

# Providing TPC-W with web user dynamic behavior

Raúl Peña-Ortiz, José A. Gil, Julio Sahuquillo, Ana Pont  
Departament d'Informàtica de Sistemes i Computadors (DISCA)  
Universitat Politècnica de València  
Valencia, Spain  
rpenya@upvnet.upv.es, {jagil,jsahuqui,apont}@disca.upv.es

## Abstract

The evolution of the World Wide Web from hypermedia information repositories to web applications such as social networking, wikis or blogs has introduced a new paradigm where users are no longer passive web consumers. Instead, users have become active contributors to web applications, so introducing a high level of dynamism in their behavior. Moreover, this trend is even expected to rise in the incoming Web.

As a consequence, there is a need to develop new software tools that consider user dynamism in an appropriate and accurate way when generating dynamic workload for evaluating the performance of the current and incoming web.

This paper presents a new testbed with the ability of defining and generating web dynamic workload for e-commerce. For this purpose, we integrated a dynamic workload generator (GUERNICA) with a widely used benchmark for e-commerce (TPC-W).

**Keywords:** Web performance evaluation, web workload generator, dynamic web workload, modeling user dynamic behavior, e-commerce

## 1 Introduction

As a consequence of the recent and incessant changes in web technology, the range of services offered through World Wide Web (Web) have suffered a continuous evolution during the last years. This evolution also has implied significant changes in web users' behavior [1].

The first generation of web-based services (called static services) were a low cost method to share a large amount of information with little or no privacy. Most of this information was offered using formatted text with only a small number of images (about 10%). As a consequence, the vast majority of its users were simply acting as consumers of content; in other words, they reviewed contents by navigating the web following the hyperlinks of the pages they visited [2]. Subsequently, dynamic contents became more and more frequent and web services evolved through their second generation. This generation could be distinguished by important changes both in its infrastructure and architecture, which allowed generation, querying and storage of dynamic content. This dynamism was extended to users' behavior [3], who changed their navigations guidelines (more and more dynamic and personalized), and consequently, their related traffic [4]. Nowadays, we are in a new web paradigm where users are no longer passive consumers, but they have become participative contributors to the dynamic web content [2].

As a system that is continuously changing, both in the offered applications and infrastructure, performance evaluation studies are a major concern to provide sound proposals when designing new web-related systems [5], such as web services, web servers, proxies or content distribution policies. As in any performance evaluation process, accurate and representative workload models should be used in order to guarantee the validity of the results. In the case of the WWW, the implicit users' dynamism makes difficult to design accurate workload models representing users' navigation.

In a previous work [6], we introduced an approach to characterize dynamic web workload, namely, Dweb model. This model is based on users' dynamism and implements the capability of changing the user behavior with time. For example, users can dynamically adopt different roles (e.g., browsing or shopping) in e-commerce; that is, they are allowed to navigate the e-commerce website with different behaviors in the same navigation session. Moreover, a dynamic workload generator named GUERNICA was implemented to show a practical application of Dweb model.

This paper proposes a new testbed with the ability of generating web dynamic workload for e-commerce. This new testbed results from the integration of GUERNICA and the commonly used TPC-W benchmark

for e-commerce [7], in order to consider user dynamic behavior when generating workload in an accurate way by using Dweb model.

The remainder of this paper is organized as follows. Section 2 discusses the reasons that motivated us to propose a new benchmark. Sections 3 and 4 present and validate our proposal, respectively. Finally, we draw some concluding remarks and future work in Section 5.

## 2 Background and motivation

The need to characterize workload in order to model and reproduce the user behavior [8] grows with the increasing importance of web applications. This need presents a special importance in e-commerce environments, where characterizing the user's workload not only has the objective of evaluating the performance of the web system, but also of modeling the behavior of those users who can become new potential clients. E-commerce applications have two main objectives: to acquire new clients and to maintain them as active users. This kind of applications presents the following characteristics: i) importance of critical information, ii) high percentage of dynamic and personalized content, iii) need for service and product quality offered to their potential clients, and iv) use of latest generation technologies. Consequently, using inaccurate models in e-commerce performance evaluation would lead to incorrect conclusions that could yield to inappropriate actions on business and system performance.

Floyd et al. [9] describe the main drawbacks when evaluating system performance using an analytical model. This is mainly due to the dynamic nature of workload and the high number of parameters that directly affect its characteristics (e.g., different protocols, types of traffic, client navigation patterns, etc). In general, three main challenges must be addressed when modeling dynamic workload. First, the user's behavior must be modeled [8]. Then, the different roles that users play in the Web must be characterized [10]. Finally, continuous changes in these behaviors must be represented [11].

There have been few but interesting efforts to define user's behavior models in order to obtain more representative workloads for specific web applications. Menascé et al. [12] introduced the Customer Behavior Model Graph (CBMG) that describes patterns of user behavior in workloads of e-commerce sites. CBMG is a workload model included in TPC-W, which is the first benchmark for e-commerce sites used in web performance studies [13]. Duarte et al. [14] applied this model for workload definition of blogspace; and Shams et al. [15] extended CBMG to capture an application's inter-request and data dependencies. Benevenuto et al. [16] introduced the Clickstream Model to characterize user's behavior in online social networks. However, these models only characterize web workload for specific paradigms or applications, do not model user's dynamic behavior in an appropriate and accurate way (first challenge), and do not consider dynamic roles that users play (second and third challenges). These shortcomings motivated us to propose a general purpose workload model called Dweb (Dynamic web workload model) [6], which permits to define dynamic web workload in an accurate way, taking into account the mentioned challenges, by introducing user's dynamic behavior models in workload characterization.

