# A Survey Paper on Comparative Study of Load Balancing Techniques in Distributed System

Shiva Bahadur Pathak
*Dept. of Computer Science and Engineering*
*Brac University*
Dhaka, Bangladesh
shivapathak67@gmail.com

Suvro Debnath
*Dept. of Computer Science and Engineering*
*Brac University*
Dhaka, Bangladesh
suvrodnath@gmail.com

Md Sabbir Hossain
*Dept. of Computer Science and Engineering*
*Brac University*
Dhaka, Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

Annajiat Alim Rasel
*Dept. of Computer Science and Engineering*
*Brac University*
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—Processes need more resources to fulfill their duties as technology progresses, which a supercomputer may not be able to manage. One solution building a distributed system in which a collection of nodes in various places are linked together and resources are shared among them. This allows for easy expansion by adding more nodes, which eliminates resource constraints. However, today, one of the most important difficulties is load imbalance, which can cause the system's performance to degrade, resulting in increasing delays in our work. Load balancing is a strategy for boosting scalability and overall system performance in distributed systems by spreading work evenly across the participating computers/nodes. Load balancing algorithms come in a number of varieties, and their effectiveness is determined by factors including response time, resource consumption, and fault tolerance. Static and dynamic load balancing algorithms are the most common types of load balancing algorithms. A few hybrid algorithms are also mentioned, which try to maintain the simplicity of static techniques while simultaneously being adaptable, as dynamic methods are. The study of performance analysis of such load balancing algorithms is presented in this review paper, with the intention that this research may aid in the development of new algorithms.**

***Index Terms*—Distributed Systems, Load Balancing, Static Load Balancing, Dynamic Load Balancing**

## I. Introduction

A Distributed system where nodes are physically separated but linked together using the communication network to form large Distributed computing system. Distributed system uses more than one processor to executed the program.

In these systems we have limited resources, resources refer to memory, hard disk, and processor speed. Since it has limited resources, it can handle limited number of users or client and their request. if the user or client increases in the system or server. it can't handle properly and an imbalance load distribution may lead to the slow performance, hang and crash issues. To solve this issue, we need to add more nodes or computers with more resources. It can be done with horizontal and vertical scaling. Now, the amount of processing time needed to execute all processes assigned to a processor is called workload of a processor [1]. One of the main challenges is to distribute the load or process among this processors or server or nodes. The main purpose of distributing task is to increase availability, improve throughput, reliability, maintain stability, optimize resources utilization, minimize communication delays and provides fault tolerant capability.

In order to solve these problems, occur we use load balancing. Load balancing is very essential in distributed system to distributing task or loads among nodes or computers in order to improve the performance of the whole systems and also minimize the cost. To minimize bottlenecks and to fully utilize available resources, incoming request demands are appropriately distributed among available system resources [2].“Figure. 1” shows a simple load balancing diagram. There is typically two types of load balancing, Hardware and Software Load Balancing. Comparing these two mostly used is software-based load balancing. There are variety open-source load balancers and load balancing software available for different O/S and application.

Load balancing methods classified in many ways. The centralized method is used when the load balancing system is controlled by one node and only that node performs the balance. However, a load balancing system that is handled by numerous nodes and those nodes do the balancing is referred to as distributed load balancing (Decentralized) [3]. There are mostly two load balancing algorithm or techniques which are broadly classified into two types - ”Static Load Balancing” and ”Dynamic Load Balancing”. The purpose of load bal-

ancing techniques, either static or dynamic, is to improve the performance by redistributing the workload among available server nodes [4]. Load distribution in static load balancing is dependent on the load at the time of node selection or let's say initial states of process, but load distribution in dynamic load balancing is done at run time based on current state of system.

The paper will describe the various Static and Dynamic algorithms and their processes and later on provide a comparison among them as follows: Section II and III will be regarding Static and Dynamic Load Balancing algorithms respectively. Section IV will cover the parameters upon which we shall conduct our comparison. Section V will be the comparison section amongst all the mentioned algorithms and lastly, we will conclude in section VI.

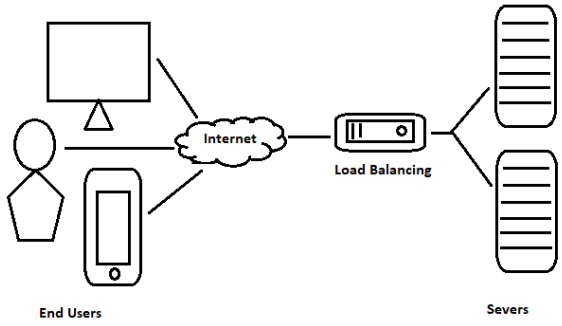


Fig. 1. General Load Balancer.

