International Journal of Engineering

Journal Homepage: www.ije.ir

Load Balancing Approaches for Web Servers: A Survey of Recent Trends

A. Shukla*, S. Kumar, H. Singh

Department of Computer Science and Engineering, Jaypee University of Engineering and Technology, Guna, MP, India

PAPER INFO

Paper history:
Received 25 August 2017
Received in revised form 06 October 2017
Accepted 12 October 2017

Keywords:
Load Balancing
Load Migration
QoS
Delay
Queueing
Cost Optimization and Web Server

ABSTRACT

Numerous works have been done for load balancing of web servers in grid environment. Reason for popularity of grid environment is to allow accessing distributed resources which are located at remote locations. For effective utilization, load must be balanced with all resources. Importance of load balancing is discussed by distinguishing the system between without load balancing and with load balancing. Various performance metrics that needed to be considered for designing an efficient load balancing algorithm are also described. Intensive review of literature of different load balancing approaches for web servers had been carried out and presented in this paper.

***doi**: 10.5829/ije.2018.31.02b.07*

1. INTRODUCTION

The performance of any web servers has been affected by web traffic usually and it makes a slow response of web server due to get overloaded. Each web server faces the problem of overloading and requires solving optimally. There are several load balancing techniques designed by developers for hardware components of web servers, but it is not much effective for common use due to cost limitation of various organizations and companies. Grid computing is the solution for optimizing the cost by making multiple machines that may be in different physical locations; behave like they like they are one large virtual machine. As user's request is increasing day by day, it is necessary to distribute the request among multiple servers / resources on the basis of their capacity. This is not possible without load balancing.

Let's take a scenario where web server is serving without load balancing as shown in Figure 1.

No load balancing simply means there is only one server which is responsible for handling all incoming requests from user.

*Corresponding Author's Email: *anjushukla.iiitb@gmail.com* (A. Shukla)

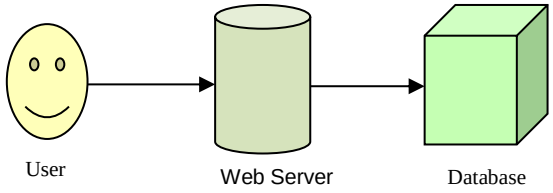

Figure 1. Web Server Functioning with No Load Balancing

Several drawback of this approach are-

- user can't use services while server is offline
- if load on the server increases, it will not able to handle request of multiple users or performance will get slower

In load balancing, load balancer is a module which is responsible for handling the user requests. After receiving request from user, load balancer checks the load on each server and sends user request to server which is lightly loaded. In this way, user feels he is the only one who is receiving service from the server as shown in Figure 2. Load balancing algorithms provide mechanism for effective resource utilization with minimum response time. Load balancer identifies the highly loaded servers and shifts the load to lightly loaded servers for improving the performance.

Please cite this article as: A. Shukla, S. Kumar, H. Singh, Load Balancing Approaches for Web Servers: A Survey of Recent Trends, International Journal of Engineering (IJE), IJE TRANSACTIONS B: Applications Vol. 31, No. 2, (February 2018) 242-248

For efficient use of resources, load balancing algorithms are used by scheduler to manage the incoming load. Impact of load scheduling algorithms and locality scheduling algorithm on web server are analyzed by Lei et al. [1]. Based on comparative analysis, LoadCache_rep algorithm is proposed. The algorithm has better self adaptation and better estimation of the back-end load. Proposed load estimation method shows performance improvement at most 14.7 % in comparison to Locality Aware Request Distribution (LARD) algorithm.

An effective load balance policy must adjust its parameters as the arrival and service characteristics of the incoming workload change. A Self-Adjusting Size-Based (SASB) load balancing policy and its prototype is implemented by Xiong et al. [2]. In SASB, the dispatcher distributed requests according to the request content size and tries to balance the load, among real servers that is measured in occupying resource time.

2. SURVEY OF LOAD BALANCING ALGORITHMS

In this section, survey of various SLB & DLB algorithms is carried out. Based on research findings, DLB algorithms are further classified in following categories- queueing based, dispatcher based, server based, delay based, cost optimization and scheduling based Dynamic Load Balancing.

