Parallelizing Multiple Group by Query in Shared-nothing Environment: A MapReduce Study Case

PAN Jie\(^1\)  Yann LE BIANNIC \(^2\)  Frédéric MAGOULES \(^1\)

\(^1\)Ecole Centrale Paris-Applied Mathematics and Systems Department (MAS)

\(^2\)SAP BusinessObjects Innovation Center

June 22, 2010
Outline

1. Introduction
2. Related works
3. Multiple Group by Query
4. Data Preparation
5. MapReduce jobs
6. Experimental Results
7. Cost Estimation Model
8. Conclusions and Future works
Outline

1. Introduction
2. Related works
3. Multiple Group by Query
4. Data Preparation
5. MapReduce jobs
6. Experimental Results
7. Cost Estimation Model
8. Conclusions and Future works
Introduction

Objective:
- **Cheap** solution for multidimensional data analysis applications

Challenges:
- Large data volume
- Short response time (ex. within 5 seconds)
- Use cheap commodity computers → scalability and fault-tolerance

Advantages of MapReduce:
- Automatic scalability and fault-tolerance
- Minimize the need of locking
- Wide range of applicative environments: cluster, multi-core hardware,
Related works

- MapReduce's implementations: GridGain, Hadoop...
- Data query languages based on MapReduce: PigLatin, HiveQL
- MapReduce-based analytic data warehouses: Greenplum, Aster
- Combination of MapReduce and parallel databases: HadoopDB

This work focuses on:
- MapReduce-based multiple group by query
Outline

1. Introduction
2. Related works
3. Multiple Group by Query
4. Data Preparation
5. MapReduce jobs
6. Experimental Results
7. Cost Estimation Model
8. Conclusions and Future works
Multiple group-by query’s form:

- select X, SUM(*), from R where condition group by X

Example:

- select B, sum(A) from R where \( I > i \) group by B;
- select C, sum(A) from R where \( I > i \) group by C;
- select D, sum(A) from R where \( I > i \) group by D;
- select E, sum(A) from R where \( I > i \) group by E;
- select F, sum(A) from R where \( I > i \) group by F;
- ...
Outline

1. Introduction
2. Related works
3. Multiple Group by Query
4. Data Preparation
5. MapReduce jobs
6. Experimental Results
7. Cost Estimation Model
8. Conclusions and Future works
Dataset used in multidimensional data analysis applications

<table>
<thead>
<tr>
<th>Dimension1</th>
<th>Dimension2</th>
<th>Dimension3</th>
<th>Dimension4</th>
<th>Dimension5</th>
<th>Dimension6</th>
<th>Measure1</th>
<th>Measure2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>Product</td>
<td>ProdCode</td>
<td>Store</td>
<td>StoreState</td>
<td>Opening Year</td>
<td>Sold</td>
<td>Revenue</td>
</tr>
<tr>
<td>Text</td>
<td>Text</td>
<td>Text</td>
<td>Text</td>
<td>Text</td>
<td>Text</td>
<td>Numeric</td>
<td>Numeric</td>
</tr>
<tr>
<td>Color1</td>
<td>Product12</td>
<td>F20</td>
<td>Store1</td>
<td>StoreState1</td>
<td>1997</td>
<td>20</td>
<td>201.50</td>
</tr>
<tr>
<td>Color1</td>
<td>Product7</td>
<td>F60</td>
<td>Store2</td>
<td>StoreState2</td>
<td>1998</td>
<td>100</td>
<td>10808.00</td>
</tr>
<tr>
<td>Color1</td>
<td>Product30</td>
<td>F90</td>
<td>Store1</td>
<td>StoreState1</td>
<td>1997</td>
<td>500</td>
<td>905.80</td>
</tr>
<tr>
<td>Color2</td>
<td>Product30</td>
<td>F90</td>
<td>Store2</td>
<td>StoreState2</td>
<td>1998</td>
<td>675</td>
<td>1120.90</td>
</tr>
<tr>
<td>Color2</td>
<td>Product4</td>
<td>F16</td>
<td>Store1</td>
<td>StoreState1</td>
<td>1997</td>
<td>505</td>
<td>4506.90</td>
</tr>
<tr>
<td>Color2</td>
<td>Product55</td>
<td>F50</td>
<td>Store2</td>
<td>StoreState2</td>
<td>1998</td>
<td>802</td>
<td>3960.50</td>
</tr>
<tr>
<td>Color2</td>
<td>Product6</td>
<td>F36</td>
<td>Store3</td>
<td>StoreState1</td>
<td>1997</td>
<td>1890</td>
<td>18506.80</td>
</tr>
<tr>
<td>Color3</td>
<td>Product2</td>
<td>F65</td>
<td>Store2</td>
<td>StoreState2</td>
<td>1998</td>
<td>2086</td>
<td>56008.90</td>
</tr>
<tr>
<td>Color3</td>
<td>Product2</td>
<td>F65</td>
<td>Store3</td>
<td>StoreState1</td>
<td>1997</td>
<td>3809</td>
<td>112017.80</td>
</tr>
</tbody>
</table>
Dataset used in multidimensional data analysis applications

```
Select sum(Revenue)
from R group by Color;
```

