Resource Management Policies for E-commerce Servers *

Daniel A. Menascé
Dept. of Computer Science
George Mason University
Fairfax, VA 22030
USA
menasce@cs.gmu.edu

Virgilio A. F. Almeida
Dept. of Computer Science
Univ. Federal de Minas Gerais
Belo Horizonte, MG 30161
Brazil
virgilio@dcc.ufmg.br

Rodrigo Fonseca
Dept. of Computer Science
Univ. Federal de Minas Gerais
Belo Horizonte, MG 30161
Brazil
rfonseca@dcc.ufmg.br

Marco A. Mendes
Dept. of Computer Science
Univ. Federal de Minas Gerais
Belo Horizonte, MG 30161
Brazil
corelio@dcc.ufmg.br

Abstract

Quality of service of e-commerce sites has been usually managed by the allocation of resources such as processors, disks, and network bandwidth, and by tracking conventional performance metrics such as response time, throughput, and availability. However, the metrics that are of utmost importance to the management of a Web store are revenue and profits. Thus, resource management schemes for e-commerce servers should be geared towards optimizing business metrics as opposed to conventional performance metrics. This paper introduces a state transition graph called Customer Behavior Model Graph (CBMG) to describe a customer session. It then presents a family of priority-based resource management policies for e-commerce servers. Priorities change dynamically as a function of the state a customer is in and as a function of the amount of money the customer has accumulated in his/her shopping cart. A detailed simulation model was developed to assess the gain of adaptive policies with respect to policies that are oblivious to economic considerations. Simulation results show that the adaptive priority scheme suggested here can increase, during peak periods, business-oriented metrics such as revenue/sec by as much as 43% over the non priority case.

*The subject matter of this paper is covered by a pending patent application at the United States Patent and Trademark Office.

1 Introduction

It has been recognized by many that congestion and poor performance can be the major impediments for the success of e-commerce. Many e-commerce sites, especially those in the financial trading business, have been facing serious problems and financial losses when customers are not allowed to trade in a timely manner. Some disgruntled customers sue online trading services if they feel they have been short changed and others just move their business elsewhere.

IT managers of Web stores have been managing allocation of resources such as processors, disks, and networks, by tracking conventional performance metrics such as response time, throughput, and availability. However, the metrics that are of utmost importance to the management of a Web store are revenue and profits. Thus, resource management schemes for e-commerce servers should be aimed at optimizing revenue/sec instead of focusing on conventional performance metrics.

Resource management policies for e-commerce sites should be based on the behavior of customers and on how they change state as they navigate through the site going from browsing to searching, selecting items, adding them to their shopping carts and paying. Resource management policies for e-commerce sites should aim at optimizing business metrics, instead of just trying to increase throughput or decrease response time. We present in this paper a family of priority-based resource management policies for e-commerce servers. Priorities change dynamically as a function of the state a customer is in, as a function of the user profile, and as a function of the amount of money the customer has accumulated in his/her shopping cart. The policies can also be tuned to provide good performance to customers who are just entering the Web store even before they add any items to their shopping carts. We believe that resource management policies such as the ones presented in this paper should be integrated into future commercial e-commerce products. This would allow e-commerce servers to handle peak loads with existing resources in a way that minimizes revenue loss due to poor quality of service.

A detailed simulation model was developed to assess the gain of our policies with respect to policies that are oblivious to monetary considerations. The simulation is driven by SURGE [2]—a workload generator for Web sites. SURGE is used to generate the requests that start a customer session. Requests generated by a customer within a session are generated from a Customer Behavior Model Graph (CBMG) that captures how users navigate through the site. The CBMG representation is introduced in this paper as a means of characterizing workloads for e-commerce sites. As an example, two types of customer profiles, heavy and occasional were considered. Each customer profile has its own CBMG. From the CBMG, one can obtain the average number of times a state is visited per entry to the Web store, the average session length, and the buy to visit ratio. The results of our simulations show that the adaptive priority scheme suggested in this paper can increase business-oriented metrics such as revenue/sec by as much as 43% over the non priority case.

The rest of this paper is organized as follows. Section two discusses new metrics for e-commerce sites. Section three describes e-commerce workloads as composed of session requests and customer behavior model graphs (CBMGs). Section four describes new resource management policies for e-commerce servers. The next section describes the simulator and the simulation environment used to analyze the new policies proposed. Section six describes the numerical results obtained. Section seven compares our work with that of others. Finally, section eight presents some concluding remarks.

2 Metrics and Measurements for E-commerce

The quality of the service provided by on-line information systems, such as Web servers, has been traditionally assessed by metrics such as response time, throughput, reliability, and availability. Response time can be measured at the server side, in which case it does not include any client and external network time, or it can be measured from the user's perspective, in which case it includes components such as browser time, network access time at the client side, ISP time (at both ends), Internet time, network access time at the server side, and server response time.

Throughput is usually measured in requests/sec or transactions/sec and determines the rate at which the system can deliver work. While throughput is important from the perspective of the site administrator, it is irrelevant to end-users who are concerned about the response time they get. When the response time is too high, users complain if they have no choice but use the system. In e-commerce, customers usually have a choice: they leave the site and move to another Web store. This translates into lost revenue for the Web store and decreased throughput.

