

Business-oriented 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

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

Virgilio A. F. Almeida
Dept. of Computer Science
Univ. Federal de Minas Gerais
Belo Horizonte, MG 30161
Brazil
virgilio@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 and shareholders 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 these dynamic policies with respect to policies that are oblivious to economic considerations. Simulation results show that the multilevel dynamic 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 importance of these results lies in the fact that e-commerce sites that use this approach will be able to improve revenue at peak times with the same server capacity.

1 Introduction

It has been recognized by many that congestion and poor performance can be the major impediments for the success of e-commerce. A recent report [24] estimated that US\$4.35 billion are lost each year in the United States due to unacceptable download times. 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.

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

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 business-oriented metrics such as 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. 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 [4], a workload generator for Web sites, augmented by a generator of e-commerce requests that mimic typical customer behavior. 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 dynamic priority scheme introduced 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 discusses how e-commerce related metrics can be obtained from CBMGs. The next section discusses how e-commerce workloads can be characterized as sets of CBMGs. Section six presents and analyzes the results of applying workload characterization methodologies to synthetic and real e-commerce logs. Section seven describes new resource management policies for e-commerce servers. The following section describes the simulator and the simulation environment used to analyze the new policies proposed. Section nine describes the numerical results obtained. Section ten compares our work with that of others. Finally, section eleven presents some concluding remarks.

2 Novel Metrics 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 its 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 frustrated customer is a quickly exiting customer. 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 The Nature of E-commerce Workloads

E-commerce workloads are composed of sessions. A *session* is defined as a sequence of requests of different types made by a single customer during a single visit to a site. Examples of requests for an online shopper are: browse, search, select, add to the shopping cart, user registration, and pay. An online trader would have different operations, such as: enter a stock order, research a fund, obtain real-time quotes, retrieve company profiles, and compute earning estimates. The allowed sequences of requests can be described by a state transition graph called *Customer Behavior Model Graph (CBMG)* [13, 14]. This graph has one node for each possible state (e.g., home page, browse (b), search (s), select (t), add (a), and pay (p)) 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 set 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 [21]. 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.

As an example, consider two customer 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.

Note that the CBMGs of Figs. 1 and 2 are just examples. CBMGs can have many other states, depending on the nature of the electronic business. The exit state is not explicitly represented in Figs. 1 and 2 to improve their readability. Transitions to the exit state are represented as arrows leaving the states of the CBMG. 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.

It is important to note that the CBMG is a characterization of the navigational pattern as viewed from the server side. This means that a transition from state i to state j is said to occur when the request to go to state j arrives at the server. Therefore, user requests that are resolved at the browser cache or at a proxy server cache are not seen by the e-commerce server and therefore are not reflected in the CBMG. However, these requests do not use resources of the e-commerce site and therefore do not have to be considered for server sizing and capacity planning purposes [15, 16]. In these cases, it is important to capture the load imposed on the server resources by the various requests submitted by a customer. It should also be noted that, in the case of e-commerce, many pages that are intrinsically static, are generated dynamically, and therefore not cached, because they contain advertisement. The problem of relying solely on server side

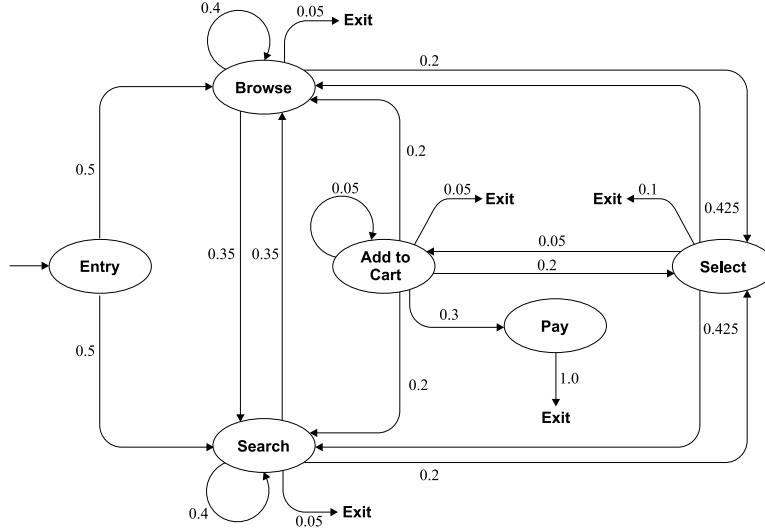


Figure 1: Customer Behavior Model Graph for an occasional buyer.

information is discussed in [23] in the context of using data mining and online analytic processing techniques (OLAP) on HTTP logs. The authors conclude that despite the fact that server side information is not 100% complete, much useful information can be discovered from them.

