Experimenting With the TPC-W E-commerce Benchmark

Mehdi Khouja, Farouk Kamoun
Université de la Manouba, Ecole Nationale des Sciences de l'Informatique (ENSI),
La Manouba, Tunisia, 2010
mehdi.khouja@cristal.rnu.tn, frk.kamoun@planet.tn

Catalina M. Lladó, Ramon Puigjaner
Universitat de les Illes Balears, Departament de Matemàtiques i Informàtica,
Palma de Mallorca, Spain, 07071
cllado@uib.es, puxi@uib.es

Abstract

The success of an e-commerce site highly depends on its performance characteristics. These, are very difficult to asses for the given software and hardware characteristics of the specific system. The TPC-W is a benchmark aimed at evaluating e-commerce sites. This paper presents an implementation of this benchmark and the experimentation process that has been carried out in order to evaluate it. A full factorial experimental design has been used with the factors, number of emulated browsers, their profile and the number of processors in the server machine. The analysis of the results is done in terms of the TPC-W main metric, Web Interaction Per Second (WIPS) and it shows the effect of the variation of the factors above mentioned on the TPC-W throughput.

Keywords: TPC-W, Benchmarking, E-commerce, Computer Evaluation.

1 INTRODUCTION

Nowadays, the Internet is deeply affecting the commerce. However, e-commerce sites are at a risk of being overwhelmed by large numbers of customers and therefore offering them a very bad performance. Hardware and software vendors and commerce service providers need to know the best configuration to use in order to fit the customers demand. One possibility for the sizing and capacity planning of computer systems is the use of benchmarks. TPC-W (specified by the Transaction Processing Performance Council), is a benchmark aimed at evaluating sites that support e-commerce activities. The business model [4] of this benchmark is an e-commerce web site that sells products over the Internet. The site provides e-business functions that allow customers to browse through selected products (e.g., best sellers or new products), search information on existing products, see product detail, place an order, or check the status of a previous order.

The benchmark results are highly depending on the benchmark specification, workload, and design and implementation of the benchmark application. Therefore experimentation with the benchmark needs to be done in order to evaluate this dependency. Clearly, the first step is to design and develop a TPC-W benchmark application, taking into account that the results can vary between different application implementations. This paper presents a TPC-W benchmark implementation that has been developed at the Universitat de les Illes Balears and the experimental design process done in order to evaluate the utility of this benchmark.

Related work can be found in [1], where a TPC-W implementation and a set of experiments conducted to study how the response time varies as the request arrival rate increases are presented. Another TPC-W implementation and its evaluation is presented in [2]. Nevertheless, those implementations are different from the one presented in this work and the experiments they perform also differ substantially.

The paper is structured as follows: while section 2 describes the TPC-W specification, section 3 shows the main characteristics of the specific implementation of the TPC-W and the system configuration that has been used to carry out the experimentation process. Following, section 4 covers the experimental methodology and its application to the case under study and section 5 analysis and discusses the experimental results obtained. Finally, conclusions and future work are covered in section 6.

2 TPC-W OVERVIEW

TPC-W is a transactional Web benchmark with a client-server architecture which represents a typical e-commere environment. It covers both, the server application and the workload generator. The e-commerce environment is characterized by multiple on-line browsing sessions, static and dynamic web pages services and accesses and updates to a database [6]. The site maintains a catalog of items that can be searched by a customer. TPC-W specifies that the site maintains a database with information about customers, items in the catalog, orders, and credit card transactions.

Three components constitute the TPC-W system: a set of Emulated Browsers (EB) which is called RBE (Remote Browser Emulator), a system under test (SUT), which mainly includes a web and a database server, and a Payment Gateway Emulator (PGE). Figure 1 shows the TPC-W environment.