2. 1. Static Load Balancing Algorithms Round Robin (RR) load balancing algorithm [3] has performance issues in heterogeneous environment where all servers are having different capacity. RR assigns requests in round robin fashion and distributes the load among servers equally without considering serving capacity of server. Low capacity servers don't perform well for the same load. Further improvement is done in RR load balancing algorithm by assigning priority to the servers. For assigning incoming requests, highest priority server is selected instead of serving capacity.

An algorithm is presented for static allocation of requests, in which requests are assigned to servers randomly [4].

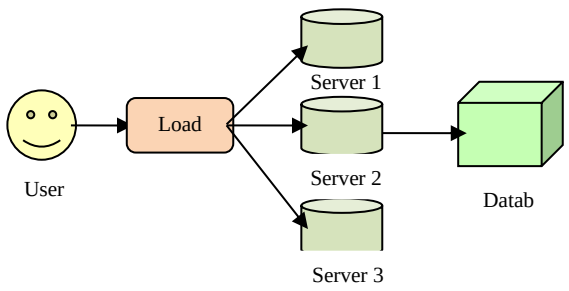

Figure 2. Web Server Functioning with Load Balancing

From implementation point of view, algorithm is easy to implement but due to random distribution of requests, one server may become overloaded or other one's may become underloaded or idle. Various factors that affect the performance of web server are analyzed by Zou [5]. An improved Weighted Round Robin (WRR) scheduling algorithm achieved the web client requests with realistic allocation in order to achieve dynamic load balancing for web server clusters. It is easy and efficient with less system resources, and ensures the forwarding of packets in high-speed with adaptable load distribution [5].

Three task scheduling algorithms RR, Modified Round Robin (MRR) and Time Slice Priority Based RR (TSPBRR) are analyzed by Samal and Mishra [6]. For removing problems of RR, Modified Round Robin came in existence, which has less waiting and response time. In MRR, time quantum is changed at every round. In TSPBRR, firstly, time slice is computed. By using computed time slice, time quantum for each round is computed. Result shows less response time in TSPBRR as compared to other variants of Round Robin algorithm.

Due to rapid changes in distributed environment, static load balancing approaches doesn't have wide scope.

2. 2. Dynamic Load Balancing Algorithm Depending on the scheduling policy and the computational environment and application requirement, various shortcomings are found: (1) Efficiency may not scale as the number of processors increases and (2) High probability that the scheduling is not done by using updated information. In an effort to address these limitations, a distributed load balancing scheme is presented by Riakiotakis [7], in which the scheduling decisions are made by the workers in a distributed fashion and implemented this method along with other two master-worker schemes for three different scientific computational kernels.

2. 2. 1. Web Server Queueing for Dynamic Load Balancing The growth of web traffic is increasing very rapidly that affects the response time of web server. Web users get frustrated due to long waiting of requests and slow response of web servers [8].

A queuing model is used to analyze the performance of web servers in reference [9]. A hypothetical upper bound on the serving capacity of web servers is defined by Elleithy and Komaralingam [9]. This asymptote defined a clear upper bound for the serving capacity of web servers and maximized the capacity boundary particularly for average length of the files served. By restricting the number of simultaneous connections, a web server may keep distance from deadlock situations

that occur whenever server load reaches its maximum capacity [9].

A memory estimation model is developed by Singh and Srivastava [10]. Methodology for calculation of the queue length, waiting time and utilization for predicting the performance metrics like buffer estimation and waiting time are presented [10].

Join Server Queue (JSQ) routing for processor sharing (PS) server farms based on multiple general settings of the applications is analyzed by Gupta [11]. In Single Queue Approximation (SQA) setting, selected queue of the server farm has analyzed based on the process of separation among queues and a state-dependent arrival rate has used to capture the effect of queues [11].

Monitoring of the network traffic based on queuing theory and its simulation is performed in a mixed network environment [12]. The network traffic monitoring is required for the calculation of confidence and efficiency parameters from steady operations of the network. Based on queuing theory and little's law network congestion rate is balanced.

An analysis to obtain the queue length distribution in a fixed range of the buffer system is done by Hernández-Orallo and Vila-Carbo [13]. Histograms used in simple traffic models for analyzing the performance in which the distribution of arrival rate has been stored by Hernández-Orallo and Vila-Carbo [13]. Hurst parameter used for reproduction of second-order. Buffer occupancy distribution has been achieved [13].

