Ranking Configuration Parameters in Multi-Tiered E-Commerce Sites

Monchai Sopitkamol [1]

Abstract

E-commerce systems are composed of many components with several configurable parameters that, if properly configured, can optimize system performance. Before upgrading existing systems to overcome performance bottlenecks, several areas of a site's architecture and its parameters may be adjusted to improve performance. This paper provides a method to rank key configurable e-commerce system parameters that significantly impact overall system performance, and the performance of the most significant Web function types. We consider both on-line and off-line parameters at each of the e-commerce system layers: Web server, application server, and database server. In order to accomplish our task, we designed a practical, ad-hoc approach that involves conducting experiments on a testbed system setup as a small e-commerce site. The configurable parameters are ranked based on their degrees of performance improvement to the system and to the most critical Web functions. The performance metrics of interest include server's response time, system throughput, and probability of rejecting a customer's request. The experiments were conducted on an e-commerce site compliant to the TPC-W benchmark.

1 Introduction

Typical e-commerce sites are complex and are composed of several connected servers. Each server generally has several configuration parameters. Settings of these parameters can have significant effects on the performance of individual servers and of the whole site. This study addresses the complex problem of designing a practical and ah-hoc methodology for ranking factors by their relevance to performance improvement. The methodology presented here is applied to an e-commerce site compliant with TPC-W as described later in the paper.

Work related to our study can be divided into three categories: Web workload characterization, e-commerce workload characterization, and performance improvement studies of Web and e-commerce sites. WWW workload characteristics have been studied at great depth [1, 2, 3, 6, 7, 15, 16, 21].

Studies by [3, 4, 10] suggested that the distribution of Web file sizes is heavy-tailed. The authors in [10] showed that WWW traffic exhibits self-similar behavior, i.e., WWW traffic is bursty at a wide range of time scales of one second and above.

A thorough examination of non-e-commerce Web traffic was accomplished in [8]. In addition to providing an in-depth analysis of the workload characteristics of Internet traffic based on Web server log analysis, the research produced a workload generator called SURGE (Scalable URL Generator). We incorporate some of the workload results from this work in generating the workload for our experiments.

Menascé et al. defined Customer Behavior Model Graphs (CBMG) as a model to characterize the navigational patterns of users of e-commerce sites [15]. Research in dynamic performance tuning of an e-commerce system can be seen in [17, 18]. In these studies, a performance model was used to dynamically direct configuration changes (system size and number of threads) of Web server and application server of a TPC-W site prototype [19] to maintain Quality of Service (QoS) metrics as desired by a site manager. The workload model used in these studies was a combination of the TPC-W model and the model defined in [8].

This paper is organized as follows. Section two provides some background information. Section three presents the factor ranking methodology. The next section presents the results obtained with the application of the methodology to an experimental testbed. Section five presents some concluding remarks.

2 Background

This section describes some important background information related to e-commerce site architectures, the performance metrics of interest for our study, and the TPC-W benchmark.

2.1 Architecture of e-commerce sites

One of the typical architectures employed by many e-commerce sites is a three-tiered architecture as shown in Fig. 1. This architecture consists of three logical layers of servers running different applications to provide support for various functionalities offered by those sites. These layers include Web server, application server, and database server. A Web server serves as a front end of an e-commerce site, where HTTP [16] requests are received from the clients, processed

[1] Department of Computer Science, George Mason University, Fairfax, VA 22030-4444, USA; monchai@cs.gmu.edu

according to the type of request, or, if necessary, forwarded to the next tier, the application layer, and replied back to the clients.

At the middle tier, application servers implement the business logic by invoking server-side applications. Examples of application server implementations are CGI [16] scripts, FastCGIs [16], server-applications, server-side scripts, ASP, JSP, and EJB. Application servers may interact with the next tier, transaction and/or database servers, if they need transaction processing to be performed, and/or need to access a database.

Finally, transaction and/or database servers take the requests, perform the corresponding transaction processing, and/or database queries or updates, and then send the results back to the application servers, which, in turn, generate HTML pages that are sent back to the client by the Web servers. These three logical layers can be implemented on one machine or can be distributed to different machines, depending upon the workload intensity level and the system capacity.