Web performance evaluation studies are supported by specific software with the aim of validating the quality of service provided by a web system under specific workload conditions. Among web performance evaluation software we can highlight benchmarks and workload generators. A benchmark is defined to reproduce workload conditions of a typical working environment with the aim of evaluating whether the system meets established quality standards. On the other hand, a web workload generator pursues a degradation in the quality of service by producing enough HTTP requests. Among the evaluated benchmarks in a previous work [17], we found that TPC-W is the best benchmark for an e-commerce system. In addition, we also concluded that GUERNICA is the only generator that reproduces in an accurate way web dynamic workload, by using Dweb model. TPC-W reproduces multiple on-line browser sessions over a bookstore. It generates dynamic workload but not in an accurate way because it is based on CBMG that only considers a partial representation of the users' dynamic behavior.

To deal with this challenge, we decide to develop a new testbed for e-commerce by extending TPC-W using GUERNICA in order to exploit Dweb model on dynamic workload characterization.

## 3 Integration between TPC-W and GUERNICA

We developed a new testbed to accomplish three main goals. First, it must define and reproduce dynamic web workload in an accurate and appropriate way. Second, it must be able to provide client and server metrics with the aim of being used for web performance evaluation studies. Finally, it should be representative of web transactional systems that have been established in recent years.

Among the evaluated benchmarks in [17], TPC-W is the best candidate to provide an appropriate testbed for our purposes, because it satisfies most of previous goals. But it does not consider users' dynamic behavior

in an accurate way on workload characterization. To achieve the three previous goals, we integrate TPC-W and GUERNICA in order to use Dweb model on the workload generation process.

Section 3.1 presents the main features and the architecture of TPC-W. After that, we introduce the main functionalities of GUERNICA and its architecture in Section 3.2. Finally, Section 3.3 shows the devised architecture of integrating TPC-W and GUERNICA.

### 3.1 TPC Benchmark™ W

TPC Benchmark™ W (TPC-W) is a transactional web benchmark that models a representative e-commerce system, specifically an on-line bookstore environment [7]. The benchmark reproduces the workload generated by multiple on-line browser sessions over a web application, which serves dynamic and static contents related to the bookstore activities (e.g., catalog searches or sales).

TPC-W provides a standard environment that is independent of the underlying technology, designed architecture and deployed infrastructure. Also, it has been commonly accepted by the scientist-technical community in many research works [13], [18], [19]. As shown in Figure 1, TPC-W presents a client-server architecture. Remote Browser Emulators (RBE) are located in the client side and generate workload towards the e-commerce web application, which is located in the server side (E-commerce server). With the aim of reproducing a representative workload, the emulators simulate real users' behavior when they navigate the website by using CBMG model. The server hosts the system under test (Server Under Test), which consists of: i) a web server and its storage of static contents, and ii) an application server with a database system to generate dynamic content. The payment gateway (Payment Gateway Emulator) represents an entity to authorize users' payments. These three main architecture components are connected together by a dedicated network.

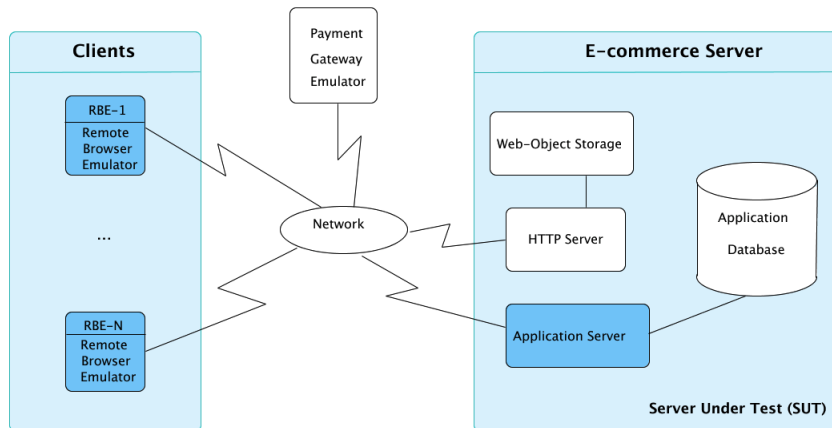


Figure 1: TPC-W architecture

A TPC-W Java implementation developed by the UW-Madison Computer Architecture Group [20] was selected as framework of our testbed. As shown in Figure 2, the architecture client side is a Java console application that provides two interfaces for workload generation; an emulator for clients simulation (EB) and a factory (EBFactory) to create, configure and manage them. These interfaces allow to define new processes for workload generation. The server side was developed as a Java web application made of a set of `Servlets`. Each `Servlet` resolves client requests by requiring the database information.

### 3.2 GUERNICA

GUERNICA (Universal Generator of Dynamic Workload under WWW Platforms) is a web workload generator developed as a result of the cooperation among the Web Architecture Research Group (Polytechnic University of Valencia), Intelligent Software Components, and the Institute of Computer Technology; thereby, bridging the gap between academia and industry.