E-commerce brings the need of novel metrics that reflect at the same time the needs of the Web store and those of users. One such metric defined here is *revenue throughput*, denoted by X^+, measured in dollars/sec generated by completed transactions. One of the goals of a Web store should be to maximize the revenue throughput. Implicit in this metric is a measure of customer satisfaction with response time, since if customers were really unhappy they would not shop at the Web store and the revenue throughput would decrease. A frustated customer is a quickly exiting customer. Therefore, another metric is *potential lost revenue/sec*. This metric, denoted by X^-, represents the rate at which dollars accumulated in shopping carts are lost when customers leave the Web store because of inadequate response time. The resource management policies discussed in this paper attempt to maximize a Web store's revenue throughput without the need to increase server capacity.

3 Understanding E-commerce Workloads

E-commerce workloads are composed of sessions. A *session* is defined by a sequence of requests of different types such as browse, search, select, add to the shopping cart, and pay. The allowed sequences of requests can be described by a state transition graph called *Customer Behavior Model Graph (CBMG)*. This graph has one node for each possible state (such as home page, browse, search, select, add, and pay) and transitions between these states. A probability is assigned to each transition. Different types of users may be characterized by different CBMGs in terms of the transition probabilities. Thus, workload characterization for e-commerce entails in determining the sets of CBMGs that best characterize customer behavior. As e-commerce sites become more sophisticated, they can process their logs to identify user profiles based on the navigation and buying patterns. These profiles could be specified as CBMGs. Thus, as a customer starts to navigate through a site, the Web store could attempt to match the customer to one of the existing profiles and assign priorities based on the user profile. The marketing and economic value of customized navigation experience made possible by the vast amount of information collected by Web servers has been pointed out in [10]. Some Web stores request that users login before they start to navigate through the site. In these cases, it may be even easier to match a customer with a profile.

In this paper, we distinguish between two user profiles: occasional and heavy buyers. The first category is composed of customers who use the Web store to find out about existing products, such as new books or best fares and itineraries for travel, but end up not buying, most of the time, at the Web store. The second category is composed of customers who have a higher probability of buying if they see a product that interest them at a suitable price. Figs. 1 and 2 show the CBMGs for occasional and heavy buyers, respectively. Another aspect of the workload model is the mix of customer types. We denote by f_o and f_h the fraction of occasional and heavy buyers who visit the Web store. In this paper, we assume a much higher percentage of occasional buyers so that the overall percentage of customers who end up buying, i.e., visiting the "pay" state is around 5%. This number tends to be typical for many Web stores [8].

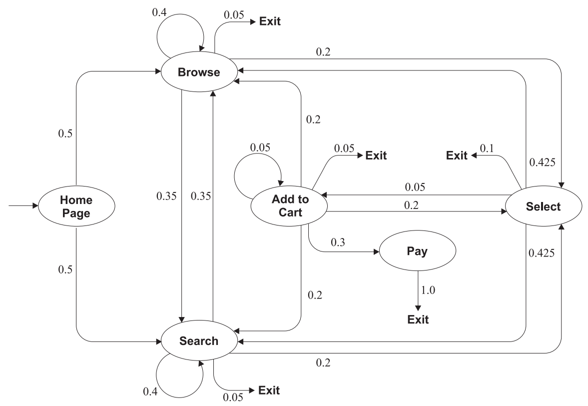

Figure 1: Customer Behavior Model Graph for an occasional buyer.

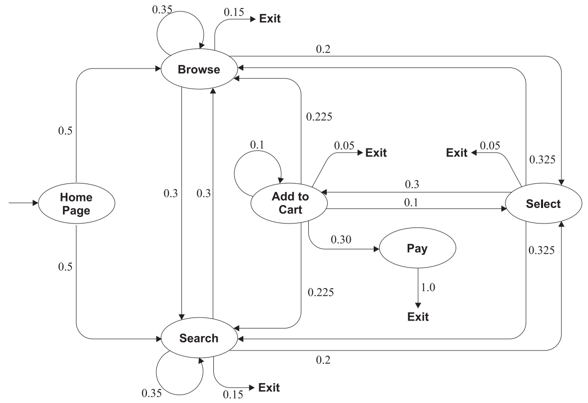

Figure 2: Customer Behavior Model Graph for a heavy buyer.

The global workload of an e-commerce site can then be characterized by groups of users with similar behavior. The behavior of each group is characterized by a CBMG with the proper transition probabilities and by the arrival behavior of customers of that group. The arrival process is characterized by the average arrival rate of the request that starts the session and by the think times within the session. A think time can be associated with each type of transition in the CBMG. For example, the think time associated with the transition between states "browse" and "select" indicates how long it takes a customer to submit a select request after receiving a reply from a browse request. Think times are not shown in Figs. 1 and 2 but the specific values used in the simulation are given in Table 1. The transitions that indicate exit from the Web store, from states other than "pay", are indicative of spontaneous exits. The resource management policies described in this paper attempt to reduce the number of customers who leave the Web store due to poor performance.