Another aspect of the nature of the e-commerce workload is the workload intensity. This aspect of the workload is characterized by two main parameters:

- the arrival rate of session initiation requests, measured in sessions/sec, for each type of session. Different session types are characterized by different CBMGs. These arrival rates reflect the mix of customers who use the site. It has been observed in many e-commerce sites that occasional buyers constitute a very large percentage of all customers. In fact, the percentage of customers who end up buying, i.e., visiting the “pay” state has been found to be around 5% [17].
- the average server-perceived think time (Z_s) between state transitions of the CBMG. This is defined as the average time elapsed since the server completed a request for a customer until it receives the next request from the same customer within the same session. Z_s is given by $2 \times nt + Z_b$ where nt is the network time between client and server and Z_b is the browser-side think time. From now on, the server-perceived think time will be simply referred to as think time. A think time can be associated with each transition in the CBMG. Think times are not shown in Figs. 1 and 2. No think times are associated with transitions to the exit state.

So, a CBMG can be more formally characterized by a pair (P, Z) where $P = [p_{i,j}]$ is an $n \times n$ matrix of transition probabilities between the n states of the CBMG and $Z = [z_{i,j}]$ is an $n \times n$ matrix that represents the average think times between states of the CBMG. A matrix similar to the transition probability matrix P is described in [23] where Online Analytical Processing (OLAP) and Data Mining techniques are used to analyze Web logs. We adopt the convention that state 1 is always the entry (y) state and state n is always the exit (e) state. Note that the elements of the first column and last rows of matrix P are all zeroes since, by definition, there are no transitions back to the entry state from any state nor any transitions out of the exit state.

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 and by the arrival behavior of customers of that group as discussed in section 5. CBMGs are also useful for answering what-if questions regarding

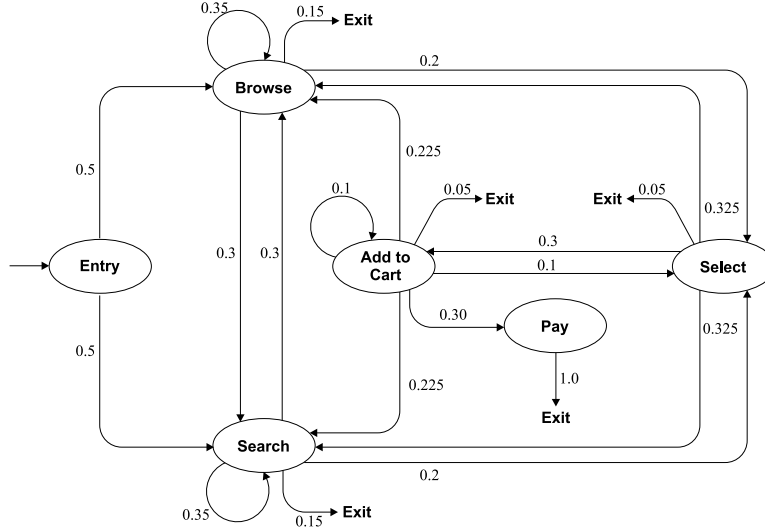


Figure 2: Customer Behavior Model Graph for a heavy buyer.

the impact of site layout changes on the site performance and business metrics. For example, a change in site layout could be reflected by an increase in the probability of going from the state search to select.

4 Metrics Derived from the CBMG

The CBMG provides useful information regarding the average number of visits V_j to each state of the CBMG for each visit to the site. The values of V_j can be obtained by solving the system of linear equations:

$$\begin{aligned} V_1 &= 1 \\ V_j &= \sum_{k=1}^n V_k \times p_{k,j} \quad \text{for } j = 2, \dots, n. \end{aligned} \quad (1)$$

The system of linear equations in (1) can be written in vector form as $\vec{V} - \vec{I} = \vec{V} \times P$ where $\vec{I} = (1, 0, \dots, 0)$. Note that since $p_{n,k} = 0 \forall k = 1, \dots, n$, $V_n = 1$. For the CBMGs of Figs. 1 and 2, this system of linear equations becomes

$$\begin{aligned} 1 &= V_y \\ p_{y,b} V_y + p_{b,b} V_b + p_{s,b} V_s + p_{a,b} V_a + p_{t,b} V_t &= V_b \\ p_{y,s} V_y + p_{b,s} V_b + p_{s,s} V_s + p_{a,s} V_a + p_{t,s} V_t &= V_s \\ p_{a,a} V_a + p_{t,a} V_t &= V_a \\ p_{b,t} V_b + p_{s,t} V_s + p_{a,t} V_a &= V_t \\ p_{a,p} V_a &= V_p \\ p_{b,e} V_b + p_{s,e} V_s + p_{a,e} V_a + p_{t,e} V_t + p_{p,e} V_p &= V_e \end{aligned} \quad (2)$$

where $V_y, V_b, V_s, V_a, V_t, V_p$, and V_e are the average number of visits to states Entry, Browse, Search, Add to Cart, Select, Pay, and Exit, respectively. Note that for each CBMG, we have a different system of linear equations because the transition probabilities are different. The solutions to the system of linear equations $\vec{V} - \vec{I} = \vec{V} \times P$ for the graphs of Figs. 1 and 2 are $(V_y = 1, V_b = 6.76, V_s = 6.76, V_a = 0.14, V_t = 2.73, V_p = 0.04, V_e = 1)$ and $(V_y = 1, V_b = 2.71, V_s = 2.71, V_a = 0.37, V_t = 1.12, V_p = 0.11, V_e = 1)$, respectively.