The main benefit of GUERNICA lies in its workload generation process, which is based on Dweb model (Dynamic web workload model) [6] in order to generate dynamic web workload in a more accurate and appropriate way, taking into account the three mentioned challenges in dynamic workload characterization. The navigation concept defines users' behavior while they interact with the web, and it facilitates the characterization of users' dynamism in the navigations. By the other hand, the concept of workload test models a set of navigations, which defines behaviors of a user by considering the capability of changing them with time.

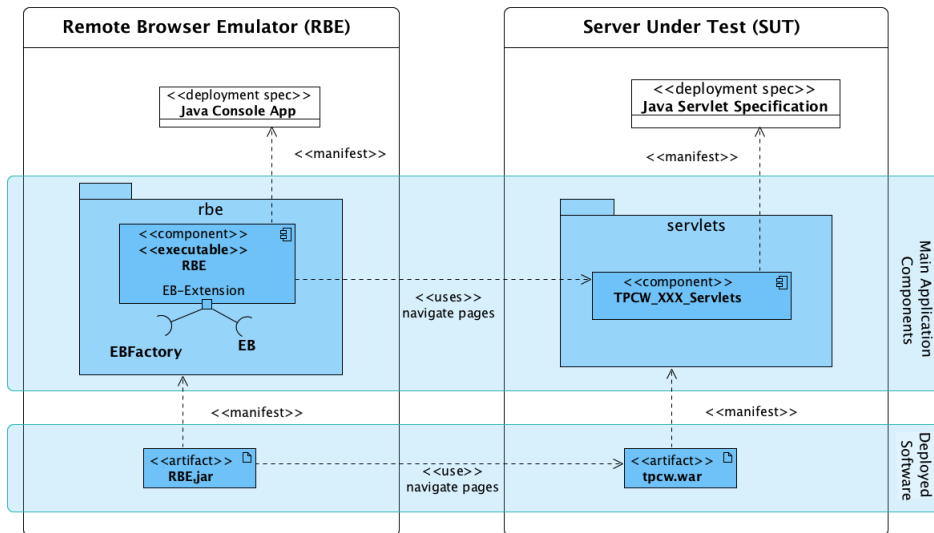


Figure 2: Main software components of TPC-W Java implementation

As shown in Figure 3, GUERNICA is a software made up of three main applications: workload generator, performance evaluator and performance tests planner. Each application permits an autonomous distribution among the different machine nodes of the main activities in the evaluation of performance and functional specifications of a web application.

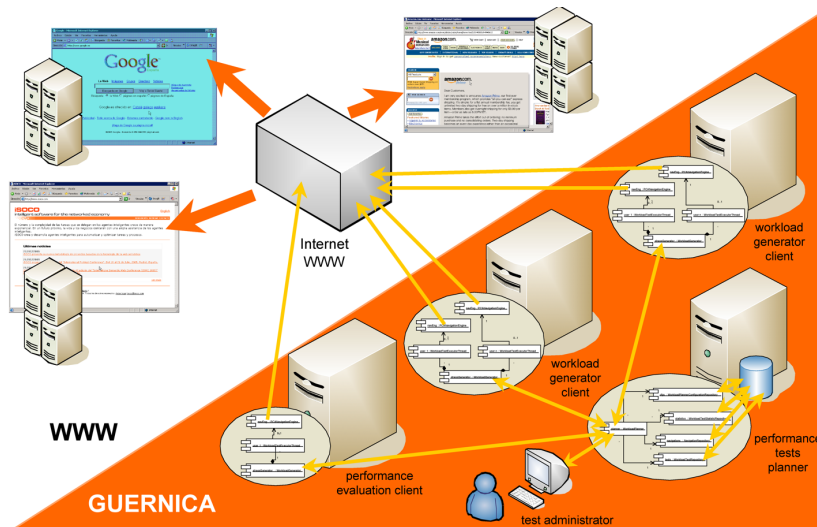


Figure 3: Main applications of GUERNICA

These three applications are defined in a software architecture based on components, as Figure 4 depicts. The main component of the architecture is the **GUERNICA.core**, which carries out the workload generation process by using Dweb. The navigation and workload test concepts are implemented by **WorkloadTest** and **WorkloadNavigation** interfaces, respectively. The **NavigationEngine** simulates the user behavior; its configuration is defined in terms of Dweb, and it is stored in a repository called **WorkloadTestRespository**. A centralized access to **GUERNICA.core** is possible by using **CoreManager**.

### 3.3 Integration architecture

The architecture of the TPC-W and GUERNICA integration is depicted in Figure 5, which is organized in three main layers:

- The top layer is defined at the client side of TPC-W and supplies the two interfaces related to the workload generation process (EB, EBFactory), as introduced in Section 3.1.
- The bottom layer is related to the process of workload generation in GUERNICA, detailed in Section 3.2.

