International Journal of Science and Engineering Applications
Volume 9-Issue 12, 133 - 138, 2020, ISSN:- 2319 - 7560

Performance Evaluation of Web Servers using Response Time and Bandwidth

Nnodi Joy Tochukwu
Bishop Okoye Spiritan Secondary School
Mmirinwanyi
Rivers State, Nigeria

Obasi Emmanuela Chinonye Mary
Department of Computer Science and Informatics
Federal University Otuoke, Bayelsa State
Nigeria

Abstract: Computer system performance can be measured using the amount of useful work accomplished by that system when compared to the time and resources utilized. Useful work here means how well the computer is doing the work it is supposed to do. Most websites work under the support of web servers that usually include hardware (for instance CPU, RAM, Disk, and network) as well as software (web services). The alarming growth in web traffic has led to performance problems and has necessitated much research activities as ways to improve web server performance. In this paper, a web server performance evaluation system that evaluates a server based on time of file execution and bandwidth has been designed and developed. Access log data sets from University of Port Harcourt web site were used to study the system and develop software for checking time of server activity in any domain where the application is executed. The evaluation system so designed and developed can be deployed in any server and the values generated can be used in making useful decisions that will mitigate certain occurrences in the future and also offers site owners information that will be useful in forecasting users visiting behaviours and the effects on the server downtime.

Keywords: Bandwidth, performance evaluation, response time, web server, workload.

1. INTRODUCTION

Computer systems are like automobiles that need periodic check to run efficiently and serve user's need. A good performance evaluation of a web server may involve measuring one or more of the following parameters: throughput, bandwidth, response time and so on. As the number of web users are increasing every day, the web servers are required to serve millions of requests per second from multiple users. This requires upgrading of the server in both hardware platform or software architecture. It then becomes very important to evaluate the performance of a web server in order to tackle the bottlenecks and optimize resource usage. Generally, the performance of a web server can be evaluated in measurable terms using one or more of these parameters which include throughput, response time, bandwidth, e.t.c.

A server is a computer system that serves the needs or requests of other systems usually referred to as clients. A **web server** is a computer program that delivers content, such as web pages using the Hypertext Transfer Protocol over the World Wide Web. This means that it delivers HTML documents and any other additional content that may be included in a document, such as images, style sheets, videos and so on. If the performance of a web server is measured by different clients that are remote to the web server, the result may be different depending on the requirements and configurations of the clients. But in this paper, we are specifically concerned with the practical analysis of the activity log files obtained from a university website in Nigeria and how a web site behaves in terms of workload, throughput, response time and bandwidth usage at peak load during a three months period of intensive new students registration and e- learning on the web server. The analysis result was used to design a system that can check the server activity based on bandwidth and response time.

2. RELATED WORKS

Performance Evaluation and measurement of web servers has been the subject of many research. Many works have been done by researchers in order to evaluate the performance of web servers. Early works in the 1990s carried out workload characterization of web servers to obtain the invariants which were used to optimize server performance. Xue et al (2003) measured the performance of web applications and quantified web page attributes that affect response time. They discovered that the most important factor speeding up web page response times was to minimize the number of embedded objects. Caching and also transfer rate were able to improve response time in request bit per second.

Abbas and Kumar (2011) evaluated the performance of web servers as perceived by a client in cases when (i) there was no data flowing between a web server and a client and (ii) there was data flowing from the web server to the client. They focused on three parameters: round trip latencies, access rate and connection throughput.

Manjur (2017) measured the performance of a web server under virtual environment (that is, the web server was hosted on a virtual machine to measure latency). He compared the results obtained with that from a dedicated machine and from the result, it was found that the difference between two sets of results was largely negligible but in some areas one approach performed better than the other.

Nguyan (2017) presented an empirical analysis of several web servers including Apache, NodeJS and NginX to precisely figure out the trade – off between different software designs to tackle the performance bottleneck and resource consuming problems. Their result showed that the performance of the web servers is considerably improved with a good memory allocation.

Adepele et al (2006) evaluated the performance of web servers on the basis of the server load and the network load and it was discovered that to improve the performance of the websites, optimization of the load on the web server was needed which requires capacity planning strategies.

