

In-Memory Data Management for Consumer Transactions

The TimesTen Approach

The TimesTen Team
TimesTen Performance Software
2085 Landings Drive
Mountain View, CA 94043
(650) 526-5100
Info@timesten.com

1. INTRODUCTION

TimesTen Performance Software is a company dedicated to high-performance data management. TimesTen™, the company's flagship product, is an in-memory relational database.

The concept of managing an entire database in main memory has been researched and studied for over fifteen years [1-7], and the benefits of such an approach have been well-understood in certain domains, such as telecommunications, for at least as long. But it is only recently, with decreasing memory prices and with the availability of 64-bit operating systems, that the size restrictions on in-memory databases have been removed. In-memory data management has become available for many more applications that are in dire need of high performance.

This paper examines the data management requirements of applications targeted at consumers and how TimesTen addresses these requirements.

2. CONSUMER TRANSACTIONS

Consumer transactions appear in a variety of applications from diverse domains. The common thread between them is that the end user is a consumer with certain expectations about response time and quality of service, and the service provider is in a highly competitive field where consumer satisfaction is essential to success. Examples abound: in the field of telecommunications, the consumer dials a phone number expecting a connection within seconds. For web-based searches, the consumer expects reasonable responses to somewhat fuzzy inquiries within seconds. For web-based trading, the consumer expects accurate, up-to-date quotes in a timely manner. For electronic commerce, the consumer expects the completion of a transaction within seconds. These expectations are independent of the amount of processing, analysis, mappings, or customization that must take place in the background.

Successful service providers must offer continuously available

and properly targeted services, timely responses even during peak loads, the ability to rapidly deploy new services, and the ability to capture and analyze consumer access patterns and affinities without degrading service.

At the data management level, these requirements translate into response times under 10 milliseconds for simple requests, transaction rates on the order of 10,000 transactions per second for simple transactions, the ability to handle hundreds of simultaneous users, and high availability. We briefly cover below how TimesTen addresses these requirements.

3. BASICS

TimesTen is an in-memory relational database [8] that supports SQL [9] through the ODBC and JDBC APIs. TimesTen databases are persistent and recoverable. Durability is ensured through checkpointing and logging. Applications may choose ACID properties for their transactions, but more relaxed options are also available for higher performance. SQL statements are optimized by TimesTen's cost-based query optimizer and applications may view and influence query plans. The system is available as a library that may be linked by applications as well as through a client/server option. High availability is provided through replication. A number of utilities are also available, including on-line backup and restore, bulk loading, and data migration between different versions of the system.

Through the support of SQL and of standard APIs, TimesTen enables service providers to deploy new, high-performance, data-intensive services rapidly. Adherence to standards reduces development time and permits interoperability of the system with existing tools.

4. PERFORMANCE

To measure the performance of the system, we developed a multi-user benchmark consisting of simple ODBC read or update transactions. Each *read transaction* issues three ODBC calls: one to execute a precompiled SQL SELECT statement, one to fetch the result, and one to commit the transaction. Each *update transaction* issues two ODBC calls: one to execute a precompiled SQL UPDATE statement and one to commit the transaction. Each transaction is run with both logging and row-level locking enabled.

Experiments were run on a 4-CPU Compaq ProLiant Xeon-based (400 MHz) system running Windows NT 4.0. With a mix consisting of 80% read transactions and 20% update transactions, and with 32 simultaneous users, throughput was measured at

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMOD '99 Philadelphia PA

Copyright ACM 1999 1-58113-084-8/99/05...\$5.00

1,050 transactions per second (TPS) when the log buffer is flushed to disk on commit and at 17,000 TPS when the log buffer is flushed to disk only when full. The latter is one of the options available in TimesTen. It guarantees atomicity of transactions and durability of all but the last few transactions, and is suitable for a certain class of applications. The average response time for a read transaction was 63 μ secs and the average response time for an update transaction was 112 μ secs for transactions that flush the log buffer to disk only when the buffer is full.

5. HIGH AVAILABILITY

For consumer transactions, high availability not only means that service must always be up, but that it is also available for the consumer when the consumer wants it, and at the level of performance that the consumer expects. Thus, maintenance operations such as database backups, administrative operations such as altering the definition of a table, compacting a database or computing statistics, and periodic operations such as database checkpoints cannot block access to the database or any portion of the database for any consumer-noticeable duration.

