

Web Server Analyzer

Prof. A.A Bhosale^{#1} Kaushal Komawar^{#2} Puja Patil^{#3} Dhanashree Jagtap^{#4} Niraja Jahagirdar^{#5} Ankur Bang^{#6}
1amolabhosale@gmail.com, 2kaushalvk2008@gmail.com, 3patilpuja1810@gmail.com,
4dhanashreejagtap1@gmail.com, 5jahagirdarnm@gmail.com, 6ankur.bang@yahoo.com

Department of Computer Engineering,
Vishwakarma Institute of Information Technology, Pune-411048, Maharashtra, India.

Abstract

Web hosting providers claim to provide service for certain parameters such as bandwidth, computation power, speed, disk space etc that may not be fulfilled completely. The end user is unaware of this fact. To solve this ambiguity, the tool: Web Server Analyzer is introduced. The main objectives of this tool include comparison between actual and stated parameters of the service provided by the web hosting providers and analyze them. Analysis will be performed considering the following parameters: stress testing, load testing, scalability, Resource utilization, Bandwidth, Speed, Thread count, Rate of response, CPU utilization and Number of hits per second. A client server application, Web Server Analyzer will generate a report accordingly for the user.

Keywords- Web hosting, web server, analysis.

I. INTRODUCTION

The client server application Web Server Analyzer, is an attempt to compare the actual and stated parameters provided by the service provider so as to generate a report giving the values of the actual parameters at the client side. The report generated will contain comparison and information about the following parameters:

- 1) CPU Utilization
- 2) Number of hits
- 3) Database operations
- 4) File Read Write operation
- 5) Load Testing
- 6) Bandwidth
- 7) Response Time

Web system consists of number of clients, a web server and a network that connects the clients to the server. HTTP is the protocol used to communicate between the client and the server. In order to measure a system's server performance, it is necessary to run some tool on the clients that generates a reproducible HTTP workload on the server. It is generally also desirable that the generated workload mimics real-world behavior. The degree to which real-world behavior is desirable depends on the intent of the test. For example, fine grained performance analysis often benefits from a smaller degree of realism as it can simplify understanding the relationship between cause and effect. In contrast, for benchmarking real-world behavior is important to ensure the results remain meaningful when compared across platforms. The focus of this paper is on the client tool necessary to generate such workloads and httpperf, an implementation of such a tool. This paper does not discuss issues on how to select or generate workloads representative of real-world behavior.

Creating a tool for measuring web server performance is a surprisingly difficult task. There are several factors that cause this, some inherent in the problem and some related to short-comings of current operating systems (OSes). A first difficulty is that a web system is a distributed system and distributed systems are inherently more difficult to measure than centralized systems that have no concurrency and a perfectly synchronized clock.

Second, unlike for other distributed systems, it is often insufficient to model a web system as a closed system with a relatively small user population. While most servers are unlikely to experience severe and prolonged overloads, recent extreme experiences with sites such as the 1996 Olympic web site or the Mars Pathfinder web site illustrate that popular servers are often so popular that they experience load levels that are more accurately modeled by an open system with an essentially infinite user population. For a test tool running on a relatively small number of client machines, this implies that the tool must be able to generate and sustain server overload—not a trivial task as described, for example, by Banga and Druschel.

Third, HTTP in general and HTTP/1.0 in particular cause connection usage patterns that TCP was not designed for. Many of these problems have been fixed in response to experiences gained from running web servers. However, a high performance test tool such as httpperf is likely to push the performance limits of the host OS in different ways than a web server would and, as a consequence, there are a number of issues that such a test tool needs to guard against. A fourth reason that makes writing a test tool difficult is that the web is by no means a static system. Almost every part of a web system—server and client software, network infrastructure, web content, and user behavior—are subject to frequent and sudden changes. For a test tool to remain useful over some period of time requires a design that makes it relatively easy to extend and modify the tool as need arises.

