RC23097 (W0401-148) January 30, 2004
Mathematics

IBM Research Report

Analysis and Control of Correlated Web Server Queues

Soumyadip Ghosh
School of Operations Research and Industrial Engineering
Cornell University
Ithaca, NY

Mark S. Squillante
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Analysis and control of correlated Web server queues

Soumyadip Ghosh[a] and Mark S. Squillante[b]

[a]School of Operations Research and Industrial Engineering, Cornell University, Ithaca NY, USA
[b]Mathematical Sciences Department, IBM T.J. Watson Research Center, Yorktown Heights NY, USA

ABSTRACT

This paper demonstrates the existence of considerable dependencies between Web server arrival and service times, as well as strong dependencies within the arrival process. We derive a heavy-traffic stochastic-process limit for Web server performance, under various control policies, that captures these forms of correlations. This includes an analysis of control policies that provide near-optimal expected response times while also maintaining good response time variance properties.

Keywords: Performance analysis, control, stochastic models, queueing theory, Web server performance.

1. INTRODUCTION

Stochastic models play an important role in the design, performance analysis, and control of Web server systems. To be most effective in each of these areas, such models must capture in sufficient detail the key characteristics of Web server workloads, the key characteristics of Web server control policies for executing these workloads, and the impact of these workload and control policy characteristics on Web server performance and related measures. Of particular interest in our present study are very efficient mathematical methods that can be exploited online to support an increasingly important range of real-time applications (e.g., dynamic control of quality of service in Web servers to best satisfy service-level agreements), as well as in efficient interactive tools often used offline for many different purposes (e.g., what-if analysis and stochastic optimization for performance and capacity planning studies).

A significant body of research has investigated the workloads found in practice at a wide variety of Web servers; e.g., refer to the special issue of *World Wide Web*.[1] Most of these studies have demonstrated that the client request patterns exhibited at many different types of Web servers have strong dependence structures and that the requested files sizes at these Web servers are independent and identically distributed (i.i.d.) according to a heavy-tailed or subexponential distribution. However, the vast majority of these studies appear to have completely ignored the issue of whether there is any correlation between the client request arrival times and the requested file sizes or the request service times. This includes studies of Web server workload characterization and generation, Web server performance models, and Web server control policies.

In this study, we consider data from various production Web servers to investigate the issues of different forms of correlation in these Web server workloads and their implications on Web server performance and control policies. The results of our data analysis demonstrate the existence of significant dependencies between the client request interarrival times and the corresponding file sizes and service times. Our analysis also shows a strong dependence structure within the interarrival process of client requests, whereas the sequence of service times is essentially i.i.d. To the best of our knowledge, this is the first study to identify and quantify significant dependencies between the arrival and service processes in Web servers (as opposed to networks, where this issue goes all the way back to Kleinrock[2]). We then investigate the source of this cross correlation between the client request interarrival times and the corresponding file sizes and service times using the data from production Web servers. The results of our data analysis support the view that such cross correlations are primarily due to certain mixing effects involving the superposition of different classes of client requests each with different service requirements.

Based on this causal model as our motivation, we next turn to consider a mathematical analysis of queueing models of Web server performance, under various control policies, that capture the dependence structure within the arrival process as well as between the arrival and service processes. Unfortunately, the research literature is relatively limited in this respect for our purposes. Connelly and colleagues[3,4] consider models in which the service time of a customer is essentially a linear scaling of the interarrival time associated with the customer, which is then extended along a few different lines in subsequent papers.[5,6] The cross correlation provided by these models are often much stronger than that observed in the Web server data used in our study, in addition to not being consistent with our causal model. Boxma et al.[7] study a model of cross correlation due to a specific type of batching, but their causal model does not match at all with the Web server architectures and workload data considered in our study. Fendick et al.[8] consider a packet communication network queue with cross correlation resulting from multiclass batching effects that are somewhat related to our Web servers. However, all of these previous studies assume that the sequence of interarrival times in their system model is i.i.d., and the vast majority of these studies assume the system arrival process to be Poisson. This is quite different from the Web servers motivating our study where the arrival process of client requests has a strong dependence structure, both in the aggregate as well as within and across the arrival processes of different classes of requests.

We therefore derive a mathematical analysis of Web server performance, focusing initially on control policies that are representative of existing Web servers. Our approach is based on establishing heavy-traffic stochastic-process limits of single-server correlated queues that capture the complex properties found in the Web servers of interest. An approximation of Web server performance for all traffic intensities is obtained from these stochastic-process limits, yielding a closed-form expression for the expected equilibrium response time that is asymptotically exact and is easily parameterized from calculations based on readily available Web server data. Our experiments with data from Web servers demonstrate the accuracy of our expected response time approximation, which is in very good agreement with simulation results across all traffic intensities and is in excellent agreement for the range of traffic intensities of greatest interest to us. Such accuracy levels are achieved by paying specific attention to the manner in which the parameters of the expected response time expression are estimated. We also exploit our analysis to investigate the sensitivity of performance measures to different forms and degrees of correlation, each of which are explicitly represented in the expected response time expression. This includes the cross correlation between interarrival and service times observed in Web servers, which can have a significant impact on performance but is not captured in previously considered Web server queueing models.

We lastly turn to extend our mathematical analysis of Web server performance to further explore certain control policy issues in Web servers, based on our causal model as motivation. The optimality results of shortest remaining processing time (SRPT) and its variants with respect to minimizing expected response times are well known.[9] Some recent studies have further argued that SRPT does not unfairly penalize large customers in order to benefit small customers, and thus have proposed the use of SRPT to improve performance in Web server systems.[10, 11] On the other hand, first-come first-serve (FCFS) is known to minimize the waiting time variance of customers.[12] Moreover, our causal model suggests that the workloads found at various Web servers consist of multiple classes of client requests based on the different service requirements of these requests. We therefore consider the corresponding multiclass priority queue (using FCFS within each class) as an alternative approach for controlling the execution of client requests in Web servers, with the goal of providing expected response times close to those obtained under SRPT while also providing significantly better response time variance properties. Our analysis of Web server performance is extended to investigate this alternative control policy and demonstrate that it exhibits the desired properties.

The remainder of the paper is organized as follows. §2 presents our data analysis, and §3 presents our queueing-theoretic analysis. We then further explore control policy issues in §4, and provide concluding remarks in §5.

2. DATA ANALYSIS

Our study is based on data from various production Web servers. In each case, the distributed architecture used to support these Web servers generally consists of multiple single-server computing nodes to which incoming requests are routed by a front-end, high-speed router. This router attempts to balance the Web server load across the set of single-server computing nodes, where each computing node directly serves the requests routed to it and, in particular, operates independently of the other nodes.