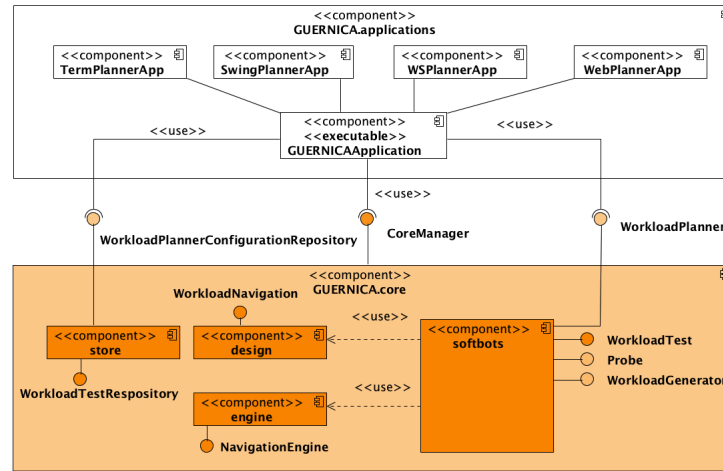


Figure 4: Main software components of GUERNICA

- Finally, the intermediate layer defines the integration between TPC-W and GUERNICA. The integration is provided by an independent Java library named TGI. This library implements a new type of emulated browser (i.e., DwebEB) that uses the GUERNICA core to reproduce users' dynamic behavior in the workload generation process. In order to simplify the new emulated browser, a workload generation engine (i.e., DwebExecutorEngine) is implemented to carry out the generation process. A browser factory (i.e., DwebEBFactory) was also developed to manage the creation and configuration of new browsers.

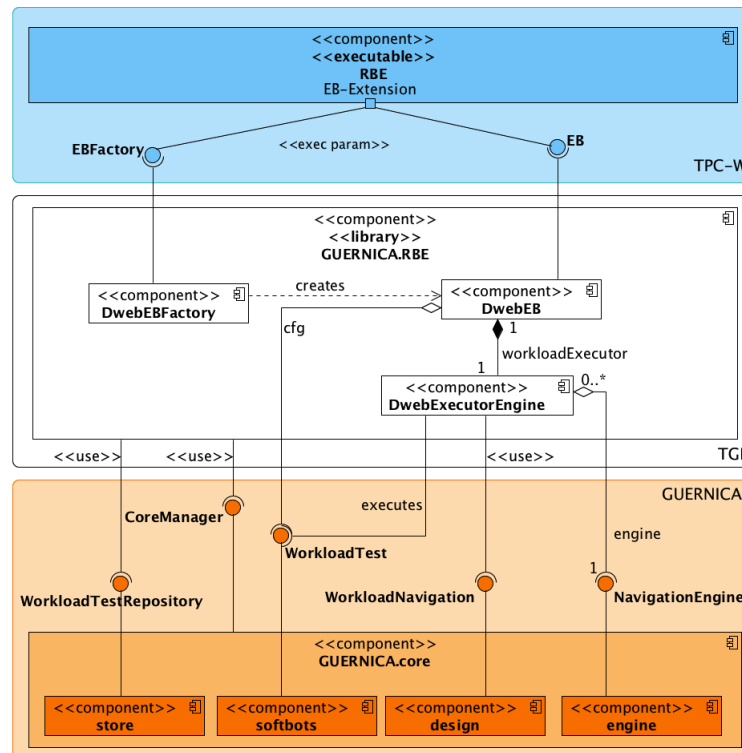


Figure 5: Architecture of the TPC-W and GUERNICA integration

#### 4 Testbed validation

This section compares the main functionalities and behavior of the devised testbed against TPC-W. We found that both implementations present similar behavior in traditional web workloads. Section 4.1 and

Section 4.2 describe the experimental setup and the main measured performance metrics in this process, respectively. The validation process is discussed in Section 4.3.

#### 4.1 Experimental setup

The experimental setup used in this study is a typical two-tier configuration that consists of an Ubuntu Linux Server back-end and an Ubuntu Linux client front-end tier. The back-end runs the TPC-W server part, whose core is a Java web application (TPC-W `web app`) deployed on the Tomcat web application server. Requests to static content of this web application, such as images, are served by the Apache web server, which redirects requests for dynamic content to Tomcat. TPC-W web application generates this type of content by fetching data from the MySQL database. On the other side, the front-end generates the workload using conventional or dynamic models. Both web application and workload generators are run on the SUN Java Runtime Environment 5.0 (JRE 5.0). Figure 6 illustrates the hardware and the software used in the experimental setup.

Given the multi-tier configuration of this environment, system parameters (both in the server and in the workload generators) have been properly tuned to avoid that middleware and infrastructure bottlenecks interfere the results. TPC-W has been configured with 100 emulated browsers and a large number of items (100,000) that forced us to review the tuning of accessing to the database (e.g., pool connection size), static content service by Apache (e.g., number of workers to attend HTTP requests), or dynamic content service by Tomcat (e.g., number of threads providing dynamic contents). For each workload, measurements were performed during several runs having a 15-minute warm-up phase and a 30-minute measurement phase.

