



WHITE PAPER

Memory-Optimized Transactions and Analytics in One Platform: Achieving Business Agility with Oracle Database

Sponsored by: Oracle Corp.

Carl W. Olofson
July 2015

IN THIS WHITE PAPER

This white paper examines the need for enterprises to move beyond the static model that limits business intelligence to the accumulation of operational data into a data warehouse or an operational data store (ODS) and the use of the resulting analytics to make decisions based on data that may be days or weeks old. The more dynamic model involves including current transactional data in analytical queries and using the results to make decisions during the business day, when they can have a more timely impact. To do this, one must have a database management system (DBMS) capable of executing complex analytical queries against live transaction data, without impacting the operational applications that are handling that data. This white paper considers the business requirements for such a DBMS, the technical characteristics of such a system, and the benefits that result from the system's use. This white paper also examines the In-Memory Option in Oracle Database as a technology to consider in this regard.

SITUATION OVERVIEW

The Challenge: To Stop Managing by Guesswork

IDC research has shown that most business managers admit to basing their intra-day decisions on personal knowledge and educated guesswork rather than on data. Data is commonly used for monthly, quarterly, annual, and multiyear planning but not for "in the moment" business decisions.

The reason for this is simple: The data is not available. Most business intelligence systems extract data on a periodic basis from operational databases and load it into analytic databases for query and reporting. These may be operational data stores, data marts, or data warehouses. Operational data stores and data marts typically have schemas similar to those of operational databases, but with indexes for query enhancement, cubes, materialized views, and so forth to enhance query performance. They may receive data through frequent extract, transform, and load (ETL) processes or from dynamic data movement tools such as change data capture (CDC) software. They are used to make short-term planning decisions, looking at the next day, the next week, and the next month. Data warehouses, which receive data from many databases and collect it in a schema carefully crafted to encompass all the source data in a way that optimizes the value of queries, receive their data through scheduled ETL jobs and are used for quarterly, semiannual, annual, and multiyear planning exercises.

Operational data stores, data marts, and data warehouses are important for business planning but are of little value in informing moment-by-moment decision making. For this, it is necessary to look at live data, not data that is hours or days old, and execute complex queries that will not slow down the transactional systems that manage it.

The problem, until recently, has been that DBMSs were not designed for this purpose. Transactional databases are typically implemented on relational DBMSs (RDBMSs) that keep the data in rows, for efficiency in updates that usually involve most of the columns in a table. The database has as few indexes as possible because index management slows down the database and so impacts transaction throughput. Because there are few or no secondary indexes, analytic queries, which generally retrieve data based on specific column values and matches of column values with those in other tables, are forced to do table scans to find the data, if they are run on the transaction database. It is for this reason that the data is moved to another database that is optimized for query.

The established approach to setting up a query/reporting database (ODS, data mart, data warehouse) has involved establishing indexes for all columns that might have value lookup operations in the queries. Many organizations now use columnar databases, which have the same relational characteristics as row-oriented databases but store the data in blocks of column rather than row data for speed of retrieval. This obviates the need for indexes and, in some cases, for cubes and materialized views.

It does require keeping the data in a database separate from the transactional database and refreshing the data on a periodic basis, which makes the inclusion of live transaction data in such queries impossible. It also means that if the transactional database schema changes, the others must be changed as well, and that if queries or reports are requested that were not anticipated when the query database was set up, it would need to be modified to satisfy such requests. All this leads to further delay in serving the business.