Let T be the random variable denoting the client request response time in this queueing network. From the law of total probability we have

$$\mathsf{E}T \quad = \quad \sum_i \mathsf{E}T_i \; \mathsf{P}[\text{ request is served on node } i \,], \tag{1}$$

where $\mathsf{E}T_i$ is the expected response time of client requests served on computing node i. Every computing node in a Web server maintains its own time-series access log of all client requests that are served by the node. Our approach therefore consists of taking the access log data from each of the computing nodes i of the Web server to characterize the statistical properties of the corresponding arrival and service processes in order to obtain $\mathsf{E}T_i$ (using the results derived in §3) and the conditioning probability in (1). Of course, when the statistical properties of the workloads at all of the computing nodes are identical, then $\mathsf{E}T_i = \mathsf{E}T_j$ for all i, j, and thus equation (1) reduces to $\mathsf{E}T = \mathsf{E}T_i$.

The access logs contain several pieces of useful information about each client request served by the corresponding computing node. This includes the arrival time epoch of the k^{th} request served on the i^{th} computing node of the j^{th} Web server, which we denote by $A_{i,k}^{(j)}$, and the number of bytes comprising this client request, which we denote by $B_{i,k}^{(j)}$, $k \in \mathbb{Z}^+$. Even though the Web servers considered in our study serve dynamic content, we use a measurement-based function of the byte size of each client request as an accurate estimate of the service time for the request. The various arguments to justify this assumption are omitted due to space limitations, and we refer the interested reader to our technical report[13] for these details. We further point out that even if the service times of dynamic pages included the additional time to generate the page, this would still not eliminate the different sources of correlation demonstrated in this section. In fact, our analysis suggests that at worse this would yield a small change in the cross correlation exhibited below.

The unit of time in the access logs available to us is one second, which is quite standard. Since there can be tens to hundreds of client requests within a second at each computing node during peak traffic periods for the Web servers of interest, the access log data for each computing node provides us with the corresponding discrete-time batch process for the number of client requests per second. However, the direct use of this batch arrival process to estimate Web server performance can overestimate measures of performance by more than an order of magnitude, as demonstrated in a recent paper.[14] We therefore apply a somewhat modified version of the methodology proposed in the paper[14] to the sequence of arrival times $\{A_{i,k}^{(j)}\}$ extracted from the access log data for each computing node in order to obtain the corresponding interarrival sequence $\{\widehat{A}_{i,k}^{(j)}\}$, $\widehat{A}_{i,k}^{(j)} \in \mathbb{R}^+$, $k \in \mathbb{Z}^+$. It is important to note, however, that our analysis also includes direct consideration of the batch arrival process where the corresponding results are completely consistent with those presented in §2 and §3; see our technical report.[13]

We identify and focus on sufficiently long stationary intervals of traffic periods found in our analysis of the access logs from each computing node of every Web server. Of particular interest are peak traffic periods, given the importance of such intervals in capacity planning, dynamic resource allocation and other applications of performance analysis and control. This further motivates our use of heavy-traffic stochastic-process limits in the next section. These stationary intervals of peak traffic are comprised of traffic periods whose lengths are on the order of several hours and consist of at least several hundred-thousand data points. Thus, the corresponding processes $\{\widehat{A}_{i,k}^{(j)}\}$ and $\{B_{i,k}^{(j)}\}$ extracted from the Web server access logs are stationary sequences. Moreover, in the interest of space, we will henceforth focus on a representative access log from a specific computing node of a particular Web server, and therefore drop some of the indices by using the stochastic processes $\{\widehat{A}_k\}$ and $\{B_k\}$ to characterize the workload found in the corresponding access log, where $\widehat{A}_k$ describes the interarrival time of the k^{th} request and B_k the file size of the k^{th} request.

We now consider the dependence structures found in the Web server workloads used in our study. Figure 1(a) plots the autocorrelation function (ACF) of the interarrival process $\{\widehat{A}_k\}$, where the ACF of the stationary process $\{\widehat{A}_k\}$ at an integer lag ℓ is defined as $\mathsf{Cov}(\widehat{A}_1, \widehat{A}_{\ell+1})/\mathsf{Var}(\widehat{A}_1)$. This figure clearly demonstrates that there is a significant amount of correlation among the interarrival times at successive lags. Figure 1(b) plots the ACF of $\{B_k\}$, showing that the individual request sizes are essentially i.i.d. Figure 1(c) plots the cross correlation function (CCF) between the two processes $\{\widehat{A}_k\}$ and $\{B_k\}$ at an integer lag ℓ defined as $\mathsf{Cov}(\widehat{A}_1, B_{\ell+1})/\sqrt{\mathsf{Var}(\widehat{A}_1)\mathsf{Var}(B_1)}$. This figure clearly demonstrates that there is a considerable amount of (positive) cross correlation (relative to the statistical independence line) between the interarrival times and service times of Web servers for small lags.

2.1. Causal Model

One possible cause of this strong cross correlation between the interarrival and service processes can be explained as follows. Suppose the processes $\{\widehat{A}_k\}$ and $\{B_k\}$ are partitioned into different classes of client requests such that the per-class processes $\{\widehat{A}_{k,c}\}$ and $\{B_{k,c}\}$ are independent for each class c and the corresponding expected file sizes are different for different classes c and c', i.e., $\mathsf{E}B_{k,c} \neq \mathsf{E}B_{k,c'}$, $c \neq c'$ (where at least some of the per-class interarrival times are not exponentially distributed). Here we allow the interarrival process $\{\widehat{A}_{k,c}\}$ to have a strong dependence structure, and we further allow cross correlations among the per-class interarrival processes $\{\widehat{A}_{k,c}\}$. The key point is to have the processes $\{\widehat{A}_{k,c}\}$ and $\{B_{k,c}\}$ be independent of each other for all classes c. It then can be easily shown, via straightforward

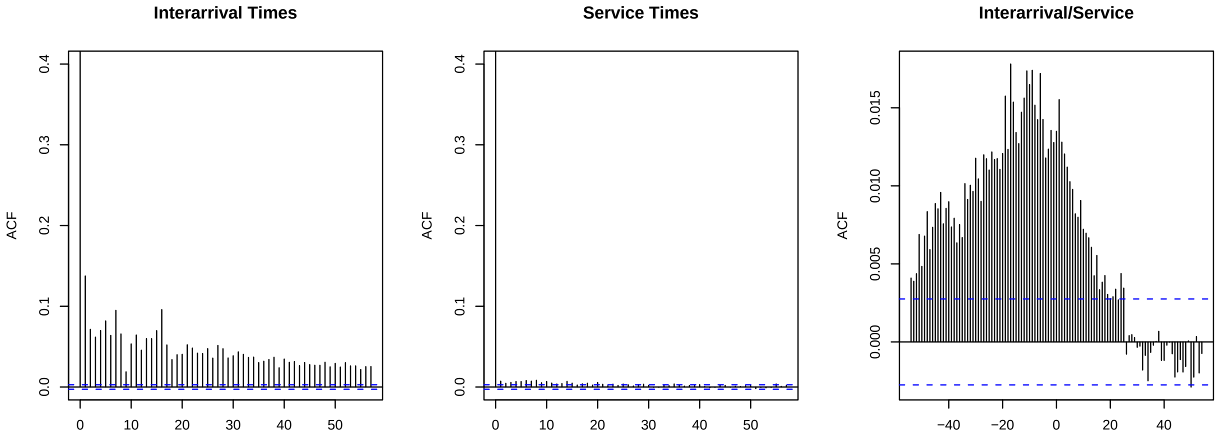

Figure 1. ACFs and CCF for the interarrival and service processes: (a) ACF of the interarrival times; (b) ACF of the service times; (c) CCF of the interarrival and service times.

calculations, that the superposition of these multiclass workload processes will yield aggregate processes $\{\widehat{A}_k\}$ and $\{B_k\}$ that have considerable cross correlation.

