STUDIA INFORMATICA 2009
Volume 30 Number 4 (87)

Grażyna SUCHACKA
Opole University of Technology, Chair of Computer Science

PROPOSAL OF USING BUSINESS INFORMATION IN A QoS MECHANISM FOR THE E-COMMERCE WEB SERVER[1]

Summary. Due to very negative and long-term consequences of a low quality of service (QoS) for e-business, a number of QoS mechanisms for Web servers were proposed. As a continuation of this research trend, the paper proposes a new way of using business information in an admission control and scheduling scheme for the e-commerce server aiming at the integration of the server system efficiency with e-business profitability.

Keywords: Web server, Quality of Web Service, QoWS, e-commerce, RFM

PROPOZYCJA WYKORZYSTANIA INFORMACJI BIZNESOWYCH W MECHANIZMIE JAKOŚCI USŁUG DLA SERWERA E-COMMERCE

Streszczenie. Tematyka pracy dotyczy problemu jakości usług ośrodków webowych. Zaproponowano nowatorski sposób wykorzystania informacji biznesowych w metodzie kontroli przyjęć i szeregowania żądań dla serwisu e-commerce. Celem metody jest połączenie aspektu wydajności serwisu webowego oraz rentowności elektronicznego biznesu.

Słowa kluczowe: Web server, Quality of Web Service, QoWS, e-commerce, RFM

1. Problem of a low quality of Web service

Contemporary Web servers are often subject to transient overloads, mainly due to a very “bursty” and unpredictable nature of a Web traffic. During such an overload a response time of admitted requests increases to an unacceptable level resulting in many request timeouts or users’ impatience and site abandonments, usually after some service has been granted. The overloaded server seems to be busy but its work is unproductive and its throughput dramatically decreases. Such a situation is particularly undesirable in case of e-commerce Web sites because it leads to many aborted user sessions and incomplete transactions. The negative consequences for e-business in the long run include a weak company image, users avoiding the site and measurable losses of revenue that could be generated.

In order to guarantee a proper quality of service at the Web server many admission control and scheduling algorithms for the server were proposed. Controlling request arrival rates at the server input turned out to be a good way of protecting the Web server from overload. On the other hand, replacing a standard FIFO scheduling policy by more sophisticated ways of handling HTTP requests proved to be capable of increasing the server throughput and offering a differentiated service levels. However, most of these schemes are not suitable for the e-commerce Web server destined for B2C (Business-to-Consumer) applications. A specific character of the users' interaction with the e-commerce Web site and especially possible financial consequences of that interaction indicate a necessity of looking at the efficiency of the server not only at the level of the computer system but also from the perspective of the business.

2. Proposal of a new QoS mechanism for the e-commerce Web server

This paper presents an idea of a new Quality of Service mechanism for the B2C e-commerce Web server that applies a business criterion to admit and schedule HTTP requests coming to the server. The idea is an application of one of customer segmentation methods known in CRM (Customer Relationship Management) into the control of the Web server. A term of a key customer is introduced and RFM (Recency, Frequency, Monetary) analysis is applied to assess a profitability of key customers of the e-commerce Web site. The more "profitable" a customer is (a term of a customer profitability will be explained in detail below) the better quality of service will be granted to process his/her requests. A new method of HTTP requests scheduling for the e-commerce server has two main objectives:

- maximizing a percentage of successfully completed key customer sessions and
- maximizing revenue generated through the successfully completed transactions.

These goals are reflected in the name of the proposed method, KARO (Key customers And Revenue Oriented scheduling). They are achieved at a limited usage of the server system resources which is realized through preventing the server from admitting the excessive number of requests. At peak times the method guarantees a precedence of a service for

[1] This work was supported by the Polish Ministry of Science and Higher Education under Grant No. N516 032 31/3359 (2006-2009).

requests from the most profitable customers of the site and for requests connected with the most advanced stages of the session at the cost of requests that are less important from the business point of view (i.e. requests from unknown customers who have stayed at the site for a long time and don't show any intend to make a purchase since their shopping carts are empty).