The rest of this paper is organized as follows: the next section gives a brief introduction on how to use httpperf. Section 3 describes the overall design of the tool and presents the rationale for the most important design choices. Section 4 discusses the current state of httpperf and some of the more subtle implementation issues discovered so far. Finally, Section 5 presents some concluding remarks.

II. LITERATURE SURVEY

Now a day's, multiple tools are available for web server analysis which examines the web server based on different parameters.

In this paper [1] they have discussed Greenajax for measuring web application performance. They have compared green ajax with classical ajax application. The green ajax outperforms than the classical ajax. It can decrease the bandwidth consumption to/from web server.

In this paper [2] they have discussed httpperf tool for measuring web server performance. This paper reports on the design and implementation of httpperf, but also discussed some of the more subtle issues that arise when attempting to measure web server performance.

In this paper [3] they have discussed multi-server construction using server virtualization technique. They have constructed several x86_64 servers based on open source server virtualization tools KVM and VirtualBox, and then they analyze their performances using open source analyzing tools ab, httpperf, and siege. Server construction using a server virtualization technique has increased due to its advantages including strengthening security, saving storage and costs, and efficient resource utilization. Particularly when the web server has a severe variation of number of users, server virtualization is used. However, constructing a multi-server with virtualization, compared to a single-server, there can be a drop in performance.

In this paper [4] they have discussed performance analysis of web caching by using cache replacement based on user behavior. The increasing growth of user over world wide web degrades the performance of system by network traffic and server heavy work load. To address this problem web caching is more efficient and scalable scheme to distribute web documents over the web.

III. EXISTING TOOLS

Following are some existing tools to measure web server performance-

1. Httpperf-Httpperf is a tool for measuring web server performance. It provides a flexible facility for generating various HTTP workloads and for measuring server performance.

2. Pylot-Pylot is a free open source tool for testing performance and scalability of web services. It runs HTTP load tests, which are useful for capacity planning, benchmarking, analysis, and system tuning.

3. Tsung-Tsung is a distributed load testing tool. It is protocol independent and can currently be used to stress HTTP, SOAP and Jabber servers.

4. Pylot-Pylot is a free open source tool for testing performance and scalability of web services. It runs HTTP load tests, which are useful for capacity planning, benchmarking, analysis, and system tuning. Pylot generates concurrent load (HTTP Requests), verifies server responses, and produces reports with metrics.

5. http_load-http load runs multiple HTTP fetches in parallel, to test the throughput of a Web server.

6. Iperf-Iperf was developed for measuring maximum TCP and UDP bandwidth performance. Iperf reports bandwidth, delay jitter, datagram loss.

IV. DOMAIN CONCEPTS

PARAMETERS:

1. CPU UTILIZATION-

CPU utilization is computer's usage of processing resources, or the amount of work handled by a CPU. Actual CPU utilization varies depending on the amount and type of managed computing tasks. Certain tasks require heavy CPU time, while others require less because of non-CPU resource requirements..

2. NUMBER OF HITS-

A hit is a request for a file made by a user-agent. User-agents include web browsers and search engine indexing programs, or spiders. Each time a web page is viewed a user-agent request the individual files that make up the page from the computer where the website is stored. A record of the hits received is automatically created and saved as part of monitoring the web server performance. This record is called a web server log.

3. DATABASE OPERATIONS -

Database operations are the various operations performed on the tables of the database. The speed of the web server affected by different joins in a database or different operations being performed is to be calculated.

4. RESPONSE TIME-

Response time is the total amount of time it takes to respond to a request for service. That service can be anything from a memory fetch, to a disk IO, to a complex database query, or loading a full web page. Ignoring transmission time for a moment, the response time is the sum of the service time and wait time.

5. BANDWIDTH-

In computer systems, the internet service providers (ISP) describe the bandwidth as data transfer rate. Basically it is the amount of data that a network can transport from one area to other during a timeframe. Now

adays, most ISPs refer to bandwidth as Kilobits perSecond or kbps, Megabits per Seconds or mbps. In some cases, when more bandwidth is required, some providers will require Gigabits per Seconds or Gbps. A high bandwidth link can transport enough data to display heavy content such as live video event, videostreaming, cloud hosting, popular websites and much more. Following are some quick facts to help you better understand dedicated server bandwidth.

