Oracle Database 11g for Data Warehousing and Business Intelligence

An Oracle White Paper
July 2007

Oracle Database 11g for Data Warehousing and Business Intelligence

Oracle Database 11g for Data Warehousing and Business Intelligence

INTRODUCTION

Oracle Database 11g is a comprehensive database platform for data warehousing and business intelligence that combines industry-leading scalability and performance, deeply-integrated analytics, and embedded integration and data-quality -- all in a single platform running on a reliable, low-cost grid infrastructure. Oracle Database 11g provides best-of-breed functionality for data warehouses and data marts, with proven scalability to 100's of TBs and market-leading performance. Oracle Database 11g also provides a uniquely integrated platform for analytics; by embedding OLAP, Data Mining, and statistical capabilities directly into the database, Oracle delivers all of the functionality of standalone analytic engines with the enterprise scalability, security, and reliability of an Oracle Database. Since data integration is a core requirement of any data warehouse, Oracle Database 11g includes a leading ETL tool, Oracle Warehouse Builder, which leverages Oracle's scalable data transformation and heterogeneous data-access capabilities.

This white-paper provides an overview of Oracle Database 11g's capabilities for data warehousing, and discusses the key features and technologies by which Oracle-based business intelligence and data warehouse system can integrate information, perform fast queries, scale to very large data volumes and analyze any data.

INTEGRATE

Today's information architecture is much more dynamic than it was just a few years ago. Businesses now demand more information, they want it sooner and they are delivering more analytics to an every-widening set of users and applications.

Oracle Warehouse Builder

To address these business requirements, Oracle Database 11g includes a leading data-integration tool, Oracle Warehouse Builder (OWB). The core features of OWB are included as a no-cost database feature, and OWB additionally has three options for specific integration requirements:

> Base product: The base product is designed to allow any Oracle customer to efficiently build a data mart or data warehouse, of any size or complexity. It includes an enterprise-ready multi-user metadata repository, data-modeling

capabilities, and wide variety of transformation and extraction techniques, and the performance and scalability of an 'ELT' architecture.

Enterprise ETL Option: This option is specifically geared to increase performance and productivity, and include the following features:

Advanced Data Load Options

Developer Productivity through reusable components

Embedded slowly changing dimension management

End to end data lineage and impact analysis

Support for advanced configuration management

Data Quality Option: As a fully integrated part of the product, a very distinct advantage over all other tools in the market, the data quality option provides support for data profiling, data rules (in essence business rules) and information compliance features.

Connectors: Connects provide optimized access for leading operational applications. Oracle provides connectors for Oracle e-Business Suite, Peoplesoft Enterprise, Siebel (CRM), and SAP R/3.

A key advantage of OWB is the breadth of functionality that it provides integrated within a single tool. Data modeling, data compliance, and data quality are core features that any enterprise data integration tools must possess. However, a key architectural advantage of Oracle Warehouse Builder is the integration of the components. Oracle Warehouse Builder provides all of its capabilities within a common repository and user interface.
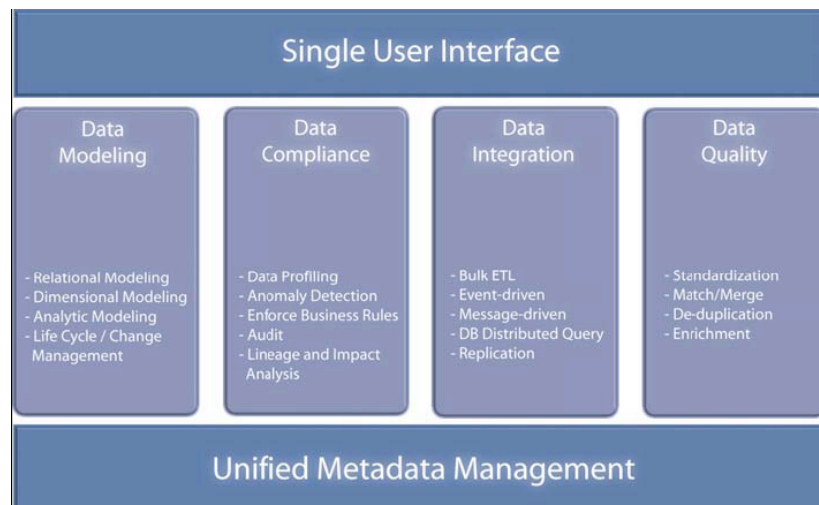
Figure 1

By providing all of these capabilities in a single tool on top of single repository, OWB resolves a long-standing challenge in data integration. Many integration solutions provide separate tools for these distinct capabilities However, it is terribly inefficient to do data-

modeling in one tool, and then ETL mapping in another tools, and then data-profiling in yet another tool. OWB provides one metadata repository and one UI for the entire integration process.

