Web Server Performance Analysis

Jijun Lu and Swapna S. Gokhale
Department of Computer Science and Engineering
University of Connecticut, Storrs, CT 06269, USA
{jijun.lu, ssg}@engr.uconn.edu

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems

General Terms

Performance, Measurement

Keywords

Queuing model, $M/G/m$, Web server performance

1. INTRODUCTION AND MOTIVATION

The Web Wide Web (WWW) has experienced an exponential growth during the last ten years and in the current era Web servers represent one of the most important sources of information and services. Web servers, which are typically based on the HTTP protocol running over TCP/IP, are expected to serve millions of transaction requests per day at an acceptable level of performance. Due to the stringent performance expectations imposed on a Web server, it is important to analyze its performance for different levels of load prior to deployment. Model-based analysis, which consists of capturing the relevant aspects of a Web server into an appropriate model, validating the model and then using the validated model to predict the performance for different settings and load levels can be used for this purpose.

A number of research efforts have focused on performance modeling and analysis of Web servers. Slothouber [14] model a Web server as an open queuing network. Van der Mei *et al.* [16] present an end-to-end queuing model for the performance analysis of a Web server, encompassing the impact of client workload characteristics, server hardware/software configurations, communication protocols and interconnect topologies. Cao *et al.* [2] use an $M/G/1/K*PS$ queuing model to model the performance of a Web server. Nossenson *et al.* [10] introduce a new $N-Burst/G/1$ queuing model with heavy-tailed service time distribution for Web server performance modeling. Squillante *et al.* [15] employ a $G/G/1$ queue to model high-volume Web sites. Kant *et al.* [6] describe a queuing network model for a multiprocessor system running a static Web workload. The model is based on detailed measurements from a baseline system and a few of its variants.

Most of the previous efforts in Web server performance analysis use a *single* server queuing model. Modern Web servers invariably process multiple requests concurrently to fulfill the workload demands placed on them. Concurrency can be achieved using a process-based, a thread-based or a hybrid architecture [9]. Thus, a Web server typically has multiple processes, or multiple threads or multiple threads within multiple processes. Each thread or process works independently and multiple threads or processes can work simultaneously to service multiple client requests at the same time. Thus considering the concurrent processing capability of modern Web servers, it is appropriate to consider this system as a multi-server system.

In this paper we use a $M/G/m$ queuing model, which consists of multiple nodes/servers to model the performance of a Web server. The performance metric we consider is the response time of a client request. Since there is no known analytically or computationally tractable method for obtaining an exact solution for the response time of a $M/G/m$ queue, we use an approximation proposed by Sakasegawa [13]. We validate the model for deterministic and heavy-tailed workloads using experimentation. Our results indicate that the $M/G/m$ queue provides a reasonable estimate of the response time of a Web server when the traffic intensity is moderately high. The conceptually simplicity of the model combined with the fact that it needs the estimation of very few parameters makes it easy to apply.

2. PERFORMANCE MODEL

Modern Web servers implement concurrent processing capability using either a thread-based, a process-based or a hybrid approach [9]. The thread-based and the process-based architectures offer distinct performance, stability and reliability tradeoffs, while the hybrid approach exploits the advantages of both and mitigates their drawbacks.

In both the thread- and the process-based architectures, to avoid the overheads of forking a process/thread for every incoming client request, the Web server can fork a pool of processes/threads a priori, at start up. We assume that the Web server consists of a static thread/process pool, with the number of threads/processes in the pool or the pool size, denoted m. The client requests arrive according to a Poisson distribution and the service time of a client request is generally distributed. For most Web servers, the capacity of the queue to hold requests when all the resources are busy is typically very large to ensure that the probability of denying a request is very low. Thus, for the purpose of modeling we assume the queue size to be infinite. The

ICWE'06, July 11-14, 2006, Palo Alto, California, USA.
ACM 1-59593-352-2/06/0007.

queuing model which coincides with these characteristics of a Web server which is capable of processing multiple requests concurrently is a $M/G/m$ queue.

In the queuing model, the arrival process is Poisson with rate λ and there are m parallel servers serving the incoming client requests. The service times of requests are independent and identically distributed random variables, with a general distribution. The general distribution has a finite mean μ^{-1} and a finite coefficient of variance c_s (the ratio of standard deviation to the mean). Let $\rho = \lambda/(m\mu)$ denote the traffic intensity. We assume that $\rho < 1$ and the queuing discipline is FCFS.

The performance metric of interest is the expected or the average response time of a client request denoted R. Except for certain special cases, there is no known analytically or computationally tractable method for deriving an exact solution for the response time of a $M/G/m$ queue and several approximation approaches have been proposed. We consider the approximation by Sakasegawa [13] for the mean number of jobs in a $M/G/m$ queue. Using the mean number of jobs, the response time can be obtained using Little's law [7].