The CBMG provides useful information regarding the average number of visits to each state for each visit to the site's home page. Consider the vector

$$\vec{V} = (V_b, V_s, V_a, V_t, V_p)$$

which represents the average number of visits to the states browse (b), search (s), add (a), select (t), and pay (p), respectively, for each visit to the home page. Let the 5×5 matrix $P = [p_{i,j}]$ denote the transition probabilities between states b, s, a, t, and p. This is the order of rows and columns in P. Then we can write the systems of linear equations $\vec{V} - \vec{V}_{hp} = \vec{V} \times P$, where $\vec{V}_{hp} = (p_{h,b}, p_{h,s}, 0, 0, 0)$ is the vector of transition probabilities from the home page state to the other states in the CBMG. For the CBMGs of Figs. 1 and 2, this system of linear equations becomes

$$\begin{aligned} p_{b,b}\ V_b + p_{s,b}\ V_s + p_{a,b}\ V_a + p_{t,b}\ V_t &= V_b - p_{h,b} && (1)\\ p_{b,s}\ V_b + p_{s,s}\ V_s + p_{a,s}\ V_a + p_{t,s}\ V_t &= V_s - p_{h,s} && (2)\\ p_{a,a}\ V_a + p_{t,a}\ V_t &= V_a && (3)\\ p_{b,t}\ V_b + p_{s,t}\ V_s + p_{a,t}\ V_a &= V_t && (4)\\ p_{a,p}\ V_a &= V_p && (5) \end{aligned}$$

Note that for each CBMG, we have a different system of linear equations because the transition probabilities are different. When necessary, we will use the superscripts o and h to differentiate between the two cases. The solution to the system of linear equations $\vec{V} - \vec{V}_{hp} = \vec{V} \times P$ for the graphs of Figs. 1 and 2 is $V^o = (V_b = 6.76, V_s = 6.76, V_a = 0.14, V_p = 0.04, V_t = 2.73)$ and $V^h = (V_b = 2.71, V_s = 2.71, V_a = 0.37, V_p = 0.11, V_t = 1.12)$, respectively.

From the vector $\vec{V}$ one can obtain some useful metrics. The first, called *average session length* and denoted by $\bar{S}$, is the average number of pages visited by a customer for each visit to the Web store. Thus, $\bar{S} = 1 + V_b + V_s + V_a + V_t + V_p$. The average session length for occasional and heavy buyers, for the CBMGs of Figs. 1 and 2 is $\bar{S}^o = 17.45$ and $\bar{S}^h = 8.03$, respectively. Another metric of interest is the *buy to visit ratio*, denoted by BV. This is equal to the ratio between the average number of customers who buy from the Web store and the total number of visits to the Web store. For each type of customer the buy to visit ratio is given by V_p. Then, $BV = f_o \times V_p^o + f_h \times V_p^h$. In our simulations we used $f_o = 0.9$. Thus, $BV = 0.9 \times 0.04 + 0.1 \times 0.11 = 0.047$.

One can find an upper bound on the revenue throughput, X^+, as

$$X^+ \leq \lambda_s \ \ (f_o \times V_a^0 \times \bar{P}^o + f_h \times V_a^h \times \bar{P}^h) \qquad (6)$$

where λ_s is the average arrival rate of session requests and $\bar{P}^o$ and $\bar{P}^h$ are the average price of an item added to the shopping cart by occasional and heavy customers, respectively. The right hand side of Eq. (6) is an upper bound for X^+ because not all customers who add items to their shopping carts end up buying. Some customers leave the Web store spontaneously, through the exit transitions in the CBMG. Others, abort the session due to very high response times.

4 New Resource Management Policies

One of the goals of an e-commerce server should be to maximize the revenue generated by the site. To that end, one must understand the behavior of Web servers and how performance may impact user behavior, in particular during peak periods when the site's resources are stressed by bursts of customers. Therefore, resource management policies should be devised to support business-oriented goals.

4.1 Limitations of Current Policies

This is a typical scenario in e-commerce sites. Sometimes, the traffic to the site becomes too high (e.g., three or four times greater than the average traffic) and the server capacity is not able to handle the customers that are trying to enter the site. Usually, the quality of service is degraded to all customers that attempt to visit the site during these peak periods. Long waiting times to interact with the Web server and refused connections are typical indicators of poor quality of service in e-commerce sites. Current resource management policies do not make any distinction between customers and do not provide differentiated quality of service. For example, an e-store may want to favor customers who really intend to purchase something. However, in current systems, resources are equally shared between customers who are making a purchase and customers who are just browsing.

General-purpose operating systems and Web servers do not provide adequate support for resource management of e-commerce sites. They allocate resources to multiple processes considering only throughput and ignore the relative importance of processes and their meaning to business goals. Operating systems usually manage the allocation of CPU time, memory and I/O, without taking into consideration any special characteristic of the workload, such as who generated the requests and what amount of money is associated with that requests.

4.2 E-commerce Oriented Resource Policies

E-commerce oriented resource management policies should be **adaptive** and evolve according to customer usage and profiles. Resources should be allocated on an economic basis. The resource allocation policies examined in this paper are geared towards CPU and disk scheduling at the various servers of a Web store. The optimization criterium for the policies is to maximize the value of the metric revenue/sec.

