Laboratoire d'Informatique et d'Automatique pour les Systèmes

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

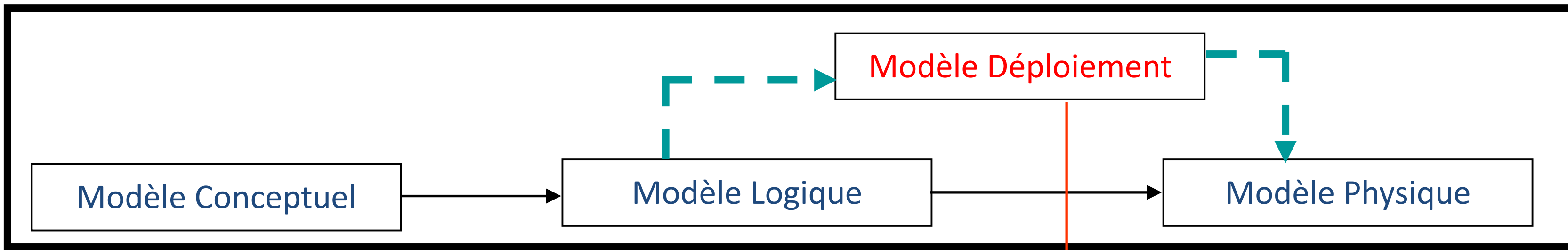
4- Our test repository

– Storage - User interface

5- Searching usage of our test repository

- Recommender system – Algorithm - Usage

Context and problematic



[Golfarelli 11, Jenkins 08, Bouarar 15]

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

DBMS Selection and evaluation

DBMS SELECTION AND EVALUATION: PERSPECTIVES AND PRACTICAL ISSUES

DBMS SELECTION AND EVALUATION: PERSPECTIVES AND PRACTICAL ISSUES

Wednesday, May 31, 1978
4:00 p.m. to 5:30 p.m.

F. Brett Berlin, Captain, USAF - Panel Chairman
Directorate of System Evaluation
Federal Computer Performance Evaluation and
Simulation Center (FEDSIM)
Washington, D.C.

James R. Deline
Vice President
DBD Systems, Inc.

J. Ron Phillips, Project Manager
Management Information Services
B.F. Goodrich Company, Chemical Division
Cleveland, Ohio

Wanda A. Reynolds, Supervisor
Data Base Administration
Texas State Comptroller
Austin, Texas

As management information systems become more and more "data centered", the software chosen to handle the data becomes more crucial to the system's overall successful operation. But how does a user procure a DBMS that is right for his or her needs? What are the questions that the user should be asking the vendor, and how should the vendor answers be evaluated? These are the central issues to be considered by this panel session. In this session, therefore, panel members will consider the DBMS user's problems in selecting DBMS software. Based upon personal experience and insight in this area, each panel member will seek to identify the central issues which must be considered in the DBMS selection process, including such problems as benchmarking and other pre-selection testing, general selection criteria, cost evaluation, transportability, security, conversion, and reliability. After presenting some of the basic issues, the panel will come to some conclusions as to how managers in industry and government can approach their own DBMS procurements.

The panel will open with a short presentation by the session chairman, in which he will attempt to set the stage for the panel discussion. In particular, the presentation will deal with the following issues:

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?

