An Overview of Data Warehousing and OLAP Technology

E. Rajakumar and R. Raja

1Assistant Professor, Department of Computer Science and Engineering, Sri Aravindar Engineering College, Sedarapet.
2Principal, Sri Aravindar Engineering College, Sedarapet, Aadhar ID: 301904009901

ABSTRACT

A data warehouse can be used to analyze integrates data from multiple data sources can retrieve a historical data and data kept in the data warehouse is non volatile in nature. The data stored in the warehouse is uploaded from the operational systems and also known as an enterprise data warehouse (EDW), is a system used for reporting and data analysis for creating trending reports for senior management such as need for data warehousing, differences between operational databases and data warehouses, relational OLAP and indexing bitmap in OLAP and architecture data warehouse were discussed in detail in this paper.

INTRODUCTION

The concept of data warehousing dates back to the late 1980s, the concept of Barry Devlin and Paul "business data warehouse" for the Data warehouse was debuted by IBM researchers Barry Devlin and Paul Murphy developed the "warehouse" for the flow of data from operational systems to decision support environments. The concept attempted to address the various problems associated with this flow, mainly the high costs associated with it, in the absence of data warehouse architecture, an enormous amount of redundancy was required to support multiple decision support environments. In larger corporations it was typical for multiple decision support environments to operate independently, though each environment served different users; they often required much of the same stored data. The process of gathering, cleaning and integrating data from various sources, such as long-term existing operational systems, was typically in part replicated for each environment. Moreover, the operational systems were frequently reexamined as new decision support requirements emerged. Often new requirements necessitated gathering, cleaning and integrating new data from “data marts” that were tailored for ready access by users.

Need for Data Warehousing:

The data needed to provide reports, analytic applications and ad hoc queries all exist within the set of production applications that support the organization. There is no reason to add the name of organization to the long list of failures and many reasons that the direct connection never works. Incase of the new releases of application software, frequently introduce changes that make it necessary to rewrite and test reports. These changes make it difficult to create and maintain reports that summarize data originating within more than one release, field names are often hard to decipher and meaningless strings of characters, application data is often stored in odd formats such as Century Julian dates and numbers without decimal points. Tables are structured to optimize data entry and validation performance, making them hard to use for retrieval and analysis. There is no good way to incorporate worthwhile data from other sources into the database of a particular application. Developing and storing metadata is the process without a data warehouse, there is no obvious place to put it. Many data fields that users are accustomed to seeing on display screens are not present within the database, such as rolled-up, general ledger balances and priority to be given to transaction processing. Reporting and analysis functions tend to perform poorly when run on the hardware that handles transactions.

Corresponding Author: E. Rajakumar, Assistant Professor, Department of Computer Science and Engineering, Sri Aravindar Engineering College, Sedarapet.
E-mail: rajkumar30980@gmail.com
Differences between Operational Database Systems and Data Warehouses:

IT systems can be divided into transactional (OLTP) and analytical (OLAP). In general, it is assumed that OLTP systems provide source data to data warehouses (Thomsen, E., 1997), whereas OLAP systems help to analyze it and differences between them is given in Table 1 and in Fig 1.

![Operational Database Systems and Data Warehouses](image)

**Fig. 1: Operational Database Systems and Data Warehouses**

<table>
<thead>
<tr>
<th>CONTENT</th>
<th>OLTP</th>
<th>OLAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source of data</td>
<td>Operational data; OLTPs are the original</td>
<td>Consolidation data; OLAP data comes from the various OLTP Databases</td>
</tr>
<tr>
<td>Purpose of data</td>
<td>To control and run fundamental business</td>
<td>To help with planning, problem solving, and decision support</td>
</tr>
<tr>
<td>What the data</td>
<td>Reveals a snapshot of ongoing business</td>
<td>Multi-dimensional views of various kinds of business activities</td>
</tr>
<tr>
<td>Inserts and Updates</td>
<td>Short and fast inserts and updates initiated by end users</td>
<td>Periodic long-running batch jobs refresh the data</td>
</tr>
<tr>
<td>Queries</td>
<td>Relatively standardized and simple queries</td>
<td>Often complex queries involving aggregations</td>
</tr>
<tr>
<td>Processing Speed</td>
<td>Typically very fast</td>
<td>Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes</td>
</tr>
<tr>
<td>Space Requirements</td>
<td>Can be relatively small if historical data is archived</td>
<td>Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP</td>
</tr>
<tr>
<td>Data base design</td>
<td>Highly normalized with many tables</td>
<td>Typically de-normalized with fewer tables; use of star and/or snowflake schemas</td>
</tr>
<tr>
<td>Backup recovery</td>
<td>Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability</td>
<td>Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method</td>
</tr>
</tbody>
</table>

**Relational OLAP (ROLAP):**

Use relational or extended-relational DBMS to store and maintain warehouse data and OLAP middle ware to support missing pieces, include optimization of DBMS backend, implementation of aggregation navigation logic, and additional parameter and services greater scalability. Multidimensional OLAP (MOLAP), array-based multidimensional storage engine fast indexing to pre-computed summarized data, hybrid OLAP (HOLAP) is combination of ROLAP and MOLAP technology - User flexibility, such as low level; relational and high-level; array Specialized SQL (Gupta and I.S. Mumick, 1999) servers specialized support for SQL queries over star and snowflake schemas.