<table>
<thead>
<tr>
<th>Dimension1</th>
<th>Dimension2</th>
<th>Dimension3</th>
<th>Dimension4</th>
<th>Dimension5</th>
<th>Dimension6</th>
<th>Measure1</th>
<th>Measure2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>Product</td>
<td>ProdCode</td>
<td>Store</td>
<td>StoreState</td>
<td>Opening Year</td>
<td>Sold</td>
<td>Revenue</td>
</tr>
<tr>
<td>Text</td>
<td>Text</td>
<td>Text</td>
<td>Text</td>
<td>Text</td>
<td>Text</td>
<td>Numeric</td>
<td>Numeric</td>
</tr>
<tr>
<td>Color1</td>
<td>Product12</td>
<td>F20</td>
<td>Store1</td>
<td>StoreState1</td>
<td>1997</td>
<td>20</td>
<td>201.50</td>
</tr>
<tr>
<td>Color1</td>
<td>Product7</td>
<td>F60</td>
<td>Store2</td>
<td>StoreState2</td>
<td>1998</td>
<td>100</td>
<td>10808.00</td>
</tr>
<tr>
<td>Color1</td>
<td>Product30</td>
<td>F90</td>
<td>Store1</td>
<td>StoreState1</td>
<td>1997</td>
<td>500</td>
<td>905.80</td>
</tr>
<tr>
<td>Color2</td>
<td>Product30</td>
<td>F90</td>
<td>Store2</td>
<td>StoreState2</td>
<td>1998</td>
<td>675</td>
<td>1120.90</td>
</tr>
<tr>
<td>Color2</td>
<td>Product4</td>
<td>F16</td>
<td>Store1</td>
<td>StoreState1</td>
<td>1997</td>
<td>505</td>
<td>4506.90</td>
</tr>
<tr>
<td>Color2</td>
<td>Product55</td>
<td>F50</td>
<td>Store2</td>
<td>StoreState2</td>
<td>1998</td>
<td>802</td>
<td>3960.50</td>
</tr>
<tr>
<td>Color2</td>
<td>Product6</td>
<td>F36</td>
<td>Store3</td>
<td>StoreState1</td>
<td>1997</td>
<td>1890</td>
<td>18506.80</td>
</tr>
<tr>
<td>Color3</td>
<td>Product2</td>
<td>F65</td>
<td>Store2</td>
<td>StoreState2</td>
<td>1998</td>
<td>2086</td>
<td>56008.90</td>
</tr>
<tr>
<td>Color3</td>
<td>Product2</td>
<td>F65</td>
<td>Store3</td>
<td>StoreState1</td>
<td>1997</td>
<td>3809</td>
<td>112017.80</td>
</tr>
</tbody>
</table>
Data partitioning

Horizontal Partitioning
- Putting different entire rows into different partitions.
- In our work: 1 horizontal partition = a certain number of entire rows (13 Dimensions + 2 Measures).