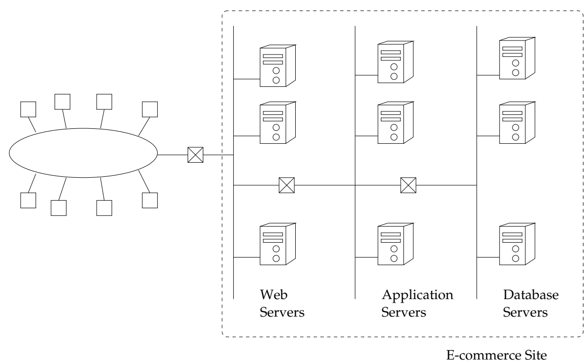

Figure 1: Multi-tiered architecture of an e-commerce site

2.2 Performance metrics of interest

In this paper, we are interested in studying three performance metrics: response time, throughput, and probability of rejection. We consider only the server-side response time, i.e., the time elapsed since a request arrives at the server until the request completes service and leaves the server. The throughput of a server or system is defined as the total number of completed requests divided by the time necessary to process these requests. The probability of rejection of an arriving request at a server is defined as a probability that the request will be rejected when it arrives at a server. This can be operationally computed as the number of arriving requests that have been rejected by a server since the server has started divided by the number of requests submitted to the server.

2.3 The TPC-W benchmark

In 2000, the Transaction Processing Performance Council (TPC) [16, 22] extended its suite of benchmarks to include a transactional Web benchmark for e-commerce systems, the TPC^{TM} Benchmark W (TPC-W). The workload is performed in a controlled Internet commerce environment that simulates the activities of a business-oriented transactional Web server. The workload exercises a breadth of system components associated with such environments, which are characterized by: multiple on-line browser sessions, dynamic page generation with database access and update, consistent Web objects, simultaneous execution of multiple transaction types that span a breadth of complexity, on-line transaction execution modes, databases consisting of many tables with a wide variety of sizes, attributes, and relationships, transaction integrity (ACID properties), and contention on data access and update.

The primary performance metric reported by TPC-W is the number of Web interactions (i.e., full Web pages returned) processed per second (WIPS). Multiple Web interactions are used to simulate the activity of a retail store, and each interaction is subject to a response time constraint.

TPC-W defines three different profiles of Web interactions by varying the ratio of browsing to ordering activities: primarily shopping (WIPS), browsing (WIPSb) and ordering (WIPSo). The browse-to-order ratios of the shopping mix, browsing mix, and ordering mix are 80%:20%, 95%:5%, and 50%:50%, respectively. The primary metrics are the WIPS rate and the associated price per WIPS ($/WIPS). The higher the WIPS rate, the faster the system, and the lower the $/WIPS, the more cost effective the system is. The response time of a Web interaction is called WIRT by TPC-W.

TPC-W emulates the activities of an on-line bookstore. An emulated browser (EB) simulates a client interacting with the on-line site by performing different Web functions available to the browser. The testbed used in the experiments conducted in this work implements an e-commerce site compliant with TPC-W's specification.

A Remote Browser Emulator (RBE) drives the TPC-W workload. It emulates users using Web browsers to request services from the System Under Test (SUT). The RBE creates and manages an Emulated Browser (EB) for each emulated user. The term RBE includes the entire population of EBs. The RBE communicates with the SUT over TCP connections.

The SUT contains all components that are part of the simulated on-line book store. This includes Web servers, application servers, database servers, and network connections between them. According to the TPC-W specification, most of the fourteen Web interactions generate requests that exercise the SUT down to the database layer although most database accesses are query related.

The connection between the EB and SUT is either nonsecure or secure, depending upon the type of Web interaction. Secure connections are implemented through Secure Socket Layer (SSL) protocol version 3 or above, with SSL_RSA_WITH_RC4_128_MD5 as the cipher suite. The digital certificate's private key must be at least 1,024-bit long.

Figure 2 a Customer Behavior Model Graph for the navigational patterns followed during TPC-W sessions. The transition probabilities vary according to the type of TPC-W session.

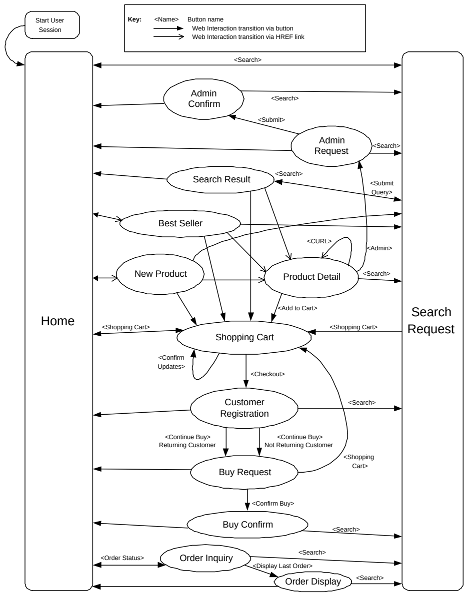

Figure 2: Web Interactions Navigation Options [22]

