Towards an open collective knowledge base
in testing results

Lahcène BRAHIMI

Lahcene.brahimi@ensma.fr

www.lias-lab.fr/members/lahcenebrahimi

Sous la direction de:
M. Ladjel BELLATRECHE
Agenda

1. Context and problematic
2. Initiative of DBMS selection
3. Testing – Types and methods
4. Our test repository
   – Storage - User interface
5. Searching usage of our test repository
   - Recommender system – Algorithm - Usage
Context and problematic

![Diagram showing the models: Modèle Conceptuel -> Modèle Logique -> Modèle Physique -> Modèle Déploiement]

<table>
<thead>
<tr>
<th>Phase</th>
<th>Specification</th>
<th>Criteria</th>
<th>Evaluation tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual design</td>
<td>- Requirements, data analysis, modeling, workloads</td>
<td>- Security, quality, understandability (usability)</td>
<td>- Expert &amp; End-users (Syntax checking, mini DB, back to users, validation rules)</td>
</tr>
<tr>
<td>Deployment design (DBMS selection)</td>
<td>- System cost, features, portability, requirements, Hardware</td>
<td>- Performance</td>
<td></td>
</tr>
<tr>
<td>Logical design</td>
<td>- Data structure, data type, attribute domain</td>
<td>- Normalization</td>
<td>- Prototype check</td>
</tr>
<tr>
<td>Physical design</td>
<td>- Hardware, storage structure, access methods</td>
<td>- Performance, response time, energy consumption</td>
<td>- Cost model, benchmarking</td>
</tr>
<tr>
<td>Implementation</td>
<td>- Special storage, storage group, data files, data loading</td>
<td>- Performance, integrity, concurrent access, security</td>
<td>- Tuning (Integrated tools in the DBMS)</td>
</tr>
<tr>
<td>Exploitation</td>
<td>- New data, access by users, new business requirements</td>
<td>- Maintenance</td>
<td>- Audit, tuning (Integrated tools in the DBMS)</td>
</tr>
</tbody>
</table>

[Golfarelli 11, Jenkins 08, Bouarar 15]
1. Why are these issues so important to the practitioner?

2. How do the DBMS vendors view the selection/evaluation criteria currently used by many of its customers?

3. What does the DBMS vendor see as the most important issue in a DBMS selection?

4. How much does the DBMS selection process cost?
# DB-Engines DBMS Ranking

[http://db-engines.com/en/ranking]

299 systems in ranking, March 2016

<table>
<thead>
<tr>
<th>Rank</th>
<th>Mar 2016</th>
<th>Feb 2016</th>
<th>Mar 2015</th>
<th>DBMS</th>
<th>Database Model</th>
<th>Score</th>
<th>Score</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Oracle</td>
<td>Relational DBMS</td>
<td>1472.01</td>
<td>-4.13</td>
<td>+2.93</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>MySQL</td>
<td>Relational DBMS</td>
<td>1347.71</td>
<td>+26.59</td>
<td>+86.62</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>Microsoft SQL Server</td>
<td>Relational DBMS</td>
<td>1136.49</td>
<td>-13.73</td>
<td>-28.31</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>MongoDB</td>
<td>Document store</td>
<td>305.33</td>
<td>-0.27</td>
<td>+30.32</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>PostgreSQL</td>
<td>Relational DBMS</td>
<td>299.62</td>
<td>+10.97</td>
<td>+35.19</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>DB2</td>
<td>Relational DBMS</td>
<td>187.94</td>
<td>-6.55</td>
<td>-10.91</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>Microsoft Access</td>
<td>Relational DBMS</td>
<td>135.03</td>
<td>+1.95</td>
<td>-6.66</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>Cassandra</td>
<td>Wide column store</td>
<td>130.33</td>
<td>-1.43</td>
<td>+23.02</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>Redis</td>
<td>Key-value store</td>
<td>106.22</td>
<td>+4.14</td>
<td>+9.17</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>SQLite</td>
<td>Relational DBMS</td>
<td>105.77</td>
<td>-1.01</td>
<td>+4.06</td>
</tr>
</tbody>
</table>

The popularity of DBMS is based on the following parameters:

1- Number of mentions of the system on websites (google, bing)
2- Frequency of technical discussions about the system on websites (Stack Overflow and DBA Stack Exchange)
3- Number of job offers
4- Number of profiles in professional networks (LinkedIn)
Functional requirements:
In the database field, the functional requirements describe:
- the functionalities
- the functioning
They are specifying: the calculation, data manipulation and processing, identification, creation, insert, delete, update and others.

