A Classification of Job Scheduling Algorithms for Balancing Load on Web Servers

Sairam Vakkalanka
School of computing, Blekinge Institute of Technology, Karlskrona, Sweden-37141

ABSTRACT: *Through this report, a classification of different job scheduling algorithms available for balancing the load on web servers is made. Types such as static and dynamic scheduling algorithms are thoroughly discussed and the strengths and weaknesses of these algorithms are put forth through this article.*

Keywords: *Load balancing, scheduling algorithms, web servers, traffic, load index*

I. INTRODUCTION

With the rapid increase and growth of World Wide Web (WWW), grew the usage of several complicated and computation-intensive applications, which require high degree of computation and higher bandwidth for the transmission of data [26]. These applications may vary from cloud based, multimedia, design and development, e-commerce etc [25]. With these options being made available for users all over the world, there is an exponential increase in the usage of network bandwidth. This increase or change is not only affected by the traffic but also by the nature of traffic, which in the era where web servers were used for the first time were used only to transfer plain texts or images [25]. Now, with the explosion of data, traffic and low bandwidth problems, balancing the load on these web servers play a vital role.

II. TRAFFIC AND ITS TYPES

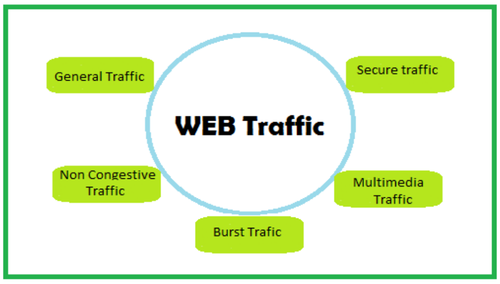
As stated earlier, load on these web servers not only depends on the traffic but also on the type of traffic. According to Kotogiannis et.al [13], traffic on these web servers can be classified into

- **General traffic**
- **Secure traffic**
- **Multimedia traffic**
- **Burst traffic**
- **Non congestive traffic**

General traffic:
This sort of traffic can be stated as the traffic generated due to request for data such as the plain text documents or static content on web pages and dynamic content [13].

Secure traffic:
This type of traffic is mostly generated by e-commerce applications, which largely run on the SSL- TTL protocol [18].

Multimedia traffic:
The multimedia traffic is a sort traffic which is generated by the streaming of data which may either be video or audio [18]

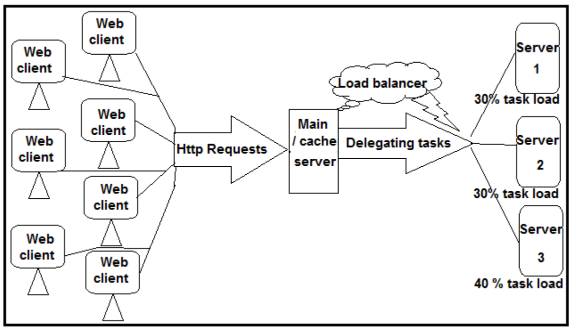


Non congestive traffic:
Though this sounds like general traffic, it is distinguished in terms of the size of the packet [13][20]. The packet size in a non congestive traffic is usually small (NCQ threshold)[13][20]. This kind of traffic never leads to jitter or delay [13][20].

Burst traffic:
This type of traffic is mainly caused due to packet which are transferred in bursts such as P2P transfers and file downloads or uploads etc [16][13].
With these different types of traffic exist different load balancing techniques. These load balancing techniques and their types are discussed in the section below.

III. LOAD BALANCING

Load balancing is used to distribute work between two or more processors, computers, networks or memory devices in order to channelize the resources in an efficient manner and to get optimized response times and throughputs [1]. Load balancing can be defined as an approach to increase and improve the performance of two or more nodes or links connected nodes by the redistribution or the reassignment of load [6][9][10]. The figure below explains how load balancing works in a web.



A. Main Goals of load balancing

According to [6][11], Balancing the load on the nodes and links in a distributed setting is always driven by the goals discussed below

- To provide, a plan B when a single node or group of nodes fail.
- To improve the overall performance of the connected nodes or network.
- To maintain the stability of the systems connected.
- To make available systems for easy future modifications.

