DATA MANAGEMENT SYSTEM EVALUATION FOR MOBILE MESSAGING SERVICES

David C. C. Ong, Rytis Sileika, Souheil Khaddaj
Faculty of Computing, Information Systems and Mathematics, Kingston University, Kingston upon Thames KT1 2EE, UK.

Radouane Oudrhiri
Systonomy Ltd, 2nd Floor , Berkeley Square House, Berkeley Square, London W1J 6BD, UK

Keywords: Database management systems, Database reliability, Mobile communication.

Abstract: A mobile messaging revolution for the mobile phone industry started with the introduction of the Short Messaging Service (SMS), which is limited to 160 characters of conventional text. This revolution has become more significant with the additional improvements in mobile devices. They have become relatively powerful with extra resources such as additional memory capacity and innovative features such as colour screen, photo camera, etc. Now Multimedia Messaging Service (MMS) takes full advantage of these capabilities by providing longer messages with embedded sound, image and video streaming. This service presents a new challenge to mobile platform architects particularly in the data management area where the size of each MMS message could be up to 100,000 bytes long. This combined with a high volume of requests managed by these platforms which may well exceeded 250,000 requests per second, means that the need to evaluate competing data management systems has become essential. This paper presents an evaluation of SMS and MMS platforms using different data management systems and recommends the best data management strategies for these platforms.

1 INTRODUCTION

Defining and creating a high performance mobile messaging platform is the main mission for today's mobile application architects. Designing a more efficient messaging protocol or platform code often has improved the overall performance of the platform, but major problems relating to the efficiency of use of those data management systems still remain. Although, there are various benchmarks and test results available (Bourke, 2005) (Sleepycat Software, 2005) (Dyck, 2002), more investigation is still required in order to produce a convincing and precise evaluation model in terms of consistency and throughput of data over a given period of time. Also, there is both a lack of accurate representation of distributions in the actual system operations and credible assessment of those non-complex data structures, which resemble mobile messaging data structures, as most of them tend to concentrates on the performance of the large database where the life span of the data store is long and huge compare with the mobile data. There is also tendency toward popular areas such as On-Line Transaction Processing and Web Database Application (Vieira, 2003), Extreme Database Test (Ercegovac, 2005) (Bourke, 2005) (Dyck, 2002), etc., which do not satisfy the criteria required for the mobile data management systems.

Recently Sleepycat Software Inc. has published a white paper entitled "Managing Data within Billing, Mediation and Rating System" (Sleepycat Software, 2005), this database performance model deals only with a small test sample, thus there were no statistically acceptable results to support the claim of the superiority of the chosen database.

The data structure of the mobile messaging platforms is simple, consisting of one or two tables with an estimated three columns each. It is considered unproductive and expensive to recreate a custom designed data management system, as current available data management systems have reached maturity in terms of their robustness and

C. C. Ong D., Sileika R., Khaddaj S. and Oudrhiri R. (2006).
DATA MANAGEMENT SYSTEM EVALUATION FOR MOBILE MESSAGING SERVICES.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - DISI*, pages 226-231
DOI: 10.5220/0002490402260231

efficiency in handling and storing data. In view of this, most of the mobile messaging platforms use a data management system such as a DBMS to handle data manipulation and storage of the platform.

This paper aims to investigate and evaluate several database management systems that are commonly used for mobile messaging services. In the quest to find the best data management strategies for mobile messaging platforms, a critical evaluation was carried out to assess current available data management systems. This evaluation focused on the SMS and MMS platforms by observing performance and quality of service in the message handling under minimum and maximum workload, where data size remains consistent throughout the evaluation period.

In this work we started by considering a number of data management systems together with the software and hardware environment. Then, an evaluation framework was constructed together with some initial evaluation of the various platforms. Several real time experiments using SMS and MMS data structures were then conducted. Finally, we present some conclusions and suggestions for future works.

2 DATA MANAGEMENT SYSTEMS

2.1 Classification

It was considered unproductive to evaluate all of the data management systems available, as this would require huge resources and time, so data management systems were categorised into three main classes based on the storage engine provided. There are Main Memory based DBMS, Disk based DBMS and Disk based data store without DBMS. DBMS products such as Oracle, MySQL, TimesTen, Microsoft SQL Server, MaxDB, etc. have offered various data management solutions under these classifications.