TimesTen addresses these issues through a number of features and utilities. First, replication is offered to guarantee system availability even in the event of software or hardware failures. Replication, coupled with TimesTen's migration utility, may be used to perform on-line software upgrades. Second, the on-line backup utility does not interfere with access to databases. Internal work required by time-consuming DDLs such as ALTER TABLE is deferred until needed and executed piecemeal. Finally, some operations are made available to the application through *time-sliced* calls where each call executes a time slice of the work, and where the work resumes on subsequent calls.

6. CONSUMER TRANSACTIONS VERSUS BUSINESS TRANSACTIONS

For the service provider, there are two aspects to running a successful business: consumer satisfaction and intelligent business management. Consumer satisfaction mandates timely, accurate and relevant responses to consumer transactions. Running a business intelligently often means analysis of events, tracking and storing access patterns, recording changes to relevant information, tracking the patterns of the changes, etc. It also often means feeding back the results of analysis into consumer transactions through techniques such as customization and targeted advertisement.

TimesTen's approach to meeting consumer satisfaction while not compromising business management is to divide the task of managing data between an in-memory DBMS and a disk-based DBMS. By partitioning data management between the application tier and the database server, the overall performance of the system improves as the database server is offloaded from some of the data management work and especially from responding to a large number of simple requests. These requests create a heavy burden of network traffic and CPU usage.

The disk-based DBMS runs in the database server. It is the permanent repository of all the business data. It performs analytical and complex transactions.

The in-memory DBMS runs in the application tier, preferably linked with the application. It provides the high performance associated with managing data in main memory and avoids network communication and context switches. Yet, it presents familiar, standard interfaces and the declarative advantages of SQL to the application. It caters to consumer transactions, and when it assists in managing the business (e.g., by capturing access patterns, creating web site personalization, or detecting fraudulent behavior), it does not degrade the performance of consumer transactions. It maintains a working set of the database and interoperates with the disk-based DBMS to perform operations such as batching inserts of captured data, propagating updates to cached data, or altering the content of its working set.

The TimesTen in-memory database is ideally suited to fill the high-performance data management needs of applications targeted at consumer transactions. Through its standard interfaces, it permits the rapid development of new services, and through its high performance, it ensures consumer satisfaction. TimesTen's first product was released at the beginning of 1998. Smallbase, the Hewlett-Packard Laboratories predecessor to TimesTen, has been shipping, embedded in Hewlett-Packard's products, since 1996.

7. REFERENCES

- [1] DeWitt, D.J., et al. Implementation Techniques for Main Memory Database Systems. *Proc ACM SIGMOD Conf*, Boston, MA, June 1984.
- [2] Eich, M.H. A Classification and Comparison of Main Memory Database Recovery Techniques. *Proc 3rd Int Conf on Data Engineering*, Los Angeles, CA, February 1987.
- [3] Garcia-Molina, H. and Salem, K. High Performance Transaction Processing with Memory Resident Data. *Proc Int Workshop on High Performance Transaction Systems*, Paris, December 1987.
- [4] Garcia-Molina, H. and Salem, K. Main Memory Database Systems: An Overview. *IEEE Trans on Knowledge and Data Engineering*, vol 4, no 6, December 1992.
- [5] Jagadish, H.V., et al. Dali: A High Performance Main Memory Storage Manager. *Proc VLDB Conf*, Santiago, Chile, September 1994.
- [6] Lehman, T.J., and Carey, M.J. A Study of Index Structures for Main Memory Database Management Systems. *Proc VLDB Conf*, Kyoto, August 1986.
- [7] Lehman, T.J., Shekita, E.J., and Cabrera, L.F. An Evaluation of Starburst's Memory Resident Storage Component. *IEEE Trans on Knowledge and Data Engineering*, vol 4, no 6, December 1992.
- [8] *TimesTen Programmer's Guide*. TimesTen Performance Software. <http://www.timesten.com>
- [9] *TimesTen SQL Reference Manual*. TimesTen Performance Software. <http://www.timesten.com>