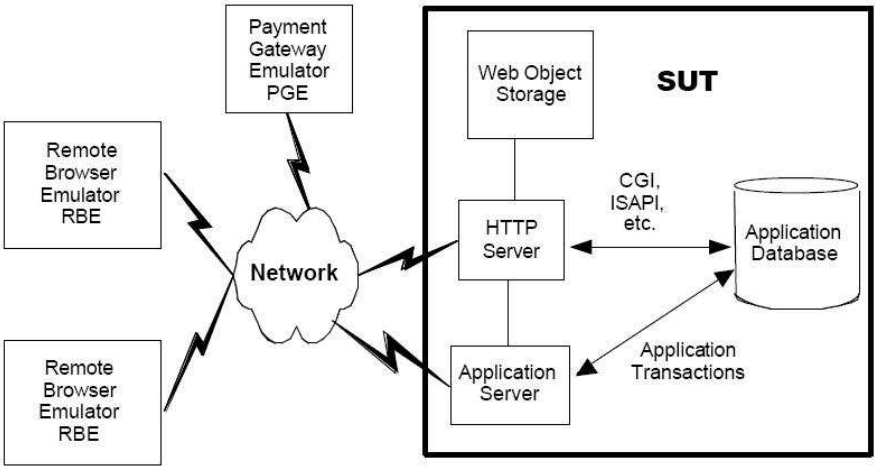

Figure 1: TPC-W environment

Following, the server application (SUT) and the workload generator (RBE) are described in more detail, as well as the performance metrics that the benchmark reports.

2.1 The Workload Generator

The emulated browsers drive the TPC-W workload. They emulate human customers sending and receiving HTML contents using HTTP and TCP/IP over a network connection. The EBs can browse, search the site's bookstore, view product details, add products to the shopping cart and buy added products. The customer activities are categorized into 14 specific web interactions, though TPC-W also classifies those into two broad categories:

- *Browse* interactions involve browsing and searching but no product ordering activity.
- *Order* interactions involve product ordering activities only.

Each of the 14 web interactions are characterized by the number of images included in the response page, the number of database table joints that require, and a maximum response time. In order to consider an interaction as valid, it needs to be executed within this maximum value. Table 1 shows these characteristics for each TPC-W web interaction.

On the client site of the benchmark, each emulated customer submits a series of requests within a user session. TPC-W defines the minimum user session duration (USMD), which is generated at the starting of a session from a exponentially distributed function with mean equal to 15 minutes. The user session duration is defined as the time elapsed between the first request executed by the EB and the current time. Therefore, when the user session duration reaches or passes the USMD, the session finishes. At the end of a session the EBs must close all the TCP/IP and SSL (Secure Sockets Layer) connections and restart a new session with the generation of a new USMD.

The EBs wait a period of time between two consecutive requests in order to simulate the user think time (TT). This time is generated using a negative exponential distribution function with mean equal to 7 seconds.

A user session is characterized by a browsing to ordering interaction ratio. TPC-W specifies three session types:

Table 1: Characteristics of TPC-W Web Interactions

Name	Dynamic Generation	Number of Joints	Number of Images	Max Response Time (s)	Interaction Type
Admin Confirm	Yes	4	5	20	Order
Admin Request	Yes	2	6	3	Order
Best Seller	Yes	3	9	5	Browse
Buy Confirm	Yes	1	2	5	Order
Buy Request	Yes	1	3	3	Order
Customer Registration	No	N/A	4	3	Order
Home	Yes	1	9	3	Browse
New Product	Yes	2	9	5	Browse
Order Display	Yes	1	2	3	Order
Order Inquiry	No	N/A	3	3	Order
Product Detail	Yes	2	6	3	Browse
Search Request	No	N/A	9	3	Browse
Search Result	Yes	2	9	10	Browse
Shopping Cart	Yes	1	9	3	Order

- Browsing mix: 95% of browsing and 5% of ordering
- Shopping mix: 80% of browsing and 20% of ordering
- Ordering mix: 50% of browsing and 50% of ordering

The customer behaviour model graph (CBMG), shown in Figure 2, specifies the navigational behaviour of the EBs through the web site. Each interaction represents a state in the graph. The transition from a state to an another is characterized by a probability of transition that depends on the EB session type. For instance, the Shopping Cart state represents a state in which items can be added or deleted from the shopping cart and customers have to go through the Customer Registration state before they can reach the Buy Request (customers need to register with the site before placing an order).

2.2 The System Under Test

The System Under Test (SUT) comprises all the components which are part of the application being emulated. This includes network connections, Web servers, application servers and database servers. It does not include the RBE or the PGE [6]. The two main components of the SUT are the database and the web server.

The TPC-W database consists of 8 tables. These tables store users, authors, books, orders and credit cards information. The database size depends on two scalability parameters: the number of emulated clients and the number of bookstore items. The second parameter must be chosen from a list of five values specified by TPC-W: 1K, 10K, 100K, 1M and 10M. Database scalability rules (see [3]) are shown in Figure 3. Database interactions must verify the ACID properties: Atomicity, Consistence, Isolation and Durability.