Another researcher (Lind, 2014) evaluated the performance of HTTP web servers on selected hardware platforms for embedded systems using load limits, performance characteristics and system resource usage. A simulated web application was used for the test and a total of five HTTP server software were tested. The overload behavior and efficiency of system resource usage differed greatly between the servers. The test results also showed that the performance varied significantly between HTTP server software running on the same hardware platform, and generally the software with limited feature sets performed best.

Jader et al (2019) reviewed different works that addressed web server performance and load balancing algorithms in the last half decades, to compare their capabilities and provide an efficient platform to build web-based system structures.

Xianghua et al (2013) presented web server performance evaluation model based on response time (MBRT) to evaluate the peak load of a Web server with given configuration. This was based on the special relationship between the response time and throughput when request rate was lower than peak load. They discovered that MBRT was simple when compared with other models and it has also been validated from both theoretical and empirical point of view in real environment.

3: RESEARCH DESIGN

3.1. Design Methodology

The design methodology adopted for the proposed system is the Water-fall Model. Water-fall model is a sequential model that divides software development process into different phases. Each phase is designed for performing specific activity during the Software Development Life Cycle (SDLC) and each phase must be completed before the next phase can begin without overlap between the phases.

3.2 Analysis of the Existing System

The World-Wide Web or the Web is a distributed hypertext-based information system. The client-server model, as it applies to the Web environment, is shown in Figure 3.1. A user accesses documents on the Web through a Web browser such as Google Chrome. The Web browser sends the user request to a Web server, which responds with the requested document. A Web server can respond to requests from multiple clients and this communication between a web server and client is always initiated by the client using Hyper Text Transfer Protocol (HTTP). Communication between a Web client and a Web server is carried out in the following manner: when a client has a request to make of a particular Web server, the client must contact that server. The server listens on a designated port for a request from a web client to establish a TCP connection. Once a TCP connection has been opened and the client has made its request, the server then parses the request and issues a response. The response includes a status code to inform the client if the request was successful or not. If the request is successful, a document is usually returned with the response. If the request is not successful, a reason for the failure is also returned to the client. Once the response is completed, the TCP connection between the Web client and Web server is closed. This process is repeated for each document that a client wishes to retrieve from a web server. The web server keeps activity log of various requests and responses issued at every given time which can be accessed for performance check of that system and for other purposes as the need arises.

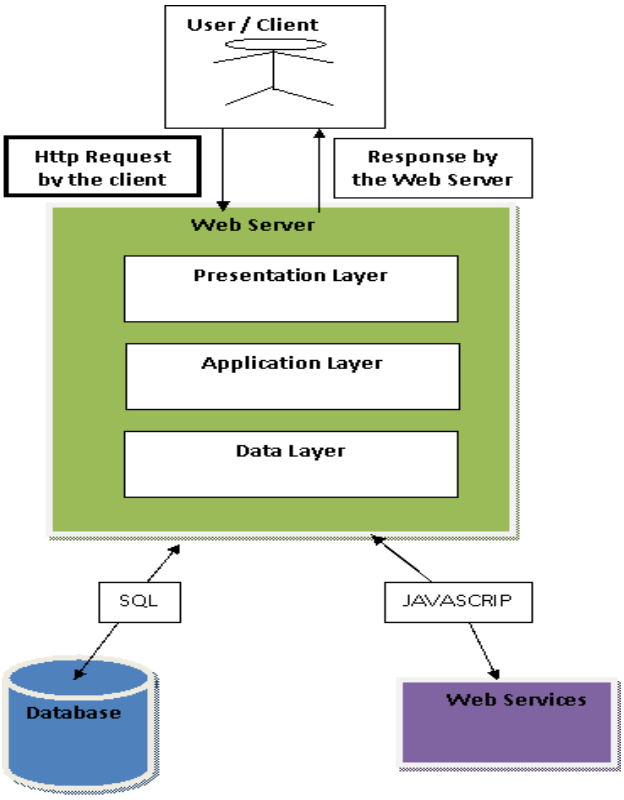

Figure 3.1: Existing System Architecture

3.2.1. Disadvantages of the Existing System

The existing system has the following drawbacks:
1. Web servers face a large number of users.
2. They must provide high availability services with low response time.
3. They must also guarantee a certain level of throughput.
4. The continuous increase of traffic on the web.