6. LOAD TESTING-

It checks the application's ability to perform under anticipated user loads. It is the process of placing a demand on a computing device or a software system and then measuring its response. Load testing is basically performed to determine system behavior under normal and anticipated peak load conditions. It also helps to identify the maximum capacity of operating of an application and also to determine which element is the cause of degradation. When loads are placed on the system beyond normal usage patterns to test system response at peak it is known as stress testing.

7. FILE READ WRITE OPERATIONS –

File system's performance depends upon factors like allocated block size of a file system, seek time, performance rate of the file system metadata, type of read/write.

It includes operations like file open, read, write, append. The counter is to be kept on the operations to count the number of actions and then calculate the web speed.

V. IMPLEMENTATION DETAIL

1. CPU UTILIZATION -

In PHP, CPU utilization is simply calculated using a function named `getrusage()` that returns the resource usage measures like number of swaps, memory required, number of page faults. Also a function named `memory_get_usage()` is used that returns the amount of memory allocated to PHP. While calculating the CPU utilization we also calculated the time required for executing the functions independently using user time i.e. the CPU time spent executing the user program and `microtime()` that returns the current unix timestamp with microseconds.

2. NUMBER OF HITS-

For calculating number of hits from a client to a server we first need to select and connect to a database. Then create a counter for each page and update the number of hits. If for a page there is no counter then create a new counter. At the end a report is generated that gives the hits for every page individually also the total number of hits. Also we get the visitor information like IP address, user agent and the total number of unique IPs that gives the number of clients.

3. DATABASE OPERATIONS-

Database operations can be calculated by firstly creating database for storing all the requests and responses. A query was fired for data retrieval process and there requests are

stored in the database and the response is received at the client side.

4. RESPONSE TIME-

For calculating response time we have measured the start of every request and the end of every response to those request. We have data for the round trip of what is sent from a browser and how long it takes the target web application to deliver what was needed. We have measured the response time in milliseconds. In PHP, time is calculated by using the function `microtime()` which returns the current timestamp with microseconds.

5. BANDWIDTH-

Bandwidth of web server is the actual amount of data transferred through the web server at one time. We have calculated bandwidth by simply measuring size of website, number of pages or size of data that we have to transfer and number of visitors that we expect. It is simply multiplication of these parameters.

6. LOAD TESTING-

The most basic reason of load testing is to determine the web server's behaviour under both normal and anticipated peak load conditions. For testing load, we started with a small number of users and then incrementally we have increased the load from normal to peak. We have observed how our web server performs during this gradually increasing load condition.

7. FILE READ WRITE OPERATIONS -

The number of times a file is read or written is to be counted. The number of operations affect the speed of the server.

VI. RESULTS

1. CPU UTILIZATION –



2. NUMBER OF HITS-

Hits

Page	Hits
The first page	4
Total Hits	4

Visitors

IP	User agent	Date & Time
116.75.42.199	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:12.0) Gecko	2016/02/17 14:08:32
116.75.14.161	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:12.0) Gecko	2016/02/09 11:32:52
Total unique IP's	2	

3. DATABASE OPERATIONS-

```

Web Server Analyser 1.0 beta
WSA

Performing the testing. Please be patient while the tests are conducted
PHP version: 5.4.45
MySQL version: 5.5.45-cll-lve
Server Software: Apache/2.4.16
Date : February 17, 2016, 5:38 pm

test_1_create_dir ... 0.0002 seconds | test_2_create_tempfile ... 0.0001 seconds | test_3_write ... 0.0076 seconds |
test_4b_append ... 0.0350 seconds | test_5_fileinfo ... 0.0050 seconds | test_6_read_1024 ... 0.1599 seconds | test_
test_6c_read_256 ... 0.1690 seconds | test_6d_read_128 ... 0.3574 seconds | test_6e_read_64 ... 0.4279 seconds |
seconds | test_6g_read_16 ... 1.5418 seconds | test_7_read_8 ... 3.1958 seconds | test_9_readdir ... 0.0159 seconds
test_del_tempdir ... 0.0001 seconds |

Tests : 17
Iterations : 300
Total time : 7 seconds
Score : 3476 (higher is better)
Run again | > Test history
    
```

