Performance of database driven network applications from the user perspective

This is the Published Version of the final output.

Performance of Database Driven Network Applications from the User Perspective

Shanyu Tang[1], Huang YongFeng[2] and Yau Jim Yip[3]
[1] Faculty of Computing, London Metropolitan University
London, London N7 8DB - UK
[e-mail: s.tang@londonmet.ac.uk]
[2] Department of Electronic Engineering, Tsinghua University
Hai Dian, Beijing 100084 - CN
[e-mail: yfhuang@tsinghua.edu.cn]
[3] School of Computing and Engineering, University of Huddersfield
Queensgate, Huddersfield HD1 3DH - UK
[e-mail: j.yip@hud.ac.uk]
*Corresponding author: Shanyu Tang

Abstract

An understanding of performance of database driven network applications is critical in combating slow performance of e-commerce web sites, besides efficient web page design, and high quality and well-maintained web server equipment. This paper describes a method of measuring performance from the user viewpoint, which can help enormously in making realistic assessment of true performance of database driven applications. The performance measurements were taken at user locations by using several specially designed JavaScript functions along with ASP scripts. A performance study is presented in this paper, comparing performance of data access using stored procedures with the traditional way of querying a database. It is generally believed that stored procedures have performance benefits as they are pre-compiled. However, our study shows that the data access approach using stored procedures provides significant benefits (by about 30%) over the traditional approach for querying a commercial MySQL database, only when retrieving a substantial amount of data (at least 10,000 rows of data).

Keywords: Performance, stored procedure, database driven application, user perspective

---

1. Introduction

The rapid and widespread use of computers and information technology has led to exponential growth of e-commerce all over the world. The information technology industry demands technological advances in the computing, networking and telecommunication fields. The industry is highly competitive; on the one hand there is a large number of firms and individual operators, but on the other hand, the growing 'download culture' is threatening some businesses as sales are declining. In many cases, an e-commerce company relies on its products, but also requires a competent management team, good post-sales services, well-organized business structure, network infrastructure and a secured, well-designed website. The most successful e-commerce web sites are driven by large database systems, which enable a large number of data to be stored and processed into information. The information is then used to satisfy customers and their future needs, thereby offering competitive deals in several markets. E-commerce web sites have generally become a single access point to the entire stakeholder – customers, investors, employees and external partners for companies. The management of their perceptions and the e-commerce web site has become important for business success. With millions of customers now online, the importance of e-commerce web sites in influencing their purchasing decisions is significant.

Many companies use their e-commerce web sites to generate revenue, either indirectly through marketing and advertisements, or directly through e-commerce applications. Loss of e-commerce web site visitors often means loss of business. Slow or unreliable performance at these e-commerce sites has real consequences. Performance is a particularly sensitive issue for e-commerce sites, where potential buyers typically must download multiple pages and data, as well as supply information to the vendor. It is estimated that slow performance (i.e. a web page takes any longer than eight seconds to load) costs e-commerce web sites as much as million pounds annually in lost revenue. High-quality web servers and Internet link equipment with vigilant maintenance have been used to ensure performance and reliability of e-commerce web sites. Good web site performance is taken for granted when the health of the server and its links is good. Web server administrators typically measure the performance of web sites, if at all, using a single workstation, often local to the web server itself. This approach does little to convey true performance as it evaluates web site performance and reliability from the inside looking out. Sole reliance on performance and connection status at the server level can badly mislead a business about what actual users experience on the web. So it is necessary to explore a way to measure performance from the user viewpoint, which is an important part of this study.

In Section 2, we review the previous work in performance measurement. Section 3 addresses the importance of stored procedures. Section 4 presents a case study, detailing experiments, performance measurement of database driven applications, data access methods, comparisons between stored procedures and the traditional approach, effects of ASP.NET, implications for web application design, and security issues. Section 5 concludes with a summary.

2. Related Work

Conceptual and empirical studies on the web site quality have been reported in the information systems literature. Research identified some factors that contribute to the overall web site

quality. Speed of delivery, reliability, content / functionality of web sites, trust, inter-activeness, and ease of use are the most important service quality dimensions for web users / customers [1].

Liu and Arnett identified information quality, system use, system design quality and playfulness as four major determinants for the success of e-commerce web sites [2]. For rating web sites and comparison purposes, web design industries are usually concerned with the web site design, access to web site and/or speed with which pages download.

Janda et al proposed five dimensions, performance, access, security, sensation and information, by studying the consumer perceptions of Internet retail service quality [3]. The performance dimension measures the page download speed of the retailer's web site, while the information dimension measures the quantity and credibility of information provided by the web site, and the security dimension deals with the privacy and financial integrity issues. McKinney et al suggested nine key constructs separating web site quality into information quality and system quality for measuring the web-customer satisfaction [4].

Similarly, six dimensions of consumer perceptions of service quality – ease of use, content, timeliness of response, accuracy of content, aesthetics and privacy, were identified and measured by Yang et al [5]. The 'ease of use' factor includes user friendliness, loading / transaction speed (depending on page download speed), search capability, and easy navigation.

