

Performance Modelling and Metrics of Database-backed Web Sites

Yuanling Zhu, Kevin J. Lü
SCISM, South Bank University
Email: zhuy@sbu.ac.uk

Abstract

Currently, Web database systems are widely used to construct Web sites for their excellent capability of providing on-line information. In this paper, we analyse the workflow of a Web database system dealing with a Web page request. We have classified the different cases and given an approximate method to model a Web database system. As the service time of database servers is a primary factor in determining the input characteristics of Web servers, it is very important to investigate the relationship between database servers and Web servers. The performance metrics is introduced on the basis of the analysis of the relationship.

1. Introduction

As more and more organisations want to extend their business to wider range of users through the Internet, database-backed Web sites are widely built as a promising new platform for client/server database applications. Many efforts have been made on improving the connectivity between databases and the Web [Gre99]. Traditional RDBMS vendors manage to make their products act as Web databases which can be accessed from the Web [Ora99] [Mic99].

Unfortunately, most of current Web database products have been criticised for poor performance. The performance of a Web site is critical. Users might give up visiting a Web site with numerous functions but poor performance. A high-performance Web site will keep its customers' loyalty and become more popular. On the contrary, a poor-performance Web site will bring about serious consequences, such as losing customers, revenue reduction and damaging the company's reputation.

Several benchmarks have been proposed for measuring the capacity of Web sites [TPC99][Spe99]. However, most of them focus on the performance evaluation of Web servers, which neglect the important impact of database servers on the overall system performance. As the process of Web servers depends on the availability of data provided by database servers,

different types of database applications will result in various workloads on Web servers. This paper presents an analytical performance model of Web servers with the consideration of database involved applications. We argue that the Web servers and database servers in a Web database system should not be investigated in isolation. The issues related to performance metrics of Web database systems are also discussed in this paper.

The remainder of the paper is organised as follows: Section 2, gives the definition of Web databases and introduces the related work. Section 3, proposes the performance model of Web database systems and analyses the performance. Section 4, presents the performance metrics of the Web database systems. Section 5, concludes the paper and discuss pending research issues.

2. Related Work

A database with a Web interface, or a database which can be accessed from Internet/Intranet using a Web browser, has been referred by many terms, such as: "Web enabled database", "database with Web connectivity", and "Web database". These terms describe the similar concept and have been used on many occasions. But there is not a generally agreed definition and agreed "term" for it.

A database-backed Web site is constructed by *Web database systems*. Such a system refers to a database with a Web interface that can provide dynamic Web pages to HTTP clients. More precisely, it can be defined like this:

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

The Web-server-as-database-client architecture is the fastest currently popular database-backed Web site architecture [Gre99]. The Web server program maintains a pool of already-open connections to one or more DBMSs. The required data is extracted from the DBMS server and go back to the DBMS client, the Web server, which sends the data back to the client browser. The Web server must wait until all or part of query result is sent back by the database server before proceeding to the next

step. This process can be modelled as a serial queuing system.

During the last few years, several studies have been carried out to model and analyse Web performance problems based on queuing network theories. [SLO96] modelled the Web server and the Internet as an open single class queuing network, from which they derived several performance results for Web servers on the Internet. [MA98] modelled the Web servers as an open multiple-class queuing network model. Similar to the studies in [SLO96], it did not consider the case of dynamic Web page requests, of which the contents is extracted from a backend database server or created by an application. In both models, the impact of database servers is not considered in system performance modelling.

However, as Web applications requiring database accesses become very common in today's Web-based computing, it is thus very important to understand the performance problems of Web systems with consideration of database servers. The performance metrics of the Web database systems also needs to be redefined.

3. Performance Model

3.1 Workload classification

The first step in any performance evaluation study is to understand and characterise the workload [MA98]. The workload can be categorised into classes based on resources usage. Each class comprises requests that are similar to each other concerning resource usage. Because of the heterogeneity of Web applications, it is not sufficient to represent the workload of Web database servers by a single class.

Before characterising the workload of a Web database system, components involved in a Web database application processing must be checked first: when a Web server receives a request with a query to databases, it parses the request and forwards the embedded query to the database server. The database server executes the query and sends the result back to the Web server. The Web server then replies the client with a response consisting of information just retrieved from the database in HTML format. It can be inferred that the workload on a Web database system can be characterised by the system resources consumption on the network, Web servers and database servers. According to our tests, the size of the requested document is the main factor to networking delays [Zhu00]. Similarly, large Web pages will demand much more system resources of Web servers. Meanwhile, database applications are often characterised by the complexity degree of the query.

Table 1 depicts a typical classification of the workload on a Web database system. The workload of a

Web database system is categorised into six classes of similar requests based on the resource usage on the primary components. For class 1, a request of this type embeds complex queries with small result sizes. It demands high CPU but low I/O time consumption on the database server. Consequently, it requires low CPU time of the Web Server to process small amount of data and produces low load on the network.

