Main–Memory & Near–Main–Memory OLAP Databases

Wo–Shun Luk
Professor of Computing Science
Simon Fraser University
Outline

• What is OLAP DB?
• How does it work?
  – MOLAP, ROLAP
• Near–Main–Memory DB
  – Partial Pre–computation of Aggregates
• Main–Memory DB
  – Online update of OLAP DB structure
What is OLAP DB?

Data Warehouse

OLAP DB

Data Analysts

Data Mining Algorithms
OLAP Database

- OLAP Database = Fact Table + Aggregates
- A fact table contains $n+1$ columns, i.e. $n$ dimension columns and a measure column.
- Aggregate data are derived from fact table and stored on disk
  - Online data analysis requirements
- Two approaches to treat aggregate data: ROLAP and MOLAP
## An Example of Fact Table

<table>
<thead>
<tr>
<th>Item</th>
<th>Customer</th>
<th>Supplier</th>
<th>Sales $</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>Telus</td>
<td>IBM</td>
<td>18,000</td>
</tr>
<tr>
<td>Disk</td>
<td>Telus</td>
<td>IBM</td>
<td>25,000</td>
</tr>
<tr>
<td>PC</td>
<td>SFU</td>
<td>Dell</td>
<td>14,000</td>
</tr>
<tr>
<td>Disk</td>
<td>BC Hydro</td>
<td>Dell</td>
<td>15,500</td>
</tr>
<tr>
<td>Printer</td>
<td>UBC</td>
<td>Compaq</td>
<td>45,000</td>
</tr>
<tr>
<td>PC</td>
<td>SFU</td>
<td>Compaq</td>
<td>14,000</td>
</tr>
<tr>
<td>Printer</td>
<td>BC Hydro</td>
<td>Compaq</td>
<td>35,200</td>
</tr>
</tbody>
</table>

**Dimensions**: Item, Customer, Supplier  
**Measure**: Sales $
ROLAP Database

• A relational OLAP (ROLAP) database contains a fact table and a set of materialized views
  – Materialized view: the result of a query is pre-computed and stored as another table.
  – Materialized views of ROLAP contained `group by` clauses

• Queries posed to the ROLAP will be sped up due to presence of materialized views.
# ROLAP Materialized Views

<table>
<thead>
<tr>
<th>Customer</th>
<th>Supplier</th>
<th>Sales $</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telus</td>
<td>IBM</td>
<td>43,000</td>
</tr>
<tr>
<td>SFU</td>
<td>Dell</td>
<td>14,000</td>
</tr>
<tr>
<td>BC Hydro</td>
<td>Dell</td>
<td>15,500</td>
</tr>
<tr>
<td>UBC</td>
<td>Compaq</td>
<td>45,000</td>
</tr>
<tr>
<td>SFU</td>
<td>Compaq</td>
<td>14,000</td>
</tr>
<tr>
<td>ICBC</td>
<td>Compaq</td>
<td>35,200</td>
</tr>
</tbody>
</table>

Group-by (Customer, Supplier)

<table>
<thead>
<tr>
<th>Supplier</th>
<th>Sales $</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>43,000</td>
</tr>
<tr>
<td>Dell</td>
<td>29,500</td>
</tr>
<tr>
<td>Compaq</td>
<td>94,200</td>
</tr>
</tbody>
</table>

Group by Supplier
Computation of Materialized Views

• Assumption: database is much larger than main memory available

• Externally sort all tuples in the fact table by some ordering of the attributes, e.g.
  – sort by columns (*Customer, Supplier*)
  – summation on the column *Sales $*
  – discard the column *Item*
MOLAP Database

• Fact table is stored as a multi-dimensional array
  – Each cell is addressed by the values in the $n$ dimension columns
  – Each cell contains a measure value.

• Some or all aggregates, e.g. sums, counts, are pre-computed, and stored somewhere.
Aggregation on a Cube

• Aggregation can be done very efficiently on the multidimensional array.
• Implementation of the multidimensional array:
  – The cube is folded into a single–column array
  – Each cell is addressed by its offset from the top of the array
The cube
MOLAP vs. ROLAP

• Downsides of MOLAP
  – Need very large address space
    • 64–bit word may not be sufficient for a large number of dimensions
  – Not feasible unless enough main memory is available

• Downsides of ROLAP
  – Need multiple passes in aggregate pre–computation
  – Example: to compute all aggregates, it needs \( n \) passes at the minimum, where \( n \) is the number of dimensions
  – There will be more passes if dimension hierarchy is considered.
Running Time

Fact Table Size

Fixed Memory Size
Example of Dimension Hierarchies

Location Dimension

- On-Campus
  - Day
  - Evening
- Off-Campus
  - Distance Ed.
  - Harbor Centre

Semester Dimension

- All
  - Fall
  - Spring
  - Summer

Division Dimension

- All
  - Graduate
  - Undergraduate
    - Lower Division
    - Upper Division
Variable Memory Size

• If memory increases,
  – the MOLAP algorithm runs on larger databases
  – the ROLAP algorithm runs faster

• Given the database, and sufficient memory, MOLAP algorithms will beat ROLAP algorithms.
OLAP DB Research

- Most research papers focus on ROLAP
- Most OLAP products adopt MOLAP strategy, or a mixed one.
- Most MOLAP/ROLAP algorithms do not consider dimension hierarchies higher than 2 levels.
Near–Main–Memory OLAP DB

• The OLAP DB is still disk resident
• Main memory is no longer the overriding limitation for algorithm developments
• NMM OLAP adopts MOLAP strategy
• Assume dimension hierarchies of arbitrary height
Why?

• Upward scalability is still important from the view point of CS, but …

• Cost of main memory decreases rapidly
  – 1G (One gigabyte) memory goes for < US$100

• OLAP problem size in general does not increase as rapidly
  – One TPC–W benchmark database is 300G in size
MOLAP Research

• How to overcome the address space size limitation:
  – break the key, i.e. the offset, in multiple parts, so that each part is within the allowable address space, e.g. 32-bit address space
  – use compression

• One main goal: to develop algorithms that minimize the use of main memory
Partial Aggregate Pre-computation

• Compute in advance only a fraction of the aggregates
  – the rest of them are computed during query processing time on when-needed basis

• How large should the fraction be?

• Which aggregates should be selected for pre-computation?
Main Memory OLAP DB

- What can one do with unlimited memory?
- Why has MM DBMS failed?
  - Audit trail requirements
  - MM DBMS is not scalable enough
- Big memory in DBMS means only big buffer
- MM OLAP DB is good only when there are useful features that are not feasible for disk-based OLAP DB
Update OLAP Dimensions

• Data analyst may want to compare different classification schemes – “What if” type of questions, e.q.
  – what if MACM courses re–labelled as CMPT or MATH courses
  – what if the year–semester dimension hierarchy be broken into two dimensions: year and semester.
Update OLAP DB

- Assume the OLAP DB is memory resident
- Research questions: If the OLAP dimension is updated,
  - should we re–compute all aggregates inside the OLAP DB (i.e. re–generate the cube), or
  - should we just update the affected aggregates?
Conclusions

• There should be more research on MOLAP, in review of low cost memory.
• A different class of OLAP DB algorithms need to be developed that run much faster than the existing ‘scalable’ algorithms, given sufficient memory.
  – The idea of Near–Main–Memory OLAP DB
• Main–Memory OLAP DB allows online display of multiple views of a OLAP DB.