This load balancing is always fruitful and has many advantages when the goals are satisfied. The advantages of load balancing are discussed in the following section.

B. Features and advantages of Load balancing

Balancing the load on servers comes with added features and benefits though increases the cost of communication and transfer between the nodes. Some of those advantages and features are listed below:

- Load balancing protects the servers from Distributed denial of Service attacks (DDos)
- Balancing the load improves the reliability of systems, reducing the crashes on the nodes caused due to overload.
- Load balancers can help buffer response from the servers and slowly send to the clients who are down, reducing the burden and waiting time on the servers.
- Load balancers have the feature of asymmetric load distribution where overloaded tasks can be assigned to servers at the backend.
- Load balancing helps in improving functionality, stability, reliability and maintainability of the servers.

Load balancing can be considered as a process which is carried out in such a way that no processes are overloaded but kept busy [1]. In order to know if a node is busy or not and to check the load on the node, Load index is calculated.

C. Load Index

Load index is used to identify or to detect an imbalance state [1]. An imbalance state occurs when the load index of a particular node is greater than the load indices of others which vary with a variation in the performance measure of interest [1]. The performance measure of interest can be anything, for example the Length of the CPU queue can be considered when the performance measure of interest is the average response time [1][3][4]. All load balancing algorithms are based on this load index and also some governing policies which are discussed below.

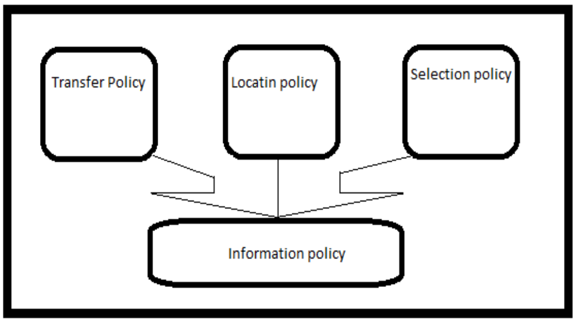
D. Load balancing policies

All load balancing algorithms are based mainly on four policies, which are responsible in keeping the systems updated with the information of workload on the nodes [1]. The four policies which govern the load balancing algorithms are as follows

- Global Information Policy
- Transfer Policy
- Location Policy
- Selection Policy

Information Policy gives all the nodes an access to the load indices of each and every node, which comes with an added cost of extra effort needed for communication in order to maintain the exact information of the nodes[1][2][5][6]

Transfer Policy determines when a node can distribute the load or transfer a job to the other node, also when a node can receive the load or retrieve a job from another node [1][6]. A node becomes eligible to transfer or receive when it crosses or reaches a certain threshold limit which is determined by the total average load on these nodes [1][6]

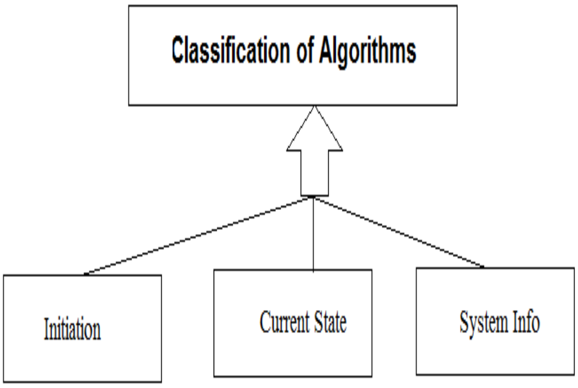


Location Policy determines which node needs to be paired with another in order to accomplish the transfer of load or job [1]. If the node is a sender then location policy looks for a receiver and vice versa [6].

Selection Policy selects the appropriate jobs from the queued jobs in order to retrieve / transfer the task to an eligible receiver / sender [1]. This policy works on the principle of minimizing the cost required to transfer the jobs from one node to the other [1][6]

IV. Scheduling Algorithms for Load Balancing

The main aim of scheduling algorithms is to improve the stability, reliability and performance of systems which are connected in a network. There exist different kinds of scheduling algorithms which are explained below:



Classification of scheduling algorithms in load balancing can be done in three ways as explained by different authors are as follows

- Classification based on Initiation
- Classification based on system information
- Classification based on state of the current system

E. Classification based on Initiation

Here, scheduling algorithms are classified based on the job transfer initiation process [6][11].