## II. Static Algorithms

When talking about static load balancing algorithm, it is called a static when during load distribution, the algorithm does not take the current state of the system into account. Often, they distribute workloads randomly or orderly, depending on the algorithm used. Since, static algorithms do not need to retain any information regarding system state so they have low communication overhead [5]. Because of this, their primary disadvantage is that they cannot adapt with the current state of the system and thus, may overload one node while some other node will be under underloaded, thus making an inefficient system. Here we shall describe 4 such algorithms, which falls under the definition of static

### A. Randomized Algorithm

This is classic algorithm where its procedure is to randomly select a node to distribute load. This is done without any consideration to current or previous workload. This is suited to special applications where load is predefined. There is no inter process communication at all since nodes do not need to share its load information although a record is maintained with anyone [6].

### B. Round Robin Algorithm

This is an algorithm where nodes receive easily and in a circular order without priority. This algorithm is simple to implement and does not require any inter process communication [7].

### C. Central Manager Algorithm

This algorithm works in a slave master system, where master node decides where load will be transferred. Upon any load changes, slave node notifies master node and this information is updated in load index and load is delivered accordingly. This requires high inter process communication which may lead to bottleneck state [8].

### D. Threshold Algorithm

Load for a node is decided upon the time of its creation and nodes upon which load will be distributed is chosen locally without any remote messages being sent. Underload, Medium and Overload are the 3 levels of load, and "Tunder and "Tupper" are the two parameters used to describe the load level.

- $Tunder > Underloaded - Load$
- $Tupper \geq Load \geq Medium - Tunder$
- $Tupper > Overloaded–Load$

All noders are considered underloaded initially, and when load level exceeds, message is sent to notify new load state to all nodes. A remote node is chosen only if a local node is overloaded otherwise load is distributed locally. Thus, low levels of inter process communication is applicable for this algorithm [9].

## III. Dynamic Load Balancing Algorithm

The essence of these algorithms is to distribute work upon the current workload on the nodes. there are three controlling strategies. Centralized distribution where one node is at the center and distributes load across all nodes. Distributed load distribution where load distribution is distributed evenly among all nodes. In Semi distributed load distribution, nodes are formed into clusters and load distribution of the overall system is done which mutual communication among all the clusters.There are different policies for dynamic load balancing algorithms which can define their implementation [10]:

- Information policy: This policy is responsible for collection information of node state and sharing it among other nodes.
- Location policy: It determines the best node for from available nodes according to the sender request.
- Selection policy: This policy determines most idle nodes for particular or selected job, basically exchange workload for particular process.

There are many algorithms that have been developed for the high performance of a system. "Figure. 2" shows a basic dynamic load balancing technique for redistributing workloads from highly loaded nodes to lightly loaded nodes [11].some of them are discuss below which falls under the definition of dynamic load balancing distribution:

### A. Central Queue Algorithm

Here, there is a central host, which maintains a cyclic FIFO queue where new activity and requests are stored. This works by the principal of receiver initiation, where node requests for activity to the central node when its processers is underworked. New activity is stored in queue until aby of the receiver node asks for an activity. This method requires high inter process communication among the nodes [9].

### B. Local Queue Algorithm

This algorithm implements the idea of local allocation of records of processing data and workload. New processes are received by the main host, the information is passed around the local load manager. The local load manager attempts to collect processes from remote hosts only if the local host is underloaded. When another load manager gets such message, it compares it with the number of processes it has, and the number of requested one. If the requested is greater than some of its processes are transferred and a confirmation message is sent detailing the number of processes transferred. This algorithm requires much less inter process communication than that of central queue algorithm [8].

### C. Least Connection Algorithm

The algorithm requires a load balancer. Its job is to keep a log on the number of connections on the nodes and higher the number of connections, the more loaded the node is considered to be in which decreases again when connection terminates. Hence when a question of load distribution is considered, the node with least connections present is considered as priority [12].

### D. Ant Colonay Optimization

This Algorithm is influenced by ant behavior. They collect information shortest path to find food from their nest and they also send information to each other by some releasing some kind of chemical. It can useful to find information of overloaded nodes and balance the nodes [13].

### E. Honey bee Optimization

This algorithm is based on foraging behavior. Where algorithm checks periodically load on each virtual machine. And find out highly loaded nodes and less loaded nodes transfer load to less node, just like scout bee searches for food [14]. Where food is a workload.