Oracle Warehouse Builder uses an extract-load-transform ('ELT') architecture. Rather than providing an external data-transformation engine, OWB executes all of its transformations within an Oracle database leveraging the scalability and performance of the database platform.

Key database integration features

The Oracle database has a broad set of extraction, loading and transformation capabilities. These features are leverages by OWB, but can be used in any environment in which data is being integrated into an Oracle data warehouse. These features include:

Database Gateways for accessing non-Oracle systems

Loader utility for fast data loads of flat files

SQL extensions for data transformations: MERGE statement

Table functions: efficient parallel user-defined transformations

Change data capture for low-latency log-based capture from Oracle databases

PERFORM

Oracle provides performance optimizations for every type of data warehouse environment. Data warehouse workloads are often complex, with different users running vastly different operations, with similarly different expectations and requirements for query performance.

Oracle meets the demands of data warehouse performance by providing:

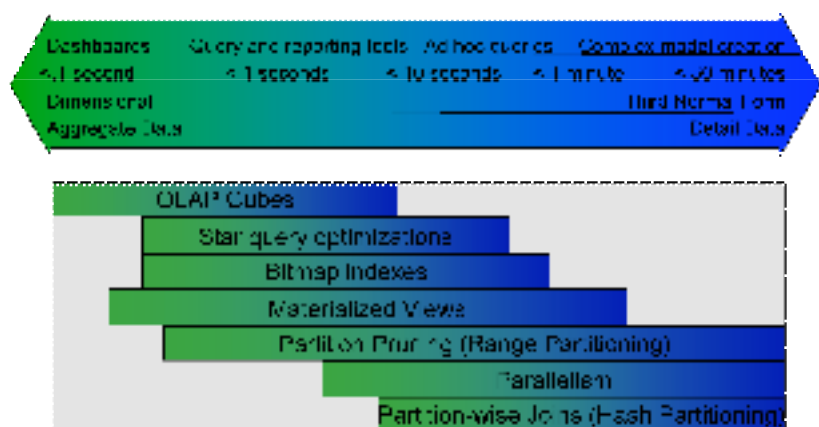a broad set of performance optimization techniques for every type of query and workload:

Figure 2

a sophisticated resource manager for ensuring performance even in databases with complex, heterogeneous workload. The Database Resource Manager allows end-users to be grouped into 'resource consumer groups', and the DBA can control exactly how resources are allocated to each group. For each group, the DBA can control the amount of CPU utilization, as well as specify policies for proactive query governing, and for query queuing.

a unique, patented read-consistency model to ensure that data loads never impact query performance. Oracle solves the challenges of concurrent access through a technology called multi-version read consistency; this unique technology has been the foundation of Oracle's concurrency model for over 15 years. Multi-version read consistency guarantees that a user always sees a consistent view of the data requested. If another user changes the underlying data during the query execution (such a trickle-feed update of a large data warehouse table), Oracle maintains a version of the data as it existed at the time the query began. The data returned to the query always reflects the state of the database (including all committed transactions) at the point in time at which the query was submitted regardless of what other updates may be occurring while the query is running. With this technology, Oracle is uniquely positioned to handle near real-time data loads within data warehouse environments.

Management Packs to automate the ongoing performance tuning of a data warehouse.

One of the most interesting new performance features in Oracle Database 11g is OLAP-based materialized views. Conventional performance-tuning methods for star schemas within data warehouses have long involved the use of summary tables (or materialized views). Using this approach, the performance of queries is improved by pre-computing one or more summaries and storing those summaries in the data warehouse. These summaries are completely transparent to the application, since the database provides 'query rewrite' capabilities in which a query against a base set of star-schema tables is rewritten to access summary data. One challenge with materialized views is that a single star schema can have thousands or even millions of possible aggregations that could be stored as summary tables. It is not practical to create a materialized view for every possible aggregation; the space utilization and maintenance costs would be too high. However, this scenario can be resolved with OLAP-based materialized views. An OLAP-based materialized view leverages the OLAP data structures, in which all possible aggregations can be quickly retrieved. Thus, a star schema with an OLAP-based materialized view provides the performance of a star schema with thousands of relational materialized views … but with a single easy-to-manage OLAP cube that is highly compressed and provides efficient update capabilities. This application of OLAP technology to general-purpose data warehouse workloads is unique to Oracle. While previous OLAP solutions were accessed via an OLAP-based API, Oracle OLAP delivers the performance benefits of OLAP to all data warehouse users using any SQL-based tool.