To demonstrate that this is a reasonably likely cause of such cross correlation in our Web servers, we take the corresponding access log data and seek to find confirmation of the properties of the above causal model by partitioning the original interarrival and service time processes from the data set into different classes based on the file size values. Specifically, we divided the file-size distribution into mutually exclusive intervals and then constructed the per-class processes $\{\widehat{A}_{k,c}\}$ and $\{B_{k,c}\}$ by assigning all requests of sizes within an interval to the corresponding class. The kernel density estimation of the file-size distribution (from the data) may provide a good initial indication of a classification of different types of requests, and the best intervals can be found by an exhaustive search. Of course, we are not claiming that this classification necessarily reflects the actual multiclass behavior of the workload processes, but rather we use it to provide support for, or against, our causal model.

We then consider the cross correlation between each of the per-class interarrival time and service time processes when there is a total of 3, 4, 5 and 6 classes. In all cases, the resulting cross correlation within each class is significantly reduced from that of Figure 1(c). Furthermore, these cross correlation values are indeed within two standard deviations of the independence line, which provides support for the conjecture of our causal model that the per-class interarrival time and service time processes are independent. Note that as the number of classes is increased, we could easily find partitions where the cross correlation between the per-class interarrival and service processes decreases steadily.

Figure 2 provides a representative sample of our results corresponding to those presented in Figure 1(c) for a partitioning of the workload into 6 classes based on file sizes. Observe that the cross correlation between the per-class processes $\{\widehat{A}_{k,c}\}$ and $\{B_{k,c}\}$ has been significantly reduced in relative magnitude (with respect to the independence line) from more than 6 times in the original data (in Figure 1(c)) to well within 2 times the independence line in each of the classes. These results provide considerable support for the conjecture of our causal model of an underlying multiclass workload where the per-class interarrival and service processes are independent and the per-class service processes have different service requirements.

3. MATHEMATICAL ANALYSIS

We next turn to consider a mathematical analysis of the queueing system representing the distributed architecture of the Web server environment motivating our study. Our starting point is equation (1) which expresses the expected client request response time of the entire Web server in terms of the expected response time of client requests at each of the computing nodes comprising the Web server. We therefore focus on each computing node of the Web server of interest and model each

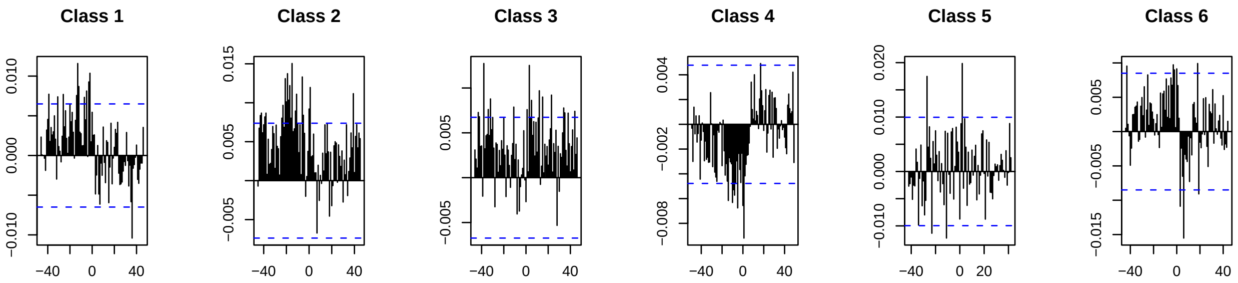

Figure 2. CCF of the interarrival and service times within each class of a 6-class workload partitioning.

node as a general single-server queue, although extensions to a multiserver queue can be readily added on top of the results derived in this section.[15] For various reasons, including the argument that the actual execution ordering of client requests at the computing nodes of many Web servers is somewhere in between pure FCFS and pure processor sharing (PS), we consider both FCFS and PS queueing disciplines in our analysis of each general single-server queue. The results of our analysis for the FCFS discipline are presented herein, because of our extension of this analysis in §4 to further investigate control policy issues, but the corresponding PS results are omitted due to space limitations. We refer the interested reader to our technical report[13] for these results and related technical details.

Let u_k represent the time between the $k-1^{st}$ and k^{th} arrivals, and let v_k represent the service time of the k^{th} arrival, $k \geq 1$. The interarrival time u_k is identical to the interarrival time $\widehat{A}_k$ of the k^{th} client request obtained from the access log of the Web server computing node of interest, and the service time v_k is obtained directly from the corresponding byte size B_k of this k^{th} request, both as discussed in §2. Define $U_k = u_1 + \ldots + u_k$ and $V_k = v_1 + \ldots + v_k$, $k \geq 1$. The sequence of successive waiting times $\{W_k; k \geq 1\}$ can then be defined in terms of the sequence $\{(u_k, v_k) : k \geq 1\}$ using Lindley's recursion[16]:

$$W_{k+1} \;=\; [W_k + v_k - u_{k+1}]^+ \;=\; D_k - \min_{1\leq j\leq k}\{D_j\} \tag{2}$$

where $d_k = v_k - u_{k+1}$, $D_k = d_1 + \ldots + d_k$, $k \geq 1$, $D_1 = 0$, $W_1 = 0$ and $(x)^+ \equiv \max\{0, x\}$. The rightmost representation of W_k in (2) describes the waiting time process as a random walk process with random steps d_i, reflected at the barrier 0.

Let u_k^n and v_k^n represent the interarrival time and the service time of the k^{th} client request in the n^{th} queue of a sequence of queues. The heavy-traffic limits are then obtained for scaled versions of the stochastic processes associated with queue n such that the traffic intensities for the sequence of queues increase successively to the critical value of 1 as $n \to \infty$. In establishing these heavy traffic limits, we will be using the theory of weak convergence of probability measures on the space D of all right-continuous functions with finite left-limits on $[0, \infty)$; refer to Billingsley[17] and Whitt.[15]

