An Investigation to the Performance of Web Database Systems

1 Introduction

The World Wide Web (WWW) has been proven to be the largest information database and an excellent medium for businesses, and the ability to populate Web sites with content derived from large databases has become the key to building enterprise Web sites [1]. Web databases, which provide Internet-based accesses to remote users, enable companies to extend their services to a wider range of users through the Internet. As business transactions are increasingly executed by means of the Web, poor performance of Web databases may bring about serious consequences, such as losing clients, revenue reduction and damaging the company's reputation. The Web database system designers believe that performance prediction of the Web database can be used to evaluate different design options to optimise performance [8,10]. After the system is developed, the performance prediction will be useful in determining the configuration of a system to meet user requirements as well as in subsequent tuning to obtain improved performance. As a result, there is an increasing need of a tool to predict the performance of Web databases. The purpose of this project is to investigate the performance of a Web database system, and to build a simulation model of a Web database to provide a means to analyse, tune and predict the performance of the Web database.

2 Background to Project

2.1 Web databases

The development of Web database has posed as a new and challenging database problem, as is shown in a number of research proposals relating the database area for data management on the Web sites [11,12]. It has aroused immense interest in the database community. A series of conferences on Web databases (e.g. International Workshop on the Web and Databases) have been held since 1998. Recent research interests have been clearly moving in the direction of specification of the structure and content of Web sites on how to find useful information quickly from the Web databases [3]. Yet there has been little research work done specially on the Web database performance.

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. Normally, an application server is needed to glue a database server and a Web server together. The main products of Web servers are Apache, Internet Information Server, Oracle Application Server and others. These main products are Oracle, SQL Server, DB2, Sybase and Informix. Web database application servers include ASP, Code Fusion, Tango, JSP, Java, Servlet, and CGI Java Servlets and Active Server Pages are two major server-side Web applications to use this method [4]. The diagram doesn't mean that three components should be involved at the same time. Actually, some products combine several of these components into a single package. For example, File Maker Pro. It is a database product, but it has a build-in Web server. [13]

2.2 Performance of Web databases

The purpose of studying Web databases performance is to provide the quality of service required by the users. At the most fundamental level, the performance of Web database systems can be described in terms of just five simple concepts: *Workload, Response time, Throughput*, *Resource utilisation* and *Bottleneck*.

The term *workload* designates all the processing requests submitted to a Web database system by the user community during any given period of time. The term *response time* refers to the time required to process a single unit of work (such as a query). It records the time between the instant of a request and the computer's *response* to that request. The throughput of a Web database system is the average rate at which the system completes jobs in an interval of time. Resource utilisation is the ratio of the busy time and idle time of resource. Bottleneck is the resource that has the longest service time in the system.

When a system is applied to the task of processing a given **workload**, **response times**, and **throughput**, are *external* measures that describe the observed performance. **Resource utilisation's** and **bottleneck** describe internally the behaviour system.

2.3 Advantages and disadvantages of simulation analysis

Analytic models have definite advantages to be considered carefully and these advantages include: conciseness in problem description, closed form solutions, ease of evaluating the impact of changes in inputs on output measures. On the other hand, there are some disadvantages, such as assumptions regarding system description which may be unrealistic and complex mathematical formulations which defy solution. For example, queueing networks, but either the assumptions required for analytic solution are somewhat unrealistic (e.g., exponential interarrival and service times). Simulation models can compensate for these disadvantages. Simulation models can describe systems which are very complex. They can be used to experiment with systems which are not yet in existence, or to experiment with existing systems without actually altering the system. A Web database system is a complex system if the optimisation, cache, connection to database servers, and etc. are considered. Analytic models can also be used in these ways only if the system is not too complex. On the deficiency side of the ledger, there are no closed form solutions to simulation models; each change of input variables requires a separate solution or set of runs. The request arrival pattern may be changed for different Web sites.

3 Aims and Objectives

The purpose of this project is to investigate the issue of performance of Web databases and to provide means sizing, analysing, tuning and predicting the performance of Web databases. The research will

1. Provide a comprehensive understanding of the possible operational mechanism that could be used in Web database systems.