3 Ranking Methodology

The purpose of the ranking approach is to sort the various factors in decreasing order of impact on the three metrics of interest: response time, throughput, and probability of rejection. Ranking is constructed by comparing the measured minimum, maximum, and the computed range of the performance metrics (i.e., response times throughputs, and probabilities of rejection) due to a factor, against the corresponding target performance metrics values. Since not all Web interactions are probabilistically accessed for the same number of times, this study focuses only on the most frequently accessed five Web interaction types: Search Request (20%), Search Results (17%), Product Detail (17%), Home (16%), and Buy Request (2.6%). The Buy Request interaction, although not the fifth ranked interaction, is considered here because it is the most accessed interaction using Secure Socket Layer (SSL) connections, compared with the other three SSL-related interactions.

TPC-W pre-defined target response times constraints for each Web interaction are shown partly in the second column to the left of Table 1. The 90% WIRT constraints specified by TPC-W stands for the 90-th percentile on response time, i.e., at least 90% of Web interactions of each type must have a WIRT less than the constraint specified (in seconds) for that Web interaction. The 10% WIPS constraints, in Web Interactions Per Second (WIPS), are derived from the 90% WIRT constraints with the help of the Interactive Response Time Law [16]:

$$WIRT = \frac{\text{number of emulated browsers}}{WIPS} - Z$$

where Z is the average think time, specified by TPC-W as 7 seconds. Therefore,

$$WIPS = \frac{\text{number of emulated browsers}}{WIRT + 7}. \qquad (1)$$

Similarly to the throughput, the probability of rejection was not pre-defined by TPC-W. The probability of rejection constraints is derived based on the system availability concept as follows. A rule of thumb used in the on-line industry, is that a fault-tolerant Web-based system should have at least a 99.99% availability [16]. This translates to a probability of rejection constraint of no greater than 0.01% for any Web interaction. Table 1 summarizes all the performance metric constraints for the selected five Web interaction types used during the ranking process.

Table 1: 90% WIRT, 10% WIPS, and Probability of Rejection Constraints for Each Web Interaction Type

Web Interactions	90% WIRT Constraint (seconds)	10% WIPS Constraint	Probability of Rejection Constraint
Buy Request	3	10.00	0.01%
Home	3	10.00	0.01%
Product Detail	3	10.00	0.01%
Search Request	3	10.00	0.01%
Search Results	10	5.88	0.01%

The ranking is also carried out for all the factors considering all fourteen types of Web interactions combined. The response time constraint of eight seconds—a de-facto industry standard [15]—was used here instead of TPC-W's defined constraints. The throughput is obtained as 6.67 WIPS using Equation 1 with WIRT equal to eight seconds and 100 emulated browsers. The probability of rejection constraint is kept at 0.01% as in the previous cases.

Before the ranking is performed, for each factor and its respective Web interaction, we compute or find, from the "best" level of each factor obtained from experimental data, the following:

- the 95% confidence interval of the average value of each performance measure (i.e., response time, throughput or probability of rejection),
- the coefficient of variation of each factor's sample data points,
- the minimum value, maximum value of the performance measures (response time, throughput, or probability of rejection) and its respective range (= maximum - minimum).

For each of the five Web interactions in Table 1, all factors are ranked according to the following steps:

Step 1: Divide the factors into three groups (Group 1, Group 2, and Group 3) with Group 1 ranked higher than Group 2 and Group 2 ranked higher than Group 3, for the response time and probability of rejection metrics. The ranks are, however, reversed for the throughput metric (i.e., Group 3 ranks higher than Group 2 and Group 2 ranks higher than Group 1). Figure 3 illustrates how the factors are grouped and the following section describes the grouping criteria. In the figure, each horizontal line represents a range of the values of the response variables, from the minimum to the maximum. The dark-circle dot inside the horizontal lines with surrounding small vertical lines are the average value of the response variable and its corresponding 95% confidence intervals, respectively.

Group 1: If the lower bound of the 95% confidence interval is larger than the requirement in Table 1 for that Web interaction.

Group 2: If the requirement in Table 1 falls within the 95% confidence interval of the average performance measure, or if the upper bound of the 95% confidence interval is smaller than or equal to the requirement and the upper bound of the range is greater than the threshold.