2.1. Idea of customer classification at the e-commerce site

The main idea of including business information into a QoS mechanism for the e-commerce server is introducing a concept of "key customers" of the online store and using information on the past customers' shopping in a process of making a decision on the order of request processing under server overload. Preferential treatment of the most loyal customers has been practiced in a traditional marketing for a long time and it is a basis of many CRM techniques. A practice of many companies confirms a Pareto Principle (also known as the 80/20 rule) which states that a dominant part of the company profit (80%) is generated by a relatively small percentage of the most profitable customers (20%). Furthermore, winning over new customers is even 5 – 6 times more expensive than retaining the loyalty of already won over ones. In the electronic commerce environment a preferential treatment of key customers seems to be especially justified by the following observations: a percentage of e-commerce customers who end up buying something is rather small (2.13% [1] – 5% [8]) and most customers of the online stores are returning customers [10, 12]. Returning customers have different navigation patterns at the e-commerce site from non-returning ones and they are characterized by a greater probability of making a purchase. That's why we differentiate between users who had just bought something in the store sometime and users who hadn't.

We define two customer classes: *KC* class for key customers, defined as users who had made a purchase in the store in the past and *OC* class for ordinary customers defined as users without any prior purchase. We assume that a user may be identified as a key customer only after logging into the site. Unless the user is logged in, he/she will be treated as an ordinary customer. However, if the user decides to log into the site, he/she may be classified as a key customer or remain the ordinary customer depending on the purchase history in the store.

In order to quantify key customers of the online store, a method called RFM (Recency, Frequency, Monetary) analysis is used. It allows to cluster customers based on their past behavioral data and to use this information to predict future customers' behavior. Recency means the time interval from the customer's last purchase until now. Frequency means the total number of times the customer has made a purchase. Monetary means the total amount of money the customer has spent in the store.

RFM analysis is a powerful technique of identifying the most valuable customers of a company. It doesn't require collecting demographic, social or economic information on the company's customers and doesn't involve big costs. Strong arguments for applying this method into electronic commerce environment are its simplicity and a lack of necessity of performing complicated computations in real time. All data required to compute RFM codes are easily accessible and can be captured by a simple observation of the customers' purchasing behavior.

2.2. Model of a user session at the e-commerce site

Users interact with the e-commerce site through a session. A user session is defined as a sequence of temporally and logically correlated requests issued by the user during a single visit to the site. Throughout the session a user performs some typical business functions such as entry at the site, registration, logging in, browsing and searching goods, looking through detailed product information, adding products to the shopping cart, placing an order, determining a way of payment and delivery and finally an order confirmation. Performing each of such business functions by a customer involves transmission of many single HTTP requests by the client internet browser. The web server intercepts the incoming requests, processes them (usually issuing complex queries to a database server), generates replies and sends them back to the client.

We model a behavior of e-commerce customers at the server using a modified state transition graph called Customer Behavior Model Graph (CBMG) [6]. CBMG is a graphical representation of relationships between user session states at the e-commerce Web site. Session states correspond to the business functions. In an original CBMG six states were specified: "*Home*", "*Browse*", "*Search*", "*Details*", "*Add*" and "*Pay*". Based on results of experiments using real Web server logs an e-commerce workload was described as two CBMGs specifying two typical customer profiles: an „occasional buyer" and a „heavy buyer". The first profile encompasses customers who use the e-commerce site to find out about products and their prices and usually end up not buying in the Web store. The second profile describes customers who have a much higher probability of making a purchase. Each customer profile is characterized by different set of transition probabilities between the states, average server-perceived think times for state transitions and arrival rates of session initiation requests. Furthermore, CBMGs provide other useful information regarding the average number of visits to each state of the graph for a single session, the average session length expressed as the average number of states visited per session and the buy to visit ratio.

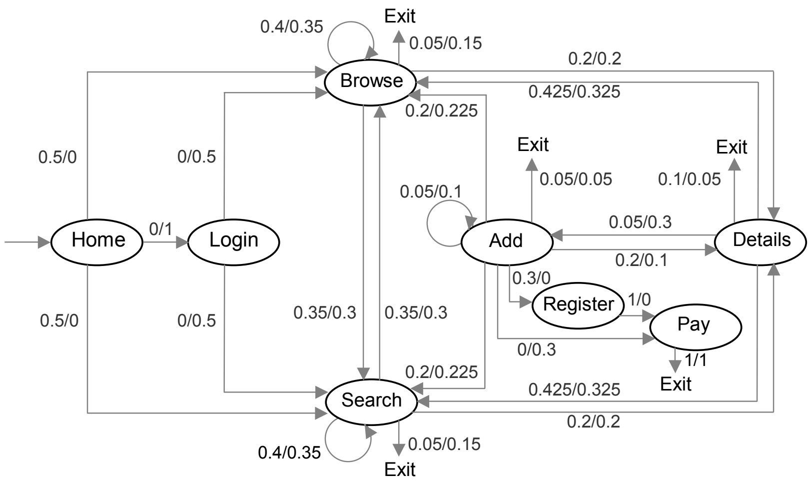

Fig. 1. Model of the ordinary customer/key customer session at the e-commerce site
Rys. 1. Model sesji zwykłego klienta/kluczowego klienta sklepu internetowego