The Solution: Memory-Optimized Analytics on Live Data

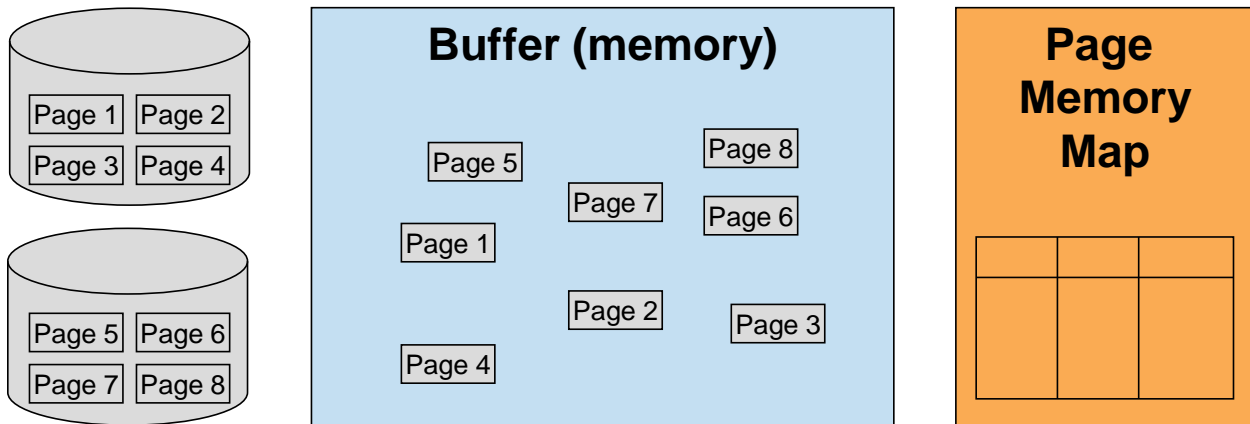
If live data is to be queried and updated at the same time, the queries must be very fast in order to avoid consuming resources on the database server and slowing down transactions. A number of vendors have created database technologies that optimize query performance by combining two key elements: query-optimized columnar organization for the data and memory-optimized database operations. In the case discussed here, however, there is an additional challenge, which is to maintain that data in a form that also supports a high-performance transactional database.

What Is Memory Optimization?

Memory optimization involves organizing the data for optimizing in-memory rather than on-disk operations. This means that instead of residing in disk buffers when in memory, the data is in a space especially arranged to ensure rapid data access. Because the data is organized this way, the database server performs fewer machine instructions to execute any given data access task. To understand why this is, consider how data is organized in a disk-optimized database (see Figure 1).

FIGURE 1

Disk-Optimized RDBMS Organization



Source: IDC, 2015

In this example, table rows are kept in disk records. The records are stored in blocks on disk, commonly called "pages." Each page has a table of contents, which indicates where on the page each record (row) is located. When a page is read into memory, it is stored in a database buffer. Alongside that buffer is a page memory map, indicating which pages are in memory and where in the buffer they are located. If data is required from a page not in memory, one of the pages in memory is "flushed" (or written) to disk to make room for another page.

There is also a directory of table and index locations on disk. If an index is used, that index is fetched into the buffer and searched for the matching entry or entries. Each entry contains a database key that identifies the location of the corresponding table row in the form of a page number and line (or block) number, which is used to retrieve the required table row. If an index cannot be used and a table scan is required, the table directory is used to find the first row of the table and then the rows are read sequentially. Since pages contain rows from more than one table, and since a table may have its rows scattered across many pages to optimize certain queries that the database administrator (DBA) knows about, pages with data that is mostly irrelevant to the query are often fetched into memory.

By contrast, memory-optimized data is kept in memory all the time, so there are no I/Os. Table locations are indicated by memory addresses rather than table directories. There are no buffers, no pages, no wasting memory with irrelevant data, and no instructions spent on index lookups, directory lookups, or other operations made necessary by a disk-oriented memory format.

Data that is optimized for query is organized in a compressed columnar format that speeds both sequential and random retrieval of data. The compression format is designed to enable "vector" processing using single instruction, multiple data (SIMD) operations to further reduce the number of machine instructions required to find the desired data.

Typically, memory-optimized databases organize their data in either a row-oriented way (for transaction processing) or a column-oriented way (for query processing). Very few are organized in such a way as to support both fast transaction and query processing against the same data.

The Value of Running Analytics on Live Data

The ability to execute queries against live transactional data enables enterprises to greatly improve their agility by adjusting decisions and processes to facts on the ground as the business day progresses.

