

# Survey on Load Balancing Techniques

**Shantanu Shukla**

*Research Scholar, Department of Computer Science & Engineering,  
Pranveer Singh Institute of Technology, Kanpur, India*

**Raghuraj Singh Suryavanshi**

*Associate Professor, Department of Computer Science & Engineering,  
Pranveer Singh Institute of Technology, Kanpur, India.*

## Abstract

Load balancing is challenging task in distributed system. Distributed system is collection of autonomous system where data is allocated at various nodes or sites. Several tasks are initiated at these sites. Due to improper distribution of tasks, system may be overloaded. In this paper, we propose various load balancing Technique that resolve the issues by applying node movement and replication technique. We have considered some load transfer to high popularity data transfer from overloaded to under loaded node. This paper describes newly developed and essential concepts for several load balancing techniques in a distributed system environment.

**Keywords:** Load balancing, Static load, Dynamic Load, Threshold, Under loaded node, Overloaded node.

## I. Introduction

Load balancing is essential part in distributed computing. It is one of the main problems which have to be solved to achieve high performance in Distributed environment. We have to consider many approaches and techniques for proper load balancing [1][2]. In many of the approaches, we transfer the load from one node to another that means from overloaded node to under loaded node [3][4]. This technique is useful but it is older one. For proper implementation of load balancing clustering of sites can also be done where one or many clusters are formed to balance the load. The notion of maximum, minimum and normal threshold is considered for proper load balancing. We have various schemes to survey on the basis of static and dynamic load balancing. In popular technique we have considered transfer of load and proper distribution of node's load balance to improve the response time and utilization of resources. Load balancer is a network device that used in distributed system, and balancing is used in cloud environment to improve the load and increase availability of application. In network, a load balancer is between the client and server machine to accept incoming traffic and after calculation by various algorithms data will send to client in limited time and approx data. A load balancer reduce the individual server load to rescue become the overloaded node and also prevent from a single point of failure. Thus load balancing is improving the overall system. But if load balancing is not done in a proper way then we have faced a very big issue in networking.

The organization of paper is as follows: Section II describes static load balancing schemes, section III

explains several dynamic load balancing approaches, section IV gives comparison of different load balancing in different aspects and finally section V conclude the paper.

## II. Static load balancing

Static load balancing [5][6] is obtain by earlier information giving in the system. The performance of the process is processed under the execution. The load information is calculated at one node and submits the information to execute in remote based node. Depending on the calculation of work load in distributed allocate the load and start the execution without concerning the previous load. Non-preemptive nature is obtained by static load balancing. System load is not dependent to current state but it requires resources and weightage of the system. In static workload load balancing algorithm distribution of load is done earlier. There are several algorithms in static load balancing for proper load distribution [4]. The discussions of several schemes are as follows.

### a) Round robin algorithm

Round robin load balancing [7] is one of the simplest approach, distributed the client information across a group of servers. This technique is followed in a small network where server is limited. In this algorithm we have multiple clients and multiple servers. When a client initiate request and pass to server, the first server is allocate the request. When second request is initiated it is allocated to the second server and third request come to the third server. But if server receive fourth request then it is allocated to the first server and create a ring type allocation. The number of requests that can be processed is limitation of the server.

In Round Robin algorithm we have defined 2 types of RR algorithm:

- A. Classic Round Robin Algo
- B. Weighted Round Robin Algo

In Classic RR algo we defined the approach of particular time slice given to each process. While in weighted Round robin algorithm we have assigned the weight to every node. That node is based on configuration and criteria to process the request. In there we have assigned a particular time and that time we execute the process. Any particular process which have higher load that process time is large. So Weighted RR is efficient for small and medium process.

**b) Randomized Algorithm**

In randomized Algorithms, we select the node at random basis not perform a particular basis. A server sends the request to node which has low load value and then the request can be processed to that node. This particular operation is performed by Load balancer. In this algorithm we have suffered a overloaded problem, any node which have loaded proceed to the overloaded node

**c) Central Manager Algorithm**

In central Manager Algorithm of load balancing, we choose a central node among many nodes and choose on the basis of how many requests this can be accept. After choosing the central node we manage the request. Transfer of load from overloaded node to lightly node and create a right path to define the main work is done by central node.

This algorithm does not work in distributed manner. We have to manage all the function to manage by a central node. This is not efficient for large and distributed network.