Palmer used media richness and design and usability principles to validate the measures of specific web site attributes that can be used to identify elements of successful web site design [6]. The attributes include page download speed, sequence, layout, arrangement, web site content that includes amount and variety of product/company information, customization, and interactivity that covers easy of navigation and responsiveness.

Aladwani and Palvia's work [7] is the first appropriate instrument to measure the web quality from users' perspective. They developed a multi-dimensional scale for measuring user-perceived web quality, and explained the user-perceived web quality with the help of four dimensions – technical adequacy, specific content, content quality and appearance and validated the instrument. They conducted an empirical study and extensive testing for the validity and reliability of the instrument and suggested generalisability of the instrument.

Ravi [8] validated the instrument developed by Aladwani and Palvia using Structural Equation Modelling methodology. Analysis revealed that the data set did not fit the Aladwani and Palvia's model well enough, i.e. the model did not reflect the overall construct in its entirety. However, further refinement of the instrument with more empirical studies is necessary.

Performance measurements from the server perspective are normally run on multiple distributed servers with massive amount of disk, RAID, SAN and such architectures, and hardly run a web server on the same server hardware as a database server. Both servers are usually mirrored and almost likely the database server is behind a secure firewall. In such an environment, if real performance comparison is to be researched, very specific test cases should be built, with time comparison taking into account network latency, amount of load on the servers, and such factors. Unfortunately, it is not easy to account for network latency, amount of load on the servers, due to their variations with time.

Banga et al [9] attempted to develop a method for Web traffic generation that can generate bursty traffic, with peak loads that exceed the capacity of the server, and their method models the delay and loss characteristics of WANs. They used the proposed method to measure the performance of widely used Web servers, and reported that actual server performance could be

significantly lower than indicated by standard benchmarks under conditions of overload and in the presence of wide area network delays and packet losses.

One promising approach to testing the functionality of web applications leverages user-session data collected by web servers [10]. User-session-based testing automatically generates test cases based on real user profiles. Andrews et al [11] suggested the application of concept analysis for clustering user sessions and a set of heuristics for test case selection. Concept analysis provided a promising means for incrementally updating reduced test suites in response to newly captured user sessions with little loss in fault detection capability and program coverage.

Effective performance testing techniques are essential for understanding whether a web-based application will meet its performance objectives when deployed in the real world. Shams et al [12] proposed a model-based approach for testing the performance of web applications. The application model that captures the dependencies for a web-based system was used to obtain a large set of valid request sequences representing how users typically interact with the application, thereby automatically constructing a synthetic workload with desired characteristics.

The good way to obtain accurate web site performance data, such as response time over the Internet, from the user perspective is for measurements to be taken at customer locations - in other words, from points on the Internet. Measuring the quality of web site from the users' perspective, will give a fast and early feedback to the company and enables it to take corrective actions, develop an appropriate e-business strategy and improve its operations.

Several researchers have used JavaScript to measure web page access time. Felten and Schneider measured the access time for a particular Web URL using a JavaScript program embedded in a Web page [13]. The JavaScript program runs on the client browser and measures the time both before and after loading a file. Yang et al. suggested slow-motion benchmarking for evaluating thin-client systems in which performance is measured by capturing network packet traces between a thin client and its respective server during the execution of a slow-motion version of a conventional benchmark application [14].

While there has been work on performance measurement, prior work has concentrated primarily on various instruments available to measure the web site quality from the user perspective [15][16][17][18]. Most of research are focused on the Internet factors (such as traffic bottlenecks on specific public network backbones) that affect users' ability to access a company's web site, little attention has been paid to the effect of the underlying data access method on a commercial database. This study is aimed to address this issue by evaluating the performance differences between the data access approach using stored procedures and the traditional approach for querying a commercial MySQL database.

3. Stored Procedures

Stored procedures are useful to someone developing massive databases with a web front- or backend [19], but also to those who develop general applications, which are somehow connected to a database. This study is about the supplement of retrieving information stored in a MySQL database to hundred, thousands or even millions of users through the World Wide Web.

Basically stored procedures are "mini – programs" which are stored and executed within the database management system (DBMS), and they could be useful in order to increase performance and to secure the actual data. Since stored procedures are executed in the DBMS

the execution logic of the stored procedures is cached and so can be accessed immediately without additional compilation.

The implementation of stored procedures in MySQL 5.0 is expected to lead to widespread enterprise adoption of the already extremely popular MySQL database [20]. It is a fundamental open source for building web-based database applications in the future. Since stored procedures enhance performance through caching the execution plan of the particular query or action within the DBMS therefore execution is possible immediately without compilation. Furthermore stored procedures represent a layer between the database and the actual application. Changing the database structure does not necessarily need to rewrite the application, since it is only calling stored procedures within the database. The stored procedures need to be adapted to the new given structure, leading to an additional level of abstraction.

Paul DuBois provides a problem-and-solution format offering examples for everyday programming dilemmas [21]. He explains clearly how and why the code works, which is essential when adapting the techniques to similar situations. Guy Harrison and Steven Feuerstein detail everything from language basics to application building to advance tuning and best practices [22]. They explain valuable information in stored programs, error handling, stored functions and triggers, as well as security basic and advanced SQL tuning, optimising and programming best practice.