Group 3: If the upper bound of the range of the performance measure is smaller than the requirement in Table 1

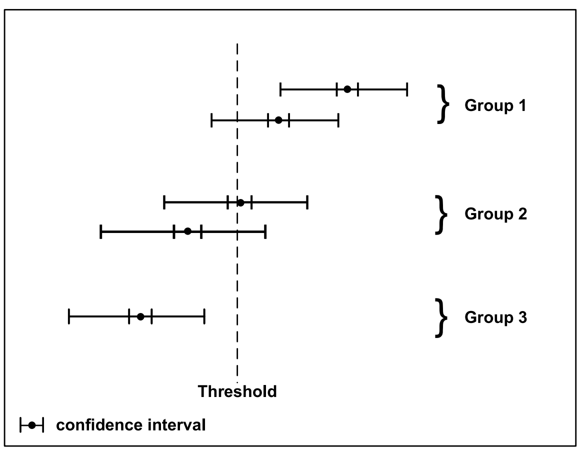

Figure 3: Dividing Factors into Three Groups

Step 2: Within each group, sort the factors as follows:

(a) Factors with larger coefficient of variation are ranked higher than those with smaller coefficient of variation. This is because a factor with a large coefficient of variation exhibits a wide variation of the performance measure compared to its mean value, which results in a larger impact on system performance.

(b) If the coefficients of variation are the same between two or more factors, the one that has a wider range of response time (throughput or probability of rejection) values than the others receives higher score or rank. This is because that factor has a wider effect on system performance.

(c) If two factors have the same range of response time (throughput or probability of rejection) values, the one with the higher maximum response time (higher maximum probability of rejection or lower minimum throughput) is ranked higher. This is because we wish to try to reduce the system's overall response time (increase throughput or decrease the probability of rejection).

(d) If the maximum response times (maximum probability of rejection or minimum throughput) of two or more factors are the same, the one with the smallest value of minimum response time (smallest value of minimum probability of rejection or largest value of maximum throughput) is ranked the highest.

Steps (a)-(d) above are iluustrated by the decision-tree of Fig. 4. L, H, and E indicate lower rank, higher rank, and equal rank, respectively.

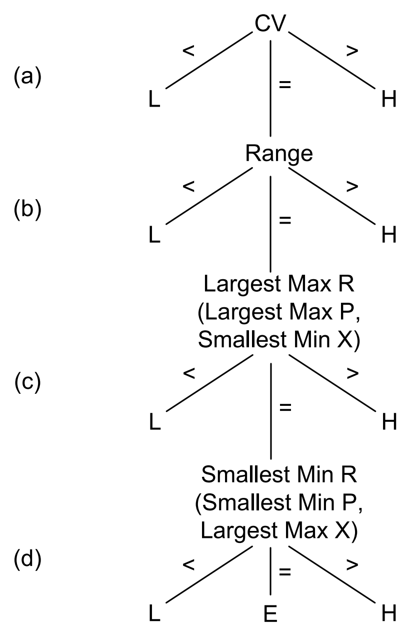

Figure 4: Factor Ranking Decision Tree

4 Applying the Methodology

4.1 Experimental Setup

The testbed used in our experiments is shown in Figure 5. Two workload generator machines were used to guarantee that contention at the workload generator would not limit the system throughput. The system is an e-commerce site built based on the TPC-W specifications [22]. From the figure, the system under test (SUT) consists of a Web server, an application server, and a database server. The workload generator modules emulate client browser sessions that submit requests to the Web server and receive responses back from the Web server after the request has been processed. After a response page is received, the client browser simulates a user "think time" before submitting the next request to the SUT by putting itself to "sleep" for a random interval specified by the TPC-W specifications. The workload intensity to the SUT is varied by changing the number of simultaneous browser sessions. The Web server receives requests from the workload generator, parses them, submits requests, if necessary, to the application server, receives the results back from it, and provides the results to the client running on the workload generator. The application server module takes requests from the Web server, processes the request, and if needed, generates database queries that are submitted to the database server. After query results are received from the database server, the application server dynamically generates a response page, which is sent to the Web server. The database server receives database access requests from the application server, executes the queries and/or updates, and sends back the results to the application server.