### F. Practical Swarm Optimization

This algorithm or techniques first check the status of each node or computer before assigning any task to that node. It assigns task such in best fit manner so that memory wastage should be minimum [15].

## IV. Comparative Qualitative Parameters

Based on the requirement of the modern load balancing era, the below parameters are set for comparison amongst the algorithms stated as mentioned as follows [16] [17]:

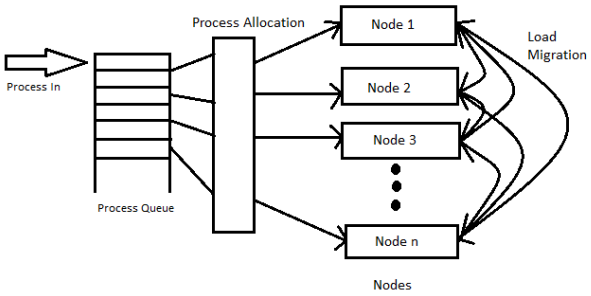


Fig. 2. Job Migration in Dynamic Load Balancing Strategy (source: Jain P. and Gupta D., 2009.).

### A. Reliability

Any system should be reliable so whenever any part of the system fails or node still the system should work. Dynamic is more reliable load balancing algorithm than static load balancing algorithm.

### B. Overhead

Defines the work done during processes like inter process communication etc. We would want it to be as low as possible for efficient performance.

### C. Throughput

Describes completion rate of tasks. The higher the throughput, better the algorithm.

### D. Process Migration

This implies the time required for migration of jobs to nodes. We would want it to be low for any algorithm.

### E. Response Time

The raw time taken for a job to be completed is response time. The value should be as minimum as possible.

### F. Resource Utilization

This depicts the level of optimization level of the algorithm in terms of usage of available resources.

### G. Fault Toleration

This shows how robust the algorithm is. This parameter is to judge if the algorithm survives failure, and if it degrades the system if an error persists.

### H. Scalability

It defines the ability of the algorithm to adapt to growth of the system, in terms of number of nodes or workloads.

### I. Drop Rate

Refers to the number of packets dropped during transmission. From node to node.

### J. Waiting Time

Time spent in ready queue is defined as waiting time; the shorter the time spent in the ready queue, the better the performance.

### K. Performance

It is used to assess the system's efficiency.

## V. COMPARISON AMONG VARIOUS LOAD BALANCING TECHNIQUES

This section is mainly to provide a comparison among all of the algorithms discussed in the paper. Static load balancing algorithms advantages are mainly in the domain of being simple in implementation and very stable. Additionally, these have low communication overheads as they do not need to monitor the current system state information [5]. This feature is however its own disadvantage as because of this, they cannot adapt with the entire system and thus may cause the system to fail. For Dynamic load balancing algorithms since they take the current state of the system into account while distributing workloads, means that these algorithms have to take the current state of all the nodes into account thus will often have higher communication overheads with complex algorithm structure. These algorithms will however provide much better and efficient performance.

### A. Comparison of Static Algorithms

The 'Table I' below lists many comparisons among static load balancing approaches.

### B. Comparison of Dynamic Algorithms

The 'Table II' below lists many comparisons among dynamic load balancing approaches.

### C. Comparison of Hybrid Load Balancing Techniques

As a result, we discover that both techniques have their own set of advantages and disadvantages, therefore research in the subject of load balancing is still ongoing, and a hybrid of the two has been produced, namely the Dynamic Biasing algorithm and the Minimum Load State Round Robin Algorithm (MLSRR). These use hierarchical model to accomplish their goal with two goals in mind: being simple like Static algorithms with being adaptive as like Dynamic algorithms.

This structure is similar to those described in refs. [18] [19]. To minimize communication overheads, load balancing is done at two levels in the defined framework: group level, which is managed by the Global Load Balancer (GLB), and node level, which is performed by the Local Load Balancer (LLB). A group is composed of several nodes, where each group has one selected representative node (DR) and only Local load balancing DR can interact with the GLB. This is shown in "Figure. 3".

*1) Dynamic Biasing Algorithm [20]:* Load balancers employ the approach of determining weights for groups and nodes. Weight is referred to as a bias, and this process of identifying biases is referred to as biasing. Each node communicates about its current load status to its group's DR, which performs local load balancing in that group and is connected to the global load balancer, at predefined periods termed state checking times. The nodes in each local load balancer are ordered depending on their load statuses during the biasing process. If the number of jobs is less than a certain threshold, the algorithm sorts the nodes in ascending order of load status. Pushing policy deployed to exchange the current status information of the system's groups and nodes. This strategy eliminates unnecessary queries from the system, lowering communication costs.