Table 1. Workload classification in a Web database system

Class	Description	Load on Database Servers		Load on Web Servers		Load on Network
		CPU	Disk	CPU	Disk	
1	Complex query with small result size	High	Low	Low	N/A	Low
2	Complex query with large result size	High	High	High	N/A	High
3	Simple query with small result size	Low	Low	Low	N/A	Low
4	Simple query with large result size	Low	High	High	N/A	High
5	No query, small file size	N/A	N/A	Low	Low	Low
6	No query, large file size	N/A	N/A	High	High	High

3.2 Performance Modelling

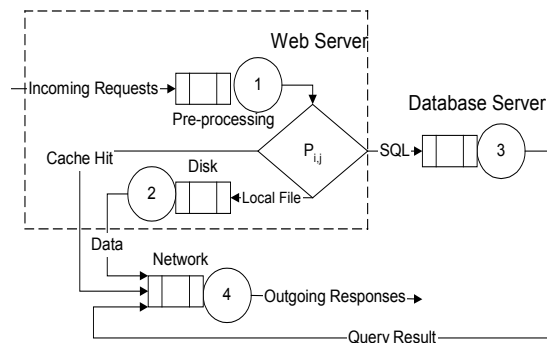


Figure 1. Queuing Network model for Web databases systems

Figure 1 illustrates the model that is used to describe a Web database system. In fact, it is an open multiple-class queuing network model. There are four load-independent queues: the first queue is the queue of incoming requests on Web servers. The second queue is for the file requests and the third queue is for the database-involved requests. The last one depicts the

response queue waiting for being sent back through the network.

Web page requests arrive at the Web server with the average arrival rate $\lambda(\lambda_1, \lambda_2, \dots, \lambda_r, \dots, \lambda_R)$, Where λ_r represents the arrival rate of the workload of class r . Upon receiving the request, the Web server will check its cache first. If the requested data is available there, then the cached data is passed to the network queue waiting for being sent back to the client. Otherwise, the request will proceed to the disk queue of the Web server if it only requests static Web pages, or proceed to the database queue if it requires access to databases. After the requested data is fetched from local disk or retrieved from databases, it goes to the network queue.

In order to present the impact of database servers on the overall performance of a Web database system, the Web database system is modelled as an open queuing network with First Come First Service (FCFS) discipline, multiple classes, and general service pattern at queuing centres containing a single server. This is a typical serial queuing system, which has exact solutions only when the service time distribution is exponential. The maximum entropy method is adopted to solve this kind of queuing network model for its numerical accuracy and low computational cost [Kou85] [Kou93].

Six classes of the workload and four queues are considered in our model. Assume the following notation for class r requests at queue i :

$\lambda_{i,r}$ Mean arrival rate

$C_{ai,r}^2$ Squared coefficient of variation of inter-arrival times

$\mu_{i,r}$ Mean service rate

$C_{si,r}^2$ Squared coefficient of variation of service times

$\rho_{i,r} = \frac{\lambda_{i,r}}{\mu_{i,r}}$ Load on server

$p_{j,t,i,r}$ Probability that a request of class t , having completed service at queue j , then changes to of class r , and queues for service at queue i

Let $\lambda_r (r=1,2,\dots,6)$ be the arrival rate of class r requests. The average arrival rate is computed as:

$$\lambda_r = \lambda \times \text{PercentOfClass}_r \quad (1)$$

Let $\lambda_i = \sum_{r=1}^R \lambda_{i,r}$ be the total mean arrival rate at queue i , the probability that an arbitrary request at queue i is of class r can be computed as:

$$\pi_{i,r} = \lambda_{i,r} / \lambda_i \quad (2)$$

p_{ji} is the probability of a request going from queue j to queue i :

$$p_{ji} = \sum_{r=1}^R \sum_{t=1}^R \pi_{j,t} p_{j,t,i,r} \quad (3)$$

The coefficients of variation of the departure process at queue i is:

$$C_{di}^2 = \frac{(1-\rho_i)\lambda_i}{\sum_{j=0}^M \lambda_j p_{ji} / [2 + p_{ji}(C_{dj}^2 - 1)]} + \lambda_i \sum_{r=1}^R \frac{\rho_{i,r}(C_{sir}^2 + 1)}{\mu_{i,r}} + 2\rho_i(1-\rho_i) - 1 \quad (4)$$

Where $\rho_i = \frac{\lambda_i}{\mu_i}$ denotes to the load on queue i .