Useful metrics can be obtained from \vec{V} . The first, called *average session length* and denoted by \bar{S} , is the average number of states visited by a customer per visit to the Web store. Thus, $\bar{S} = \sum_{k=1}^{n-1} V_k$. The average session length for occasional (o) and heavy (h) 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, BV is given by V_p , the average number of visits to the pay state per visit to the site. Suppose that in the case of Figs. 1 and 2, 90% of customers who initiate sessions are occasional buyers. Then, $BV = 0.9 \times 0.04 + 0.1 \times 0.11 = 0.047$.

5 Characterizing E-commerce Workloads Based on CBMGs

With the growing importance of performance for e-commerce sites, comes an increasing need to characterize and model their workloads. A workload model can be: i) a trace of an actual e-commerce site, ii) an artificially generated stream of requests that mimic a population of customers, or iii) a set of parameters that represent the resource usage of an actual workload. Resource usage-oriented workload models are useful for performance modeling and capacity planning purposes. We describe below, with the help of Fig. 3, the steps needed to characterize the workload of an e-commerce site in terms of CBMGs [13]:

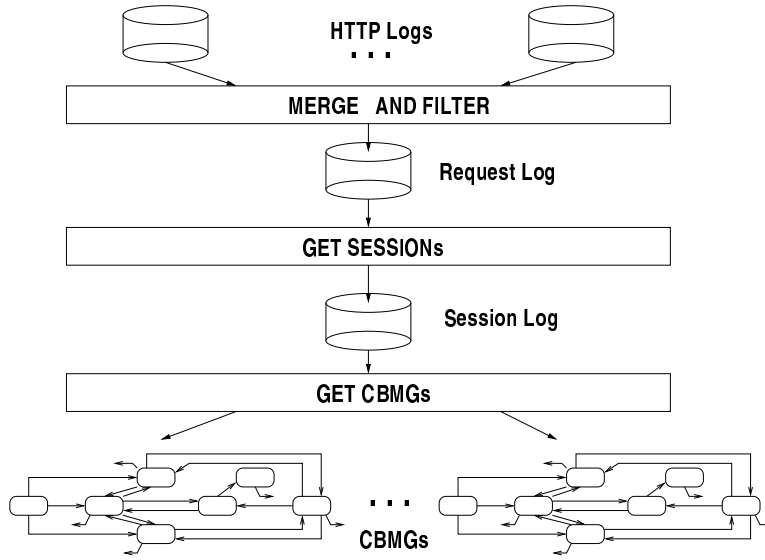


Figure 3: Workload characterization methodology.

The first step consists of merging and filtering HTTP logs to discard irrelevant entries such as errors and others. If a site has more than one HTTP server, more than one HTTP log will be generated. All HTTP logs must be merged into a single log using the timestamp. Clock synchronization services such as the ones available in Linux and NT can be used to facilitate merging of distributed logs. The result of this step is a request log denoted by \mathcal{L} . Each line in \mathcal{L} is assumed to have the following information (uid, request_type, request_time, exec_time), where

- uid is an identification of the customer submitting the request. Cookies, dynamic URLs, or even authentication mechanisms can be used to uniquely identify requests as coming from the same browser during a session [20].
- request_type indicates the type of request. Examples include a GET on the home page, a browse

request (i.e., a GET on another page), a request to execute a search, a selection of one of the results of a search, a request to add an item to the shopping cart, or a request to pay.

- `request_time` is the time at which the request arrived at the site.
- `exec_time` is the execution time of the request. Even though this value is not normally recorded in the HTTP log, servers can be modified to record this information.

The next step takes as input the request log and generates a session log \mathcal{S} . The k -th entry in the log \mathcal{S} is composed of the tuple (C_k, W_k) where $C_k = [c_{i,j}]$ is an $n \times n$ matrix of transition counts between states i and j of the CBMG for one session, and $W_k = [w_{i,j}]$ is an $n \times n$ matrix of accumulated think times between states i and j of the CBMG for one session. To illustrate the notation, consider that for a given session, there were 3 transitions between states s and t , and that the think times for each of the transitions were 20 sec, 45 sec, and 38 sec, respectively. Then, $c_{s,t} = 3$ and $w_{s,t} = 20 + 45 + 38 = 103$ sec.

