Asian Journal of Applied Science and Technology (AJAST)
Volume 1, Issue 1, Pages 155-157, February 2017

Survey: Deployment of Virtual Cloud Server with Raspberry Pi

S.Adhiyaman[1], P.Anushree[2], P.Priyanka[3] and Mishmala Sushith[4]

[1]*UG Scholar, Department of CSE, KIT-Kalaignarkarunanidhi Institute of Technology. Email: adhiyamaninfotech@gmail.com*
[2]*UG Scholar, Department of CSE, KIT-Kalaignarkarunanidhi Institute of Technology. Email: panutech13@gmail.com*
[3]*UG Scholar, Department of CSE, KIT-Kalaignarkarunanidhi Institute of Technology. Email: priyasept2695@gmail.com*
[4]*Associate Professor & Head, Department of Information Technology, KIT-Kalaignarkarunanidhi Institute of Technology. Email: mishmala@gmail.com*

Article Received: 15 February 2017 *Article Accepted: 26 February 2017* *Article Published: 28 February 2017*

ABSTRACT

This paper presents the design of a new Web server architecture called the Asymmetric Multi-Process Event Driven (AMPED) architecture, and evaluates the performance of an implementation of this architecture, the Flash Web server. The Flash Web server combines the high performance of single-process event-driven servers on cached workloads with the performance of multi-process and multi-threaded servers on disk-bound workloads. Furthermore, the Flash Web server is easily portable since it achieves these results using facilities available in all modern operating systems. The performance of different Web server architectures is evaluated in the context of a single implementation in order to quantify the impact of a server's concurrency architecture on its performance. Furthermore, the performance of Flash is compared with two widely used Web servers, Apache and Zeus. Results indicate that Flash can match or exceed the performance of existing Webservers by up to 50% across a wide range of real workloads. We also present results that show the contribution of various optimizations embedded in Flash.

Keywords: Manufacturing, Cloud computing, Informatics, Computer architecture and Supply chains.

1. INTRODUCTION

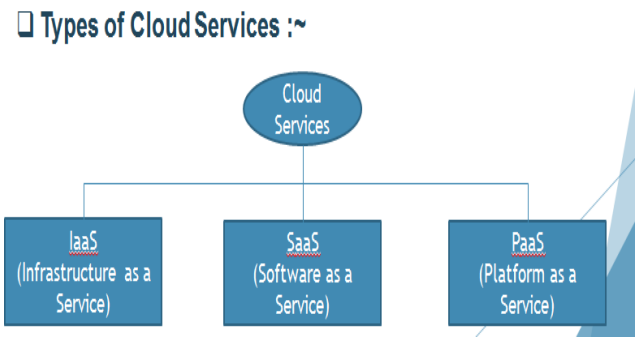
The performance of Web servers plays a key role in satisfying the needs of a large and growing community of Web users. Portable high-performance Web servers reduce the hardware cost of meeting a given service demand and provide the flexibility to change hardware platforms and operating systems based on cost, availability, or performance considerations. Web servers rely on caching of frequently-requested Web content in main memory to achieve throughput rates of thousands of requests per second, despite the long latency of disk operations. Since the data set size of Web workloads typically exceed the capacity of a server's main memory, a high-performance Web server must be structured such that it can overlap the serving of requests for cached content with concurrent disk operations that fetch requested content not currently cached in main memory.

Web servers take different approaches to achieving this concurrency. Servers using a single-process event driven (SPED) architecture can provide excellent performance for cached workloads, where most requested content can be kept in main memory. The Zeus server and the original Harvest/Squid proxy caches employ the SPED architecture1. On workloads that exceed that capacity of the server cache, servers with multi-process (MP) or multi-threaded (MT) architectures usually perform best. Apache, a widely-used Web server, uses the MP architecture on UNIX operating systems and the MT architecture on the Microsoft Windows NT operating system.

This paper presents a new portable Web server architecture, called asymmetric multi-process event-driven (AMPED), and describes an implementation of this architecture, the Flash Web server. Flash nearly matches the performance of SPED servers on cached workloads while simultaneously matching or exceeding the performance of MP and MT servers on disk-intensive workloads.

Moreover, Flash uses only standard APIs and is therefore easily portable. Flash's AMPED architecture behaves like a single process event-driven architecture when requested documents are cached and behaves similar to a multi-process or multi-threaded architecture when requests must be satisfied from disk. We qualitatively and quantitatively compare the AMPED architecture to the SPED, MP, and MT approaches in the context of a single server implementation.

Finally, we experimentally compare the performance of Flash to that of Apache and Zeus on real workloads obtained from server logs, and on two operating systems.



1.1 IaaS

Infrastructure as a Service (IaaS) is a form of cloud computing that provides virtualized computing resources over the Internet. IaaS is one of three main categories of cloud computing services, alongside Software as a Service (SaaS) and Platform as a Service (PaaS).

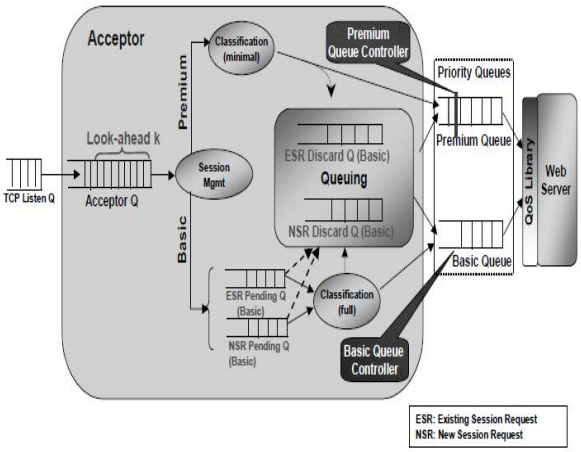
© 2017 AJAST All rights reserved.