Customer Behavior Model Graphs specified in [6] have been used in a lot of research as a basis of the e-commerce Web site workload model both for simulation experiments [4, 14] and for experiments with a real Web server prototype [13]. We propose to augment the original graph with two additional states corresponding to Web interactions "*Register*" and "*Login*". We assume that every key customer logs in after entering the site and every ordinary customer navigates through the site without being logged in and he/she registers only before finalizing a transaction. The modified CBMG used to model a session of ordinary customer class and a session of key customer class is presented at Fig. 1.

2.3. Key customers' segmentation and valuation scheme

For each customer who had made at least one purchase in the Web store there exists an appropriate record in a customer database. Each customer record contains purchase history data that must be properly coded, i.e. it contains at least three pieces of information [3, 7]: a date of the last purchase in the Web store, a counter for the total number of times the customer has made a purchase (incremented by one every time a purchase is made) and a counter for the total monetary amount spent by the customer in the store (incremented by the amount of purchased items every time a purchase is made).

These three pieces of data for all key customers are used to construct RFM codes. A way of constructing recency, frequency and monetary codes is described in detail below.

We apply the most practical and common method of computing a recency code which is based on a calendar and divides customers on the basis of their recency values according to the following ranges: 0-3 months, 4-6 months, 7-12 months, 13-24 months and above 25 months. To the most recent group one assigns and inserts in each customer record the number

5. The next group receives the number 4, etc. After that operation every customer in database has a recency code of 5, 4, 3, 2 or 1.

A frequency code is constructed using the Behavior Quintile Scoring method. All customer records are sorted by frequencies with the greatest frequencies at the top. Then the records are divided into five groups on the basis of five ranges of the frequency values, with four cutoffs generated every 20% of the highest frequency value. Thus the highest group includes customers who bought in the store more frequently than 80% of the highest frequency value – these customers receive frequency code of 5. On the other hand the lowest group includes customers who bought not more often than 20% of the highest frequency value – they receive a score of 1. In this way there are similar values of frequencies in each of five groups although the numbers of customers in each group may be different. After this step every customer record has a two digit code which varies from 55 (which means the most recent and the most frequent customers) down to 11 (which means the oldest and the least frequent customers).

A monetary code is constructed like a frequency code, after sorting customer records by amounts of money spent in the store with the greatest amounts at the top. After this step every customer has a three digit code in his/her customer record, from 555 down to 111. All in all, there are 125 possible RFM cells.

There are a few possible approaches to calculate a single RFM score for a customer. In different types of business individual components R, F and M may be of different importance for the final single score and therefore they may receive different weights. Typically recency is considered the most important component, the second is frequency and the third is the monetary value [2, 5, 7, 9]. In order to schedule key customers' requests at the server input we apply the following way of calculating a single RFM value for a key customer *k*:

$$RFM_k = w_R \times R_k + w_F \times F_k + w_M \times M_k \quad (1)$$

where w_R, w_F, w_M are weights assigned to the corresponding behavioral variables R, F, M and R_k, F_k, M_k mean values of the appropriate behavioral variables for a key customer *k*. For example, if the weights w_R, w_F, w_M are equal to 3, 2 and 1 respectively then the maximum possible value of a single RFM score will amount to 30 (i.e. 3×5+2×5+1×5) and the minimum possible value will amount to 6 (3×1+2×1+1×1). Customers with the highest values of RFM score are probably the most valuable for the Web store and are characterized by the highest probability of repeated visits and purchases in the future [9].

For the given Web store RFM cell codes and a resulting single RFM values are recalculated periodically (e.g. every month) for all key customer records in the customer database. Moreover, after each next purchase made by a key customer the corresponding RFM codes are updated. For each customer who just made the first purchase in the store a proper record in database is created and appropriate RFM values are computed (a three digit code of a new purchaser will be probably equal to 511 giving a single score of 18).