Once the session log \mathcal{S} is generated, a clustering analysis is performed on it to generate a synthetic workload composed of a relatively small number of CBMGs. The centroid of a cluster determines the characteristics of the CBMG. Any number of clustering algorithms could be used. We used the k -means clustering algorithm [7, 8, 15, 16] to characterize e-commerce workloads based on synthetic and real logs as discussed in the next section. Clustering algorithms require a definition of a distance metric to be used in the computation of the distance between a point and a centroid. Assume that the session log is composed of M points $X_m = (C_m, W_m)$, $m = 1, \dots, M$ where C_m and W_m are the transition count and accumulated think time matrices defined above. Our definition of distance is based on the transition count matrix only since this is a factor that more clearly defines the interaction between a customer and an e-commerce site. We define the distance $d_{a,b}$ between two points X_a and X_b in the session log as the Euclidean distance

$$d_{a,b} = \sqrt{\sum_{i=1}^n \sum_{j=1}^n (C_a[i,j] - C_b[i,j])^2}. \quad (3)$$

6 Results of Workload Characterization

We conducted experiments aimed at characterizing e-commerce workloads based on CBMGs using both synthetic and real log. Synthetic logs were generated by generating the first request of a session using SURGE [4] and then generating the following requests for that session from CBMGs that represent different user types. The details about the experiments are reported in [13]. In this section, we report on the results of these experiments.

6.1 Experiments with Synthetic Logs

We obtained six clusters from our synthetic logs. The `GetSessions` procedure (see Fig. 3) was applied to the request log \mathcal{L} with 340,000 lines representing HTTP operations. The number of sessions identified by `GetSessions` in log \mathcal{L} is 20,000. Table 1 reports the values of some parameters of the centroids of the six clusters obtained from the analysis of the 20,000 sessions. The first line shows the percentage of sessions that fall into each cluster. For instance, cluster 1 represents almost half of all the sessions. Line 2 shows the *Buy to Visit Ratio* (BV). Session length indicates the average number of shopper operations requested by a customer for each visit to the electronic store. Line 5 exhibits the *Add to Shopping Cart Visit Ratio* (V_a), that represents the fraction of times a customer adds an item to his/her shopping cart. However, this operation does not necessarily imply a buy operation, as can be noticed from the comparison between its values and the BV's values. The last line of the table indicates the number of browsing and searching operations associated with customers of each cluster. The natural question that arises now is: What kind of conclusions can we draw from the above characterization of the e-commerce workload?

Cluster	1	2	3	4	5	6
% of the Sessions	44.28	28	10.6	9.29	6.20	1.5
BV Ratio (%)	5.7	4.5	3.7	4	3.5	2
Session Length	5.6	15	27	28	50	81
AV Ratio (%)	11	15	21	20	32	50
$V_b + V_s$	3.6	11.4	20	23	39	70

Table 1: Resulting Cluster Attributes

In the sample we analyzed, we can note two very different behavior patterns. Cluster 1, that represents the majority of the sessions (44.28%) has a very short average session length (5.6) and the highest percentage of customers that buy from the store. On the other extreme, we notice that cluster six represents a small portion of the customers, that exhibit the longest session length and the smallest buying ratio. In order to try to correlate these parameters, we plotted in Fig. 4 the percentage of customers who buy as a function of the average session length. We can observe that for this sample, an interesting pattern was found: the longer the session, the less likely it is for a customer to buy an item from the Web store.

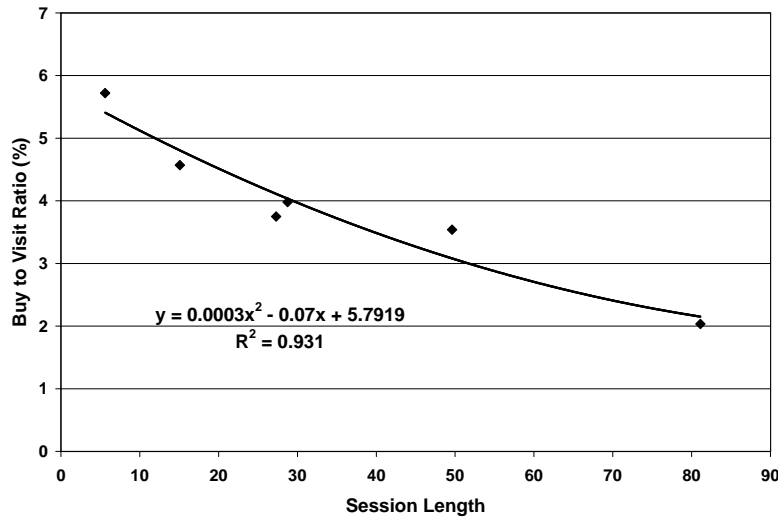


Figure 4: Buy to Visit Ratio vs. Session Length

6.2 Experiments with Real Logs

Obtaining HTTP logs from actual e-commerce sites can be a challenge since these logs may contain information that is quite revealing about the nature and degree of success of the business. We were able to obtain an HTTP log file from an e-commerce company that was kind enough to provide us with a sanitized version of their logs. Due to a non-disclosure agreement, we cannot name the company nor provide information on sale-related matters. Notwithstanding, we provide in this section some results of the analysis we carried out on these logs.

We worked with six log files with a total of 17,727,573 HTTP requests. After eliminating requests for images, we were left with 6,004,813 HTTP requests. These are requests for HTML files and execution of

cgi scripts. After running the GetSessions algorithm on 628,573 requests, we identified a significant number of very small sessions and a few very large sessions, due to accesses by robot agents. In order to analyze meaningful sessions, we decided to work with sessions whose length was greater than 3 and smaller than 20 requests. As a consequence, we were able to identify 34,811 sessions, described by CBMGs, with an average size of 8.46 requests each for one of the logs that had almost 295,000 filtered requests. We noticed from the clustering analysis of the real logs that the buy probability tends to decrease as sessions get longer. It is interesting to observe that the same result was found in the analysis of the synthetic logs (see Fig. 4).

