Oracle Database In-Memory - a game changer for Data Warehousing?

Hermann Baer, Maria Colgan
Oracle USA
Redwood Shores, CA USA

Keywords:

Oracle Database 12c Release 12.1.0.2, Database In-Memory, Data Warehousing, Real-time analytics

Introduction

Customers are building data warehousing systems with more or less one basic goal in mind: to deliver value through the analysis of data. The bedrock of a solid data warehousing solution is a scalable, high-performance hardware infrastructure. With the Oracle Exadata Database Machine Oracle designed an Engineered System that delivers extreme performance with built-in functionality for optimized data processing, delivering order-of-magnitude performance gains for large-scale data warehousing environment along with very efficient data storage.

While Exadata tackles one major requirement for high-performance data warehousing – high bandwidth IO – Oracle Database In-Memory tackles another requirement that becomes increasingly more important for today agile data warehousing environments: interactive, real-time queries. Oracle Database In-Memory is useful for every data warehousing environment. It is entirely transparent to applications and tools, so that it is simple to implement for any existing environment

Database In-Memory – Concepts

Oracle Database has traditionally stored data in a row format. In a row format database, each new transaction or record stored in the database is represented as a new row in a table. That row is made up of multiple columns, with each column representing a different attribute about that record. A row format is ideal for online transaction systems, as it allows quick access to all of the columns in a record since all of the data for a given record are kept together in-memory and on-storage. A column format database stores each of the attributes about a transaction or record in a separate column structure. A column format is ideal for analytics, as it allows for faster data retrieval when only a few columns are selected but the query accesses a large portion of the data set.

Illustration. 1: Oracle's unique dual-format architecture

Oracle Database In-Memory (Database In-Memory) provides the best of both worlds by allowing data to be simultaneously populated in both an in-memory row format (the buffer cache) and a new inmemory column format.

With Oracle's unique approach, there remains a single copy of the table on storage, so there are no additional storage costs or synchronization issues. The database maintains full transactional consistency between the row and the columnar formats, just as it maintains consistency between tables and indexes. The Oracle Optimizer is fully aware of the column format: It automatically routes analytic queries to the column format and OLTP operations to the row format, ensuring outstanding performance and complete data consistency for all workloads without any application changes.

But reading data from memory can be orders of magnitude faster than reading from disk, but that is only part of the performance benefits of In-Memory: Oracle additionally increases in-memory query performance through innovative memory-optimized performance techniques such as vector processing and a new in-memory algorithm.

Key features include:

- **In-memory Column Store**. Data is stored in a compressed columnar format when using Oracle Database In-Memory. A columnar format is ideal for analytics, as it allows for faster data retrieval when only a few columns are selected from a table(s), especially when the query accesses a large portion of the rows from those table(s). Compression is a fundamental component of In-Memory, since enables more data to be stored in memory. Columnar data is very amenable to efficient compression; data is typically compressed 2-20x, often with better performance than non-compressed columnar data.

- **SIMD Vector Processing.** When scanning data stored in the IM column store, Database In-Memory uses SIMD vector processing (Single Instruction processing Multiple Data values). Instead of evaluating each entry in the column one at a time, SIMD vector processing allows a set of column values to be evaluated together in a single CPU instruction, for example in applying a where-clause predicates. In this way, SIMD vector processing enables the Oracle Database In-Memory to scan and filter billion of rows per second.

- **In-Memory Aggregation.** Analytic queries require more than just simple filters and joins. They require complex aggregations and summaries. A new aggregation algorithm, specifically optimized for the join-and-aggregate operations found in typical star queries, has been introduced with Oracle Database 12c Release 12.1.0.2. This algorithm allows dimension tables to be joined to the fact table, and the resulting data set aggregated, all in a single in-memory pass of the fact table.

Unlike a pure in-memory database, not all of the objects in an Oracle database need to be populated in the IM column store. The IM column store should be populated with the most performance-critical data, while less performance-critical data can reside on lower cost flash or disk. Thus, even the largest data warehouse can see considerable performance benefits from In- Memory.

Data Warehousing Information Reference Architecture

Data Warehousing is not a new paradigm, and Oracle has endorsed "Data Warehousing Reference Architecture" for over a decade (see here) to provide a blue print for building a relational Enterprise Data Warehouse.

