Admission Control Schemes Guaranteeing Customer QoS in Commercial Web Sites

Maria Kihl and Niklas Widell

Department of Communication Systems, Lund Institute of Technology, Sweden

Abstract Many commercial web sites, as web stores, have recently experienced performance problems due to the growth of Internet trading. One way to improve a site's performance during overload is to introduce admission control mechanisms. In this paper we develop and investigate two admission control schemes specifically for distributed commercial web sites. One of the schemes is request-based and the other one is session-based. A queuing network model is used to investigate a distributed site representing a web store. We find that both schemes improve the site's performance during overload. However, while the session-based control scheme guarantees a good customer QoS, the request-based scheme generates a large amount of so called angry customers.

1. INTRODUCTION

Commercial web services, so-called e-commerce services, are becoming an important part of the Internet. Some examples are web stores, e-traders, web auctions and Internet banks. Unfortunately, many problems related to performance have been recognized for commercial web sites. There is a tendency that customers leave and never come back to sites that perform poorly.

The concept of customer session is of particular interest for commercial web sites. During a session, the customer sends a number of HTTP GET requests to the web site. For a commercial web site to be successful, it is obviously important that as many customers as possible complete their sessions since only a customer that completes his/her session may generate some revenue.

Studies show that one of the main QoS demands that customers have on a commercial web site is short response times, that is the time it takes to download a page (Nielsen [15], Bhatti *et al.* [5]). If the site is dimensioned correctly this will not be a problem during normal traffic flows. However a popular web store may receive too much traffic during sales or promotions. Internet banks usually have a high offered load at the end of each month. When more customers arrive than the site is designed for, the response times will increase which means that customers may start to abandon the site. To solve this problem, admission control mechanisms must be implemented in order to guarantee a high performance.

The objective of the admission control mechanism is to maintain an acceptable load in the server cluster even when the arrival rate is above the site's capacity. The mechanism can be based on two basic principles: rejection of

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35620-4_43

D. Gaïti et al. (eds.), *Network Control and Engineering for QoS, Security and Mobility*

requests or content adaptation. In the first case, some requests are rejected during overload. In the second case, the site delivers less resource intensive content to the customers during overload. However, a content adaptation mechanism must always be used in combination with a rejection mechanism. Therefore, we will in this paper only discuss rejection based admission control mechanisms. For more details about content adaptation, see Abdelzaher and Bhatti [1].

An admission control mechanism with rejections can either be request-based or session-based. In a request-based mechanism there is an upper limit to the number of requests processed at the same time in the site, whereas a session-based mechanism limits the number of ongoing customer sessions.

Only a few papers have investigated admission control mechanisms for web sites. Iyengar [9] analysed a web server and found it necessary to have an admission control in order to obtain good performance. Bhatti and Friedrich [4] investigated a simple request-based admission control scheme. Abdelzaher *et al.* [2] used basic control theory to investigate a request-based scheme in combination with content adaptation. Cherkasova and Phaal [8] developed a session-based scheme. Bhoj *et al.* [6] developed a middleware implementation with session-based admission control.

The schemes above use the server utilization or queue length to detect overload. However, server utilization is not directly related to the response time, since a bursty arrival process may cause long delays even if the average server utilization is low (see Kleinrock [10]). Lu *et al.* [11] proposed a control scheme guaranteeing so called relative delays. However, this scheme requires a single Apache web server and it is not a real admission control scheme since it only change the number of processes allocated for each customer class. The scheme is not tested during overload.

Further, the papers above only consider single web servers, which mean that the suggested control schemes may not be suitable for distributed sites. In a distributed web site it may be difficult to have any detailed assumptions about the back-end server architecture. This means that control variables as server utilization and queue length may be difficult to use in an admission control scheme.

Therefore, we propose an admission control scheme specifically for distributed commercial web sites. The objectives of the scheme are first to protect the site from breaking down and second to guarantee that the customer QoS demands are obeyed. The proposed scheme use the processing delay, that is the time it takes for the site to process an HTTP request, as control variable. The processing delay is directly related to the customers' response times, since the response time consists of a network delay and a processing delay.