SCALE

The size of the largest data warehouses is growing exponentially. Today, more and more business processes are becoming more completely automated and more data is collected a more granular levels, so these data volume increases show no signs of abating.

With the Oracle Database, there are four key capabilities to enable scalability: partitioning, compression, clustering, and parallelism

Partitioning

Oracle Partitioning is essential for managing large databases. It enables a "divide and conquer" technique for managing the large tables in the database, especially as those tables grow.

Although your database may have twice as much data next year as it does today, your end-users are not going to tolerate their application running twice as slow, your database is not going to be given twice as much time to complete maintenance and batch processing, and your IT managers are not going to double the hardware budget for the data warehouse. Partitioning is the feature that allows a database to scale for very large datasets while maintaining consistent performance, without unduly increasing administrative or hardware resources. Partitioning breaks up large tables up into smaller pieces, and thus allows data management windows and many end-user queries to be maintained at constant performance level even as the data grows.

Oracle has been developing its partitioning technology for over 10 years, since its introduction in Oracle8. In Release 11g, Oracle provides nine methods for partitioning tables, along with the capability for DBA's to define custom partitioning schemes; a rich set of administrative commands for partitioned tables; and a partition adviser to guide administrators on how best to implement partition.

Partitioning also enables ILM ("Information Lifecycle Management") strategies within the Oracle database. A single table, when partitioned, can be distributed across multiple storage tiers. Old, less-frequently accessed data, corresponding to older partitions, can be stored on less expensive storage devices. For large databases, this approach can provide significant cost savings.

Compression

Since its introduction in Oracle9i Release 2, compression has grown to be one of the most popular features for data warehouse customers. As customers look to store larger volumes of data, compression is a natural solution. Oracle's compression algorithms provides a unique mechanism for compressing data stored in relational tables with virtually no negative impact on query performance. With typical compression ratios ranging from 2:1 to 5:1, the popularity of this feature comes as no surprise.

In Oracle Database 11g, compression has been enhanced to support frequent, small updates and inserts. In previous releases, data was compressed when inserted into a table using 'direct-path' operations ('bulk' operations in which thousands of rows are

inserted when using a single SQL statement or when using the load utility). For smaller updates and inserts into a compressed table, the new data would not be compressed in these earlier releases. This new enhancements in 11g brings the benefits of partitioning to data warehouse tables which are 'trickle-fed' or otherwise updated frequently.

Real Application Clusters

Real Application Clusters enable a single database to scale across multiple servers. For data warehousing, RAC provides a solution to 'scale-out' to grow to hundreds of CPU's. The architecture of RAC provides unique benefits for Oracle data warehousing. In addition to the capability of adding additional raw computing power to a data warehouse system, RAC also enables high-availability against node failure.

RAC also delivers a tremendous amount of flexibility for managing multiple workloads within a single database. Using 'RAC services', different applications be assigned to separate services, which in turn can run on separate nodes of a cluster. For example, data mart users could be assigned to one set of nodes while ETL operations could be assigned to another set of nodes. A key advantage of RAC is the ability to dynamically change and adjust these workloads. One could easily and dynamically allocate more nodes to ETL processing at nighttime, for example, while re-allocating those nodes to query processing during the day.

Parallelism

Parallelism is the ability to leverage multiple processors towards the execution of a s single database operation. Oracle's scalable parallel execution architecture dynamically adjusts to meet all user demands. For large operations, Oracle parallelizes database operations across all available nodes and cpu's. For smaller operations, Oracle dynamically adjust the parallelism to maximize overall system throughput while the Database Resource Manager ensure that high-priority operations continue to get the appropriate resources.

ANALYZE

Technologies such as OLAP, statistics, and data mining are hardly new to data warehousing and business intelligence (indeed, all of these technologies mostly predated data warehousing). Data warehousing practitioners have been able to purchase products with each of these capabilities for years. However, OLAP products typically have their own calculation engine, statistics products have their own data engine, and data mining products have their own mining engines. In short, an enterprise-wide business intelligence environment would maintain at least four different types of 'data engines', each requiring their own servers, their own management infrastructure, their own security administration, their own high-availability infrastructure. Each engine has its on API's and its own set of developer tools and end-user tools. The complexity and cost of replicating entire stacks of BI technologies is significant.