3.3 Analysis of the Proposed System

The major function of Web Servers is to provide documents to web clients that request them. In order to improve the performance of Web servers, knowledge of the workloads that these servers are required to handle is needed and also the variables that can affect the performance of these systems are also required. Knowledge of these workloads can be acquired by analyzing logs of web server's activity. These logs are extremely valuable, because they provide a snapshot of actual requests to Web servers. In this research, log files obtained from University of Port Harcourt web server were examined to locate workload invariants. These variants were used to create a workload model of a web server, and also develop a

system that can check server activity in that domain to identify possible performance enhancements for web servers.
In the proposed system, web server evaluator (UPHevaluator) layer is introduced to configure and conduct evaluation test and to also provide real time information about the current test iteration results as shown in figure 3.2. This side is accessible from outside the web server to avoid traffic that may influence the test result.

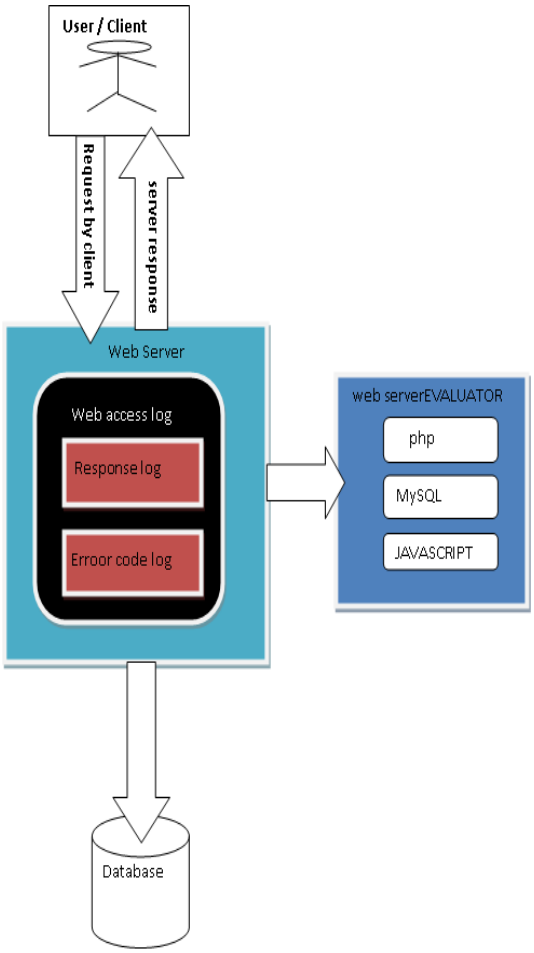

Figure 3.2: Proposed System Architecture

The plot of the raw data is shown in figure 3.3. From the graph in figure 3.3, it can be seen that the highest number of unique visitors, number of visit, pages and hits occurred during the month of July, 2019 while the highest bandwidth occurred in September,2019. It can be inferred that the bandwidth is not directly proportional to the number of unique visitors, their number of visits, hits and pages. The bandwidth depends on other factors not considered in this work.

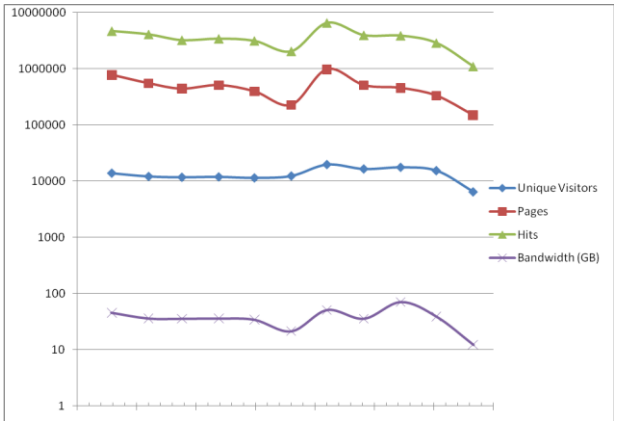

Figure 3.3: Access Log Characteristic Graph

In figure 3.4, the graph of server response in relation to error logs is clearly shown; the highest error is that of partial content. It is much more likely to result from slow internet connection and file sizes of pages on the web server. This condition is understood since many students visited the site from their Phones and PDAs. Document not found error is about 100,000kb (100mb out of 400GB) in bandwidth and 79,565hits (approximately 80,000 hits out of 38,000,000 hits). The graph clearly shows falling of error on bandwidth and hits as other error parameters are evaluated.

