Generating Bursty Web Traffic for a B2C Web Server

Grażyna Suchacka

Chair of Computer Science, Opole University of Technology,
Sosnkowskiego 31, 45-272 Opole, Poland
g.suchacka@po.opole.pl

Abstract. The paper deals with the problem of emulating highly bursty Web traffic that can be observed at inputs of Web servers hosting online Web stores. This problem is related to the broad issue of Web server performance prediction and evaluation through simulation experiments. Based on up-to-date results on real Web server workload analyses a workload model has been proposed. It combines a model of a user session at a Business-to-Consumer (B2C) Web site with HTTP-level workload models for business and non-business Web servers. The proposed model has been implemented in a workload generator. Based on statistics registered during a simulation experiment, a burstiness factor has been computed for the generated workload, which has proven to be highly variable and bursty.

Keywords: Web traffic, burstiness, Web server, e-commerce, B2C, Business-to-Consumer, simulation.

1 Introduction

Web performance prediction and evaluation is currently a hot research issue. Due to the problem of the quality of Web service (QoWS), perceived by Internet users mainly through long response times, there has been a lot of research aiming at improving quality of service both in Web server nodes and in the network. This paper addresses some QoWS issues on the Web server side – they are related to modeling and generating Web traffic typical of Web servers hosting online stores, i.e. Business-to-Consumer (B2C) Web sites.

In reality, Web traffic is characterized by some unique properties having a significant negative impact on Web server performance. In particular, designing a workload model for a B2C Web server requires including user navigational patterns at B2C Web sites [1,2,3]. One needs also to take into consideration unique characteristics and invariants typical of real Web traffic, especially its high variability – so-called "burstiness". Real Web traffic has been proven to be bursty across several time scales, i.e. self-similar. Burstiness means that peak HTTP request rates during bursts exceed the average request rate by factors of five to ten and thus can easily surpass the server capacity [4,5,6,7].

Section 2 discusses two main approaches applied to generate Web traffic for a Web server. Section 3 presents a workload model proposed for a B2C Web

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2011, CCIS 160, pp. 183–190, 2011.

server, i.e. for the server hosting a retail store Web site. Burstiness of the Web traffic generated in a simulation experiment according to the proposed workload model is evaluated in Sect. 4. The results are summarized in Sect. 5.

2 Approaches to the Web Server Workload Generation

There are two main approaches to generate a stream of HTTP requests for an evaluated Web server system. The first one is a *trace-based* approach which consists in reproducing workload data recorded in a real Web server log file. Such workload is characteristic of only one actual Web site and may not be representative in general. It is also a kind of a "black box" and thus recognition of reasons for the system behavior and adjustment of the workload to different scenarios may be difficult. Moreover, Web server traces for e-commerce traffic are not publicized because of a sensitive financial aspect of companies' revenues.

Due to the aforementioned reasons we decided to apply a *distribution-driven* approach. It consists in specifying key Web workload characteristics by probability distribution functions and generating a workload according to parameters of the workload model. Distributions are the result of a detailed workload characterization for a few representative Web sites. The resulting sequence of requests has a synthetic nature but one can easily modify workload conditions by changing individual distributions or their parameters.

Research on Web traffic characteristics has resulted in developing a number of benchmarking tools, i.e. computer programs used to evaluate performance and scalability of highly accessed Web servers. Web benchmarks can generate a representative Web workload (usually allowing to customize the workload model) and collect some statistics on simulation results. However, the analysis of popular, freely available benchmarking tools, such as httperf, SPECweb99, SURGE, S-Clients, WebBench and WebStone, has indicated their low suitability for e-commerce Web servers, mainly due to very simplified workload models and their incapability of providing session- and business-oriented performance metrics [8,9]. Therefore, we decided to work out a workload model typical of B2C Web sites based on the up-to-date literature and to develop a new workload generator.