When the SUT needs to finalize an order interaction, it uses a third component which is the PGE. The PGE generates an authentification identifier and then the buying transaction is accomplished. The connections between the SUT and the PGE are over the secure socket layer.

2.3 Performance Metrics

TPC-W has two types of performance metrics: a throughput metric and a cost to throughput ratio metric. There are three throughput metrics depending of the type of session. The main one measures the average number of valid Web Interactions executed Per Second during a simulation interval in which all the sessions are of the shopping type (WIPS). As said above, valid interactions are those which the response time is under the maximum response time specified by TPC-W (see Table 1). There are also two secondary throughput metrics corresponding to the other TPC-W mixes: the WIPb, measures the average number of valid Web Interactions completed during an interval in which all sessions are of the browsing type. The other, called WIPSo, measures the average number of valid Web Interactions per second completed during an interval in which all sessions are of the ordering type.

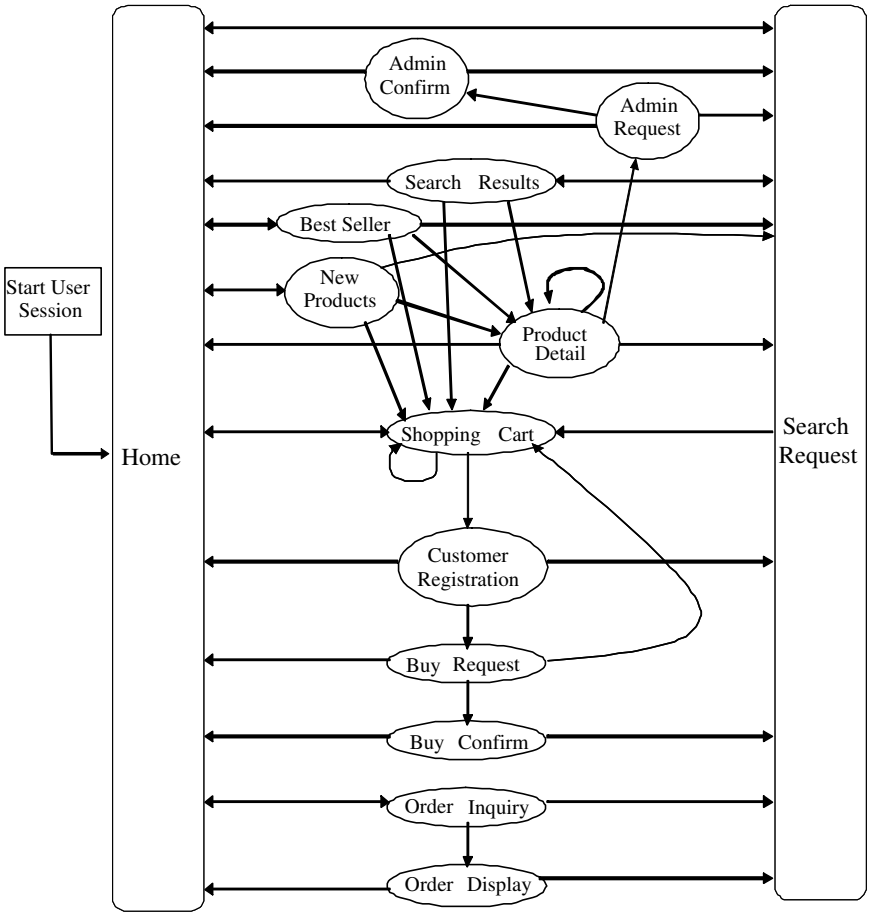

Figure 2: TPC-W Customer Behaviour Model Graph

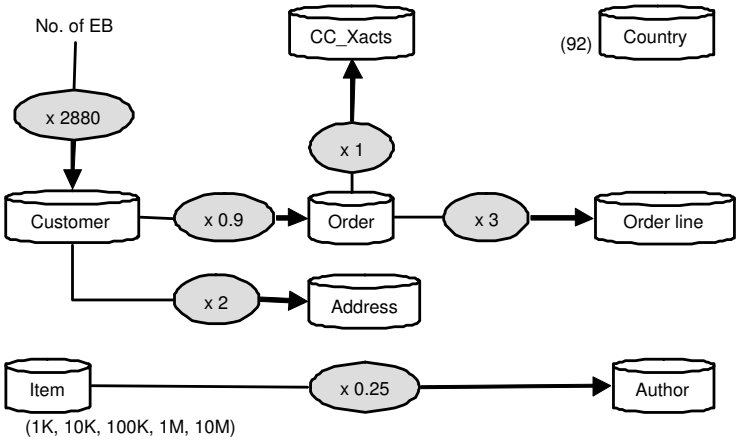

Figure 3: Database Scalability Rules

The TPC-W specification states that in order to consider the value reported for the WIPS as valid, it must satisfy the following bounding conditions: the upper bound corresponds to the WIPS that would be reported for a response time equal to zero and the lower bound is set as 50% of the upper bound. Since the user think time has a mean value of 7 seconds, the upper bound is $EB/7$ and the lower bound is $EB/14$.

The cost related metric specified by TPC-W is $\$/WIPS$ and indicates the ratio between the total cost of the system under test and the number of WIPS measured during a shopping interval. Total cost includes purchase and maintenance cost for all hardware and software components.

