Chapter 1.25
Performance Analysis of a Web Server

Jijun Lu
University of Connecticut, USA

Swapna S. Gokhale
University of Connecticut, USA

ABSTRACT

With the rapid development and widespread use of the Internet, Web servers have become a dominant source of information and services. The use of Web servers in business and critical application domains imposes stringent performance requirements on them. These performance requirements cast a direct influence on the choice of the configuration options of the hardware and the software infrastructure on which a Web server is deployed. In addition to the selection of configuration options, for a given level of load and a particular hardware and software configuration, it is necessary to estimate the performance of a Web server prior to deployment.

INTRODUCTION AND MOTIVATION

The World Wide Web (WWW) has experienced an exponential growth in the last 10 years and today Web servers are 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 with acceptable performance, which may be defined in terms of transaction throughput and latency experienced by the users (Van der Mei, Hariharan, & Reeser, 2001). The stringent performance requirements imposed on Web servers have a direct influence on the configuration options of the hardware and software infrastructure used for deployment. Hardware configuration options may include the capacity and the number of processors and caching strategies. Software configuration options may include the number of server threads/processes to serve client requests, the buffer size, and the scheduling discipline. Prior to deployment, for a given level of load, it is necessary to determine the hardware and software configuration options that would provide acceptable server performance.

One of the ways of estimating the performance of a Web server is by conducting actual measurements. While the measurement-based approach may be viable to estimate the performance for a given set of configuration options, it is cumbersome and expensive for "predictive" or "what-if" analysis and for an exploration of a set of alternative configurations. 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 is an attractive alternative to the measurement-based approach.

Web servers receive and process a continuous stream of requests. As a result, a vast majority of their time is spent waiting for I/O operations to complete, making them particularly apt to fall under the category of I/O intensive applications (Ling, Mullen, & Lin, 2000). The performance of such I/O intensive applications (responsiveness, scalability, and throughput) can be improved dramatically if they are provided with the capability to process multiple requests concurrently. Thus, modern Web servers invariably process multiple requests concurrently to enhance their performance and to fulfill their workload demands. Considering the concurrent processing capability, we propose the use of a multiserver *M/G/m* queue to model a Web server with an I/O intensive workload. The performance metric of interest is the response time of a client request. Since there is no known analytically or computationally tractable method to derive an exact solution for the response time of the *M/G/m* queue, we use an approximation proposed by Sakasegawa (1977). 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 for moderately high traffic intensity. The conceptual simplicity of the model combined with the fact that it needs the estimation of very few parameters makes it easy to apply.

The balance of the article is organized as follows: First, we present the performance model of a Web server. We then discuss the workload characteristics used for the experimental validation of the model, followed by a description of the experimental infrastructure used for validation. Subsequently, we present and discuss the experimental results. Research related to the present work is summarized next. Finally, we offer concluding remarks and directions for future research.

PERFORMANCE MODEL

We describe the performance model of a Web server in this section. Towards this end, we first provide an overview of the software architecture of a Web server. Subsequently, we discuss the rationale for modeling a Web server using an *M/G/m* queue and present an analytical expression to compute the approximate response time.

Web Server Software Architecture

Modern Web servers implement concurrent processing capability using a thread-based, a process-based, or a hybrid approach (Menasce, 2003). An example of a thread-based server is the Microsoft IIS server (Microsoft Corporation, n.d.), a process-based server is the Apache HTTP server 1.3, and a hybrid server is the Apache HTTP server 2.0 (Apache Software Foundation, n.d.).

In both the thread-based and process-based architectures, to avoid the overheads of forking a process/thread for every request, the Web server can fork a pool of processes/threads at start-up. If all these threads/processes are busy, either additional threads/processes can be forked or the request waits in a queue. In the former case, the size of the thread/process pool changes dynamically, whereas in the latter case the size of the thread/process pool is fixed and a new request