3 Workload Model of a B2C Web Server

Many studies have characterized Web server workload at the HTTP level, i.e. they described key workload characteristics (such as the number of objects per Web page, object sizes and requests interarrival times) by the most adequate probability distribution functions. Thanks to applying heavy-tailed distributions, such as Pareto or lognormal ones, the resultant Web traffic usually reflects an extremely variable and self-similar nature of a real Web traffic.

On the other hand, few studies for e-commerce Web servers characterize their workload at the high level. They identify and model different types of customers

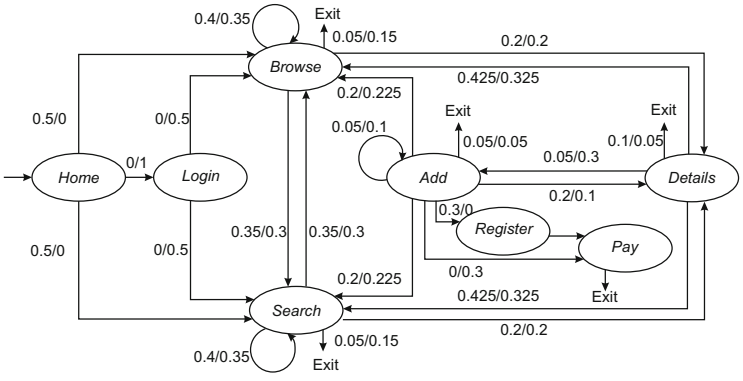

Fig. 1. Modified CBMG used to model *occasional buyer* session/*heavy buyer* session

and customer navigational patterns at the e-commerce site, i.e. a structure of a user session at the site [1,2,3]. Two user session models are well-established in the literature: a state transition graph called Customer Behavior Model Graph (CBMG) [1] and a Web interaction diagram specified in TPC-W benchmark [10], recommended by Transaction Processing Performance Council [11]. Both models specify e-commerce workloads which mimic activities of a retail bookstore Web site. They are based on a workload analysis for representative B2C Web sites and have been widely used in up-to-date research in the QoWS area.

In order to emulate highly variable Web traffic typical of B2C Web sites, we have proposed combining some results on HTTP-level Web workload characterization with a user session model based on the modified CBMG [12].

We decided to use the CBMG because it models two different customer profiles (a *heavy buyer* profile and an *occasional buyer* profile), as well as is more legible and easy to extend. Our modified graph is presented in Fig. 1. Beside six session states distinguished in the original graph (*Home, Browse, Search, Details, Add* and *Pay*), two additional states have been introduced: *Login* and *Register*. Thus, there are eight possible sessions states in our session model:

- *Home* – Entry to the home page of a retail store Web site;
- *Login* – User's logging into the site;
- *Browse* – Browsing the site contents, e.g. browsing items in various categories, bestsellers or new products;
- *Search* – Searching for products according to specific keywords;
- *Details* – Viewing a page containing detailed information on a selected product;
- *Add* – Adding a selected product to a shopping cart;
- *Register* – User's registration at the site;
- *Pay* – Finalizing a purchase transaction, including operations involved in purchase confirmation and making a payment online.

We assume that every occasional buyer navigates through the site without being logged on. Only when he/she decides to finalize a purchase transaction does he/she have to register. On the other hand, every heavy buyer in our session model logs on straight away after entering the site.

Each node of the graph corresponds to one session state. Each arrow between two states k and l means a probability $p^c_{k,l}$ of transition from state k to state l for customer profile $c \in \{heavy\ buyer,\ occasional\ buyer\}$. Arrows labeled with *Exit* mean a user's decision to leave the site without any specific reason. For given customer class c, each transition from state k to state l is characterized by a mean value of server-perceived user think time $u^c_{k,l}$. This is the time interval the user needs to analyze a downloaded Web page and to issue the next page request. User think time is modeled according to an exponential distribution with a maximum of 10 times the mean [10]. Its mean value is equal to 15 seconds for all transitions with the following exceptions [1]: $u^c_{Search,Details} = 30\,\text{s}$, $u^c_{Details,Add} = 45\,\text{s}$, $u^c_{Add,Pay} = u^c_{Add,Register} = 25\,\text{s}$, $u^c_{Register,Pay} = 60\,\text{s}$, for $c \in \{heavy\ buyer,\ occasional\ buyer\}$.

