IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 4 Issue 6, June 2017
ISSN (Online) 2348 – 7968 | Impact Factor (2016) – 5.264
www.ijiset.com

A Comparative Study of Load Balancing Techniques in Distributed Systems

Vinita Mathur

Department of Computer Science, Jodhpur Institute of Engineering and Technology

Jodhpur, Rajasthan, India

Abstract— Need of high speed processing has brought us into an era where we cannot fulfil our processing demands with just a single computation unit. Therefore, we moved towards distributed systems, where we have multiple processing devices connected in a network. The capabilities of these systems provide us tremendous amount of processing power which can handle some of the most complex computations of the world. This is not as simple as it sounds. The distributed systems are prone to some problems such as improper use of resources, bandwidth, computation time, power wastage. This leads to an immediate urge to address these problems because, we want to maximize user satisfaction, reduce response time, enhance resource utilization along with increment in job scheduling speed and performance ratio.

In order to not let these problems, occur we use load balancing. this paper shows, the comparative study of various load balancing techniques available for distributed system. Parameters used for comparison are: throughput, associated overhead, fault tolerance, migration time, response time, resource utilization, scalability and performance.

Keywords— Distributed Systems; Load Balancing; Performance Parameters

I. INTRODUCTION TO LOAD BALANCING

Load balancing is the process of dividing the workload evenly among processors/nodes. To improve performance of system, different kind of load balancing techniques are used to distribute or redistribute workload among nodes. Goal of load balancing in distributed system is to distribute workload among node to maximize throughput, optimize resource use, minimize response time and avoid overload of any single node.

II. NEED OF LOAD BALANCING

There are a lot of nodes available in distributed system which are working independently and linked with each other using some communication channel. Each node may have a different processing capacity. The workload has to be effectively distributed among all nodes of distributed system on the basis of their processing capability in such a way that all process can complete in minimum possible time. With this we have to reduce idle time of any resource. For all this we require load balancing.

III. TYPES OF LOAD BALANCING TECHNIQUES

1. Static Load Balancing – It is a master slave configuration based load balancing technique. Master node analyses the workload and state of all the slaves. On the basis of these it decides how workload should be distributed among the slaves. Master node keeps record of workload assigned to every slave. slave nodes process the assigned workload individually according to their capabilities and returns the results to master node. The master node now compiles the results received from all the slave nodes and generates a final result which is return to user as output. once the load is assigned to a slave, it cannot be transfer to another slave during run time. Therefore, such load balancing techniques which do not allow transfer of workload during run time are known as static load balancing techniques[1][2][4].

2. Dynamic Load Balancing – This technique considers current state and presently available load of all nodes and determines how much load should be handled by each node. On the basis of this, it migrates load from more loaded nodes to less loaded nodes in such a way that load on every node becomes almost equivalent in terms of resource utilization. Advantage of this technique is that, failure of any node will affect only performance of distributed system unlike static load balancing technique where the distributed system completely fails when any node fails[2][5].

IV. STATIC LOAD BALANCING TECHNIQUES

1. Random Allocation – This technique uses the

method of random number generation for processor selection. the tasks are then allotted to these randomly selected processors. thus, the probability of task allocation to each processor becomes 1/N, where N is the number of processors. This means that in a distributed system the processing devices are randomly selected for task allotment. Major advantage of this technique is, it requires less inter process communication because every node maintains its own load record. But problem with this technique is that there may be a situation in which some nodes gets overloaded while the others remains under loaded[1].

2. Round Robin – This technique distribute load evenly among all the nodes in cyclic fashion without any priority i.e. load is assigned to nodes in series and will back to first node if last node has been reached. Load index is maintained by each node independent of allocation from remote node. There is no requirement of inter process communication thus this technique gives tremendous performance for special purpose application[2][3].

3. Central Manager – In central manager technique, central processor gathers all slave processors' load information and then select a slave processor that have least load to assign a job. a message is sent by the slave nodes to central manager, if any change occurs in load. this technique requires high level of inter process communication which may cause the bottleneck state[7].

4. Threshold Based Allocation – In this technique load is characterized in three classes: under load, medium load and overload. Two important parameters tlower and tupper are used to describe these three terms:

Under load - load<tlower

Medium load - tlower <= load<=tupper

Over load - load>tupper.

Here load is assigned immediately after the creation of a node. Nodes for new processes are selected locally without sending remote messages. Every node keeps a private copy of system's load. Initially all the nodes are considered under loaded. As a node gets overloaded then that node sends its new load state to all other nodes and regularly updating them about its load state. This technique requires large number of process allocation and low inter process communication[10].

V. DYNAMIC LOAD BALANCING TECHNIQUES

1. Central Queue – This technique uses a queue and a queue manager for load balancing. In this technique, every incoming activity (i.e. load) is added into a queue. This queue is maintained in master node. Each slave node of distributed system maintains a threshold value that specifies whether the resource is over loaded or underloaded. When any slave node gets under loaded, then it requests the master node to send load to it. The master node then removes the load from the queue in FIFO fashion and assigns it to the requesting slave node. If there is no activity in the queue, then the request sent by slave node is buffered by the master node. When the load arrives the slave, nodes are assigned the load according to the incoming order of requests[1].