Then we use simulations to investigate and compare one request-based and one session-based version of the proposed control scheme. We show that both schemes improve the site's performance during overload periods. Further, we show that the throughputs of completed sessions are similar. However, with a request-based scheme the number of so-called *angry* customers increases rapidly when the offered load increases. Each angry customer represents a revenue loss for the site. With a session-based scheme the number of angry customers may be kept at a minimum level.

2. DISTRIBUTED COMMERCIAL WEB SITES

A distributed web site can be divided into a front-end system and a back-end system, see Figure 1. The front-end system has direct communication with the customer. The front-end then communicates with the back-end, if necessary. The back-end is invisible to the customer.

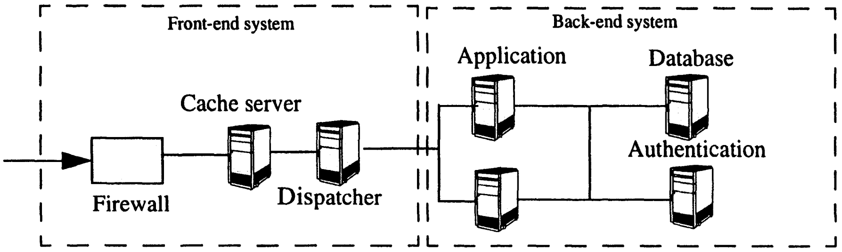

Figure 1. General structure of an e-commerce site.

The front-end part consists of a firewall, a cache server and a dispatcher. The firewall works at wire speed and has no impact on performance. The cache server caches static web objects, such as plain HTML pages and pictures. This means that only requests for dynamic web pages reach the other servers. The dispatcher distributes incoming requests to the back-end servers using some load balancing algorithm. If there is an admission control mechanism in the site, it is placed in the dispatcher.

The back-end servers on the inside of the dispatcher are typically connected with one or two high-speed LANs, such as a switched Fast Ethernet. The application servers receive the dynamic HTTP requests from the customers. They reply the HTTP request by returning the requested page or by running requested server side scripts. If a script requires the services of one of the other back-end servers, the application server generates and sends a new request to the respective server. The database server has one or more databases containing customer data, inventory, catalogues and other data. The authentication server is used to verify the identity of a customer.

3. CUSTOMER QOS

It is important to know what the customers expect when they visit a commercial web site, that is the customers' *QoS demands*. From a customer's point of view, a number of general QoS issues have been recognized as important, see for example [5][15]. One important QoS issue for the customers is the site's performance. If customers find that the site they have entered have performance problems they may abandon the site and do their business somewhere else instead. For a customer, good performance means that the response times, i.e. the time to download a page, are short.

One fact is that most customers visiting a commercial web site, for example a web store, will not make a purchase. Studies show that as few as 5% of the customers buy something when they visit a web store [15]. However, it is important to take care of all types of customers since every customer who visits the site is a potential buyer (now or in the future).

Also, customers believe that if the site is heavily loaded it is important to receive information about this rather than to just see the long response times [5]. If a customer must be rejected, a discount or some kind of 'coupon' should be offered as an incentive to go back to the site. A customer experiencing long response times without getting any warning may hesitate to visit the site again. Even customers making a purchase may not visit the site again if they believe that the response times are too long.

One way to measure the customer QoS is to measure the rate of so called *happy* and *angry* customers (Menascé *et al.* [14]). A happy customer completes his/her session and may thereby generate some revenue for the site. On the other hand, customers that are rejected or abandon the site in the middle of their sessions are defined as angry customers. These customers have already spent some of their time in the site without being able to finish their sessions. Therefore they may hesitate to visit the site again. Those customers that are rejected before they have spent any time in the site are not classified as angry. We assume that the site informs them about the load situation and rewards them if they come back another time.