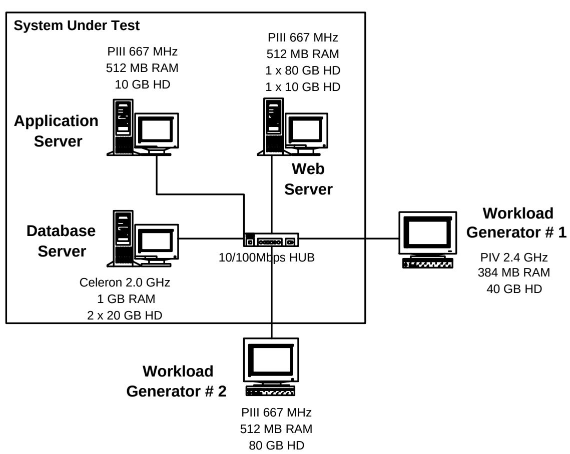

Figure 5: Experimental Testbed Setup

4.2 Workload Models

In order to generate "real world" workloads to the SUT, we used a combination of the TPC-W workload model [22] and the Scalable URL Reference Generator (SURGE) model [8]. Table 2 summarizes the workload model used in our experiments and the following paragraphs describe the terminology in detail.

4.3 Workload Intensity Levels

In order to stress test the testbed, we subjected the SUT to a heavy workload intensity level using 100 emulated browsers (EB). Another factor for workload configuration in the TPC-W specification is the number of items in the database. To stress test the SUT, 100,000 books was chosen as a scaling factor. As a result, the combinations of the two configuration parameters (number of EBs and number of books) that would generate heavy workload level for our experimental environment is 100 EBs with 100,000 items.

Table 2: Workload Model

Category	Distribution
User Think Time	Pareto: $f(x) = ax^{-(a+1)}$, $a = 7/6$, truncated at 70 secs
User Session Minimum Duration	Negative exponential: $f(x) = \mu e^{-\mu x}$, $1/\mu = 15$, truncated at 60 mins
Objects per Page	Varied (under TPC-W's specifications)
HTML Objects Size	Varied (under TPC-W's specifications)
In-Line Object Size	45%=5KB, 35%=10KB, 15%=50KB, 4%=100KB, 1%=250KB

4.4 Initial Database Population Configurations

The number of rows for each database table is specified by TPC-W as a function of the number of emulated browsers (EB) and number of items. Table 3 shows the number of rows for each database table for the system configuration with 100 EBs and 100,000 books.

Table 3: Cardinality of the Database Tables

Table Name	# of Rows
Customer	288,000
Country	92
Address	576,000
Orders	259,200
Item	100,000
Order_Line	777,600
Author	25,000
CC_XACTS	259,200

4.5 Workload Characterization

The TPC-W specification defines fourteen different types of Web interactions based on the navigational options that a typical e-commerce site offers to its customers: home, shopping cart, customer registration, buy request, buy confirm, order inquiry, order display, search request, search result, new products, best sellers, product detail, admin request, and admin confirm.

4.6 Configuration Parameters

Twenty-eight factors from the three-tiered architecture are chosen for the experiments. The following list briefly describes the factors, organized by the layer in which they reside and sorted in an alphabetical order. The Web server layer uses Microsoft Internet Information Server (IIS) 5.0, the application server layer uses Apache Tomcat 4.1, and the database layer uses Microsoft SQL Server 7.0. Factors 1-13 are the Web server factors, 14-16 the application server factors, and 17-28 the database server factors.

1. Application Optimization: Whether to allow perfor-

mance optimization of only the foreground applications (more processor resources are given to the foreground program than to the background program), or all applications (all programs receive equal amounts of processor resources) [9].

2. Application Protection Level: Whether applications are run in the same process as Web services (low), in an isolated pooled process in which other applications are also run (medium), or in an isolated process separate from other processes (high) [20].

3. Connection Timeout: Sets the length of time in seconds before the server disconnects an inactive user [13].

4. HTTP KeepAlive: Whether to allow a client to maintain an open connection with the Web server [13].

5. ListenBacklog: Set the maximum active connections held in the IIS queue [11].

6. Logging Location: Sets a specific disk and path where the log files are to be saved [20].

7. MaxCachedFileSize: Sets the size of the largest file that IIS will cache [20].

8. MaxPoolThreads: Sets the number of I/O worker threads to create per processor [11].

9. MemCacheSize: Sets the size of the virtual memory that IIS uses to cache static files [20].

10. Number of Connections: Sets the maximum number of simultaneous connections to the site [13].

11. Performance Tuning Level: Sets the performance optimization level of IIS to the expected total number of accesses to the Web site per day [13].

