Institut de Recherche en Informatique de Toulouse

CONSULTING.TECHNOLOGY.OUTSOURCING

Benchmark for OLAP on NoSQL Technologies

Max Chevalier
Mohammed EL Malki
Arlind Akopliku
Olivier Teste
Ronan Tournier

RCIS MAI 2015

Plan

1 ***Context - Objective***

2 ***Related work***

3 ***Contributions***

4 ***Experimentations***

STAR SCHEMA BENCHMARK

- Benchmarking data warehouses is a means to evluate :
  - the performance of systems
  - impacts of different technical choices.
- SSB is a reference benchmark : to support data warehousing application
- SSB is Developed on relational model
- Relational model shows their limitations to manage big data

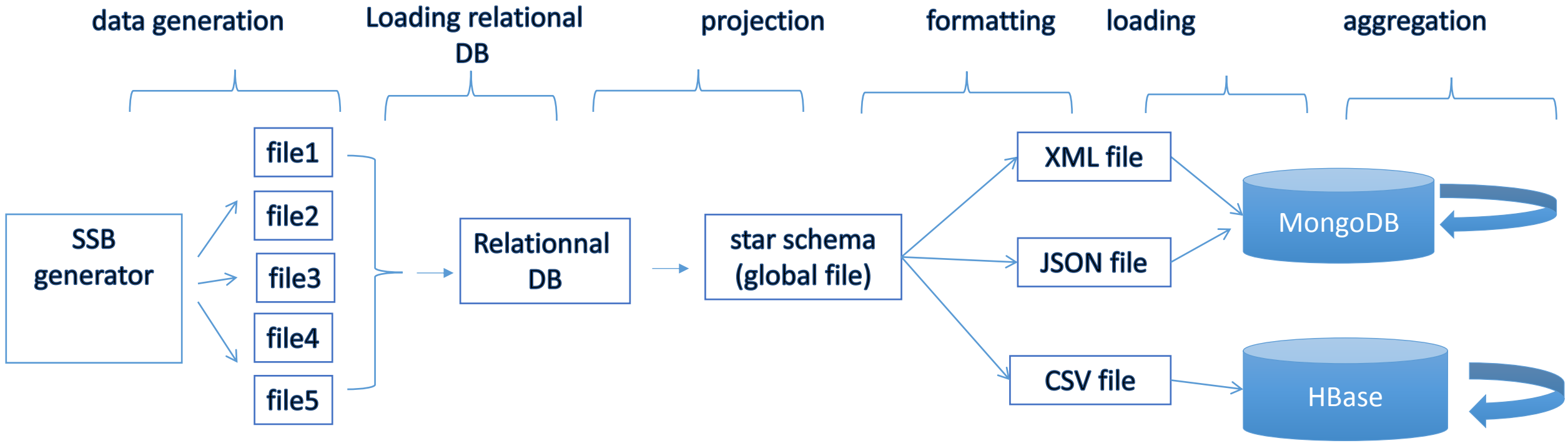
NoSQL

4 models

- key – value: Redis
- Document: MongoDB , CouchDB
- Column: Hbase, Cassandra
- Graph: Neo4j

NoSQL databases are known to be non-relational, horizontally scalable and distributed.

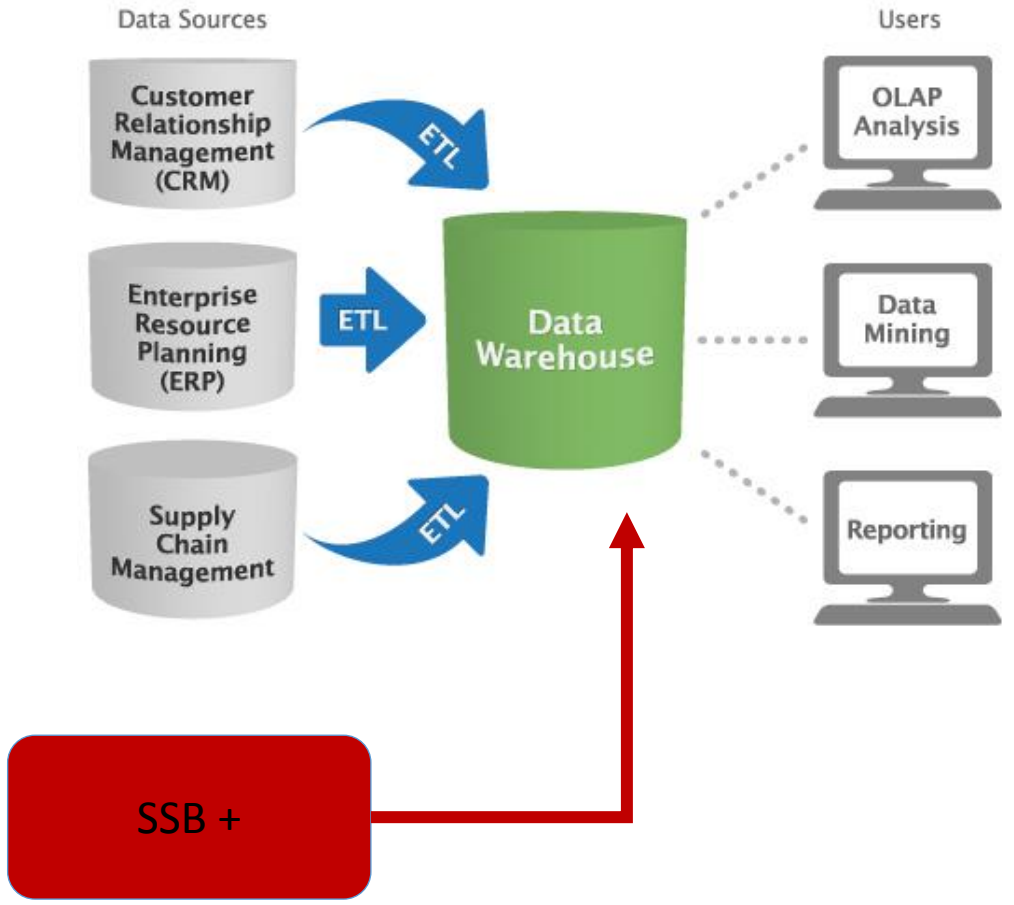
to use a the ssb benchmark in NoSQl models



