An Overview of Data Warehousing and OLAP Technology

Presenter: Otto
Discussion: Jim
Mar. 07 2006

Outline
- What is decision support
- What is a data warehouse
- Why we need it, and how it differs from a regular RDBMS
- Difference between OLAP and OLTP
- Typical OLAP architecture
- Database Design Methodology
- Star and Snowflake schemas
- Challenges of materialized views
- Implementation of the OLAP Server
- Metadata requirements

Motive for a Data Warehouse
- Businesses have a lot of data, operational data and facts.
- This data is usually in different databases and in different physical places.
- Decision makers need to access information (data that has been summarized) virtually on the single site.
- This access needs to be fast regardless of the size of the data, and how old the data is.

What is decision support
- Decision support systems are a class of computerized information systems that support decision making activities.
- Decision support systems usually require consolidating data form many heterogeneous sources: these might include external sources.
  - Such us stock market feeds.

What is data warehouse
- Data warehousing provides architectures and tools for business executives to systematically organize, understand and use their data to make strategic decisions. - Jiawei Han
- A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management’s decision making process.

Difference between OLAP and OLTP

<table>
<thead>
<tr>
<th></th>
<th>OLTP</th>
<th>OLAP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Users</strong></td>
<td>Clerk, IT professional</td>
<td>Knowledge worker</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>Day to day operations</td>
<td>Decision support</td>
</tr>
<tr>
<td><strong>DB Design</strong></td>
<td>Application-oriented</td>
<td>Subject-oriented</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>Current, up-to-date</td>
<td>Historical, summarized,</td>
</tr>
<tr>
<td></td>
<td>detailed</td>
<td>multidimensional...</td>
</tr>
<tr>
<td><strong>Usage</strong></td>
<td>repetitive</td>
<td>Ad-hoc</td>
</tr>
<tr>
<td><strong>Access</strong></td>
<td>Read/write</td>
<td>Lots of scans</td>
</tr>
<tr>
<td><strong>Unit of work</strong></td>
<td>Short, simple transaction</td>
<td>Complex query</td>
</tr>
<tr>
<td><strong># rec accessed</strong></td>
<td>tons</td>
<td>Millions</td>
</tr>
<tr>
<td><strong># users</strong></td>
<td>thousands</td>
<td>Hundreds</td>
</tr>
<tr>
<td><strong>DB size</strong></td>
<td>100 MB-GB</td>
<td>100 GB-TB</td>
</tr>
<tr>
<td><strong>Metric</strong></td>
<td>Transaction throughput</td>
<td>Query throughput</td>
</tr>
</tbody>
</table>
Why Do we Separate DW From DB?

- Performance reasons:
  - OLAP requires special data organization that supports multidimensional views.
  - OLAP queries would degrade operational DB.
  - OLAP is read only.
  - No concurrency control and recovery.

- Decision support requires historical data.
- Decision support requires consolidated data.

Typical OLAP architecture

Database Design Methodology

- Most data warehouses use a star schema to represent the multi-dimensional model.
- Each dimension is represented by a dimension-table that describes it.
- A fact-table connects to all dimension-tables with a multiple join. Each tuple in the fact-table consists of a pointer to each of the dimension-tables.
- The links between the fact-table in the centre and the dimension-tables form a shape like a star. (Star Schema)

Example of Star Schema

Database Design Methodology (con't)

- Each dimension is represented by one table.
- Un-normalized (introduces redundancy)
  - Ex: (Vancouver, BC, Canada, North America)
    (Victoria, BC, Canada, North America)

Normalize dimension tables ➔ Snowflake Schema

Example of Snowflake Schema
What Is the Best Design??

Performance benchmarking can be determine what is the best design.

- Snowflake schema: Easier to maintain dimension tables when dimension table are very large (reduces overall space).
- Star schema: More effective for data cube browsing (less joins).

Challenges of Materialized views

The challenges in exploiting materialized views are like in using indices.
- Identify the views to materialize.
- Exploit the materialized views to answer queries.
- Efficiently update the materialized views during load and refresh.

The currently industrial solution—Views consist of joins of the fact table with a subset of dimension tables. But it’s a little more complex, if using the snowflake schema.

Why?? → (much joins)

Implementation of the OLAP Server

ROLAP: Relational OLAP - data is stored in tables in relational database or extended-relational databases. They use an RDBMS to manage the warehouse data and aggregations using often a star schema.
- They support extensions to SQL
- Advantage: Scalable.
- Disadvantage: No direct access to cells.

MOLAP: Multidimensional OLAP - implements the multidimensional view by storing data in special multidimensional data structures.
- Advantage: Fast indexing to pre-computed aggregations. Only values are stored.
- Disadvantage: Not very scalable.

Metadata requirements

- Administrative metadata
  - Source database and their contents
  - Back-end and front-end tools
  - Definitions of the warehouse schema
  - Pre-defined queries and reports
  - Data mark locations and contents
  - Data refresh and purging policies
  - User profiles and user access control policies

Metadata requirements

- Business metadata
  - Business terms and definitions
  - Ownership of data
  - Charging policies

- Operational metadata
  - Data lineage: history of migrated data and sequence of transformations applied
  - Currency of data: active, archived, purged
  - Monitoring information: warehouse usage statistics, error reports, audit trails
Discussion

• How does the heterogeneity in data warehouses differ from the topics that we’ve discussed in data integration? What are some applications that you would use data integration for? A data warehouse? Can you think of any applications for which “both” would be a good solution?

• Do you think that materialized views are more important on an RDBMS? Why? Which one do you think is easier to use materialized views in? why?

• Do you think that star schemas are more useful in data warehouses than in RDBMSs? Why or why not?