Speed Up Apache Server Performance by Using Heterogeneous Architecture of Content Delivery Network

Rajiv Gupta, Sukhpreet Kaur
M.Tech IT, Head (CSE)
SUSCET Tangori, Mohali,
rajivgupta4010@gmail.com, er.sukhpreetkaur@gmail.com

Abstract: Cloud computing offers better computing through improved utilization and reduced administration and infrastructure costs. Cloud computing is the sum of Software as a Service (SaaS) and Utility Computing. Apache HTTP Server, a modified version of the Apache Software Foundation's HTTP Server that utilizes a pipelined execution of Apache's request cycle. We discuss Apache's original architecture, the modifications necessary for implementation of pipelined execution, and analyze its run time. Ultimately, we hoped to increase throughput of Apache but fall short because of unbalanced request phases and pipelining overhead. This paper discuss the basis for developing advanced and efficient content delivery solutions that are scalable, high performance, and cost-effective. It introduces techniques to enable coordination and cooperation between multiple content delivery services.

Keywords: Apache, xampp, content delivery networks, optimization, client connectivity, client characterization, server adaptation.

1. INTRODUCTION
Apache HTTP server is a software (or program) that runs in the background under an appropriate operating system, which supports multi-tasking, and provides services to other applications that connect to it, such as client web browsers. It was first developed to work with Linux/Unix operating systems, but was later adapted to work under other systems, including Windows and Mac. The Apache binary running under UNIX is called HTTPD (short for HTTP daemon), and under win32 is called Apache.exe.

Web performance has been a key focus of researcher over the last few years. User-perceived latency has a strong bearing on how long users would stay at a Web site and the frequency with which they return to the site. A Web site that is trying to retain users thus has a strong incentive to reduce the time to glass” (the delay between the browser click and the delivery and display of the resource on the user's screen). For Web sites that have a critical need to retain users beyond the first page there is a strong motivation to deliver the content quickly to the user. Given the vagaries of network delays, presence of intermediaries, and the user's network connectivity, the server has a strong incentive to deliver the most suitable (dynamically generated or statically selected) content quickly to the user.

A web server's job is basically to accept requests from clients and send responses to those requests. A web server gets a URL, translates it to a filename (for static requests), and sends that file back over the internet from the local disk, or it translates it to a program name executes it, and then sends the output of that program back over the internet to the requesting party. The word, web server, can refer to the machine itself, or the software that receives requests and sends out responses.

Figure 1: Client Server communication

A. Key Apache performance metrics
If you’re using Apache as your web server, you want to be informed of bottlenecks or performance problems before they manifest themselves as user-facing issues. Some of the most important metrics that will help you gauge your server’s health include the rate of requests, request latency, and error rates.

In general, there are a few types of metrics you will want to monitor:
1. Throughput and latency metrics
2. Resource utilization and activity metrics
3. Host-level resource metrics
4. Errors

B. Cloud Computing
Cloud computing is a general term for the delivery of hosted services over the internet. Cloud computing is the delivery of computing services—servers, storage, databases, networking, software, analytics and more—over the Internet (“the cloud”). Services offered by cloud computing includes online service to send email, edit documents, watch movies or TV, listen to music, play games or store pictures and other files, it is likely that cloud computing is making it all possible behind the scenes Companies offering these computing services are called cloud providers and typically charge for cloud computing services based on usage, similar to how you are billed for water or electricity at home.

A. Types of cloud services
Most cloud computing services fall into three broad categories: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). These are sometimes called the cloud computing stack, because they build on top of one another. Knowing what they are and how they are different makes it easier to accomplish your business goals.

Infrastructure-as-a-service (IAAS)
The most basic category of cloud computing services. With IaaS, you rent IT infrastructure—servers and virtual machines (VMs), storage, networks, operating systems—from a cloud provider on a pay-as-you-go basis.

Platform as a service (PAAS)
Software-as-a-service (SaaS) is a method for delivering needed infrastructure of servers, storage, network and databases for development. SaaS is designed to make it easier for developers to quickly create web or mobile apps, without worrying about setting up or managing the underlying infrastructure of servers, storage, network and databases needed for development.

**Software as a service (SAAS)**

Software-as-a-service (SAAS) is a method for delivering software applications over the Internet, on demand and typically on a subscription basis. With SaaS, cloud providers host and manage the software application and underlying infrastructure and handle any maintenance, like software upgrades and security patching. Users connect to the application over the Internet, usually with a web browser on their phone, tablet or PC.

**C. Load balancing**

A web server has the dispatcher to balance the incoming request to the servers. The main aim of dispatcher in load balancing is to transfer the request to the server that is available at that time. The front end is responsible to balance the requests by making decisions regarding the transfer so that the load is transferred efficiently to the server which can process the request at that moment. Web servers load information is used in making the decisions by the front end. The front end may send a series of the requests to the number of the web servers. In this, the front end distributes the load to the server with least load at that moment. The server and the front end exchanges the information about the load with each other to make the effective decisions. With the help of the correct decision about the load balancing, the service quality is improved and the system becomes more robust. The load balancer decides how to forward the requests and the decision is made correspond to the CPU load percentage on the particular virtual machine.