A web server gets overloaded when large numbers of requests are received. For measuring the overloading and serving capacity of server, a queuing algorithm is presented in reference [14]. Presented model is compared with RC and QSC algorithms, and provides better result in both homogeneous and heterogeneous environment.

Sometimes a server becomes unavailable for certain reasons like server breakdown, up gradation of server etc. A server may go on vacation when it has no pending request in its queue. This kind of system is called vacation queueing system. An expression for mean waiting time of a vacation queueing system is derived in reference [15]. Many models already exist for vacation queueing system but they don't consider server time out. Server time out also considered and extended for N-Policy scheme where time out is measured [15].

2. 2. 2. Dispatcher Based Dynamic Load Balancing

Dispatching algorithm for web server is presented in reference [16], the server with less capacity always serves fewer requests because it does not have sufficient processing power. Otherwise, the response time may increase rapidly and the drop time may also increase. By using architecture presented in reference [16], remaining capacity of DNS server and MAIL server is used to manage everything and there is no need of any other web server in the load balancing system.

A policy based on the state of pre-emption is presented in reference [17], which performs better than Dynamic and Least Connected (LC) based on the response-time factor. It happened due to information available to the dispatcher. This depicted the execution of LC as a practical policy. Further, it has enhanced as Adaptive LC which improved more based on the response-time factor of LC by adjusting the incoming traffic rates.

A model is developed in reference [18] which is fusion of DNS & Dispatcher based dynamic load balancing approach.CPU utilization is considered as a major metric to select the web server to serve the request. Two tests are performed for performance measurement. First one is without load balancing with one web server and second one is with load balancing in which two web servers are used for load balancing. Result shows improvement in response time, percentage error and throughput.

The dispatching problem in a size and state-aware multi-queue system with Poisson arrivals and queue-specific task sizes is considered in reference [19]. By size and state awareness the dispatcher knows the size of an arriving task and the remaining service times of the tasks in each queue. By queue specific task sizes, the time to process a task depends on the chosen server. The main focus is on minimizing the mean response time by Markov Decision Process (MDP) approach. First, size-aware relative values of states are derived with respect to the response time in an M/G/1 queue operating under various disciplines. For First in First out (FIFO) and Last in First out (LIFO), the size-aware relative values turn out to be insensitive to the form of the task size distribution. The relative values are then accomplished in developing efficient dispatching rules in the strength of the first policy iteration [19].

2. 2. 3. Server based Load Balancing Algorithm

Several approaches exist for web server load balancing system based on the server affecting parameters and content of request types. Several benefits and limitations of these algorithms are analyzed below:

The applications of World Wide Web place extensive performance demands on network servers. The ability to measure the effect of these demands is important for optimizing the various software components that make up a web server. For benchmarking web servers is unable to generate client request rates that increase the ability of the server being tested, even for short periods of time. Moreover, it fails to model important characteristics of the Wide Area Network (WAN) on which most servers are deployed (e.g., delay and packet loss). Various drawbacks are examined by Banga and Druschel [20] when measuring

web server capacity using a synthetic workload. A method is proposed for web traffic generation which can produce traffic with peak loads that exceed the capacity of the server. The results shows that actual server performance can be significantly low than shown by standard benchmarks under conditions of overload and in the presence of wide area network delays and packet losses [20].

Two main aspects based on the performance characteristics of the web server are analyzed [21] and introduced the Gini Performance Coefficient (GPC) as a scale-invariant metric which is used for evaluation of performance regularity. A quantitative benchmark has provided in reference [21] that complements the system capacity metric such as maximum throughput for measuring the system functioning for calculating the values of GPC for several representative systems which were used in the public SPECweb96 analysis.

With Load balancing, Improvement in performance of a parallel and distributed system is done. In past researches, most of the authors have not considered about uncertainty and inconsistency in state information. A new approach is presented using fuzzy logic in reference [22], which gives better response time in comparison to RR and Randomized Algorithm (RA) respectively 30.84% and 45.45%.

Modern web-server systems use various servers to handle an increased user demand. In traditional DNS-based load balancing model, a DNS dispatched requests to web servers based on their load status. In reference [23], Random Early Detection method is presented with the perception that the chance for a web server to become overloaded was directly proportional to its current load in near future [23].