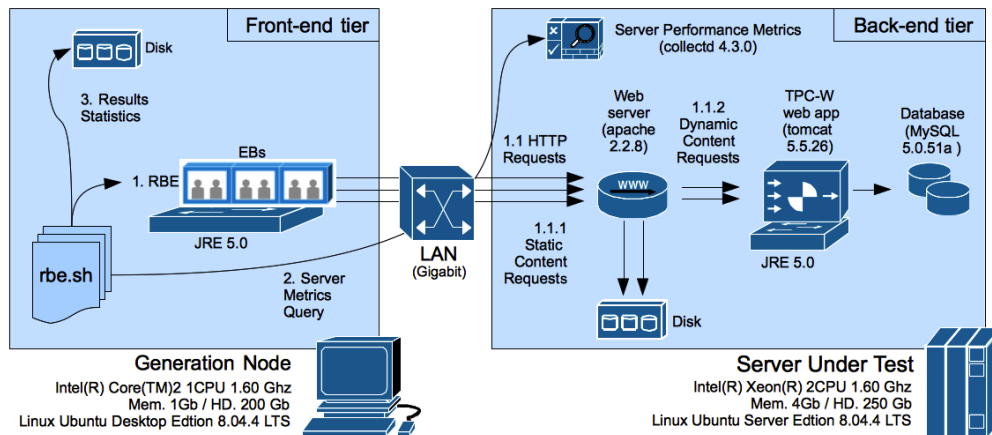


Figure 6: Experimental setup

#### 4.2 Performance metrics

Table 1 summarizes the performance metrics available in the experimental setup. The main metrics measured on the client side are the response time and the total requests per page. On server side, the study collects server performance statistics required by TPC-W specification (i.e., CPU and memory utilization, database I/O activity, system I/O activity, and web server statistics) as well as other optional statistics that allow a better understanding of the system behavior under test and the techniques to improve performance when applying a dynamic web workload. The collected metrics can be classified in two main groups: metrics related with the usage of main hardware resources, and performance metrics for the software components of back-end. We use a middleware named `collectd`<sup>1</sup> which collects system performance statistics periodically, and allows us to standardize the performance evaluation.

<sup>1</sup><http://collectd.org/>

Resource		Metric	Description/Formula
Client Side		WIRT	Web Interaction Response Time (WIRT) is defined by TPC-W by $t2 - t1$ , where $t1$ is the time measured at emulated browser when the first byte of the first HTTP request of the web interaction is sent by the emulated browser to the server, and $t2$ is the time measured at emulated browser when the last byte of the last HTTP response that completes the web interaction is received by the emulated browser from the server.
		$Req_{page}$	Requests per Page ( $Req_{page}$ ) are the total number of connections for a page requested by emulated browsers and accepted by the server.
		$\overline{WIRT}$	$\overline{WIRT} = \frac{\sum_{i \in Pages} WIRT_i * Req_i}{\sum_{i \in Pages} Req_i}$
Server Side	Hardware	CPU	$U_{CPU}$
		Memory	$U_{memory}$
		Disk	$U_{disk}, X_{disk}$
		Network	$U_{net}, X_{net}$
	Software	Apache	$X_{apache}, CPU_{apache}, MEM_{apache}$
		Tomcat	$X_{tomcat}, CPU_{tomcat}, MEM_{tomcat}$
		MySQL	$X_{mysql}, CPU_{mysql}, MEM_{mysql}$
			Metrics for hardware resources include utilization and throughput for the disk and the network.
			Performance metrics for software components of server include: throughput, CPU and memory utilization, processes or threads, etc

Table 1: Performance metrics classification

### 4.3 Results

This section compares the devised testbed against TPC-W. According to the TPC-W specification, the full CBMG for the on-line bookstore consists of 14 unique pages and the associated transition probability. Figure 7 depicts an example of a simplified CBMG for the search process of on-line bookstore, showing that customers may be in several pages (i.e., Home, Search request, Search results, and Product Detail) and may transit among these pages according to the arcs' weight. The numbers in the arcs indicate the probability of making that transition. For example, the probability of going to the Product detail page from the Search results page is 0.6195 and means that after a search, regardless of whether a list of books is returned or not, the Product detail page will be visited 61.95% of the cases. This detailed book is a result of the search or a member of the banner for latest books, which is included in all pages of the website.

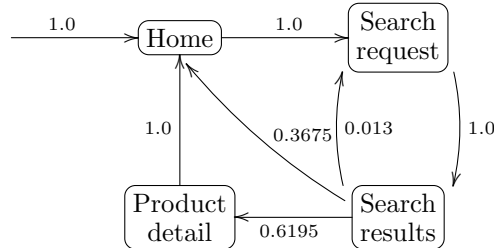


Figure 7: Example of a simplified CBMG

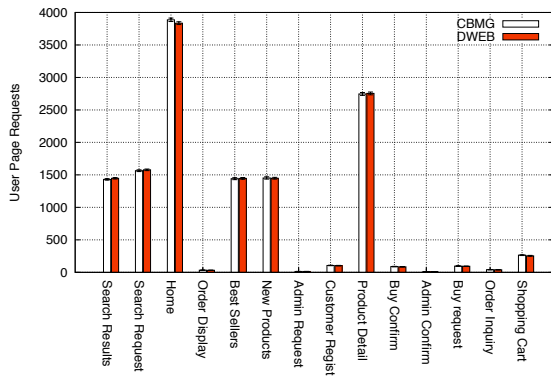
Three scenarios are defined by the TPC-W specification when characterizing the web workload: shopping, browsing, and ordering. The shopping scenario presents browsing and ordering activities; the browsing scenario consists of significant browsing activity and relatively little ordering activity; and the ordering scenario mixes significant ordering activity and relatively little browsing activity. In order to consider these scenarios, TPC-W needs to define three web workloads based on different CBMGs.

