International Journal of Scientific Research in Computer Science and Engineering
Vol.5, Issue.4, pp.56-59, August (2017)

Review Paper

E-ISSN: 2320-7639

Dynamic Load Balancing Algorithms for Heterogeneous Web Server Clusters

Deepti Sharma[1*], Vijay B. Aggarwal[2]

[1*]Dept. of IT, Jagan Institute of Management Studies, Rohini, India
[2]DIT, JIMS, Rohini, Delhi, India

Corresponding Author: deeptisharma@jimsindia.org

Available online at: www.isroset.org

Received 10th Jun 2017, Revised 16th Jul 2017, Accepted 12th Aug 2017, Online 30th Aug 2017

Abstract Today, the World Wide Web is growing at an increasing rate and occupied a big percentage of the traffic in the Internet. These systems lead to overloaded and congested proxy servers. Load Balancing and Clustering of web servers helps in balancing this load among various web servers. In this paper we have given solution for load balancing Clustering using three different algorithms and frameworks. The objective of this paper is to enhance the execution time of a server, increasing reliability and optimizing the system performance. These are achieved through simulations on the proposed methods.

Keywords: Dynamic Load Balancing, Cluster System, load sharing, web traces

I. INTRODUCTION

In current scenario, Internet is acting as a backbone of communication by offering an efficient and stable network for millions of effected workers round the globe. Among various (email, file sharing, Newsgroup, websites, Instant messaging / chat, fax server, search engine, www) services offered by Internet, www is the most popular. Web is an Internet-based computer network of documents that allows users on one computer to access information stored on another through the world-wide network. To host web pages web servers are required. Web server is a combination of host computer, server software and communication protocol that delivers requested web pages to web clients. In the WWW environment each client request includes a user friendly name (Like www.google.com), which in turns convert into an IP address by DNS through a separate address mapping request. DNS will make use of a list which has an entry of all domain name registered and all IP address corresponding to a single domain name.

This paper is organized as follows: Section II explains problem statement of research work. Section III describes the related work in this area. Section IV illustrates the research objectives. Section V explains proposed framework by defining proposed algorithms. At last, section VI provides future use and is concluded with references in section VII.

II. PROBLEM STATEMENT

With the ever-increasing dependence on the Internet, the traffic on the World Wide Web has increased at an explosive rate causing a rapid increase in the request rate to popular Web sites. When number of requests increases from a particular website, the response time from that website also increases, because of any of the reason:

-Bandwidth of web server
- HTTP Request Type
-Level of traffic volume on the web site.

It is really important to properly handle the slow response time problem to avoid any delay or shutdown or bottleneck.
If there is only one web server responding to all the incoming HTTP requests for your website, the capacity of the web server may not be able to handle high volumes of incoming traffic. In order to achieve web server scalability, more servers need to be added to distribute the load among the group of servers, which is also known as a *server cluster*. The load distribution among these servers is known as **load balancing**.
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. A web cache stores copies of document passing through it, subsequent request of such documents can be satisfied from these cached documents if certain conditions met. The main drawback of this technique is its 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.*

