grid: A Survey of Recent Trends”, Springer Science+ Business Media New York 2014, DOI 10.1007/s11277-014-1975-9
3. Neeraj Rathore, Inderveer Chana,” Variable threshold-based hierarchical load balancing technique in Grid”, Springer-Verlag London 2014, DOI 10-1007/s00366-014-0364-z

4. Neeraj Rathore, Inderveer Chana, “A cognitive Analysis of Load Balancing and job migration Technique in Grid”, 2011 World Congress on Information and Communication Technology, 978-1-4673-0126-8/11, 2011 IEEE
5. B.Priya, Dr. T. Gnanasekaran, “Grid Architecture for Scheduling and Load Balancing-An Assessment”, ISBN No.978-1-4799-3834-6/2014, IEEE
6. Hemant Kumar Mehta, Manohar Chandwani, Priyesh Kanungo, “ A Modified Delay Strategy for Dynamic Load Balancing in Cluster and Grid Environment”, 2010 IEEE
7. Kai Lu Riky Subrata, Albert Y. Zomaya, “ An Efficient Load Balancing Algorithm for Heterogeneous Grid Systems Considering Desirability of Grid Sites”, 2006 IEEE
8. Liang Guangmin, “ Adaptive Load Balancing Algorithm over Heterogeneous Workstation”, 2008 IEEE
9. Belabbas Yagoubi and Meriem Medebber, “ A Load Balancing Model for Grid Environment”, 2007 IEEE
10. Sunita Bansal, Chittaranjan Hota,” Efficient Algorithm on Heterogeneous Computing System”, 2011 International Conference on Recent Trends in Information System, 2011 IEEE
11. N. Malarvizhi, Dr. V. Rhymend Uthariaraj “Hierarchical Load Balancing Scheme for Computational Intensive Jobs in Grid Computing Environment”. 978-1-4244-4787-9/09/\$25.00, 2009 IEEE, ICAC 2009.
12. Ruchir Shah, Bhardwaj Veeravalli, Senior Member, IEEE, and Manoj Misra, Membe, IEEE “On the Design of Adaptive and Decentralized Load –Balancing Algorithms with Load Estimation for Computational Grid Environments”. IEEE Transaction on Parallel and Distributed Systems, Vol.18, No. 12, December 2007, Digital Object Identifier no. 10.1109/TPDS.2007.1115.