Main Memory based DBMS manages, caches and stores data in the main memory (Altýnel, 2003). It is the fastest in transaction management and storage but the data stored will be completely lost in the event of a system failure. DBMS products such as TimesTen or MySQL also utilise a main memory engine for their data store engine.

Disk based DBMS provides a persistent storage to disk. It uses data cached in a memory buffer to handle transaction before storage to disk (Altýnel, 2003). It is considered safe because data that have been successfully stored to the disk will not be lost in case of system failure. Disk based DBMS like Microsoft SQL Server and Oracle incorporates persistent data storage engines to provide a persistent storage solution. Unlike other established DBMS products, MySQL utilises third party data storage engines besides its own (e.g. InnoDB and BerkeleyDB) for the persistent storage solution.

Functionalities provided in DBMS are often considered a waste of resources if the data structures like mobile message services are simple and small. In Disk based DBMSs, data storage engines (e.g. InnoDB, BerkeleyDB, etc) are used to store and retrieve data from the disk. Hence, Disk based data store without DBMS could be considered a solution for managing a simple mobile messaging data structure.

2.2 Database Selection

For each classification, the database that was considered the best for this evaluation was selected based on cost, efficiency, performance, reliability, popularity and their general availability.

In the mobile messaging industry, operators cannot afford data loss in the event of system failure. TimesTen was selected to represent a Main Memory based DBMS (Team, 2002) because it provides an automatic secondary persistent data management disk solution. In the event of system failure, data loss will be minimal.

There are various Disk based DBMS products on the market. These products have gained popularity due to their high reliability and integrity, as there is a very small amount of data loss in the event of system failure. Popularity of MySQL in the data management solutions has grown recently (Red Herring, 2005) (Jupitermedia, 2005) because it offers a cheaper alternative to the established Disk based DBMS product such as Oracle and Sybase. MySQL with MyISAM engine was therefore recommended to represent the Disk based DBMS class.

BerkeleyDB is a transaction safe storage engine with a page locking facility. It is viewed as safest as a data storage solution, as it only requires minimal processing overhead before data is safely stored. It was therefore chosen in the Disk based DBMS (i.e. MySQL) to provide a transaction safe solution to meet the data management requirement. It is considered easy to fit onto practically any data management system and efficient in handling simple data structures. Recently several of the established IT corporations such as Amazon.com, Cisco Systems, Google, Teligent, Mitel, Motorola and Sun Microsystems have looked into BerkeleyDB for their data management solutions (Hedlund, 2003) (Okmianski, 2003) (Anzarouth, 2003). For these reasons, BerkeleyDB has been chosen to represent the Disk based data store without DBMS class.

3 CONSTRUCTION OF EVALUATION FRAMEWORK & PLATFORM

3.1 Hardware and Software Environment

Performance may depend on the server and database environment configuration and specification. Evaluation was conducted in a fixed server environment, where databases involved in the evaluation were installed. A cluster with 2 x 2.4GHz (Intel Xeon HT) processor, 1.5GB of RAM, 36GB U-SCSI 10k internal HDD and a disk array (3 x 4 disk volumes (RAID 1+0)) 10k for the shared HDD, which housed Linux RedHat ES 3 with the 2.4 kernel as the operating system was used as the base evaluation platform. Java 2 Platform, Standard Edition (J2SE) 1.5 was adopted to execute the evaluation model.

It is acknowledged that the performance of the Main Memory based DBMS or Disk based DBMS may improve dramatically simply by the installation of more memory or faster HDD in the server. For the purpose of this evaluation, basic hardware and software specifications had to be fixed to ensure that an even assessment could be performed. Optimisation of individual databases was done to the best of our knowledge and experience.