Non-functional requirements: describe how the system will do:
- the security,
- the performance (response time, refresh time, processing time, data import/export, load time),
- the capacity (bandwidth transactions per hour, memory storage),
- the availability,
- the data integrity,
- the scalability
- the energy, etc.

Non-functional requirements are difficult to test.
### Testing – Types and methods

[Golfarelli 11, Tort 11, Haftmann 07]

<table>
<thead>
<tr>
<th>Type</th>
<th>Actor</th>
<th>Objective</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural testing</td>
<td>Designer / developer</td>
<td>Testing all phases of the life cycle of the database design</td>
<td>- Conceptual - Logical</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Physical - Exploitation</td>
</tr>
<tr>
<td>Functional testing</td>
<td>End users</td>
<td>Testing a database as a final product</td>
<td>- Checking data integrity and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>consistency</td>
</tr>
<tr>
<td>Non-functional testing</td>
<td>Developper/ Administrator</td>
<td>Testing the interaction between the applications and their underlying</td>
<td>- Performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>databases</td>
<td></td>
</tr>
</tbody>
</table>

Two testing methods exist to answer the question above:

1. **Simulation**
   - Mathematical cost model
   - Formal methods

   **is based on parameters related to the principal dimensions of the database:**
   - Schema (length of an attribute)
   - Platform (disk page size)
   - Workload (selectivity factors)
   - DBMS - Operating system
   - Access methods and algorithms
   - Metrics

2. **Hardware experimentation**
## Hardware experimentation

### Testing environment

1. **Laboratory**
   - Time: 14/05/2015
   - Platform: Dell precision T1500
   - CPU: Intel Core i5 2.27GHz
   - Memory: 4GB of DDR3
   - Operating system: Ubuntu 14.04 LTS kernel 3.13
   - Deployment: Centralized
   - DBMS: Oracle 11gR2
   - Dataset: Star Schema Benchmark (SSB) datasets
   - Size: 100 GB
   - Workload: SSB queries
   - Access methods: Materialized views
   - Algorithm: Nondominated Sorting Genetic Algorithm NSGA II
   - Metrics: Response time CPU_Cost IO_Cost Energy
   - Hypothesis: Without cache

The same dimensions that are repeated in the testing (Platform, DBMS, operating system, workload, dataset, metrics ...)

2. **TPC (The Transaction Processing Performance Council)**
   - These tests are stored in websites of TPC such as TPC-H benchmarking
   - It uses the same dimensions
   - The tests are not really usable
Simulation : Our test repository

1- **Storage**: Test repository allowing persisting all environment of testing results;
2- **Usage**: Repository exploitation in order to deal with the problem of DBMS and platform selection.
Test repository – User interface

Manifest:
Excerpt of the test repository meta-model.
Basic concepts

Query-per-Hour Performance (QphH@size): This metric represents the number of queries executed for one hour relative to the size of the database.

Similarité: It is a comparison between two objects to determine the most important and useful relationships between them.

Distance Euclidienne:  
\[ D_E = \sqrt{\sum_{i=1}^{k} (X_i - Y_i)^2} \]

Normalisation: Resize all the attributes of data in the range 0-1  
\[ S_i = \frac{X_i - \text{MIN}(X_i)}{\text{MAX}(X_i) - \text{MIN}(X_i)} \]
Searching usage of our test repository

Our recommender system:

Our algorithm:

**Step 1** - analyzing of the company manifest to identify the presence of dimensions;
**Step 2** - getting a fragment of the data cube satisfying these dimensions (using Slice and Dice);
**Step 3** - normalizing all the dimension’s values using formula (2);
**Step 4** - computing the similarity between the company manifest and each instance of the data cube fragment. Note that an instance represents a test;
**Step 5** - selecting the best propositions based on the result of sorting. Indeed, tests are sorted in relation to similarity results for each DBMS.
**Step 6** - the company can choose its favorite DBMS based on its requirements such as price.
Example:
Process of our recommender system (1)

<table>
<thead>
<tr>
<th>Algorithm's steps</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><img src="example1.png" alt="Example" /></td>
<td>- Platform dimension</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DBMS dimension</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Dataset dimension</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Metrics dimension</td>
</tr>
<tr>
<td>Step 2</td>
<td><img src="example2.png" alt="Example" /></td>
<td><img src="results_table.png" alt="Results Table" /></td>
</tr>
</tbody>
</table>