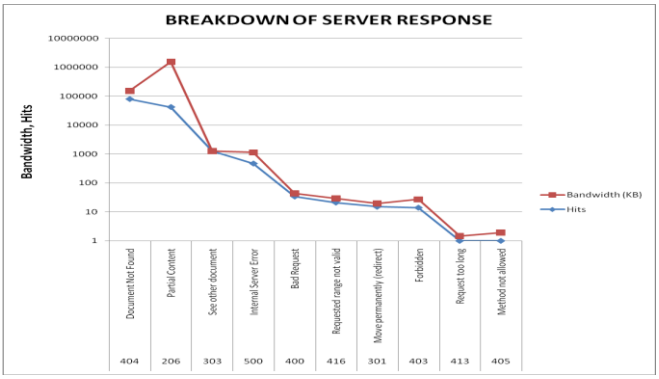

Figure 3.4: Breakdown of Server Response

3.4 Analysis of File Size Distribution

Images get more blurred; video gets less clear and often server variation restrict some of the file format from executing directly in them. For instance some web servers do not allow the execution of tiff image files but jpeg are allowed. Other file types are executed at variance depending on the web server and the application program powering it.
Apache web servers do not allow file name variation involving upper case and lower case mix up but internet information service (IIS) allows mix of lower case and upper case. Server users need to check some of this conditionality in their system.

From the graph of File Type Distribution in Figure 3.5, it is clear that the images generate the highest bandwidth on the

site, followed by Acrobat files and finally java scripts, HTML files and Cascaded Style sheets. This indicates that the higher the file size, the higher the possible bandwidth the file requires.

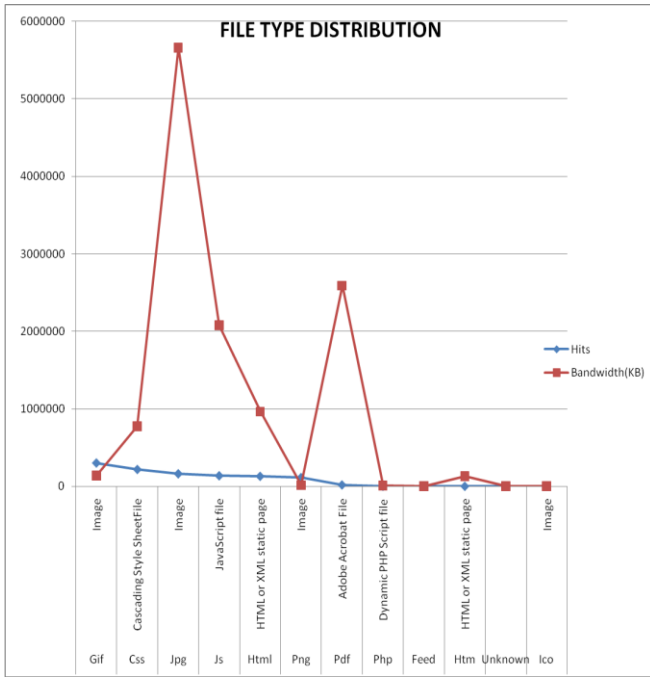

Figure 3.5: File Type Distribution Graph

3.5 Advantages of Proposed System

The proposed system has the following advantages:

1. The performance evaluator system stores all the measurements and parameters for measurement.

2. It provides a real time report on the web server being tested.

3. It evaluates a system using its own workload invariants which helps website owners to make informed decision concerning their own website or system.

3.6 Algorithm of the proposed system

The following steps are taken by the new system to evaluate the performance of web servers:

a. Set the server workload.

b. The parameter is checked based on the solution verified and selected for attention. This will lead to execution of the system process involved in the performance evaluation core using the parameters that have been fetched from the server.

c. Once the parameters are fully collected, the check will lead to decision whether the parameters are okay or not depending on the evaluation benchmark that is guided by what is core in the server.

d. If the parameters are Ok the server performance are evaluated. Otherwise the performance bottlenecks are identified and server tuning is done to enhance the server performance.

e. The server log data are fetched and new parameters generated and used in evaluating server performance.

f. The final performance solution is then presented for verification. If the verification is as expected a performance evaluation must have been concluded.

g. However, if the solution did not take into considerations all the factors required, the lapses are resubmitted for a fresh parameter checks and the cycle continues.