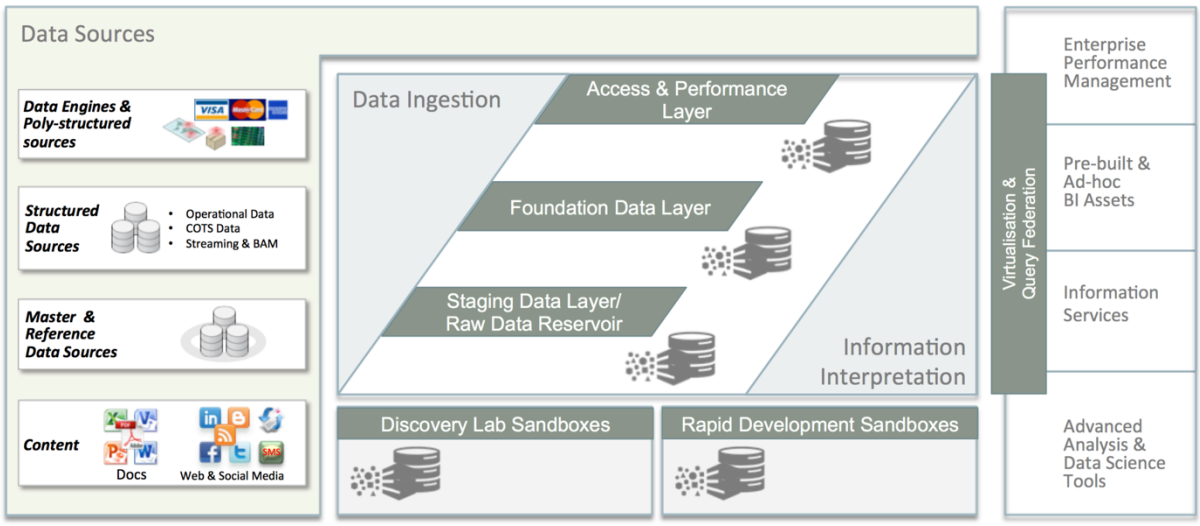

Illustration. 2: Oracle's Information Reference Architecture

The Information Reference Architecture has three main parts. It comprises of the Data Sources, the Data Warehouse itself, and the Information Access layer (incl. visualization and query federation). The core data warehouse, built upon a relational database, is the primary analytic database for storing much of a company's core transactional data: financial records, customer data, point-of-sale data and so forth. Within the core data warehouse you can find three different layers of data representation. Those are not necessarily implemented and physically instantiated in every data warehousing environment, but larger successful deployments tend to have some form of all three layers, at minimum in a logical form. These layers are namely:

Staging Data Layer: this layer acts as a temporary storage area for data manipulation before it enters the data warehousing. Data is either physically staged or only made accessible for the data warehousing processing (newer big data and data warehousing architectures tend to enrich the formerly purely relational environment with raw data reservoirs residing outside the database, with transparent access and potentially some pre-processing the data before it hits the data warehouse[1]).

Foundation Data Layer: This layer is sometimes referred to as the atomic data warehouse that stores data on the lowest possible level of granularity. It represents the heart of the data warehouse. The data is stored in a normalized fashion close to Third Normal Form (3NF) for storage efficiency; the data is stored in a completely subject-neutral manner.

Access and Performance Layer: While the Foundation Layer acts as the main source of information in the data warehouse, it provides little support to navigate through the rather complex data model to find the information necessary nor does it help to optimize the processing for any specific application. The Access and Performance layer adds the information access component to the architecture by providing a more business, end-user-friendly view of the data. You can also find additional performance-improving data structures, such as pre-aggregated information, in some implementations.

The impact of Database In-Memory on the Data Warehousing Information Reference

[1] A broader discussion of the trends in big data and data warehousing is beyond the scope of this article.

Architecture
Oracle Database In-Memory transparently accelerates analytic queries by orders of magnitude, enabling real-time business decisions. Using Database In-Memory, businesses can instantaneously run analytics and reports that previously took hours or days. Businesses benefit from better decisions made in real-time, resulting in lower costs, improved productivity, and increased competitiveness.

This begs the ultimate question: do you have to forget anything you have learned and implemented over the years with this new groundbreaking technology? What does it mean for you to adopt Oracle's new groundbreaking Database In-Memory? In short, nothing.

However your data warehousing environment looks like, nothing you have built over the years is void. Oracle's Database In-Memory technology provides instantaneous benefits for any existing environment. It enables the ultimate agile data warehousing environment, addressing all existing and future requirements.

Staging Data Layer: Database In-Memory provides performance speed-up for data extraction of relational source systems where the post-extraction transformation efforts are minimal. Data can be extracted much faster and with less resource impact on the source system. Under some circumstances, the direct raw data access of the operational source system avoids the physical replication of data into an ODS layer completely.

Foundation Data Layer: With a subject-neutral and non-performance optimized data representation the foundation layer will benefit most from Database In-Memory. Data can be readily accessed and analyzed for any subject area without the need of detailed analysis and development of a performance and access layer. The most valuable information is readily available for analysis, providing real-time insight even for the unknown questions. Note, however, that this does not make the implementation of any Access and Performance Layer void. It only shifts the main purpose of the Access and Performance layer towards controlled access, away from pure performance enablement (this is a trend that started with the introduction of Exadata already).

Access and Performance Layer: The purpose of this layer has steadily shifted away from a pure and necessary performance-enabling layer towards enabling ubiquitous data access to end user in business-understandable terms and to improve the quality of the data analysis. KPIs are pre-created and defined in the access and performance layer so that every known and well-understood data analysis is processed in the same agreed-upon way. The pre-creation is often done through secondary performance-enabling structures like materialized views, a data structure often used in dimensional modeling. Materialized views, just like normal tables, will benefit from Database In-Memory and make fast processing even faster. Ultimately, even with Database In-Memory and the achievable fast real-time analysis of any data, you should not forget about the most proven and pragmatic performance enabling measurement: avoiding unnecessary work and redundant, repetitive processing to begin with.

Contact address:

Hermann Baer, Maria Colgan

Oracle USA
400 Oracle Parkway
Redwood Shores, CA 94065

Phone: +1.650.506.7000
Fax: +1.650.506.7000
Email hermann.baer@oracle.com, maria.colgan@oracle.com
Internet: www.oracle.com