**Results Table**

<table>
<thead>
<tr>
<th>DBMS</th>
<th>Test</th>
<th>Size</th>
<th>CPU</th>
<th>Memory</th>
<th>QphH</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSQL Server</td>
<td>Test1</td>
<td>1000</td>
<td>2.8</td>
<td>1536</td>
<td>588831</td>
</tr>
<tr>
<td></td>
<td>Test2</td>
<td>3000</td>
<td>2.5</td>
<td>3072</td>
<td>725686</td>
</tr>
<tr>
<td></td>
<td>Test3</td>
<td>3000</td>
<td>2.5</td>
<td>3072</td>
<td>700392</td>
</tr>
<tr>
<td></td>
<td>Test4</td>
<td>3000</td>
<td>2.8</td>
<td>3072</td>
<td>461837</td>
</tr>
<tr>
<td></td>
<td>Test5</td>
<td>10000</td>
<td>2.8</td>
<td>4096</td>
<td>652239</td>
</tr>
<tr>
<td>Oracle</td>
<td>Test6</td>
<td>1000</td>
<td>1.5</td>
<td>64</td>
<td>9853</td>
</tr>
<tr>
<td></td>
<td>Test7</td>
<td>3000</td>
<td>2.88</td>
<td>512</td>
<td>198907</td>
</tr>
<tr>
<td></td>
<td>Test8</td>
<td>3000</td>
<td>3</td>
<td>1024</td>
<td>205792</td>
</tr>
<tr>
<td></td>
<td>Test9</td>
<td>10000</td>
<td>1.5</td>
<td>288</td>
<td>108099</td>
</tr>
<tr>
<td></td>
<td>Test10</td>
<td>30000</td>
<td>1.6</td>
<td>1024</td>
<td>156980</td>
</tr>
<tr>
<td>DB2</td>
<td>Test11</td>
<td>100</td>
<td>3.6</td>
<td>4</td>
<td>1894</td>
</tr>
<tr>
<td></td>
<td>Test12</td>
<td>300</td>
<td>3</td>
<td>10165</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test13</td>
<td>1000</td>
<td>1.7</td>
<td>32</td>
<td>20221</td>
</tr>
<tr>
<td></td>
<td>Test14</td>
<td>1000</td>
<td>1.9</td>
<td>32</td>
<td>26156</td>
</tr>
<tr>
<td></td>
<td>Test15</td>
<td>3000</td>
<td>2.6</td>
<td>16</td>
<td>38672</td>
</tr>
</tbody>
</table>
### Example:

**Process of our recommender system (2)**

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Table in above with the following formulas:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( N_1 = \frac{\text{Size}_{\text{Test1}} - \min(\text{Size})}{\max(\text{Size}) - \min(\text{Size})} )</td>
</tr>
<tr>
<td></td>
<td>( N_{\text{Test1}} = \frac{\text{Distance}_{\text{Test1}} - \min(\text{Distance})}{\max(\text{Distance}) - \min(\text{Distance})} )</td>
</tr>
<tr>
<td></td>
<td>( \text{Distance}<em>{\text{Test1}} = \sqrt{\sum</em>{i=1}^{3} (N_{\text{Manifest}} - N_{\text{Test1}})^2} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DBMS</th>
<th>Test</th>
<th>Size</th>
<th>N1</th>
<th>CPU</th>
<th>N2</th>
<th>Memory</th>
<th>N3</th>
<th>QphH</th>
<th>Distance</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MySQL Server</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test1</td>
<td>1000</td>
<td>0,03</td>
<td>2,8</td>
<td>0,62</td>
<td>1536</td>
<td>0,37</td>
<td>588831</td>
<td>0,19</td>
<td>0,17</td>
<td></td>
</tr>
<tr>
<td>Test2</td>
<td>3000</td>
<td>0,10</td>
<td>2,5</td>
<td>0,48</td>
<td>3072</td>
<td>0,75</td>
<td>725686</td>
<td>0,59</td>
<td>0,52</td>
<td></td>
</tr>
<tr>
<td>Test3</td>
<td>3000</td>
<td>0,10</td>
<td>2,5</td>
<td>0,48</td>
<td>3072</td>
<td>0,75</td>
<td>700392</td>
<td>0,59</td>
<td>0,52</td>
<td></td>
</tr>
<tr>
<td>Test4</td>
<td>3000</td>
<td>0,10</td>
<td>2,8</td>
<td>0,62</td>
<td>3072</td>
<td>0,75</td>
<td>461837</td>
<td>0,57</td>
<td>0,50</td>
<td></td>
</tr>
<tr>
<td>Test5</td>
<td>10000</td>
<td>0,33</td>
<td>2,8</td>
<td>0,62</td>
<td>4096</td>
<td>1,00</td>
<td>652239</td>
<td>0,87</td>
<td>0,77</td>
<td></td>
</tr>
<tr>
<td>Test6</td>
<td>1000</td>
<td>0,03</td>
<td>1,5</td>
<td>0,00</td>
<td>64</td>
<td>0,01</td>
<td>9853</td>
<td>0,64</td>
<td>0,57</td>
<td></td>
</tr>
<tr>
<td>Test7</td>
<td>3000</td>
<td>0,10</td>
<td>2,0</td>
<td>0,66</td>
<td>512</td>
<td>0,12</td>
<td>198907</td>
<td>0,10</td>
<td>0,09</td>
<td></td>
</tr>
<tr>
<td>Test8</td>
<td>3000</td>
<td>0,10</td>
<td>3</td>
<td>0,71</td>
<td>1024</td>
<td>0,25</td>
<td>205792</td>
<td>0,14</td>
<td>0,12</td>
<td></td>
</tr>
<tr>
<td>Test9</td>
<td>10000</td>
<td>0,33</td>
<td>1,5</td>
<td>0,00</td>
<td>288</td>
<td>0,07</td>
<td>108099</td>
<td>0,70</td>
<td>0,62</td>
<td></td>
</tr>
<tr>
<td>Test10</td>
<td>3000</td>
<td>1,00</td>
<td>1,6</td>
<td>0,05</td>
<td>1024</td>
<td>0,25</td>
<td>156960</td>
<td>1,13</td>
<td>1,00</td>
<td></td>
</tr>
</tbody>
</table>