Other performance metrics that can be reported are CPU and memory utilization, database I/O activities, etc.

3 TPC-W IMPLEMENTATION AND SYSTEM CONFIGURATION

The evaluated implementation of the TPC-W has been developed in Java. Figure 4 shows the specific architecture used in this implementation.

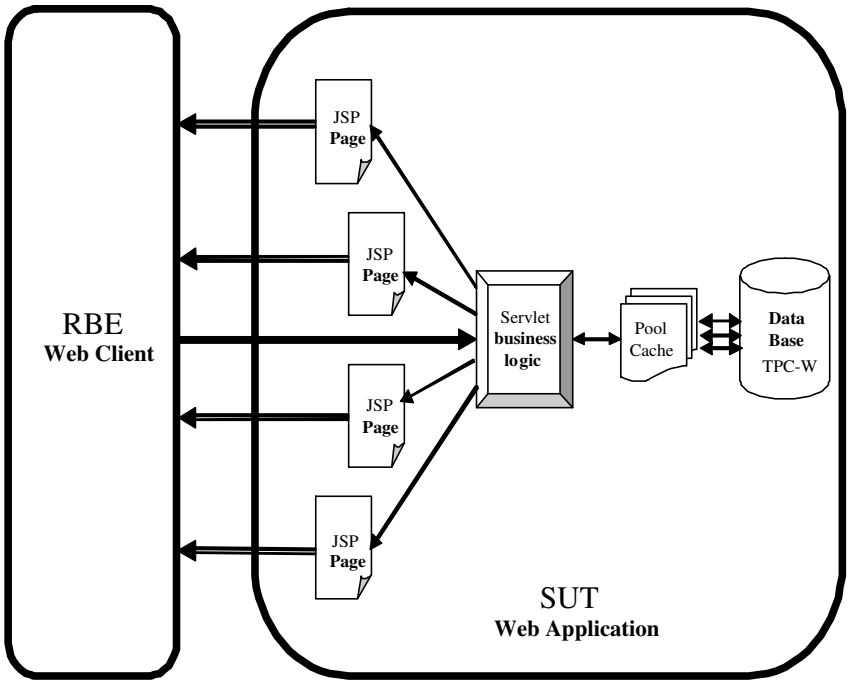

Figure 4: TPC-W Architecture

The RBE component has a graphical interface allowing the experimenter to introduce the following execution parameters:

- Web Server IP address.
- Number of emulated clients.
- Type of mix.
- Simulation time.

The TPC-W web site is implemented using Java Server Page (JSP). The web site is based on the servlet-centric approach. Client requests arrive to the central servlet, which takes charge of processing them, retrieving the required information from the data base through the PoolMan (an embeddable object pooling and caching library) and, once done all required actions, routing the response information to the JSP front-end pages. The access to the database is done through JDBC (Java Database Connectivity).

For the experimentation process, three different computers are used, one for each component of TPC-W (RBE, SUT and PGE). Table 2 shows the SUT configuration in the experimentation. The number of bookstore items in the database is 1000.

Table 2: SUT Configuration

Component	Description
Operating System	Microsoft Windows 2000 Server
CPU	2 x Intel Pentium XEON 2.4 GHZ
RAM	2 GB
HTTP Server	Apache 2.0.44
Servlet Engine	Tomcat 3.3.1
Java Virtual Machine	Java 2 Runtime Environment, Standard Edition 1.3.1
Database	Oracle 9i Enterprise Edition Release 9.2

4 EXPERIMENTAL DESIGN FORMULATION

Once the measurement tool is developed and the physical configuration established, the logical configuration of the measurement environment has to be specified. Next, the measurement phase has to be carried out and the reliability of the results has to be tested. This section describes the workload characteristics, as well as the subsequent experimental design formulation.