To achieve this goal, the policies analyzed here use priorities based on user profiles, on current session length (S), on the amount of money accumulated in the customer's shopping cart ($\$sc$), and on the states visited in the CBMG. Three priority classes are considered: high, medium, and low. Customers transition between these priority classes as shown in the priority transition diagram of Fig. 3. The transitions between priority classes are labeled by conditions which may depend on the state visited (h, b, s, a, t, or p), the session length S, and the

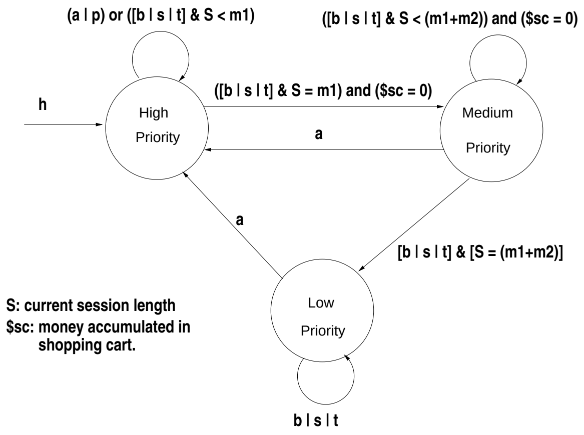

Figure 3: Priority scheme for e-commerce sites.

accumulated money in the shopping cart $\$sc$. As shown in the figure, all users entering the Web store from the home page start at high priority and remain there if they add at least one item to their shopping cart or their session length is less than a threshold m_1. The rationale is to give high priority to users who are in the early stages of exploring the store so that they do not give up before they are given an opportunity to find out what the store has to offer. After the session length exceeds the threshold m_1 and the customer has not added anything to its shopping cart, its priority is lowered to Medium. He/she stays at that level of priority while no items are added to the shopping cart and the session length has not reached the second threshold $m_1 + m_2$. When this happens, the customer's priority is lowered to Low. The priority is changed to High from Medium or from Low if an item is added to the shopping cart. Note that by making m_2 large enough, the priority scheme can be reduced to a two-class priority scheme. By making m_1 large enough, the priority framework of Fig. 3 reduces to a singe-class priority scheme. Thus, by varying the values of m_1 and m_2 one can generate a large number of policies.

Subclasses may be defined within each priority class according to user profiles. For example, if the profiles heavy buyer and occasional buyer are defined, higher priority would be given to all customers with a heavy buyer profile over customers with an occasional buyer profile.

Within each each priority subclass, we use the following disciplines at processors and disks:

- CPU: processor sharing (PS) for classes Medium and Low and ordered by $\$sc$ for the High priority class. In other words, if $\$sc_1 > \$sc_2$ then customer 1 has higher non-preemptive priority at the CPU than customer 2.
- disks: FIFO for classes Medium and Low and ordered by $\$sc$ for the High priority class.

5 A Framework to Compare The Policies

Because no current system implements all the ideas and concepts proposed in this paper, we need to construct an experimental environment to simulate adaptive systems, customer behavior, and workloads that conform to those systems. We built an e-commerce site simulator which is a hybrid between a trace-driven simulator and a discrete event simulator. As shown in Fig. 4, we use a realistic Web workload generation tool to generate a stream of requests to start customer sessions and use the CBMGs and the simulator to generate the individual requests that make up a session.

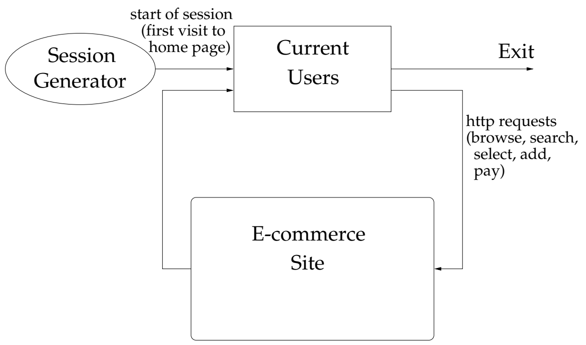

Figure 4: Overview of the e-commerce site simulator.

The workload of an e-commerce site is highly dependent on the interaction between customers and the store site. For example, after deciding on which item to buy after a successful search in the store's database, a customer may leave the store instead of completing the purchase, if it takes too long to receive a response from a request. In order to mimic this process and be able to evaluate changes in the user behavior caused by the proposed adaptive schemes of the site, we use a hierarchical simulation model for workload generation.

5.1 Hierarchical Workload Generation

The hierarchical simulation model is constructed as follows. First, the model creates service requests that initiate customer sessions. Once a session starts, the HTTP requests of the session are generated according to the CBMG of the workload. For each HTTP request, the simulator generates specific requests for the site facilities, such as http servers, CGI servers, DB servers, and LANs. Each facility simulates the execution of a request, demanding service from its main resources, such as CPU, disks and network bandwidth.

At the higher-level, we use SURGE [2] to generate requests to start sessions at the e-commerce site. SURGE is used to guarantee burstiness and realistic interarrival distributions for the service requests. At session start time though, we do not know yet the sequence of HTTP requests that will make up the session because the system is adaptive and the customer behavior depends on the system performance. Thus, the next request to be generated by the user is a function of the CBMG and the system responsiveness.

When a new session is initiated, the customer is assigned to a profile (i.e., Heavy Buyer or Occasional Buyer) represented by a CBMG. During a session, the customer goes through the states of the CBMG, and each of these states generates a single HTTP request to the store site, that processes it and sends the response. Then, based on the current state, the transition probabilities and think times associated with each transition in the CBMG, and the response time of the last request, a new state is computed and a new request is generated for the site. Let us describe how request timestamps are generated. Assume that a customer is at time t_1 at a given state and decides to generate a transition to another state. Let us assume that the think time (Z) is drawn from an exponential distribution. Then the timestamp for the new request will be $t_1 + Z + 2 \times RTT$, where RTT is the network roundtrip time.