12. Resource Indexing: Whether to allow Microsoft Indexing Service to index a specific Web directory and files in that directory [20].

13. worker.ajp13.cachesize: This is not an IIS parameter but it is a configurable parameter of the Web server. It specifies the maximum number of sockets that can be opened between two Tomcat out-of-process processes [5].

14. acceptCount: Sets the maximum queue length for incoming connection requests when all possible request processing threads are in use [5].

15. minProcessors: Specifies the number of request processing threads that are created when a Tomcat connector is first started [5].

16. maxProcessors: Specifies the maximum number of request processing threads to be created by a Tomcat connector, which determines the maximum number of simultaneous requests that can be handled [5].

17. Cursor Threshold: Tells SQL Server whether to execute all cursors synchronously, or asynchronously [14].

18. Fill Factor: Sets the default fill factor for indexes when they are built [14].

19. Locks: Sets the amount of memory reserved for database locks [23].

20. Max Server Memory: Sets the maximum amount of memory, in MB, that can be allocated by SQL Server to the memory pool [23].

21. Max Worker Threads: Determines how many worker threads are made available to the SQL Server process from the operating system [14].

22. Min Memory Per Query: Sets the amount of physical memory in KB that SQL Server allocates to a query [14].

23. Min Server Memory: Sets the minimum, in MB, to be allocated to the SQL Server memory pool [23].

24. Network Packet Size: Sets the packet size that SQL Server uses to communicate to its clients over a network [14].

25. Priority Boost: Whether to allow SQL Server to take on higher priority than other application processes in terms of receiving CPU cycles [14].

26. Recovery Interval: Defines the maximum time, in minutes, that it will take SQL Server to recover in the event of a failure [23].

27. Set Working Set Size: Specifies that the memory that SQL Server has allocated cannot be paged out for another application's use [23].

28. User Connections: Defines the maximum number of concurrent user connections allowed to SQL Server [23].

4.7 Experimental Results

For each Web interaction type, the rankings can be used to indicate which factors improve most significantly the performance of that interaction type for each performance metric of interest. Tables 4, 6, 7, 8, and 9 present ranking results for all 28 factors by Web interaction type across all three performance metrics (response time, throughput, and probability of rejection). The factor ranked number one is indicated in bold in each of these tables. For example, Table 4 shows that the factor that has the highest impact on the response time of a buy request is Cursor Threshold. This factor also has the highest impact on the probability of rejection. However, for the throughput, the factor that has the highest impact is Number of Connections.

Table 4: Factor Ranking for Buy Request Web Interaction

Buy Request	Rank		
Factor	WIRT	WIPS	Prej
Fill Factor	9	5	9
Set Working Set Size	11	25	11
Application Protection Level	17	14	17
Network Packet Size	4	15	4
Locks	17	9	17
MaxCachedFileSize	12	7	12
Http KeepAlive	18	4	18
minProcessors	19	3	19
Application Optimization	6	26	6
Resource Indexing	17	11	17
Performance Tuning Level	13	24	13
ListenBacklog	17	22	17
Number of Connections	16	**1**	16
Recovery Interval	2	6	2
Connection Timeout	8	2	8
Logging Location	17	21	17
MaxPoolThreads	15	18	15
worker.ajp13.cachesize	3	16	3
maxProcessors	20	10	20
Cursor Threshold	**1**	19	**1**
User Connections	5	13	5
Max Worker Threads	14	17	14
Priority Boost	22	12	22
Min Memory Per Query	21	23	21
MemCacheSize	7	8	7
acceptCount	10	20	10

5 Concluding Remarks

This section summarizes the major findings of this study. Table 5 shows the highest ranking factors for the five most important Web interaction types and for each of the three performance metrics (response time, throughput, and probability of rejection). The table shows that Set Working Set Size is the highest ranked factor for Search Results for the three performance metrics. This indicates that under heavy load, a proper setting of the amount of main memory available to the SQL server is crucial for good performance.

It can also be seen from the table that Cursor Threshold is the best factor to tune to improve the response time of Buy Request and Search Request Web interactions. This table also shows that eight factors, namely Cursor Threshold, Number of Connections, Set Working Set Size, HTTP Keep Alive, MaxCachedFileSize, Connection Timeout, worker.ajp13.cachesize, and MaxPoolThreads, are the ones to be tuned to improve the response time, throughput, and probability of rejection of the five most important Web interactions.