The Join-Idle Queue (JIQ) algorithm consist two load balancing systems: primary and secondary, which communicate through a data structure called I-queue. Together, it serves to decouple the discovery of idle web servers from the process of task assignment. I-queue is a list of a subset of processors that are idle. All processors are accessible from each of the dispatchers [24]. At a task arrival, the dispatcher consults its I-queue. If the I-queue is non-empty, the dispatcher deletes the first inactive processor from the I-queue and directs the task to this idle processor. If the I-queue is empty, the dispatcher directs the task to a randomly chosen processor. The challenge with distributed dispatchers is the uneven distribution of incoming tasks and inactive processors at the dispatchers [24].

An algorithm has suggested [25] for service queue in which load of every web server has computed and balances its load dynamically. There are several load parameters like memory, CPU and network utilization exchange their load value with a centralized system cyclically. In this approach, central node also maintains a waiting queue for each server called as service queue.

Various types of load balancing strategies for web server, their advantages; disadvantages and comparison depending on various parameters are described in reference [26]. An Improved Weighted Round Robin (IWRR) algorithm is presented in reference [27]. Proposed algorithm is compared RR and WRR algorithm. Result shows that IWRR is more suitable for homogeneous or heterogeneous tasks with heterogeneous resources.

2. 2. 4. Dynamic Load Balancing Delay Models

In the process of load balancing, the tasks migrate from one node to another. Due to this task migration, various types of delay have introduced such as communication delay and transfer delay, etc. These delays affect the load balancing models. In this section, various delay models have introduced which are analyzed the effects of delay due to several load balancing techniques used for the web servers.

A delay system is presented in reference [28] for load balancing of cluster of computer nodes used for parallel computations [29]. Stability of the system by using the load balancing algorithms is shown. In order to ensure the performance, a limit has formed in the size of controller gains due to delay in order to ensure performance.

The effect of delays in the exchange of data among nodes by a nonlinear time-delay model and the preventive effects forced on a load balancing strategy are shown in reference [29]. There were two Test-Beds in two different real environments- The first implementation was over a local area network, whereas the second one was over the Planet-Lab and the result drawn from the two Test-Beds were unswerving with each other and high gains were shown to be incompetent [30].

A closed-loop controller has used depending upon the local queue size and tasks calculated for the queue among other nodes. The control system has achieved faster resolution times in contrast to the model proposed by Tang et al. [30]. The time varying delays arising in the closed-loop load balancing process has simulated on the OPNET and has shown good harmony of the nonlinear time delay model with the actual implementation [31].

A polling-based approach is analyzed in reference [31] among the methods applicable to fetching feeds, which is based on a specific schedule for visiting feeds. While the existing polling-based approaches have focused on the allocation of fetching resources to feed in order to either reduce the fetching delay or increase the number of fetched entries, for optimizing both objectives a proposal is present by Singh and Kumar [32].

Delay due to load balancing is analyzed in reference [32] based on factors such as average queue length and

average waiting time of tasks. Ratio Factor Based Delay Model (RFBDM) is presented, which is an improvement of delay model presented in reference [33]. Three cases are considered-zero delay, Deterministic delay and Random delay in heterogeneous environment. In RFBDM average queue length and average waiting time is minimized and functioning of web server is also improved. In this model, fixed delay is considered, so further enhancement is also required.

2. 2. 5. Cost Optimized Dynamic Load Balancing A model based on an incremental tree is presented in reference [34]. In this model, each machine generates a two-level sub-tree communicated to the machine about its computing elements. The root of the tree as a virtual node associated with the machine. The machines of resources have been communicated by these sub-trees. These are combined to develop a three-level sub-tree. Eventually, these sub-trees are linked together to generate another four-level tree called load balancing generic model [34].

A method for reducing cost is presented by Singh and Kumar [35] by selecting the processor which has least cost for executing the job. Processor availability of selected processor is also checked before assigning job to processor. If two jobs selects the same processor, then job which has least cost will execute first, and other job has to wait in queue.