**Indexing OLAP Data: Bitmap Index:**

Index on a particular column each value in the column has a bit vector, bit-op is fast and the length of the bit vector number of records in the base table was the \( i^{th} \) bit is set if the \( i^{th} \) row of the base table has the value for the indexed column not suitable for high cardinality domains (many different values) given in Fig 2.
Indexing OLAP data and joining indices was as follows: Join index: \( JI(R-id, S-id) \) where \( R \) (\( R-id, \ldots \)) \( S \) (\( S-id, \ldots \)), traditional indices map and the values to a list of record ids, it materializes relational join in \( JI \) file and speeds up relational joining rather costly operation in data warehouses, join index relates the values of the dimensions of a star schema to rows in the fact table illustrated from fact table \emph{Sales} and two dimensions \emph{city} and \emph{product}. A join index on \emph{city} maintains for each distinct city a list of \( R\)-IDs of the tuples recording the sales in the city (ditto for join index on \emph{item}) and join indices can span multiple dimensions. Efficient Processing OLAP queries determine which operations should be performed on the available cuboids, transform drill, roll, etc. into corresponding SQL and/or OLAP operations, and it can be illustrated from the figure dice = selection + projection determines to which materialized cuboids (s) the relevant operations should be applied. Exploring indexing structures and compressed against dense array structures in MOLAP.

\textbf{On-Line Analytical Processing to On Line Analytical Mining (OLAM):}

The advantage of online analytical mining were high quality of data in data warehouses, contains integrated, consistent, cleaned data available information processing structure surrounding data warehouses for ODBC, OLEDB, web accessing, service facilities, reporting and OLAP tools such as OLAP-based exploratory data analysis mining with drilling, dicing, pivoting. On-line selection of data mining (Sristava, D., \emph{et al.}, 1996) functions integration and swapping of multiple mining functions and tasks were presented in Fig 3.

\textbf{Data Warehouse Architecture:}

The architecture in Fig4 is quite common, might and it customize the warehouse’s architecture for different groups within the organization, do this by adding data mart which are systems designed for a particular line of
business and Fig 4 illustrated on example were purchasing sales, and inventories are separated in this example a financial analysis’s of historical data for purchase and sales.

![Architecture of a Data Warehouse](image)

**Fig. 4: Architecture of a Data Warehouse**

**Data Warehouse Development and Recommended Approach:**

Enterprise warehouse was collection of all the information about subjects spanning the entire organization (Fig 5), data mart a subset of corporate-wide data that is the value to specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart, virtual warehouse is a set of views over operational databases and may be materialized in future.

![Data Warehouse Development: A Recommended Approach](image)

**Fig. 5: Data Warehouse development**

**Conceptual Modeling of Data Warehouses:**

Star schema in which a fact table in the middle connected to a set of dimension tables, It contains a large central table that is (fact table given Fig 6), a set of smaller attendant tables (dimension table), and one for each dimension

![Star Schema](image)

**Fig. 6: Star Schema**
Snowflake schema is refinement of star schema where some dimensional hierarchy is further splitting (normalized) into a set of smaller dimension tables (Fig 7), forming a shape similar to snowflake. However, the snowflake structure can reduce the effectiveness of browsing, since more joins will be needed.

Multiple share dimension tables where viewed as a collection of stars, called galaxy schema or fact tables’ fact constellation (Fig 8).

**Concept Hierarchies:**

![Diagram of Concept Hierarchies](image)

**Fig. 9:** (a) Concept Hierarchies

**Fig. 9:** (b) Concept Hierarchies
Concept hierarchies are defined as defined by grouping values for a given dimension or attribute, resulting in a set-grouping hierarchy.

**Data Cube:**
A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions named as *sales* data warehouse.

**Design of Data Warehouse:**
Four views regarding the design of a data warehouse and they are top-down view allows selection of the relevant information necessary for the data warehouse, data source view exposes the information being captured, stored, and managed by operational systems, data warehouse view take care of consists of fact tables and dimension tables, business query view sees the perspectives of data in the warehouse from the view of end-user.

**Three Data Warehouse Models:**
Enterprise warehouse collects all of the information about subjects spanning the entire organization, data mart a subset of corporate-wide data that is of value to specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart warehouse, and a set of views over operational databases. Only some of the possible summary views may be materialized.

**Metadata Repository:**
Meta data is the data defining warehouse objects. It has the following kinds of description of the structure of the warehouse schema, view, dimensions, hierarchies, derived data defines, data mart locations and contents of operational meta-data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (warehouse usage statistics, error reports, audit trails). The algorithms used for summarization the mapping from operational environment to the data warehouse, data related to system performance of warehouse schema, view and derived data definitions of business data and business terms and definitions, ownership of data, charging policies.