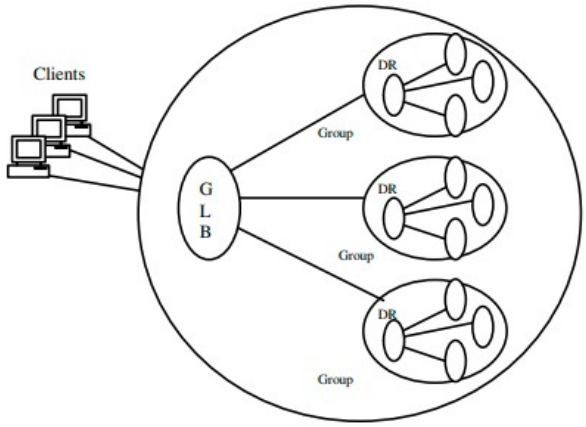


Fig. 3. Gobal Load Balancer (source: Iman Barazandeh, S S Mortazavi and A M Rahmani,2009.).

*2) Minimum Load State Round Robin Algorithm (MLSRR) [21]:* At periods known as state checking time, each node communicates their current load state to its group's DR in this technique. Based on the state information of the nodes, each local load balancer distributes received loads from the global load balancer to the nodes in their respective groups. This method is built in such a manner that the group or node that is chosen as the minimum in one biasing may not be chosen as the minimum in the following biasing. To decrease communication overheads, a pushing policy is used

In a computer simulation, the outcomes of Dynamic Biasing and MLSRR algorithms are compared to Random Biasing Algorithm and Round Robin Algorithm by assessing three criteria: Drop Rate, Throughput, and Response Time. The table below depicts the results.

TABLE III
COMPARISON BETWEEN TWO HYBRID LOAD BALANCING ALGORITHM

| | Dynamic Biasing Algorithm | MLSRR | Randomized Algorithm | Round Robin |
|---|---|---|---|---|
| Drop Rate | Low | Low | High | High |
| Throughput | Higher | Higher | High | Low |
| Response Time | Low | Low | High | High |

## VI. CONCLUSION AND FUTURE SCOPE

Load balancing's ultimate purpose is to distribute workload evenly among all nodes in order to achieve maximum performance while keeping overheads to a minimum. Response time can only be improved with efficient load balancing algorithms. The comparison of several load balancing methods on the

TABLE I
COMPARISON BETWEEN STATIC LOAD BALANCING ALGORITHMS.
(SOURCE: BENIWAL, P AND GARG,2014. MR. VIPINWANI, 2018)

| | Round Robin | Central Manger | Threshold Algorithm | Randomized Algorithm |
|---|---|---|---|---|
| Reliability | Less | Less | Less | Less |
| Adaptability | Less | Less | Less | Less |
| Overhead | Low | High | Low | Low |
| Throughput | Low | Low | Low | High |
| Process Migration | No | No | No | No |
| Response Time | Less | Less | Less | Less |
| Resource Utilization | Less | Less | Less | Less |
| Fault Tolerant | No | Yes | No | No |
| Scalability | High | Low | High | Low |
| Waiting Time | More | Low | High | Low |
| Performance | Low | Low | Low | Less |

TABLE II
COMPARISON BETWEEN DYNAMIC LOAD BALANCING ALGORITHM.
(SOURCE: BENIWAL, P AND GARG,2014. VINITA MATHUR,2017)

| | Central Queue | Local Queue | Least Connection | Ant Colony Optimization | Honey Bee Optimization | Practical Swarm Optimization |
|---|---|---|---|---|---|---|
| Overhead | High | High | High | High | High | Low |
| Throughput | High | High | Low | Low | Low | High |
| Process Migration | No | Yes | No | Yes | Yes | Yes |
| Response Time | More | More | Less | Less | Less | Less |
| Resource Utilization | Less | More | More | More | More | Less |
| Fault Tolerant | Yes | Yes | No | Yes | Yes | Yes |
| Scalability | Low | High | High | High | High | Low |

basis of specific factors is carried out in this study. The comparison demonstrates that static load balancing algorithms are more stable than dynamic load balancing algorithms, yet dynamic load balancing techniques are preferred over static load balancing algorithms owing to their ability to function accurately in distributed systems increasing the overall efficiency of the system [16] . In addition, two hybrid algorithms have been presented and compared to certain traditional load balancing algorithms that are based on a hierarchical structure and regulate loads in two layers of groups and nodes. The first technique allocates the arrival loads to groups and nodes depending on their present load condition with certain biases. The second technique picks the group and node with the lowest load status and sends incoming loads to that group and node [20]. Hopefully, this study will aid in the development of novel load balancing algorithms in the future, and also making more hybrid algorithm also make distributed systems simpler and more efficient.