An optimized mechanism for resource allocation is presented by Shukla et al. [36]. Tasks are submitted by user with their workload and each resource has its availability. By checking resource availability, a least cost resource is selected. User submits task in task pool where tasks reside for getting the resource. Scheduler is responsible for task allocation to a resource. However, before assignment of tasks, Resource Cost Monitor (RCM) searches for min cost resource and sends the information to scheduler. After getting information from RCM, scheduler assign task to resource and kept in allocated resource list. The algorithm gives the lower execution cost and average waiting time in compared to existing algorithms [36].

2. 2. 6. Scheduling Based Dynamic Load Balancing Various task based scheduling exists for improving the efficiency of grid environment [37-39]. An approach for selecting the fittest resource is presented in reference [37]. Task scheduler is in-charge of transmitting tasks to resources depending on scheduling algorithm. Resources are categorized into L discrete levels (r1, r2….rl) from smallest to largest. Each task has a resource requirement Ri and a closeness factor ($0<C<1$). Predicted Execution Time (PET) is also computed for a job at each resource.

Two scheduling algorithm to assign suitable resources to a job is presented in reference [40] and performing adaptive load balancing algorithms between clusters. Balance threshold is used to adapt the changes of environment when load changes. Two algorithms differ with each other by mechanism of cluster selection.

A task scheduling algorithm based on budget is presented in reference [41]. Algorithm works in two phases-task selection phase and processor selection phase. For selecting the task, priority is assigned by computing the rank. For processor selection, worthiness of all processor is calculated and selects the processor with highest worthiness value.

3. CONCLUSIONS

This paper presents a detailed survey of load balancing techniques based on various parameters: category, contribution, research gap, future scope. Analysis of various issues is done for different load balancing algorithms (static, dynamic, web server queueing, dispatcher, web server, delay, cost and scheduling based load balancing) from year 1991 to July-2017.

On the basis of survey, a web server system used several resources for applying the load balancing strategy. These resources are getting costlier day-by-day so it is not easy to support by small organization or industry. Therefore, an efficient resource allocation plays an important role in load balancing. An adequate radiance has been thrown to these techniques. Detailed description of existing approaches, strength, limitation, and future scope has been analyzed and presented.

4. FUTURE SCOPE

An efficient resource allocation approach plays an important role in load balancing. It can affect various metrics- cost, waiting time, response time, throughput, penalty ratio and many more. A lot of work exists in this field but it still requires more exploration that has great significance for improving the performance of overall system.

5. REFERENCES

1. Lei, Y., Gong, Y., Zhang, S. and Li, G., "Research on scheduling algorithms in web cluster servers", ***Journal of Computer Science and Technology***, Vol. 18, No. 6, (2003), 703-716.
2. Xiong, Z., Yan, P. and Wang, J., "A self-adjusting size-based load balance policy for web server cluster", in Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on, IEEE., (2005), 368-374.
3. Kumar, B. and Richhariya, V., "Load balancing of web server system using service queue length", ***M. tech Scholar (CSE)***

Bhopal. http://www. ijetae. com/files/Volume4Issue5 /IJETAE_0514_14. pdf Publicerad, Vol. 5, No. 5, (2014).