Regarding the checking test, we contrasted both workload characterization approximations (i.e., CBMG and Dweb) for each scenario. This test considers 50 emulated browsers because the Java implementation of the TPC-W generator presents some limitations in the workload generation process. The measurements were performed repeating 50 runs and obtaining confidence intervals with a 99% confidence level.

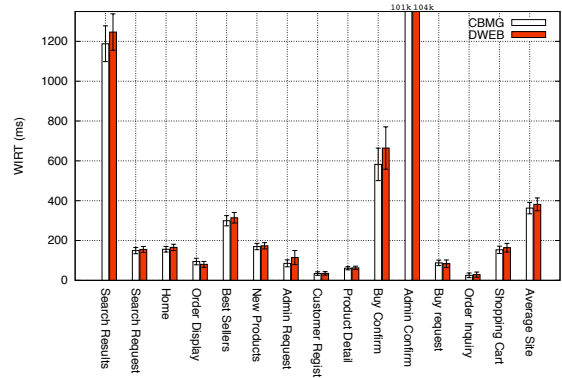
For illustrative purposes, this section presents results of a subset of the most significant metrics when running TPC-W for different scenarios (i.e., shopping, browsing, and ordering) defined by CBMG and Dweb. Note that we are able to model the same workloads by using only the navigation concept of the Dweb model, and disabling all the parameters used to include user dynamism in the workload characterization.

Figures 8 and 9 depict client and server performance metrics for the shopping scenario. As shown in Figure 8(a), both approximations generate a similar number of page requests. Figure 8(b) shows that the response time is, on average, 5% higher for Dweb than for CBMG, because some pages (e.g., Search results or Buy confirm) present very wide confidence intervals in this scenario. However, this difference does not affect the server performance metrics, since as observed in Figure 9 the highest utilization is lower than 40% in both cases. The CPU and memory usages are rather low and similar in both cases (see Figure 9(a)). Incoming and outgoing traffics do not increase network utilization more than 2% in any workloads (see Figure 9(b)). Finally, the disk utilization is lower than 0.2% in both workloads (not represented).



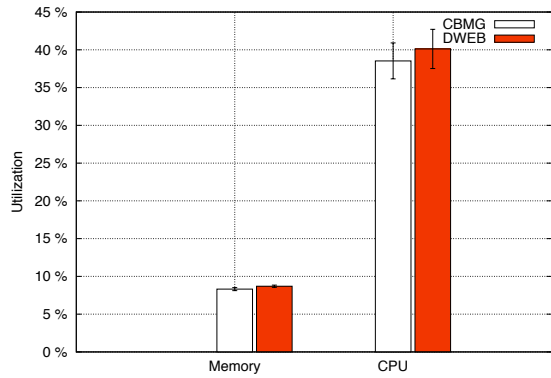


(a) User page requests

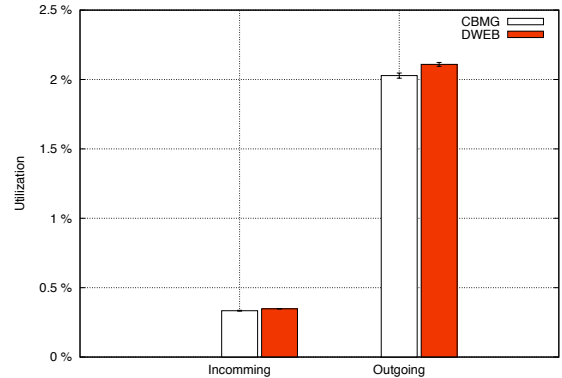


(b) WIRT

Figure 8: Client metrics obtained for the shopping scenario in the testbed



(a) Server memory and CPU utilization



(b) Server network utilization

Figure 9: Server metrics obtained for the shopping scenario in the testbed

Browsing scenario results are illustrated by Figures 10 and 11. Both workloads generate a similar number of page requests and response time as shown in Figure 10. On the other hand, the server is characterized by a middle level of stress in both cases. CPU usages are 50%, while usages for memory, network and disk are low, as observed in Figure 11.