3.6.1 Overall System Flowchart of the Proposed System

The system flowchart of the proposed system is shown in figure 3.6.

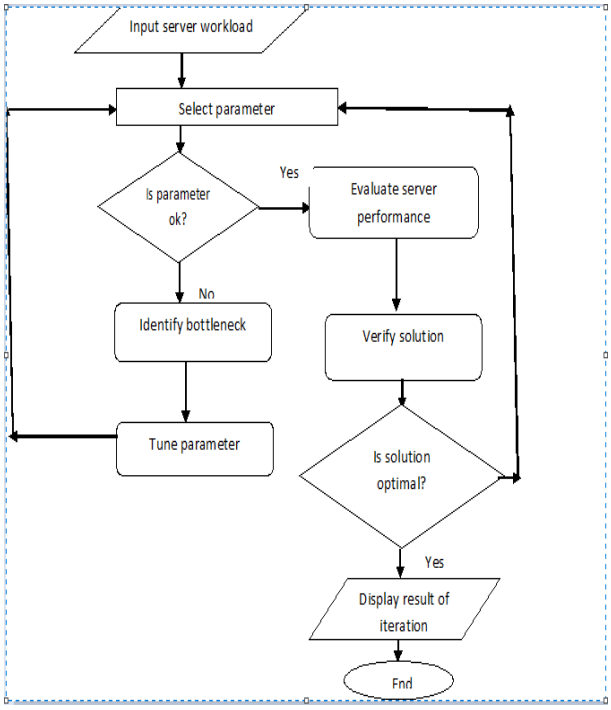

Figure 3.6: Flowchart of the proposed system

4. IMPLEMENTATION AND DISCUSSION OF RESULTS

Implementation process which is the coordination of the web log data collected from university of Port Harcourt web server and programming codes wer`e written to conform to the requirements set out in the system specification. Considering the fact that the project involves the use of interfaces for input of data and display of output, data generated from the web

server for the purpose of the performance evaluation and for referencing the data for output, the system was developed using the following tools and languages. The programming languages selected for the purpose of implementation of the system developed in this project were HTML, PHP, and JavaScript. PHP was selected as the server side logic language while JavaScript was selected as the database server that will be used for the operation. HTML was selected as the language by which the web server and client browsers communicate for encoding information for web documents. **Apache Web Server** was employed to construct the web server and its associated data management system which was used as the relational database management system. Apache-HTTPD was used in this system to capture the timing required for execution of files and passed into the program for use in the computation of the response time required in the evaluation of the performance of the web server. Apache serves the PHP pages to the client browser for the display of the time used in the execution of each of the iterations. JavaScript was used mainly because the function calls provided an iteration that recalls the PHP code automatically whenever it has to recompute the server time. This system consists of three major modules, which are:

1. Parameter capture Module
2. Performance Evaluation Module
3. Result Data Module

Parameters capture Module

The Application module works directly with the PHP. It is an open source application module, which can be easily modified, and it is for capturing of server clock values before and after the execution of a single module in the system program. An application module consists of the response time capture and display offers The Application module works directly with the PHP. It is an open source application module, which can be easily modified, and it is for capturing of server clock values before and after the execution of a single module in the system program. An application module which consists of the response time capture and display offers the user the opportunity to view the activity time on their browser. All required information must be provided and displayed in the browser in order to successfully complete its execution.

Performance Evaluation Module

Flexibility and Availability in this application enables the user to view this interface and interact with this module in two separate interfaces before the access logs from the system can be used in the evaluation of the performance. This module allows the performance evaluator to examine the minimum time interval and the maximum time interval as well as the average time for all the execution done in the system. This generated information can be used to make decision on whether the server is performing well or not as shown in figure 4.1.