4. Sharma, S., Singh, S. and Sharma, M., "Performance analysis of load balancing algorithms", ***World Academy of Science, Engineering and Technology***, Vol. 38, No. 3, (2008), 269-272.
5. Zou, S., Analysis and algorithm of load balancing strategy of the web server cluster system, in Communications and information processing. 2012, Springer.699-706.
6. Samal, P. and Mishra, P., "Analysis of variants in round robin algorithms for load balancing in cloud computing", ***International Journal of computer science and Information Technologies***, Vol. 4, No. 3, (2013), 416-419.
7. Riakiotakis, I., Ciorba, F.M., Andronikos, T. and Papakonstantinou, G., "Distributed dynamic load balancing for pipelined computations on heterogeneous systems", ***Parallel Computing***, Vol. 37, No. 10-11, (2011), 713-729.
8. Gilly, K., Juiz, C. and Puigjaner, R., "An up-to-date survey in web load balancing", ***World Wide Web***, Vol. 14, No. 2, (2011), 105-131.
9. Elleithy, K.M. and Komaralingam, A., "Using a queuing model to analyze the performance of web servers", in International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medecine on the Internet, Rome, Italy, January., (2002), 21-27.
10. Singh, L. and Srivastava, R., "Memory estimation of internet server using queuing theory: Comparative study between m/g/1, g/m/1 & g/g/1 queuing model", ***a a***, Vol. 2, (2007), 393-397.
11. Gupta, V., Balter, M.H., Sigman, K. and Whitt, W., "Analysis of join-the-shortest-queue routing for web server farms", ***Performance Evaluation***, Vol. 64, No. 9-12, (2007), 1062-1081.
12. Kamali, S.H., Hedayati, M., Izadi, A.S. and Hoseiny, H.R., "The monitoring of the network traffic based on queuing theory and simulation in heterogeneous network environment", in Computer Technology and Development, 2009. ICCTD'09. International Conference on, IEEE. Vol. 1, (2009), 322-326.
13. Hernández-Orallo, E. and Vila-Carbo, J., "Network queue and loss analysis using histogram-based traffic models", ***Computer Communications***, Vol. 33, No. 2, (2010), 190-201.
14. Singh, H. and Kumar, S., "Wsq: Web server queueing algorithm for dynamic load balancing", ***Wireless Personal Communications***, Vol. 80, No. 1, (2015), 229-245.
15. Ibe, O.C., "M/g/1 vacation queueing systems with server timeout", ***American Journal of Operations Research***, Vol. 5, No. 02, (2015), 77-88.
16. Pao, T.-L. and Chen, J.-B., "The scalability of heterogeneous dispatcher-based web server load balancing architecture", in Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT'06. Seventh International Conference on, IEEE., (2006), 213-216.
17. Ungureanu, V., Melamed, B. and Katehakis, M., "Effective load balancing for cluster-based servers employing job preemption", ***Performance Evaluation***, Vol. 65, No. 8, (2008), 606-622.
18. Singh, H. and Kumar, S., "Dispatcher based dynamic load balancing on web server system", ***International Journal of Grid and Distributed Computing***, Vol. 4, No. 3, (2011), 89-106.
19. Hyytiä, E., Penttinen, A. and Aalto, S., "Size-and state-aware dispatching problem with queue-specific job sizes", ***European Journal of Operational Research***, Vol. 217, No. 2, (2012), 357-370.
20. Banga, G. and Druschel, P., "Measuring the capacity of a web server under realistic loads", ***World Wide Web***, Vol. 2, No. 1-2, (1999), 69-83.
21. Ling, Y., Chen, S. and Lin, X., "On the performance regularity of web servers", ***World Wide Web***, Vol. 7, No. 3, (2004), 241-258.
22. Karimi, A., Zarafshan, F., Jantan, A., Ramli, A.R. and Saripan, M., "A new fuzzy approach for dynamic load balancing algorithm", ***arXiv preprint arXiv:0910.0317***, (2009), 1-5.
23. Yang, C.-C., Chen, C. and Chen, J.-Y., "Random early detection web servers for dynamic load balancing", in Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on, IEEE., (2009), 364-368.
24. Lu, Y., Xie, Q., Kliot, G., Geller, A., Larus, J.R. and Greenberg, A., "Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services", ***Performance Evaluation***, Vol. 68, No. 11, (2011), 1056-1071.
25. Kanagaraj, G., Shanmugasundaram, N. and Prakash, S., "Adaptive load balancing algorithm using service queue", ***Jurnal. ICCSIT***, (2012).
26. Ali, M.F. and Khan, R.Z., "The study on load balancing strategies in distributed computing system", ***International Journal of Computer Science and Engineering Survey***, Vol. 3, No. 2, (2012), 19-30.
27. Devi, D.C. and Uthariaraj, V.R., "Load balancing in cloud computing environment using improved weighted round robin algorithm for nonpreemptive dependent tasks", ***The Scientific World Journal***, Vol. 2016, (2016), http://dx.doi.org/10.1155/2016/3896065.
28. Birdwell, J., Chiasson, J., Abdallah, C., Tang, Z., Alluri, N. and Wang, T., "The effect of time delays in the stability of load balancing algorithms for parallel computations", in Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, IEEE. Vol. 1, (2003), 582-587.
29. Ghanem, J., Abdallah, C., Hayat, M., Dhakal, S., Birdwell, J., Chiasson, J. and Tang, Z., "Implementation of the load balancing algorithm over a local area network and the internet", in Decision and Control, 2004. CDC. 43rd IEEE Conference on, IEEE. Vol. 4, (2004), 4199-4204.
30. Tang, Z., Birdwell, J.D., Chiasson, J., Abdallah, C.T. and Hayat, M.M., "A time delay model for load balancing with processor resource constraints", in Decision and Control, 2004. CDC. 43rd IEEE Conference on, IEEE. Vol. 4, (2004), 4193-4198.
31. Jee, C., Lim, J., Shin, Y., Yang, Y. and Park, J., "A resource allocation policy for delay minimization in fetching capacitated feeds", ***World Wide Web***, Vol. 16, No. 1, (2013), 91-109.
32. Singh, H. and Kumar, S., "Analysis & minimization of the effect of delay on load balancing for efficient web server queueing model", ***International Journal of System Dynamics Applications (IJSDA)***, Vol. 3, No. 4, (2014), 1-16.
33. Birdwell, J.D., Chiasson, J., Tang, Z., Abdallah, C., Hayat, M.M. and Wang, T., Dynamic time delay models for load balancing. Part i: Deterministic models, in Advances in time-delay systems. 2004, Springer.355-370.
34. Yagoubi, B. and Slimani, Y., "Dynamic load balancing strategy for grid computing", ***Transactions on Engineering, Computing and Technology***, Vol. 13, No., (2006), 260-265.
35. Singh, H. and Kumar, S., "Optimized resource allocation mechanism for web server grid", in Electrical Computer and Electronics (UPCON), 2015 IEEE UP Section Conference on, IEEE., (2015), 1-6.
36. Shukla, A., Singh, H. and Kumar, S., "An improved optimized resource allocation mechanism for web server grid", in Parallel, Distributed and Grid Computing (PDGC), 2016 Fourth International Conference on, IEEE., (2016), 438-442.
37. Chang, R.-S., Lin, C.-F. and Chen, J.-J., "Selecting the most fitting resource for task execution", ***Future Generation Computer Systems***, Vol. 27, No. 2, (2011), 227-231.

