ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences &Management Research

Issues and Challenges of Load Balancing in Web Server Clusters: A Survey

Deepti Sharma, Asst. Professor (IT)*
*Department of Computer Science, Jagan Institute of Management Studies, Affiliated to GGSIPU, Rohini, Delhi

ABSTRACT
In recent years, people are shifting from PC's to laptops to smartphones. Services are becoming easy day by day. This will sometimes, leads to poor response time or denial of services from the server. To overcome this situation, multiple server under single name working collectively as a web server cluster.As users are connected in big number and information is shared in big size, the quality of the services needs to be maintained. Web server cluster performance can be enhanced using Load Balancing (LB) mechanism. LB is the process of re-distributing the work load within the web servers or among the web server clusters. LB can be done for both homogeneous and heterogeneous systems. While doing LB in heterogeneous systems, there are various issues and challenges that come across. Thus there is a fundamental need to examine these issues. In this paper, we have examined various issues likes scalability, congestion handling, fault tolerance, performance and scheduling of cluster of web servers.

INTRODUCTION
Due to rapid development, Internet has become one of the most important aspects of life. Over the internet, there are various types of heterogeneous devices with different configuration, operating system and bandwidth. Due to this, complexity of Internet is increasing day by day. The traffic generated over the internet in every single hour is big in number. While accessingInternet, request is generated through client's machine and served by web server or a cluster of web servers. A cluster of web servers can refer to collection of either hardware or the software that helps to deliver web content through internet.

In today's world, response from a web site matters a lot for client. Sometimes, if response from web site is too slow, clients avoid that website and would never visit that website again. With this, response time plays crucial role. In addition to this, a website can be accessed by multiple clients at a time. Thus it might end up with overloading on the webservers or on the web server clusters.

Overloaded requests lead to slow response time of the web servers. To resolve this situation, we need a mechanism that handles the overloaded requests. If there are increase in number of requests that are generated, that mechanism helps in distribution of requests within and between the server clusters. It helps the server to be load balanced. Load balancing mechanism is the answer to all the issues discussed above. The main task of load balancing is to distribute the tasks to the web server's within and between the web server clusters on evenly basis. In computing, LB distributes workloads across multiple computing resources such as computers, a computer cluster, network links, central processing units or disk drivers.

Load balancing are of two types; when we assign the constant workload for the computation to the processor, this is called *static load balancing*. And, when we have variable workload for the computation, and that can be changed during computation that is said to be *dynamic load balancing*.In static load balancing work is distributed statically without giving emphasis on runtime events. This will leads to a stage where it is impossible to judge the work load at the initial stages for the future usage. But, in dynamic load balancing, every time the new workload arrives, the distribution of work load by the master processor is done dynamically. In static load balancing we can have best performance. But in dynamic load balancing, we can use Load Balancing in best way to process work load dynamically.

While doing load balancing, various issues and challenges come across and they must be handled properly for high performance and minimum response time. In this paper, we try to focus on some of the issues that must be taken care of. The issues are congestion handling, scalability, performance, availability, types of request and scheduling.Load balancing helps in ensuring all the issues and takes care of the performance. In the next section, these issues are discussed in detail.

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences &Management Research

ISSUES IN LOAD BALANCING

Following are various issues and challenges while doing load balancing in web server clusters:

Scalability

Scalability can be defined as: a) utilize available processing power and memory on a multiprocessor system fully. b) Ability to balance the load onto servers with disaster recovery from any disaster. c) It can also be defined in terms of the ability to serve large number of clients with sufficient performance and reply the request in minimum time. d) It is the ability to support different types of networks (like wireless, phone line etc) and machines (like laptop, mobile, tablets). e) It can have the ability to support accessing remote applications from device of heterogeneous platform.

A system, whose performance improves after adding hardware, proportionally to the capacity added, is said to be a scalable system[3]. Scalability can be done horizontally or vertically. To scale horizontally (or scale out) means to add more nodes to a system, such as adding a new computer to a distributed software application.[3].Vertical Scalability means to add resources to a single node, *making it more powerful,* in a system, typically involving the addition of CPUs or memory to a single computer [3].

Scalable system should not effect: a) Efficiency b) Reliability c) cost d) performance e) response time f) self-overhead of the system software g) extensibility.

