A STUDY OF WORKLOAD CHARACTERIZATION IN WEB BENCHMARKING TOOLS FOR WEB SERVER CLUSTERS

Syed Mutahar Aaqib[1], Lalitsen Sharma[2]
[1]Research Scholar, [2]Associate Professor
University of Jammu, India

Abstract: *This paper presents some web server benchmarking techniques and tools that can be used to evaluate the performance of standalone as well as cluster web server systems. Various qualities and properties that should be provided by benchmarking tools are discussed. The focus is on comparing properties of some popular open source benchmarking tools which are readily available. This paper highlights the drawbacks in existing tools with respect to web clusters as currently no tool provides functionality to efficiently evaluate the performance of cluster web server systems.*

Keywords: *Benchmarking, Performance Analysis, Web Server Clusters.*

1. Introduction

The explosive growth of the World Wide Web (WWW) has caused tremendous load on the web content providers. Many researchers have studied the implications of the increased load on the performance of web-based services and suggested ways and means to improve the performance, while others have evaluated the performance and scalability of web servers. In this paper some of web server benchmarking techniques employed to evaluate standalone web server as well as web server clusters are studied

and the functionality provided by various benchmarking tools which implement these techniques are compared. Various benchmarking tools which are publically available and whose source code is freely available are analyzed: httperf [1], WebStone[2], WebBench[3], SURGE[4,5], S-Clients[6], SPECweb2005[7],WebPolygraph[8], TPC-W[9]. Based on the study carried out , it was found out that none of the tool is efficient in evaluating performance of distributed web server clusters, although most of them can be efficiently utilized to test the performance of a standalone web server system. Normally these benchmark tools test the saturation level of the web server by requesting the objects (static and dynamic workload files) as quickly as possible or at a constant/ increasing request rate. Some of the tools which model realistic workloads of user session work for only static workloads. The ever increasing need for dynamic multimedia e-commerce services with subject to security restrictions demands a modern benchmarking tool which should model and examine these complex interactions efficiently and effectively.

2. GOALS OF BENCHMARKING STUDY

While evaluating the performance of a web systems, factors that can influence the results should be taken into account: web server software, operating system, hardware configuration, network environment. In a web server cluster, this environment is more complicated by the presence of multiple other components. Thus, a complete performance evaluation of all layers and components involved in a web server cluster is too difficult if not impossible. This study thus focuses mainly on benchmarking tools used in the design and prototype phase when different architectures must be evaluated and alternative solutions must be compared. This study is thus based on comparing the properties of various benchmarking techniques and tools and does not consider factors like network/ hardware and operating system that in most cases are implied.

3. TECHNIQUES OF WORKLOAD GENERATION FOR WEB SERVER CLUSTERS

The characterization of the workload generated by the web benchmarking tool is one of the distinguishing features among all the tools as it represents

a prime aspect of benchmarking and tries to model the real-world traffic pattern observed by the web server systems.

Closed v/s Open loop Model

In an open loop model, a number of clients issue requests to retrieve workload objects characterized by periodic client inter-arrival times. Clients keep sending the requests without waiting for the response from the server. This situation provides a realistic view of the offered workload. In closed model, a pre-determined number of clients send requests to retrieve workload objects only after receiving previous server responses. This model becomes unrealistic and is not acceptable for a cluster web server system. Closed loop model prevents concurrent transfer of responses from the server. An open loop model thus is a viable solution when modeling a client behavior especially in case of Clustered web server systems.

There are several possibilities to generate the web requests which normally fall in four categories.

i) Trace based techniques.
ii) File-list based techniques.
iii) Analytical distribution based technique.
iv) Hybrid techniques.

i) Trace based techniques.

Trace-based approach [10, 11], allows a benchmark tool to reproduce the user behavior in a realistic manner. The web workload is based on pre-generated trace logs derived by server access logs [12]. The workload characteristics can be mimicked by replaying the requests as logged in the trace. An alternative is to create an abstract model of a web site and extract session-oriented high level information through trace analysis [13]. However the validity of the representativeness of a trace recorded is questionable issues as it represent state of the system under specific conditions and thus is general not valid in all environments and under futuristic varying workload demands that may arise in near future .

As the trace-based approach requires access to web server access logs files which are normally of high business value, companies and organizations are reluctant to hand over their traces for free (or even at all).