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

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

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

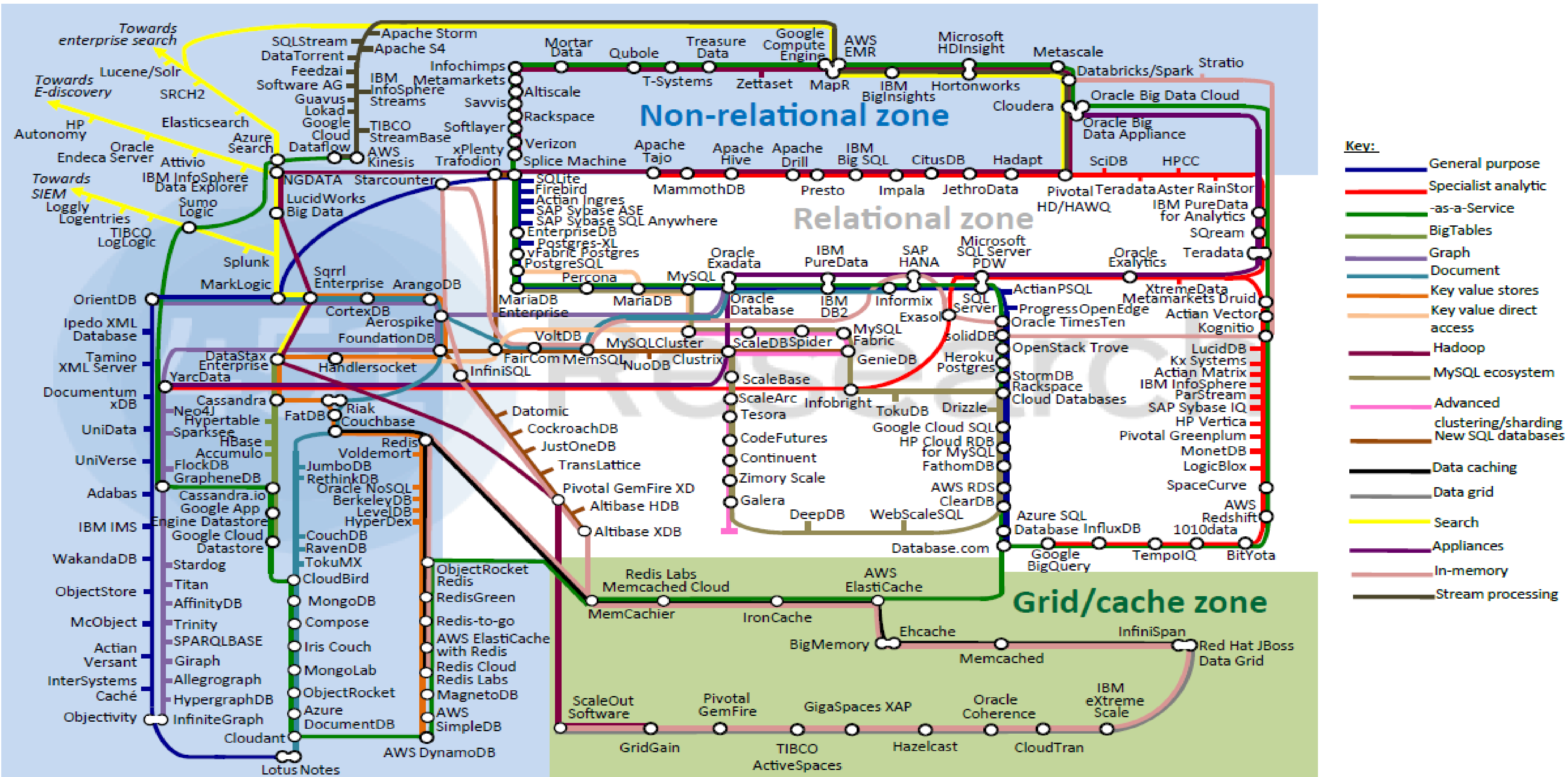
4- Number of profiles in professional networks (LinkedIn)

Landscape DBMS

451 Research

Data Platforms Landscape Map

OCTOBER 2014



Testing – Functional and non-functional requirements

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]

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

Two testing methods exist to answer the question above:

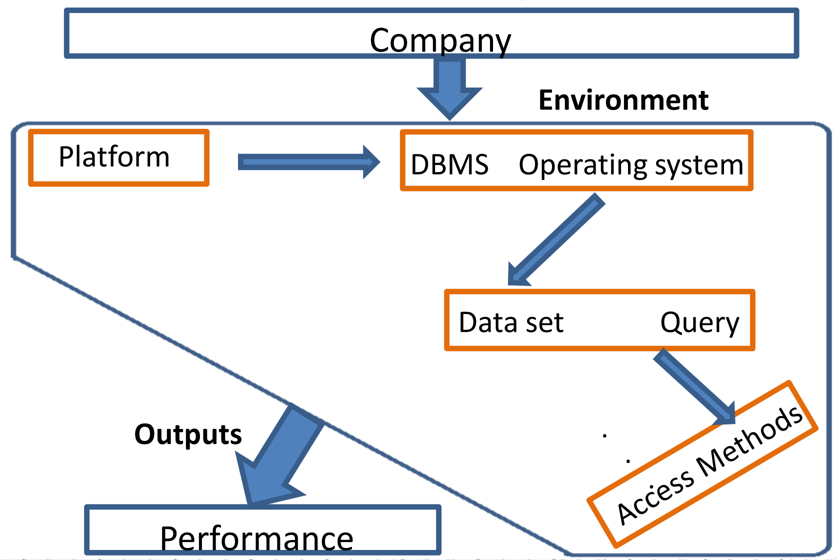
1- Simulation

- 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.