Objective

We propose an extension to the Star Schema Benchmark (SSB) that supports distributed and NoSQL systems: columns-oriented and documents-oriented.

We focus only on tow models:
Column-oriented:vertitcal partionning
Document-oriented: Horizontal partitionning



To evaluate NoSQL systems

Related works

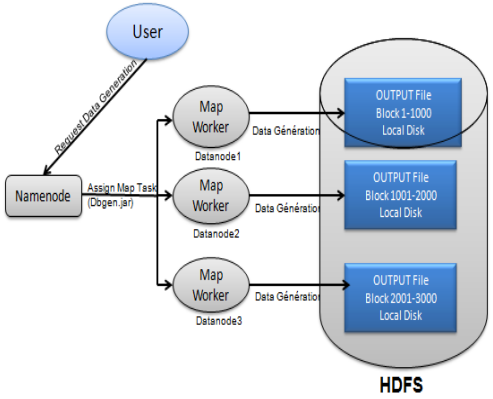
Categories / Criteria		NoSQL Model	HDFS	Star schema (conceptual model))	logical model	Generated Format according to model
DSS benchmark	**TPC-SSB**			x	x	csv
Big data benchmark	**Bigbench**	x	x			
Nosql benchmark	**CNSSB**	Columns	x	x	x	csv
	SSB+	Documents & Columns	x	x	x	csv, json, xml

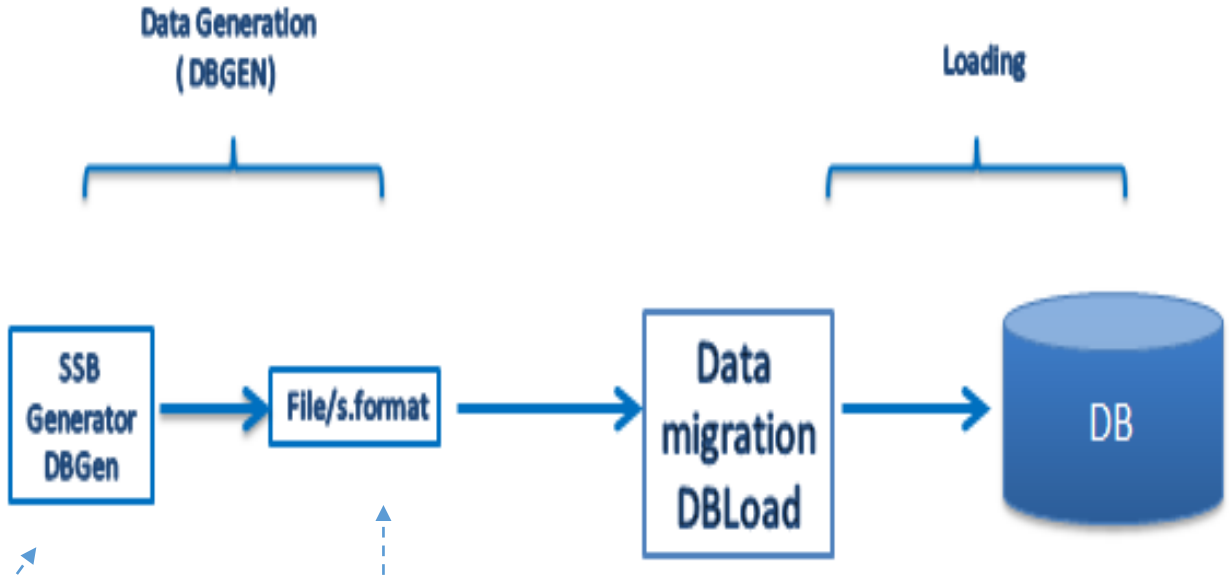
SSB+: Data Generation

We have parallelize Data Generation using hadoop:
- MapReduce: to distribute processing generating data on Datanodes).