**3D Data cube Example:**

![Fig. 10: Data cube Example](image-url)

![Fig. 11: Data cube Example](image-url)
Efficient Data Cube Computation:

Data cube can be viewed as a lattice of cuboids, the bottom-most cuboids is the base cuboids, the top-most cuboids (apex) contains only one cell. Materialization (pre-computation) of data cube, materialize every (cuboids) (full materialization), none (no materialization), or some (partial materialization), selection of such cuboids to materialize is based on size, sharing, access frequency, etc. cube definition and computation in DMQL defines cube sales[item, city, year]: sum(sales_in_dollars) and compute cube sales transform into a SQL-like language (with a new operator cube by, introduced by Gray et al.1996) example SELECT item, city, year, SUM (amount), FROM SALES CUBE BY item, city, year Need compute the following Group-Bys(year, item, city), (year, item), (year, city), (item, city), (year), (item), (city) cube computation: ROLAP-Based Method of efficient cube computation methods were ROLAP-based cubing algorithms (Agarwal et al. 1996), Array-based cubing algorithm (Zhao et al. 1997), Bottom-up computation method (Bayer and Ramakrishnan 1999).

ROLAP-based cubing algorithms were sorting, hashing, and grouping operations are applied to the dimension attributes in order to reorder and cluster related tuples grouping is performed on some sub aggregates as a “partial grouping step”. Aggregates may be computed from previously computed aggregates, rather than from the base fact table. Hash/sort based methods (Agarwal et al. VLDB 1996) (Berson and S.J. Smith, 1993) in this method smallest-parent computing a cuboids’ from the smallest previously computed cuboids’, Cache-results as caching results of a cuboids from which other cuboids are computed to reduce disk I/Os, a mortise-scans is computing as many as possible cuboids at the same time to amortize disk reads, share-sorts was sharing sorting costs across multiple cuboids when sort-based method is used and share-partitions sharing the partitioning cost across multiple cuboids when hash-based algorithms are used.

Multi-way Array Aggregation for Cube Computation:

Partition arrays into chunks (a small sub cube which fits in memory), compressed sparse array addressing: (chunk_id, offset), compute aggregates in “multi way” by visiting cube cells in the order which minimizes the # of times to visit each cell, and reduces memory access and storage cost. The planes should be sorted and computed according to their size in ascending order. keep the smallest plane in the main memory, fetch and compute only one chunk at a time for the largest plane and Limitation of this method were computing well only for a small number of dimensions, if there are a large number of dimensions, “bottom-up computation” and iceberg cube computation methods (Donjerkovice, D. and R. Ramakrishnan, 1999) can be explored. Another way of Multi-Way Array Aggregation for Cube Computation is the planes should be sorted and computed according to their size in ascending order and keep the smallest plane in the main memory, fetch and compute only one chunk at a time for the largest plane and limitation associated with this method is computing well only for a small number of dimensions, in case of a large number of dimensions, “bottom-up computation” and iceberg cube computation methods can be explored.

Discovery-Driven Exploration of Data Cubes:

Hypothesis-driven: Exploration by user, huge search space, discovery-driven (Sarawagi et al. ’98), pre-compute measures indicating exceptions, guide user in the data analysis, at all levels of aggregation and exception; significantly different from the value anticipated, based on a statistical model and visual cues such as background color are used to reflect the degree of exception of each cell. Computation of exception indicator can be overlapped with cube construction. Selfexp confirms the degree of surprise of the cell value, relative to other cells at the same level of aggregation. Inexp the degree of surprise somewhere beneath the cell, if one were to drill down from it and path the degree of surprise for each drill-down path from the cell.

Data Warehouse Back-End Tools and Utilities:

Data extraction gets data from multiple, heterogeneous, and external sources such as data cleaning (detect errors in the data and rectify them when possible), Data transformation (convert data from legacy or host format to warehouse format) load, sort, summarize, consolidate, compute views, check integrity, and build indices and partitions and refresh propagate the updates from the data sources to the warehouse.

Data Warehouse Usage:

Three kinds of data warehouse applications, information processing, supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs and analytical processing, multidimensional analysis of data warehouse data, supports basic OLAP operations, slice-dice, drilling, pivoting data mining, knowledge discovery from hidden patterns and supports associations constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools of differences among the three tasks.
Conclusion:

Data warehouse a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management’s decision-making process. A multi-dimensional model of a data warehouse star schema, snowflake schema, fact constellations and data cube consists of dimensions and measures. OLAP operations includes drilling, rolling, slicing, dicing and pivoting OLAP servers such as ROLAP, MOLAP, HOLAP. Efficient computation of data cubes Partial against full and no materialization, multi way array aggregation, bitmap index and join index implementations.

REFERENCES