Assuming the sequence $\{u_j^n, v_j^n\}$ is stationary, we define a sequence of queues with $\alpha_n = \mathsf{E}u_1^n$ and $\beta_n = \mathsf{E}v_1^n$ which vary such that $\beta_n/\alpha_n = \rho_n \to 1$ as $n \to \infty$. For our model of each computing node, we fix the arrival rate for all queueing systems to the arrival rate λ from the corresponding data set and construct our sequence of queueing systems by changing the service rate such that $1 - \rho_n = n^{-1/2}$. Hence, $\alpha_n = 1/\lambda$ for all n and $\beta_n = \rho_n/\lambda = (1 - n^{-1/2})/\lambda$. We then define a random element $(\widehat{U}^n, \widehat{V}^n)$ in the product function space $D \times D$ corresponding to the n^{th} queue, where

$$(\widehat{U}^n(t), \widehat{V}^n(t)) \;=\; (n^{-1/2}[U^n_{\lfloor nt\rfloor} - \alpha_n nt],\; n^{-1/2}[V^n_{\lfloor nt\rfloor} - \beta_n nt]), \qquad t \geq 0,$$

and $\lfloor x\rfloor$ is the greatest integer less than or equal to x. Let $\widehat{D}^n$ and $\widehat{W}^n$ be the random elements associated with the differences and waiting times in (2) defined as

$$\widehat{D}^n(t) \;=\; n^{-1/2} D^n_{\lfloor nt\rfloor} \qquad \text{and} \qquad \widehat{W}^n(t) \;=\; n^{-1/2} W^n_{\lfloor nt\rfloor}, \qquad t \geq 0.$$

3.1. General Results

We now are able to present the following important result for our completely general $G/G/1$ queue of each Web server computing node, which parallels the results in Iglehart and Whitt[18]; see also Whitt.[15]

Theorem 3.1.

(a). If $(\widehat{U}^n, \widehat{V}^n) \Rightarrow (\widehat{U}, \widehat{V})$ in $D \times D$ where $(\widehat{U}, \widehat{V})$ has continuous paths with probability 1 and $\widehat{E}^n(t) = n^{-1/2}(\alpha_n - \beta_n)nt \to \alpha e$, $-\infty < \alpha < \infty$, as $n \to \infty$, then $\widehat{D}^n \Rightarrow \widehat{D} = \widehat{V} - \widehat{U} - \alpha e$ in D, where $e(t) = t$.

(b). If $\widehat{D}^n \Rightarrow \widehat{D}$ in D, then $\widehat{W}_n \Rightarrow \widehat{W} = f(\widehat{D})$ in D, where $f(x)(t) = x(t) - \inf\{x(s) : 0 \leq s \leq t\}$, $t \geq 0$, is a functional on D that represents the reflective barrier at 0.

(c). If, in addition, the limit $(\widehat{U}, \widehat{V})$ is a two-dimensional Brownian motion with a symmetric covariance matrix having elements σ_{11}^2, σ_{22}^2 and σ_{12}^2, then $\widehat{D}$ is a Brownian motion (BM) and $\widehat{W}$ is a reflected Brownian motion (RBM) with drift $-\alpha$. Furthermore, if $\alpha > 0$, then $\widehat{W}$ has an exponential equilibrium distribution $\widehat{W}(\infty)$ with expectation

$$\mathsf{E}\widehat{W}(\infty) = (\sigma_{11}^2 + \sigma_{22}^2 - 2\sigma_{12}^2)/2\alpha. \tag{3}$$

Proof. The first two parts are elementary consequences of the continuous mapping theorem (e.g., see Whitt[15]). Since $\widehat{E}^n$ converges to the linear functional αe, the joint convergence given in part (a) can be extended to $(\widehat{U}^n, \widehat{V}^n, \widehat{E}^n) \Rightarrow (\widehat{U}, \widehat{V}, \alpha e)$. From (2), $\widehat{D}^n = \widehat{U}^n - \widehat{V}^n - \widehat{E}^n$, and thus (a) holds by the continuous mapping theorem applied to the subtraction mapping. Similarly, for part (b), notice that $\widehat{W}^n = f(\widehat{D}^n) = f(\widehat{U}^n - \widehat{V}^n - \widehat{E}^n)$, and again we obtain the convergence by the continuous mapping theorem applied to the barrier function f. The last part follows from the fact that $\widehat{D}^n \Rightarrow \widehat{U} - \widehat{V} - \alpha e$ is a BM with drift $-\alpha$ and a diffusion coefficient $\sigma_{11}^2 + \sigma_{22}^2 - 2\sigma_{12}^2$, and thus $f(\widehat{D})$ is an RBM. If an RBM has a negative drift, then its equilibrium distribution is exponential with the stated expectation. □

A consequence of the above theorem is the existence of Brownian limits for the $\widehat{D}^n$ and $\widehat{W}^n$ processes. The BM limits further imply that the following central limit theorems (CLTs) hold:

$$\begin{aligned} n^{-1/2}[U_n - \alpha_n n] &\Rightarrow N(0, \sigma_A^2), & (4)\\ n^{-1/2}[V_n - \beta_n n] &\Rightarrow N(0, \sigma_S^2), & (5)\\ n^{-1/2}[(V_n - U_n) - (\beta_n - \alpha_n)n] &\Rightarrow N(0, \sigma_{AS}^2), & (6) \end{aligned}$$

where the variance parameters coincide with the functional central limit theorem (FCLT) parameters, i.e., $\sigma_A^2 = \sigma_{11}^2$, $\sigma_S^2 = \sigma_{22}^2$ and $\sigma_{AS}^2 = \sigma_{11}^2 + \sigma_{22}^2 - 2\sigma_{12}^2$, the numerator in (3).

The variance parameters in the CLT limits also can be identified more generally in the case where the CLTs hold for $\alpha_n \to \overline{\alpha}$ and $\beta_n \to \overline{\beta}$ (and not necessarily $\overline{\alpha} = \overline{\beta}$) as $\sigma_A^2 = \overline{\alpha}^2 C_A^2$, $\sigma_S^2 = \overline{\beta}^2 C_S^2$ and $\sigma_{AS}^2 = \sigma_A^2 + \sigma_S^2 - 2\overline{\alpha}\overline{\beta} C_{AS}^2$, where C_A^2, C_S^2 and C_{AS}^2 are the asymptotic variability parameters defined as

$$\begin{aligned} C_A^2 &= \lim_{n\to\infty} n\frac{\mathsf{Var}(U_n)}{(\mathsf{E}U_n)^2} = \lim_{n\to\infty} \frac{\mathsf{Var}(U_n)}{n\alpha_n^2}, & (7)\\ C_S^2 &= \lim_{n\to\infty} n\frac{\mathsf{Var}(V_n)}{(\mathsf{E}V_n)^2} = \lim_{n\to\infty} \frac{\mathsf{Var}(V_n)}{n\beta_n^2}, & (8)\\ C_{AS}^2 &= \lim_{n\to\infty} n\frac{\mathsf{Cov}(U_n, V_n)}{(\mathsf{E}U_n)(\mathsf{E}V_n)} = \lim_{n\to\infty} \frac{\mathsf{Cov}(U_n, V_n)}{n\alpha_n\beta_n}. & (9) \end{aligned}$$

The existence of these limits imply that the variance of the quantities U_n and V_n and the covariance between them increases at the same rate as n, which is equivalent to assuming that the processes are weakly dependent.[15]