Vertical Partitioning
- Putting different columns into different partitions.
- In our work: 1 vertical partition = 1 Dimension + 2 Measures.
Data Distribution

With horizontal partitioning:

- Each worker node holds equal number of partitions (in this work, 1, 2, 10 and 20 partitions/worker)

With vertical partitioning:

- Small worker number: replicate all partitions over every worker node
- Large worker number:
  1. partitions are divided into regions
  2. worker nodes are grouped into regions
  3. replicate the vertical partitions over nodes within region (hybrid partitioning)
Data Distribution under Vertical Partitioning

Figure: 3 partitions in 1 region vs 3 partitions in 3 regions
Indexation: Lucene Inverted Index

Figure: Lucene Index
Outline

1. Introduction
2. Related works
3. Multiple Group by Query
4. Data Preparation
5. MapReduce jobs
6. Experimental Results
7. Cost Estimation Model
8. Conclusions and Future works
GridGain’s MapReduce Framework

Multiple mappers one reducer

**Figure:** GridGain’s MapReduce framework
Figure: Data flow of GridGain MapReduce
Master’s Processings: Start-up and Closure

A MR task thread starts

Create mappings between mappers and available worker nodes

Serialize mappers and arguments

Send out mappers to worker nodes

MR task thread ends GridGain instance “wait”

Activated by receiving sub-result(s)

De-serialize sub-result

More sub-results to de-serialize?

Yes

Execute reducer

No

End

Waiting

Waiting

startup
closure

Figure: Start-up and closure processing on Master node
Figure: The processing on worker node
Mapper’s Calculations under Horizontal Partitioning

Mapper: Filter + Aggregate over all the group by columns \(\rightarrow\) List<FacetCollector$^1$>

Original data block:

<table>
<thead>
<tr>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>...</th>
<th>D13</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

13 dimensions
2 measures
1000K rows

Filtered data block:

<table>
<thead>
<tr>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>...</th>
<th>D13</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\checkmark)</td>
<td>d11</td>
<td>d25</td>
<td></td>
<td></td>
<td>m1</td>
<td>m2</td>
</tr>
<tr>
<td>(\checkmark)</td>
<td>d11</td>
<td>d25</td>
<td></td>
<td></td>
<td>m3</td>
<td>m4</td>
</tr>
<tr>
<td>(\checkmark)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\checkmark)</td>
<td>d12</td>
<td>d26</td>
<td></td>
<td></td>
<td>m5</td>
<td>m6</td>
</tr>
<tr>
<td>(\checkmark)</td>
<td>d12</td>
<td>d24</td>
<td></td>
<td></td>
<td>m7</td>
<td>m8</td>
</tr>
</tbody>
</table>

\(\checkmark\) indicates the dimensions that are included in the query.

$^1$ FacetCollector: aggregated values over 1 column’s distinct values
Mapper’s Calculations under Horizontal Partitioning

Mapper: Filter + Aggregate over all the group by columns → List<FacetCollector>¹

---

¹ FacetCollector is a set of aggregated measure values over 1 dimension’s distinct values
Reducer’s Calculations under Horizontal Partitioning

Reducer: aggregate over `List<List<FacetCollector>>`

<table>
<thead>
<tr>
<th>Sub-result_1</th>
<th>Sub-result_2</th>
<th>Sub-result_n</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D1</strong></td>
<td><strong>Sum(M1)</strong></td>
<td><strong>Sum(M2)</strong></td>
</tr>
<tr>
<td><strong>D2</strong></td>
<td><strong>Sum(M1)</strong></td>
<td><strong>Sum(M2)</strong></td>
</tr>
<tr>
<td><strong>D3</strong></td>
<td><strong>Sum(M1)</strong></td>
<td><strong>Sum(M2)</strong></td>
</tr>
<tr>
<td>D4</td>
<td>Sum(M1)</td>
<td>Sum(M2)</td>
</tr>
<tr>
<td>D3</td>
<td>Sum(M1)</td>
<td>Sum(M2)</td>
</tr>
<tr>
<td>D3</td>
<td>Sum(M1)</td>
<td>Sum(M2)</td>
</tr>
<tr>
<td>D4</td>
<td>Sum(M1)</td>
<td>Sum(M2)</td>
</tr>
<tr>
<td>d45</td>
<td>ma</td>
<td>md</td>
</tr>
<tr>
<td>d43</td>
<td>mb</td>
<td>me</td>
</tr>
<tr>
<td>d44</td>
<td>mc</td>
<td>mf</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Aggregation order:

- D1
- D2
- D3
- D4
- d45
- d43
- d44
- ...
Mapper’s Calculations under Vertical Partitioning

Filter + Aggregate over one group by column $\rightarrow$ FacetCollector

**Original data block:**

<table>
<thead>
<tr>
<th>D1</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 dimension
2 measures
10000K rows

**Filtered data block:**

<table>
<thead>
<tr>
<th>D1</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>d11</td>
<td>m1</td>
<td>m2</td>
</tr>
<tr>
<td>d11</td>
<td>m3</td>
<td>m4</td>
</tr>
<tr>
<td>d12</td>
<td>m5</td>
<td>m6</td>
</tr>
<tr>
<td>d12</td>
<td>m7</td>
<td>m8</td>
</tr>
</tbody>
</table>

1 dimension
2 measures
15K rows

scan + aggregate

<table>
<thead>
<tr>
<th>D1</th>
<th>Sum(M1)</th>
<th>Sum(M2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>d11</td>
<td>m1+m3</td>
<td>m2+m4</td>
</tr>
<tr>
<td>d12</td>
<td>m5+m7</td>
<td>m6+m8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Reducer: aggregate over $\text{List<FacetCollector>}$

Reducer’s Calculations under Vertical Partitioning

**Sub_result_1**

<table>
<thead>
<tr>
<th>$D_1$</th>
<th>Sum($M_1$)</th>
<th>Sum($M_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{11}$</td>
<td>$m_a$</td>
<td>$m_d$</td>
</tr>
<tr>
<td>$d_{12}$</td>
<td>$m_b$</td>
<td>$m_g$</td>
</tr>
<tr>
<td>$d_{13}$</td>
<td>$m_c$</td>
<td>$m_f$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Sub_result_2**

<table>
<thead>
<tr>
<th>$D_2$</th>
<th>Sum($M_1$)</th>
<th>Sum($M_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{21}$</td>
<td>$m_h$</td>
<td>$m_k$</td>
</tr>
<tr>
<td>$d_{22}$</td>
<td>$m_i$</td>
<td>$m_l$</td>
</tr>
<tr>
<td>$d_{23}$</td>
<td>$m_j$</td>
<td>$m_m$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Sub_result_3**

<table>
<thead>
<tr>
<th>$D_3$</th>
<th>Sum($M_1$)</th>
<th>Sum($M_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{31}$</td>
<td>$m_{n}$</td>
<td>$m_q$</td>
</tr>
<tr>
<td>$d_{32}$</td>
<td>$m_{o}$</td>
<td>$m_{r}$</td>
</tr>
<tr>
<td>$d_{33}$</td>
<td>$m_{p}$</td>
<td>$m_{s}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Sub_result_4**

<table>
<thead>
<tr>
<th>$D_1$</th>
<th>Sum($M_1$)</th>
<th>Sum($M_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{11}$</td>
<td>$m_{a'}$</td>
<td>$m_{d'}$</td>
</tr>
<tr>
<td>$d_{12}$</td>
<td>$m_{b'}$</td>
<td>$m_{g'}$</td>
</tr>
<tr>
<td>$d_{13}$</td>
<td>$m_{c'}$</td>
<td>$m_{f'}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Sub_result_5**

<table>
<thead>
<tr>
<th>$D_2$</th>
<th>Sum($M_1$)</th>
<th>Sum($M_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{21}$</td>
<td>$m_{h'}$</td>
<td>$m_{k'}$</td>
</tr>
<tr>
<td>$d_{22}$</td>
<td>$m_{i'}$</td>
<td>$m_{l'}$</td>
</tr>
<tr>
<td>$d_{23}$</td>
<td>$m_{j'}$</td>
<td>$m_{m'}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Sub_result_6**

<table>
<thead>
<tr>
<th>$D_3$</th>
<th>Sum($M_1$)</th>
<th>Sum($M_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{31}$</td>
<td>$m_{n'}$</td>
<td>$m_{q'}$</td>
</tr>
<tr>
<td>$d_{32}$</td>
<td>$m_{o'}$</td>
<td>$m_{r'}$</td>
</tr>
<tr>
<td>$d_{33}$</td>
<td>$m_{p'}$</td>
<td>$m_{s'}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
### Outline