4. QUEUING NETWORK MODEL

In this section a queuing network model for a distributed commercial web site is described. The site includes one dispatcher, M application servers (APP), one database server (DB) and one authentication server (AS). The firewall and the cache server in Figure 1 are not modelled since they are seldom the performance bottlenecks. The servers are connected with two high-speed LANs. We assume that the LANs do not cause any delays.

4.1. The Dispatcher

The dispatcher is modelled as in Figure 2. Incoming requests are placed in a queue. Each request is parsed in order to identify the customer session (see [3] for more details about session identification). After that, the request is sent to the admission control mechanism. All admitted requests are then transferred to the load balancer, which distributes the requests to one of the application servers. The processing time for a request is denoted x_f. We assume that the admission control and load balancing algorithms are simple enough not to cause any heavy processing in the server.

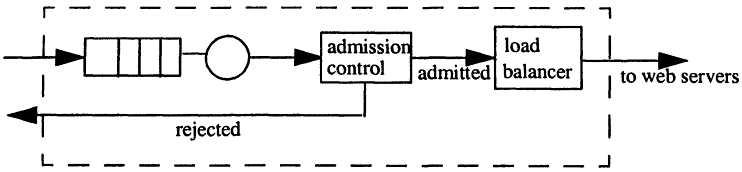

Figure 2. A model of a dispatcher.

4.2. The Back-end Servers

The back-end servers are modelled as single servers with priority queues. Each back-end server executes a specific operation, for example an application server executes dynamic scripts. The processing time for an operation is denoted x_{APP}, x_{DB} and x_{AS} depending on the server that executes the operation. A request may need several operations before completed. In this case, the request is sent between the required servers. We assume that the transmission times between the servers are close to zero. When a request has been completed, the reply is sent directly to the customer.

5. CUSTOMER BEHAVIOUR MODELS

A customer behaviour model describes how the customers send requests to the site. The model should accurately mimic the arrivals of customer requests.

The arrival of new customers can be modelled as a stochastic process. Paxson and Floyd [16] show that within one-hour intervals, the Poisson process may be used to model user-initiated session arrivals in the Internet. They analyse FTP and TELNET sessions, however the result should also be applicable to customer arrivals at commercial sites.

During a session, the customer sends HTTP requests for varying types of data. There are a limited number of request types. In for example a web store a typical set of requests may be Home, Search, Browse, Select, Add and Pay. When a customer has sent a request to the site, he/she waits for a reply. When the reply has arrived, the customer either sends a new request or decides to leave the site.

It is important to develop accurate models for the session length distribution. In [8] the session lengths, that is the number of requests a customer sends during a session, were exponentially distributed. In [6] each session included 40 requests. Menascé *et al.* [13] develop a so-called Customer Behaviour Model Graph (CBMG). In this model, the session lengths are almost geometrically distributed. Arlitt *et al.* [3] measures the customer behaviour in a real e-commerce site. From their results we have proved that the geometrical distribution is an accurate model for the customer session length.

We have used a customer behaviour model where the customer sends requests according to a probability distribution. A customer sends a request of type *y* with probability *p(y)*. With probability *p(leave)* the customer instead leaves the site. Between each request, the customer has a think time. With this model, the session lengths become geometrically distributed.

A customer that has sent a request of type y will wait S_y seconds. If the reply has not arrived after this time, the customer becomes impatient and abandons the site by pressing the stop button. One problem in web sites is how to detect these customers, denoted timed-out customers. If the site processes a request belonging to a timed-out customer, the processing is wasted. Carter and Cherkasova [7] develop a method for detecting timed-out customers. Therefore, we assume that timed-out customers can be detected.

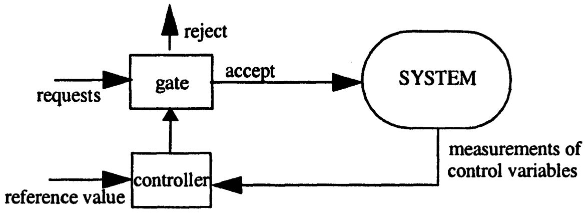

Figure 3. An admission control mechanism

6. ADMISSION CONTROL MECHANISMS

A good admission control mechanism improves the site's performance during overload by only admitting a certain amount of customers at a time into the site. The fundamental observation is that it is sometimes better to reject some customers so that other customers may finish their tasks and thereby generate some revenue for the site.