2. Least Loaded First – In this technique, the master node keeps a track of activity queue of slave nodes. Whenever the master node encounters incoming load on the distributed system, it assigns the load to that slave node which has the smallest activity queue[1].

3. Cycle Stealing with Central Queue – This technique uses two queues for load balancing. One queue contains the list of short jobs and another queue contains list of long jobs. Short server is responsible to handle short jobs and long server is responsible to handle long jobs. If the long server is idle then it can process short jobs. Similarly, if the short server is idle, it can process long job[9].

4. Ant Colony Optimization – This load balancing technique concept is influenced by ant's behaviour. A colony of ants collectively performs useful tasks like find the shortest path between nest and food source and sharing this information with other ants by lay some chemical substances i.e. Pheromone. The ants are attracted by Pheromone. So, all ants follow the path which have high probability of Pheromone on ground for finding food and return to nest. This collective intelligence of ants is transformed into useful load balancing technique. Ant first traces overloaded nodes then go back to fill the under loaded nodes, so that load can be balanced. This technique is used to find optimal resource allocation for each task in distributed system. This technique tries to minimize response time[4].

5. Honey bee optimization – The basic idea of this technique is to simulate the foraging behaviour of Honey bee colonies. Special kind of bee in hive, scout bees will search for new food source when there is shortage of food in their hive. After finding the source of food it will come back to its hive and perform vibration or waggle dance. After seeing this waggle dance other bees can easily determine the amount and distance of food. The other kind of bee called forager bee will follow the scout bee to reach at that new food source. This procedure will repeat until the food in that new source gets reduce. Same concept followed by this load balancing technique in which the algorithm checks periodically load on each virtual machine. Then it identifies the highly loaded and less loaded nodes. After that it transfers load from highly loaded nodes to less loaded nodes so that the loads are evenly distributed. The workload to be done can be considered as bee. It searches for a less loaded node (food source), when bee finds suitable node then assignment of workload takes place. Next coming workload is also assigned to same node. This assignment process will continue until this less loaded node reaches to threshold value. When the threshold value is reached, the algorithm again starts searching less loaded node and same procedure is repeated with this new less loaded node[6].

6. Practical Swarm Optimization – This technique first checks status of each node before assigning task to node. This technique assigns the task to the nodes in best fit manner so that memory wastage should be minimum. This technique has a higher throughput[2].

7. Equally Spread Current Execution – This technique tackles the process according to their priorities. It determines the priority according to the size of process. Initially it distributes the workload to nodes randomly according to priority, then it transfers the load to node which is lightly loaded. Load balancer distributes the load on the different nodes. This load balancer maintains the queue of the process that are currently using or waiting for nodes. The load balancer then continuously checks this queue as well as list of nodes. If there is a node available that can execute the process in queue, then load balancer assign the process to that node. If there is a node, node1 that is free and another node, node2 that needs to be freed from some processes, then load balancer assign some process of node2 to node1 to reduce overhead of node2. The load balancer selects the process that matches the criteria for execution at the present time. Although this technique provides better results but this also requires a lot of computational overhead[2].

8. Semi Distributed Load Balancing – In this technique, the nodes of a distributed system are divided into various clusters. A central node is appointed in each cluster which takes care of load balancing in that cluster[8].

9. Centralized Load Balancing – In this technique, there is only a single node responsible for the management of load of the whole system. The remaining nodes of the systems interacts with the central node[2].

VI. PERFORMANCE PARAMETERS

1. Throughput: Throughput is the amount of data transferred successfully from one node to another node in a specific time period.

2. Associated Overhead: Overhead involves amount of additional work which is required to properly implement the load balancing technique like migration of task, inter process communication, movement of data etc.

3. Fault Tolerance: The load balancing technique should work properly in case of any failure condition. The performance of the technique depends upon amount of failure. A small failure can result in large performance degradation.

4. Migration Time: Migration time is the time required to move a process from one node to another node. Sometimes it may require to redistribute load when overload condition occurs at a node or due to some other problem. Migration time should be low to maintain the system performance.

5. Response Time: Response time is amount of time is required to complete a task by distributed system while implementing load balancing technique. This time should be minimum to enhance the performance.

6. Resource Utilization: It means how the available resources can be used in proper and optimized way. This is required to get good performance.

7. Scalability: It defines the ability to handle gracefully a growing amount of workload.it should be high for better performance.