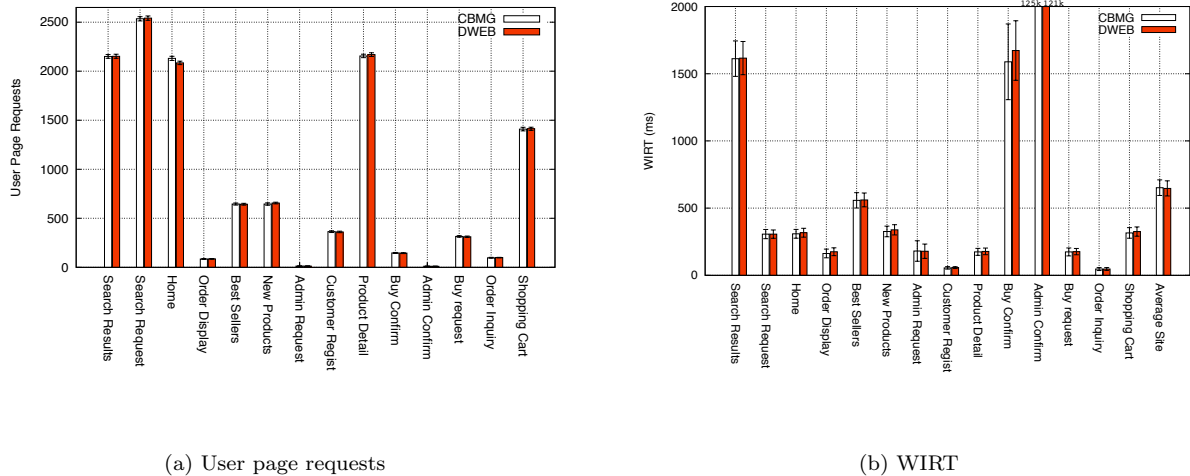


Figure 10: Client metrics obtained for the browsing scenario in the testbed

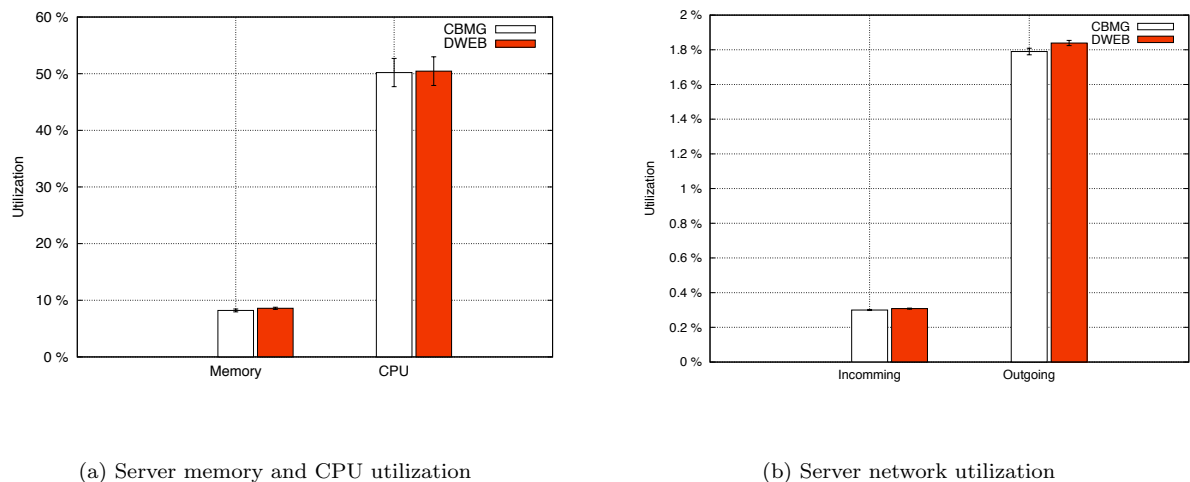


Figure 11: Server metrics obtained for the browsing scenario validation

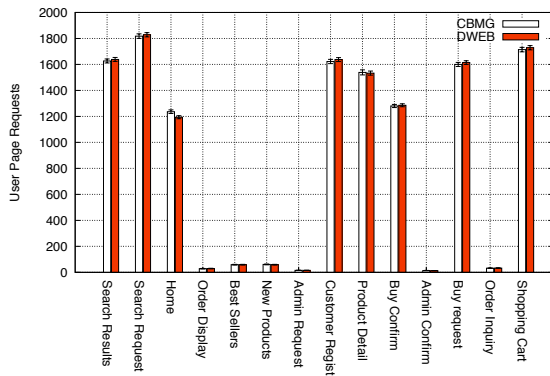
Figure 12 and Figure 13 depict results for the ordering scenario. The former shows that both workloads present similar levels on client metrics. The latter presents how the highest server's utilization is lower than 40% in both cases.

Finally, we can conclude that the Dweb model and GUERNICA can generate accurate traditional workloads for web performance studies based on TPC-W. Moreover, due to their designs, our new testbed can be used to generate web workloads with users' dynamic behavior.

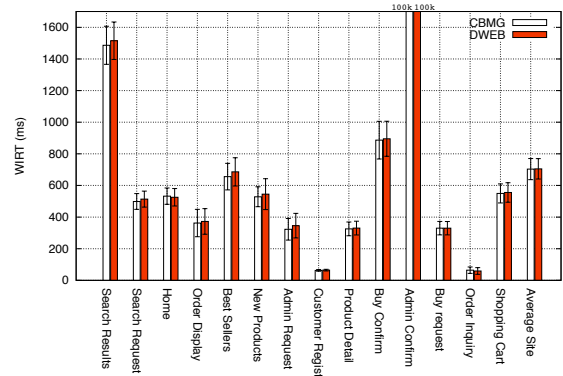
## 5 Conclusions and future work

The evolution of the World Wide Web from the first generation to the second and third generations has involved a new paradigm where users are no longer passive consumers, but they become participative contributors to the dynamic web content accessible on the web. Consequently, users are characterized by a more dynamic behavior. This dynamism is the main shortcoming to model representative web workload to carry out performance evaluation studies.

This paper has proposed a new e-commerce testbed with the ability of generating web dynamic workload on web performance evaluation, by considering user dynamic behavior in an appropriate and accurate way. To this end, we used GUERNICA as a dynamic workload generator, and integrated it with TPC-W benchmark. Also, we contrasted the new testbed main functionalities and behavior against the benchmark.