2. Examine the factors that may affect the performance of Web databases.

3. Identify the metrics of Web database performance.

4. Investigate potential optimised query strategies for a Web database system.

5. Develop tools to model workload.

6. Investigate the approach to estimate the web database performance and make a general model based on it.

7. Implement, calibrate and validate the model.

8. Develop software tools for estimating the performance of a Web database, capable of system tuning, bottleneck identification and prediction of the performance of the database.

4 Context

4.1 Project description

The average response time for a Web database depends on the amount of time consumed in Internet transmission, Web server service and Web database service. There is little we can do to reduce Internet transmission time, but we have a large space employed to improve the performance by reducing Web server and database service time. With the fast changing technologies on the Web, the study of the performance of Web databases has taken on a great deal of additional complexity. In addition to considering the different databases such as Oracle or SQL Server for the Web site, and development of environment of the Web server, such as ASP or Java servlet, this project needs especially to consider factors like different database Web connectivity, cache setting between Web server and database.

The core task of the project is to use a simulation approach to model the performance of Web databases. The questions to be addressed deal with the effects of changes in certain configuration of the system and operating policies on the system performance. The system is entered when a request is arrival and the server deal with the request, and a response page is generated. The server operations included in the system are: initialising a thread for the request, the web server parsing the request, querying the database server, and transmitting the respond page out.

4.2 Web database model building method

The data collection method should be tailored to the particular system setting if the data are to be collected efficiently and accurately. Determine a theoretical probability distribution which is appropriately similar to the real sample, and use it in the model. Exogenous variables, referred to as input variables, for this system include: 1) Number of maximum threads 2) Number of processors 3) Arrival pattern of requests. Endogenous variables as output variables for this system include: 1) Response time 2) Throughput of the server system 3) Resource utility (bottleneck) 4) Number of request waiting to process 5) Number of response waiting to transmit 6) Number of refusing respond

The measures which we choose to minimise or maximise in configuring the system are referred to as the objective function. For the web database system, we decide that minimisation of customer waiting time and maximisation the server capacity are a

desirable measure of system performance. Using event tracking, the simulation logic notes when events, and hence changes in the system status, occur and only checks the system at these times, changing the affected variables.

Programming language will be GPSS, or C++. If the model is programmed in a general purpose language, the analyst must program in detail the collection of all performance statistics. To do this simulation languages permit copious statistics collection using simple program statements.

Each operation to the database will be analysed and changed into workload first and then the value of Web database metrics will be calculated accordingly. The model should be validated by means of tests on some real Web databases. A new optimised approach may be developed during the construction of the model. Finally, a tool based on the model should be developed to help people to estimate the performance of Web databases.

A Web database model will be set up for capacity management and planning. To construct such a model, the project will follow the following major steps:

a. Base line model construction and its parameterising
   A baseline model is an analytical model built to represent the current state of the system and workload [7]. The model is to be built from a standard system and workload to monitor sources. It should be appropriate to the hardware and operating system in use. A typical real Web database will be built up. The database server will connect to LAN and to Internet through a gateway. The transport protocols used throughout the networks should be TCP/IP.

   A base line model, including a workload model and a performance model, will be set up based on the database. Performance metrics and workload characterisation will be prepared for the model.

b. Validating the base line model
   During the experimental execution of the prepared workload on the system, the system will be monitored and the collected data will be analysed by statistic software. Actual measurement data of the system being modelled should be compared against their results from the performance model. Only when the result is acceptable, then the next step may be continued. Otherwise the work should go back to Step a to address the source of the errors.

c. Judging the model by changing some of its components
   Once the base line model is validated, it will be considered as a predictable model. The real system should be monitored when some components are changed, e.g. changes in hardware, like CPU or Disk, or settings, like the size of cache, database size, especially changes in the configuration of a bottleneck. The measured data should be checked against the result from the predictable model. If the result is not acceptable, the work should go back to Step a and modify the base line model. [8,9]

The whole procedure is an iteration process until the satisfied model is built up.