7 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.

7.1 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.

7.2 E-commerce Oriented Resource Management 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 revenue throughput.

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. 5.

The transitions between priority classes are labeled by conditions which may depend on the state visited, the session length S , and the accumulated money in the shopping cart $\$sc$. As shown in the figure, all users entering the Web store start at high priority and remain there if they add at least one item to their shopping cart, in which case $\$sc > 0$, 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

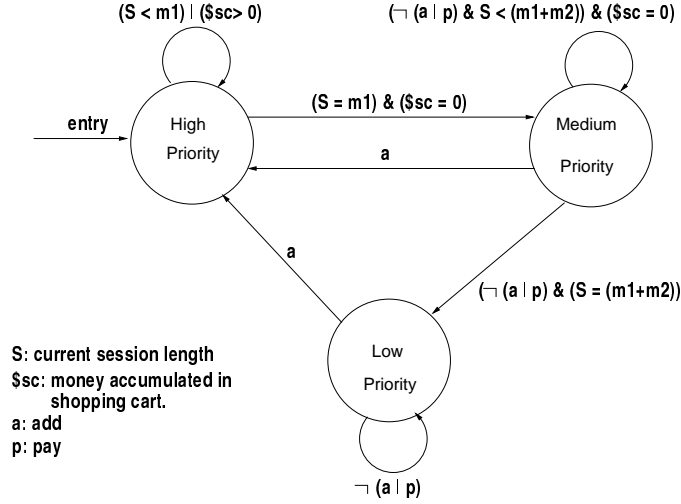


Figure 5: Priority scheme for e-commerce sites.

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. 5 reduces to a single-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.

The family of priorities discussed above can be generalized to a multilevel priority scheme in which there are P priority levels with 1 being the highest priority and P the lowest as shown in Fig. 6. In this case, there are $P - 1$ thresholds m_1, \dots, m_{P-1} . Transitions to the highest priority can occur from any of the other priority levels as soon as customers add items to their shopping carts.

8 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. 7, 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.

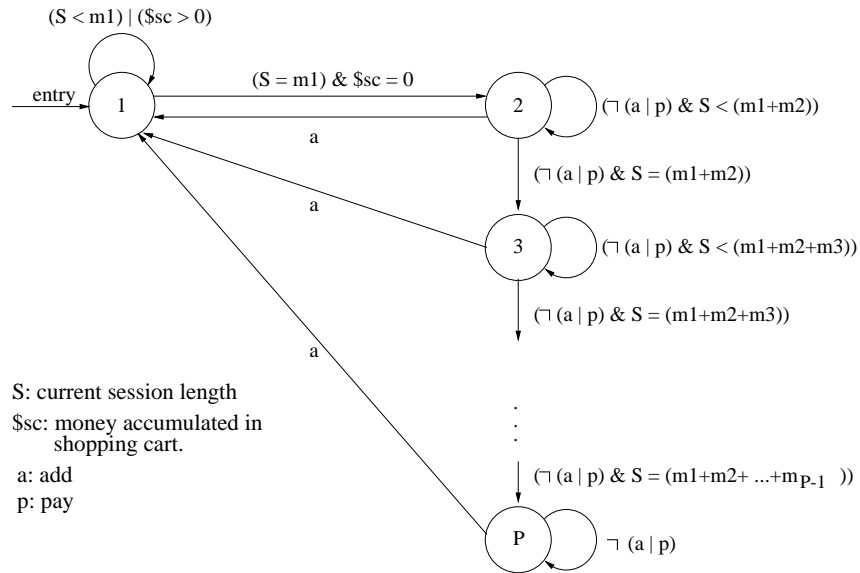


Figure 6: Multilevel priority scheme for e-commerce sites.

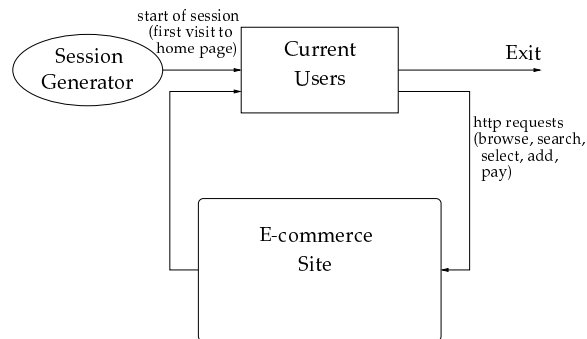


Figure 7: Overview of the e-commerce site simulator.

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.

8.1 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 [4] 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. 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

$$timeout = c_2(state) + c_1 \times session_length \quad (4)$$

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.

8.2 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 2 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 2 were obtained by analyzing results of a large number of 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.