While the CLTs in (4) – (6) are most useful for our heavy-traffic analysis, the asymptotic variability parameters in (7) – (9) are most useful for calculating performance measures based on our heavy-traffic limits from the data at the corresponding Web server computing node. In particular, we can estimate the heavy traffic steady-state waiting time in our general $G/G/1$ queue of each Web server computing node from the values of C_A^2, C_S^2 and C_{AS}^2. From our formulation of the sequence of queueing systems, we have in the heavy traffic limiting sequence $\alpha_n = 1/\lambda$, $\beta_n = \rho_n/\lambda = (1 - n^{-1/2})/\lambda \to 1/\lambda$, and $n^{1/2}(\alpha_n - \beta_n) \to 1/\lambda$. Hence,

$$\sigma^2_{AS} \;=\; \frac{1}{\lambda^2}\,(C^2_A + C^2_S - 2C^2_{AS}), \tag{10}$$

which implies that

$$\mathsf{E}\widehat{W}(\infty) \;=\; \frac{1}{\lambda}\,\frac{(C^2_A + C^2_S - 2C^2_{AS})}{2}. \tag{11}$$

From previous heavy-traffic stochastic-process limit theorems,[15, 18, 19] we know that the limits of the sequences of normalized equilibrium waiting time expectations exist and that they coincide with the expected equilibrium measure of the corresponding RBM. It then follows for the heavy-traffic regime

$$\lim_{n\to\infty} n^{-1/2}\,\mathsf{E}W^n_\infty \;=\; \lim_{n\to\infty}(1-\rho_n)\,\mathsf{E}W^n_\infty \;=\; \mathsf{E}\widehat{W}(\infty) \;=\; \frac{1}{\lambda}\,\frac{(C^2_A + C^2_S - 2C^2_{AS})}{2}, \tag{12}$$

$$\mathsf{E}W^n_\infty \;\sim\; \frac{\mathsf{E}\widehat{W}(\infty)}{(1-\rho_n)} \;\approx\; \frac{\mathsf{E}\widehat{W}(\infty)}{(1-\rho)} \;=\; \frac{1}{\lambda}\,\frac{(C^2_A + C^2_S - 2C^2_{AS})}{2(1-\rho)} \;=\; \overline{\beta}\,\frac{(C^2_A + C^2_S - 2C^2_{AS})}{2(1-\rho)}, \tag{13}$$

where $f(n) \sim g(n)$ denotes that $\lim_{n\to\infty} f(n)/g(n) = 1$. In order to obtain the corresponding steady-state waiting time approximation for any traffic intensity $\rho < 1$, it is important to note that the Brownian approximation in (13) actually represents the conditional expectation $\mathsf{E}[W^\rho(\infty) \mid W^\rho(\infty) > 0]$. Thus, upon unconditioning,[16] we have

$$\mathsf{E}W^\rho(\infty) \;=\; \frac{\rho\,\overline{\beta}}{1-\rho}\,\frac{(C^2_A + C^2_S - 2C^2_{AS})}{2}, \tag{14}$$

with the corresponding steady-state response time expressed as

$$\mathsf{E}T^\rho(\infty) \;=\; \overline{\beta} \;+\; \frac{\rho\,\overline{\beta}}{1-\rho}\,\frac{(C^2_A + C^2_S - 2C^2_{AS})}{2}, \tag{15}$$

where the three quantities in the numerator of the last term can be easily estimated from the data at the corresponding Web server computing node using the asymptotic variability relations in (7), (8) and (9).

3.2. Application of General Results

The first step in exploiting the general results of §3.1 for our purposes consists of showing that the conditions of Theorem 3.1 are satisfied for our general $G/G/1$ queue of each Web server computing node, especially focusing on the BM assumption of part (c) in the theorem. This is because having the BM condition of (c) hold implies that the conditions of part (a) are also satisfied, which in turn implies that the assumption of part (b) holds.

Let us first consider the case of $\widehat{V}^n$. Recalling from §2 that the service times of client requests in the Web server access logs from each computing node are i.i.d., we shall henceforth suppose that the v^n_i are i.i.d. with a squared coefficient of variation C^2_s. Then, by Donsker's theorem,[15] since $\beta_n \to \overline{\beta} = 1/\lambda$, we have

$$\widehat{V}^n(t) \;=\; n^{-1/2}[V_{\lfloor nt\rfloor} - \beta_n nt] \;\Rightarrow\; \widehat{V},$$

where $\widehat{V}$ is a zero drift BM with variance constant $\sigma^2_{22} = (1/\lambda^2)C^2_s$.

Considering the case of $\widehat{U}^n$, we recall from our statistical analysis in §2 that the interarrival process has a strong dependence structure. We can still have the desired BM limit as $n \to \infty$ provided that the correlation structure does not grow too fast. In fact, the existence of the limit in (7) is a necessary condition. The sufficiency conditions to be satisfied by an arrival process in order for the BM limit to exist, however, are stronger; refer to Whitt.[15] For example, if a process is stationary with uniform mixing, or if a process is ergodic and martingale properties hold, then the BM limit is known to exist. While these conditions can be difficult to check with complete certainty, many stochastic processes such as martingales, discrete-time Markov chains, continuous-time Markov chains, Markovian Arrival Processes (MAPs) and regenerative processes have been shown to satisfy these conditions; see Billingsley[17] and Whitt.[15] Whitt[15] further notes that in practical applications it is often reasonable to assume that the FCLT is valid if the asymptotic variance parameter is finite (which holds in our case).

Turning to the data from §2 for each computing node comprising the Web server, we note that recent studies using the same data sets have shown that these stationary arrival times are consistent with and can be accurately modeled by various instances of a MAP.[20, 21] We shall therefore conclude that the computing node arrival process is a MAP, which is further supported by the strong accuracy of our approximation in equations (14) and (15) obtained using the results of Theorem 3.1 as illustrated later in this section. This implies, as noted above, that the corresponding BM limit exists. The interarrival process $\widehat{U}^n$ is the inverse process associated with the MAP computing node arrival process. Hence, by Corollary 13.7.2 in Whitt,[15] we then have

$$\widehat{U}^n(t) \;=\; n^{-1/2}[U_{\lfloor nt \rfloor} - \beta_n nt] \;\Rightarrow\; \widehat{U},$$

where $\widehat{U}$ is a zero-drift BM with variance constant $\sigma_{11}^2 = (1/\lambda^2)C_A^2$.

The next step is to establish that the joint convergence for $(\widehat{U}^n, \widehat{V}^n)$ also holds. Clearly, the convergence of the marginals $\widehat{U}^n$ and $\widehat{V}^n$ does not necessarily imply that this joint-convergence will hold, and thus the processes will have to satisfy some stronger conditions to ensure that the joint limit exists. We know from the Cramer-Wold device (e.g., see Billingsley[17]) that the joint limit for $(\widehat{U}^n, \widehat{V}^n)$ exists if, and only if, the sum $m_1\widehat{U}^n + m_2\widehat{V}^n$ converges for every combination of $(m_1, m_2) \in \mathbb{R}^2$. The asymptotic variability parameter of the sum, if it exists, is then $(m_1^2 C_A^2 + m_2^2 C_S^2 + 2m_1 m_2 C_{AS}^2)/4$, and thus it is finite as long as C_{AS}^2 is finite. Our statistical analysis in §2 of the data from each computing node comprising the Web server shows that this is indeed the case. Hence, in a similar vein as our findings for the FCLT of $\widehat{U}$ to hold, we conclude that C_{AS}^2 is finite and thus the FCLT for the sum holds.