To discuss various scalability issues, we have to focus on the topology and communication mode.
Various topologies can be defined as follows:

a) **The Virtual Cluster Approach:** the virtual cluster master accepts new tasks, and later migrates them to one of the physical clusters. The virtual cluster master tracks the state and location of a migrated task throughout its lifecycle.
b) **The multi-Level Virtual Cluster Approach:** this is an extension of the virtual cluster approach. It is used when the limits of the former do not allow it to scale any further. The idea is adding more hierarchies to the system. But, one of the demerits is that, a complex decision making algorithm is used.
c) **Virtual Machine-Borrow Approach:** in this approach, the base level (local) clustering mechanism remains similar to the earlier approach. Inter-cluster resource sharing happens by virtual transfer of machines between clusters as opposed to transfer jobs between clusters. A machine that has been borrowed will start reporting its availability indicators to the new(borrower) cluster master.
d) **Hierarchical Global Master Approach:** in this approach, machines are pooled to form base clusters, typically each site/project having one base cluster. Each cluster has its own master and makes autonomous load balancing and job scheduling decisions using predefined policies.
e) **Centralized systems:** communication is completely centralized with many clients connecting directly to a single server.
f) **Ring topology:** communication between the servers coordinates the sharing of the system status. It is a physical closed loop consisting of point-to-point links.
g) **Hierarchical topologies:** it has a tree-like structure and provides an extremely fast way of searching through information that is organized in a consistent fashion.
h) **De-centralized systems:** Peers communicates symmetrically and has equal roles and have no single point of control.
i) **Hybrid topologies:** combine multiple topologies into one system.

Based on various topologies discussed above, it can be decided which topology is best suited for our architecture to make it more scalable. Partitioning the cluster system with some topologies (discussed above) is the key for building scalable cluster computing systems. Thus, we can say software topology greatly affect scalability of cluster system. In real world, it is a key problem how to select suitable design pattern of scalable cluster system software.

Congestion Handling

Congestion is a problem that occurs on shared networks when multiple users contend for access to the same resources (bandwidth, buffers, and queues)simultaneously. It is a state occurring in part of a network when the message traffic is so heavy that it slows down network response time. Many web sites today are suffering from

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences &Management Research

severe congestion as thousands of requests arrive at them every second. To deal with congestion problem, various solutions can be considered such as replacing existing machine with a faster model, distribute requests among various web servers or cluster systems.

Congestion is the issue in which, when number of requests for a server exceeds from its maximum limit (depends upon the processing speed and memory) how this situation can be handled by the server. Congestion handling is a mechanism which deals with the issue of congestion. In load balancing, we take care of congestion, when number of total requests exceeds the number of maximum request a server can handle, then on the central server, load balancing mechanism would be implemented.

There are two methods/approaches, which can be used for load balancing. First is, simply reject the request and other one is placing a request in a queue and process these requests one by one. Rejection is not a good option to deal with congestion handling as if the user's request is rejected every time, he may avoid the web site to surf.

In another approach, placing a request in a queue might be a good option. In this, server may take more time in processing but a reply back to the user is mandatory. In this, user has the assurance of getting the resultant output, but with delay.

Thus, Load balancing technique can be used to reduce the traffic congestion and unbalancing of the load over the network. Load balancing scheme can be used to perform:

a) Selecting the path which has less congestion over the network.
b) Minimizing the end-to-end delay.
c) Minimizing the loss of packets over the network due to congestion
d) Performance can be improved, by load balancing

Fault Tolerance

A fault can be stated as the unexpected behaviour shown by the system, can also termed as a malfunction. Faults can be raise due to many factors including hardware failure, software bugs, operator errors, and network problems. Faults can be classified as transient, intermittent or permanent.

A fault tolerant system is one which is able to tolerate the fault at any given instant of time. In other words, "it is the ability of a system to respond gracefully to an unexpected hardware or software failure. The main motive in designing a fault-tolerant system is reliability which can be provided by using multiple instances of CPU's, memories, disks, and power supplies for the single system. If one instance is not working, another continues with the operation.

There are various ways to make a system fault tolerant. Clustering plays an important role in high performance computing. Clusters can be asymmetric or symmetric. Asymmetric cluster is one where a standby server exists, which takes over the primary server at the time of failure. In symmetric cluster, each server acts as a primary server. If one server fails, remaining servers can continue. One approach is proposed in [2], in which three main parts of cluster failure are failure detection, failure recovery and overloaded detection. This approach provides better load balancing along with fault tolerance in asymmetric cluster environment.

