Performance Issues of a Web Database

Yi Li, Kevin Lü

School of Computing, Information Systems and Mathematics
South Bank University
103 Borough Road, London SE1 0AA
{liy, lukj}@sbu.ac.uk

Abstract. Web databases are becoming an efficient tool used to manage the web sites. In this paper we analyse the performance of a typical Web database system with different sizes of web pages and different sizes of database tables. Since a web server and a database server work simultaneously, the response time in dealing with a request to the database can not be seen simply as the web server service time plus database service time. The performance metrics and optimisation suggestions are made on the basis of the analysis of the relationship between them. Initial experiments are designed to investigate how a Web database system works and what affects its performance, in particular, the response time. We explored the different ways of sending query result files. An analysis of the initial test results and suggestions on improving the Web database system performance are presented.

1 Introduction

The World Wide Web (WWW) has developed into the largest information database and, potentially, the most convenient medium for businesses. Web sites used for conducting business transactions have, in fact, become one of the most promising factors that determine the success or failure of an enterprise [1]. Good performance of Web databases provides a company with a definite edge over competition while poor performance makes it seriously handicapped.

Hence to ensure good performance of Web databases is absolutely essential for business institutions as well as for any type of enterprises. A Web database can be seen as an integrated system of a web server and database servers. If data files are stored in web databases, then queries to the databases normally take longer time than requests to static Web pages or Web site script pages. Therefore they directly determine the Web server performance [2]. It is very important to understand how Web databases work and to make them work with maximum efficiency.

The objectives of this study are: to investigate through experiments the way a Web database system works; to find performance matrices to accurately describe the Web database system; to find ways of tuning the system to get better performance; and to estimate the performance of the system. In this paper we present initial experiments on a Web database system performance and an analysis of the test results. Relationships between query result file size, paged result size, database base table size, network throttle, and the response time are examined.

The remainder of the paper is organised as follows: Section 2, discussion of the related work. Section 3, characteristics of the web database system to be investigated. Section 4, experiments and results. Section 5, analysis of the results of the experiments and the web database model, and Section 6, conclusions and suggestions for further research.

2 Related Work

The database research community has an immense interest in the management of Web information. A series of conferences on Web databases (e.g. International Workshop on the Web and Databases) have been held since 1998 [3]. The demand for extending the functions of databases to the Web has presented new challenges to database researchers and developers as can be seen in some proposals relating the database area for data management on the Web sites [2].

Most recent efforts of Web database developers are directed towards exploring the specifications of the structure and content of Web sites on how to get useful information quickly from Web databases. Efforts have been made on modelling the Web as an infinite, semi-structured set of objects and providing a set of language for managing and restructuring data coming from the Web [13].

In relation to measurement of the capacity of Web sites, several benchmarks have been advanced by different institutions. The first published of these was the SGI Webstone benchmark [4]. SPECweb99, a standardised benchmark, was developed by the Standard Performance Evaluation Corporation (SPEC) to measure the maximum number of simultaneous connections. It asks for a predefined benchmark workload that a Web server is able to support while still meeting specific throughput and error rate requirements [5]. TPC-W is a new Web application benchmark recently launched by the Transaction Processing Performance Council (TPC) to measure the performance and price performance of computer systems used for transactional Web environment [6].

There are no suitable performance metrics and benchmarks for measuring the Web database system yet. Current performance studies either focus on the Web server or on the database server. We think there is a need to conduct a performance study taking the Web database as an integrated system.

3 Architecture of a Web Database System

A Web database is an integrated system of Web severs and database servers, which enables users to access on-line information in a platform-independent manner through Web browsers.

Web servers and database servers work together in a Web database as an integrated system. Typical Web databases are three-tier or N-tier [7]. The users run standard browsers on the client side and the requests are transferred over the Internet to Web database systems. A Web server can create a number of threads and each thread serves a request from a client.