The measure under study is the main throughput metric, WIPS. A full-factor experimental design is used since there are three factors which influence we are intending to study, namely, *Number of Processors*, *Number of Emulated Clients*, *Emulated Clients Profile*. Table 3 shows the different levels for these factors, making a $2 * 14 * 3$ factorial design.

Table 3: Design Factors and Levels

Factor	Levels
Number of Processor	2 levels : 1, 2
Number of Emulated Clients	14 levels : 10, 20, 30, ..., 130
Emulated Clients Profile	3 levels : Shopping mix, Browsing mix, Ordering mix

Further, to isolate the possible effects due to the instability of the real system where the benchmarks are executed, more than one measure for each experiment (where each combination of the different levels of factors is an experiment) is obtained. In this study, five measures (or replications) for each experiment have been used.

5 EXPERIMENTAL RESULTS

The full factorial design with replications is used because it identifies the influence of each of the factors as well as the interaction between these factors [5]. Additionally, the replications allow for the isolation of the influence of experimental errors.

The first step in the experimental design process is the analysis of variance (ANOVA). This analysis revealed that the influence of the number of EBs, their profile and their interaction, on the throughput value obtained (WIPS) is statistically significant. As shown in Table 4, ANOVA allocates the total variation of WIPS due to the type of clients as 42.15% (SS_M/SS_T), due to the number of EBs as 27.43% (SS_B/SS_T), and due to the interactions between them as 25.38% (SS_{MB}/SS_T). However, the effect due to the variation of the number of processors is 0.17% (SS_P/SS_T).

5.1 The Type of Mix Effect

Figure 5 shows the variation of the WIPS depending on the number of EBs, for the three different types of mixes when the SUT is running with one processor. Similarly, figure 6 shows the variation of the WIPS depending on the number of EBs, for the three different types of mixes when the SUT is running with two processors.

As shown in the above mentioned graphs, the WIPS increases more or less linear as the number of EBs increases until it reaches a maximum value from which the WIPS drop, due to the saturation of the system. The number of EBs in the system when this maximum is reached it varies substantially depending on the type of mix we are looking at. For instance, when executing the SUT with one processor, the maximum occurs when there are 90 clients for the *shopping* mix, 110 clients for the *browsing* mix and 30 clients for the *ordering* mix. Similar results are shown when executing the SUT with 2 processors.

Table 4: Analysis of Variance for the throughput

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F	P-value
Processor	$SS_P = 10,46$	1	10, 46	13, 94	0, 0002
Mix	$SS_M = 2557,62$	2	1278, 81	1704, 29	0, 00
EB	$SS_B = 1664,89$	12	138, 74	184, 90	0, 00
Processor∗Mix	$SS_{PM} = 16,19$	2	8, 098	10, 79	0, 00
Processor∗EB	$SS_{PB} = 11,98$	12	0, 99	1, 33	0, 19
Mix∗EB	$SS_{MB} = 1540,01$	24	64, 167	85, 51	0, 00
Processor∗Mix∗EB	$SS_{PMB} = 32,41$	24	1, 35	1, 8	0, 013
Error	$SS_E = 234,10$	312	0, 75		
Total	$SS_T = 6067,72$	389			