- Sender initiated algorithms
- Receiver initiated algorithms
- Symmetric algorithms
- If sender initiates the process, then the algorithms pertaining to the sender are considered as sender initiated algorithms [6][11].
- If the receiver initiates the process, then the algorithms which fall under this category are considered to be receiver initiated algorithms [6] [11].
- If both sender and receiver simultaneously initiate then they are considered to be symmetric algorithms [6] [11].

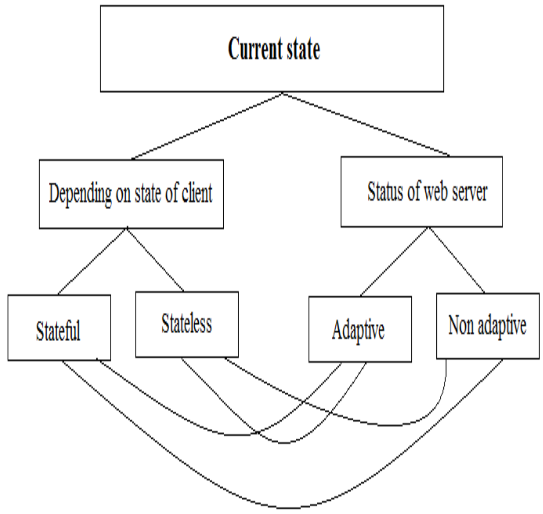
F. Classification based on state of current system

Depending on the state of the systems, load balancing algorithms can be classified into two ways

❖ Depending on the state of client request
❖ Depending on the status of the web server

- **Depending on the state of client request**

If algorithms need information regarding connection requests made by nodes or clients connected in a network [13], then they are classified into



State full Algorithms

These are those algorithms which require the information regarding connection requests made by the nodes [13].

Stateless algorithms

These are those algorithms which do not require the information regarding the connection requests made by the nodes [13].

- **Depending on the status of the web server**

Based on the status of the server [13], algorithms can be classified in to two ways

Adaptive algorithms

These are those algorithms which require the status of the server [13].

Non adaptive

These are those algorithms which do not require the status of the server [13].

These are again combined into four categories namely,

➢ Stateless non adaptive
➢ State full adaptive
➢ State full non adaptive
➢ Stateless adaptive

- **Stateless non adaptive**

These algorithms do not take into regardsystem information where it may be the client connection status or the status of the web server [13]. Algorithms such as Random and round robin algorithms come under this category stateless non adaptive algorithms [13][15][19].

- **State full adaptive**

These are those algorithms which make use of information from both servers and nodes, which is based on the ratio of Number connection requests at a node to the average connection requests received with a particular time interval [13].

t2-t1 : R$_i$ = $|ci2 - ci1 / \frac{1}{n} \sum_{i=1}^{n} Ci1$.. [13][21][24].

Least loaded algorithm which falls under this category makes use Weighted round robin method [13] [14].

- **Stateless adaptive**

These are those algorithms which take into consideration the server side information and are not concerned with current state of the client [13].

Fastest response time algorithm falls under this category of stateless adaptive algorithms..

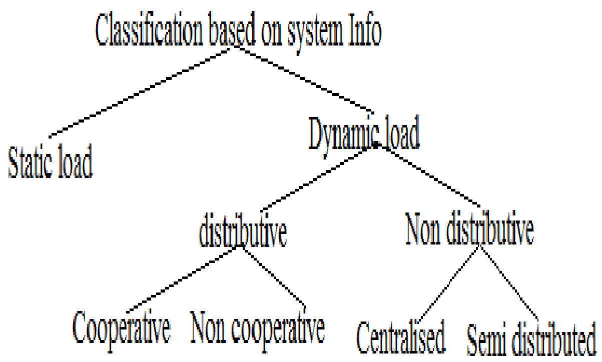
- **State full non adaptive**

These are those type of algorithms which take into account information pertaining to the client requests [13].

Algorithms such as weighted round robin algorithm, list based weighted round robin algorithm, Least connection-weighted least connections algorithm, Shortest expected delay, Never queue scheduling algorithm, Destination hashing locality based scheduling algorithm etc [13][21][22][23][24].