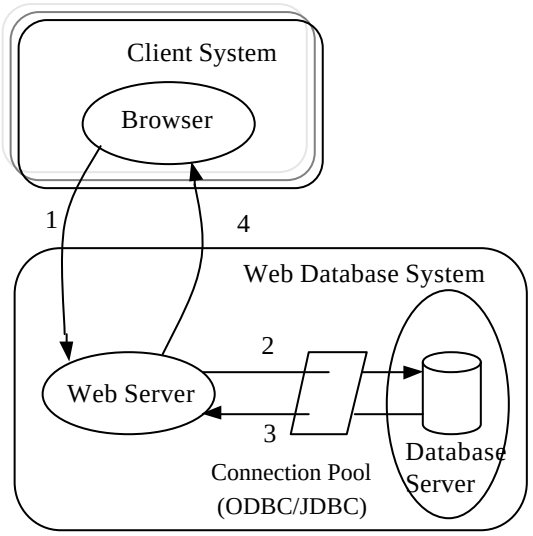

Fig. 1. Processing a Web database query at a typical Web database system

Figure 1 shows a typical Web database processing procedure. The following outline the details of a query processing in a Web database system.

Step 1. Query requests are sent from clients to a Web database system via Internet.

Step 2. Each request is served by a thread called initialisation. Then the Web server deals with the request. When the Web server finds a database query, it will dispatch the query request to the database server through an interface (the connection pool in Figure 1).

Step 3. A query result made by the database server is given back to the Web server through the same interface.

Step 4. The Web Server incorporates the query result into a response Web page, and then sends the page back to the client.

The Web server and the database server in a Web database system may work concurrently. The two servers can run either on the same computer or on two different computers. From the clients' point of view, they are one integrated system, which acts as a Web database. Web servers play the role of a service provider and data consumer, responsible for interacting with clients and requesting data from a database on their behalf. The database server acts as a data provider in charge of data storage and data manipulation.

Pooling technology is used in most Web database systems to get better performance. There are usually two different types of pools. One is a thread pool in the Web server and the other is a connection pool to the database server.

Thread pools eliminate the overheat of creating a new thread when an additional request reaches the server. Instead of creating a new thread for each request, the server creates a pool of threads when it starts up. If more CPU processing power is available, then by increasing the number of threads in the pool the server capacity will be improved because an available CPU processing power can be used to deal with more requests concurrently. But too many threads will consume a lot of resources and will weaken the server performance.

The same mechanism is applied in the connection pool. At the early stage of the Web database development, Common Gateway Interface (CGI) was the only way used to connect a Web server to a database. The CGI architecture, which gave poor performance, was gradually replaced by the Application Programming Interface (API) approach [8], in which an interface between the server and back-end applications is created by using a dynamic linking or shared object. The ODBC pool is a typical connection pool. Vendors of database systems integrate databases with the Web server through their own API to act as an extension to the Web server that can communicate with databases through ODBC [9].

4 Experiments

A series of tests are designed to examine how a Web server and database servers work together and what may affect the response time of a Web database system. The following relationships are examined namely, relationships between: 1) query result file sizes and response time; 2) table sizes of a database and response time; 3) the network throttle and response time; and 4) paged query result file sizes and response time.

4.1 Experimental Design

The tests we carried out simulate the activities of a Web database system and clients' Web browsers. In our experiments, Windows NT is used as the platform with Microsoft Internet Information Server as the Web server, MS SQL Server as the database server, and the Active Server Pages as the Web site script language. A thread pool initialises new requests and an ODBC pool is connected to the database server.

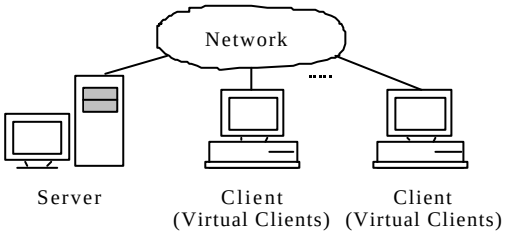

Fig. 2. Components for the tests

The tests involve three primary components: a Server, Clients, and a Network. During the tests, a client runs a virtual client program and the server responds with a result file. Figure 2 shows these components of our tests.

A Web server and a database server are running on the same Server computer. The Server uses a set of prepared sample ASP pages and sample databases, which simulate those that a server might provide for its clients. The prepared ASP pages and databases simulate Web databases of various sizes.