[Roukh 2015]

Laboratory	LIAS/ENSMA
Time	14/05/2015
Platform	Marque: 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

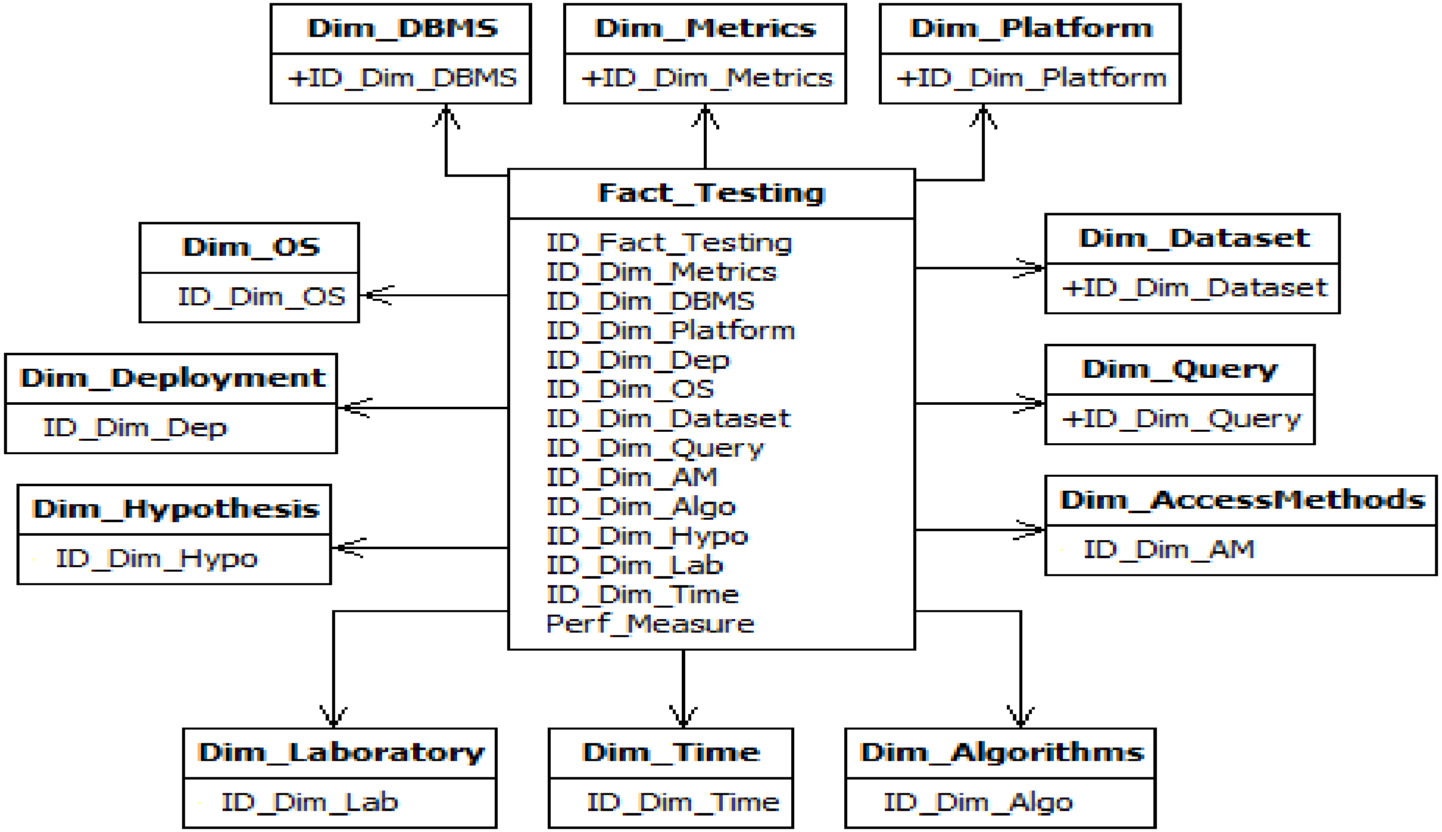
The same dimensions that are repeated in the testing (Platform, DBMS, operating system , workload, dataset, metrics …)