Make Intra-Day Decisions Based on Facts

Instead of guessing or depending on personal experience or "intuition" to make intra-day business decisions, managers and staff can now use up-to-the-minute data. The result should be faster, more confident choices; better overall performance; and fewer mistakes or blunders. This also means that the enterprise can take advantage of late-breaking developments to act on opportunities that might otherwise simply pass by. For example:

- A retailer uses a combination of current sales, inventory, and warehouse data to determine item restocking, reordering, and even pricing to achieve minimal inventory management costs and ensure a steady flow of business.
- A logistics firm uses current traffic data (which is, of course, exegetical) together with pickup and delivery orders, including late-breaking pickup requests, to route trucks, saving time and money on fuel and better serving customers.
- A manufacturing firm adjusts production on the line based on the cost, availability, and timely delivery of parts (carefully managing the whole supply chain), applying the just-in-time model to a minute-by-minute, hour-by-hour level of management, ensuring steady output with the lowest possible cost.

Use Business Intelligence to Drive Business Processes

Because live data is not usually available to enterprises, they tend to follow fixed, prescribed business processes. These processes have defined stages and decision points, and although the designers of such processes try to take known possible problems and exceptions into account, these process definitions often don't fit perfectly with the way business actually needs to be done. This forces staff to either fudge the process or force customers, partners, suppliers, and so forth into formal procedures that seem cumbersome and partly unnecessary and can alienate partners and suppliers and lose customers.

When live analytical data can be taken into account, business processes can become more flexible and fluid because they are defined not by fixed stages but by the data. The result is more agility and greater success.

The Oracle Database In-Memory Option

To address the need for this type of query support against live transactional data, Oracle offers the In-Memory Option for Oracle Database. This capability, which is available on any environment supported by Oracle Database, including, of course, Oracle Engineered Systems, is designed to deliver all the memory optimization advantages for query that are described in this white paper, allowing the user to query live transaction data while also improving transactional processing speed.

How It Works

The In-Memory Option allows users to identify some tables with an in-memory attribute, which causes them to be loaded into the in-memory column store. When transactions are performed on those tables, they are managed in row format, but the updates are simultaneously sent to the column store and applied there, so the data in the column store always matches the data in the row store. Because all queries for such tables are sent to the column store, no secondary indexes are required for the

row-format tables, and none need be maintained on disk. Because queries act on the in-memory column store, they have no significant negative impact on transaction processing.

Benefits

The absence of secondary indexes on memory-optimized tables means that no index maintenance is necessary, so transactions are sped up. Also, such tables do not need to be arranged in a special way on disk volumes; rather, they can be packed together. This can both save on storage space and greatly reduce administrative time and effort in maintaining those tables. The overall benefits of using the In-Memory Option for Oracle Database may be summarized as follows:

- High-speed query capability on live transactional data can enable management and staff to make timely, intra-day decisions based on the current state of the business.
- Business processes can be redesigned to be data driven, resulting in a more agile enterprise that is responsive to changing operational conditions and market opportunities.
- Fewer indexes and packed tables mean lower storage costs and much less administrative and operational time and effort managing those tables.
- Because the row store operates in a manner compatible with the way it has worked in the past, no data conversion or configuration changes of any kind are required to take advantage of this feature, though simplification can actually improve performance.

SAP Certification

Although SAP certified Oracle Database 12c for use with SAP Business Suite on March 31, 2015, the In-Memory Option was certified later, on June 30. This means that SAP customers using Oracle Database 12c can employ the In-Memory Option with confidence, knowing that its use is approved and supported by SAP.

Customer Experience

AT&T Wireless

AT&T Wireless is a leading provider of wireless services in the United States. As part of its service, the company offers WiFi hotspots at over 45,000 locations in the United States and in some parts of Canada, which may be accessed at no charge by AT&T Wireless customers. Approximately 27-30 million customers use those hotspots. Data about that usage is tracked in an Oracle Database for analysis and service improvement. Reports show quality of service, volume of data, and how well AT&T's WiFi service is meeting the company's contractual obligations to customers. SAP BusinessObjects products are used for the analytics. Daniel Huls, Senior Technical Director for AT&T WiFi, said the company considered other in-memory columnar databases but decided to go with the Oracle Database In-Memory Option largely because the staff was already familiar with Oracle Database and the product required no changes to the applications or migration of the data.