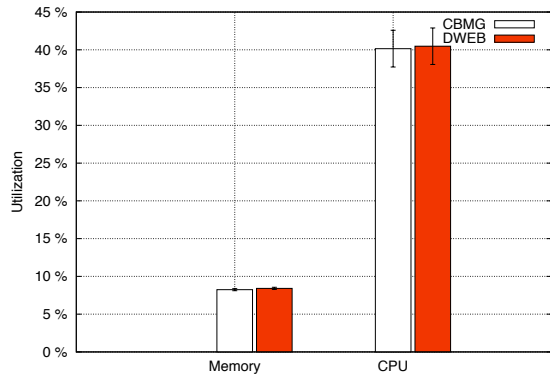


(a) User page requests

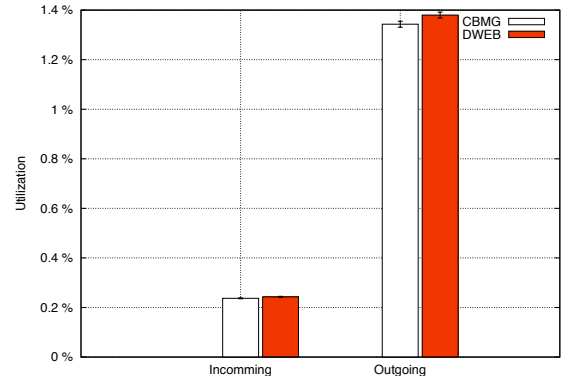


(b) WIRT

Figure 12: Client metrics obtained for the ordering scenario validation



(a) Server memory and CPU utilization



(b) Server network utilization

Figure 13: Server metrics obtained for the ordering scenario validation

As for future work we plan to demonstrate that our workload model is a more valuable alternative, because it is able to reproduce user dynamic behavior on workload characterization. With this aim, we should quantify the effect of using dynamic workload on web performance evaluation studies.

## Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation under grant TIN-2009-08201.

## References

- [1] P. Rodriguez, "Web Infrastructure for the 21st Century," in *18th International World Wide Web Conference*, 2009.
- [2] G. Cormode and B. Krishnamurthy, "Key differences between Web 1.0 and Web 2.0," *First Monday Journal*, vol. 13, no. 6, January 2008.
- [3] T. O'Reilly, "What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software," *O'Reilly Online Publishing*, 2005.
- [4] G. Abdulla, "Analysis and Modeling of World Wide Web Traffic," Ph.D. dissertation, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia, May 1998.
- [5] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. Performance Evaluation Review, July 1998, pp. 151–160.
- [6] R. Peña-Ortiz, J. Sahuquillo, A. Pont, and J. A. Gil Salinas, "Dweb model: representing Web 2.0 dynamism," *Computer Communications Journal*, vol. 32, no. 6, pp. 1118–1128, April 2009.
- [7] "TPC BENCHMARK(TM) W Specification. Version 1.8," Transaction Processing Performance Council, Tech. Rep., February 2002.
- [8] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, "Changes in Web client access patterns: Characteristics and caching implications," *World Wide Web*, vol. 2, pp. 15–28, 1999.
- [9] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392–403, January 2001.
- [10] H. Weinreich, H. Obendorf, E. Herder, and M. Mayer, "Off the beaten tracks: exploring three aspects of web navigation," in *15th international conference on World Wide Web, WWW 2006*, 2006, pp. 133–142.
- [11] S. Goel, A. Broder, E. Gabrilovich, and B. Pang, "Anatomy of the long tail: ordinary people with extraordinary tastes," in *3rd ACM international conference on Web search and data mining*, 2010, pp. 201–210.
- [12] D. A. Menascé and V. A. F. Almeida, *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.
- [13] R. C. Dodge JR, D. A. Menascé, and D. Barbará, "Testing e-commerce site scalability with TPC-W," in *Computer Measurement Group Conference*, 2001, pp. 457–466.
- [14] F. Duarte, B. Mattos, J. Almeida, V. A. F. Almeida, M. Curiel, and A. Bestavros, "Hierarchical characterization and generation of blogosphere workloads," Computer Science, Boston University, Tech. Rep., 2008.
- [15] M. Shams, D. Krishnamurthy Ph.D, and B. Far, "A model-based approach for testing the performance of web applications," in *International Workshop on Software Quality Assurance*. ACM, November 2006, pp. 54–61.
- [16] F. Benevenuto, T. R. de Magalhães, M. Cha, and V. A. F. Almeida, "Characterizing user behavior in online social networks," in *9th ACM SIGCOMM conference on Internet measurement conference*, 2009, pp. 49–62.
- [17] R. Peña-Ortiz, J. Sahuquillo, J. A. Gil Salinas, and A. Pont, "WEB WORKLOAD GENERATORS: A survey focusing on user dynamism representationa," in *International Conference on Web Information Systems and Technologies*, May 2011.
- [18] C. Amza, A. Chanda, and A. Cox, "Specification and implementation of dynamic web site benchmarks," *IEEE International Workshop on Workload Characterization*, pp. 3–13, 2002.
- [19] D. García and J. García, "TPC-W e-commerce benchmark evaluation," *Computer*, vol. 36, no. 2, pp. 42–48, February 2003.
- [20] H. W. Cain, R. Rajwar, M. Marden, and M. H. Lipasti, "An architectural evaluation of Java TPC-W," in *7th International Symposium on High Performance Computer Architecture*, 2001, p. 229.