The architecture of the e-commerce site for our bookstore is shown in Fig. 8. It is composed of one authentication server, 3 HTTP servers, 3 application 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.

Table 3 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 as $D_{CPU}(BD, BT) = 0.00258 \times (BD + BT) + 0.0027$ 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 4 shows the site’s components involved in the processing of the requests associated with each CBMG state. The simulation model was developed in C and made use of the discrete event simulation environment SMPL [12]. 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

Think Times: 15 sec for all transitions, except for: Select to Add: 45 sec; Add to Pay: 25 sec; Search to Select: 30 sec										
Percent of Technical Books Selected: 20%										
Average Book 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:										
	<table border="1"> <thead> <tr> <th></th> <th>Technical</th> <th>Non-Technical</th> </tr> </thead> <tbody> <tr> <td>By Keyword</td> <td style="text-align: center;">20</td> <td style="text-align: center;">25</td> </tr> <tr> <td>By Author</td> <td style="text-align: center;">11</td> <td style="text-align: center;">16</td> </tr> </tbody> </table>		Technical	Non-Technical	By Keyword	20	25	By Author	11	16
	Technical	Non-Technical								
By Keyword	20	25								
By Author	11	16								
Avg. size (in KB) of pages returned by browse requests: 20										

Table 2: Workload parameters.

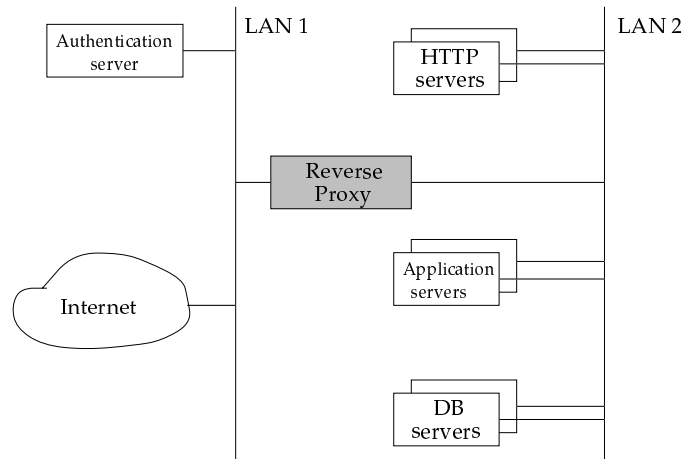


Figure 8: Architecture of the Web store.

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.

9 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 throughput (X^+), in \$/sec, for occasional and heavy buyers, 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

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

Table 3: 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 4: Components of the site configuration involved in the request execution.

throughout the simulations reported here is 4. The value of m_1 varied as shown in the figures.

Figs. 9 and 10 show the variation of the revenue throughput 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. 9 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.

For occasional buyers, as seen in Fig 10, 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. 9 and 10 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. 9 can be better understood by analyzing Fig. 11 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 leave the site due to

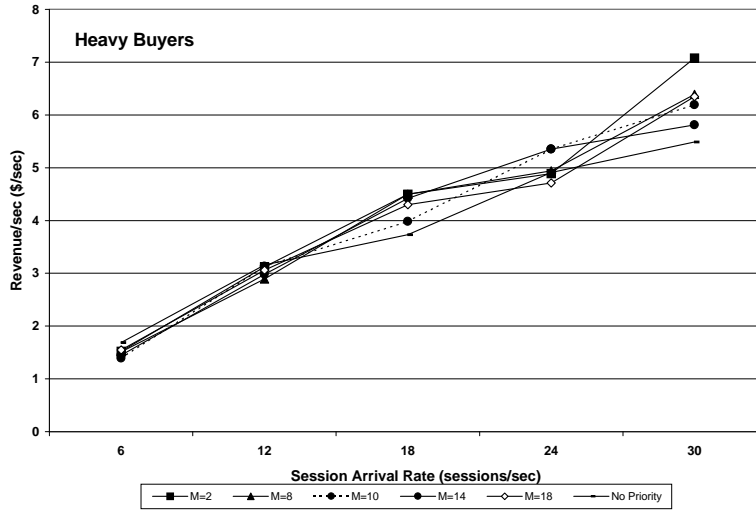


Figure 9: Revenue/sec vs. session arrival rate for heavy buyers.

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 12 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. 9 and 12 shows that angry customers lost by the priority scheme do not have anything in their shopping carts and therefore do not contribute to X^- .

Finally, Fig. 13 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.