3.2 Data Structure

The data structure for SMS consists of a single table, which has 10 bytes of index and 1,200 bytes for data. This platform gives a simple and easy way to handle and store conventional short messages. In order to handle long messages embedded with audio, graphics, video and data, MMS was devised. MMS has two tables, one of which is designed for the index and metadata of the message and the other was created for storing the actual data of the message. The index table consisted of 10 bytes of index and 1,200 bytes of metadata. The data table for the MMS has 10 bytes of index and 100,000 bytes of message data.

The mixture of tasks is based on the proportional distribution of tasks in the actual system observed in the SMS and MMS platforms. The evaluation model would replicate the actual distribution of these tasks. This is to ensure the model is able to mimic the processes in the actual platform environment.

Each process in the SMS evaluation model consists of a number of tasks; typically, it has 4 insertion, 12 selection, 1 updating and 4 deletion tasks of the SMS data. The index table for the MMS has a similar proportion of tasks as described in the SMS. The proportion of tasks for the data table consisting of 4 insertion, 8 selection and 4 deletion tasks was adopted. Distribution tasks for SMS and MMS platforms presented above were based on the functional requirements described in the ETSI Standard titled “Technical realization of the Short Message Service” for the GSM 3.40 (ETSI, 2003).

3.3 Evaluation Framework

The purpose of this evaluation was not to test the system under extreme conditions, to breaking point, but rather to evaluate an optimal level of performance for the system when it reaches a consistent level. The challenge was to control the state of the database during testing and to order the test runs in such a way that a measurable figure could be observed at the same time maintaining an optimal operation state (Haftmann, 2005). Thus, the evaluation framework must achieve three identified aims to fulfil this purpose. These are: to define and achieve a consistent level in the system before any measurement is taken; to reach and maintain maximum data throughput to the database; and finally to maintain the same mixture of tasks executed and at the same time ensuring randomness of the tasks.

A single test framework was not considered a justifiable evaluation, as a maximum throughput of the database and true randomness in the mixture of the tasks could not thus be achieved. Multi-threading therefore was introduced to the evaluation system to ensure maximum throughput of the database and to ensure that randomness was achieved. Four threads were therefore introduced to the framework. Without such multi-threading, measurements taken from repeated evaluations would be invariable. Conversely, introducing the threading in the system meant that there could be only slight variation in the measurements taken should the same evaluation be repeated. Variation would exist mainly due to the randomness of multiple task requests created by the threading. There is a difference in time of the transitional period from one task to another. The difference in task combinations in this random environment contributes to the overall differences in the measurement. Observations could be based on these variations, when performance of the database might be considered inconsistent and unstable if huge differences among these variations were to be observed.

The examination of the evaluation result is based on the ten tests carried out for each of the test categories. The speed of the database to execute requested tasks is measured based on the average time it takes to complete the 10 tests. The sequence of tasks executed in each of the 10 tests is different due to the randomness introduced into the framework by the multi-threading. The standard deviation is taken for the 10 tests conducted to measure the level of variation of the performance of the database under different sequences of the tasks to check for the database consistency. If the standard deviation is high, we may conclude that consistency and reliability of the database is low, as the performance will have varied over the different mixture of tasks. Conversely, if the standard deviation is low, we may conclude that consistency and reliability of the database is high.

Table 1: Low Volume SMS Test, Summary of 10 Test Results.

	MySQL	TimesTen	BerkeleyDB
Total number of tasks / test	1,050,000	1,050,000	1,050,000
Average number of tasks / millisecond executed in 10 tests	12	84	23
Time spent / test (minute)	1:24	0:12	0:45
Standard deviation, σ	84	3,476	2,951

Table 2: Medium Volume SMS Test, Summary of 10 Test Results.

	MySQL	TimesTen	BerkeleyDB
Total number of tasks / test	2,100,000	2,100,000	2,100,000
Average number of tasks / millisecond of 10 tests	12	89	27
Time spent / test (minute)	2:50	0:23	1:16
Standard deviation, σ	80	2,738	2,870

Table 3: High Volume SMS Test, Summary of 10 Test Results.