Also to reconstruct the user session from trace logs can be complicated with the presence of a proxy server through which all the requests may be arriving which may lead to incorrect characterization of workload[12].

ii) File-list based technique.

In file-list based technique, the next request for the object to be retrieved is chosen on the basis of its access frequency specified in a file with a predetermined inter-arrival request time. This file lists the web objects to be retrieved along with their access frequencies which are determined by the analysis of the trace logs. This technique generates non-realistic workload and thus lacks flexibility. Tools based on this technique are not able to model bursty, session oriented web traffic.[14,15,5]. As this techniques fails to provide support for modeling the session-oriented nature of the web traffic, its not suitable for evaluation involving realistic workloads unless some support to define the characteristics of a user session is provided. Also the size of the fileset should be examined to ensure the server caching is fully utilized,

iii) Analytical distribution based technique.

Analytical distribution based techniques employs mathematical models for specification web workload distribution. The next request for the object is generated according to the parameters of this model. Random values based on probabilistic distribution are generated to mimic all the characteristics of request stream. Alternately, all the session based information and resulting sequence of requests and generated beforehand and stored in a trace file to be later used by the workload generator. The workload model can be used to evaluate performance under different environment/conditions by simply changing the parameters of the mathematical model.

iv) Hybrid technique.

Another technique combines the features of the file-list and analytical approaches. This techniques employs stochastic models to characterize the session oriented workloads. The objects to be retrieved are specified using a file-list while as session oriented parameters are modeled through analytical approaches.

The emulated clients be that open or closed loop model, should be capable of supporting for various requests mechanisms (GET, POST, and HEAD) and provide the functionality of HTTP/1.0 and HTTP/ 1.1 while issuing requests for objects specified in the workload distribution. HTTP/1.1 provides features like persistent connections, pipelining and encoding[12] which have a significant impact on the performance of the web server system[5,19]. Furthermore, support for session tracking via cookies and to emulate security mechanisms of HTTPS/SSL encryption should be provided in the client emulators.

Also, to reduce the latency time experienced by the users, the emulated client should be allowed to use issue requests for embedded objects using multiple parallel connections.

4. Comparison Results of Web Benchmarking Tools

In this section, the manner in which the selected web benchmarks specify their workloads is studied and analyzed in detail. In order to evaluate different test configuration and scenarios most of the benchmarks under study can be customize, tailored and extended to fit the workload model. Benchmarks like SPECweb2005 and TCP-W don't provide much functionality to modify the configuration of the workload model, because of the standardized goals of these benchmarks.

Httperf [1] is based on an open model. It allows two approaches to issue requests: hybrid and trace-driven [10]. Both of these approaches allow request generation for both static and dynamic objects as well as session-oriented workload characterization. Hybrid approach involves specifying requested URLs along with session oriented values such as user think times. In case of trace-driven approach, user sessions are defined in a trace file. Httperf provides support for HTTP/1.0, HTTP/1.1 and SSL protocols. It also allows support for session reuse and multiple concurrent connections.

WebStone[2] is file-list based benchmark following a closed loop model where request streams are specified in a filelist. It allows generation of requests for both static and dynamic content. However since the maximum size of filelist is 100 files, It's difficult to model workloads of cluster based

web server systems which consist of thousands of files. Also there is no provision to specify session-oriented workload. WebStone supports only HTTP/1.0 without keep-alive, also there's no support for encryption and authentication.

WebBench [3] is based on a hybrid approach and follows a closed loop model. It provides support for request workload characterization for static, dynamic objects. Support for HTTP/1.0 and HTTP/1.1 is also provided . However there's no provision to specify session oriented information of requests.

S-Clients[6] is a non-realistic web benchmarking stress tool. It involves a single file which is requested after a specified time period [6]. This benchmark can well be used to measure the peak performance of a web server but it does not exercise load on the file system as workload comprises of a single file which is always fetched from the cache. Also, there's no support for encryption or session oriented behavior as it supports only HTTP/ 1.0.

SURGE[4] is a closed loop analytical distribution based web benchmarking tool which derives empirical analysis of web server usage to reproduce real-world characteristics[4,5]. In SURGE , a single process executes in an infinite loop, alternating between requests and thinking times producing bursty traffic. It provides support for HTTP/1.0 and HTTP/ 1.1 protocols whereas no provision is there for any security protocol. Also, only one connection can be active at a time. The basic limitation of SURGE is that it can handle only static content and thus is not usable in case of multi-tier cluster web server systems.

