Benchmarking Summarizability Processing in XML Warehouses with Complex Hierarchies

By Chantola KIT
Marouane HACHICHA
Jérôme DARMONT
Decision Making:

1. **Business Intelligence (BI)** is famed for complex analysis
   - OLAP is a notable BI tool for multi-dimensional analysis

2. **DWS**: collection of historical and concurrent data
   - XML is widely used to represent *complex hierarchical* data
Effectiveness of Summarizability processing on complex hierarchies

- Benchmarks are used to support performance evaluation
- Existing XML data warehouse benchmark: XWeB
  - Complex hierarchies are not scalable
XML Data Example

Sales
  └── sale
       └── sale#1
            ├── part
            │     └── Part#1
            │          └── type3
            │               └── LARGE
            │                   └── type2
            │                       └── PLATE
            │                            └── type1
            │                                └── TIN
            └── customer
                 └── Customer#1
                      ├── nation
                      │     └── USA
                      └── region
                           └── AMERICA
            └── supplier
                 └── Supplier#1
                      ├── date
                      │     └── 25/06/1998
                      └── f_quantity
                           └── 100
            └── f_totalamount
                 └── 2,800
Non-Strict Hierarchies

- Supplier#1 is located in Europe and Africa;
- Europe contains two suppliers: #1 and #2
- Total quantity supplied by Europe is 200 (wrong)
Incomplete Hierarchies

- Part#1 has no type3 (LARGE) level
- Total quantity of PLATE or TIN part is 0 (wrong)
Related Work

- Relational Decision Support Benchmark
  - TPC: TPC-H and TPC-DS [TPPC'12]
  - SSB [VLDB/TPCTC’09]
  - DWEB [IJBIDM’07]

- XML benchmarks: Michigan [VLDB’02], MemBer [SIGMOD’05], X-Mach, XMark [VLDB/EEXTT’02], XOO7[CIKM’01], and XBench [ICDE’04]

- XML decision support benchmarks: XWeB [VLDB/TPCTC’10]
  - Only one complex hierarchy workload
  - Complexity lies only on part-category dimension
  - Query on complex hierarchies is limited
  - Complex hierarchy is not scalable
Objective

Extending XWeB with:

- Scalable complex hierarchies
- Summarizability processing
Benchmarking Summarizability Processing in XML Warehouses with Complex Hierarchies

Data Model

```
Sales
  -*
  |   sale
    +   ?
  +____-__-
part   customer   supplier   date   f_quantity   f_totalamount
  *     *         *         ?     ?          ?
  type3 nation     nation     day   ?          
  *     ?         ?         ?     month     ?
  type2 region     region     ?     ?          year
  *     ?         ?         ?     ?          ?
  type1
```

?: 0-1 (incomplete)
-: 1 only (simple)
*: 0-many (complex)
+: 1-many (non-strict)
Generating Incomplete Hierarchies

Randomly delete *ip* hierarchical levels

- *ip*: incomplete percentage

![Diagram]

Type3 level of Part#1 is randomly deleted
Generating Non-strict Hierarchies

Randomly generate $np$ non-strict hierarchies

- $np$: non-strict percentage

1. Randomly generate an array of $n$ non-strict hierarchies
   - $n$: number of non-strict hierarchies. Ex. $n = 4$

2. Convert the array into Hierarchical XML Data
1. Generate \( n \)-non-strict array (as in slide #12)
2. Randomly delete some levels from non-strict array
3. Convert the array into Hierarchical XML Data

**Generating Complex Hierarchies**

<table>
<thead>
<tr>
<th>4-non-strict-hierarchy array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier#1</td>
</tr>
<tr>
<td>Supplier#2</td>
</tr>
<tr>
<td>Supplier#1</td>
</tr>
<tr>
<td>Supplier#2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>complex-hierarchy array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier#1</td>
</tr>
<tr>
<td>Supplier#2</td>
</tr>
<tr>
<td>Supplier#1</td>
</tr>
<tr>
<td>Supplier#2</td>
</tr>
</tbody>
</table>

sale#1

```plaintext
supplier#1
  | FRANCE
  | EUROPE
  | ALGERIA
  | AFRICA

supplier#2
  | ASIA
  | GERMANY
```
## Query Workload