10 Related Work

A few mechanisms have been developed to support quality of service in Web servers. All of them [1, 6, 18] 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

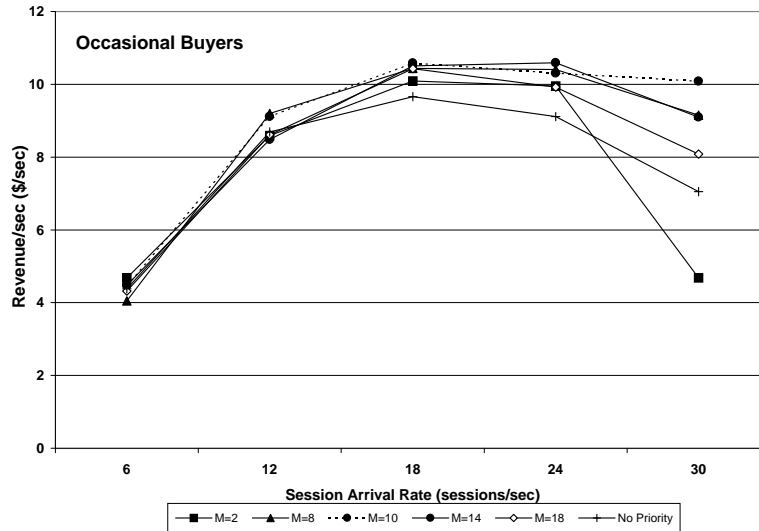


Figure 10: Revenue/sec vs. session arrival rate for occasional buyers.

for CPU scheduling only. In [18], 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 [6]. 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 [10] presents an overall description of a Hewlett-Packard product (WebQoS) that implements the session-based admission control scheme proposed in [6]. 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 [6, 10] 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 does 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 13. 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 [6, 10] treat an e-commerce workload as composed of sessions, that consists of many individual HTTP requests with no interdependence.

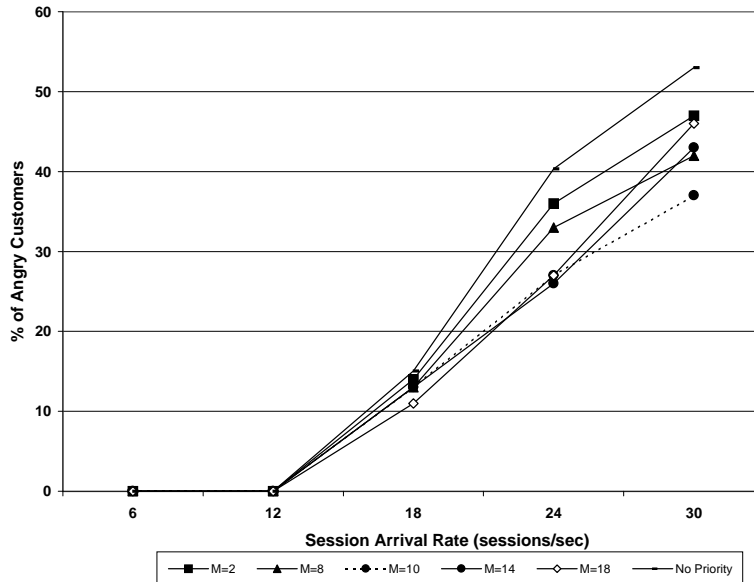


Figure 11: Percent of angry customers vs. session arrival rate

Other references analyze performance models of Web sites. The work in [11] 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 [15], 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 [22] 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.

11 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 the performance of e-business sites in terms of business goals. Traditional quality of service models for the WWW are associated with two viewpoints: client-side performance, usually measured by response time, and server-side performance, usually 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 will shop at the Web store and the revenue throughput will 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 related requests of different types. From the CBMG representation, we showed how to analytically derive useful metrics, such as average number of visits to each state per visit to the store, average session length, and the *buy to visit ratio*. CBMGs are also useful for answering what-if questions regarding the impact of changes in site layout on the site performance and business metrics. The paper described how clustering techniques can be applied to e-commerce logs in order to characterize e-commerce workloads in terms of CBMGs. These characterizations are useful in establishing useful relationships between observed metrics. For example, we

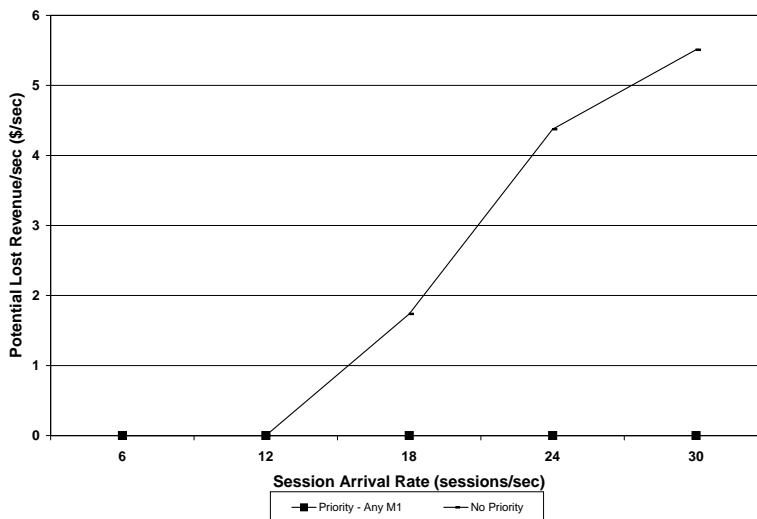


Figure 12: Potential lost revenue/sec vs. session arrival rate

observed in both synthetic and real logs that the buy to visit ratio decreases significantly with session length.