**D. Content Delivery Network**

A content delivery network (CDN) is a system of distributed servers (network) that deliver pages and other Web content to a user, based on the geographic locations of the user, the origin of the webpage and the content delivery server. This service is effective in speeding the delivery of content of websites with high traffic and websites that have global reach. The closer the CDN server is to the user geographically, the faster the content will be delivered to the user. CDNs also provide protection from large surges in traffic.

**How CDNs Work**

Servers nearest to the website visitor respond to the request. The content delivery network copies the pages of a website to a network of servers that are dispersed at geographically different locations, caching the contents of the page. When a user requests a webpage that is part of a content delivery network, the CDN will redirect the request from the originating site's server to a server in the CDN that is closest to the user and deliver the cached content. CDNs will also communicate with the originating server to deliver any content that has not been previously cached. The process of bouncing through CDNs is nearly transparent to the user. The only way a user would know if a CDN has been accessed is if the delivered URL is different than the URL that has been requested.

**II. PROCEDURE AND EXPERIMENTAL RESULTS**

For the deployment, we are using Digital Ocean services, which provide cloud infrastructure worldwide. Digital Ocean provides developers cloud services that help to deploy and scale applications that run simultaneously on multiple computers. Digital Ocean offers virtual servers (VPS), or “droplets” using Digital Ocean terminology, use KVM as hypervisor.

**Response Time Calculation**

The response time, RT is calculated with the help of the following formula-

\[ RT = Fintm - Arrtm + TDelay \]

Where Arrtm is the arrival time of user request and Fintm is the finish time of user request.

The transmission delay, TDelay is calculated with the help of using the following formulas

\[ TDelay = Tlatency + Ttransfer \]

where, TDelay is the transmission delay T latency is the network latency.

Ttransfer is the time taken to transfer the size of data of a single request from source location to destination. TDelay is considered to be same in every case and hence it is considered as zero.

**III. PERFORMANCE MEASUREMENT**

The experiment is conducted using loadimpact next generation performance testing tool. Loadimpact also provide a k6 extension. Using k6, researchers can run fast, daily load tests, on their own machines (local testing, behind the firewall), as part of their Continuous Integration process. QA testers can run more complex tests in the cloud with Load Impact Cloud Execution. In this test we are using 50VMs for load testing where average Load time is around 455ms and maximum requests performed by server. The experiment is conducted using load impact next generation performance testing tool.
Loadimpact also provide a k6 extension. Using k6, researchers can run fast, daily load tests, on their own machines (local testing, behind the firewall), as part of their Continuous Integration process.

Table 1: Comparison between CDN & Heterogeneous CDN

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Before CDN</th>
<th>After CDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of VMs</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Average load time</td>
<td>455ms</td>
<td>44ms</td>
</tr>
<tr>
<td>Max Request rate</td>
<td>301 ps</td>
<td>631 ps</td>
</tr>
<tr>
<td>Max Response time</td>
<td>11.4 s</td>
<td>1.7 s</td>
</tr>
</tbody>
</table>

In this test we are using 50VMs for load testing where average load time is around 455ms and maximum requests performed by server is

IV. CONCLUSION AND FUTURE SCOPE

This research explored the recent trends in the cloud computing technology. The main issue remained is content delivery, caching and the resource utilization. Therefore, the content delivery should be more dynamic and efficient to improve the performance of the cloud computing technology. In the load balancing mechanism, as described in the thesis, we have to tackle with the situation of efficient loading of the workload. The existing work considered several load distribution techniques that manage the load among various virtual machines and assigns load corresponding to their priority and states. There is an issue of overloading which means the resources may be over utilized and hence there increases the response time. There was also an issue of security and encryption. Content delivery network reduces power consumption if cached content is served during the request so we need to reduce the power consumption if content is delivered directly from the server. The analysis of the results shows that response time of the server is reduced as compared to the other algorithms. A resource allocation policy that takes into consideration resource utilization would lead to a better energy efficiency, as an idle server consume 70% of power with 0% utilization, as per by power model. Hence the proposed work is also energy efficient.

REFERENCES

[2]. Zend PHP. http://php.net/
[3]. CP Congestion Control https://www.net.t-labs.tu-berlin.de/teaching/computer_network/03.07.htm
[5]. Balachander Krishnamurthy AND Craig E. Wills, “Improving Web Performance by Client Characterization Driven Server Adaptation” at AT&T Labs–Research
[8]. I-Hsin Chung, Active harmony, “Towards tomatc Performance Tuning”.
[15]. Shujie Cui, Rizwan Asgha, “Multi-CDN: Towards Privacy in Content Delivery Networks”, 2018