AT&T started working with the technology in August 2014, and it was in production by October. It was implemented on a 2-node RAC cluster with a 100GB shared global area (SGA) and 80GB of memory per node. The database is about 20TB in size, running under a VMware configuration that assigns 8 cores to each database server node out of a total of 32 on each physical server (so it can use the other system resources for other workloads). VMware, Huls said, adds less than 5% overhead to the database. The company uses NetApp storage.

45,000 venues can access the data, though there are normally not more than a few hundred concurrent users at any given time. AT&T performs a full ETL operation every morning and incremental data loads on an hourly basis.

Huls said that after implementing the In-Memory Option, query time dropped from an average of 400 seconds to 10 seconds. Some run much faster, and users have noticed. According to Huls, one user commented, "We expected these reports in 10 seconds but never sub-second." Huls said that although the company did not expect improvement in load times, it found that load times were reduced, sometimes by as much as 50%, though the average load is about 10-15% faster than before.

Huls said that this initial implementation was basically effortless, and now that AT&T is becoming more familiar with the technology, it is looking at ways of optimizing its operations further. The company may do more frequent or complete loads, or even stream data into the database, for instance. Users are also able to execute more extensive analytics than was practical before, making their work more timely and effective.

FUTURE OUTLOOK

This kind of in-memory information management technology will open the door for fundamental changes in the ways that we use business data.

The Analytic-Transaction Platform

This approach to database management enables a new class of application, one that incorporates analytic queries into transaction processing and conditions the behavior of those transactions based on query results. Such functionality was not possible before because current data could not be queried in this way and query performance was not good enough to incorporate in a transactional application. Such an approach anticipates the emergence of an analytic-transaction platform, having at its core a memory-optimized database such as the one described here, with a constellation of other data that may be used to enrich the application's behavior. This topic is discussed more extensively in *The Analytic-Transactional Data Platform: Enabling the Real-Time Enterprise* (IDC #253100, December 2014).

Integration of Contextual Data

The analytic-transaction approach necessitates access to a range of contextual data that may be kept in other databases, including both relational and nonrelational databases, as well as other sources such as Hadoop. Such a platform requires speedy access to and compilation of such data and full integration with the core memory-optimized database. The full analytic-transaction platform is not currently available, but with key elements such as the Oracle Database In-Memory Option, it seems well on its way.

CHALLENGES/OPPORTUNITIES

Oracle is not the only vendor seeking to address this opportunity area. Each vendor has specific technical and marketing challenges, so the competitive outcome is hard to predict. Oracle should press its advantage, in the extensive extant use of Oracle Database around the world, as well as its growing acceptance in the cloud, to drive the adoption of this technology.

CONCLUSION

It has been clear for some time that memory optimization is the future of all DBMS technology and that it would play a key role in enhancing the strategic significance of relational databases in the enterprise. The ability to optimize both transactions and analytical queries involving the same data represents a new milestone in this process. Up to now, business managers and staff have been, relatively speaking, fumbling in the dark as they conduct business on a day-to-day basis. With the ability to perform high-speed analytic queries on transactional data, all that changes, and the following improvements become possible:

- Decisions can be made based on the current state of play based on certain knowledge.
- Business processes can be data driven rather than scripted, yielding greater business agility.
- Delays represented by data movement and maintenance of ODS or data mart schemas for tactical decision support become a thing of the past. Any question can be asked of live data, and the response is immediate.

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

Global Headquarters

5 Speen Street
Framingham, MA 01701
USA
508.872.8200
Twitter: @IDC
idc-insights-community.com
www.idc.com

Copyright Notice

External Publication of IDC Information and Data – Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2015 IDC. Reproduction without written permission is completely forbidden.