Then the coefficients of variation of the inter-arrival processes, i.e., the differential of requests transferring from queue i to queue j is:

$$C_{dj,i}^2 = 1 + \sum_{r=1}^R \sum_{t=1}^R \pi_{j,t} p_{j,t,i,r} (C_{dj}^2 - 1) \quad (5)$$

The squared coefficient of variation of arrivals at queue i is:

$$C_{ai}^2 = \left(\lambda_i / \sum_{j=0}^M \frac{\lambda_j p_{ji}}{C_{dj,i}^2 + 1} \right) - 1 \quad (6)$$

The squared coefficient of variation of service time of queue i is:

$$C_{si}^2 = \frac{\mu_i}{\rho_i} \sum_{r=1}^R \frac{\rho_{i,r}(C_{sir}^2 + 1)}{\mu_{i,r}} - 1 \quad (7)$$

Finally, the mean number of requests in the queue i is given by:

$$\bar{n}_i = \frac{\rho_i}{2} \left(1 + \frac{C_{ai}^2 + \rho_i C_{si}^2}{1 - \rho_i} \right) \quad (8)$$

Let $\bar{n}_{i,r}$ be the mean number of requests of class r in queue i , it can be calculated by:

$$\bar{n}_{i,r} = \pi_{i,r} (\bar{n}_i - \rho_i) + \rho_{i,r} \quad (9)$$

Since a request's waiting time at the queue does not depend on its class, the mean waiting time per class r , is (by Little's Law):

$$W_r = \bar{n} / \lambda - 1 / \mu \quad (10)$$

And the mean response time per class r is:

$$R_r = \bar{n} / \lambda + 1 / \mu_r - 1 / \mu \quad (11)$$

3.3 Performance Analysis

The model in this paper is a serial queuing network, which implies that all the data must be read from disks or retrieved from databases before going to the network queue. When the size of requested data is very big, the Web server will wait for a long time before it can process the request. In fact, some parallelism exists between the database server and the Web server. The Web server can start sending part of data back to client while the database server is still retrieving the rest of the required data. There is an overlap between the time spent on the Web server and the database server, especially in the case of large amount of data being extracted from databases. This feature is examined by a simple experiment summarised below:

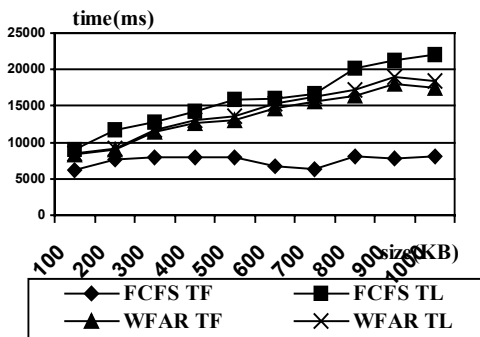


Figure 2. response time with varying result sizes

The experiments were conducted using one client machine and one server in a 10Mbps Ethernet environment (the network bandwidth is limited to 193Kbps to simulate the typical Internet connection). Microsoft Windows NT IIS and SQL Server are adopted as the Web server and backend database server. The database is accessed by an ASP file, which embeds a simple query performing a table scan. Query result sizes selected range from 100KB to 1MB. Two strategies have been tested on their impacts to the T_F (the time that the user need to receive the first page of data) and T_L (the time that the user need to receive all the data) with varying result sizes:

1. First Come First Service (FCFS): When the Web server receives first packet of data from the database server, it begins to send the result back to users.
2. Wait For All Result (WFAR): The Web server will wait until all the result has received from database

server before it begins sending the result back to users.

Figure 2 shows that users can always see the first page earlier if they adopt the FCFS strategy. With the increase of the result size, the duration between the first page and last page arrival time increases sharply. It can be explained by the overhead of too many small packages. Under the WFAR strategy, the Web transaction always ends earlier despite that it's very late when the Web server begin to send data. In this case, the Web server and the database server work in sequence. When the query result size is quite big, users have to wait for a long time to see the first page. The times, FCFS T_F and WFAR T_L , can be viewed as the optimal first page arrival time and the optimal last page arrival time, respectively. As a compromise, carefully selected buffer sizes on the Web server will result in both acceptable first page arrival time and last page arrival time.

Table 2. Resource usage of each class of workload on queue l and different workload

Class	Mean service rate			Service time deferential			Load(%)					
	Disk	DBS	Net	Disk	DBS	Net	1	2	3	4	5	6
1		50	500		3	1	0	0	0	0	0	0
2		20	50		4	3	0	0	0	0	0	0
3		200	500		1	1	26	31	36	41	45	50
4		40	50		2	3	34	33	32	31	30	29
5	200		500	1		1	30	25.5	21	16.5	13	8.5
6	40		50	2		3	10	10.5	11	11.5	12	12.5