The request generating process is repeated until the customer leaves the site. There are two ways to leave the site. In the first one, a customer decides to leave the site spontaneously, without any specific reason. This case is represented by the "exit arrow" in each state of a CBMG. A probability is assigned to the exit transition in each state. The second reason to leave is poor performance. The system is slow to answer a request and the customer gets impatient and quits the site. This phenomenon is modeled by a "customer impatience factor", defined as a function of two parameters: timeout and number of retries. The timeout is defined as

$$timeout \;=\; c_2(state) + c1 \times session\_length \qquad (7)$$

where $c_2(state)$ is a constant that depends on the state of the CBMG and c_1 is fixed at 0.1. The values of c_2 used in our simulations are $c_2(b) = 9, c_2(s) = 9, c_2(t) = 8, c_2(a) = 8$, and $c_2(p) = 30$. A retry means that a user presses the "STOP" button and the "LOAD" button of the browser in an attempt to get the response. Occasional buyers are assumed to retry once and heavy buyers three times.

5.2 Simulation Model

To assess the impact of the resource management policies described here, we simulated an electronic bookstore. Customers are assumed to search for books by either keyword or by author. Two types of books were considered: technical and non-technical. Table 1 specifies the percent of technical books selected by customers of our online bookstore, the average price and price range for each type of book, the percentage of searches by author and keyword, and the average size of a page returned by searches by keyword and author on technical and non-technical books. The distribution of book prices is a truncated Gaussian distribution. The values in Table 1 were obtained by analyzing results of many searches performed in various existing online bookstores. There are many possible values for the parameters used in the simulation studies. Due to space constraints we are only presenting here results for a limited set of values.

Think Times:		
all transitions: 15 sec:	exceptions:	
	Select to Add:	45 sec
	Add to Pay:	25 sec
	Search to Select:	30 sec
Percent of Technical Books Selected: 20%		
Book Price Parameters (Distribution: truncated Gaussian)		
Average Prices:		
Technical Books	$45.00	
Non-Technical Books	$18.00	
Price Range:		
Technical Books	[$5.00,$100.00]	
Non-Technical Books	[$5.00,$60.00]	
Searches by author: 60%		
Searches by keyword: 40%		
Avg. size (in KB) of pages: returned by Search requests:		
	Technical	Non-Technical
By Keyword	20	25
By Author	11	16
Avg. size (in KB) of pages returned by Browse requests: 20		

Table 1: Workload parameters.

The architecture of the e-commerce site for our bookstore is shown in Fig. 5. It is composed of one authentication server, 3 HTTP servers, 3 CGI servers, 3 DB servers, and two 100-Mbps LANs. Each DB server has three disks with 9 msec average seek time, 4.7 msec average latency, and 30 MB/sec transfer rate.

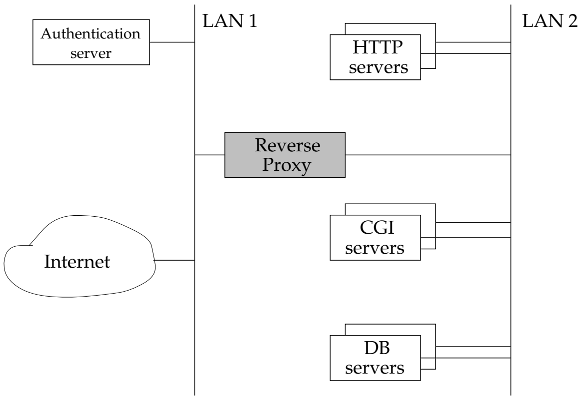

Figure 5: Architecture of the Web store.

Table 2 contains additional parameters used by the simulator. These parameters along with the LAN bandwidth and disk performance characteristics were used to obtain the service demands for each type of request at each state. The service demand D_{CPU}, in seconds, at the CPU of each server was computed using the function $D_{\mathrm{CPU}}(BD, BT) = 0.00258 \times (BD + BT) + 0.0027$ where BD is the number of KB read/written from disk and BT is the number of KB transmitted.

Associated with each state defined in the CBMG is a sequence of operations to be performed at the store site, i.e., the sequence of the site's components a request in that state has to go through. Ultimately, this sequence, together with the load in each of the store's components, determines the response time seen by the customer. Table 3 shows the site's components involved in the processing of the requests associated with each CBMG state.

Parameter Name	Parameter Description	Value
SHttpReq	Size of an HTTP req (includes protocol ovhd)	0.358 KB
RTT	Avg. round trip time (RTT)	0.125 sec
SCgiDBReq	Avg. size of a request to a CGI or DB server (includes ovhd)	0.258 KB
SCGIFile	Avg. size of a file containing a CGI script	2 KB 2 KB
BAddReq	Avg. KB read/written at the DB Server per Add request	2 KB
SStaticPage	Avg. size of a static page	2 KB
BSelecReq	Avg. KB read/written at the DB Server per Select request	3 KB
BPayReq	Avg. KB read/written at the DB Server per Pay request	2 KB

Table 2: Request/Reply size parameters.

CBMG State	Site's Components
H	HTTP
B	HTTP,CGI,DB
S	HTTP,CGI,DB
T	HTTP,CGI,DB
A	HTTP,CGI,DB
P	HTTP,CGI,DB,CGI,AS

Table 3: Components of the site configuration involved in the request execution.