	MySQL	TimesTen	BerkeleyDB
Total number of tasks / test	4,200,000	4,200,000	4,200,000
Average number of tasks / millisecond executed in 10 tests	12	88	10
Time spent / test (minute)	5:48	0:47	6:33
Standard deviation, σ	160	2,000	158

4 SMS AND MMS EXPERIMENTS

Each experiment was divided into three critical areas, based on the different volume of traffic. There are; Low volume test to observe performance of the databases in low levels of data intensity and database size; Medium volume test to examine performance of the databases when it handles medium volumes of data intensity and database size; and High volume test to study databases performance under high data intensity and database size. These experiments are based on a proportion of distribution of tasks described in section 3.2.

4.1 Evaluation of the SMS Platform

Throughout this evaluation, data sizes inserted into the database were the maximum allowed by the length of each column. Tables 1, 2 and 3 show the summary results of 10 concurrent tests conducted for each database under different classifications of data intensity. 1,050,000 task requests were sent to databases for the low volume test, 2,100,000 tasks for the medium volume test and 4,200,000 requests were executed for the high volume test.

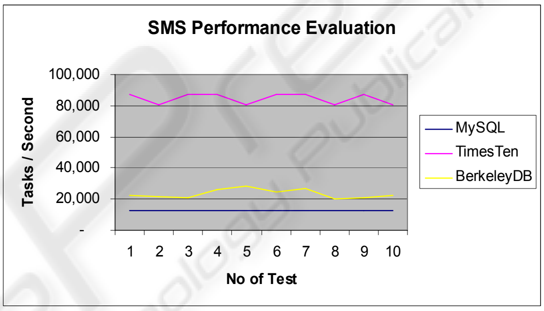

Figure 1: Performance evaluation for Low Volume SMS Test.

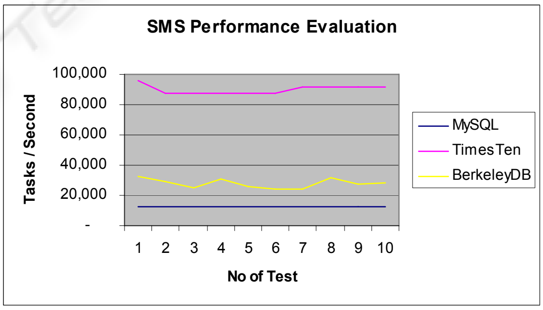

Figure 2: Performance evaluation for Medium Volume SMS Test.

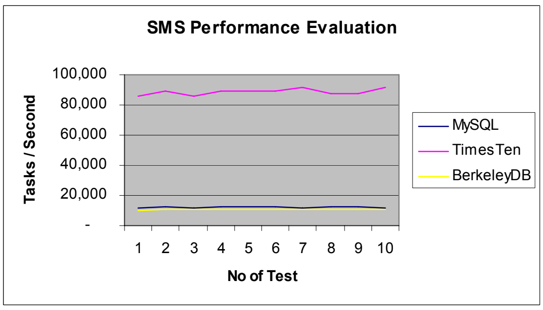

Figure 3: Performance evaluation for High Volume SMS Test.

Based on the result shown in figures 1, 2 and 3, performance of BerkeleyDB degraded dramatically as the volume of tasks increased. There is a reliability issue for BerkeleyDB and TimesTen as the standard deviation for each test is high. But, TimesTen has overcome this problem by delivering a far superior service compared with other databases. Going on this evidence, Main Memory based DBMS is considered the most desirable choice as the data management strategy for SMS services, provided it has a third party or its own plug-in secondary persistent storage.

It is acknowledged that the standard deviation score is insignificant in relation to the high number of tasks executed by databases but it is enough to carry some weight when the time taken to complete the tasks among the different databases is considered.

4.2 Evaluation of the MMS Platform

The MMS platform consists of two tables, the index and the data tables. It is considered a good strategy to find various database combinations in search of the best solution to manage index and data tables for the MMS platform, rather than just stick to one type of the database for both tables. The SMS evaluation observations from section 4.1 are considered essential in this selection process.

Table 4: Low Volume MMS Test, Summary of 10 Test Results.