2. PURPOSE

In this section, we briefly describe the basic processing steps performed by an HTTP (Web) server. HTTP clients use the TCP transport protocol to contact Web servers and request content. The client opens a TCP connection to the server, and transmits a HTTP request header that specifies the requested content. Static content is stored on the server in the form of disk files. Dynamic content is generated upon request by auxiliary application programs running on the server. Once the server has obtained the requested content, it transmits a HTTP response header followed by the requested data, if applicable, on the client's TCP connection. For clarity, the following discussion focuses on serving HTTP/1.0 requests for static content on a UNIX-like operating system. However, all of the Web server architectures discussed in this paper, fully capable of handling dynamically-generated content. Likewise, the basic steps described below are similar for HTTP/1.1 requests, and for other operating systems, like Windows NT.

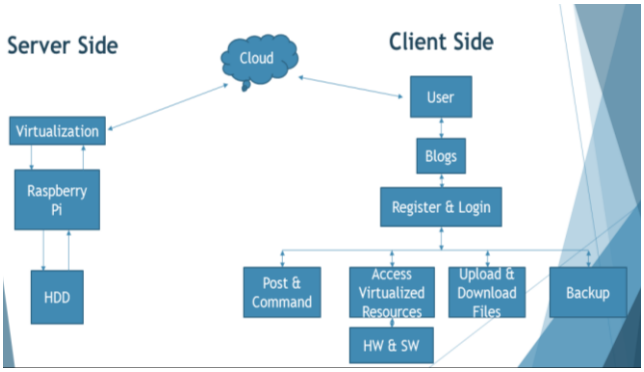
3. SURVEY

While the traffic-aware request processing model dictates how many requests (*m*) should be processed by the acceptor in each cycle, this model does not dictate *which* requests must be processed. When it picks up a request for processing from the acceptor queue, the acceptor has to perform session management in order to determine the class of the request and whether the request pertains to an existing session. In order to ensure that premium requests receive higher priority starting at the earliest point in the Web2K server, the acceptor holds basic requests that it receives in one of two pending request queues. (Two queues are necessary for the acceptor to ensure that requests from existing sessions receive priority over requests from new sessions).



However, when the acceptor detects a premium request, it immediately processes the request, adding the request to the premium queue (as shown in Figure 3) from which the web server receives new requests. Since basic requests cannot be held in the pending queues forever, the acceptor uses a parameter called the *look-ahead* (*k*) to determine when it should process basic requests. The look-ahead represents the maximum number of requests that the acceptor will scan from the acceptor queue looking for premium requests before it decides to process a basic request. When the look-ahead is set to one, the acceptor treats both premium and basic requests in the same manner. The larger the look-ahead value, greater the prioritization provided for premium requests as compared to basic requests.

4. BLOCK DIAGRAM FOR VIRTUAL CLOUD SERVER DEPLOYMENT WITH INTERNET OF THINGS



5. CONCURRENCY MODELS USED BY POPULAR WEB SERVERS

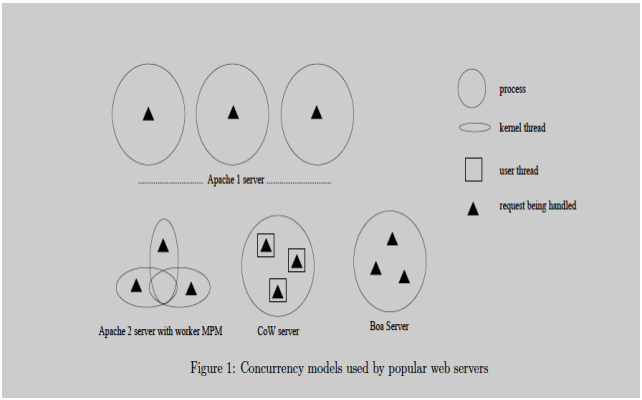


Figure 1: Concurrency models used by popular web servers

6. WHY A PORTABLE VIRTUAL WEB SREVER AND WEB HOSTING

A Personal Web Server or personal server in short, allows users to store, selectively share, or publish information on the web or on a home network. Unlike other types of web servers, a personal web server is owned or controlled by an individual, and operated for the individual's needs, instead of by a company. It can be implemented in different ways:

1) As an Appliance.
2) As a general-purpose server, such as a Linux server; this may be located at the owner's home or in a data center.
3) In a shared hosting model, where several users share one physical server by means of virtualization, or virtual hosting.
4) As one feature of a computer that is otherwise also used for other purposes.

7. CONCLUSION

Thus, the hosting is very easy, because cloud storage can be accessed remotely. Hardware and Software resources can be

© 2017 AJAST All rights reserved.

accessed remotely. Any type of files can be uploaded and retrieved whenever needed.

REFERENCES

[1] Preeti Bhoj, Srinivas Ramanathan, Sharad singhal, "Web2K: Bringing QOS to Web Servers", *Copyright Hewlett-Packard Company*, 2000.

[2] Vivek.S.Pai, Peter Druschel and Willy Zwaenepoel, "Flash: An Efficient and Portable Web Server", *Proceeding of the 1999 USENIX Annual Technical Conference, Monterey.*

[3] Akshat Verma, Puneet Ahuja, and Anindya Neogi, "Cost Aware Application Placement in Virtualized Systems", *LNCS 5346*, pp. 243–264, 2008.

© 2017 AJAST All rights reserved.