A client application is running on a computer to simulate one/many client's browser(s) and virtual clients. The client application can be run on more than one client computer. Each virtual client makes one connection and a page request to the server at a time. This design enables each client computer to simulate more than one client.

In these tests, the network refers to the communication links between the client computers, and the server computer. It is a 10Mbps Ethernet network connected by a hub.

4.2 Performance Metrics

Traditionally, response time and throughput are the two most important performance metrics for both Web information systems and database systems [11][12]. But the Web database has its unique character that is different from any other database applications. Suppose the result size of a Web database query is 1MB, a client may only need to receive the first page to read at first, expecting pages to follow will be available by the time when he finishes reading the first page. The client does not mind whether the whole result file arrives at the same time or in sequence. If the client can receive the sub-sequential pages just before completing the page in hand, he will feel being served effectively, and would be satisfied with the service. Meanwhile, clients do not want the last page to arrive too late. Therefore, we adopt the following two metrics as our response time metric:

TTFB - time for a client to receive the first response byte from the server.

TTLB - time for a client to receive the last response byte from the server.

In addition to the above two kinds of response time, the following new factors need to be considered to get a clearer description of the performance of a Web database system.

Result Arrival Rate (RAR) is used to present the time intervals between the two consecutive pages. RAR is measured in terms of bits per second (bps) or pages per second (pps). A reasonable RAR can improve the system resource utilisation without degrading the service quality provided for users.

Throughput Ratio (TR) indicates how the capacity of the Web server matches the capacity of the database server in a Web database system. The throughput of a Web server and the throughput of a database server both refer to the amount of data they can process in a certain time. If their throughput ratio is nearly equal to 1, i.e., their processing speeds match each other, and the Web database system can achieve its maximum throughput.

4.3 Results

A set of tables (10,000, 20,000 and 60,000 records) were created for the tests, and the length of a record ranged from 86 bytes to 256 bytes. So, there were totally 6 tables used in our tests. Queries to the database tables were adopted according to the standard SQL.

In order to control cache effect on the performance, tests carried out were of two types. One is called *single query* tests, in which in order to reduce the cache effect to a minimum we tested the queries one at a time once the Server was started up. The other is called *mixed* tests, in which we tested a request in a simulated environment.

Response time vs. query result file size. A number of queries were tested to get the response time of different result file sizes. Figure 3 shows the Web database system response time versus the result file size from 10k to 100k in *single* query tests.

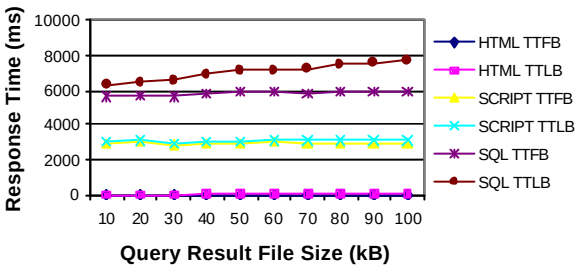

Fig. 3. Web database system response time vs. query result file size (*10k-100k single query tests*)

Our tests show that when the result file size increases the response time for queries to the database increases about 7.8 times more than the script response time, and the rate of increase of the script response time is about 1.5 times bigger than that of HTML as shown in figure 3. Therefore, users may ignore the small increases in response time of the static HTML when its result file size increases, but they can not ignore the increase in the response time of a Web database system when its result file size increases.

Based on our experiments, we found that the response time of HTML request is the shortest and the response time of database request is the longest among the three classified requests when the query result file sizes are the same. Figure 3 shows two other features: 1) Linear Relationship between Response Time and Query Result File Size. 2) The increase rate of database query response time is much higher than that of other requests in relation to the increase of the query result file size.

Our other test results confirm that the linear relationship that exists before the result file size is increased up to 1000kB under a *mixed* non-queueing environment.

The response time of a script request may be close to an HTML request because the script is running inside a Web server and can use API directly to improve the performance. The response time of a database query is much longer than that of an HTML request, because not only the query running in the database needs a relatively longer time, setting up a connection to the database and the communication of the query result data between the Web server and the database server also need time. When the response file size increases, the Web server time increases and so does the database server time. Further more, the communication time increases too. Therefore, the increase rate of database query response time is much higher than that of other requests in relation to response file size increases.