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

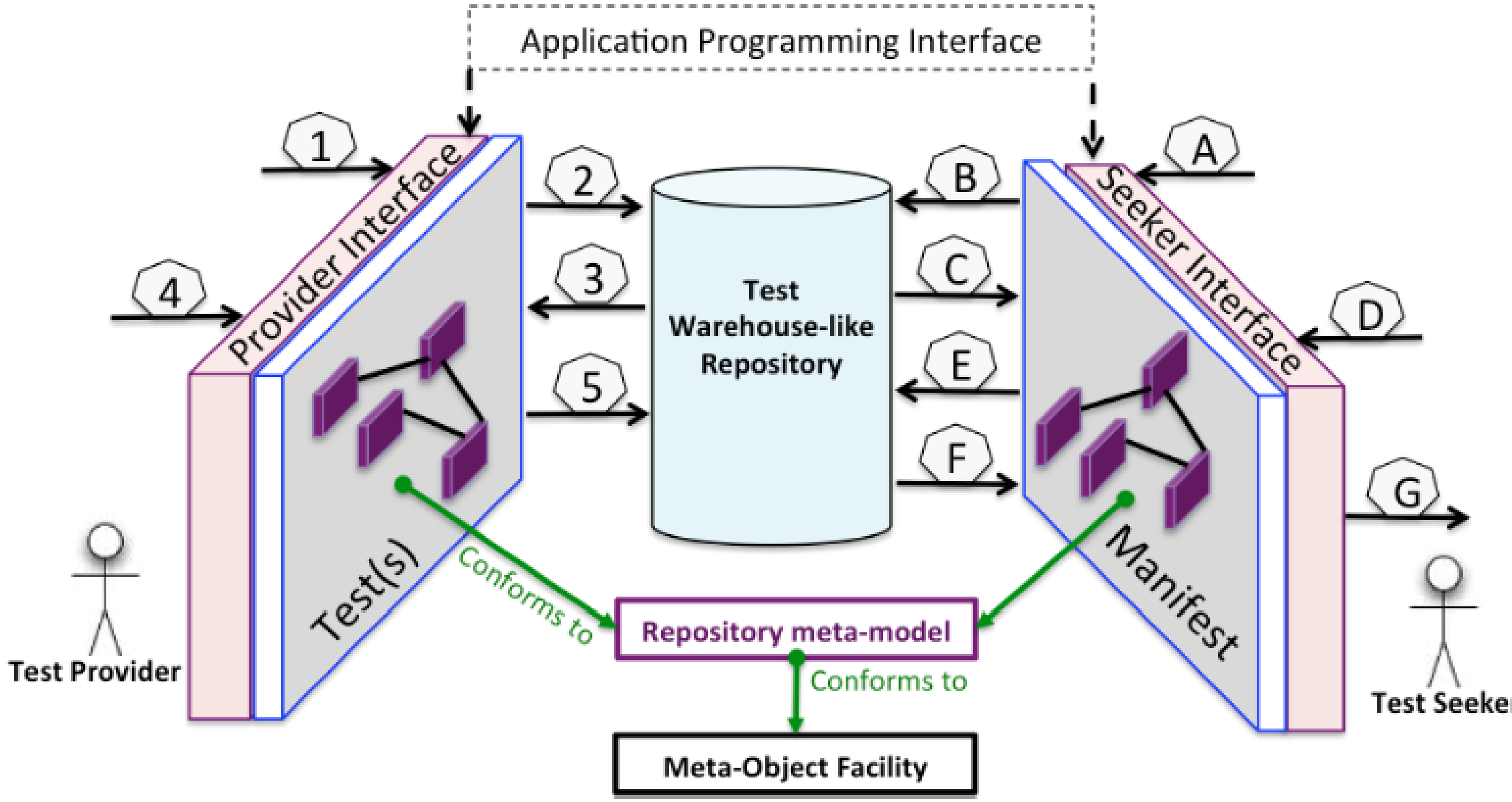
Simulation : Our test repository

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.