RFM scores for key customers are read from the customer database every time after the customer logged into the site and was identified as a key customer. RFM scores are used in a scheduling algorithm so that requests from the most profitable customers (with the highest RFM values) take the precedence of service over requests from the ordinary customers whose shopping carts are empty and whose sessions aren't at the state "*Pay*". RFM values for ordinary customers whose sessions are at a stage "*Pay*" are assumed to be zero.

2.4. Dynamic changes of a session priority

In our QoS mechanism for the e-commerce server HTTP requests belonging to different user sessions are offered differentiated levels of service depending on the session priority. The idea of dynamic changes of session priorities is based on the approach described in [6]. We define four priorities: Highest, High, Medium and Low. A session priority (i.e. a customer priority) changes dynamically along with the session progress based on a customer class (*KC* or *OC*), a session state, an amount of money corresponding to products in the shopping cart and a session length (computed as the number of session states visited during the session so far). Session priority is verified and updated at arrivals of requests for new Web pages only. Thus all subsequent requests for objects embedded in the Web page have the same priority.

Fig. 2 illustrates a way of changing the session priority. Customers who logged into the site and were identified as key customers have the Highest priority (P1) all the time of the session duration. Ordinary customers receive the Highest priority only when they are ready to finalize a transaction (i.e. their session state is "*Pay*"). If an ordinary customer logs on in the middle of the session and is classified as a key customer, his/her session will receive the Highest priority. The rationale is that in the long run key customers are most important for the online store even if they have empty shopping carts and are only browsing the Web site. On the other hand, all customers regardless of their class must have a guarantee of a fast service if they are ready to make a purchase.

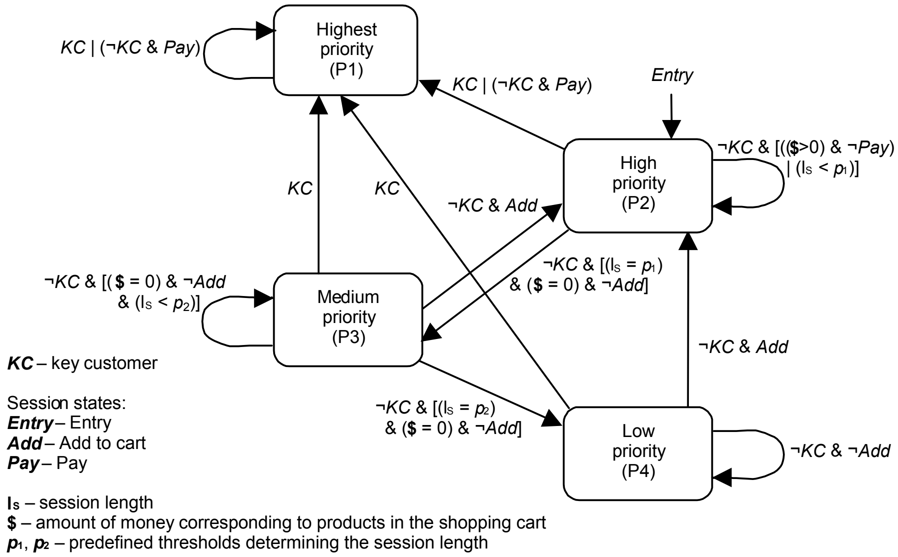

Fig. 2. Dynamic session priority scheme
Rys. 2. Diagram obrazujący dynamiczną zmianę priorytetu sesji użytkownika

All customers entering the Web site receive a High priority (P2). Ordinary customers stay at this level of priority unless their session length reaches the first threshold p_1 or if they have not empty shopping cart (but the session state is not "*Pay*"). If an ordinary customer has some products in a shopping cart and he/she wants to finalize a transaction (the session state is "*Pay*"), the session priority will be changed to the Highest level. However, if the ordinary customer's session length reaches a threshold p_1 and his/her shopping cart is still empty, the priority will be lowered to Medium.

Medium priority (P3) is assigned to the sessions of ordinary customers whose session length is longer than p_1 but not longer than p_2 and their shopping carts remain empty (and they aren't going to add something to the cart i.e. the session state is not "*Add*"). As soon as an ordinary customer adds something to the cart, the session priority will be raised to High. But if his/her shopping cart is still empty and the session length reaches the second threshold p_2, the priority will be lowered to Low.