There are two basic types of admission control schemes that may be used in commercial web sites: Request-based and Session-based. In a request-based scheme there is an upper limit of the number of ongoing TCP connections in the site. With a request-based scheme, customers may be rejected in the middle of their sessions. In a session-based control scheme, only the first request in a customer session is sent to the admission control. Once a customer has been admitted to the site, the site guarantees that the customer may complete his/her session.

As can be seen in Figure 3, an admission control mechanism consists of two parts: a *gate* and a *controller.* The controller measures one or more so called *control variables.* Using the control variables, the controller decides the rate at which requests can be admitted to the system. The gate rejects those requests that cannot be admitted. A notification message should be sent to a rejected customer, in order to inform about the current load status of the system. The requests that are admitted proceed to the rest of the system.

6.1. Control Variables

The choice of control variables is an important issue when developing an admission control scheme. First, the control variables must be easy to measure.

Second, the value of a control variable must accurately show the status of the controlled system. Finally, the control variables must in some way relate to the QoS demands that the customers may have on the system.

Traditionally, *server utilization* or *queue lengths* have been the variables most used in admission control schemes (see, for example, Wildling and Karlstedt [18]). In classical so called Stored Program Control (SPC) systems, like telephone exchanges or Service Control Points (SCPs), the main objective of the control scheme was to protect a centralized system from overload. The main customer QoS demand was the blocking probability and, therefore server utilization and queue length were two appropriate variables that were easy to measure.

However, commercial web sites introduce new control problems that have to be solved. First of all, the sites are usually distributed. In a distributed web site it may be difficult to have any detailed assumptions about the back-end server architecture. If a distributed middleware is used in the site, the underlying server architecture is hidden for the application layer. This means that system variables as server utilization and queue length may be difficult to use in an admission control scheme. Second, the one customer QoS demand for a commercial web site is short response times. This means that the server utilization, for example, is not directly related to the main customer QoS demand since a bursty arrival process may cause long delays even if the average server utilization is low (see Kleinrock [10]).

Therefore, a better choice of system variable is the *processing delay* for a request. A customer's response time consists of a network delay and a processing delay. In an overloaded site, the processing delay is probably the major component in the customer response time. Rajamony and Elnozahy [17] develop a performance monitor that measures the processing delay for each request by tagging them when they pass the firewall. This type of monitor may be used in the site in order to obtain accurate measurements of the delay.

6.2. Proposed admission control scheme

In this section we describe a simple admission control scheme that may be used in a distributed commercial web site. The control scheme uses the processing delay as system variable and is therefore independent of the back-end server cluster. It may be implemented as either a request-based or a session-based scheme.

6.2.1. The gate

The gate uses a dynamic window mechanism. There is an upper limit, the so-called window W, for the number of requests (in the request based case) or sessions (in the session based case) that may be processed at the same time in the site. If there are W ongoing requests (sessions) when a request arrives at the gate, it is placed in a waiting queue with *10* places. If the waiting queue is full, the request is rejected.

6.2.2. The controller

The controller updates *W* by measuring the processing delay for the back-end server cluster. If a completed request has had a processing delay higher than *8* seconds, W is decreased with one. If instead *20* requests have had a delay lower than *7* seconds, W is increased with one. The limits for the delay are chosen according to [5]. Other values may of course be used. *W* is always between *1* and *500.*

6.2.3. Several Customer Classes

In the control scheme described above we have assumed that all customers belong to the same class. However, the scheme may easily be generalized to several customer classes with different priorities, for example premium and basic [4]. If it is assumed that the back-end web servers use per-class queuing (see, for example, [2]), the QoS for premium customers is independent of the basic customers. The admission control scheme must use one controller per class. If the processing delay for a customer of any class becomes too long, the controller decreases the window for basic customers. If the window for basic customers is zero, the window for premium customers is decreased. The window for basic customers is increased only when the premium customers have an acceptable processing delay and no premium customers are rejected.