Table 5: Best Ranking Factors by Web Interaction Type for 100 EBs/100K Items for Each Performance Metric

WIRT	Best-Ranking Factors
Buy Request	Cursor Threshold
Home	Number of Connections
Product Detail	Number of Connections
Search Request	Cursor Threshold
Search Results	Set Working Set Size
WIPS	Best-Ranking Factors
Buy Request	Number of Connections
Home	HTTP KeepAlive
Product Detail	Set Working Set Size
Search Request	MaxCachedFileSize
Search Results	Set Working Set Size
Prej	Best-Ranking Factors
Buy Request	Cursor Threshold
Home	Connection Timeout
Product Detail	worker.ajp13.cachesize
Search Request	MaxPoolThreads
Search Results	Set Working Set Size

References

[1] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira. “Characterizing Reference Locality in the WWW,” In IEEE Conference on Parallel and Distributed Information Systems, Miami Beach, Florida , Dec. 1996.

[2] M. F. Arlitt and C. L. Williamson. “Web Server Workload Characterization: The Search for Invariants. ” In Proc. of the ACM SIGMETRICS ’96 Conference, Philadelphia, PA, Apr. 1996.

[3] M. Arlitt et al, “Workload Characterization of a Web Proxy in a Cable Modem Environment,” ACM Performance Evaluation Review, 27 (2), Aug. 1999, pp. 25-36.

[4] M. Arlitt and C. Williamson, “Internet Web Servers: Workload Characterization and Performance Implications,” IEEE/ACM Tr. Networking, Vol. 5, No. 5, pp. 631-645, Oct. 1997.

[5] A. Bakore et al., “Professional Apache Tomcat,” Wrox Press Inc, 2002.

[6] G. Banga and P. Druschel, “Measuring the capacity of a Web server under realistic loads,” World Wide Web, 2(1-2):69-83, Jan. 1999.

[7] P. Barford, M. Crovella, “Measuring Web Performance in the Wide Area,” Special Issue on Network Traffic Measurement and Workload Characterization, Aug. 1999.

[8] P. Barford and M. Crovella. “Generating Representative Web Workloads for Network and Server Performance Evaluation.” In Proc. of the ACM SIGMETRICS Conference, Madison, WI, Jul. 1998.

[9] M. Baute, “White Paper: Multi-Tier Performance Tuning,” http://www.gasullivan.com/whitepapers/PerformanceTuning A.pdf.

[10] M. Crovella and A. Bestavros, ”Self-Similarity in World- Wide Web Traffic: Evidence and Possible Causes,” IEEE/ACM Tr. Networking, Vol. 5, No. 6, pp. 835-846, Dec. 1997.

[11] B. Curry et al, ”The Art and Science of Web Server Tuning with Internet Information Services 5.0,” http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/iis/iis5/maintain/optimize/iis5tune.asp

[12] R. Dodge, “Improving E-Commerce System Performance with Dynamic System Tuning, ” Ph.D. Dissertation, George Mason University, May 2001.

[13] “IIS 5.0 Documentation,” Microsoft Windows 2000 Help.

[14] B. McGehee, “SQL Server Configuration Performance Checklist,” http://sql-server-performance.com/sql_server_performance_audit5.asp

[15] D. A. Menascé and V. A. F. Almeida,“Scaling for E-Business: technologies, models, performance, and capacity planning,” Prentice Hall, Upper Saddle River, NJ, 2000.

[16] D. A. Menascé and V. A. F. Almeida,“Capacity Planning for Web Services: Metrics, Models, and Methods,” Prentice Hall, Upper Saddle River, NJ, 2002.

[17] D. A. Menascé, D. Barbara and R. Dodge, “Preserving QoS of Ecommerce Sites Through Self Tuning: A Performance Model Approach,” Proc. 2001 ACM Conference on E-commerce, Tampa, FL, Oct. 14-17, 2001.

[18] D. A. Menascé, “Using Performance Models to Dynamically Control E-Business Performance,” Proc. 11th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems, Aachen, Germany, Sep. 11-14, 2001.

[19] D. A. Menascé, “TPC-W: a benchmark for E-commerce,” IEEE Internet Computing, May/June 2002.

[20] Microsoft, “Windows 2000 Performance Tuning, White Paper,” Mar. 22, 2000, http://www.microsoft.com/windows2000/professional/evaluation/performance/reports/perftune.asp

[21] J. E. Pitkow,“Summary of WWW Characterizations,” Proc. World Wide Web Conf. 2(1), Jan. 1999, pp. 313.