Notice that, as pointed out in Fendick et al.,[8] since the limit process for $\widehat{W}^n$ is inferred only from the limit for the process $\widehat{D}^n = \widehat{U}^n - \widehat{V}^n - \widehat{E}^n$, the existence of $\widehat{D}$ is technically sufficient for the limit $\widehat{W}^n \Rightarrow \widehat{W}$ to hold. Here $\widehat{D}^n$ is a translation of a linear combination of the $\widehat{U}^n$ and $\widehat{V}^n$ with asymptotic variability parameter given by $C_A^2 + C_S^2 - 2C_{AS}^2$. We can therefore reasonably suppose that the limit for the $\widehat{D}^n$ and $\widehat{W}^n$ processes holds given that the constants C_A^2, C_S^2 and C_{AS}^2 are finite.

Finally, the approximation for the steady-state waiting time in (14) is dependent on the underlying processes $\widehat{U}^n$ and $\widehat{V}^n$ only through the asymptotic parameters $\overline{\beta}$, C_A^2, C_S^2 and C_{AS}^2. Moreover, this expression is invariant to what distribution is chosen or how the dependence structure is modeled, as long as the asymptotic parameter values do not change. This feature is of course true in general for approximations obtained from Brownian limits.

3.3. Numerical Experiments

Let us now turn to consider the accuracy of our expressions for the steady-state waiting time and response time in (14) and (15), respectively. Figure 3 plots a representative sample of the expected waiting time and response time measures as a function of the traffic intensity ρ, together with the corresponding empirical steady-state waiting and response times obtained directly from the sequences $\{\widehat{A}_k\}$ and $\{B_k\}$ using Lindley's recursion in (2). The asymptotic variability constants C_A^2, C_S^2 and C_{AS}^2 in (14) and (15) are estimated from the data at the corresponding Web server computing node through the asymptotic variability parameters in (7) – (9). This involves estimating the asymptotic variance parameters σ_A^2, σ_B^2 and σ_{AB}^2, which from the CLTs are $\mathsf{Var}(U_n)/n$, $\mathsf{Var}(V_n)/n$ and $\mathsf{Var}(V_n - U_n)/n$ as $n \to \infty$. As in our analysis above, to preserve the statistical properties of the interarrival times in the Web server data set, we consider a sequence of queues with the workload increasing to the critical value of 1 by steadily scaling up the service times via the single-server queue capacity.

We observe from these results that our closed-form expressions for the expected equilibrium waiting time and response time in (14) and (15) provide excellent approximations for traffic intensities $\rho > 0.5$. In fact, the confidence intervals for our expected waiting time approximation completely overlap with those from Lindley's recursion applied to the Web server data when $0.5 < \rho < 1$. Moreover, in comparing the expected response time curves, we find that they are essentially indistinguishable for $\rho > 0.5$. Although our primary interest is in moderate to heavy traffic intensities (e.g., $\rho > 0.5$), we observe that the expected response time curve from our approximation in (15) differs with the corresponding curve from Lindley's recursion by very small margins under lighter traffic intensities. We further observe that our approximations provide an upper bound on the expected waiting and response time measures across all traffic intensities, at least for the Web servers used in our study.

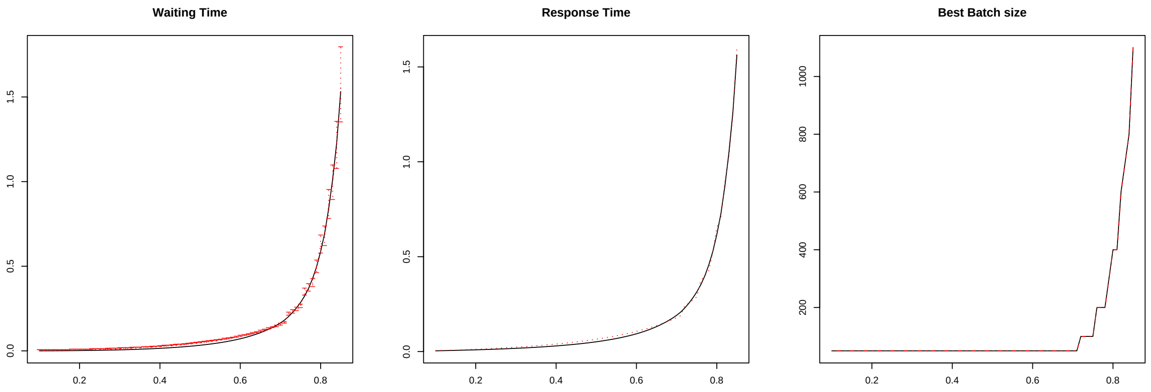

Figure 3. Accuracy of our approximations as a function of the traffic intensity ρ: (a) Expected waiting time from (14) and (2), together with confidence intervals; (b) Expected response time from (15) and (2); (c) Best batch size for estimating approximation parameters.

Remark 3.2. Since the desired estimators are asymptotic limits, we employ steady-state estimation techniques such as batch-means to obtain these estimates of C_A^2, C_S^2 and C_{AS}^2 and to produce confidence intervals for the corresponding estimate of expected waiting time. The key idea behind the batch-means estimates is to divide (or section) a single sample path into intervals (or batches) which can be assumed to be approximately independent. We observe from our experiments that the results of our approximation can be quite sensitive to the batch size selected for these calculations, where our empirical results in Figure 3(c) show that the best batch size is relatively small for light traffic intensities and that the best batch size increases as the traffic intensity ρ increases. This demonstrates the increasing impact of correlations on our steady-state performance measures as ρ increases. The results in Figure 3(a)-(b) were obtained using the best batch sizes from our empirical analysis, which ranged from 50 to 2400.

It is important to note that heavy-traffic approximations are well known to have the potential to be inaccurate at light to moderate traffic intensities. This has been observed in a number of studies, many of which have developed heuristics to address such inaccuracies. These heuristics are often based on some form of interpolation between light-traffic and heavy-traffic formulas, one of the most notable being the approach proposed by Reiman and Simon.[22] In our study we have instead focused on issues related to the steady-state estimation methods used to obtain the values of the parameters of our heavy-traffic approximations, as explained in Remark 3.2. This appears to be an interesting area of research which we plan to explore in more detail as part of future work.