**d) Threshold Algorithm**

In threshold algorithm nodes are assigned by the weight and each node defines a threshold value. Node capacity define their limitation which should not be crossed the threshold limit. The update load of a server machine could be in one of the following three levels:

- A. Underloaded
- B. Normal
- C. Overloaded

if  $load < threshold$  (Underloaded)  
if  $load > threshold$  (overloaded)  
if  $load \leq threshold$  (Normal)

In initial level all the nodes are under loaded, when the load state crosses the threshold limit, node is called overloaded node. When any node is reached to threshold limit it will not accept the other request.

**e) Min-Min LB Algorithm**

In this algorithm, all information is predefined in node. We have to create a set of all pending jobs where we have to calculate execution time. The jobs with minimum execution time are selected then after the selection of the node and transfer of task take place. The job is processed according to the selection time until all jobs are assigned in node to execution. The advantage of this algorithm is that the node having less execution time will processed first. The drawback of this algorithm is that node will have more execution time and the nodes which are not processed in time will lead to starvation.

**f) Min-Max LB Algorithm**

This algorithm will work opposite [7] as compare to min-min algorithm where minimum node execute first. In this algorithm, after finding mean completion time of all the nodes is calculated, then the nodes whose completion time is greater

than mean value is selected and those jobs will be reprocessed to the node which has least completion time.

**III. Dynamic Load Balancing**

In this technique, multiple nodes are responsible to make correct load balancing decision. In dynamic approach [3][4][5][7] the node is not certain and request is not in a simple manner. Any time Load will be less and another time load will be high. Therefore, we have assigned the dynamic balancing algorithm which gives correct result in run time environment. Every node contains the information of one another node or neighbor node. Failure of a node is not neglected in distributed scenario, we have new concept to make the system as fault tolerant. So in today approach we perform the dynamic load balancing instead of static load balancing. Discussion of several dynamic algorithms are as follows:

**a) Nearest Neighbor Algorithm(NNA)**

In nearest neighbor algorithm, each processor performs the load balancing algorithm. Under this algorithm we search neighborhood node if succeed to find overloaded node in neighbor then migrate the request otherwise those neighbor is also send load to own neighbor. Load information is exchange locally means we have performed the work in limited environment. In this algorithm, all the processes maintain the adjacent processor and load distribution of all the nodes is equally divided. For successfully local load distribution global load balancer is considered. The nearest neighbor algorithm is further categorized as [6]

- A. Diffusion Method
- B. Dimension Exchange Method

**b) Random Algorithm (RAND)**

In random base Load balancing Algorithm [3], if the workload (which is greater than threshold limit) is specified by any node, it is migrated to a randomly selected node. It does not check node's information, whether it is overloaded or under loaded. This algorithm neither maintain any local load information nor send any load information to other processors. Further, it is easy to design and simple to implement. But it causes so much communication overhead due to the random selection of highly loaded node. So information exchange is locally organized and each new node is selected random process. And important point in this algorithm, the load is distributed asynchronous.

**c) Adapting contracting with Neighbor(ACWN)**

In this algorithm, we maintain a table and it is updated every time to transfer the load. So ACWN does require to maintain the local load information and also the load information of the neighbors is exchanging the load time to time basis [3]. Hence, random algorithm is different from the ACWN in a respect that ACWN automatically finds the destination node which is less loaded in neighbors. So ACWN is important in Dynamic load balancing approach. and ACWN information exchange is locally based.

**d) Cyclic Algorithm**

In Cyclic based load balancing algorithm [4], the outcome of RAND algorithm is slightly change to perform in the cyclic algorithm. The load is assigned in a system via cyclic manner. The advantage of this algorithm is that process which receive the information will not received request for particular time, unless the cycle is not completed at one rotation. Information is stored by the source process. If the source is fail then all of the information sends back to particular node. In this algorithm the information exchanged is locally basis.

**e) Probabilistic**

In probabilistic based load balancing algorithm, information about each node is maintained and according the information node is given the probability. On the basis of probability, the node which is higher probable will accept the information from the less probable node means under loaded node accepts the load from highly loaded node. In order to find the probability, information about each node is calculated by formula load/total load.

**f) Prioritized Random Algorithm**

In priority random load balancing [5], we have to select the node's load randomly which have overloaded node like random algorithm. In PRAND, a priority is assigned to each node, the node having higher priority will send the load to other node having less priority.

**g) Threshold and least**

In Threshold and least Algorithm [4], these are separate algorithm but used similar approach. These used partial knowledge obtained by the message exchange method. In Threshold algorithm, if the load is below the particular threshold then we have received the load from another overloaded node. If no node is available for transfer then overall system is underloaded. After some time we calculate the load and search the node to transfer. After a particular time if no proper node has been found to transfer the load then the process is executed locally. LEAST algorithm is a part of Threshold algorithm. We calculate the least response time, which node have response time is low that nodes are highly active to accept the load. These communications are uniformly local. So, both algorithms are simple and important for less busy environment.