Index / *Data*	MySQL MySQL	TimesTen MySQL	MySQL BerkeleyDB	TimesTen BerkeleyDB
Total number of tasks / test	370,000	370,000	370,000	370,000
Average number of tasks / millisecond executed in 10 tests	4	5	5	7
Time spent / test (minute)	1:26	1:18	1:11	0:54
Standard deviation, σ	119	43	223	631

Table 5: Medium Volume MMS Test, Summary of 10 Test Results.

Index / *Data*	MySQL MySQL	TimesTen MySQL	MySQL BerkeleyDB	TimesTen BerkeleyDB
Total number of tasks / test	1,110,000	1,110,000	1,110,000	1,110,000
Average number of tasks / millisecond executed in 10 tests	4	4	4	4
Time spent / test (minute)	5:07	4:14	5:23	4:40
Standard deviation, σ	140	386	1,021	1,415

Table 6: High Volume MMS Test, Summary of 10 Test Results.

Index / *Data*	MySQL MySQL	TimesTen MySQL	MySQL BerkeleyDB	TimesTen BerkeleyDB
Total number of tasks / test	1,850,000	1,850,000	1,850,000	1,850,000
Average number of tasks / millisecond executed in 10 tests	3	3	2	2
Time spent / test (minute)	10:44	9:38	17:57	19:07
Standard deviation, σ	178	296	46	103

Main Memory based DBMS is considered unsuitable for the data table, owing to the limitation of the hardware (i.e. RAM) to store such a huge amount of data. Main Memory based DBMS or Disk based DBMS for index table will be combined with Disk based DBMS or Disk based data store without DBMS.

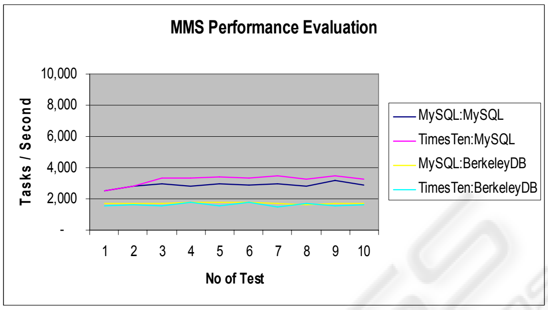

Figure 4: Performance evaluation for Low Volume MMS Test.

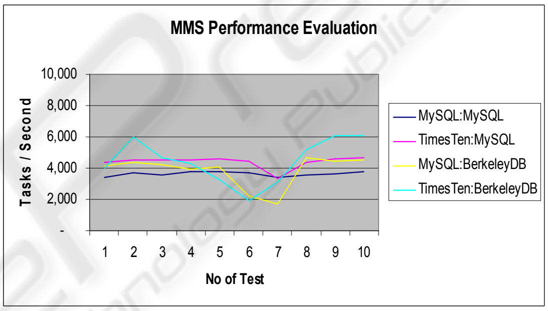

Figure 5: Performance evaluation for Medium Volume MMS Test.

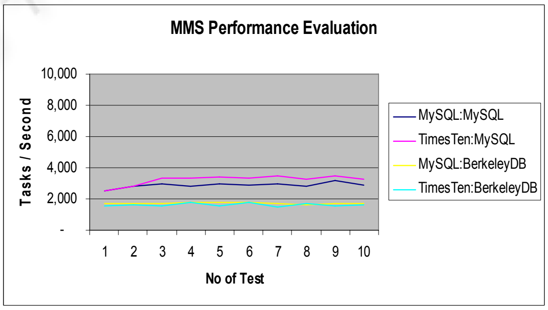

Figure 6: Performance evaluation for High Volume MMS Test.

There are four viable combinations to consider, which are; MySQL for both index and data tables; TimesTen for the index table with MySQL for the data table; MySQL for the index table with BerkeleyDB for the data table; and finally TimesTen for the index table with BerkeleyDB for the data table.

As the message size (i.e. 100,000 bytes) to be inserted was not viable for the evaluation platform, data size inserted into the database for this table was reduced to 30,000 bytes.

A table 4, 5 and 6 shows the summary of 10 concurrent tests with the results conducted for each database under different classifications of data intensity. 370,000 tasks request were sent to databases for the low volume test, 1,100,000 tasks for the medium volume test and 1,850,000 requests were executed for the high volume test.