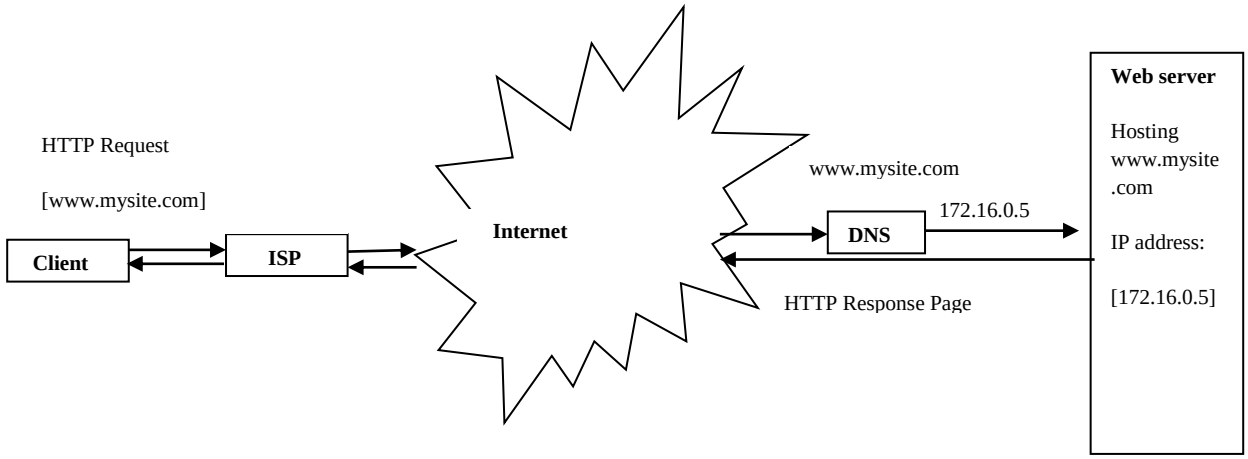

Figure 1. Routing of HTTP request to Web server through ISP and DNS

III. RELATED WORK

Based on above stated problem, the following papers have been reviewed.

[1] have studied various Static(Round Robin and Randomized Algorithms, Central Manager Algorithm, Threshold Algorithm*)* and Dynamic (Central Queue Algorithm, Local Queue Algorithm) Load Balancing Algorithms in distributed system. The performance of these algorithms is measured by following parameters: Overload Rejection, Fault Tolerant, Forecasting Accuracy, Stability, Centralized or Decentralized, Nature of Load Balancing Algorithms, Cooperative, Process Migration and Resource Utilization. Their result shows that static Load balancing algorithms are more stable with such parameters. **[2]** have proposed a DDLB (Dynamic Distributed Load Balancing) scheme for minimizing the average completion time of application running in parallel and improve the utilization of nodes. In proposed scheme, instead of migrating the process for load balancing between clusters, they split the entire process into job and then balance the load. In order to achieve their target they will make use of MA (Mobile agent) to distribute load among nodes in a cluster. **[3]** have worked together to find a load sharing policy in heterogeneous distributed environment where half of the total processor have double the speed of others except only two types of jobs: first class and generic. They have studied only non-preemptive job scheduling policies where scheduler have exact information about queue length of all processor and queuing time of dedicated jobs in fast processor. **[4]** has proposed a genetic algorithm (GA) to dynamically schedule heterogeneous task on heterogeneous system in a distributed environment to minimize total execution time. GA uses historical information to exploit the best solution and completes it process in three steps: Selection, Crossover and Random mutations. The GA algorithm is only performed if there are more unscheduled tasks than processors. **[5]** Wang have designed and implemented a load balancing system based on fuzzy logic and proved that, this algorithm not only effectively reduces the amount of communication messages but also provides considerable improvement in overall performance such as short response times, high throughput, and short turnaround time. **[6]** have proposed a Load Balancing scheme for (SAMR) Structured Adaptive Mesh Refinement Application on distributed systems. In proposed scheme they consider the heterogeneity of processor and dynamic load on system and divide the complete Load balancing process into two phases Global Load Balancing and Local Load Balancing. The size based polices previously used to balance loading in a web cluster framework accomplishes the goal of minimum job response time and job slow down [7] [8] and [16**]**.

IV. RESEARCH OBJECTIVES

A critical challenge today is to process huge data from multiple sources. Users are becoming progressively more dependent on the web for their daily activities such as electronic commerce, reservations and stock trading. Therefore the performance of a web server system plays an important role in success of many internet related companies. The need for a web server cluster system arises from the fact that requests are distributed among these web servers in a proficient manner. For load balancing various factors like I/O overhead, job arrival rate, processing rate may be considered to distribute the jobs to various nodes so as to derive maximum efficiency and minimum wait time for jobs.

Aims and Objectives of Research

- To effectively distribute the shared resources in a cluster framework.
- Distribute the load among the servers in the cluster to provide the best possible response time to the end user by keeping track of the load and available resources on the servers.
- Balances the load on the web servers using its knowledge of the past workload distribution.
- A simple load balancing simulation model is created, under certain assumptions, and to effectively set its interval boundaries according to current workload characteristics is presented.
- To develop a new load balancing policy, this distributes the workloads to the servers by making note of both client activity and system load.
- The objective is to provide a framework which is expected to be more effective, acceptable and robust for load balancing based on server load.
- Further, in order to facilitate the input for clients, proper tools are used that allows input data to be fed easily.
- The proposed framework provides a load balancing approach where outputs are defined as to properly handle all issues and challenges in the model.
- To emphasizes more on high throughput of data on low-latency of job executions, to accomplish huge execution advantages.
- To follow trace driven simulations utilizing both synthetic and real trace to assess the execution of new policy.