Response time vs. database table size. To see how much the database size affects the response time, we changed the database size from 10,000 records to 650,000 records in the tests. Figure 4 shows the testing result with the same requests to a different size database. A large database has a bit longer response time than a small one but the difference is small. The response time increase rate of a large database with the result file size is almost the same as that of a small database.

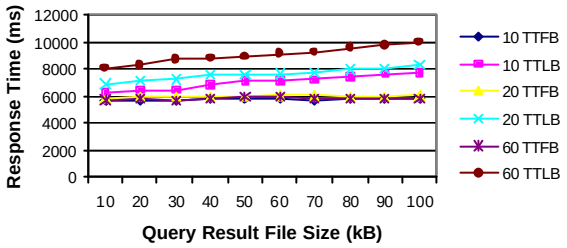

Fig. 4. Web database system response time vs. query result file size (*10k–100k 10k, 20k&60k records single query tests*)

Changing database size means changing database service time. The linear relationship between response time and query result file size remains. Figure 4 shows that the linear relationship is unchanged and the values are not increased significantly either. This is because a database server is efficient in querying data and the transferring time of query result file between the database server and the Web server is almost unchanged because of the same size of query result file. Therefore, the increase in database service time by data added to the database is small. If a query is very expensive to operate, an obvious increase in Web database system response time will occur when the size of a database increases.

Response time vs. network throttle. The Web server may work in two modes. In Mode 1 the data from database are processed and sent to clients one by one, on a first come first served basis. In Mode 2, only when all the data arrive are they processed in the Web server and then, sent to clients altogether. All the above tests were performed by means of Mode 1. There produced a slightly different result when we changed to using Mode 2, and the network transfer rate was limited to T1, which is 1.544Mbps to connect to the network.

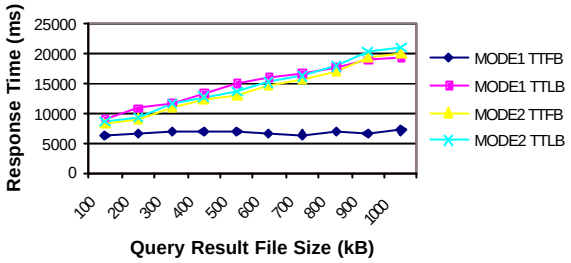

Fig. 5. Web database system response time vs. query result file size (*100k - 1000k Mode 1&2*)

Figure 5 shows that Mode 2 response time is normally shorter than Mode 1's. When the response file size is increased to a certain value, Mode 2 response time is longer than Mode 1 and the bandwidth of connection to the network becomes the bottleneck device.

The reason that Mode 2 line exceeds Mode 1 line is that when the result data get to a certain value, the data can not be sent out to the network immediately and queueing occurs. We do not recommend the use of Mode 2 when the response file size is big, because the queueing time will be counted.

Mode 1 is not the best way to gain good performance either, because its TTLB is longer than Mode 2 normally. It is better to send out the response data whenever they accumulate to a certain amount. Thus, it can control the queueing time by insuring that the server send the data out immediately, and it can also shorten TTLB by preventing the server from switching operations too frequently.

Response time vs. query result file paged size. When a result file is over 3 or 4 pages, it is better to divide it into several small pages (normally 1 or 2 pages to display) and send them to the client separately. We tested the response time of different size pages.

We defined 256 bytes for each record in a result page. The 62.5kB file was divided into pages of different records ranging from 8 to 15 records. Figure 6 shows the results.

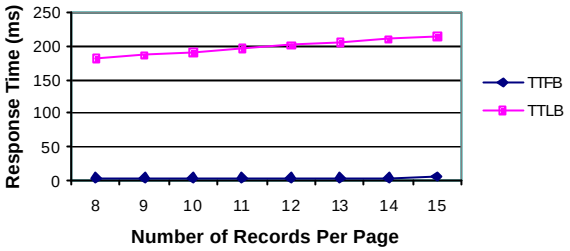

Fig. 6. Web database system response time vs. different numbers of records per page (*8-15 records mixed tests*)

If a result page with 8 records is changed to one with 15 records, the response time will increase about 18% according to Figure 8. We assume that clients do not care much about whether it is 8 or 15 records on the first result page they read. Then, using 8 rather than 15 records page the result file will have much better performance for the system (over 15% response time reduced).