3. EXPERIMENTAL VALIDATION

The hardware platform hosting the experimental infrastructure comprises of one sever and one client. The two machines have the following configuration. The server is a Dell OptiPlex GX260 (Intel Pentium 4 processor at 2.4GHz, 1GB of RAM, 40GB hard driver and Intel PRO 1000 MT network adapter) and the client is an IBM ThinkPad T40 (Intel Pentium-M processor at 1.5GHz, 1GB of RAM, 40GB hard driver and Intel PRO 100 VE network adaptor). The server machine is installed with Microsoft IIS 5.1. The two computers are connected via a LAN.

In a multi-threaded server, as the size of the thread pool increases, each thread may experience some performance degradation. However, when the thread pool size is below a threshold, the performance degradation is negligible [17]. As a rule of thumb, to ensure negligible degradation, the size of the thread pool should be two times the number of CPUs on the host machine [8, 12]. Thus for the single processor server in our testbed, the size of the thread pool is chosen to be 2. The maximum queue length used in the experimental validation is very large compared to the number of threads, due to which the assumption of infinite queue length needed to apply the $M/G/m$ model is reasonable.

We consider the service scenarios where the clients request static files one at a time, and the Web server responds to the client with the file. We consider two types of workloads, namely, deterministic and heavy-tailed. For deterministic workload, all the client requests are for the same file size in a single experiment. The heavy-tailed workload was generated based on the Pareto distribution, which has been widely used to model heavy-tailed characteristics in a variety of phenomenon [11, 3, 5, 1, 4].

Our results indicate that the $M/G/m$ queue provides a reasonable estimate of the service response time for low to moderately high traffic intensities for both deterministic and heavy-tailed workloads.

4. ACKNOWLEDGMENTS

This research is supported in part by a grant from the University of Connecticut Research Foundation.

5. REFERENCES

[1] P. Barford and M. Crovella. Generating representative Web workloads for network and server performance evaluation. In *SIGMETRICS'98/PERFORMANCE'98 Joint International Conference on Measurement and Modeling of Computer Systems*, pages 151–160, 1998.

[2] J. Cao, M. Andersson, C. Nyberg, and M. Kihl. Web server performance modeling using an M/G/1/K*PS queue. In *10th International Conference on Telecommunications (ICT'03)*, pages 1501–1506, 2003.

[3] J. Charzinski. HTTP/TCP connection flow characteristics. *Performance Evaluation*, 42(2-3):149–162, 2000.

[4] M. Crovella, M. S. Taqqu, and A. Bestavros. *Heavy-Tailed Probability Distributions in the World Wide Web.* A practical Guide To Heavy Tails: Statistical Techniques and Application. Birkhauser, Boston, 1998.

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On the power-law relationships of the Internet topology. In *ACM SIGCOMM*, 1999.

[6] K. Kant and C. R. M. Sundaram. A server performance model for static Web workloads. In *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS'00)*, pages 201–206, 2000.

[7] L. Kleinrock. *Queueing systems, Volume 1: Theory.* John Wiley & Sons, New York, 1976.

[8] Y. Ling, T. Mullen, and X. Lin. Analysis of. optimal thread pool size. *ACM SIGOPS Operating System Review*, 34(2):42–55, 2000.

[9] D. Menascé. Web server software architecture. *IEEE Internet Computing*, 7(6):78–81, 2003.

[10] R. Nossenson and H. Attiya. The N-burst/G/1 model with heavy-tailed service-times distribution. In *Proceedings of 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'04)*, pages 131–138, 2004.

[11] V. Paxson and S. Floyd. Wide area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[12] J. Richter. *Advanced Windows (3rd Edition).* Microsoft Press, 1996.

[13] H. Sakasegawa. An approximation formula $L_q \doteq \alpha\rho^\beta/(1-\rho)$. *Annals of the Institute of Statistical Mathematics*, 29(1):67–75, 1977.

[14] L. Slothouber. A model of Web server performance. In *Proceedings of the Fifth International World Wide Web Conference*, 1996.

[15] M. S. Squillante, D. D. Yao, and L. Zhang. Web traffic modeling and Web server performance analysis. In *Proceedings of the 38th Conference on Decision and Control*, pages 4432–4439, 1999.

[16] R. D. van der Mei, R. Hariharan, and P. Reeser. Web server performance modeling. *Telecommunication Systems*, 16(3-4):361–378, 2001.

[17] D. Xu and B. Bode. Performance study and dynamic optimization design for thread pool systems. In *Proceedings of the International Conference on Computing, Communications and Control Technologies (CCCT'04)*, 2004.