Low priority (P4) is assigned to the ordinary customers who stay in the Web store for a long time (session length is longer than the threshold p_2) and don't have any items in their shopping carts. In reality such customers have a very low probability of making a purchase so they are considered to be the least valuable for the online store. But if such a customer decides to add any item to the cart, his/her priority is raised to High again.

2.5. Admission control and scheduling of HTTP requests

Control of the e-commerce Web server changes depending on the level of the server workload. Three ranges of workload are defined: Low Load, Medium Load and High Load.

During Low Load of the Web server all HTTP requests coming to the server are admitted and scheduled according to FIFO policy. When the server workload reaches a level defined as Medium Load all requests are admitted and scheduled according to the new mechanism using a dynamic session priority scheme. This scheduling policy is applied also during High Load of the server but additionally an admission control is performed.

All HTTP requests coming to the server receive a priority according to the scheme described in a section 2.4. Each request is identified with regards to the session it belongs to. If a request belongs to a key customer session then it is going to receive the Highest priority. A request belonging to an ordinary customer session is subject to further classification process. In case of the request for an object embedded in a Web page a session priority is read and assigned to the request. In case of the request for a new Web page a session priority is verified. Based on the session state, the amount of money accumulated in a shopping cart and the session length, the session priority is determined and updated if there is such a need. Then the priority is assigned to the request.

A request priority determines a way of the request handling at a server. During High Load of the server some requests with Low or Medium priority may be rejected in order to prevent the Web server from overload and to ensure the integrity of sessions with High and Highest priorities. Depending on the priority the admitted request is added to one of four priority queues (Fig. 3). Each priority queue is destined for requests with different priority. Requests waiting in the Highest priority queue are scheduled according to customer RFM values and the requests with the same RFM value are scheduled according to the value of goods in the shopping cart. In other three priority queues requests are scheduled according to FIFO policy. Among the priority queues a strict or weighted priority scheduling policy is applied.

After processing of each request some information required by the admission control and scheduling policy at the server has to be recorded. Furthermore, after making a purchase by a customer the information on the date and amount of money spent in the store is recorded in a proper customer record and the customer RFM scores are updated.

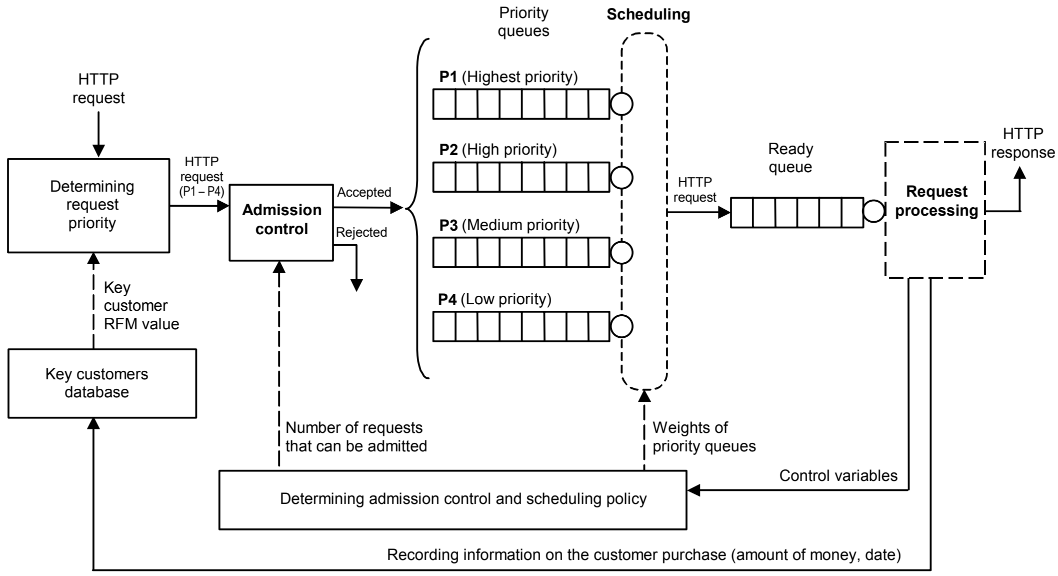

Fig. 3. Architecture of a Web server with a dynamic session priority scheme
Rys. 3. Architektura serwisu WWW dla proponowanej metody obsługi żądań

3. Results of simulation experiments

We developed a simulation tool consisting of a session-based workload generator and an e-commerce Web server system simulator. The tool was implemented in C++ with a toolkit for modeling complex systems CSIM19 [11]. Using our tool we run a series of experiments comparing the efficiency of the e-commerce Web server system under FIFO and KARO scheduling policies. Every single experiment was run for a constant session arrival rate. In consecutive experiments the session arrival rate was gradually increased.