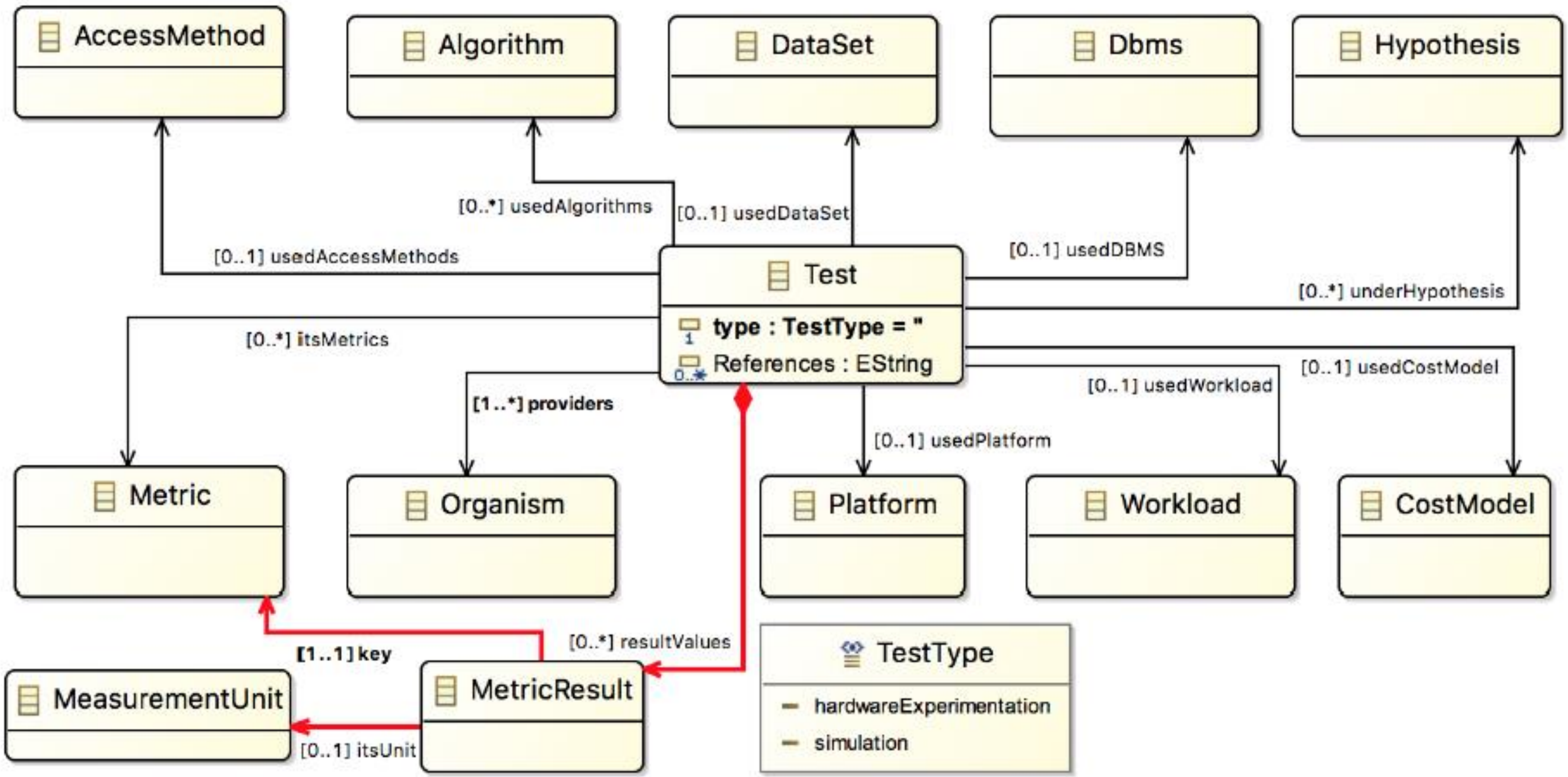


Test repository – User interface

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:

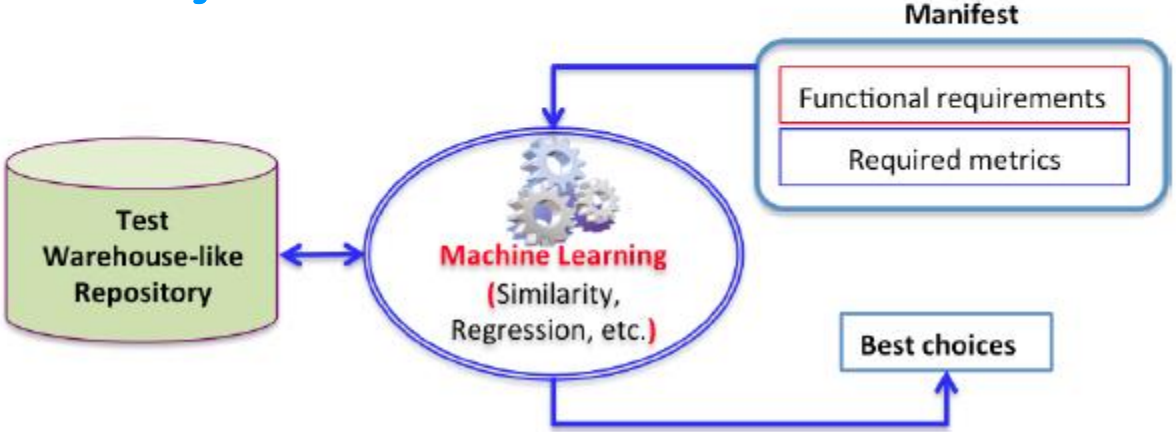
$$D_E = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

Normalisation: Resize all the attributes of data in the range 0-1

$$S_i = \frac{X_i - MIN(X_i)}{MAX(X_i) - MIN(X_i)}$$

Searching usage of our test repository

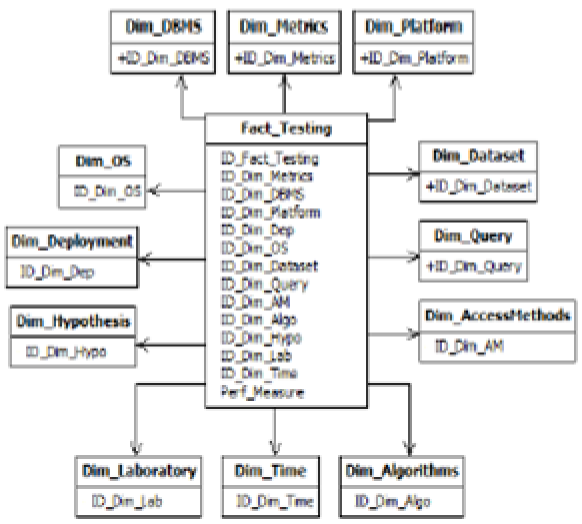
Our recommender system:



Our algorithm:

Step1- analyzing of the company manifest to identify the presence of dimensions;
Step2- getting a fragment of the data cube satisfying these dimensions (using Slice and Dice);
Stap3- normalizing all the dimension's values using formula (2);
Step4- computing the similarity between the company manifest and each instance of the data cube fragment. Note that an instance represents a test;
Step5- selecting the best propositions based on the result of sorting. Indeed, tests are sorted in relation to similarity results for each DBMS.
Step6- the company can choose it favorite DBMS based on its requirements such as price.

Example: Process of our recommender system (1)