Table 1: Results for Static Load Balancing Techniques

	THROUGHPUT	ASSOCIATED OVERHEAD	FAULT TOLERENCE	MIGRATION	RESPONSE TIME	RESOURCE UTILIZATION	SCALABILITY
RANDOM ALLOCATION	LOW	LOW	NO	NO	LESS	LESS	LESS
ROUND ROBIN	LOW	LOW	NO	NO	LESS	LESS	HIGH
CENTRAL MANAGER	LOW	HIGH	YES	NO	LESS	LESS	LOW
THRESHOLD BASED ALLOCATION	LOW	LOW	NO	NO	LESS	LESS	HIGH

Table 2: Results for Dynamic Load Balancing Techniques

	THROUGHPUT	ASSOCIATED OVERHEAD	FAULT TOLERENCE	MIGRATION	RESPONSE TIME	RESOURCE UTILIZATION	SCALABILITY
CENTRAL QUEUE ALGORITHM	HIGH	HIGH	YES	YES	MORE	LESS	LOW
LEAST LOADED FIRST	HIGH	HIGH	YES	YES	MORE	LESS	LOW
CYCLE STEALING WITH CENTRAL QUEUE	HIGH	HIGH	YES	YES	MORE	LESS	HIGH
ANT COLONY OPTIMISATION	LOW	HIGH	YES	YES	LESS	MORE	HIGH
HONEY BEE OPTIMISATION	LOW	HIGH	YES	YES	LESS	MORE	HIGH
PRACTICAL SWARM OPTIMISATION	HIGH	LOW	YES	YES	LESS	LESS	LOW
EQUALLY SPREAD CURRENT EXECUTION	HIGH	HIGH	YES	YES	LESS	MORE	HIGH
SEMI DISTRIBUTED LOAD BALANCING	LOW	HIGH	YES	YES	MORE	LESS	LOW
CENTRALISED LOAD BALANCING	LOW	HIGH	YES	YES	MORE	LESS	LOW

VII. CONCLUSION

load balancing is used to distribute workload evenly on multiple nodes. There are different techniques to implement this task. In this paper, we compare different kind of load balancing techniques on basis of some parameters. This comparison shows although static load balancing techniques are more stable and faster but dynamic load balancing techniques are always better than static in terms of throughput, associated overhead, fault tolerance, migration, response time, scalability. Dynamic load balancing technique provides better performance than static load balancing technique in distributed system. Each technique has its own advantage and disadvantage and there is no absolutely perfect balancing algorithm exists but we can use depending on the need. Still after analysing these techniques we can conclude that random load balancing technique is naive technique while cyclic stealing with central queue provides best performance, high efficiency and throughput.

VIII. FUTURE SCOPE

Load balancing techniques presented in this paper are promising, but still there is scope for future research. We can use mobile agent technology for load balancing in distributed system. Mobile agents are autonomous programs that can migrate through the nodes of the distributed system to accomplish some tasks on behalf of some user. With adaptive learning and automation added to agents, the agent can be used to develop a new load balancing and scheduling technique.

IX. REFERENCES

[1] S.Saranya, Dr.R.Manicka Chezian "Comparative Study on Load Balancing Techniques in Cloud Computing", International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 10, October 2016.

[2] Deepali Mate, Sonali Ghumatkar, Priyanka Bharat "A Survey on Load Balancing Strategies in Cloud Computing", International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 10, December 2014.

[3] Nusrat Pasha, Dr. Ravi Rastogi, "Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014.

[4] Ayushi Sharma, Umang Singh "Study on Load Balancing Techniques in Ant Colony Optimization for Cloud Computing", International Journal of Computer Applications.

[5] Dr.S.Suguna, R.Barani "Simulation of Dynamic Load Balancing Algorithms", Bonfring International Journal of Software Engineering and Soft Computing, Vol. 5, No.1, July 2015.

[6] Anju Baby, Jeno Lovesum "A Survey on Honey Bee Inspired Load Balancing of tasks in Cloud Computing", International Journal of Engineering Research & Technology, Vol. 2 Issue 12, December 2013.

[7] Deepika, Divya Wadhwa, Nitin Kumar "Performance Analysis of Load Balancing Algorithms in Distributed System" Research India Publications, Volume 4, Number 1 (2014), pp. 59-66.

[8] Ahmad, Ishfaq and Ghafoor, Arif, "Semi-Distributed Load Balancing for Massively Parallel Multicomputer Systems" (1991). Electrical Engineering and Computer Science Technical Reports. Paper 103.

[9] Mor Harchol-Balter, Cuihong Li, Takayuki Osogami, Alan Scheller-Wolf, Mark S. Squillante "Analysis of Task Assignment with Cycle Stealing under Central Queue", IEEE Distributed Computing Systems, 2003.

[10] Monika Kushwaha, Saurabh Gupta "Various Schemes of Load Balancing in Distributed Systems- A Review" International Journal of Scientific Research Engineering & Technology, Volume 4, Issue 7, July 2015.

[11] Ali M.Alakeel "Guide to Dynamic Load Balancing in Distributed Computer Systems" International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

X. AUTHOR PROFILE

Vinita Mathur completed her B.Tech. from Vyas Institute of Engineering & Technology, Jodhpur (Rajasthan) in year 2013. She is currently pursuing M.Tech. in Computer Science from Jodhpur Institute of Engineering and Technology, Jodhpur (Rajasthan). Her areas of interests include Distributed Systems, Web Development and Networking.