V. PROPOSED FRAMEWORK

In this research, we propose three new algorithms

a. Framework to solve load balancing in web servers based on effective parameters.
b. Framework to solve load balancing in web servers based on workload characterization and map reduce approach.
c. A framework based on response time to achieve efficient load balancing in web server clusters.

Also we discuss detailed description of process for model development with input and output parameters, designing of algorithms for performing load balancing in web servers, provision of control charts, templates and other materials. We also discuss comparative analysis of different algorithms with the help of graphical representation. Finally, the conclusion is made through extensive analysis to ensure the distribution of requests among different web servers based on different algorithms and conditions.

We hereby propose three efficient algorithms each of which distributes the request among web server cluster in order to balance the load among these servers.

Algorithm 1: Framework To Solve Load Balancing In Web Servers Based On Effective Parameters

The primary contribution of this algorithm is to propose a framework for running web server cluster system in web environment based on collected requirements and to present its implementation on one of the web services.

The second contribution is to present an experimental analysis of running this framework in web environment to prove the proposed research and to present the evaluation of experiments based on the various parameters such as speed of processing, response time, server utilization and cost efficiency.

For the purpose of validation, different parameters are defined like availability, Health Check, Throughput, Resource Utilization and Efficiency.

Algorithm 2: Framework to Solve Load Balancing in Web Servers Based on Workload Characterization and Map Reduce Approach

The load on the web servers are balanced using the information of the older data. Using actual workload traces a map reduce programming approach is applied to characterize the workload on the basis of key (protocol). A simple load balancing simulation model is created, under certain assumptions, and setting server's parameters according to protocol and threshold value. To forward requests for the same type of content to the same server. It has been proved that directing tasks of similar type to the same servers reduces the slowdown in a web server. A detailed comparison of performance with Adaptload and ACCS approach that aims to balance the load will also be presented.

Algorithm 3: A Framework Based on Response Time to Achieve Efficient Load Balancing in Web Server Clusters

This algorithm distributes traffic based on minimum response time from the servers and minimum pending jobs. The response time is the time interval between sending a request server and receiving the first response from the server. A data structure is maintained to store the number of requests sent to each server and pending jobs on that server.

The model is validated with two algorithms:

- Round Robin Algo – All the requests are sent to each server one after another. So all servers will have equal number of requests.
- Pending Jobs Algo – Selects the server with least response time and least pending requests. Eg, server 1 has less response time and 0 pending requests, so it will be given priority . So each server will handle different number of requests.

Overall results, performance analysis and comparative analysis are also demonstrated. Algorithm has been validated by using different number of requests ranging from 100 to

10000. The output metrics consists of log statements, timestamp for request received and sent, number of requests handled by each web server, time taken by each server for each request, pending requests on each server, transactions per second, number of bytes read and transferred, total processing time to complete all requests.

VI. CONCLUSION AND FUTURE SCOPE

We have proposed three algorithms which are described above for dynamic load balancing in web server clusters. The future work will be to implement these algorithms and to present the experimental results and the analysis of the proposed methods with the existing approach. The experimental analyses will be carried out using various tools like Advanced java programming, Map Reduce Programming (Hadoop), Tomcat, Apache and Ngnix Web Servers, Java Servlets, Jetty embedded Server and SoapUI. Algorithms will be validated on real data set: Live Data from Website http://cricscore-api.appspot.com/ and real traces from http:// ita.ee.lbl.gov.

VII. REFERENCES

[1] S.Sharma Sandeep, S.Singh and M. Sharma,"*Performance Analysis of Load Balancing Algorithms*", World Academy of Science, Engineering and Technology, 2008.

[2] N. Nehra, R.B. Patel, and V.K. Bhat, "*A Framework for Distributed Dynamic Load Balancing in Heterogeneous Cluster*", Journal of Computer Science vol. 3, Issue 1, pp. 14-24, ISSN 1549-3636, 2007.

[3] Helen D. Karatza and Ralph C. Hilzer, "*Load Sharing In Heterogeneous Distributed Systems*", Proceedings of 34th Winter Simulation Conference (WSC'02), ISBN:0-7803-7615-3, 2002.