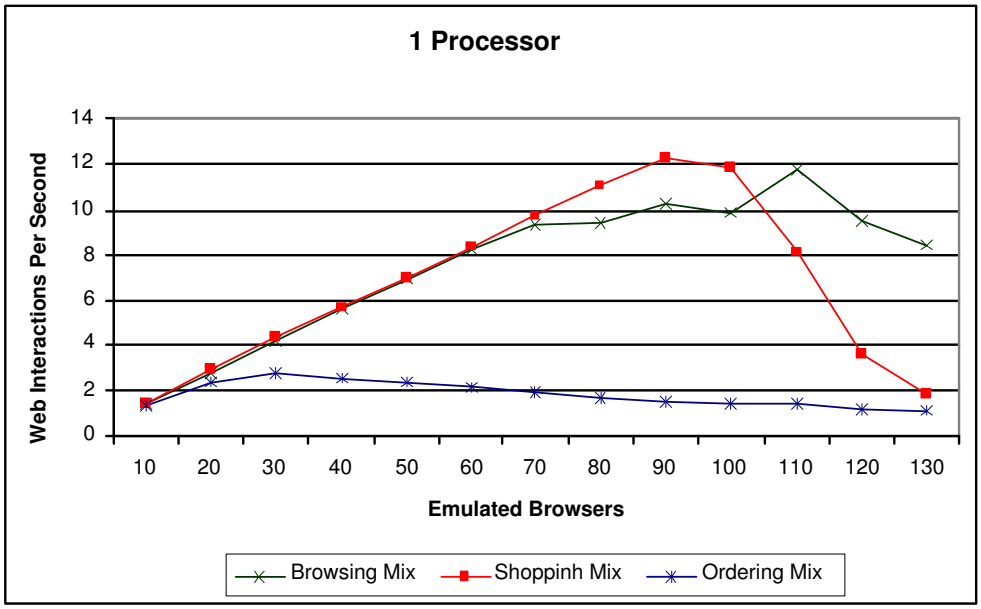

Figure 5: WIPS for 1 Processor

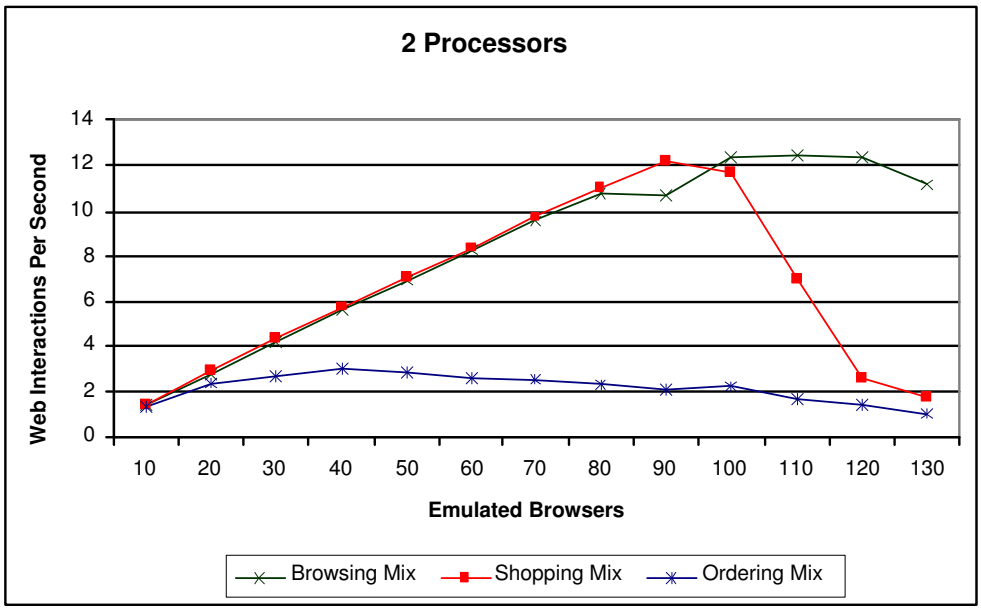

Figure 6: WIPS for 2 Processors

Therefore, it is shown that there is a significant difference between the *shopping*, *browsing* and *ordering* mixes. In fact the *ordering* mix (which leads to the server saturation very much earlier than the other mixes) induces more interaction between the SUT and the PGE for the order transaction authentification. This step requires the use of SSL connections which have a much higher response time when compared with non SSL connections. Being the WIPS the inverse of the response time, the site throughput is much lower when executing the *ordering* mix than when executing the other two mixes. Comparing the *browsing* and the *shopping* mix, the difference is not as substantial. Even though the *shopping* mix involves 15% more ordering interactions than the *browsing* one, the results do not show such a difference in terms of the throughput. Clearly, not only the use of SSL connections influences the throughput results.

5.2 The Processor Effect

The number of processors in a web server is a key factor for e-commerce server scalability. In order to evaluate the processor influence on WIPS, the benchmark has been executed using one and two processors. Figures 7, 8 and 9 show the results obtained for the *browsing*, *ordering* and *shopping* mix respectively when the number of EBs increases from 10 to 130.