1. Introduction
2. Related works
3. Multiple Group by Query
4. Data Preparation
5. MapReduce jobs
6. Experimental Results
7. Cost Estimation Model
8. Conclusions and Future works
Environment Configuration

**Softwares:**
- GridGain version 2.1.1
- Java version 1.6.0
- JVM heap maximum 1536M

**Hardwares:**
- a Grid5000’s Sophia site
- CPU 2 single-core at 2.0GHz
- Memory 2GB RAM
- Disk 80GB

**Data set**
- 1.2GB
- 10,000,000 rows
- 15 columns = 13 D + 2 M

**Data preparations**
- Horizontal partitioning or vertical partitioning
- Indexed with Lucene

**MG-Query selectivities:**
- 1.05%, 9.96%, 18.6%, 43.1%

**Aggregate over:**
- 5 group by columns

---

*Grid5000’s Sophia site*

An infrastructure distributed in 9 sites around France, for research in large-scale parallel and distributed systems (https://www.grid5000.fr/)
Speed-up Measurements: MapReduce-based IMPL. over Horizontally Partitioned Data

<table>
<thead>
<tr>
<th>Query</th>
<th>Execution time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0105</td>
<td>869</td>
</tr>
<tr>
<td>0.0996</td>
<td>1259</td>
</tr>
<tr>
<td>0.186</td>
<td>2097</td>
</tr>
<tr>
<td>0.431</td>
<td>4098</td>
</tr>
</tbody>
</table>

Observations:
- lrg. selectivity > sm. selectivity
- sm. nb\_job/node > lrg. nb\_job/node

Figure: Speed-up with horizontally partitioned data without combiner.
Speed-up Measurements: MapCombineReduce-based IMPL. over Horizontally Partitioned data

About combiner:
Function: pre-aggregating

Advantage: reduce communication cost

Observations:
For sm. job no. (eg. 1, 2), MR > MCR
For lrg. job no. (eg. 10, 20), MCR > MR

Figure: Speed-up with horizontally partitioned data with combiner.
Speed-up measurements: Vertically Partitioned Data

Figure: Speed-up with vertically partitioned data.
Speed-up measurements: Vertically Partitioned Data

Observations:

- speed-up ↗ when worker nb ↗
- MR-based IMPL. > MCR-based IMPL.
- lrg. selectivity queries > sm. selectivity queries
Performance Affecting Factors

Selectivity decides the workload of aggregation.
When // aggregation, lrg. selectivity queries > sm. selectivity queries

\(nb_{job}/node\)
Running multiple mappers on one node may degrade performance because of resource contention.

<table>
<thead>
<tr>
<th>Job number on 1 node</th>
<th>1 Mapper’s Average Execution time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>170</td>
</tr>
<tr>
<td>2</td>
<td>208</td>
</tr>
<tr>
<td>3</td>
<td>354</td>
</tr>
<tr>
<td>4</td>
<td>417</td>
</tr>
<tr>
<td>5</td>
<td>537</td>
</tr>
</tbody>
</table>

Table:
Query selectivity=0.01, vertical partitioning, 3 regions
Outline

1. Introduction
2. Related works
3. Multiple Group by Query
4. Data Preparation
5. MapReduce jobs
6. Experimental Results
7. Cost Estimation Model
8. Conclusions and Future works
Cost Estimation Model (MapReduce-based IMPL.)