7. SIMULATIONS

We have used simulations to investigate the performance of an overloaded commercial web site. In particular we have investigated customer QoS during overload both when the site is uncontrolled, that is without admission control, and when the site uses either a request-based or a session-based control mechanism as described in section 6.2.

7.1. Site Architecture

The web site consists of one dispatcher, two application servers, one database server, and one authentication server. The application servers work with the same speed and have access to all data. The web site represents a web store, which means that the requests belong to one of the following types: Browse, Search, Select, Add and Pay.

Table 1. Required servers

Request	Servers
Browse	APP, DB
Search	APP, APP, DB, DB, APP
Select	APP, DB
Add	APP, DB
Pay	APP, DB, APP, AS

Table 1 shows the servers required to complete each request type. As can be seen in the table, a request may need to be processed more than once in a particular back-end server. This is due to the dynamic scripts that have to be processed before the requested web page can be returned. For each script some data has to be read or written in the database or the authentication server. We assume that a search query in average consists of two words. Therefore two item lists have to be constructed and merged (see, Meira *et al.* [12]). In the browse request, only one item list has to be constructed.

7.2. Customer Behaviour

New customers arrive at the site according to a Poisson process with mean λ customers per second. The customers send requests according to a probability distribution $p(i)$ where i is the request type.The probability distribution is given by: *p(Browse)*=0.37, *p(Search)*=0.36, *p(Select)*=0.15, *p(Add)*=0.015 and *p(Pay)*=0.005. We assume that a customer leaves the site after a Pay request. With probability *p(leave)*=0.1, the customer leaves the site without buying anything. This probability distribution corresponds to the occasional buyer in [14].

7.3. Simulation Parameters

The following processing times have been used in the simulations: x_f=1 msec, x_{APP}=10 msec, x_{DB}=5msec, and x_{AS}=10 msec. The processing time in the front-end server is chosen so that the load on the server is kept below one for all simulation cases. When an admission control mechanism is used, a customer never becomes impatient ($S_y = \infty$). This makes it easier to understand the behaviour of the control schemes. When the system is uncontrolled, the maximum waiting time for a customer is 8 seconds. The mean think time between each request is 5 seconds.

8. RESULTS AND DISCUSSION

Discrete event simulation was used to investigate the system described above. The results shown here are steady-state averages. All 95%-confidence intervals were within 5% of the average value.

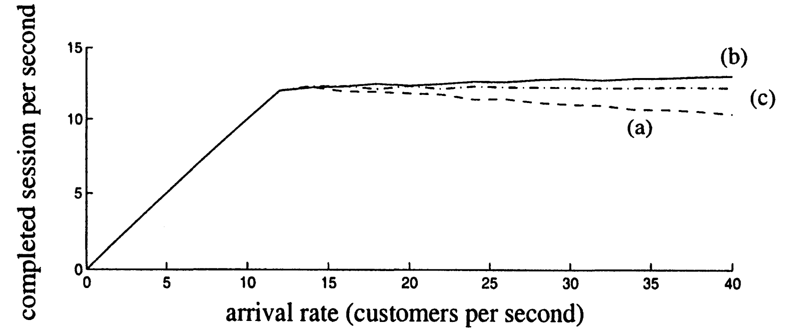

Figure 4. Rate of completed sessions: (a) no admission control (b) request-based scheme (c) session-based scheme.

8.1. Completed Sessions

One of the main performance metrics for a commercial web site is the rate of completed sessions, which is the so-called *goodput*, since only completed sessions may generate revenue. Figure 4 shows the goodput for varying arrival rates. The schemes reject customers during overload (the saturation point indicates the system capacity), which means that other customers may complete their sessions. In the uncontrolled system it is the customers themselves that act as a control mechanism, since a customer abandons the site if the response time is higher than 8 seconds.