<table>
<tr><th>Algorithm's steps</th><th>Example</th><th>Result</th></tr>
<tr><td>Step 1</td><td>Organism My Company
Platform CPU: 2.8 Ghz -Memory: 76.8 Gbytes
DBMS ?Unknown
Data Set TPC-H datasets -Size : 800 GB
Mteric QphH</td><td>- Platform dimension
- DBMS dimension
- Dataset dimension
- Metrics dimension</td></tr>
<tr><td>Step 2</td><td>
</td><td>
<table>
<tr><th>DBMS</th><th>Test</th><th>Size</th><th>CPU</th><th>Memory</th><th>QphH</th></tr>
<tr><td rowspan="5">MSQL Server</td><td>Test1</td><td>1000</td><td>2,8</td><td>1536</td><td>588831</td></tr>
<tr><td>Test2</td><td>3000</td><td>2,5</td><td>3072</td><td>725686</td></tr>
<tr><td>Test3</td><td>3000</td><td>2,5</td><td>3072</td><td>700392</td></tr>
<tr><td>Test4</td><td>3000</td><td>2,8</td><td>3072</td><td>461837</td></tr>
<tr><td>Test5</td><td>10000</td><td>2,8</td><td>4096</td><td>652239</td></tr>
<tr><td rowspan="5">Oracle</td><td>Test6</td><td>1000</td><td>1,5</td><td>64</td><td>9853</td></tr>
<tr><td>Test7</td><td>3000</td><td>2,88</td><td>512</td><td>198907</td></tr>
<tr><td>Test8</td><td>3000</td><td>3</td><td>1024</td><td>205792</td></tr>
<tr><td>Test9</td><td>10000</td><td>1,5</td><td>288</td><td>108099</td></tr>
<tr><td>Test10</td><td>30000</td><td>1,6</td><td>1024</td><td>156960</td></tr>
<tr><td rowspan="5">DB2</td><td>Test11</td><td>100</td><td>3,6</td><td>4</td><td>1894</td></tr>
<tr><td>Test12</td><td>300</td><td>3</td><td>32</td><td>10165</td></tr>
<tr><td>Test13</td><td>1000</td><td>1,7</td><td>32</td><td>20221</td></tr>
<tr><td>Test14</td><td>1000</td><td>1,9</td><td>32</td><td>26156</td></tr>
<tr><td>Test15</td><td>3000</td><td>2,6</td><td>16</td><td>38672</td></tr>
</table>
</td></tr>
</table>

Example: Process of our recommender system (2)

Step 3

Table in above with the following formulas:

$$N_1 = \frac{Size_{Test1} - Min(Size)}{Max(Size) - Min(Size)}$$

$$N_{Test_1} = \frac{Distance_{Test_1} - Min(Distance)}{Max(Distance) - Min(Distance)}$$

$$Distance_{Test_1} = \sqrt{\sum_{i=1}^{3}(Ni_{Manifest} - Ni_{Test_1})^2}$$

DBMS	Test	Size	N1	CPU	N2	Memory	N3	QphH	Distance	N
MSQL Server	Test1	1000	0,03	2,8	0,62	1536	0,37	588831	0,19	0,17
	Test2	3000	0,10	2,5	0,48	3072	0,75	725686	0,59	0,52
	Test3	3000	0,10	2,5	0,48	3072	0,75	700392	0,59	0,52
	Test4	3000	0,10	2,8	0,62	3072	0,75	461837	0,57	0,50
	Test5	10000	0,33	2,8	0,62	4096	1,00	652239	0,87	0,77
Oracle	Test6	1000	0,03	1,5	0,00	64	0,01	9853	0,64	0,57
	Test7	3000	0,10	2,9	0,66	512	0,12	198907	0,10	0,09
	Test8	3000	0,10	3	0,71	1024	0,25	205792	0,14	0,12
	Test9	10000	0,33	1,5	0,00	288	0,07	108099	0,70	0,62
	Test10	30000	1,00	1,6	0,05	1024	0,25	156960	1,13	1,00
DB2	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
	MANIFEST	800	0,02	2,8	0,62	768	0,19		0,00	0,00

Step 4

DBMS	QphH	N
MSQL Server	588831	0,17
Oracle	198907	0,09
DB2	10165	0,18

A Case study

	Dataset	Workload	Platform	DBMS
Case 1	✓	✓	✓	?
Case 2	✓	✓	?	?

Case1

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

	Oracle	MSQL Server	DB2	Sybase
Q3	1300.74	29.94	162.45	177.4
Q7	1327.01	36.69	1110.05	167.19
Q19	1124.39	10.07	1627.62	98.39

	Oracle	MSQL Server	DB2	Sybase
CPU	1.3	2.8	1.9	2.8
Proc	64	4	8	2
Threads	64	120	32	4
Cores	64	60	16	4
Memory	256	1536	32	16
Q3	143.68	41.32	159.55	4357.91
Q7	528.36	33.57	861.13	2792
Q19	376.78	3.01	1081	929.72

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)