The simulation model was developed in C and made use of the discrete event simulation environment SMPL [6]. For all measured values, the 95% confidence intervals were within 5% of the mean values.

To conclude this section, it is worth mentioning that a real log (consisting of HTTP requests) would be useless for our purpose of testing this new adaptive mechanism. The reason is that the user behavior, in terms of request generation, is a function of the CBMGs and system performance. A log of a system that does not implement the type of resource management scheme proposed here would not have the sequences that would result from the interaction of customers with a site that implements such policies.

6 Performance of New Resource Management Policies

Many analyses were made with the use of the simulator. Due to space limitations we only present here a few of them. The metrics plotted in the figures that follow are: revenue/sec (X^+) for occasional and heavy buyer customers, percent angry customers $\%A$, defined as the percent of customers who leave the site due to poor performance, potential lost revenue/sec (X^-) for the no priority case and for the priority scheme suggested here, and average response time for the priority and no priority cases. All graphs in this section depict the variation of a metric as a function of the session arrival rate λ_s. The value of m_2 used throughout the simulations reported here is 4. The value of m_1 varied as shown in the figures.

Figs. 6 and 7 show the variation of revenues/sec X^+ as a function of λ_s for various values of m_1 for heavy and occasional buyers, respectively. In both figures, the no priority case is also shown. It can be observed from Fig. 6 that for heavy buyers, the revenue/sec always increases as the system load increases and in virtually all cases, the revenue/sec is better with the priority scheme than with no priorities, especially for higher load values. For example, for $\lambda_s = 30$ sessions/sec, the value of X^+ for $m_1 = 2$ is 29% higher than for the no priority case. We also see from the figure that X^+ is very sensitive to m_1 as the load varies.

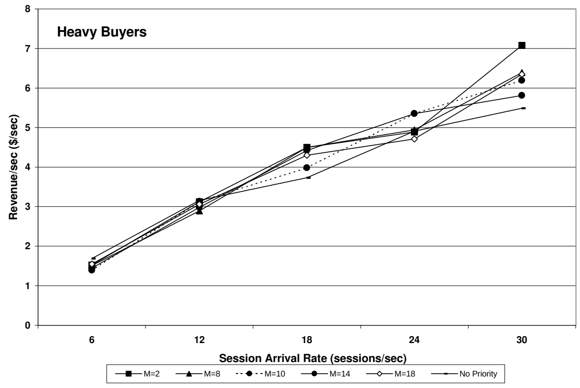

Figure 6: Revenue/sec vs. session arrival rate for heavy buyers.

For occasional buyers, as seen in Fig 7, the value of X^+ increases with λ_s up to a certain value of λ_s and then starts to decrease. The reason is that for heavy loads, occasional buyers experience much higher response times than heavy buyers and are more likely to exit the site by timeouts. As with the heavy buyers case, the value of X^+ is almost always better with priorities than without. The maximum value of X^+ for occasional buyers was observed to be \$10.59/sec and it occurred for $\lambda_s = 24$ sessions/sec for $m_1 = 14$. For heavier loads, e.g. $\lambda_s = 30$ sessions/sec, the maximum value of X^+ is 43% higher than that for the non-priority case. This happens for $m_1 = 10$. Note that a very small value of m_1 ($m_1 = 2$) for occasional buyers, hurt them more than in the case of no priority. This is because, occasional buyers will very quickly move into medium or low priority and will very likely timeout due to high response times. To help put the values of revenue/sec reported here in proper perspective, a Web store with the sustained revenue/sec shown in Figs. 6 and 7 translates into a yearly revenue of \$338 million assuming 16-hour days at that revenue/sec rate and 365 days/year, for the priority case and $m_1 = 10$.

The behavior of Fig. 6 can be better understood by analyzing Fig. 8 that shows the variation of $\%A$ with the load. As it can be seen, for load intensity values up to 12 sessions/sec, there are virtually no angry customers. However, as the load intensity increases, more and more customers become angry. The no-priority case displays the highest percent of angry customers and the $m_1 = 10$ case is the one with the lowest value of $\%A$ for very high loads. For $\lambda = 30$ sessions/sec, 53% of the customers

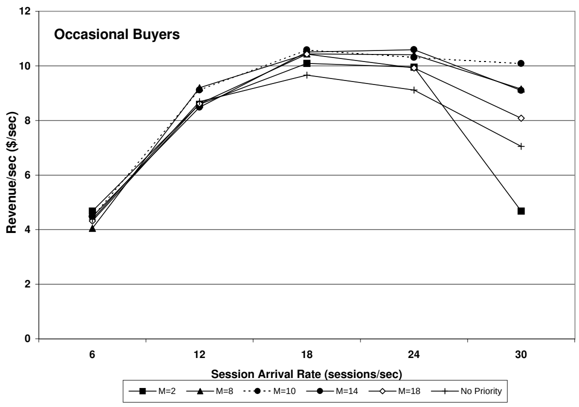

Figure 7: Revenue/sec vs. session arrival rate for occasional buyers.

leave the site due to poor performance in the no-priority case while for the priority case with $m_1 = 10$ only 37% leave the site for this reason.