As can be seen, the goodputs are surprisingly similar. If one think of traditional load control theories, the throughput in the request based case should decrease when the load increases. In the request-based scheme, customers may be rejected also when they are in the middle or at the end of their session. If such a customer is rejected, capacity will be wasted and the throughput should thereby decrease. However, this is not the case here. We have proven mathematically that this phenomenon is due to the geometrically distributed session lengths.

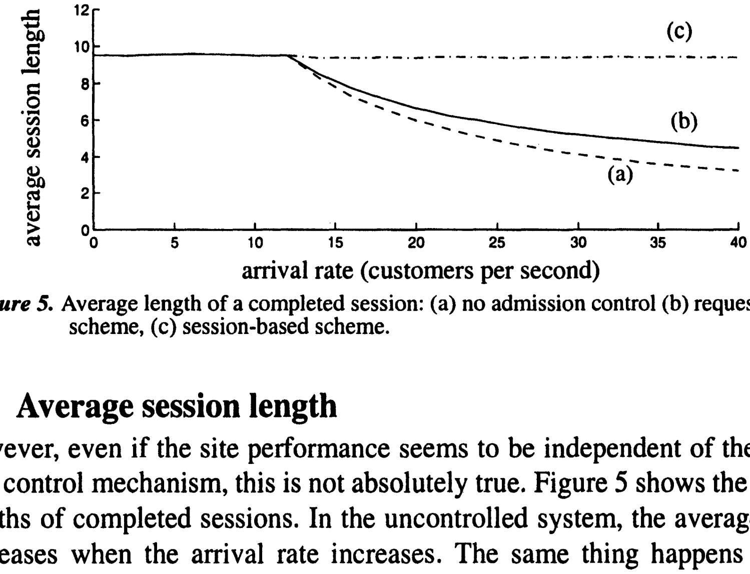

Figure 5. Average length of a completed session: (a) no admission control (b) request-based scheme, (c) session-based scheme.

8.2. Average session length

However, even if the site performance seems to be independent of the admission control mechanism, this is not absolutely true. Figure 5 shows the average lengths of completed sessions. In the uncontrolled system, the average length decreases when the arrival rate increases. The same thing happens when a request-based scheme is used. This means that in these two cases the system favours short sessions. Since longer session may generate a larger income for the site, this kind of behaviour is probably not good for business. On the other hand, with a session-based scheme the average session lengths remain the same even for very high arrival rates. This is because the session-based scheme guarantees that an admitted customer can complete his/her session irrespective of how long it is.

8.3. Angry Customers

Another important performance metric for a commercial site is the percentage of so called angry customers. Angry customers are those that have to leave the site before their sessions are completed. Each angry customer represents both goodwill and revenue loss. Figure 6 shows the percentages of angry customers for varying arrival rates. With a session-based scheme, only new customers may be rejected. As discussed in section 3, these customers are not defined as angry which means that the percentage of angry customers is zero for all arrival rates. The request-based scheme generates a considerable amount of angry customers since customers may be rejected in the middle of their sessions. The uncontrolled system has an even worse behaviour.

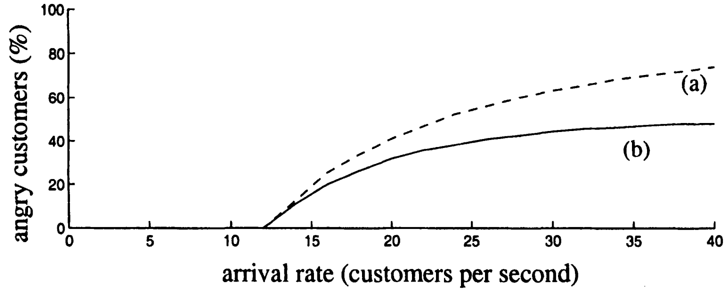

Figure 6. Percentages of angry customers: (a) without admission control (b) request-based scheme.

9. CONCLUSIONS

Many commercial web sites have recently experienced performance problems due to the growth of Internet trading. For commercial web sites it is crucial to maintain high customer QoS also during overload periods, since the site is dependent on customers finishing their tasks in order to get some profits. The best way to guarantee customer QoS during periods of overload is to implement admission control mechanisms in the site. The admission control mechanism acts as a gate that only admits an acceptable number of requests into the site.