Russell Dyer packs the information on MySQL into the concise, comprehensive, and extraordinarily easy-to-use format [23]. His important resource documents the details that experienced users need to take full advantage of this powerful database management system. Jeremy Zawodny and Derek Balling explain indexing and optimisation in depth to make better use of these key features [24]. Practical replication, backup, and load-balancing strategies with information that go beyond available tools to discuss their effects in real-life environments are clarified as well as supporting techniques, advanced configuration, benchmarking, and investigating logs.

4. Case Study

4.1 Experiments

The goal of our experiments was to simulate an e-commerce web application environment based on a commercial database. For a normal e-commerce database, it is necessary to identify desired data, which should be saved in order to support the business strategy and operation. Fundamentally four areas, such as Customer Details, Product Details, Supplier Details, and Order Details, are generally important so as to ensure the business operations within an online environment, nevertheless throughout the development additional alteration might have been made.

The experiments include the use of the dynamic approach of prototyping in combination with ASP and ASP.NET as scripting languages and MySQL as database server.

MySQL was chosen as database server for the e-commerce web application as it can be used on Unix, Linux, Windows or the Mac OS X platform. The reason for using this database is due to the fact that it is available for free; it is quick and efficient and it functions on most existing hardware, and no powerful and expensive server is needed. Furthermore there is a large community, which communicates through different mediums such as forum, web blogs, IRC channels and mailing lists. MySQL can be used in different program environments regardless the language as it is compatible to several languages, including JAVA, ASP, .NET, PHP, Perl

etc. London Metropolitan University provided the necessary infrastructure to use MySQL for web applications.

Active Server Pages (ASP) is a powerful web application tool to program dynamic web pages. ASP allows the developer to create dynamically HTML code, which is interpreted by all browsers. It is an alternative to CGI or PHP scripts. Complex web database applications can be developed and maintained through ASP applications, because ASP's built-in objects enable them to create their own dynamic, database-driven e-commerce solutions.

ASP.NET is one of the most advanced programming languages so far for developing web applications particularly for those database-backed. It is a language-independent technology used to develop dynamic web applications within the .NET Framework.

4.2 Performance Measurement

The purpose of the performance measurement is to quantify the benefits and cost of the data access with stored procedures within an e-commerce application based on a MySQL database, which was installed on the Unix server.

The performance measurements were taken on a single processor 1.5 GHz Pentium-4 machine (User machine) with 512 MB memory and 80 GB hard disk drive, running Microsoft Windows XP Professional with IIS that supports ASP, which was linked with the Unix server (www3.north.londonmet.ac.uk) where a MySQL database was installed. In order to minimize the effect of the user machine on the performance measurement, other windows programs were closed when the performance tests were conducted, because workload on the user machine can affect the display time of large amount of query results.

```
var enter;
var exit;
var time_dif;

function start() {
        enter = new Date();
}

function stop() {
        window.document.open();
        window.document.writeln('stopping');
        exit = new Date();
        time_dif = (exit.getTime() - enter.getTime());
        window.document.write('The execution needed: ');
        window.document.write(time_dif +'ms');
        window.document.close();
}
```

Fig. 1. JavaScript functions used.

Measurements can be taken of a single web page or of multi-page downloads. These consist of HTML pages along with all their component images. Each measurement consists of an HTTP GET request (rather than a "ping" that merely checks the accessibility of an IP address). The GET request is made of a web page or other object on a web server. The measurement starts at that point and ends after the page is downloaded.

The execution cost of a query on a MySQL database was measured by means of implementing several specially designed JavaScript functions (**Fig. 1**) along with ASP or ASP.NET scripts on the user machine running XP and IIS that supports ASP. The detail is

given below.

The `start()` function is used to specify the starting point when a web page requests a connection to the MySQL database server. The execution of the `stop()` function enables the calculation of the elapsed time for retrieving and displaying a chunk of data from a database, i.e. time_dif. As "`new Date()`" creates a new `Date` object, and the `getTime()` function returns the current time based on the user machine, so `time_dif` returns the execution time.

Fig. 2 is the skeleton of the JavaScript embedded ASP or ASP.NET programs running on the user machine, used to measure the execution cost of a query on a MySQL database on the Unix server. The program calls the `start()` function first, creating a date object capable of marking the starting time of the measurement. The MySQL ODBC 3.51 Driver is used to connect the MySQL database server via ADODB.Connection. The recordset of database tables is retrieved by using either the traditional approach (i.e. the complete SQL statement) or stored procedures. The content of the recordset is then displayed on a ASP or ASP.NET web page. The calling of the `stop()` function enables the calculation of the difference between the starting – point and the end – point of the transaction executed within the ASP or ASP.NET page. The times are measured in milliseconds and are displayed.
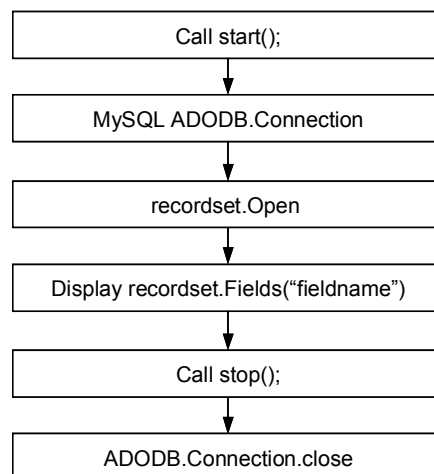
```
Call start();
    │
    ▼
MySQL ADODB.Connection
    │
    ▼
recordset.Open
    │
    ▼
Display recordset.Fields("fieldname")
    │
    ▼
Call stop();
    │
    ▼
ADODB.Connection.close
```

Fig. 2. The skeleton of JS embedded ASP or ASP.NET programs for performance measurement.

