

Zachary Jensen

Dr. Clement - CS 360

5 February, 2015

Zack Vs. Apache

ABSTRACT

This report compares the performance of an Apache server to the custom web server that I have created. The purpose behind the report is to better understand how my custom server compares to that of an existing, widely used web server. Testing will be done in order to compare how the two servers' response times compare to the size of the response. The results of these experiments will be used to see what tradeoffs, if any, occur when using my own server as compared to Apache's.

INTRODUCTION

With the amount of traffic occurring over the internet increasing by the second, it is vital to ensure that web servers are running as efficiently as possible, to reduce the amount of time taken to send information across the web. A study Dr. Fiona Fui-Hoon Nah of Missouri University concluded that, on average, a user loses their flow of thought after one second of waiting for a web page to download. The study also showed that after just 10 seconds, most users simply give up on a page and move on¹. This information is extremely valuable to anyone looking to find or keep an audience with users over the Internet, and with money or more on

¹ "A Study of Web Users' Waiting Time", Fui Hoon Nah, Proceedings of the 2000 information resources management association international conference on Challenges of information technology management in the 21st century

the line, there is very little room to error. In order to keep the focus of users, it is essential that web servers are running fast enough to avoid steering users away.

In order to know how well a server is performing, one must compare to the current standard. One of the standards today is Apache. By using apache as a benchmark, I will be able to see how my code compares to that of an internationally used piece of software.

RELATED WORK

After researching related work, a past experiment was found to be very similar to what we will be doing². The study consisted of comparisons between CPU and Memory usage between Node.js and Apache servers on basic files. The results show that, after running many different comparison experiments, Node.js was able to handle more requests faster than the Apache server. In monitoring the CPU and Memory usage, the designer was able to determine that this was a result of an increase in both categories of usage, which he claims to have been expected.

EXPERIMENTAL SETUP

I will be running my experiments on a Macbook Pro running on OSX Yosemite as the operating system. I downloaded Apache's Jmeter³ in order to compare results of web client downloads from both my server as well as Apache's. The original testing plan was to work with JMeter to analyze the response speed of the two servers based on the size of the file requested. However, JMeter would not properly analyze my custom server at first, and as such I was forced

² "Change(b)log: Benchmarking Node.js - Basic Performance Tests against Apache PHP." Change(b)log. January 1, 2013. Accessed February 4, 2015. <http://zgadzaj.com/benchmarking-nodejs-basic-performance-tests-against-apache-php>.

³ <http://jmeter.apache.org/>

⁴ <http://www.hpl.hp.com/research/linux/httpperf/>

to use HTTPerf⁴. I later discovered a small error with my server, and returned to using JMeter as it has a much simpler interface and I was able to export results directly to a spreadsheet. I set JMeter to run with 100 users, with a 10 second ramp-up period. I chose 5 files, ranging from 52 bytes to 1.3 megabytes, in order to catch a range of response times. Each server used the same files in the same location on my hard drive.

RESULTS

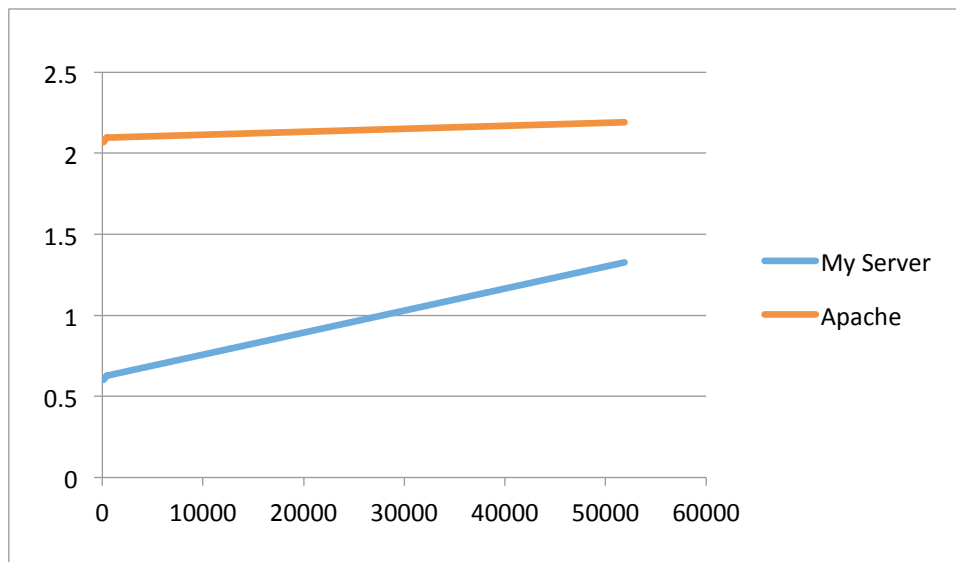
I first ran JMeter on my server to test the response speed's relation to the response size, with the following results:

My Server	
Response Size (bytes)	Response Time (ms)
135	0.602067183
412	0.626556017
51879	1.326666667
559921	962.8449848
1302333	6187.397338

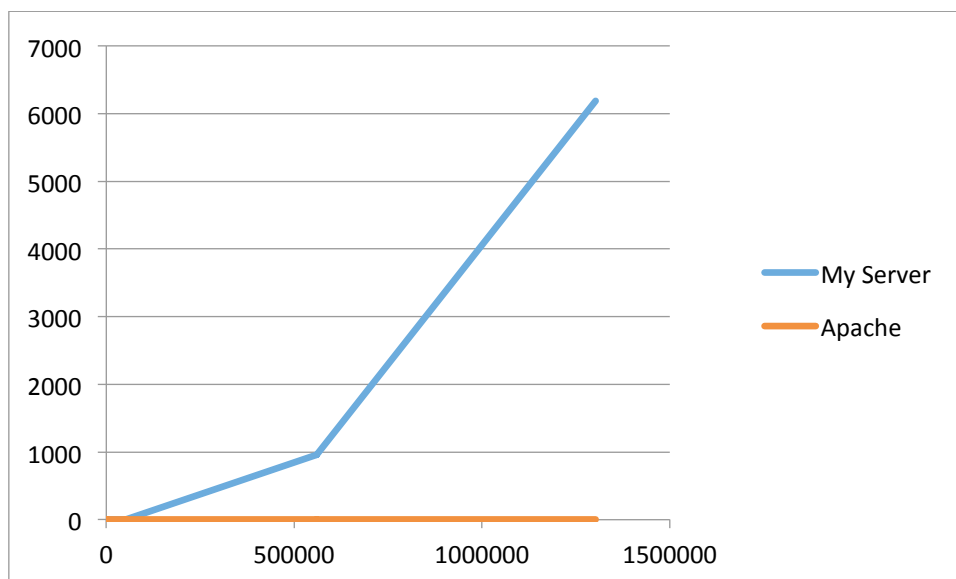
I then ran the Apache server, with the following results:

Apache	
Response Size (bytes)	Response Time (ms)
343	2.0675
621	2.095
52090	2.19
560132	2.47
1302545	4.218

This table compares the response time in milliseconds to the response size in bytes, for the smallest of the three files. Here we see that my server is significantly quicker to respond than Apache.



Next we can see that as file size increases, my server slows down exponentially, whereas the Apache server slows down only by a few milliseconds. This graph shows all 5 files



CONCLUSIONS

While my web server was quicker than apache with small file downloads, as the file size increased my performance decreased greatly. Apache remained consistent as file size increased, showing that it is a better all around server. If a client only needed access to small files, however, it appears that a homemade web server such as my own would be more efficient.