**Startup cost (Master)**
- Mapping jobs to nodes: \( C_m \times nb_{mapper} \)
- Serialization: \( C_s \times size_{mapper} \times nb_{mapper} \)

\[
Cost_{startup} = C_m \cdot nb_{mapper} + C_s \cdot size_{mapper} \cdot nb_{mapper}
\]

**Each Mapper Job (Worker)**
- De-serialization: \( C_d \times size_{mapper} \)
- Mapper cost: \( Cost_{mapper} \)
- Serialization: \( C_s \cdot size_{intermediate} \)

\[
Cost_{worker} = f\left(\frac{nb_{mapper}}{node}\right) \cdot \left( C_d \cdot size_{mapper} + Cost_{mapper} + C_s \cdot size_{intermediate}\right)
\]

**Closure cost (Master)**
- De-serialization: \( C_d \cdot \sum_{i=1}^{nb_{mapper}} size_{intermediate_i} \)
- Reducer: \( Cost_{reducer} \)

\[
Cost_{closure} = C_d \cdot \sum_{i=1}^{nb_{mapper}} size_{intermediate_i} + Cost_{reducer}
\]

**Communication**

\[
Cost_{comm} = C_n \cdot (nb_{mapper} \cdot size_{mapper} + \sum_{i=1}^{nb_{mapper}} size_{intermediate_i})
\]
Cost Estimation for Horizontal Partitioning-based IMPL.

Mapper Cost

\[ \text{Cost}_{\text{mapper}} = \text{Selectivity} \cdot \text{size}_{\text{block}} \cdot \left[ C_{\text{slct}} + (D + M) \cdot C_{\text{read}} + M \cdot nb_{\text{GB}} \cdot C_{\text{add}} \right] \]

Intermediate output size

\[ \text{size}_{\text{intermediate}} = \sum_{i=1}^{nb_{\text{GB}}} nb_{DV_i} \cdot \text{size}_{\text{aggregate}} \]

Reducer cost

\[ \text{Cost}_{\text{reducer}} = C_{\text{add}} \cdot M \cdot nb_{\text{block}} \cdot \sum_{i=1}^{nb_{\text{GB}}} nb_{DV_i} \]

Total cost

\[ \text{Cost}_{hp} = \text{Selectivity} \cdot \text{size}_{\text{block}} \cdot f\left(\frac{nb_{\text{mapper}}}{\text{node}}\right) \cdot \left[ C_{\text{slct}} + (M + D) \cdot C_{\text{read}} + M \cdot nb_{\text{GB}} \cdot C_{\text{add}} \right] + \sum_{i=1}^{nb_{\text{GB}}} nb_{DV_i} \cdot \left[ f\left(\frac{nb_{\text{mapper}}}{\text{node}}\right) \cdot (C_s + C_d + C_n) \cdot \text{size}_{\text{aggregate}} + C_{\text{add}} \cdot M \cdot nb_{\text{block}} \right] \]
Cost Estimation for Vertical Partitioning-based IMPL.

Mapper cost

\[ Cost_{mapper} = \text{Selectivity} \cdot \text{size}_{region} \cdot \left[ C_{slct} + (M + 1)C_{read} + M \cdot C_{add} \right] \]

Intermediate output size

\[ \text{size}_{intermediate} = \text{nb}_{DV_{currentGB}} \cdot \text{size}_{aggregate} \]

Reducer cost

\[ Cost_{reducer} = C_{add} \cdot M \cdot \text{nb}_{region} \cdot \sum_{i=1}^{nb_{GB}} \text{nb}_{DV_{i}} \]

Total cost

\[ Cost_{vp} = f(\text{nb}_{mapper/\text{node}}) \cdot \text{Selectivity} \cdot \text{size}_{region} \cdot \left[ C_{slct} + (M + 1) \cdot C_{read} + M \cdot C_{add} \right] \cdot f(\text{nb}_{mapper/\text{node}}) \cdot C_s \cdot \text{nb}_{DV_{currentGB}} \cdot \text{size}_{aggregate} + (C_d + C_{add} \cdot M + C_n) \cdot \text{nb}_{region} \cdot \sum_{i=1}^{nb_{GB}} \text{nb}_{DV_{i}} \]
Conclusions

- We realized multiple group by query with MapReduce (GridGain).
- Prepare data by horizontal and vertical partitioning and indexing.
- Measure speedup performance on Grid'5000.
- Formal cost analysis.
- Larger job number $\neq$ better performance.
- Performance affecting factors: selectivity and $nb_{job}/node$
Future works

- Find more performance affecting factors (e.g. filtered data distribution).
- Intermediate data compression.
- Separate the index searching from filter and aggregate; Determine mapper number on the fly, considering query selectivity.
- $nb_{job}$/node factor to be examined on multi-core hardware.