Another solution for the fault tolerant problem can be to develop building a web-server architecture with the use of cluster having non-dedicated workstation. This type of servers can fulfil our purpose by assigning workload dynamically to the non-dedicated workstations in response to the variable load burst. By using this scheme/ principle, an approach is designed in [2] which make the ongoing request to transfer between the workstation in the cluster when one station fails or become overloaded with the arriving request.

For building up a high performance reliable and scalable web servers, web server clustering is an important architecture. By using server clustering scheme with fault tolerance capability, a highly reliable web service can be built. The migration of requests between different workstation and recovery can be effectively attained by taking care of the user's request.

Web server performance is a critical issue in building efficient web servers. It becomes more critical with the explosive growth of traffic on the World Wide Web. During peak periods, a Web server might have to service

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences &Management Research

several requests per second. If the server cannot adequately handle the request traffic, the server will fail to satisfy some requests and result in unacceptably slow responses for other requests. There are two types of requests which can be satisfied by Web servers: requests for static files and requests for dynamic pages created by programs executing on the server. The probability of the CPU being a bottleneck increases with the percentage of requests for dynamic pages.When dynamic pages are required, techniques such as fast API's for invoking server programs and caching can be employed to keep the overhead of server programs generating the dynamic pages as low as possible.
Various strategies were used to solve slow response problem.

- Increase server bandwidth: Every time it is not feasible to increase the bandwidth of the web server with increased traffic moreover traffic will not be same all the times.
- Answer only text Request: Request variance cannot be overlooked always, since website offers all kind of pages then web server has to respond back to all the pages.
- Web Proxy caching: is a technique of caching web documents in order to reduce bandwidth usage and server load. The main drawback of this technique is that using stale responses from cache without checking they are changed on server or not.
- Mirror web site: A mirror site is a copy of a website or set of files hosted at a remote location. This option is useful only when live mirror technique is used which automatically updates the mirror copies as soon as the original is changed.
- Monolithic web server: Advance hardware support for web server.
- Cluster web server: A strategy where multiple web servers are used for handling all incoming request.

Among various solutions available to solve this slow response problem cluster web server is the best option. In order to solve such problems (slow response), you have to ensure a solution that can provide maximum availability and scalability by applying a load balancing scheme between geographically distinct web server. The way to handle load dynamically on web server should remain to be really big concern.

In literature, various approaches have been introduced. If resources are used in proper manner, better performance in dynamic load balancing system can be achieved. It has been proved in [3], where a load metric is proposed that contains information about both system load and resource utilization. This metric works fine without prior knowledge of resource requirements. It thus shows that the dynamic load balancing systems provides performance improvement.

A test bed approach is also proposed that is used to evaluate the performance of different load balancing schemes. It gives World Wide Web scenarios and allows variable load generation and performance measurement. Various experiments are also performed using this test bed architecture.

Scheduling
As the Internet is developing rapidly, the Web technologies have been changing people's way of life. But with users increasing, the traditional web system seems to be poor performance. Now the Web Cluster systems based on single system image are widely used to improve the performance.

The performance of a Web Cluster is subject to three main factors: the performance of a single Web server, the routing mechanism and the scheduling strategy. **Scheduling** is the process of deciding how to commit resources between varieties of possible tasks. Before execution, processes need to be scheduled and allocated with resources

Two categories of scheduling algorithms are:

a) Traditional load balancing scheduling algorithms like Random, RR, Select_n, Select_2 and Stat_select_2.
b) Locality scheduling algorithms like Cache_only, Shash, Shash_load, LoadCache_rep and LoadCache

In ***Random* scheduling algorithm**, back-end nodes are chosen randomly without considering anything else when scheduling requests whereas in *Round Robin (RR),* back-ends are chosen in round-robin manner. In *Load scheduling algorithm,* a scheduler considers every back-end node's current load information and always chooses the least loaded back-end node. In *Select_2* algorithm also, current back-ends' information is taken into account but a scheduler first chooses two nodes randomly, and then compares their loads to choose the lighter loaded

ISSN 2349-6193
Impact Factor: 2.805

IJESMR

International Journal OF Engineering Sciences &Management Research

one. Similarly in *Stat_Select_2*: a scheduler first selects two nodes randomly, taking their loads as weights respectively, and chooses the one with lighter load according to relative probability of weights.
*LoadCache_rep*algorithm uses both back-ends' loads and the cache locality and the request is distributed between different nodes if there is overloading of nodes. *Static hash with load (Shash_load)* takes care of load of the request. If at any point of time, load imbalance happens due to the Shash, it will distribute *the load evenly*.