<table>
<thead>
<tr>
<th>Q21</th>
<th>Q23</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum of $f_quantity$, $f_totalamount$</td>
<td>max of $f_totalamount$</td>
</tr>
<tr>
<td>from part, customer, supplier, date</td>
<td>from date, part, supplier, customer</td>
</tr>
<tr>
<td>group by part, customer, supplier, date</td>
<td>group by month, type2, nation, region</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q22</th>
<th>Q24</th>
</tr>
</thead>
<tbody>
<tr>
<td>min of $f_quantity$</td>
<td>average of $f_totalamount$</td>
</tr>
<tr>
<td>from customer, part, supplier, date</td>
<td>from supplier, part, customer, date</td>
</tr>
<tr>
<td>group by nation, type3, nation, day</td>
<td>group by region, type1, region, year</td>
</tr>
</tbody>
</table>
Performance Metrics

- Quantitative metric: response time; the execution time of the query workload
- Qualitative metric: verifying the result whether the summarizability issues are correctly handled
  - Resulted groups are not duplicated
  - Total of aggregation values is equal to grand total
  - Average value is the division of total and its number
  - Min is the least value
  - Max is the highest value
Experimental Study

Summarizability processing using:
- Our proposed approach: Query Based Approach (QBS) [COMAD’12]
- Previous approach: Pedersen’s approach (Pedersen) [VLDB’99]
## Experimental Study (Cont.)

### Dataset size (KB)

<table>
<thead>
<tr>
<th>No. Facts</th>
<th>50,000</th>
<th>100,000</th>
<th>150,000</th>
<th>200,000</th>
<th>250,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>27,700</td>
<td>55,390</td>
<td>82,800</td>
<td>110,577</td>
<td>138,015</td>
</tr>
<tr>
<td>Incomplete 5%</td>
<td>27,626</td>
<td>55,242</td>
<td>82,543</td>
<td>110,249</td>
<td>137,573</td>
</tr>
<tr>
<td>Non-strict 5%</td>
<td>28,669</td>
<td>57,328</td>
<td>85,671</td>
<td>114,422</td>
<td>142,786</td>
</tr>
<tr>
<td>Complex 5%</td>
<td>28,376</td>
<td>56,742</td>
<td>85,791</td>
<td>113,252</td>
<td>141,319</td>
</tr>
<tr>
<td>Incomplete 50%</td>
<td>25,020</td>
<td>50,030</td>
<td>74,769</td>
<td>99,842</td>
<td>124,601</td>
</tr>
<tr>
<td>Non-strict 50%</td>
<td>35,412</td>
<td>70,826</td>
<td>105,914</td>
<td>141,397</td>
<td>176,527</td>
</tr>
<tr>
<td>Complex 50%</td>
<td>32,522</td>
<td>65,031</td>
<td>97,263</td>
<td>129,839</td>
<td>162,088</td>
</tr>
</tbody>
</table>
Exp. Results of Simple Hierarchy Grouping

![Bar chart showing the time in milliseconds (ms) for different numbers of facts and hierarchies, comparing QBS, Pedersen without Overhead, and Pedersen with Overhead.]
Exp. Results of QBS Simple Hierarchy Group Matching

- QBS without Overhead, without Group Matching
- QBS with Overhead, without Group Matching
- QBS with Overhead, with Group Matching

Benchmarking Summarizability Processing in XML Warehouses with Complex Hierarchies
Exp. Results of Pedersen Simple Hierarchy Group Matching

![Graph showing time in ms versus number of facts for different scenarios.]

- **Pedersen without Overhead, without Group Matching**
- **Pedersen without Overhead, with Group Matching**
- **Pedersen with Overhead, with Group Matching**

Benchmarking Summarizability Processing in XML Warehouses with Complex Hierarchies
Exp. Results of Complex Hierarchy Grouping

5%

7,000
70,000
700,000
7,000,000

1D 2D 3D 1D 2D 3D 1D 2D 3D 1D 2D 3D

50,000 100,000 150,000 200,000 250,000

Time (ms)

Number of Facts

QBS Pedersen without Overhead Pedersen with Overhead

5% 50%

Benchmarking Summarizability Processing in XML Warehouses with Complex Hierarchies
Exp. Results of QBS Complex Hierarchy Grouping

5%

50%

Incomplete  Non-strict  Complex
Conclusion

- First XML data warehouse benchmark with complex hierarchies
- Conform to Gray’s criteria: relevance, portability, scalability, and simplicity
- Experimentation addressing summarizability processing:
  - Run-time summarizability management is feasible
  - Run-time of group matching process is still costly
- **Future work:**
  - Improve group matching process
  - Integrate with previous XML benchmarks: XWeB
QUESTIONS?

chantola.kit@univ-lyon2.fr
marouane.hachicha@univ-lyon2.fr
jerome.darmont@univ-lyon2.fr

Benchmark preliminary version:
http://eric.uni-lyon2.fr/~ckit/DOLAP12.zip