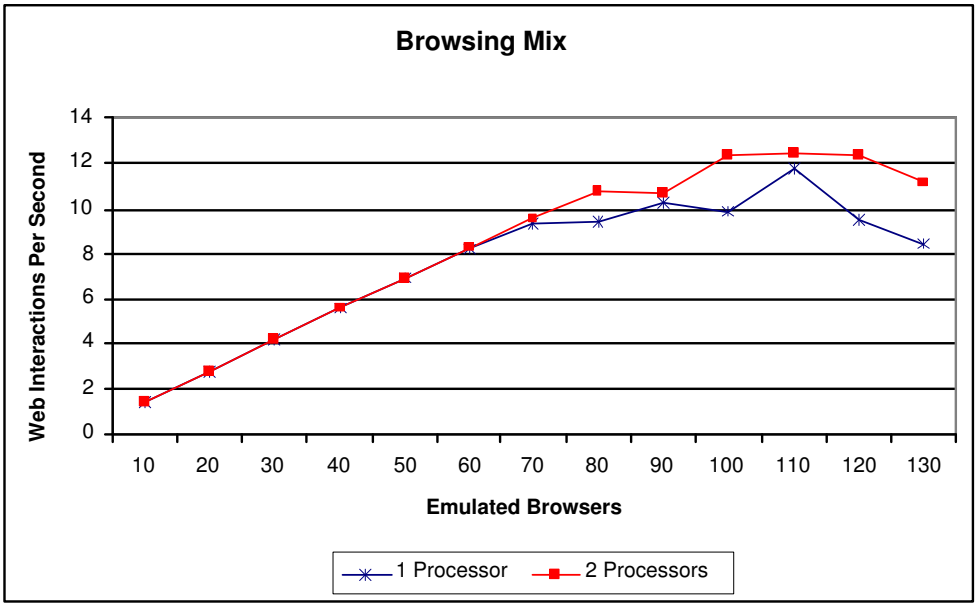

Figure 7: Browsing Mix

The 3 graphs have two well differentiated phases, the initial one, where the WIPS increase linearly as the number of EBs do, in which the throughput is equal to the best one that can be obtained (EB/7). For the *browsing* and *ordering* mix this initial phase is a bit longer in the 2-processors case than in the 1-processor case, though the difference is very subtle. The *shopping* mix results are even more disappointing since the 1-processor and 2-processors results are exactly the same for this initial phase.

Looking at the second phase of the graphs for the browsing and ordering mix, a higher number of WIPS can be observed in the 2-processors case, as a natural follow up of the first phase. On the contrary, the *shopping* mix results for this second phase show even better WIPS for the 1-processor case than for the 2-processor case. This could be due to the overhead introduced when using 2-processors since they need to communicate with each other. Very recently, we also learned that even though Oracle detects automatically the number of CPUs in the system, for Oracle databases to be concurrently and efficiently accessed by more than one processor, they have to be instantiated specifically. We have done some preliminary experiments using this sort of instantiation and also have added some *hints* to the data base queries for a more efficient use of the 2-processors. Unfortunately, the results obtained are not any better than the previous ones. More experiments will be carried out in the near future to confirm these first set of results.