4.RESPONSE TIME-

5. BANDWIDTH-

```

Transferring 512 KB
-----
Transferred 512 KB in 2.693 seconds, 190.123 KB/s
    
```

6.LOAD TESTING-

```

Web Server Analyser 1.0 beta
WSA

Performing the testing. Please be patient while the tests are conducted
PHP version: 5.4.45
MySQL version: 5.5.45-cll-lve
Server Software: Apache/2.4.16
Date : February 17, 2016, 5:42 pm

test_1_small_page ... 0.0308 seconds | test_1b_small_page ... 0.0288 seconds | test_1c_
seconds | test_1e_small_page ... 0.0288 seconds | test_2_medium_page ... 0.0746 seconds
test_2c_medium_page ... 0.0666 seconds | test_2d_medium_page ... 0.0670 seconds | te
... 0.2403 seconds | test_3b_large_page ... 0.2252 seconds | test_3c_large_page ... 0.248
test_3e_large_page ... 0.2763 seconds | test_4_huge_page ... 0.4945 seconds | test_4b_
seconds | test_4d_huge_page ... 0.4131 seconds | test_4e_huge_page ... 0.3662 seconds

Tests : 20
Iterations : 500
Total time : 4 seconds
Score : 11609 (higher is better)
Run again | > Test history
    
```

7.FILE READ WRITE OPERATIONS -

```
Performing the testing, Please be patient while the tests are conducted
HP version: 5.4.45
MySQL version: 5.5.45-cll-lve
Server Software: Apache/2.4.16
Date: March 6, 2016, 9:22 am

test_1_create_dir ... 0.0001 seconds | test_2_create_tempfile ... 0.0001 seconds | test_3_write ... 0.1
test_4b_append ... 0.0038 seconds | test_5_fileinfo ... 0.0031 seconds | test_6_read_1024 ... 0.1221
test_6c_read_256 ... 0.2231 seconds | test_6d_read_128 ... 0.3224 seconds | test_6e_read_64 ... 0.6
seconds | test_6g_read_16 ... 1.9928 seconds | test_7_read_8 ... 3.2181 seconds | test_9_readdir ...
test_del_tempdir ... 0.0001 seconds |

Tests: 17
Iterations: 300
Total time: 8 seconds
Score: 3173 (higher is better)
Run again | » Test history
```

CONCLUSION

Web Server Analyzer is a client-server application designed to generate a report giving the comparative study between the various parameters. The tool is based on windows, linux. An open source application, this tool is secure and uses http protocol for sending and receiving requests. Also, the comparative study of existing tools conclude that web server analyzer is a better application since it compares and generates report over 7 different parameters.

ACKNOWLEDGEMENT

We wish to express our gratitude to Prof. A.A.Bhosle Department of computer engineering VIIT, Pune for valuable help and guidance in implementation of the product. Also we are thankful to Mr. Kaushal Komawar from hostingrecipe.com for his idea and constant support.

REFERENCES

- [1] “Analysis of Web Application Using Green Ajax” IEEE 2010.
- [2] David Mosberger, Tai Jin “httpperf- A Tool for Measuring Web Server Performance Analysis” from Hewlett Packard, March 1998.
- [3] Sung-Jae Jung, Yu-Mi Bae and Wooyoung Soh “Web Performance Analysis of Open Source Server Virtualization Techniques” International Journal of Multimedia and Ubiquitous Engineering [Vol.6, No.4,] October, 2011.
- [4] Anita Madaan, Niraj Madaan, Poonam Jain “Performance Analysis of Web Caching Through Cache Replacement Based on User Behavior” International Journal of Advanced Research in Computer Science and Software Engineering, Vol.2, Issue 12, December 2012.