The scheduler is concerned mainly with throughput, latency, response time, turnaround time and waiting time. In distributed computing system, scheduling the tasks is an important issue when there is more number of requests. When the tasks are scheduled efficiently, it increases the performance and efficiency of overall distributed computing system. Scheduling tasks is the problem which depends upon various factors like characteristics of the machines, characteristics of the tasks to be scheduled and the objective function. Other issue can be accessing the resources like channels, processing, etc. that are needed to complete the tasks. Resources in the distributed system can be in limited amount, and can handle limited requests at a time. Another challenge or issue can be how to provide the resources to the whole distributed system?

The biggest challenge in distributed system is to distribute the tasks to the systems. The main problem arises when distribution is done to attain high performance, minimizing execution time, communication delays and maximizing the resource utilization.

CONCLUSION AND FUTURE WORK

This paper has examined issues and challenges in load balancing in heterogeneous web server clusters. These issues are scalability, congestion, fault tolerance, performance and scheduling.

A Scalable system is one whose performance improves after adding hardware, proportionally to the capacity added. Congestion is a problem that occurs on shared networks when multiple users contend for access to the same resources (bandwidth, buffers, and queues) at the same time. A fault can be stated as the unexpected behaviour by the system, can also termed as a malfunction. Faults can be raised due to many factors including hardware failure, software bugs, operator errors, and network problems. A System which can tolerate this fault is known as fault tolerant system. Web server performance is a critical issue for sites which service a high volume of requests. Performance is significantly affected by the percentage of requests f o r dynamic HTML pages; dynamic HTML pages adversely affect server performance. Scheduling is the process of deciding how to commit resources between varieties of possible tasks. Before execution, processes need to be scheduled and allocated with resources. We found that these issues play important role while doing load balancing. These issues can be taken further for doing research in Load Balancing in heterogeneous web servers

REFERENCES

1. *Yousofi Ahmad, Banitabamostafa and YazdanpanahSaeed (2011), A Novel Method for Achieving Load Balancing in Web Clusters Based on Congestion Control and Cost Reduction, IEEE Symopsium on Computers & Informatics, IEEE 2011*
2. *PatilAshwin et al. (2011), Fault Tolerance in Cluster Computing System, International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2011*
3. *Tiwari Ajay and KanungoPriyesh ((2010), Dynamic Load Balancing Algorithm for Scalable Hetrogeneous Web Cluster with Content Awareness, Trendz in Information Sciences & Computing (TISC) IEEE, Dec 2010*
4. *Gondhi Naveen kumar and Dr. Pant Durgesh (2009), An evolutionary Approach for Scalable Load Balancing in Cluster Computing, IEEE International Advance Computing Conference (IACC 2009), March 2009*
5. *TuBibo et al. (2006), Design Patterns of Scalable Cluster System Software, Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, 2006*
6. *Ho Lai Kuen et al (2004), Improving web server Performance by a clustering-based Dynamic Load Balancing Algorithm, Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04), IEEE 2004*

ISSN 2349-6193
Impact Factor: 2.805

IJESMR
International Journal OF Engineering Sciences &Management Research

7. *Haddad Ibrahim and Butler Greg (2004), Experimental Studies of Scalability in Clustered Web Systems, Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04), IEEE 2004*
8. *Ling Yibei, Chen Shigang, Lin Xiaola (2003), Towards Better Peformance Measurement of Web Servers, ICICS-PCM, December 2003*
9. *YingChun LEI et al. (2003), Research on Schedulling Algorithms in web cluster Servers, Journal of Computer. Science & Technology, Vol. 18, No.6, pp.703-716, March 2003*
10. *Hong Jonghyuck and Kim Dongeseung (2002), Hierarchical Cluster for Scalable Web Servers, Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER'01), IEEE 2002*
11. *NimmagaddaSrinivas and HarariIian, Scalaibility issues in Cluster Computing Operating Systems, Intel Corporation Cluster computing, users.crhc.illinois.edu/steve/wcbc99/wcbc-99-nih.pdf*