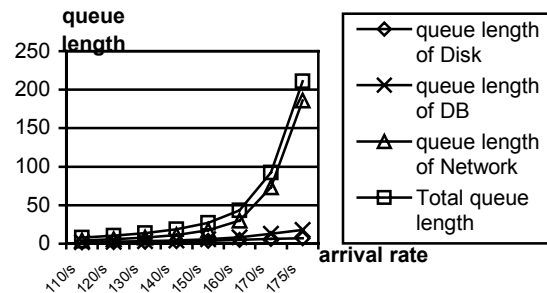


Figure 3. Queue length with different workload

In order to simulate this feature by the model proposed in last section, a request of class 4 can be seen as several requests of class 3. Table 2 shows the initial parameters of the model according to the class of workload and lists some workload distribution with the same load on the database server. Figure 3 shows the queue length of each resource under these workloads. When some requests of class 4 was replaced by some requests of class 3 with the same load on the database server, the queue length of the database server decreased. Moreover, the queue length of

the network decreased too. It is attribute to the less workload on the network and higher resource utilisation.

4. Performance Metrics of Web databases

A typical Web database system user receives and views the results of a query in the format of pages, which is different from the centralised or client/server database systems, where a user normally require to receive the entire result at one time. As a result, the user's expectation to the performance of a Web database system is also different in several ways.

Suppose the result size of a Web database query is 10MB, the user does not mind whether the whole result file has arrived at the same time or arrived in sequence. On the contrary, if the user can receive the result page by page on time, he/she would *feel* being served effectively and satisfy with the service. In addition to the demand of "read earlier", users also do not expect the last page arrive too late. It's the common sense of any database users to complete a transaction as early as possible. Unlike the conventional *response time*, new measures presenting the *First Page Arrival Time* (FPAT) and the *Last Page Arrival Time* (LPAT) should be examined together to describe the performance of Web database systems.

Meanwhile, if users received data page by page, they also do not expect some pages arrive in a much slower speed than others do. Web database systems should enable the query result to arrive at the client at an acceptable rate besides providing the first page earlier on. *Result Arrival Rate* (RAR) presents 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 to users.

From the viewpoint of a Web site administrator or an Internet Service Provider (ISP), the most concern is to achieve the ultimate utilisation of all their investments on hardware and software. *Throughput Ratio* (TR) indicates if the capacity of the Web server matches the capacity of the database sever 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 unit of time. If their throughput ratio is nearly equal to 1, i.e., their processing speeds match to each other, the Web database system can achieve its maximum throughput.

The above four newly identified factors are essential for measuring the performance of Web databases, which would be helpful to provide a clearer picture about the performance of Web database systems. As our investigation progresses further, we believe that there could be more factors to be identified to measure the Web database performance.

5. Conclusion

In this paper, components involved in a web database transaction processing are examined and their relationships to the performance of Web database systems are described. With the consideration of the aspects of Web database applications, workload on a Web database system is categorised into multiple classes with different resource usage. Based on this, a Web database system is modelled as a multiple-class open queuing network. With this model, performance metrics of a web database system can be approximately calculated. New performance metrics is also introduced to reflect web database users' demands.

The understanding and management of the performance of Web database systems is still an area in its infancy. It is critical to provide simple yet flexible ways to describe the performance of Web database systems. Otherwise, the performance management tasks like capacity planning, application sizing and performance tuning would be impossible to proceed.

6. Reference

- [Gre99] Philip Greenspun, 1999. *Interfacing a Relational Database to the Web, Chapter 13*, Philip and Alex's Guide to Web Publishing, Morgan Kaufmann Publishers
- [Kou85] Kouvatso, D.D., 1985. *Maximum Entropy Methods for General Queueing Networks*. In D. Potire, editor, *Modelling Techniques and Tools for Performance Analysis*, Pages 589-608. North-Holland, Amsterdam, The Netherlands, 1985.
- [Kou93] Kouvatso, D.D., Denazis S.G., 1993, *Entropy Maximised Queueing Networks with Blocking and Multiple Job Classes*, *Performance Evaluation* 17(3),1993
- [MA98] Menasc, D., Almeida, V., 1998. *Capacity Planning for Web Performance*, Prentice Hall Publisher
- [Ora99] *Oracle Application Server Technical Document*, 1999. <http://technet.oracle.com>
- [Mic99] *Application Server Page Technical Document*, <http://msdn.microsoft.com/library/>, 1999
- [Slo96] Slothouber, L., *A model of Web Server Performance*. <http://louvx.biap.com/WebPerformance/modelpaper.html>, 1996
- [Spe99] *SPECweb99 Release 1.01 Specification*, <http://www.sprc.org/osg/web99/>, 1999
- [TPC99] *TPC BENCHMARK™ W (Web Commerce) Draft Specification*, <http://www.tpc.org/>, November 19, 1999
- [Zhu00] Zhu Y., Lu, K.J. 2000, *Web Databases and Related Performance Issues*, Submit for publication.