Although the above measurement method may not be complicated in comparison with the performance measurements to be taken on network servers, it serves the purpose, i.e. measuring the execution time of data queries from the user perspective. The simplicity of our measurement method can also minimize the effect of the method itself on the 'real' execution time of data queries.

This case study was to compare different data access technologies. The user machine - used to run the performance measurement software - was restarted after each run of data queries on a MySQL table, in order to ensure that there were no bias caused by executing one application before another. Otherwise, executing one application first would cause the database to cache the results partially before the other application is executed.

The experiments (applications) were executed multiple times to remove the variability caused by system load and other factors (e.g. network latency, amount of load on the servers). The performance results are the averages of execution times from five runs of data queries on a table containing numbers of rows. The measurements had variation of 5.1% probably due to differences in the function calling time and system idle fluctuation.

There may be another issue, i.e., how to consider actual (target) users' practical environment (including client and network) in such a testing. This case study tested one experimental client. In some cases, separating network and client effects from the server effect are needed for correct analysis and comparison.

4.3 Data Access Methods

There are two methods of querying a database, the traditional approach that involves the transfer of the complete SQL statement, and stored procedures just calling a procedure within the database management system (DBMS).

4.3.1 The traditional SQL approach

The following SQL embedded ASP scripts were used to connect and open a MySQL database (on www3.north.londonmet.ac.uk) and then retrieve the recordset of database tables.

```
<%
DIM connString, conn
ConnString = "DRIVER={MySQL ODBC 3.51 Driver};
SERVER=www3.north.londonmet.ac.uk; PORT=3307;
DATABASE=dbname; UID=userid; PASSWORD=userpw; OPTION=3"

SET conn = Server.CreateObject("ADODB.Connection")
conn.Open(connString)

SET rs = Server.CreateObject("ADODB.recordset")
rs.open "SELECT * FROM table_name", conn
%>
```

The embedded SQL query, SELECT * FROM table_name, was to select every filed from the table_name table of the dbname MySQL database.

4.3.2 Data access using stored procedures

To evaluate the performance differences between using stored procedures and the traditional SQL approach, stored procedures were used to query data stored on the MySQL database on the Unix server (www3.north.londonmet.ac.uk).

A stored procedure is a collection of SQL statements - often with variables, looping, and branching - that are stored and processed as a unit on a database [25][26]. Stored procedures are compiled to be faster than embedded SQL. When a SQL statement is sent to the SQL database server, the query processor will parse it, analyse it, and create a query plan for execution. Not only does this slow down the queries, but it can create serious overhead when the number of users increases. Stored procedures are already compiled and their query plan resides on the server.

The following ASP scripts were used to call a stored procedure called sp_sp10(), which is shown in **Fig. 3**.

```
SET rs = Server.CreateObject("ADODB.recordset")
rs.open "call sp_sp10()", conn
```

The sp_sp10() stored procedure contains a SQL query SELECT * FROM table10, which selects every filed from the table10 table of the MySQL database.

```
-- ---------------------------
-- Procedure structure for sp_sp10
-- ---------------------------
DELIMITER ;;
CREATE DEFINER=`userid1`@`%` PROCEDURE `sp_sp10`()BEGIN
SELECT * FROM table10;
END;;
DELIMITER ;

-- ---------------------------
-- Procedure structure for sp_sp100
-- ---------------------------
DELIMITER ;;
CREATE DEFINER=`userid1`@`%` PROCEDURE `sp_sp100`()BEGIN
SELECT * FROM table100;
END;;
DELIMITER ;

-- ---------------------------
-- Procedure structure for sp_sp1000
-- ---------------------------
DELIMITER ;;
CREATE DEFINER=`userid1`@`%` PROCEDURE `sp_sp1000`()BEGIN
SELECT * FROM table1000;
END;;
DELIMITER ;
```

Fig. 3. Some stored procedures used.

4.4 Comparisons between Stored Procedures and the Traditional Approach

In order to examine the performance differences between the traditional way of executing a query on a MySQL database and the approach using stored procedures, several foundations need to be in place.

Five different database tables, called `table10`, `table100`, `table1000`, `table10000` and `table100000`, were created first on the MySQL database. The tables were filled with 10, 100, 1000, 10000 and 100000 records. All the tables have one attribute (field) called number. They were filled by writing a Stored Procedure and executing it within the MySQL console environment.