From the above tests, we have come to the following findings.

1. The test results show that the time of response of a Web database is far more sensitive to the response file size. That is, if the response file size increases, then the response time for the Web database query will increase much more than the response time of the static Web pages or the Web site script pages.
2. Clearly, the response file size has immediate effects on the maximum permitting query rate [10]. Therefore, the rate of queries to the Web database is the main factor that determines the capacity of the whole system.
3. The response file size being so important to the performance of the system, it should be divided into separate pages to be sent to the clients separately and make the size in each page small within the limit of satisfactory service to the clients.
4. Because the time between TTFB and TTLB of database queries is much longer than that of static pages, metric RAR can be used to show clearly the service quality of the system. And if we know the system's TA, we can adjust the ratio of HTML, script and database pages according to workload on the system in order to get the best match between the Web server and the database server.

Conclusions

Several attributes related to the performance of Web databases are investigated from a perspective of taking Web servers and database servers as an integrated system. The importance of performance study of Web database systems cannot be over emphasised because queries to the database system are dealt with over much longer response time compared with queries to the script or HTML files.

New metrics for Web database systems are suggested to evaluate the system. They are: time for the first response byte from the server (TTFB), time for the last response byte from the server (TTLB), Result Arrival Rate (RAR), and Throughput Ratio (TR). These metrics are essential to accurately describe and evaluate a Web database system.

Our tests confirm that the way to send query results back to clients is essential to good performance of the system. To gain better performance the technology of paged query result file should be used for queries with a very large result file. The result page size is a key factor contributing to better performance.

We are currently working on experiments, which multiple users access to a Web database system at the same time with different arrival rate and different types of queries. The study aims to find a way to predicate and tune the performance of a Web database.

References

1. Daniela Florescu, Alon L. and Dan S.: Optimization of Run-time Management of Data Intensive Web Sites. Proceedings of the 25th VLDB Conference. Edinburgh, Scotland, (1999)
2. Daniel A. Menasce and Virgilio A.F. Almeida: Capacity Planning for Web Performance, Metrics, Models, and Methods. Prentice Hall, (1999)
3. Goldman, R., Mchugh, J., and Widom: Proceedings of the 2nd International Workshop on the Web and Databases. Philadelphia, Pennsylvania, June (1999)
4. Blakeley, Michael: WebStone FAQ. Silicon Graphics, Inc. 9 Nov (1995)
5. SPECweb99 Release 1.01 Specification: http://www.sprc.org/osg/web99/ November 8, (1999)
6. TPC BENCHMARK™ W (Web Commerce) Draft Specification: http://www.tpc.org/, November 19, (1999)
7. Robert Orfali, Dan Harkey and Jeri Edwards: Client/Server Survival Guide. Wiley (1999)
8. Ron Vetter: Web-Based Enterprise Computing. Computer, May (1999)
9. Leland Ahlbeck: Improving the Performance of Data Access Components with IIS 4.0. http://msdn.microsoft.com/workshop/server/components/daciiisperf.asp
10. Louis P. Slothouber, Ph.D, StarNine Technologies.: http://louvx.biap.com/white-papers/performance/overview.html
11. Lü K. J, Dempster E. W., Tomov N., Williams M. H.: Verifying a Performance Estimator for Parallel DBMS. Proceedings of Eura-Par 98, LNCS 1470. Springer (1998)
12. Lü K. J, Dempster E W, Tomov N T, Taylor H, William M H, Pua C S, Burger A, and Broughton P.: An Analytical Tool for Predicting the Performance of Parallel Relational Databases. Journal of Concurrency: Practice and Experience. John Wiley & Sons (1999) 1-16
13. Florescu, D.: Database Techniques for the Wide Web: A Survey. SIGMOD Record, 27(3), (1998) 59-74