G. Classification based on the system information

Based on the system information required algorithms can be classified in to two types



- Static Load balancing algorithms
- Dynamic load balancing algorithms

Static load balancing algorithms

Algorithms which fall under this category require prior knowledge of the system and do not depend on the current state of the system [6]. Here, while balancing the load on the servers, the performance of the servers is determined and known prior to execution of new tasks [6]. With the information obtained from the previous tasks or before starting a new task, the load on the server is distributed based on the performance statistics obtained earlier. Here a master processor distributes the work and the slaves process estimate and calculate the load and send the results to the back to their master [6][8]. Keeping in mind to minimize the communication costs, the main goal of static load balancing algorithms is to reduce the execution times of the tasks [6].

Algorithms such as Round robin, randomized algorithm, Central manager algorithm, threshold algorithm etc fall under this category of static load balancing algorithms [6].

Dynamic load balancing Algorithms

Here, in dynamic load balancing algorithms, load balancing is done not based on prior information of the system but based on the current state of the system [6][7][12] The main difference between the static and dynamic algorithms is the calculation of load [6].

Central queue algorithm and local queue algorithm fall under this category of dynamic load balancing algorithms [6]. There are two kinds of dynamic load balancing algorithms:

- Distributed dynamic algorithms
- Non- distributed load balancing algorithms

Distributed dynamic algorithms

In the distributed algorithms, the execution and initiation of load balancing algorithm is carried out by all nodes connected and the resulting load which is calculated is shared and communicated by all the nodes in two ways[6], they are as follows:

- Co-operatively distributed
  Here, in this setting, the nodes in a distributed mode work collectively and achieve objective goals [6].
- Non co- operatively distributed
  In this type of distributed dynamic algorithms the nodes which are connected work individually to obtain objectively local goals [6].

Non-distributed dynamic algorithm

In the non distributed dynamic algorithm, not all nodes connected in a network or in system participate in the act of load balancing but only a single or a few nodes perform take up the responsibility of balancing the nodes [6]. The communication and sharing of load balance is done in two ways in non distributed algorithms, they are as follows:

- Centralized non-distributed setting
  Here in this setting only a single node is responsible for balancing the load in system, all other nodes communicate with this single node [6].
- Semi-distributed setting

Here in this type of setting, nodes connected in a system are grouped into clusters and a single node in each cluster is responsible for the balancing of load, where the remaining clusters have to communicate with this central node in the cluster [6]. The overall load balancing is carried out by these collection of central nodes [6][11].

V. Analysis and Discussion

An analysis made on the obtained results has led in identifying the benefits and shortcomings of scheduling algorithms. The advantage of Round robin algorithm is that it does not require much inter process communication but it has an drawback of not being able to achieve the expected levels of performance [6]. Similarly, the drawback of central manager algorithms is that it requires high levels of inter process communication which might create bottle neck problems [6].

VI. Conclusion

Through this report, different types of scheduling algorithms present for load balancing on web servers are thoroughly discussed, classified and evaluated. Also, benefits and shortcomings of these algorithms were identified. A complete classification and analysis of the different load balancing algorithms for web servers was discussed.

References

[1] Abbas Karimi, Faraneh Zafarshan, Adzan b. Jantan, A.R Ramli, M.Iqbal b. saripan, “A new fuzzy approach for dynamic load balancing algorithm”, *International Journal of Computer Science and Information security*, vol 6, no .1, 2009.

[2] D. L. Eager, E. D. Lazowska, and J. Zahorjan, "Adaptive load sharing in homogeneous distributed systems," *IEEE Trans. Softw. Eng.,vol.* 12, pp. 662-675, 1986.

[3] D. L. Eager, E. D. Lazowska, and J. Zahorjan, "A comparison of receiver-initiated and sender-initiated adaptive load sharing (extended abstract)," SIGMETRICS Perform. Eval. Rev., vol. 13, pp. 1-3, 1985.

[4] M. Livny and M. Melman, "Load balancing in homogeneous broadcast distributed systems, *" in Proceedings of the Computer Network Performance Symposium.* College Park, Maryland, United States: ACM,1982, pp. 47-55.

[5] W. Leinberger, G. Karypis, and V. Kumar, "Load Balancing Across Near-HomogeneousMulti-Resource Servers," presented at Proceedings, *9thHeterogeneous Computing Workshop (HCW 2000)* Cancun, Mexico, 2000.