Each single Web interaction corresponds to a single Web page request, which can be assigned to one of the eight sessions states. All Web pages are modeled as dynamic pages with static and dynamic objects. Execution of each Web page request requires processing many HTTP requests, namely the first hit for an HTML page and following hits for all objects embedded in it. Since, to the best of our knowledge, there is no B2C workload model at the HTTP level, we decided to combine results of several workload studies for business [13,14] and non-business [15,16,17] Web servers in our model. We believe this combination is well justified given a common base of these studies, which is a Web server.

E-commerce traffic analyses have not shown a significant dependence of page sizes on a method type (e.g. GET or POST) or on a session state [4], so we model a composition of a Web page apart from the session state. The number of static objects per page (including the base HTML file) is modeled by a Pareto distribution with the scale parameter equal to 1.33 and the shape parameter equal to 2. The number of dynamic objects per page is obtained by a geometric distribution with the success probability 0.8 and then is incremented by 1 in order to ensure at least one dynamic object for a page. Sizes of HTML files are obtained from a hybrid function, where a body follows a lognormal distribution with the mean 7.63 and the standard deviation 1.001, while a tail follows a Pareto distribution with the scale parameter equal to 1 and the shape parameter equal to 10 240.

Sizes of embedded static objects are obtained from lognormal distribution with the mean 8.215 and the standard deviation 1.46. Sizes of embedded dynamic objects are obtained from Weibull distribution with the scale parameter equal to 0.0059 and the shape parameter equal to 0.9. Interarrival time of hit requests at the Web server is modeled by a Weibull distribution with the scale parameter equal to 7.64 and the shape parameter equal to 1.705.

4 Evaluating Burstiness of the Generated Web Traffic

Web traffic burstiness has been evaluated after carrying out simulation experiments and using an approach described in [6] for request arrival rates registered during the experiment in subintervals of duration 100 milliseconds and 1 second.

The proposed workload model has been implemented in a workload generator, written in C++. The workload generator is an integral part of a simulation tool, which allows one evaluating a B2C Web server system performance under different scheduling policies for different server load levels. The workload generator is responsible for generating and transmitting to the Web server system a stream of HTTP requests emulating the session-based server workload. Based on some input parameters it generates user sessions at a given session arrival rate λ_s, i.e. it initiates the given number of user sessions per minute. A pre-specified parameter Δ_{KC} denotes a percentage of generated heavy buyer sessions in the observation window (so a percentage of occasional buyer sessions is $100 - \Delta_{\mathrm{KC}}$).

The experiments discussed here have been carried out for a session arrival rate of 100 sessions per minute ($\lambda_s = 100$), where 10% of all user sessions were generated according to the heavy buyer profile ($\Delta_{\mathrm{KC}} = 10$). The generated workload was monitored in a 3-hour observation window after a 10-hour preliminary phase of the experiment (this time means the internal simulation time which differs from the "real world" time). HTTP request arrival rates at the Web server input have been registered for 100 milliseconds and 1 second intervals.

A visual inspection of the number of HTTP requests arriving at the Web server system confirms the high variability of the generated Web traffic. Figure 2 presents Web traffic bursts in slots of 100 milliseconds for a 3 minute period and Fig. 3 presents bursts in slots of 1 second for a 30 minutes period. High variations in request arrival rates in the figures indicate a significant burstiness of the generated traffic on fine time scales.