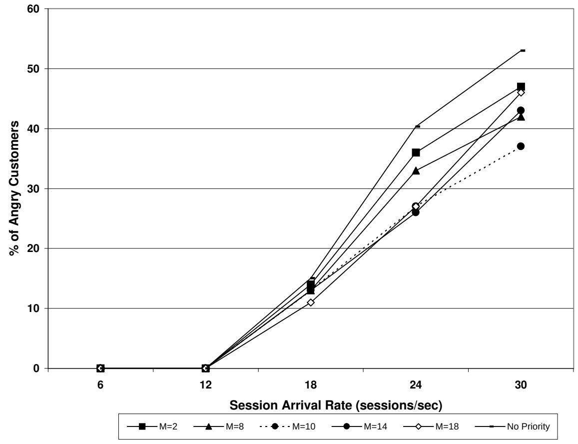

Figure 8: Percent of angry customers vs. session arrival rate

Figure 9 shows the potential lost revenue/sec X^- for all customers. This is the number of dollars/sec that were in customers' shopping carts and do not turn into sales because customers leave the site due to poor performance. As the figure indicates, with the priority scheme and all values of m_1 used, there is no potential lost revenue. However, for the no priority case, as the load increases, the potential loss in revenue increases. For $\lambda_s = 30$ sessions/sec, the potential lost revenue/sec is almost 50% of the combined revenue/sec brought to the Web store by heavy and occasional customers combined. An analysis of Figs. 6 and 9 shows that angry customers lost by the priority scheme do not have anything in their shopping carts and therefore do not contribute to X^-.

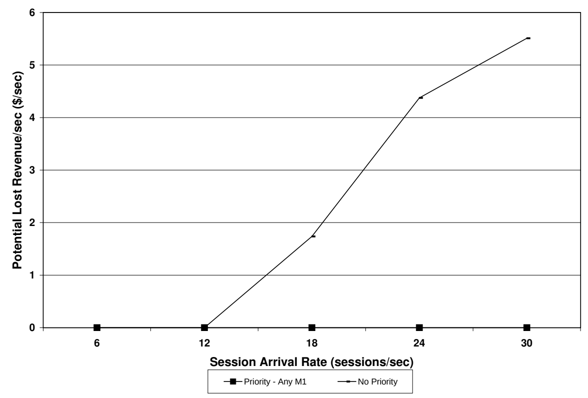

Figure 9: Potential lost revenue/sec vs. session arrival rate

Finally, Fig. 10 shows the average response time for all types of customers as a function of λ_s for the no priority case and for the priority case and different values of m_1. It is interesting to note that as λ_s increases, the average response time for the collection of all customers is higher for the priority case and values of $m_1 = 8, 10, 14$ and 18. Note that the improvements of the new metrics sometimes occur at the expense of the traditional performance metrics. Only the $m_1 = 2$ case presents a lower response time. The combined revenue of the site is maximized for $m_1 = 10$ for $\lambda_s = 30$ sessions/sec. However, for this value of λ_s and m_1 the average response time is virtually the worst among all cases. This illustrates the fact that optimizing conventional performance metrics may be detrimental to business-oriented goals and vice-versa.

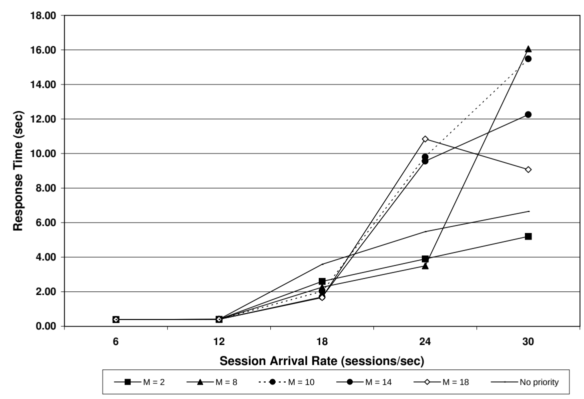

Figure 10: Response times vs. session arrival rate

7 Related Work

A few mechanisms have been developed to support quality of service in Web servers. All of them [1, 3, 9] aim at achieving good response time or high throughput. They are not adaptive to customer's behavior and they do not consider the new business-oriented performance metrics proposed in this paper. In [1], the authors propose a notion of quality of service by associating priorities with requests based on which document they are requesting, not where they come from. The metric for quality of service is latency in handling the HTTP requests. The priority mechanism is static and the study is restricted to priority schemes for CPU scheduling only. In [9], the authors develop a quality of service model that implements algorithms for scheduling CPU, memory and networking resources. In their model, a site can determine how requests for various pages should be served. Therefore, the notion of quality of service is implemented by setting priorities among page requests and by associating constraints on resource usage. The proposed mechanisms are clearly not oriented to e-commerce sites, which exhibit workloads that are quite different from streams of page requests. Our paper deals with new characterizations for e-commerce workloads and proposes resource control policies that are based on representations of customer's behavior and are oriented towards business goals.

An admission control mechanism to improve performance of overloaded Web servers is proposed and analyzed in [3]. The authors introduce the notion of session, consisting of many individual HTTP requests. The main goal of the session-based admission control mechanism is to prevent the overload of a Web server. The control scheme is based on the server CPU utilization. The analysis focuses only on the throughput gains obtained by an admission control mechanism that aims at guaranteeing the completion of any accepted session. Reference [4] presents an overall description of a Hewlett-Packard product (WebQoS) that implements the session-based admission control scheme proposed in [3]. The product also schedules HTTP requests into three different priority queues and controls the server resource allocation. Requests are classified into different levels of priority and can be reclassified to new priorities when specific URLs are requested.