Our closed-form expressions for the expected equilibrium waiting time and response time in (14) and (15) represent very efficient mathematical methods that can be exploited online to support an increasingly important range of real-time performance and control applications, such as the dynamic control of resource allocation in Web servers to best satisfy performance guarantees as part of service-level agreements. These results can be equally well exploited in efficient interactive tools often used offline for many different purposes, such as what-if analysis and stochastic optimization as part of performance and capacity planning studies. In addition to these obvious applications of our results, the expression in (14) can be used directly to gain key insights into the performance impact of correlations and variability within and between the interarrival and service processes. Specifically, the expected equilibrium waiting time varies linearly with changes in either of $\overline{\beta}$, C_A^2, C_S^2 or C_{AS}^2 while keeping the other parameters constant. Moreover, additional variability or correlation within the interarrival or service processes tends to increase the expected waiting time, while an increase in the (positive) cross-correlation between the two processes tends to decrease the expected waiting time. The rate of increase with increasing C_A^2 or C_S^2 is the same, whereas the rate of decrease with increasing C_{AS}^2 is twice as large.

Lastly, we note that, in addition to the significant dependence structure within the arrival process (as captured in C_A^2), the cross correlation between the arrival and service processes (as captured in C_{AS}^2) observed in the access logs of the computing nodes comprising the various Web servers used in our study can also be quite significant. Thus, Web server queueing models from previous studies that do not capture these correlation factors will yield expected response time

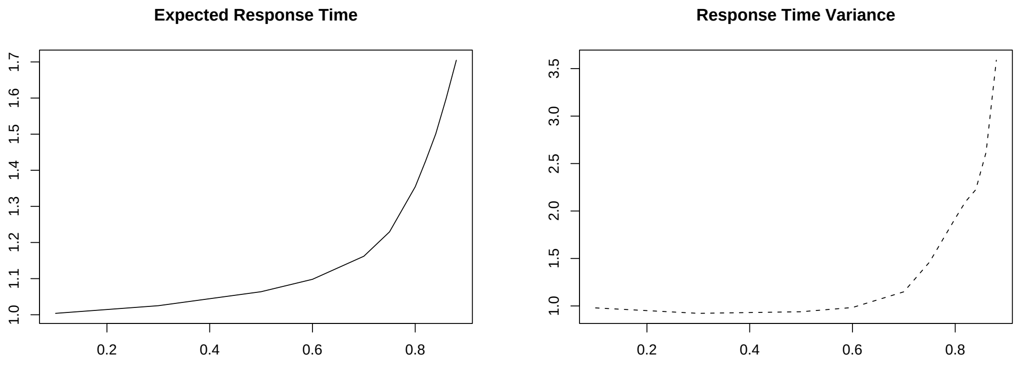

Figure 4. Ratios of expected response time and response time variance for the control policies as a function of the traffic intensity ρ: (a) Expected response time ratio of MCPQ to SRPT; (b) Response time variance ratio of SRPT to MCPQ.

estimates with considerable errors from what will be experienced in practice, at least for the Web server environments of interest in our study.

4. CONTROL POLICY ISSUES

The causal model of §2 suggests that the workloads found at various production Web servers consist of multiple classes of client requests based on the different service requirements of these requests. We therefore consider the corresponding multiclass priority queue (MCPQ) as an alternative to existing control policies for scheduling the execution of client requests in Web servers, with the goal of providing expected response times close to their optimal values obtained under SRPT[9] while also providing significantly better response time variance properties. Serving requests within each class according to an FCFS discipline can reduce the waiting time variance,[12] while the priority discipline among the classes can provide an execution ordering somewhat close to SRPT provided that the service time variability within each class is relatively low.

In this section our analysis of Web server performance is extended to investigate these control policy issues. We start by considering the representative workload data set from §2 and use simulation to estimate the first two moments of the client request response times under SRPT. For comparison, we use a variant of the partitioning from §2 of this workload into different classes of client requests (for the case of 5 classes), and obtain via simulation the first two moments of the client request response times (taken over all classes) in the corresponding MCPQ. In both cases, preemptive versions of the scheduling mechanisms are considered. A representative sample of our relative expected response time and response time variance results are provided in Figure 4.

We first observe that, with the possible exception of heavy traffic intensities, the foregoing goal is achieved: Expected response times under the MCPQ are relatively close to those obtained under SRPT while yielding significantly smaller response time variances (note the difference in scale between the y-axis of the two plots). Moreover, even at heavy traffic intensities, our results illustrate a very interesting tradeoff between a non-negligible relative increase in the expected response time and a significant relative decrease in the response time variance (more than a factor of 3.5). It also should be noted that there obviously will be less preemption overhead incurred under the MCPQ than under SRPT (which is not modeled in the results of Figure 4). Furthermore, if non-preemptive versions of both approaches are employed instead, we would expect the mean response time results in Figure 4 to be even closer with relatively little changes to the response time variance results in Figure 4.

In addition to these performance results, the approach based on the MCPQ has the added advantage of not having to know precisely the service times of each client request. Instead, one only needs to be able to partition the workload into different classes where the service times within each class are relatively close to each other and the service times across classes are relatively different. Along these same lines, it is important to note that the partitioning of the Web server workloads into multiple classes taken from §2 can be improved for our purposes here. Recall that our objective in §2 was to obtain a partitioning that yielded mutually independent arrival and service processes within each class, in order to support the conjecture of our causal model for cross correlation between the arrival and service processes. Our objective for the partitioning here is very different in that improvements over the results presented in Figure 4 can be achieved by obtaining a partitioning that takes into account the goals of this section. These issues are the subject of future work.

Given the above potential benefits of the MCPQ, we next extend our mathematical analysis to consider the general G/G/1 queue of each Web server computing node under this control policy for scheduling the execution of client requests. Our (preliminary) approach consists of first decomposing the MCPQ into separate per-class queues and then solving the resulting per-class queues based on the mathematical analysis derived in §3. Specifically, we exploit the strict ordering of the priority classes and the properties of the preemptive priority queueing discipline to isolate the per-class queues by decomposing the per-class performance characteristics in a hierarchical manner such that the analysis of the decomposed queue for each class k in isolation is based on the solution for the decomposed queues of classes $1, \ldots, k-1$.

Our starting point is to obtain the expected response time of client requests served on computing node i in the corresponding MCPQ, which is then used in equation (1) to obtain the overall expected system response time as in the previous section. From the law of total probability we have

$$\begin{aligned} \mathsf{E}T_i &= \sum_k \mathsf{E}[\, T_i \mid \text{request belongs to class } k \,] \; \mathsf{P}[\text{ request belongs to class } k \,], \\ &= \sum_k \mathsf{E}T_{i,k} \; \mathsf{P}[\text{ request belongs to class } k \,], \end{aligned} \tag{16}$$

where $\mathsf{E}T_{i,k}$ is the expected response time of class k requests served in the MCPQ on computing node i. Let $\mathcal{C}_k$ represent the capacity of the Web server node i from the perspective of class k. Since class 1 has the highest priority, we set $\mathcal{C}_1$ to be the overall capacity of the Web server node. Define $\widetilde{\rho}_k = \lambda_k \widetilde{\beta}_k$ and $\widetilde{\beta}_k = \mathcal{C}_k^{-1} \mu_k^{-1}$, where μ_k^{-1} is the expected offered service time of class k requests.