4.3 Significance of the proposed research

A Web database is one of the most powerful ways of building and maintaining a good Web site and it can be cheaper than a conventional Web site. A Web site development can be seen as a process in three phases: 1. Pre-setting up of a Web site; Web developers need to select the hardware (such as T1 line or T3) and software (such as what kind of database?) to meet the user requirements, such as under 8s respond time with a 10,000-hit per day. The correct selection will avoid costly mistakes in the construction of the Web systems and help developers avoid wasting time in unproductive tuning experiments after the event. 2. Maintaining a Web site; For the existing Web database system, the Web masters should find the bottleneck of the system and tune the system to get the best performance. 3. Expanding a Web site; When the Web database needs to be expanded due to the successful operation of the Web site, the developers should know the performance of the new Web database and upgrade the system accordingly.

To complete the operations of the above mentioned phases, suitable tools are required to predict the performance of the Web database systems for the target applications. They will determine whether a given hardware/software configuration will meet a user's requirements in performance and how performance would change with changes to the load. Such tools will rely on an accurate model of throughput, utilisation, and perhaps also the response time for given system configurations and workloads. The tools developed based on this model can be used to predict the performance of the Web site.

5 Proposed Plan of Work

The main tasks of the project are as follows:

a) Familiarise with Web development tools, DBMSs and analytical modelling
b) Examine the relationship between the Web server and the database server
c) Examine the relationship between cache and the Web server and the database server
d) Examine the relationship between the query execution plan and the database server
e) Evaluate the existing query execution plan
f) Plan the analytical model
g) Evaluating different Web databases performance
h) Design and implement the tool for estimating Web databases performance
i) Study the way of improving Web database performance
j) Evaluate and refine the final model tool
k) Final documentation

The detailed work plan is expected to be as follows:

task	Months from start											
	1	4	7	10	13	16	19	22	25	28	31	34
a	√											
b		√	√									
c			√	√								
d				√	√							

e					√	√	√					
f		√	√	√	√							
g			√	√								
h			√	√	√	√						
i				√	√	√	√	√	√			
j								√	√	√	√	
k											√	√

6 References

1. Daniela Florescu, Alon L. and Dan S. (1999) "*Optimization of Run-time Management of Data Intensive Web Sites*", Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, 1999.

2. Robert Orfali, Dan H. and Jeri E. (1999) *Client/Server Survival Guide (3rd Ed.)*, Wiley.

3. Daniela F., Alon L. and Alberto M. (1999) "Database Techniques for the World-Wide Web: A Survey", ???

4. Ron Vetter (1999) "*Web-based enterprise computing*", Computer 5/1999

5. Jon Siegel (1998) "OMG OVERVIEW: CORBA and the OMA in Enterprise Computing", COMMUNICATIONS of the ACM, Oct. 1998, Vol. 41, Number 10.

6. Richard Grimes (1997) *Professional DCOM Programming*, Wrox.

7. Raj Jain (1991) *The art of computer systems performance analysis, techniques for experimental design, measurement, simulation and modeling*, John Wiley & Sons, Inc.

8. Daniel A. Menasce and Virgilio A.F. Almeida (1998) *Capacity Planning for Web Performance, Metrics, Models, and Methods*, Prentice Hall.

9. Daniel A. Menasce, V. A. F. Almeida and L. W. Dowdy (1994) *Capacity Planning and Performance Modeling, From Mainframes To Client-Server Systems*, Prentice Hall.

10. Neven T., Euan D., Williams M. H , Peter J.B. and Albert B. (1999) "Approximate Estimation of Transaction Response Time", *The Computer Journal*, Vol.42 No.3, 1999, pp. 241-250.

11. Proceedings of the International Workshop on the Web and Databases, Valencia, Spain, March 1998.
    http://www.dia.uniroma3.it/webdb98/

12. ACM SIGMOD Workshop on the Web and Databases Philadelphia, Pennsylvania, USA- June 3-4, 1999.
    http://www-rocq.inria.fr/~cluet/webdb99.html

13. John Paul Ashenfelter, *Choosing a Database for Your Web Site*, Wiley, 1999.