[6] K.ramana, A subrhamanyam, A. Ananda rao, “Comparitive analysis of distributed webserver sstems load balancing using qualitative parameters”, VSRD-IJCSIT, Vol. 1 (8), 2011, 592-600

[7] S. Malik, “Dynamic Load Balancing in a Network of Workstation”, 95.515 Research Report, 19 November, 2000.

[8] Derek L. Eager, Edward D. Lazowska , John Zahorjan, “Adaptive load sharing in homogeneous distributed systems*”, IEEE Transactions on Software Engineering*, v.12 n.5, p.662-675, May 1986.

[9] G. R. Andrews, D. P. Dobkin, and P. J. Downey, "Distributed allocation with pools of servers," *in ACM SIGACT-SIGOPS Symp. Principles of Distributed Computing*, Aug. 1982, pp. 73-83

[10] Zhong Xu, Rong Huang, "Performance Study of Load Balancing Algorithms in Distributed Web Server Systems", *CS213 Parallel and Distributed Processing Project Report*

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

[12] Y.Wang and R. Morris, "Load balancing in distributed systems,*" IEEE Trans. Computing,* C-34, no. 3, pp. 204-217, Mar. 1985

[13] S.kontogianis, S. Valsamidis, P. Eframidis, A.karakos, “An adaptive load balancing algorithm for cluster based web systems”, http://skontog.gr/papers/duthtr-12-07.pdf

[14] Batheja, J., and Parashar, M, ”A framework for adaptive cluster computing using javaspaces”, *Cluster* Computing vol. 6-3 (2003), 201–213.

[15] Cardellini, V., Casalicchio, E., Colajanni, M., and Yu, P. S,” The state of the art in locally distributed web-server systems” *ACM Computing Survey*s vol. 34-2 (2002),263–311

[16] Kant, K.; Won, Y, “Server capacity planning for web traffic workload” *IEEE Transactions on Data and Knowledge Engineering*, v.11, n.5, p.731{47.

[17] Cardellini, V., Colajanni, M., and Yu, P. S. “Geographic load balancing for scalable distributed web systems”, *In Proc. of 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (2000*), pp. 20–28.

[18] Casalicchio, E., and Colajanni, M, “A client-aware dispatching algorithm for web clusters providing multiple services”, *WWW ACM (2001),* 535–544.

[19] Colajanni, M., Yu, P. S., and Dias, M. D, “Analysis of task assignment policies in scalable distributed web-server systems”, IEEE *Trans. on Parallel Distributed Systems,* vol. 9-6 (1998), 585–597

[20] Mamatas, L., and Tsaoussidis, V, “ A new approach to service differentiation: Noncongestive queueing”, *In Proc. of International Workshop on Convergence of Heterogeneous Wireless Networks (2005),* pp. 78–83.

[21] O’Rourke, P., and Keefe, M, “Performance Evaluation of Linux Virtual Server” , *In Proc. of 15th System Administration Conference, LISA* (2001), pp. 79–92.

[22] Weinrib, A, and Shenker, S, “Greed is not enough: Adaptive load sharing in large heterogeneous systems”, In *Proc. of IEEE INFOCOM’88* (1988), pp. 986–994.

[23] Zhang, W, “Linux server clusters for scalable network services” , *In Proc. of Ottawa Linux Symposium* (2000), pp. 437–456.

[24] Zhang, W, “Build highly-scalable and highly-available network services at low cost*” Linux Magazine* (November 2003).

[25] Teixeira, M, M. Santana, M. J.Santana, R. H. C, “Analysis of Task Scheduling Algorithms in Distributed Web-server Systems”, *In International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2003)*, p.655{63. Montreal, Canada, jul., 2003.

[26] Jiani Guo and Laxmi Narayan Bhuyan, ”Load Balancing in a Cluster-Based Web Server for Multimedia Applications” , *IEEE Trans. Parallel Distrib. Syst.* 17, 11 (November 2006), 1321-1334.

ABOUT THE AUTHOR

Sairam Vakkalanka is now pursuing masters in software Engineering at Blekinge Tekniska Högskola (Blekinge Institute of technology), Karlskrona, Sweden.