**h) Reception**

In the reception-based algorithm [8], nodes having below the threshold value will find the overloaded node by random manner or by using the approach of polling and transfer the load from overloaded to lightly loaded node. This algorithm is similar to receiver based approach and transfers the information on locally basis.

**i) Centralized information and centralized decision**

In centralized information algorithm [4], information is available on a single node. All other nodes communicate with central node, and decision is based on a single based approach. Central node is a master node of algorithm. When a over loaded node wants to transfer a load, it requests a central server and that central server search for a lightly loaded node.

Every node in the system connects the central server machine that server is responsible to transfer the node and decision is also the centralized. But this algorithm suffers from major problem that if the server is not performed well or failed, all the nodes to proceed to failed state.

**j) Centralized information and distributed decision**

In Central based information [4] the information is centrally but the decision of the node is distributed. In this approach, server broadcasts the information about load from each node. So each node contains the information of each other. Information about overloaded node can be find easily and transfer the load from heavy node to light node take place without interruption the server. This algorithm is efficiently performed the work because of less message information exchange. It is also robust in nature because failure of central node is not affect the algorithm to perform the load operation perfectly. But drawback of this algorithm is that we have connected the global environment, much of the information is not in updated.

**k) The Shortest Expected Delay (SED) Strategy**

Shortest Expected Delay based algorithm [5] is based on expected time delay. After each job completion, delay value is calculated from source to destination node and select those nodes which node's delay is minimal. This algorithm is similar to based on greedy based algorithm. In this approach, each node performed the role according to its best functioning and join the queue which have minimum expected delay time in completion the node.

**l) Never Queue**

Never queue strategy is separate algorithms from the other ones. In this algorithm [8], job is sent to each destination throw the sending server and calculation of cost is done. Main working of this algorithm is that immediate request can be processed and not wait for queue basis. This technique minimize the delay time into further reaching process, so the result is minimize delay from overall process. But the drawback of this algorithm is server does not proceed incoming job until a faster server is available to proceed the job efficiently. So never queue algorithm is good for quick response.

**m) Tilling or Direct Neighborhood Algo**

In Tilling or DN algorithm [4] the information exchange is local. On performing the load distribution we balance the process from same window. The load is distributed among the process in the window. Communication between the processes is uniformly local and all the task will independently execute. In this algorithm we have balance node to transfer the neighbor directly in same window.

**n) X tiling Algo**

In this approach we consider a hypercube. Balancer processor are connected in hypercube [5]. The load is distributed among processes in hypercube. Node communication is globally uniform and initialization of the algorithm is periodic. In periodic approach, node can never be find the random approach. X-tilling approach is locally exchange information.

This algorithm is synchronous based that means we have synchronize the nodes in hypercube manner.

**o) Throttled based Algorithm**

In the Throttled Load Balancer (TLB), we have to maintain a record of the state in each node and create a table. When a request is submitted, it searches the predefined node in table and if a match is found on the basis of availability of the node and also selected nodes have same size, then the request is granted otherwise negate value is returned and the request is queue back to the table. This is very slow load balancing process. Therefore, we cannot consider the algorithm for large load balancing process. In this algorithm, we have ignored the request which have more loaded and compress the request to rectify the load.

**p) Divide and conquer**

In this approach, we divide load from process then perform the load operation. After performing the load, we merge in the pair of two nodes then these nodes perform the load balancing. Same approach will be applied for all nodes and in last we found the totally merge (conquer) nodes and finally all nodes is balanced. In this algorithm, we process the request if any large request is not execute sufficiently then we divide the sub request into process.

**q) Pipelining**

This algorithm is based on pipelining. The first stage of output will works as an input for the second stage, hence a pipe is created, called virtual pipe. In pipelining-based approach [5] if the first load is processed then we start second process without interrupting the first one. We have calculated the load of many nodes at a time because we will assume the loads are in pipelining manner and these pipelines is not interrupt to one another.

**r) Dynamic Ratio**

Before explaining dynamic ratio, we first explain the static ratio method. In static we define the ratio which server perform and how much is ram provide. We will see how much capacity and request can be handled by the node. In static we define manually in each server ratio.

In dynamic ratio method [9] we define each server has different ram and all the server is managed by central unit called load balancer. In load balancer all the information in stored in table. In dynamic ratio we define ratio in dynamic basis means each time when we have processed the request, ratio will be changed.