We first consider the class 1 queue in isolation. Since the lower priority classes do not interfere with the execution of class 1 requests under the preemptive priority queueing discipline, then from equation (15) the expected response time of class 1 requests is given by

$$\mathsf{E}T_{i,1} \;=\; \frac{\widetilde{\rho}_1 \, \widetilde{\beta}_1}{1-\widetilde{\rho}_1} \, \frac{(C_{A,1}^2 + C_{S,1}^2 - 2C_{AS,1}^2)}{2} \;+\; \widetilde{\beta}_1. \tag{17}$$

Then, as a first-order approximation, class 2 sees a server with capacity that has been reduced by class 1 requests, and thus we set $\mathcal{C}_2 = \mathcal{C}_1(1-\widetilde{\rho}_1)$ and recursively obtain the solution for the next class. In general, assuming we have obtained the expected response times for the higher priority classes $1, \ldots, k-1$, the expected response time of class k requests is given by

$$\mathsf{E}T_{i,k} \;=\; \frac{\widetilde{\rho}_k \, \widetilde{\beta}_k}{1-\widetilde{\rho}_k} \, \frac{(C_{A,k}^2 + C_{S,k}^2 - 2C_{AS,k}^2)}{2} \;+\; \widetilde{\beta}_k, \tag{18}$$

where $\mathcal{C}_k = \mathcal{C}_1(1 - \sum_{k'=1}^{k-1} \widetilde{\rho}_{k'})$. Of course, in estimating the parameters $C_{A,k}^2$, $C_{S,k}^2$ and $C_{AS,k}^2$ we would need to include the various forms of correlation both within and across classes, and we follow the approach presented in §3.

The foregoing analysis is clearly a first-order approximation. However, comparisons between these closed-form expressions and the simulation results in Figure 4 demonstrate that the relative errors are less than $5-10\%$ for $\rho > 0.5$ and they are always within 15% for all values of ρ. We are exploring more accurate and robust approximations, based on our heavy-traffic stochastic-process limits of §3, for the expected response times in the MCPQ considered in this section. The collection of these results and those in (17) and (18) can then be used together with bounds and approximations on the variance of the BM and RBM limits in §3 to investigate in more detail the key tradeoff between small relative increases in the expected response time and large relative decreases in the response time variance under the multiclass priority control scheme, especially at heavy traffic intensities.

5. CONCLUSIONS

In this paper we used data from various production Web servers to demonstrate the existence of considerable dependencies between the arrival times and the service times of client requests, in addition to a strong dependence structure within the arrival process. Our data analysis investigated the likely causes of this cross correlation, and our queueing analysis demonstrated that such cross correlation can have a significant impact on performance (independent of the cause of such correlations) which has not been captured in previously considered Web server performance models. An approximation of Web server performance was derived, based on heavy-traffic stochastic-process limits, that captures both the correlations within the arrival process and the correlations between the arrival and service processes. We then demonstrated the accuracy of our asymptotically-exact approximation, which is excellent across all traffic intensities and is especially accurate for the range of traffic intensities of greatest interest to us. Such accuracy levels are achieved by paying specific attention to the manner in which the parameters of the expected response time expression are estimated. Our mathematical analysis was then extended to further investigate certain control policy issues in Web servers, demonstrating the ability to provide expected response times relatively close to their optimal values obtained under SRPT while also providing much better response time variance properties.

REFERENCES

1. F. Douglis, ed., *World Wide Web*, Special Issue on Workload Characterization and Performance Evaluation, vol. 2, Baltzer, June 1999.
2. L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill, 1964.
3. B. W. Conolly, "The waiting time process for a certain correlated queue," *Op. Res.* **16**, pp. 1006–1015, 1968.
4. B. W. Conolly and N. Hadidi, "A correlated queue," *J. Appl. Prob.* **6**, pp. 122–136, 1969.
5. B. W. Conolly and Q. H. Choo, "The waiting time process for a generalized correlated queue with exponential demand and service," *SIAM J. Appl. Math.* **37**(2), pp. 263–275, 1979.
6. I. Cidon, R. Guerin, A. Khamisy, and M. Sidi, "Analysis of a correlated queue in a communication system," in *Proc. IEEE INFOCOM '93*, **1**, pp. 209–216, 1993.
7. S. C. Borst, O. J. Boxma, and M. B. Combe, "Collection of customers: A correlated M/G/1 queue," in *Proc. ACM SIGMETRICS and PERFORMANCE '92 Joint Conference*, pp. 47–59, June 1992.
8. K. W. Fendick, V. R. Saksena, and W. Whitt, "Dependence in packet queues," *IEEE Trans. on Comm.* **37**, pp. 1173–1183, 1989.
9. L. E. Schrage, "A proof of the optimality of the shortest remaining processing time discipline," *Op. Res.* **16**, pp. 687–690, 1968.
10. M. E. Crovella, R. Frangioso, and M. Harchol-Balter, "Connection scheduling in Web servers," in *Proc. USENIX Symposium on Internet Technologies and Systems*, pp. 243–254, October 1999.
11. N. Bansal and M. Harchol-Balter, "Analysis of SRPT scheduling: Investigating unfairness," in *Proc. ACM SIGMETRICS Conference*, pp. 279–290, June 2001.
12. J. F. C. Kingman, "The effect of queue discipline on waiting time variance," in *Proc. Cambridge Phil. Soc.*, **58**, pp. 163–164, 1962.
13. S. Ghosh and M. S. Squillante, "Analysis of correlated queues based on Web server data," tech. rep., IBM Research Division, 2002.
14. C. H. Xia, Z. Liu, M. S. Squillante, L. Zhang, and N. Malouch, "Analysis of the performance impact of drill-down techniques for Web traffic models," in *Proc. International Teletraffic Congress Conference*, August 2003.
15. W. Whitt, *Stochastic-Process Limits*, Springer-Verlag, New York, 2002.
16. J. W. Cohen, *The Single Server Queue*, North Holland, 1982.
17. P. Billingsley, *Convergence of Probability Measures*, Second edition, Wiley, New York, 1999.
18. D. L. Iglehart and W. Whitt, "Multiple channel queues in heavy traffic. I," *Adv. Appl. Prob.* **2**, pp. 150–177, 1970.
19. J. F. C. Kingman, "On queues in heavy traffic," *J. Royal Stat. Soc.* **B24**(2), pp. 383–392, 1962.
20. A. Riska, M. S. Squillante, S.-Z. Yu, Z. Liu, and L. Zhang, "Matrix-analytic analysis of a MAP/PH/1 queue fitted to Web server data," in *Advances in Algorithmic Methods for Stochastic Models*, G. Latouche and P. Taylor (eds.), World Scientific, 2002.
21. B. Ray and M. S. Squillante, "A nonlinear model of Web server traffic and implications on long-range dependence," tech. rep., IBM Research Division, 2002.
22. M. I. Reiman and B. Simon, "An interpolation approximation for queueing systems with Poisson input," *Op. Res.* **36**, pp. 454–469, May-June 1988.