It was necessary to create several ASP web pages with the skeleton as shown in **Fig. 2**, which were used to display the query data within tables. The JS functions shown in **Fig. 1** were then used to measure the mean execution speed of retrieving and displaying the data by calculating the difference between the starting – point and the end – point of a certain transaction executed within the appropriated ASP page. The execution times were measured in milliseconds.

Table 1. Mean execution costs for different database tables

Database table	Number of rows	Mean execution costs (ms)	
		Traditional approach	Stored procedure approach
table10	10	16	47
table100	100	313	453
table1000	1,000	3,297	2,516
table10000	10,000	23,593	17,187
table100000	100,000	452,765	359,047

To obtain accurate and real values, five repeat performance measurements were carried out on the same user machine for each database table respectively. The measured execution costs in milliseconds were then averaged. Additionally the connection process within ASP was included into the measurements; however, since the database connection is identical in each test it is possible to disregard the time consumed for that matter. The performance results for five different database tables are listed in **Table 1** and shown in **Fig. 4**.
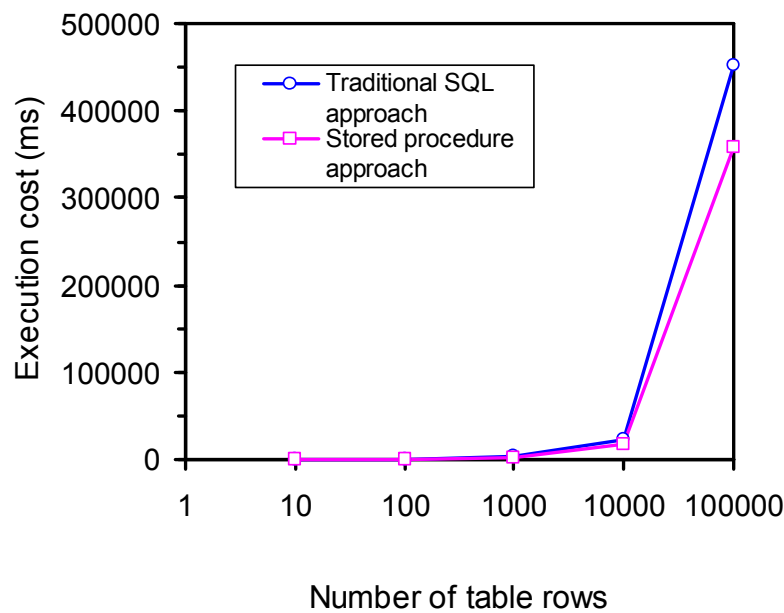

Fig. 4. Comparisons of performance between stored procedures and the traditional SQL approach.

Clearly the strength of stored procedures is noticeable. The data stored in the first two database tables are retrieved faster through the traditional SQL approach, i.e. low execution costs in milliseconds (ms), but as soon as the number of rows of the database table exceeding 10000 data rows, the data access using stored procedures is faster and therefore more efficient while dealing with the database tables with huge data inside such as `table100000` with 100000 rows. It is anticipated that the advantage of stored procedures should even be greater when more complex and nested queries or transaction commands are used [27].

As already discovered in earlier stages of this study, it is safe to mention that MySQL stored procedures are fast, useful and efficient since MySQL takes huge advantages of caching all executed queries. Possibly experiences with stored procedures may vary between different users, techniques and areas of application. The results discussed here are in connection with Active Server Pages in a web application environment based on a MySQL database.

4.5 Effect of ASP.NET

As many database driven web applications have migrated to a ASP.NET environment, it is necessary to evaluate the performance differences between data access over a MySQL database using stored procedures and the traditional SQL approach in a ASP.NET environment.
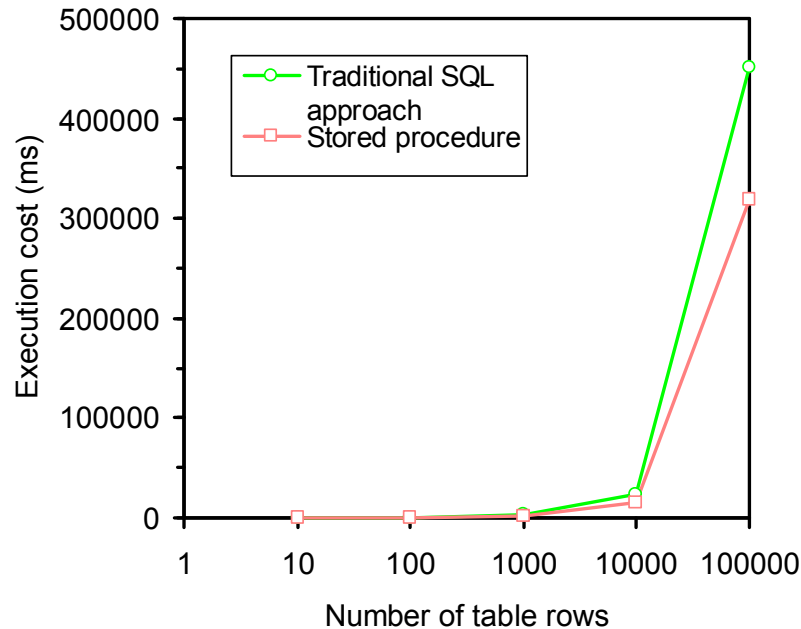
Fig. 5. Comparisons of performance between the traditional SQL approach and stored procedures in a .NET environment.