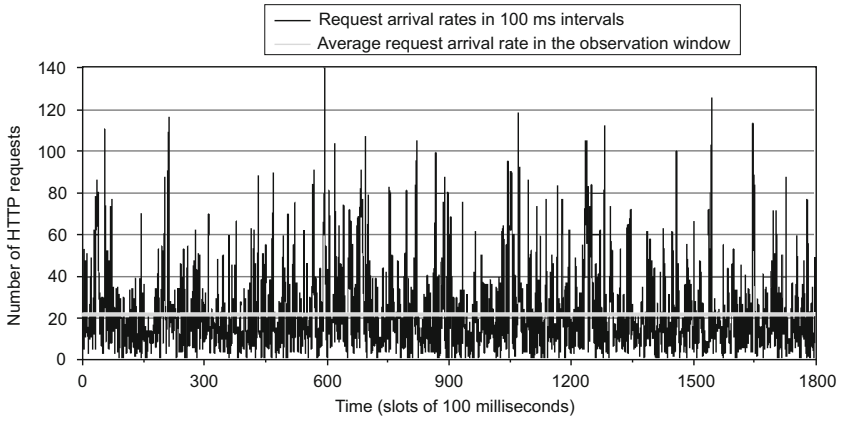

Fig. 2. Web traffic bursts in slots of 100 milliseconds (session arrival rate is equal to 100 sessions per minute)

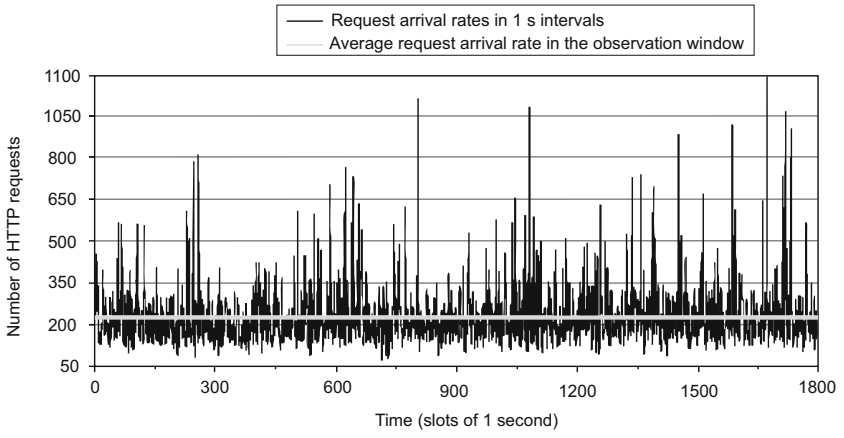

Fig. 3. Web traffic bursts in slots of 1 second (session arrival rate is equal to 100 sessions per minute)

Let L be the total number of HTTP requests generated during the simulation experiment in the interval of $T = 3$ hours $= 10\,800$ seconds. Let also λ be the average arrival rate of requests registered in the experiment, given by:

$$\lambda = \frac{L}{T} \ . \tag{1}$$

We consider the time interval T divided into n equal subintervals of duration t, called epochs. Let l_k be the number of HTTP requests that arrive in epoch k and λ_k be arrival rate of requests during epoch k, given by:

$$\lambda_k = \frac{n \times l_k}{T} \ . \tag{2}$$

Let also l^+ be the total number of HTTP requests that arrive in epochs in which the epoch arrival rate λ_k exceeds the average arrival rate λ registered in the interval T in the simulation experiment.

The burstiness parameter b is defined as the fraction of time during which the epoch arrival rate exceeds the average arrival rate λ:

$$b = \frac{Number\ of\ epochs\ for\ which\ \lambda_k > \lambda}{n} \ . \tag{3}$$

If generated Web traffic is not bursty, it is uniformly distributed over all epochs and so:

$$l_k = \frac{L}{n} \text{ and } \lambda_k = \frac{\frac{L}{n}}{\frac{T}{n}} = \frac{L}{T} = \lambda \ . \tag{4}$$

Such situation means that that there are no epochs in which $\lambda_k > \lambda$ and thus $b = 0$.

For the Web traffic generated according to our workload model, the burstiness factors for epochs of 100 milliseconds and 1 second were equal to 0.34 and 0.39, respectively, which indicates a significant degree of variability and confirms results illustrated in Fig. 2 and 3.

Experiments performed for other levels of load intensity have shown that a degree of burstiness is slightly lower for higher session arrival rates, i.e. for higher server workloads. However, in all cases significant burstiness of traffic at the Web server input was observed.