In this paper we have proposed and investigated a simple and feasible admission control scheme for distributed commercial web sites. The proposed scheme uses the processing delay as control variable. In a distributed site, the processing delay is the performance metric most related to customer QoS. Further, a control mechanism based on processing delays becomes independent of both the site architecture and any middleware implementation that may be used. The investigations show that the admission control scheme should be session-based. A session-based control scheme guarantees that admitted customers may complete their sessions with good QoS thereby minimizing the number of angry customers. Request-based schemes favour short sessions and generate a high rate of angry customers.

References

1. T.F. Abdelzaher and N. Bhatti, "Web content adaptation to improve server overload behavior", Computer Networks, Vol 31, 1999, pp 1563-1577.
2. T.F. Abdelzaher, K.G. Shin and N. Bhatti, "Performance guarantees for web server end-systems: a control theoretic approach", IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 1, Jan 2002, pp 80-96.
3. M. Arlitt, D. Krishnamurthy and J. Rolia, "Characterizing the scalability of a large web-based shopping system", ACM Transactions on Internet Technology, Vol. 1, No. 1, Aug 2001.
4. N. Bhatti and R. Friedrich, "Web server support for tiered services", IEEE Network, Sept/Okt 1999, pp 64-71.
5. N. Bhatti, A. Bouch and A. Kuchinsky, "Integrating user-perceived quality into web server design", Computer Networks, Vol. 33 (2000), No. 1-6, pp 1-16.
6. P. Bhoj, S. Ramanathan and S. Singhal, "Web2K: bringing QoS to web servers", HP-Labs Technical Report, HPL-2000-61, 2000.
7. R. Carter and L. Cherkasova, "Detecting timed-out client requests for avoiding livelock and improving web server performance", Proc. of 5th IEEE Symposium on Computers and Communications, 2000, pp 2-7.
8. L. Cherkasova and P. Phaal, "Predictive admission control strategy for overloaded commercial web server", Proc. of 8th International Symposium on Modeling Analysis and Simulation of Computer and Telecommunication Systems, 2000, pp 500-507.
9. A. Iyengar, E. MacNair and T. Nguyen, "An analysis of web server performance", Proc. of Globecom'97, 1997, pp 1943-1947.
10. L. Kleinrock, Queueing Systems Vol.1, John Wiley & Sons, 1975.
11. C. Lu, T.F. Abdelzaher, J.A. Stankovic and S.H. Son, "A feedback control approach for guaranteeing relative delays in web servers", Proc. of the 7th IEEE Real-Time Technology and Applications Symposium, 2001, pp 51-62.
12. W. Meira Jr., D. Menascé, V. Almeida and R. Fonesca, "E-representative: a scalable scheme for e-commerce", Proc. of 2nd International Workshop on Advanced Issues of E-commerce and Web-based Information Systems, 2000, pp. 168-175.
13. D. Menascé, V. Almeida, R. Fonseca and M. Mendes, "A methodology for workload characterization of e-commerce sites", Proc. of ACM Conference on Electronic Commerce, 1999, pp 119-128.
14. D. Menascé, V. Almeida, R. Fonseca and M. Mendes, "Business-oriented resource management policies for e-commerce servers", Performance Evaluation, Vol 42(2000), pp 223-239.
15. J. Nielsen, "Why people shop on the web", http://www.useit.com/alertbox/990207.html, 1999.
16. V. Paxson and S: Floyd, "Wide area traffic: the failure of Poisson modeling", IEEE/ACM Transactions on Networking, Vol. 3, No. 3, June 1995, pp 226-244.
17. R. Rajamony and M. Elnozahy, "Measuring client-perceived response-times on the WWW", Proc. of the 3rd USENIX Symposium on Internet Technologies and Systems, 2001.
18. K. Wildling and T. Karlstedt, "Call handling and control of processor load in an SPC system, a simulation study", Proc. of the 9th International Teletraffic Congress, 1979.