The performance results comparing the stored procedure approach and the traditional SQL approach for five database tables in a ASP.NET environment are shown in **Fig. 5** and listed in **Table 2**. For database tables with less number of rows, such as `table10` and `table100`, the execution times for retrieving and displaying all the data stored in the `number` field of the MySQL database using stored procedures were comparable to those when the traditional SQL approach was used. The benefits of stored procedures did not appear, which is probably due to the fact that it was still necessarily taken time for the stored procedures to be initialised in the ASP.NET environment.

Table 2. Performance results in a .NET environment

Database table	Number of rows	Mean execution costs (ms)	
		Traditional approach	Stored procedure approach
table10	10	18	40
table100	100	310	412
table1000	1,000	3,201	2,389
table10000	10,000	23,123	15,435
table100000	100,000	451,467	318,936

However, when the number of table rows increased, better performance in terms of the execution time was achieved with stored procedures. For the `table1000` table, the execution time taken reduced from 3201 milliseconds (ms) to 2389 ms. The reduction in the execution time for the `table10000` table was from 23123 ms to 15435 ms, achieving up to 33.2% performance improvement over the traditional approach. The execution for the `table100000` table speeded up by 29.4%, from 451467 ms to 318936 ms. Stored

procedures give a great improvement in the performance of data access on database tables with increasing numbers of queries being involved.
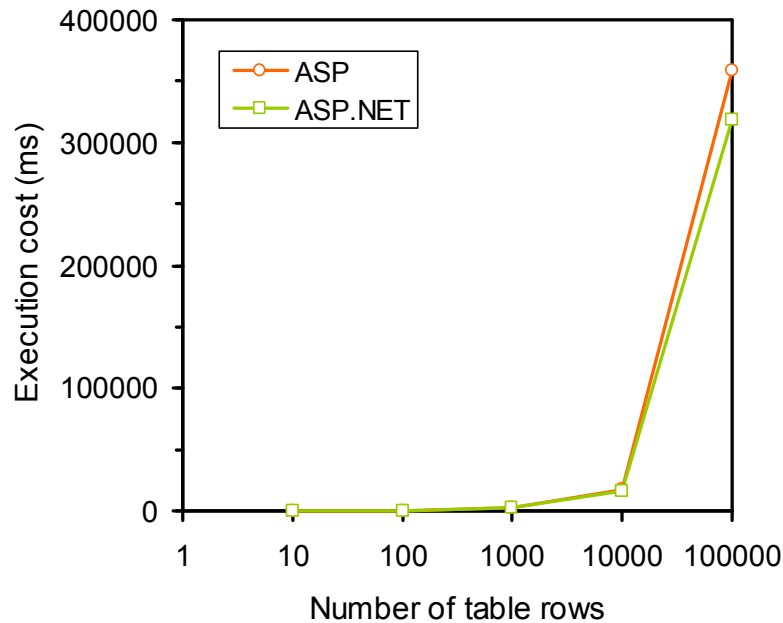

Fig. 6. Performance comparisons between ASP and ASP.NET.

Performance comparisons between ASP and ASP.NET when stored procedures being employed to query the MySQL database are shown in **Fig. 6**. The results indicate that ASP.NET gives slightly execution time improvements on the web application system. Less execution time was taken for the `table100000` table in the .NET environment, showing that the reduction in the execution time was remarkable with increasing numbers of queries. The better performance achieved in the ASP.NET environment could be attributed to the advantages of ASP.NET over ASP, such as integration and optimisation of web data access [26].

4.6 Performance Improvement with Stored Procedures

Stored procedures are just around for a short period of time in MySQL, however they are already proven within other Relational Database Management Systems (e.g. Microsoft SQL Server, Oracle and PostgreSQL), given that stored procedures are written programs executed and implemented within RDBMS. The possible complexity of stored procedures contributes to performance advantage of web applications (i.e. less execution cost), as the network traffic would be drastically reduced between the participating nodes (client and server) if stored procedures were employed.

It is possible to locate certain application logics (e.g. looping, checking, multiple transaction statements), which are not required user interaction into a stored procedure. Obviously if the logic is located within the RDBMS there will not be any communication between the server and the client during that period of time and would therefore reduce the required network capacity [24]. The entire feature is initiated through only one message from the client to the server, the so-called "call" (e.g. `call sp_procedure(parameters)`) [21][22]. It is the

call that can be specialised through the embedding of one or more parameters of different nature (e.g. in, out or inout parameters). Calling stored procedures is apparently much faster, comparing the amount of code transferred during the execution of several INSERT, UPDATE, DELETE or SELECT SQL statements, so it is noticeable that the transferred data are significantly larger than those while using a stored procedure.