5 Concluding Remarks

In the paper, a proposed workload model typical of B2C Web servers has been discussed and evaluated with respect to the variability in request arrival rates at the Web server input. The model combines a high-level model of a user session at a B2C Web site with HTTP-level workload characteristics identified for business and non-business Web servers. The model has been implemented in a workload generator integrated with a Web server system simulator. Simulation experiments have been carried out, in which request arrival rates at the server were registered for subintervals of duration 100 milliseconds and 1 second. A burstiness factor computed for the registered data indicates that the Web traffic generated according to the proposed workload model is highly variable and thus it can be applied to generate input traffic in experiments evaluating performance of e-commerce Web server systems.

References

1. Menascé, D.A., Almeida, V.A.F., Fonseca, R., Mendes, M.A.: Business-Oriented Resource Management Policies for E-Commerce Servers. Performance Evaluation 42(2-3), 223–239 (2000)
2. Song, Q., Shepperd, M.: Mining Web Browsing Patterns for E-commerce. Computers in Industry 57(7), 622–630 (2006)
3. Wang, Q., Makaroff, D.J., Edwards, H.K.: Characterizing Customer Groups for an E-commerce Website. In: ACM Conference on Electronic Commerce, pp. 218–227. ACM Press, New York (2004)
4. Kant, K., Venkatachalam, M.: Transactional Characterization of Front-End E-Commerce Traffic. In: IEEE GLOBECOM 2002, vol. 3, pp. 2523–2527 (2002)
5. Makineni, S., Iyer, R.: Performance Characterization of TCP/IP Packet Processing in Commercial Server Workloads. In: IEEE WWC-6, pp. 33–41 (2003)
6. Menascé, D.A., Almeida, V.: Capacity Planning for Web Services: Metrics, Models and Methods. Prentice Hall, Upper Saddle River (2002)
7. Vallamsetty, U., Kant, K., Mohapatra, P.: Characterization of E-Commerce Traffic. Electronic Commerce Research 3, 167–192 (2003)
8. Andreolini, M., Cardellini, V., Colajanni, M.: Benchmarking Models and Tools for Distributed Web-Server Systems. In: Calzarossa, M.C., Tucci, S. (eds.) Performance 2002. LNCS, vol. 2459, pp. 208–235. Springer, Heidelberg (2002)
9. Suchacka, G.: Generowanie obciążenia o specyfice e-commerce dla serwera webowego. Nowe Technologie Sieci komputerowych 2, 183–193 (2006)

10. García, D.F., García, J.: TPC-W E-Commerce Benchmark Evaluation. IEEE Computer 36(2), 42–48 (2003)
11. Transaction Processing Performance Council, `www.tpc.org`
12. Borzemski, L., Suchacka, G.: Web Traffic Modeling for E-commerce Web Server System. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) 16th Conference on Computer Networks, CN 2009, Wisła, Poland. CCIS, vol. 39, pp. 151–159. Springer, Heidelberg (2009)
13. Shi, W., Collins, E., Karamcheti, V.: Modeling Object Characteristics of Dynamic Web Content. Journal of Parallel and Distributed Computing 63(10), 963–980 (2003)
14. Xia, C.H., Liu, Z., Squillante, M.S., Zhang, L., Malouch, N.: Web Traffic Modeling at Finer Time Scales and Performance Implications. Performance Evaluation 61(2-3), 181–201 (2005)
15. Barford, P., Bestavros, A., Bradley, A., Crovella, M.: Changes in Web Client Access Patterns: Characteristics and Caching Implications. WWW 2(1-2), 15–28 (1999)
16. Cardellini, V., Casalicchio, E., Colajanni, M., Mambelli, M.: Web Switch Support for Differentiated Services. ACM Performance Evaluation Review 29(2), 14–19 (2001)
17. Casalicchio, E., Colajanni, M.: A Client-aware Dispatching Algorithm for Web Clusters Providing Multiple Services. In: 10th International WWW Conference, pp. 535–544 (2001)