Weights assigned to the components recency, frequency and monetary were all equal to 3. Session length thresholds p_1 and p_2 were equal to 2 and 30, respectively. Among the priority queues a strict priority scheduling policy was applied i.e. all higher priority requests were scheduled before lower priority requests. A percentage of arrived key customer sessions was about 10. We assumed that if a page response time exceeds 8 seconds the user grows impatient and leaves the site without any retries and his/her session is aborted. Simulation results are presented in Fig. 4 – 6.

Fig. 4 presents the e-commerce system throughput in the number of successfully completed user sessions as a function of the session arrival rate. The system under FIFO scheduling policy reaches its maximum capacity at about 65 new sessions per minute and after that point the throughput continues to drop rapidly. A sharp slope of the curves for all user sessions and key customer sessions is due to long page response times much exceeding 8 seconds and thus leading to users' impatience and aborted sessions. At a session arrival rate of 75 none of the sessions is successfully completed (although the system throughput in the number of completed HTTP requests still increases). Fig. 4 shows that KARO ensures much higher and stable system throughput under overload, especially for key customer sessions. However, Fig. 5 reveals that this method isn't able to prevent the abortion of all *KC* sessions. It is caused by the fact that the e-commerce system bottleneck resource is a back-end server and request scheduling at the Web sever input isn't sufficient to guarantee a high quality of service for all high priority requests.

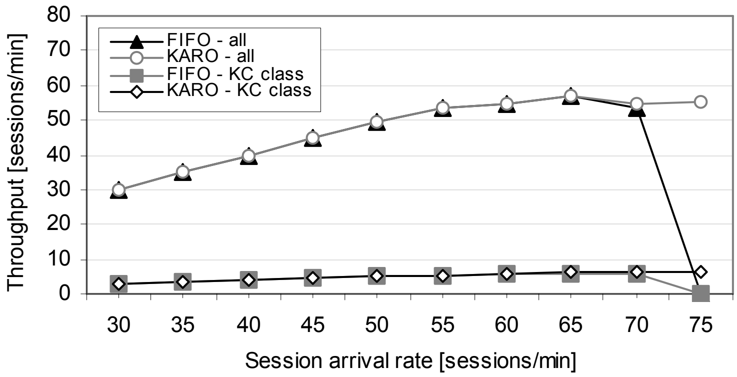

Fig. 4. System throughput as the number of user sessions completed per minute
Rys. 4. Przepustowość systemu jako liczba pomyślnie ukończonych sesji na minutę

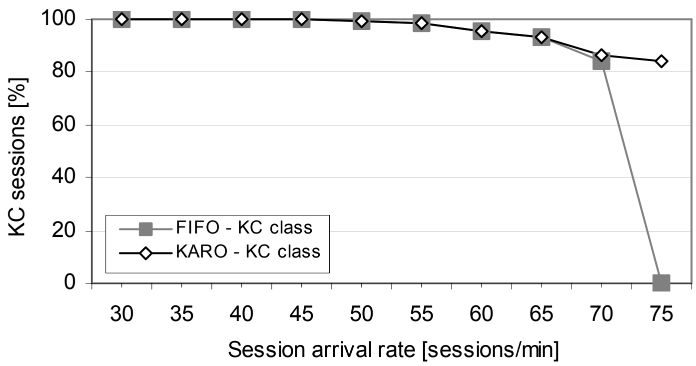

Fig. 5. Percentage of successfully completed key customer sessions
Rys. 5. Odsetek pomyślnie ukończonych sesji kluczowych klientów

Fig. 6 shows the revenue per minute resulting from the successful purchase transactions and also potential revenue losses per minute corresponding to the money accumulated in shopping carts of the sessions that had been aborted due to long page response times. For FIFO scheduling revenue rate grows until the system nears its maximum capacity in completed sessions and then drops rapidly. The potential revenue losses also increase along with the increase in the session arrival rate and paradoxically start to decrease under high load. The reason is the fact that the sessions are aborted early and the customers haven't had a chance of adding any item to the carts. KARO guarantees much higher revenue under the system overload but for medium load levels the potential revenue losses are still significant.