Additional advantages of storage procedures are portability, migratebility as well as the independency of behaviour to the used application language. It is not relevant to which language is used (e.g. ASP, Java, PHP, etc.), given that significant logics are saved within the database instead of the accessing application. As a result the business – logic of the business can change without the necessity of altering the whole application. The term portability becomes important when the database is moved to a different server, since they are possible to be executed within every MySQL 5.0 environment. Even the move into a different Database Management System is comfortably possible with no or a little effort, depending on the target database system. For that reason stored procedures are implemented fairly close to the SQL: 2003 standard.

This case study shows that stored procedures did not help improve performance when small volumes of data are transferred from the server to a client. This is probably due to the fact that it is still necessarily taken time for the stored procedures to be activated / initialised in MySQL database Server. Once being activated, stored procedures are very cost effective when applications return large number of rows.

4.7 Implications for Web Application Design

As stored procedures are pre-compiled, it is expected that they would deliver better performance whatsoever. However, our experimental results have demonstrated that stored procedures' performance benefits can only be achieved under certain conditions that a substantial amount of information (e.g. >= 10,000 rows of data) is retrieved. This means compiled stored procedures should be used for retrieving data from large database tables with at least ten thousands rows of data; in contrast, normal SQL queries can be embedded in a ASP/ASP.NET, JSP, PHP or Perl web page to deal with data access to small database tables.

It is worth mentioning that, in the perspective of practical web programming, programmers normally have access to databases via server-side programming (e.g. ASP/ASP.NET, JSP, PHP or Perl), and the right to implement stored procedures on databases has to be granted by the server administrators.

4.8 Security Issues

Stored procedures offer other advantages over embedded SQL: easier maintenance and security [21]. Using stored procedures eliminates the tedious and inefficient task of formatting variables and SQL strings with single quotes, double quotes, and so on, so it can be read by the database.

Stored procedures improve security in two ways: Users can only access the stored procedure and not the underlying tables in a database; this means they can not access sensitive information which has to be stored with full SQL statements if not using stored procedures.

Furthermore, Steganography, one of the data hiding technologies [28][29], can be used to further improve security of data access in combination with the use of stored procedures for database query.

5. Conclusions

Our performance study from the user perspective, using specially designed JavaScript functions along with ASP scripts, demonstrates that stored procedures could give significant benefits of performance over the traditional SQL approach of querying a MySQL database when large volumes of data are transferred from the server to a client. The results also indicate that ASP.NET gives slightly execution time improvements on the web application system.

The measurement method presented in this paper is capable of measuring performance of database driven network applications. The method provides a way to measure performance form the user viewpoint, and can help enormously in making realistic assessment of true web site performance, and troubleshooting problems.

Our case study has shown that stored procedures' performance benefits are significant only when retrieving a large amount of information stored in MySQL database with ten thousands rows of data.

References

[1] C.S. Sohn, "Customer evaluation of Internet-based service quality and intention to re-use internet-based services," Dissertation, Southern Illinois University, IL: Carbondale, 2000.

[2] C. Liu, and K.P. Arnett, "Exploring the Factors Associated with Web Site Success in the Context of Electronic Commerce," *Information & Management*, Vol.38, No.1, pp.23-34, 2000.

[3] P.J. Trocchia, and S. Janda, "How do consumers evaluate Internet retail service quality?" *Journal of Services Marketing*, Vol.17, No.3, pp.243-253, 2003.

[4] V. McKinney, K. Yoon, and F.M. Zahedi, "The Measurement of Web-Customer Satisfaction: An Expectation and Disconfirmation Approach," *Information Systems Research*, Vol.13, No.3, pp.296-315, 2002.

[5] Z. Yang, R.T. Peterson, and L. Huang, "Taking the Pulse of Internet Pharmacies," *Marketing Health Services*, summer, pp.5-10, 2001.

[6] J.W. Palmer, "Web Site Usability, Design and Performance Metrics," *Information Systems Research*, Vol.13, No.2, pp.151-167, 2002.

[7] A.M. Aladwani, and P.C. Palvia, "Developing and validating an instrument for measuring user perceived web quality," *Information & Management*, Vol.39, No.6, pp.467-476, 2002.

[8] S. Ravi, "Measurement of User-Perceived Web Quality," in *Proc. of 12th European Conference on Information Systems* (ECIS), Turku, Finland, 2004.

[9] G. Banga, and P. Druschel, "Measuring the capacity of a Web server under realistic loads," *World Wide Web*, Vol.2, No.1-2, pp.69-83, 1999.

[10] S. Elbaum, G. Rothermel, S. Karre, and M. Fisher, "Leveraging User Session Data to Support Web Application Testing," *IEEE Trans. Software Eng.*, Vol. 31, No. 3, pp. 187-202, 2005.

[11] A. Andrews, J. Offutt, and R. Alexander, "Testing Web Applications by Modelling with FSMs," *Software and Systems Modelling*, Vol.4, No.3, pp.326-345, July 2005.

[12] M. Shams, D. Krishnamurthy, and B. Far, "A model-based approach for testing the performance of web applications," in *Proc. of the 3rd International Workshop on Software Quality Assurance*, pp.54-61, 2006.