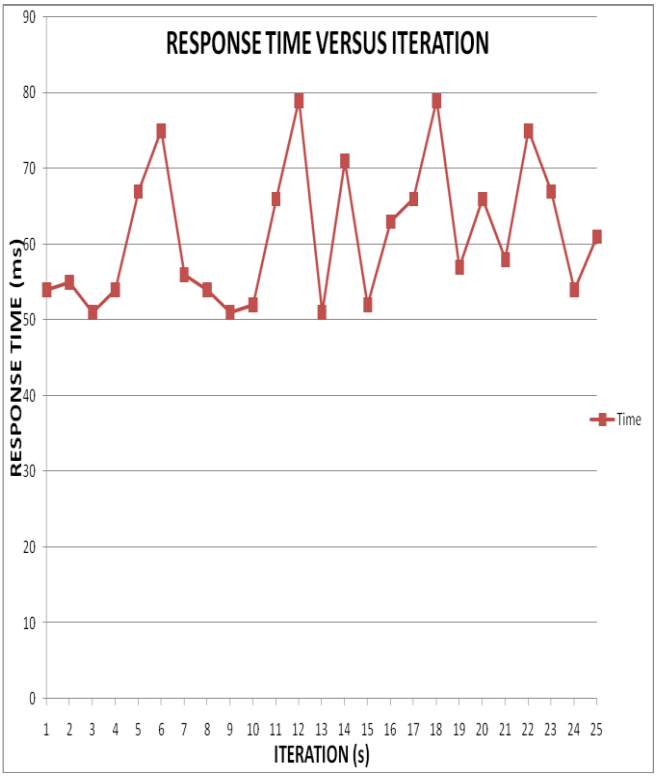

Figure 4.1: Performance Evaluation graph

Result Data Module:

This module is available in the data section of the system on the browser pages. When the system is running on a remote server, the performance result data can be displayed on the browser too showing the result data of the particular remote web server where it has been deployed. After clicking the button, a confirmation message will be displayed and the performance data generated will be displayed

5. CONCLUSION

The project has successfully studied the concepts of web server system and its associated parameter required for the evaluation of the system. We have also been able to develop a system that is useful for the evaluation of the web server using the access log data provided from the University of Port Harcourt web site. The access log takes care of logic and decision for use in the implementation of the service needed for the performance evaluation. It also offers us a more articulated direction on the possible users' behaviours on the site which is necessary to evaluate the system.
From the results obtained, it can be concluded that: the bandwidth size is not entirely a function of number of visitors, their number of visits, the pages visited and the hits, the breakdown of server response negatively imparts on the access log characteristics by reducing the size of the bandwidth and that the response time of the web server depends mostly on the bandwidth as well as the file types and sizes. From the result generated and plotted in the graph, it is clear that the server is performing well based on its response time and the low level of error generated from the access log within the period of investigation. It is also clear that the site designer needs to work on the images and pdf files so that the ratio of image bandwidth to other file size will be minimal. The evaluation system so designed and developed will be

deployed and the values collected used in making useful decisions that will mitigate against certain occurrences in the nearest future. It also offers site owners information that can be used in forecasting users visiting behaviours for instance how the server will fair during students registration or other related concerns.

REFERENCES

[1] Abbas A.M and Kumar R. (2011) A Client Perceived Performance Evaluation of Web Servers. Part II. CCIS, pp. 307-316

[2] Adepele William, Arlitt M., Willams C., Barker K. (2006).Web Workload characterization: Ten Years later. Pp 16-18

[3] Fowler S., Hameseder K and Peterson A. (2012).An empirical Evaluation of Web System Access for Smartphone Clients. A Journal of Networks, 7(11)

[4] Jader O.H, Zeebaree R.M and Zebari R.R. (2019). A State of Art survey For web Server Performance Measurement And Load Balancing Mechanisms. International Journal of Scientific and Technology Research, 8(12).

[5] Lind D. (2014). Performance evaluation of HTTP web servers in embedded systems. KTH Engineering and Management.

[6] Manjur Kolhar. (2017).Web Server Performance Evaluation in a Virtualization Environment. International Journal of Advanced Computer Science Applications 8(20).

[7] Nguyan V.N. (2017).Comparative Performance Evaluation of Web Servers: Journal of Science: Comp. Science and Com. Eng. 31(3).

[8] Xianghua X, Tingting X, Yin Y and Wan J. (2013) Performance Evaluation Model of Web servers Based on Response Time, IEEE Conference Anthology.

[9] Xue Liu, Luisha, Yixin Diao, Steven Froehlidi, Josephl. Hellerstein (2003): Online Response Time Optimization of Apache Web Server. Pp 461-478 or pp 1-18.