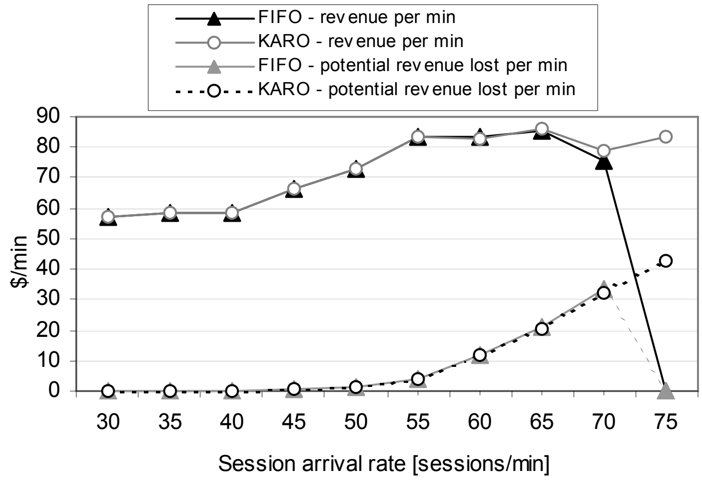

Fig. 6. Revenue and potential revenue losses per minute
Rys. 6. Przychód oraz straty potencjalnego przychodu na minutę

4. Conclusions

Quality of service mechanisms for e-business Web servers should take into consideration not only an aspect of the server system efficiency but also a necessity of maximizing the company revenue. A few e-commerce site characteristics have been included in scheduling algorithms so far, such as session states, transition probabilities between the states, amount of money corresponding to items in a shopping cart, user profiles. However, according to the best author's knowledge none of the scheduling algorithms for the e-commerce server proposed a way of combining a short-term aspect of the revenue maximization with a long-term perspective of retaining the customer loyalty and business profitability.

This paper describes an idea of a new admission control and scheduling algorithm called KARO and presents results of simulation experiments comparing the efficiency of the e-commerce Web server system under FIFO and KARO scheduling policies. Simulation results clearly demonstrate that our method can ensure much higher quality of service under the system overload in terms of traditional and business performance measures. However, scheduling of requests at the Web sever input isn't sufficient to guarantee a high quality of service for all high priority requests since the e-commerce system bottleneck lies in a database server. Such results motivated us to consider modifications of our KARO method by scheduling of not only HTTP requests at the Web server input but also scheduling of dynamic requests at the back-end server input. Our future works concern implementing such modifications in the simulator and running simulation experiments for a variety of workload scenarios for business-to-consumer e-commerce Web site.

BIBLOGRAPHY

1. Buystream, E-Metric Research Group: Measure Twice, Cut Once – Metrics For Online Retailers, http://www.techexchange.com/thelibrary/online_retail_metrics.html.
2. Chen M.-C, Chiu A.-L., Chang H.-H.: Mining Changes in Customer Behavior in Retail Marketing. Expert Systems With Applications, Vol. 28, Issue 4, May 2005, p. 773÷781.
3. Hughes A., M.: Making Your Database Pay Off – Using Recency, Frequency and Monetary Analysis. Database Marketing Insititute, January 2001.
4. Kihl M., Widell N.: Admission Control Schemes Guaranteeing Customer QoS in Commercial Web Sites. IFIP Conference NetCon'02, Vol. 235, 2002, p. 305÷316.
5. Liu D., R., Shih Y.-Y.: Integrating AHP and Data Mining for Product Recommendation based on Customer Lifetime Value. Information and Management, Vol. 42, Issue 3, March 2005, p. 387÷400.
6. Menascé D., A., Almeida V., A., F., Fonseca R., Mendes M., A.: Business-Oriented Resource Management Policies for E-Commerce Servers. Performance Evaluation, Vol. 42, Issue 2-3, September 2000, p. 223÷239.
7. Miglautsch J., R.: Thoughts on RFM Scoring. ISSM Electronic Journal, Issue 27, 2001.
8. Nielsen J.: Why people shop on the Web, February 1999, updated: April 2002, http://www.useit.com/alertbox/990207.html; access date: March 12, 2008.
9. Novo J.: Turning Customer Data into Profits with a Spreadsheet. The Guide to Maximizing Customer Marketing ROI, Booklocker.com, 3rd edition, 2004.
10. Pecaut D., K., Silverstein M., J., Stanger P.: Winning the Online Consumer: Insights into Online Consumer Behavior. Boston Consulting Group, March 2000, http://www.bcg.com.
11. Schwetman H.: CSIM19: A powerful tool for building system models. In Proc. of the 2001 Winter Simulation Conference, 2001.
12. Silverstein M., Stanger P., Abdelmessih N.: Winning the Online Consumer 2.0. Converting Traffic into Profitable Relationship. A report by BCG, February 2001.
13. Totok A., Karamcheti V.: Improving Performance of Internet Services Through Reward-Driven Request Prioritization. IWQoS'06, June 2006, p. 60÷71.
14. Zhou X., Wei J., Xu C.-Z.: Resource Allocation for Session-Based Two-Dimensional Service Differentiation on e-Commerce Servers. IEEE TPDS, 17(8), 2006, p. 838÷850.