## REFERENCES


[1] Soundarabai .P ,Sahai .R , Venugopal .R, Patnaik .L : COMPARATIVE STUDY ON LOAD BALANCING TECHNIQUES IN DISTRIBUTED SYSTEMS, International Journal of Information Technology and Knowledge ManagementVolume 6, No. 1, pp. 53-60 , December 2012.

[2] Branko Radojevic, Mario Žagar, "Analysis of Issues with Load Balancing Algorithms in Hosted Cloud Environments", Opatija, Croatia, MIPRO 2011, May 23-27, 2011.

[3] Sandeep Singh Waraich. Classification of dynamic load balancing strategies in a Network of Workstations. Fifth International Conference on Information Technology: New Generations. April 2008. pp. 1263 - 1265.

[4] Abhijit A. Rajguru, S.S. Apte "A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-1, Issue-3, August 2012.

[5] D. L. Eager, E. D. Lazowska. and J. Zahojan. "A Comparison of Receiver-Initiated and Sender-Initiated Adaptive Load Sharing". Proc. of the 1985 ACM SIGMETRICS Conference on Measurement and Modelling of Computer Systems. August 1985. pp. 1-3.

[6] Mohsen and Hossein Delda "Balancing Load in a Computational Grid Applying Adaptive, Intelligent Colonies of Ants". Informatica 32 (2008) 327–335.

[7] Abubakar, Haroon Rashid and Usman, "Evaluation of load balancing strategie", National Conference on Emerging Technologies, 2004.

[8] Sandeep Sharma, S.Singh and Meenakshi, "Performance analysis of load balancing algorithms", World academy of science, 2008.

[9] Daniel Grousa, Anthony T. , “Non-Cooperative load balancing in distributed systems”. Journal of Parallel and Distributing Computing, 2005.

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

[11] Jain P. and Gupta D. “An Algorithm for Dynamic Load Balancing in Distributed Systems with Multiple Supporting Nodes by Exploiting the Interrupt Service”. Academy Publisher, 232-236, 2009.

[12] Suriya and Prashanth , “Review of load balancing in cloud computing”. IJCS, vol.10 issue.1, Jan 2013. 25.

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

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

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

[16] Beniwal, P and Garg, A “A Comparative Study Of Static And Dynamic Load Balancing Algorithms.” in Int Journal of Advance Research in Computer Science and Management Studies, December 2014

[17] Mr. VipinWani ,”A Comparative Study of Various Load Balancing Strategies for Performance Analysis in Distributed System”, International Journal for Research in Engineering Application and Management(IJREAM),2018

[18] Ming-Chang Huang, S. Hossein Hosseini and K. Vairavan. A ReceiverInitiated Load Balancing Method In Computer Networks Using Fuzzy Logic Control. IEEE. GLOBECOM 2003. vol. 7. Dec. 2003. pp. 4028 – 4033.

[19] Hyo Cheol Ahn, Hee Yong Youn, Kyu Yeong Jeon, and Kyu Seol Lee. Dynamic Load Balancing for Large-scale Distributed System with Intelligent Fuzzy Controller. IEEE International Conference on Information Reuse and Integration. Aug 2007. pp.576 – 581.

[20] Iman Barazandeh, S S Mortazavi and A M Rahmani. “Two New Biasing Load Balancing Algorithms In Distributed Systems.” in 2009 First Asian Himalayas Int Conference on Internet, Nov. 2009

[21] P. Beaulah Soundarabai, Sandhya Rani A., Ritesh Kumar Sahai, Thriveni J., and K.R. Venugopal,” COMPARATIVE STUDY ON LOAD BALANCING TECHNIQUES IN DISTRIBUTED SYSTEMS” International Journal of Information Technology and Knowledge Management , Volume 6, No. 1, pp. 53-60, December 2012

[22] Zahra Mohammed Elngomi and Khalid Khanfar ,”A Comparative Study of Load Balancing Algorithms: A Review Paper”, International Journal of Computer Science and Mobile Computing, Vol.5 Issue.6, pg. 448-458,June- 2016

[23] Vinita Mathur , ”A Comparative Study of Load Balancing Techniques in Distributed Systems”,International Journal of Innovative Science, Engineering and Technology, Vol. 4 Issue 6, June 2017