Our work differs from the one proposed in [3, 4] in many ways. A fundamental difference is that our resource control mechanism is driven by business-oriented metrics (e.g., revenue/sec, lost_revenue/sec, etc) and do not try to optimize conventional metrics such as response time, throughput or server utilization. In contrast, our paper proposes resource management policies that aim at optimizing business-oriented metrics and, sometimes this can occur at the expense of traditional metrics, as shown in Fig 10. In fact, the metrics proposed in this paper are an aggregation of many QoS properties, such as timeliness, availability and security. Our metrics allow reasoning about and planning e-commerce sites at high levels of abstraction. Another difference is the way we characterize e-commerce workloads, as a combination of typical requests such as browse, search, select, add, and pay. A customer behavior model graph is used to represent different classes of customers and the dependencies existing among the several services of an e-commerce site. References [3, 4] treat an e-commerce workload as composed of sessions, that consists of many individual HTTP requests with no interdependence.

Other references analyze performance models of Web sites. The work in [5] proposes a workload characterization for e-commerce servers, where customers follow typical sequences of URLs as they move towards the completion of transactions. However, the authors do not propose any mechanism to guarantee quality of service of e-commerce servers. In [7], analytic models for analyzing web servers are presented. Finally, some experimental projects of operating systems have made use of economic and financial notions in the past. For example, the Spawn system [11] explores issues of fairness in resource distribution, currency as a form of priority, price equilibria, the dynamics of transients, and scaling in large distributed systems.

8 Concluding Remarks

In this paper we have introduced novel concepts for managing and evaluating e-commerce sites. We defined a set of new performance metrics to assess Web site performance in terms of business goals. Traditional quality of service models for the WWW are associated with two viewpoints: client-side performance that is usually measured by response time and server-side performance, represented by throughput. The metrics proposed here combine the two views. For example, *revenue throughput*, measured in dollars/sec, implicitly represents customer and site behavior. If a customer is happy with the site service, he/she would shop at the Web store and the revenue throughput would increase.

To address the e-commerce workload characterization issue, we introduced a state transition graph called Customer Behavior Model Graph (CBMG), that describes a customer session, i.e., a sequence of requests of different types. From the CBMG representation, we showed how to derive useful workload information, such as average number of visits to each state (i.e., that corresponds to a specific request) for each visit to the store's home page. Also the average session length and the *buy to visit ratio* were analytically obtained from the CBMG.

In order to maximize the revenue generated by a site and support its business goals, we proposed an **adaptive** scheduling policy. Priorities change dynamically as a function of the state a customer is in and as a function of the amount of money the customer has accumulated in his/her shopping cart. A detailed simulation model was developed to assess the gain of adaptive policies with respect to policies that are oblivious to economic considerations. Simulations results show that for an overloaded site, an increase of 29% in revenue could be obtained by the priority schemes of the adaptive policy, when compared to the no priority policy. We also show that the number of angry customers that leave a site due to poor performance could be decreased in 16% by the use of priority policies. The potential lost revenue could be reduced in 50% by the priority schemes. Overall, we observed a gain in the business metrics of an e-commerce site when adaptive policies for managing resources are used.

Acknowledgement

The authors would like to thank Paul Barford and Mark Crovella of Boston University for making SURGE available.

References

[1] J. Almeida, M. Dabu, A. Manikutty and P. Cao, Providing Differentiated Levels of Service in Web Content Hosting, *Proc. First Workshop on Internet Server Performance*, ACM SIGMETRICS 98, June 1998.

[2] P. Barford and M. Crovella, Generating Representative Web Workloads for Network and Server Performance Evaluation, in *Proc. 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Madison, July 1998.

[3] L. Cherkasova and P. Phaal, Session Based Admission Control: A Mechanism for Improving the Performance of an Overloaded Web Server, HPL-98-119, HP Labs Technical Reports, 1998.

[4] HP Enterprise Computing, http://www.useit.com/alertbox/990207.html

[5] D. Krishnamurthy and J. Rolia, Predicting the Performance of an E-Commerce Server: Those Mean Percentiles, in *Proc. First Workshop on Internet Server Performance*, ACM SIGMETRICS 98, June 1998.

[6] M. MacDougall, *Simulating Computer Systems: Techniques and Tools*, The MIT Press, 1987.

[7] Menascé, D. A., and V. A. F. Almeida, *Capacity Planning for Web Performance: metrics, models, and methods*, Prentice Hall, Upper Saddle River, NJ, 1998.

[8] Nielsen, J., http://www.useit.com/alertbox/990207.html

[9] R. Pandey, J. Barnes, R. Olsson, Supporting Quality of Service in HTTP Servers, in *Proc. Seventeenth Annual SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 1998.

[10] C. Shapiro and H. Varian, *Information Rules: a strategic guide to the network economy*, Harvard Business School Press, 1999.

[11] C. Waldspurger, T. Hogg, B. Huberman, J. Kephart and W. Stornetta, Spawn: A Distributed Computational Economy, *IEEE Trans. on Software Engineering*, Vol. 18, No. 2, 1992.