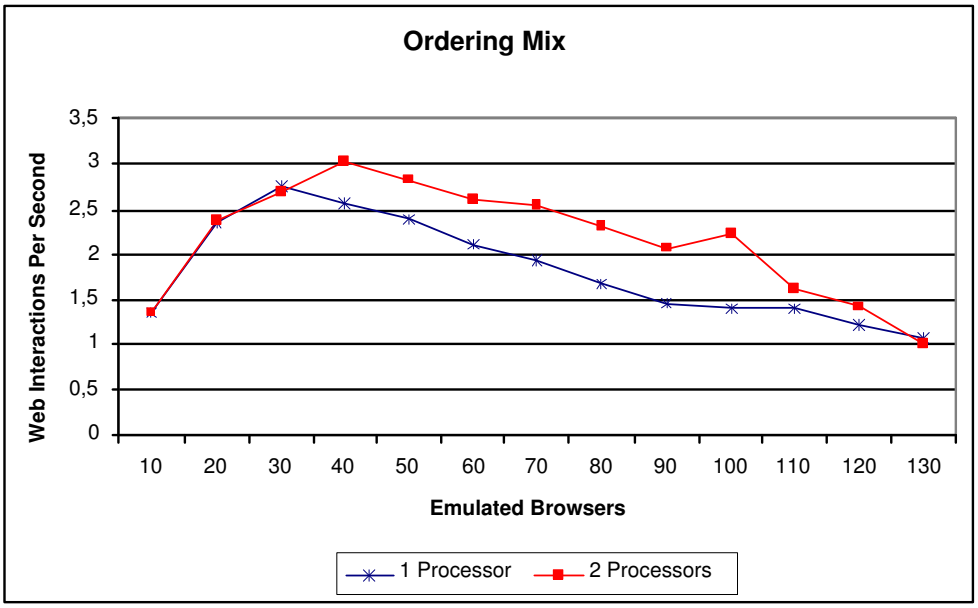

Figure 8: Ordering Mix

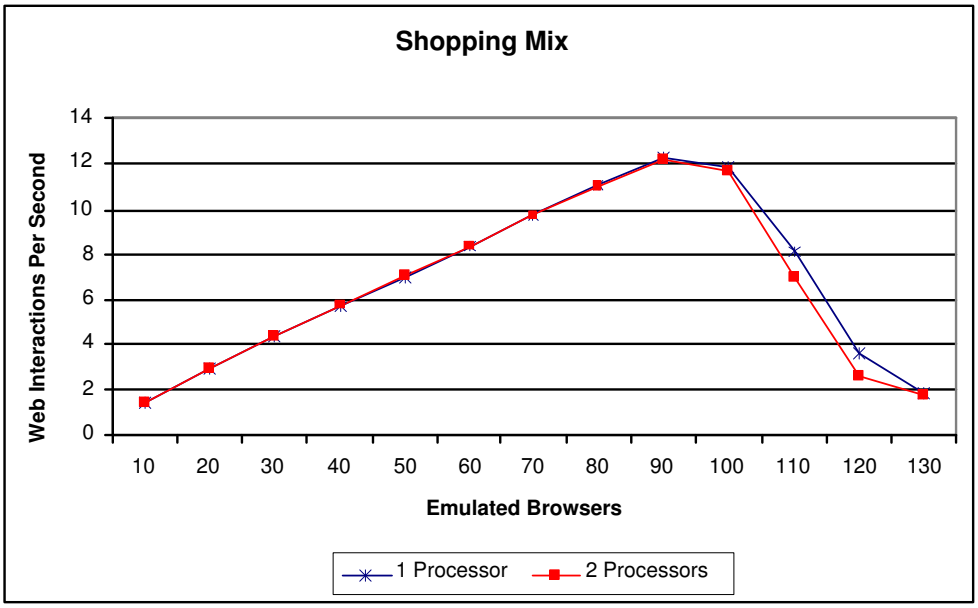

Figure 9: Shopping Mix

6 CONCLUSIONS AND FUTURE WORK

Due to the fast growing of the Internet, web servers are faced to the increasing number of fast, secure and highly available service requirements. Choosing the hardware and software configuration of an e-commerce site is not an easy task to do. Benchmarking techniques can be used to compare possible alternatives and the TPC-W is a benchmark oriented to e-commerce environments. We have implemented this benchmark and carried out a set of experiments in order to test its power to help the evaluation of e-commerce sites configurations. The analysis of the results show that the influence on the throughput results of the number of EBs, their profile and their interaction, is statistically significant. However, influence due to the variation of the number of processors is minimal. It is difficult to asses the scalability of the benchmark, though it seems that the CPU is not the bottleneck of the TPC-W implementation tested. In this sense, the bottleneck needs to be found and the benchmark implementation modified in order to remove this bottleneck. Future work also includes a more detailed study of the scalability, running the benchmark in a multiprocessor machine (with as much as 16 processors). Moreover, this implementation will be also compared with the one presented in [2], running both implementations on exactly the same hardware configuration in order to compare the influence of the software design and implementation on the throughput results.

References

[1] R. Dodge D. A. Menasce and D. Barbara. Testing e-commerce site scalability with tpc-w. *Computer Measurement Group Conference*, December 2-7 2001.

[2] Daniel F. Garcia and Javier Garcia. Tpc-w e-commerce benchmark evaluation. *IEEE Internet Computing*, February 2003.

[3] Daniel. A. Menasce. TPC-W: A Benchmark for E-commerce. *IEEE Internet Computing*, May/June 2002.

[4] Daniel A. Menasce and Virgilio A.F. Almeida. *Scaling for E-Business*. Prentice Hall Inc., 2000.

[5] D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley and Sons, Inc., N.Y, 5th edition, 2001.

[6] The Transaction Performance Processsing Council (TPC). TPC Benchmark W Specification Version 1.8, February 2002.