| **Oracle**  |       |      |     |      |     |        |     |      |          |    |
| Test11     | 100   | 0,00 | 3,6 | 1,00 | 4   | 0,00   | 1894 | 0,42 | 0,37     |
| Test12     | 300   | 0,01 | 3   | 0,71 | 32  | 0,01   | 10165| 0,20 | 0,18     |
| Test13     | 1000  | 0,03 | 1,7 | 0,10 | 32  | 0,01   | 20221| 0,55 | 0,49     |
| Test14     | 1000  | 0,03 | 1,9 | 0,19 | 32  | 0,01   | 26156| 0,46 | 0,41     |
| Test15     | 3000  | 0,10 | 2,6 | 0,52 | 16  | 0,00   | 38672| 0,22 | 0,19     |

| **DB2**     |       |      |     |      |     |        |     |      |          |    |
| Test11     | 800   | 0,02 | 2,8 | 0,62 | 768 | 0,19   | 0,00 | 0,00 |          |

<table>
<thead>
<tr>
<th>DBMS</th>
<th>QphH</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL Server</td>
<td>588831</td>
<td>0,17</td>
</tr>
<tr>
<td>Oracle</td>
<td>198907</td>
<td>0,09</td>
</tr>
<tr>
<td>DB2</td>
<td>10165</td>
<td>0,18</td>
</tr>
</tbody>
</table>
A Case study

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Workload</th>
<th>Platform</th>
<th>DBMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Case 2</td>
<td>✓</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

**Case 1**

Manifest 1

- Metric Result: Unknown of ResponseTime metric
- Organism: My Company
- Platform: CPU: 2.8 Ghz - Thread: 60 - Processor: 4 - Core: 24 - Memory: 768 Gbytes
- Dbms: Unknown
- Data Set: TPC-H datasets - Size: 800 GB
- Workload: TPC-H queries (Q3, Q7, Q19)
- Metric: ResponseTime

**Case 2**

Manifest 2

- Metric Result: Unknown of ResponseTime metric
- Organism: My Company
- Platform: Unknown
- Dbms: Unknown
- Data Set: TPC-H datasets - Size: 800 GB
- Workload: TPC-H queries (Q3, Q7, Q19)
- Metric: ResponseTime

**Results**

<table>
<thead>
<tr>
<th>Oracle</th>
<th>MSQL Server</th>
<th>DB2</th>
<th>Sybase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q3</td>
<td>1300.74</td>
<td>29.94</td>
<td>162.45</td>
</tr>
<tr>
<td>Q7</td>
<td>1327.01</td>
<td>36.69</td>
<td>1110.05</td>
</tr>
<tr>
<td>Q19</td>
<td>1124.39</td>
<td>10.07</td>
<td>1627.62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Oracle</th>
<th>MSQL Server</th>
<th>DB2</th>
<th>Sybase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q3</td>
<td>143.68</td>
<td>41.32</td>
<td>159.55</td>
</tr>
<tr>
<td>Q7</td>
<td>528.36</td>
<td>33.57</td>
<td>861.13</td>
</tr>
<tr>
<td>Q19</td>
<td>376.78</td>
<td>3.01</td>
<td>1081</td>
</tr>
</tbody>
</table>
Summary

- Warehouse covering different aspects of the testing environment (12 dimensions).

- Recommender system dedicating to recommend DBMS and platform for given requirements.

- Storage part (Dimensions détails)

- Usage part (Query similarity)