Oracle Database provides a completely different approach. Oracle has implemented OLAP, Data Mining, and statistics inside of it database engine. Instead of moving data

from a data warehouse to other analytic engines for further analysis, Oracle has instead brought the advanced analytic algorithms into its database, where the data resides. Moreover, Oracle provides SQL access to all of its analytics, so that they can be implemented with any SQL-based tool or application environment.

Beyond the considerable advantages of rationalizing the back-end data architecture of an enterprise business intelligence environment, the integration of analytics within the Oracle Database provides a host of advantages unavailable to stand-alone environments. For example, does your standalone OLAP server scale to hundreds of CPU's or clusters of servers? How easily does your data-mining engine integrate into your user authentication server? And can it transparently implement all of your data security policies? How easily can you integrate the results of your statistical analysis with your data warehouse data? Within Oracle Database, all of these issues are solved simply due to the deep integration of OLAP, Data Mining and statistics.

Data Mining

Oracle Data Mining is powerful software embedded in the Oracle Database that enables you to discover new insights hidden in your data. Oracle Data Mining helps businesses to target their best customers, find and prevent fraud, discover the most influential attributes that affect Key Performance Indicators (KPIs), and find valuable new information hidden in the data. Oracle Data Mining helps technical professionals find patterns in their data, identify key attributes, discover new clusters and associations, and uncover valuable insights.

To address these business problems, Oracle Data Mining allows companies find new information from their data using a wide range of state-of-the-art algorithms. Data mining algorithms are machine-learning techniques for analyzing data to discover patterns and relationships. Oracle provides multiple algorithms since different algorithms are effective for different types of analysis and different business problems.

Most data mining algorithms can be broadly separated into "supervised learning" and "unsupervised learning" data mining techniques. Supervised learning requires the data analyst to identify a target attribute or dependent variable (for example, customers who bought a specific product). The supervised-learning technique then sifts through data trying to find patterns and relationships between other attributes and the target attribute (for example, the characteristics that indicate whether a prospective customer is likely to buy a specific product). Supervised learning algorithms with Oracle Data Mining include Naïve Bayes, Decision Tree, General Linear Models, and Support Vector Machines.

The other broad category of data-mining algorithms is for "unsupervised learning." In these scenarios, there is not 'target attribute'; instead the data mining algorithms seek to find associations and clusters in the data independent of any a priori defined business objective. These algorithms include Enhanced k-Means Clustering, Orthogonal Partitioning Clustering, Association Rules (market basket analysis), and Nonnegative Matrix Factorization.

Oracle Data Mining includes Oracle Data Miner, a graphical user interface for data analysts to build, evaluate, and apply data mining models. Oracle Data Miner guides the data analyst through the data mining process with complete flexibility and presents results in graphical and tabular formats. Oracle Data Miner can generate the PL/SQL code associated with a Mining Activity.

OLAP

Oracle OLAP is a full-feature online analytical processing (OLAP) engine embedded in the Oracle Database. Oracle OLAP enhances data warehouses by improving query performance (as discussed in the performance section) and by adding enriched analytical content.

The core feature of Oracle OLAP is cubes. Managed within the Oracle database, this data structure stores data within a highly optimized multidimensional format. Cubes provide scalable and compressed storage of dimensional data, fast incremental update, fast query performance, and the ability to compute or store advanced analytical calculations.

Oracle's strategy with Oracle OLAP is to bring these core OLAP advantages into the data warehouse. This is achieved be exposing the key capabilities of Oracle OLAP via standard SQL, so that any business intelligence tools or other SQL-based application can leverage OLAP.

The key SQL integration features include the ability to access OLAP cubes via SQL views, which expose a multidimensional cube as a set of dimension table sand fact tables; cube-based materialized views which allow OLAP cubes to be used as materialized views; data-dictionary integration so that SQL-based query tools as well as SQL developers can quickly identify all of the cubes and their attributes; and the integration of cube access paths within the Oracle optimizer.

CONCLUSION

The Oracle Database is the market leader for data warehousing, built upon a solid foundation of scalability and performance, and augmented innovative features such as Oracle's unique read-consistency model for near-real-time data warehouses, a flexible and powerful set of table partitioning capabilities, the utilization of OLAP technology to enhance relational environments, and the unmatched support for grid architectures. Oracle Database 11g also provide the most comprehensive database platform for data warehouses and other business intelligence applications, by embedding data-integration, OLAP, and data mining capabilities. Moreover, the Oracle Database 11g provides enterprise-ready availability, security and manageability – key requirements for data warehouses as much as any other enterprise system.

ORACLE