[4] J. P. Andrew and J. Thomas Naughton, "*Framework for task scheduling in heterogeneous distributed computing using genetic algorithms*", Artificial Intelligence Review, Volume 24, Issue 3, pp 415-429, 2005.

[5] Kapil B. Morey, Sachin B. Jadhav, "*Grid Computing Approach for Dynamic Load Balancing*", International Journal of Computer Sciences and Engineering, Vol.4, Issue.1, pp.40-42, 2016.

[6] Z. Lan, V. E. Taylor and G. Bryan, "*Dynamic Load Balancing of SAMR Applications on Distributed Systems*", Supercomputing, ACM/IEEE Conference, 2001.

[7] F. Bonomi, "*On job assignment for a parallel system of processor sharing queues*", IEEE Transactions on computers, 39(7), 858-869, 1990.

[8] E. Bachmat and H. Sarfati, "*Analysis of SITA policies*", Performance Evaluation, 67(2), 102-120, 2010.

[9] A. Riska et al, "*ADAPTLOAD: effective balancing in clustered web servers under transient load conditions*", In Distributed Computing Systems, Proceedings. 22nd International Conference (pp. 104-111), IEEE, 2002

[10] R.S Sajjan, R.Y. Biradar, "*Load Balancing and its Algorithms in Cloud Computing: A Survey*", International Journal of Computer Sciences and Engineering, Vol.5, Issue.1, pp.95-100, 2017.

[11]Crescenzi, P., Gambosi, G., Nicosia, G., Penna, P., & Unger, W., "*On-line load balancing made simple: Greedy strikes back*", Journal of Discrete Algorithms, Vol 5, Issue 1, pp. 162-175, 2007

[12]Niu, Y., Chen, H., Hsu, F., Wang, Y. M., & Ma, M. *A Quantitative Study of Forum Spamming Using Context-based Analysis*, In NDSS, 2007

[13] A. Garg, "*A Framework to Optimize Load Balancing to Improve the Performance of Distributed Systems*", International Journal of Computer Applications, Volume 122 – No.15, pp. 24-27, July 2015.

[14] I. Psaras and L. Mamatas , "*On demand connectivity sharing: Queuing management and load balancing for user provided networks*", Computer Networks, Vol. 55, Issue 2,pp. 399-414, 2011.

[15] Md. A.M. Ibrahim, "*Cluster of heterogeneous computers: Using mobile agents for improving load balance*", International Journal of Science and Technology Education Research, Vol 1, Issue 7, pp. 143-146, 2010.

[16] Archana B. Saxena, Deepti Sharma, "*Framework for threshold based centralized load balancing policy for heterogeneous systems*", published in an International Research Journal "Oriental Journal of Computer Science and Technology" (OJCST), Vol. 4, Issue 1, pp.99-105 (2011) ISSN: 0974-6471, May 2011.

Authors Profile

Deepti Sharma is an Associate Professor in Department of Computer Science at Jagan Institute of Management Studies, Rohini, Delhi. She is MPhil, MCA and pursuing her PhD in Computer Science from IGNOU (thesis submitted). She has more than 14 years of teaching experience. Her research areas include "Load Balancing in Heterogeneous Web Server Clusters", Big Data Analytics, Distributed Systems and Mobile Banking on which papers have been published in National and International conferences and journals. Various seminars, workshops and AICTE sponsored FDP have been attended.

Vijay B. Aggarwal was awarded Ph.D Degree by University of Illinois in USA in 1973 for his research work in the areas of Super Computers, Array Processors, Cray XMP and Data Base Management Systems. He has been faculty member of Computer Science Deptt at Colorado State University and University of Vermont in USA. Dr. V.B. Aggarwal has been Head & Professor of Computer Science at University of Delhi and Professor at Dept of Electrical Engg and Computer Science at University of Oklahoma, USA. Currently he is Dean (Infotech), DIT, JIMS, Rohini, Delhi. In 2001 Dr. V.B. Aggarwal was elected to the prestigious office of Chairman, Delhi Chapter, Computer Society of India. He has been associated as a computer subject Expert with NCERT, CBSE, AICTE and Sikkim Govt Technical Education Department. Presently he has been nominated as Computer Subject Expert in Academic Council of Guru Govind Singh Indraprastha University in Delhi. Prof. V.B. Aggarwal has authored more than 20 Computer Publications which are very popular among the students of schools, Colleges and Institutes.