[22] Transaction Processing Performance Council, http://www.tpc.org.

[23] E. Whalen et al, ”SQL Server 2000 Performance Tuning: Technical Reference,” Microsoft Press, 2001.

Table 6: Factor Ranking for Home Web Interaction

Home	Rank		
Factor	WIRT	WIPS	Prej
Application Optimization	22	19	12
MaxCachedFileSize	3	9	3
minProcessors	21	17	4
Priority Boost	22	19	10
worker.ajp13.cachesize	16	3	6
Application Protection Level	17	15	11
Http KeepAlive	7	**1**	15
Logging Location	22	10	12
Min Memory Per Query	8	7	16
Network Packet Size	5	2	12
Set Working Set Size	9	11	17
Locks	2	18	8
Performance Tuning Level	22	19	18
User Connections	13	14	12
MaxPoolThreads	4	6	14
acceptCount	14	4	9
Recovery Interval	12	22	2
MemCacheSize	6	21	7
Number of Connections	**1**	13	13
Connection Timeout	15	16	**1**
ListenBacklog	20	8	12
Max Worker Threads	18	19	12
Resource Indexing	22	19	12
Cursor Threshold	11	12	12
maxProcessors	10	5	12
Fill Factor	19	20	5

Table 7: Factor Ranking for Product Detail Web Interaction

Product Detail	Rank		
Factor	WIRT	WIPS	Prej
acceptCount	17	20	5
Priority Boost	18	3	2
Application Optimization	18	3	5
Logging Location	15	10	5
Min Memory Per Query	21	6	11
Network Packet Size	22	5	12
Set Working Set Size	6	**1**	7
Connection Timeout	24	18	13
MaxPoolThreads	4	22	4
MemCacheSize	13	21	5
Locks	3	11	5
Http KeepAlive	18	3	10
Recovery Interval	16	9	5
MaxCachedFileSize	8	14	8
minProcessors	10	19	5
Performance Tuning Level	5	23	6
Number of Connections	**1**	13	5
Cursor Threshold	9	17	5
Max Worker Threads	11	16	5
worker.ajp13.cachesize	20	12	**1**
ListenBacklog	14	8	5
maxProcessors	23	2	3
Application Protection Level	19	15	5
Resource Indexing	7	3	5
Fill Factor	2	4	5
User Connections	12	7	9

Table 8: Factor Ranking for Search Request Web Interaction

Search Request	Rank		
Factor	WIRT	WIPS	Prej
Network Packet Size	5	6	3
Application Optimization	10	10	5
ListenBacklog	8	20	15
worker.ajp13.cachesize	9	18	16
Cursor Threshold	**1**	11	10
MemCacheSize	2	14	11
Priority Boost	10	10	4
MaxCachedFileSize	10	**1**	17
Application Protection Level	4	15	12
Http KeepAlive	10	10	17
Logging Location	10	9	9
Min Memory Per Query	7	13	14
Performance Tuning Level	10	10	17
Set Working Set Size	10	5	17
minProcessors	6	22	13
Recovery Interval	11	2	18
Locks	3	7	7
MaxPoolThreads	10	16	**1**
acceptCount	14	3	21
Number of Connections	10	17	8
maxProcessors	10	12	17
Max Worker Threads	10	21	2
Connection Timeout	12	8	19
Resource Indexing	10	10	6
Fill Factor	15	19	22
User Connections	13	4	20

Table 9: Factor Ranking for Search Results Web Interaction

Search Results	Rank		
Factor	WIRT	WIPS	Prej
worker.ajp13.cachesize	15	17	15
Application Optimization	20	9	20
Connection Timeout	10	19	10
MaxCachedFileSize	9	6	9
ListenBacklog	17	10	17
maxProcessors	11	12	11
Cursor Threshold	5	15	5
Application Protection Level	2	21	2
Http KeepAlive	20	9	20
Logging Location	20	8	20
Min Memory Per Query	18	2	18
Network Packet Size	14	3	14
Performance Tuning Level	20	9	20
Set Working Set Size	**1**	**1**	**1**
Max Worker Threads	8	9	8
minProcessors	12	16	12
MaxPoolThreads	3	11	3
MemCacheSize	4	14	4
acceptCount	16	4	16
Number of Connections	7	13	7
Priority Boost	20	9	20
Recovery Interval	13	7	13
Locks	6	20	6
Resource Indexing	20	9	20
Fill Factor	21	18	21
User Connections	19	5	19