In order to maximize the revenue generated by a site and support its business goals, we proposed a dynamic multilevel resource management 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. The importance of the results discussed in this paper lies in the fact that e-commerce sites that use our dynamic priority scheme will be able to improve revenue at peak times with the same server capacity.

The priority scheme suggested here requires modification to Web server, operating system, and DBMS software. While this could be seen as a drawback, many e-commerce sites already rely on open source software such as Apache and Linux. This makes it easier to implement the techniques discussed here. Developers of software for e-commerce may be interested in implementing these technologies in their products.

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.

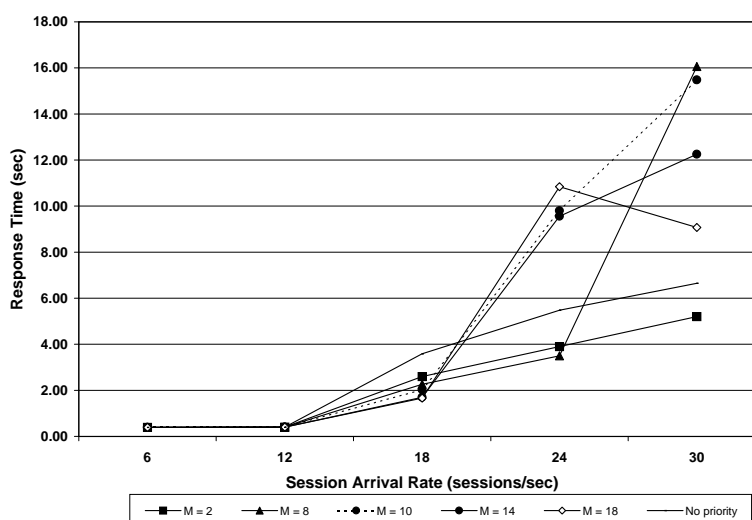


Figure 13: Response times vs. session arrival rate

- [2] V. Almeida, N. Ziviani, V. Ribeiro, and W. Meira, Efficiency Analysis of Brokers in the Electronic Marketplace, *Proc. of the 8th International World Wide Web Conference*, Toronto, May 1999.
- [3] M. Arlitt and C. Williamson, Web server Workload Characterization: the search for invariants, in *Proc. 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Philadelphia, May 1996.
- [4] 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.
- [5] M. Calzarossa and G. Serazzi, Workload Characterization: A Survey, *Proceedings of the IEEE*, Vol. 81, No. 8, August 1993.
- [6] 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.
- [7] B. Everitt, *Cluster Analysis*, Halsted Press, New York, 1980
- [8] D. Ferrari, G. Serazzi, and A. Zeigner, *Measurement and Tuning of Computer Systems*, Upper Saddle River, Prentice Real, 1983.
- [9] GVVU's WWW User Surveys, http://www.gvu.gatech.edu/user_surveys/
- [10] HP Enterprise Computing, <http://www.useit.com/alertbox/990207.html>
- [11] 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.
- [12] M. MacDougall, *Simulating Computer Systems: Techniques and Tools*, The MIT Press, 1987.
- [13] Menascé, D. A., V. A. F. Almeida, R. Fonseca, M. A. Mendes, A Methodology for Workload Characterization of E-commerce Sites, *Proc. 1999 ACM Conference on Electronic Commerce*, Denver, CO, November, 1999.

- [14] Menascé, D. A., V. A. F. Almeida, R. Fonseca, M. A. Mendes, Resource Allocation Policies for E-commerce Servers, in *Proc. Second Workshop on Internet Server Performance*, in conjunction with ACM Sigmetrics'99, Atlanta, GA, May 1, 1999.
- [15] Menascé, D. A., and V. A. F. Almeida, *Capacity Planning for Web Performance: metrics, models, and methods*, Prentice Hall, Upper Saddle River, NJ, 1998.
- [16] Menascé, D. A., V. A. F. Almeida, L. W. Dowdy, *Capacity Planning and Performance Modeling: from mainframes to client-server systems*, Prentice Hall, Upper Saddle River, 1994.
- [17] Nielsen, J., <http://www.useit.com/alertbox/990207.html>
- [18] 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.
- [19] Pitkow, J. and P. Pirolli, Mining Longest Repeating Subsequences to Predict World Wide Web Surfing, *Proc. 2nd. Usenix Symposium on Internet Technologies and Systems*, Boulder, CO, October 1999.
- [20] Treese, G. W. and L. C. Stewart, *Designing Systems for Internet Commerce*, Addison Wesley, Reading, MA, 1998.
- [21] C. Shapiro and H. Varian, *Information Rules: a strategic guide to the network economy*, Harvard Business School Press, 1999.
- [22] 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.
- [23] Zaiane, O. R., M. Xin, and J. Han, Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs, *Proc. Advances in Digital Libraries Conf. (ADL'98)*, Santa Barbara, CA, April 1998, pp. 19–29.
- [24] Zona Research, The Economic Impacts of Unacceptable Web Site Download Speeds, White Paper, 1999 (www.zonaresearch.com).