Web polygraph benchmark allows complete specification of web workload along with session-oriented request stream. It can issue requests in both open and/or closed loop model. Probability distribution can be used to specify web pages, popularity of files, cacheability at the client , server delays due to network congestion[8]. It also supports both HTTP/1.0 and HTTP/1.1 protocols. Web polygraph is suitable for Web server cluster systems as web workloads are already configured to layer-4 and layer-7 web clusters.

SPECweb2005 and TPC-W benchmarks defines only standardized web workloads which cannot be customized or modified by the users. They cannot be thus used to model workloads for different categories of web sites. The basic workload of SPECweb2005 comprises of both static and dynamic content along with support for secure services. SPECweb2005 follows a closed model because of the fixed number of clients are executed during each experiment.

The TPC-W benchmark specification defines the details of the web services and content at the site and the workload offered by clients[16,9]. It also specifies a database schema oriented to e-commerce transactions for an online bookstore together with its web interface. TPC-W defines web interations that are web page traversals which form particular actions such as browsing , searching and ordering. Request streams are session oriented, with think times separating web page retrievals. It also supports secure connections because some clients actions like (online buying) require SSL/TLS encryption.

5. Conclusion

In this paper, various benchmarking techniques and tools that are used to evaluate the performance of standalone as well as cluster web server systems are studied. It is found that although most of the web benchmark tools are effective when employed to analyze a standalone web server system, none of these benchmarks address all issues related to the analysis of cluster web server systems. Many popular tools, such as SURGE and WebStone do not support dynamic requests and other security protocols. Also, many of the benchmarks can't sustain realistic web workloads and difficulty in emulating realistic dynamic and secure web workloads. Hence it is concluded that there is still a lot of room for further research and implementation in developing web benchmarks for cluster web server systems.

6. Acknowledgements

The authors are thankful to Prof. Devanand, Head, Department of Computer Science and IT, University of Jammu, for his kind support.

7. References

[1] D. Mosberger & T.Jin . httperf- A tool for measuring web server performance. ACM Performance Evaluation Review , 26(3):31-37, Dec. 1998

[2] Mindcraft. WebStone. http://www.mindcraft.com/webstone/.

[3] Ziff Davis Media. WebBench.

http://www.etestinglabs.com/benchmarks/webbenchwebbench.asp

[4] P. Barford &M.E. Crovella. Generating representative Web workloads for network and server performance evaluation. In proc. of ACM performance 1998, 151-160, July 1998.

[5] P. Barford & M.E. Crovella. A performance evaluation of Hyper Text Transfer Protocols. In Proc. of ACM sigmetrics 1999, 188-197, May 1999.

[6] G. Banga & P.Druschel. Measuring the capacity of a web server under realistic loads. World Wide Web, 2(1-2):69:89, May 1999.

[7] Standard Performance Evaluation Corp. SPECweb2005,

http://www.spec.org/osg/web2005/

[8] Web Polygraph. http://www.web-polygraph.org/

[9] Transaction Procession Performance Council. TPC-W.

http://www.tpc.org/tpcw/

[10] M. Arlitt. Characterizing Web user sessions. ACM Performance Evaluation Review, 28(2):50-63, Sept 2000

[11] D.A. Menasce & V.A.F. Almeida. Scaling for E-business. Technologies, Models, Performance and Capacity planning. Prentice Hall, NJ, 2000.

[12] B. Krishnamurthy & J. Rexford. Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement. Addison- Wesley, MA, 2001

[13] S. Manley, M. Seltzer & M. Courage. A Self-scaling and self-configuring benchmark for web servers. In proc. of ACM Sigmetrics 1998 Conf, 170-171, June 1998.

[14] M.Arlitt & C.Williamson. Internet Web servers: Workload Characterization and performance implications. IEEE/ACM Trans. on networking, 5(5):631-645, Oct, 1997.

[15] M. F. Arlitt & T.Jin. A workload characterization study of the 1998 world cup web site. IEEE Network, 14(3):30-37, May 2000

[16] D. A. Menasce. TPC-W: A benchmark for e-commerce. IEEE Internet Computing, 6(3): 83-87, May 2002