38. Hu, Z., Mukhin, V., Kornaga, Y., Lavrenko, Y. and Herasymenko, O., "Distributed computer system resources control mechanism based on network-centric approach", ***International Journal of Intelligent Systems and Applications***, Vol. 9, No. 7, (2017), 41-51.
39. Singh, R., "An optimized task duplication based scheduling in parallel system", ***International Journal of Intelligent Systems and Applications***, Vol. 8, No. 8, (2016), 6-37.
40. Lee, Y.-H., Leu, S. and Chang, R.-S., "Improving job scheduling algorithms in a grid environment", ***Future Generation Computer Systems***, Vol. 27, No. 8, (2011), 991-998.
41. Arabnejad, H. and Barbosa, J.G., "A budget constrained scheduling algorithm for workflow applications", ***Journal of Grid Computing***, Vol. 12, No. 4, (2014), 665-679.

Load Balancing Approaches for Web Servers: A Survey of Recent Trends

A. Shukla, S. Kumar, H. Singh

Department of Computer Science and Engineering, Jaypee University of Engineering and Technology, Guna, MP, India

P A P E R I N F O

Paper history:
Received 25 August 2017
Received in revised form 06 October 2017
Accepted 12 October 2017

Keywords:
Load Balancing
Load Migration
QoS
Delay
Queueing
Cost Optimization and Web Server

چکیده

کارهای متعددی برای متعادل کردن بار سرورهای وب در محیط شبکه انجام شده است. دلیل محبوبیت محیط شبکه، اجازه دسترسی به منابع توزیع شده است که در مکان های دور قرار دارند. برای استفاده موثر، بار باید با تمام منابع متعادل شود. اهمیت توازن بار با تشخیص سیستم بین بدون تعادل بار و با تعادل بار بحث شده است. معیارهای عملکرد مختلفی که مورد نیاز برای طراحی یک الگوریتم موثر متعادل کننده بار در نظر گرفته شده است نیز شرح داده شده است. بررسی گسترده ای از ادبیات روش های متعادل کننده بار برای سرورهای وب انجام شده و در این مقاله ارائه شده است.

***doi**: 10.5829/ije.2018.31.02b.07*