Recenzent: Dr inż. Piotr Gaj

Wpłynęło do Redakcji 4 października 2009 r.

Omówienie

Stały rozwój usługi WWW, a także charakterystyczne cechy ruchu webowego przyczyniają się do przeciążeń serwerów webowych, a w konsekwencji do problemu zapewnienia jakości realizowanych przez nie usług na odpowiednim, przewidywalnym poziomie. Skutecznym sposobem polepszenia jakości usługi WWW jest wzbogacenie ośrodków webowych o mechanizmy klasyfikacji, kontroli przyjęć i szeregowania żądań, jako alternatywy stosowanej obecnie w Internecie usługi best-effort i szeregowania FIFO (ang. *First In First Out*).

W pracy przedstawiono koncepcję nowatorskiej metody obsługi żądań HTTP w serwisie webowym dla aplikacji handlu elektronicznego typu B2C (ang. *Business-to-Consumer*), która została nazwana KARO (ang. *Key customers And Revenue Oriented scheduling*). Główna idea metody polega na zastosowaniu analizy RFM (ang. *Recency, Frequency, Monetary*) do segmentacji klientów sklepu internetowego, a także na preferowaniu w warunkach dużego obciążenia serwisu żądań od kluczowych klientów oraz żądań związanych z finalizacją transakcji zakupu. Odbywa się to kosztem żądań mniej istotnych z biznesowego punktu widzenia, czyli żądań od nieznanych użytkowników sklepu internetowego (w tym agentów, robotów itp.), których sesja z witryną trwa już od dłuższego czasu, ale koszyk zakupów pozostaje pusty.

Przedyskutowano motywację oraz proponowaną koncepcję zastosowania metody RFM w środowisku handlu elektronicznego. Opisano przyjęty model sesji użytkownika na witrynie e-commerce. Przedstawiona została metoda segmentacji i oceny klientów sklepu internetowego na podstawie historii zakupów, a następnie metoda dynamicznej zmiany priorytetu sesji oraz koncepcja algorytmu kontroli przyjęć i szeregowania żądań w serwisie webowym na podstawie priorytetów sesji. Przedstawiono wyniki wstępnych eksperymentów symulacyjnych, których celem było porównanie wydajności serwisu webowego działającego na podstawie proponowanej metody KARO i szeregowania FIFO.

Uzyskane wyniki pokazały, że proponowana metoda zapewnia wyższą jakość usług przeciążonego serwisu webowego zarówno pod względem tradycyjnych miar wydajności (np. przepustowości systemu), jak również miar biznesowych, związanych z osiąganymi przychodami. Okazało się jednak, że kontrola przyjęć i szeregowanie żądań HTTP na wejściu serwisu webowego nie są w stanie zagwarantować wysokiej jakości usług dla wszystkich żądań priorytetowych, ponieważ „wąskie gardło" systemu stanowi jego zaplecze, odpowiedzialne za generowanie zawartości dynamicznej. Uzyskane wyniki stanowią motywację do dalszej rozbudowy metody KARO o szeregowanie żądań dynamicznych na wejściu serwisu zaplecza. Plan dalszych prac związany jest przede wszystkim z implementacją odpowiednich modyfikacji w środowisku symulacyjnym oraz przeprowadzeniem badań dla różnych scenariuszy obciążenia serwisu webowego.

Address

Grażyna SUCHACKA: Opole University of Technology, Chair of Computer Science,
ul. K. Sosnkowskiego 31, 45-272 Opole, g.suchacka@po.opole.pl