Based on the results shown in figures 4, 5 and 6, the implementation of MySQL as a data table is seen as faster than using BerkeleyDB. Although both MySQL and TimesTen managed to execute almost the same number of tasks as an index table, TimesTen is the fastest in executing all the tasks both using MySQL and BerkeleyDB as the data table, when compared with MySQL. Thus, using Main Memory based DBMS as an index table and Disk based DBMS as a data table is the most desirable combination data management strategy for MMS services, but it must be kept in mind that using two different databases as a solution is not always a wise choice.

It is acknowledged that the standard deviation score is insignificant in relation to the high number of tasks executed by databases but it is enough to carry some weight when the time taken to complete the tasks among the different databases is considered.

5 CONCLUSION AND FUTURE WORK

It is the intention of this paper to provide various viable data management strategies for mobile messaging platforms. The observations produced from the various tests could be viewed as a guideline in selecting the best data management strategies that meet this design requirement. Recommendations given in this paper are aimed at high performance systems, which may not be valid in other circumstances. New proposals therefore should be made based on the result of these evaluations in order to meet any new system design requirement.

Selection of the data management platform often depends on the customer. For the customer, cost is often a top priority in the selection process. Main Memory based DBMS are best in term of performance but not pricing, it is expensive to upgrade the memory in the system and license fee costs are high. At the other end of the scale, BerkeleyDB license fees cost less compared with those of other database licenses. Upgrading the disk to a high specification HDD is a cheap option that may solve the performance issue with BerkeleyDB and Disk based DBMS. The customer may not always need a high performance data management system and may be more concerned with the consistency, reliability and integrity of the system. Disk based DBMS seen to present the best choice for this requirement.

Regarding the prospect of advancing mobile technologies, further review of the data management strategies should be conducted with consideration given to live video streaming for the mobile devices and clustering solutions for data management systems.

REFERENCES

Ercegovac, V., 2005. The TEXTURE Benchmark: Measuring Performance of Text Queries on a Relational DBMS. Proceedings of the 31st VLDB Conference, Trondheim, Norway.

Haftmann, F., 2005. Parallel Execution of Test Runs for Database Application Systems. Proceedings of the 31st VLDB Conference, Trondheim, Norway.

Vieira, M., 2003. A Dependability Benchmark for OLTP Application Environments. Proceedings of the 29th VLDB Conference, Berlin.

Altýnel, M., 2003. Cache Tables: Paving the Way for an Adaptive Database Cache. Proceedings of the 29th VLDB Conference, Berlin.

Bourke, T., 2005. Using MySQL to benchmark OS performance. NewsForge, OSTG Inc., <http://software.newsforge.com/software/04/12/27/1238216.shtml?tid=152&tid=72&tid=29>.

Team, TimesTen., 2002. Mid-tier Caching: The FrontTier Approach, SIGMOD’02, Madison.

Sleepycat Software, Inc., 2005. Managing Data within Billing, Mediation and Rating Systems, Sleepycat Software Inc.

Dyck, T, 2002. Server Databases Clash, eWEEK. Ziff Davis Publishing Holdings Inc. <http://www.eweek.com/article2/0,4149,293,00.asp>.

Hedlund, N., 2003. Berkeley DB Transactional Data Store – Teligent, Sleepycat Software Inc. (Hedlund, 2003)

Jupitermedia, Corporation., 2005. Developer.com Product of the Year 2004 Results, Jupitermedia Corporation. <http://www.developer.com/lang/other/print .php/3307861>.

Okmianski, A., 2003. Berkeley DB Transactional Data Store – Cisco, Sleepycat Software Inc.

Anzarouth, R., 2003. Berkeley DB Transactional Data Store – Mitel, Sleepycat Software Inc.

Red Herring, Inc., 2005. RH-100 Europe: The Ikea of Databases, Red Herring Inc. <http://www.redherring.com/Article.aspx?a=11919&hed=RH-100+Euro pe%3a+The+Ikea+of+Databases>.

ETSI, 2003. Technical realization of the Short Message Service, GSM 3.40 Vs. 3.5.0. European Telecommunications Standards Institute.