[13] Edward W. Felten, and Michael A. Schneider, "Timing Attacks on Web Privacy," in *Proc. of 7th ACM Conference on Computer and Communications Security*, Nov. 2000.

[14] S. Jae Yang, Jason Nieh, and Naomi Novik, "Measuring Thin-Client Performance Using Slow-Motion Benchmarking," *ACM Transactions on Computer Systems*, Vol. 21, No.1, pp.87-115, 2003.

[15] A. Parasuraman, "Technology Readiness Index (TRI): A Multiple Item Scale to Measure Readiness to Embrace New Technologies," *Journal of Services Research*, Vol.2, No.4, pp.307-320, 2000.

[16] J. Song, and F.M. Zahedi, "Web Design in E-commerce: A theory and Empirical analysis," in *Proc.*

of Twenty-second International Conference on Information Systems, pp.205-219, 2001.

[17] R. Agarwal, and V. Venkatesh, "Assessing a Firm's Web Presence: A heuristic Evaluation Procedure for the Measurement of Usability," *Information Systems Research*, Vol.13, No.2, pp.168-186, 2002.

[18] J. Cox, and B.G. Dale, "Key quality factors in Web site design and use: an examination," *International Journal of Quality & Reliability Management*, Vol.19, No.7, pp.862-888, 2002.

[19] T.M. Connolly, and C.E. Begg, *Database Systems*, Pearson Education, 2005, ISBN 0321210255.

[20] MySQL web site. Http://www.mysql.net/.

[21] Paul DuBois, *MySQL Cookbook*, O'Reilly Media, Inc., ISBN: 059652708X.

[22] Guy Harrison, and Steven Feuerstein, *MySQL Stored Procedures Programming*, O'Reilly Media, Inc., 2005.

[23] Russell Dyer, *MySQL in a Nutshell*, O'Reilly Media, Inc., ISBN: 0596007892.

[24] Jeremy Zawodny, and Derek Balling, *High Performance MySQL*, ISBN: 0596003064.

[25] S. Tang, W. Lu, and F. Marir, "Optimisation of interactive web data access," in *Proc. of International Conference on eCommerce Engineering*, pp.16-21, 2001. ISBN 7-900066-54-3.

[26] S. Tang, T. Wang, and Y. Huang, "Evaluation of Web Data Access for an Enterprise Web Application Built on a .NET Platform," in *Proc. of 2nd International Conference on Internet Technologies and Application*, 2007. ISBN: 978-0-9468-8154-3.

[27] P. Roy, S. Seshadri, S. Sudarshan and S. Bhobe, "Efficient and Extensible Algorithms for Multi Query Optimisation," in *Proc. of the SIGMOD' 2000 International Conference*, Dallas, 2000.

[28] Y. Huang, K. Zhou, Z. H. Wang, "Multiple Data Channel Data Hiding Mechanisms based on VOIP," *Journal of Harbin Industrial University*, pp.38, 2006.

[29] Y. Huang, S. Tang, and Y. J. Yip, "A New Security Architecture for SIP Based P2P Computer Networks," *Journal of Computer Science, Informatics and Electrical Engineering*, Vol.2, No.1, pp.63-73, 2008.

Shanyu Tang is a senior lecturer in informatics and multimedia in the Faculty of Computing at London Metropolitan University (UK) where he teaches web programming and information security. His current research focuses on steganography, trusted communications, and fractal data modelling. He is the principal grant holder of two external funded research projects. He has a BSc degree in statistics and a PhD, his PhD was awarded by Imperial College London in 1996, and his doctorial study was supported by an ORS Award and a Hong Kong Scholarship. He has over 30 referred publications on web security, groupware and fractal data modelling, and is a member of IEEE professional society.

Huang YongFeng is an associate professor in the Department of Electronic Engineering, Tsinghua University, Beijing. He received his PhD in computer science and engineering from Huazhong University of Science and Technology in 2000. His research interests include VoIP, P2P IP TV, multimedia network and next generation Internet. He has published five books and over 50 research papers on computer network and multimedia communication. Dr Huang, as one of the principal researchers, has designed and constructed China Education and Research Network (CERNET), which is the second largest computer network in China. He is the principal/joint grant holder of 8 external funded research projects.

Yau Jim Yip Eur Ing Professor Yau Jim Yip, BSc PhD CITP FBCS CEng FIMechE. Professor Yip is Dean of School of Computing and Engineering, University of Huddersfield, UK. His research areas are Knowledge Based Systems, Knowledge Engineering, Knowledge Management and Artificial Intelligence. Funded projects include applications in chemical and process industry, textile, manufacturing and construction industry, geographical applications and environmental management. He is the UK Co-Director of the Web Intelligence Consortium and serves on conference committees. He is the principal/joint grant holder of external funded research projects worth £1.3m with 92 publications. He has supervised 34 research fellows and students with 9 PhD/MPhil completions. He is an EPSRC proposal assessor, EU registered expert and Panel Judge on Esprit/Framework research proposals. He has industrial consultancy and collaboration with 14 companies in 23 projects. He has been external examiner for taught and research degrees and the consultant at 19 UK and overseas universities.