**IV. Comparison of various techniques in different aspects**

Aspects/Algorithm	Performance	Throughput	Overhead	Fault tolerance	Migration time	Response time
RR	Avg	Avg	Less	No	Less	Min
WRR	Good	Good	Less	Yes	Normal	Min
RAN DO-MIZE D	Bad	Avg	Less	No	Less	Min

CM	Avg	Avg	Normal	Yes	Nor-mal	Min
THRESHOLD	Good	Good	More	No	more	Max
MIN-MIN	Avg	Avg	Less	No	Less	Min
MIN-MAX	Avg	Avg	Less	No	Less	Min
NNA	Avg	Good	Less	No	Less	Min
RAND	Bad	Good	More	No	Less	Min
ACWN	Avg	Good	More	Yes	Normal	Min
CYCLIC	Avg	Avg	Normal	Yes	more	Max
PROBABILISTIC	Avg	Avg	Normal	Yes	Less	Min
PRA ND	Avg	Good	More	Yes	Less	Min
Threshold & Least	Good	Good	Less	No	Less	Min
Reception	Avg	Avg	Normal	No	Nor-mal	Max
CICD	Avg	Avg	More	No	Less	Min
CIDD	Good	Good	More	Yes	more	Min
NQ	Avg	Avg	Less	No	Less	Min
TILLING	Avg	Good	Less	No	Less	Min
Divide & conquer	Good	Avg	More	Yes	Normal	Max
Pipelining	Avg	Good	Normal	Yes	Nor-mal	Min
Dyn. Ratio	Good	Avg	More	Yes	More	Max

**V. Conclusion**

This paper presents a detailed survey of load balancing algorithms in distributed environment. Distributed system is collection of nodes where multiple tasks are submitted at different nodes. Due to random arrival of tasks at different nodes, it may possible that few nodes are heavily loaded and few of them are lightly loaded. In this paper, we present several static and dynamic load balancing approaches under different categories. But in the entire algorithm we discuss mainly overloaded and underloaded process with threshold value. Also we explain merits and demerits of each algorithm.

**Acknowledgement**

This work is done under the DLSR project (Distributed load balancing and system recovery) governed by Uttar Pradesh Council of Science and Technology and supported by PSIT College Kanpur.

## References

- [1] Neeraj Rathore & Inderveer Chana " Load balancing and job Migration Techniques in Grid-A Survey of Recent Trends" In wireless Personal communication ISSN 0929-6212
- [2] Narjes Soltani and Mohsen Sharifi, " A load balancing algorithm based on replication and movement of data items for Dynamic structured P2P System " International Journal of Peer to Peer Networks (IJP2P) Vol.5, No.3, August 2014
- [3] P. Beulah Soundarabai, Sandhya Rani , Ritesh Kumar Sahai, Thriveni J., K.R. Venugopal and L.M. Patnaik " Comparative Study of load balancing Techniques in Distributed System" International Journal of Information Technology and Knowledge Management December 2012, Volume 6, No. 1, pp. 53-60
- [4] Zubair khan, Ravidra singh, Jahagir alam , Shailesh Saxena "classification of load balancing condition for parrel and distributed system" IJCSI vol 8 Issue 5 september 2011
- [5] Rakesh kumar katare and mamta kumari " A comparative study of various load balancing algorithm in parallel and distributed multiprocessor system" International Journal of Computer Applications (0975 – 8887) Volume 169 – No.10, July 2017
- [6] Zahra Mohammed Elngomi and Khalid Khanfar " A Comparative Study of Load Balancing Algorithms: A Review Paper" IJCSMC, Vol. 5, Issue. 6, June 2016, pg.448 – 458
- [7] Nischol Mishra and Nitin kumar mishra " Load Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing- A Systematic Review " International Journal of Computer Applications (0975 – 8887) Volume 131 – No.18, December 2015
- [8] Dr.Chinthagunta Mukundha, Nampally Venkatesh and Kamatagi Akshay " A Comprehensive Study Report on Load balancing Techniques in Cloud Computing" International Journal Of Engineering Research And Development e-ISSN: 2278-067X, p-ISSN: 2278-800X, [www.ijerd.com](http://www.ijerd.com) Volume 13, Issue 9 (September 2017), PP.35-42
- [9] Paulsingh Beulah Soundarabai , Venugopal K R, Ritesh Kumar sahai and Lalit M patnaik " Comparative Techniques on load balancing in distributed system" International Journal of Information Technology and Knowledge Management December 2012, Volume 6, No. 1, pp. 53-60