All outputs are stored in HDFS

It's necessary to generate data in storage format used for the model to optimize loading stage.

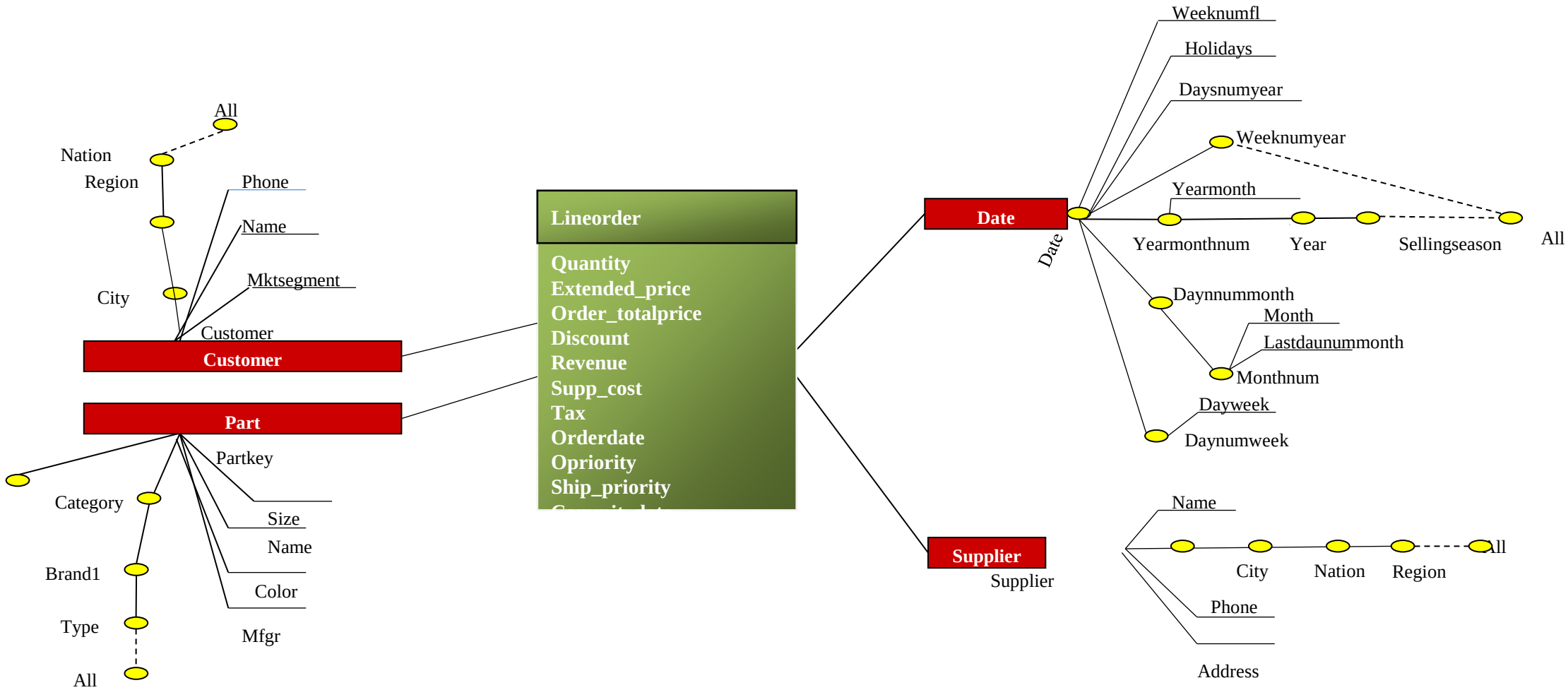




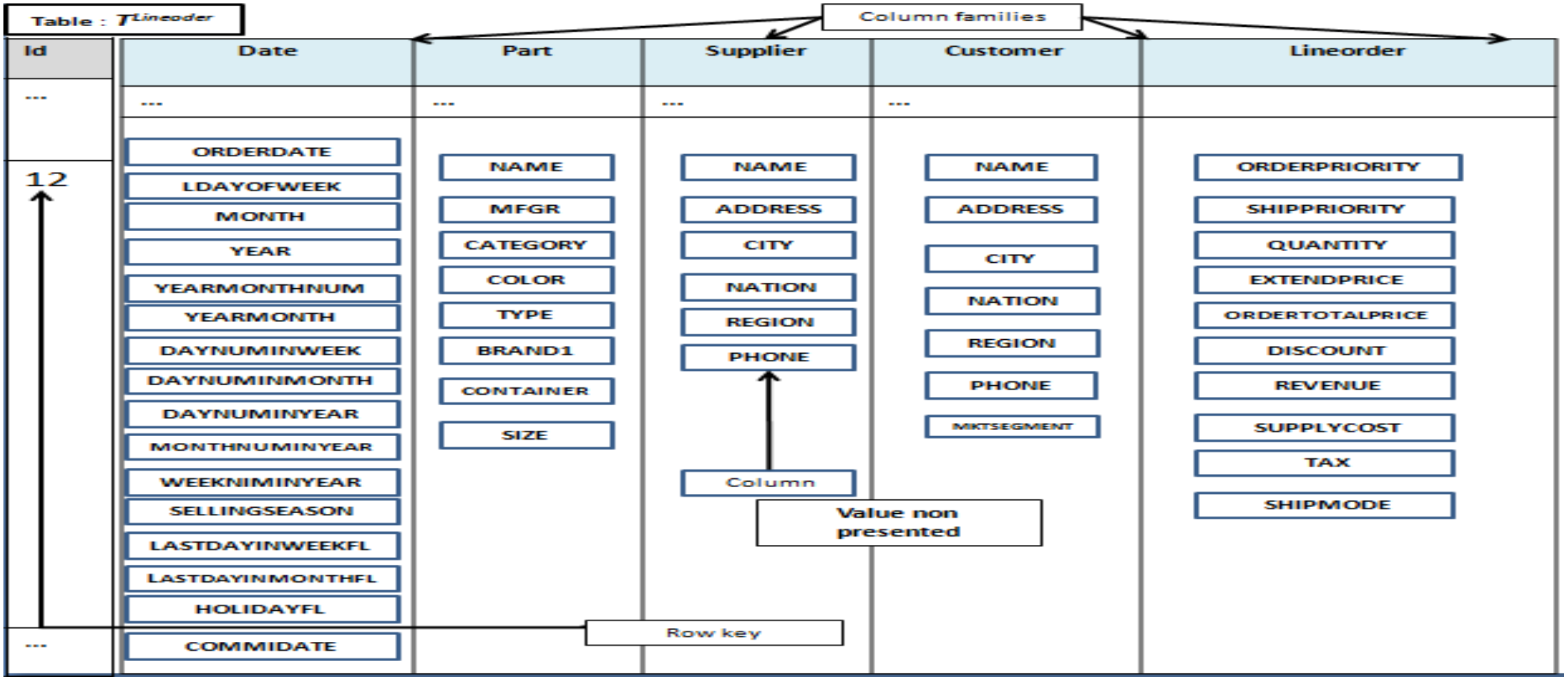
Different format:
CSV, JSON, XML

Different scripts according to Nosql model and logical schema

Muldimentionnal schema SSB



SSB+: supported NoSQL model



SSB+: supported NoSQL model

```
{
( Id: 220914
  date: {
                D_orderdate:
                D_date:
                D_year:
                D_yearmonthnum
                D_yearmonth:
                D_daynuminweek:
                D_daynuminmonth:

                 }
  part: {
                P_name:
                P_mfgr:
                P_category:
                }
  supplier: {
         terme : irak
         Catégorie : moyen-orient
         }
  Customer: {
                c_name:
                c_address:
                c_city:
                c_nation:
                C_region:
                C_phone:
                c_mktsegment:
         }
Lineorder: {
                Orderpopriority:
                Shippriority:
                Quantity:
                Extendedprice:
                Tax:
                Revenue:
                Discount:
                Supplycost:
                Ordertotalprice:
         }
}
```

V_{att} not represented

Legend

{}	Collection
()	Document
Att: {}	nested document
Att:	attribute
V_{att}	valeur de l'attribute att

Experimental Environment

Motivation

Validate feasibility

Comparing with SSB

Environment

Cluster : 3 Nodes

(commodity hardware: i5 quad-core, 8GB RAM, 4TB disk)

NoSQL Software

HBase, MongoDB

Experiment 1. Scale factor:

SSB does not generate the expected data size .
It generates between 0.56 and 0.58 times the amount of the expected data size .

SSB+ improves scaling (ratio 0.97)

Configuration	sf=1	sf=10	sf=100	sf=1000
SSB, normalized	987M	5.6G	59G	589G
SSB+, normalized	978M	9,7G	97G	976G
SSB+, denormalized	3.9M	39G	390G	3900G

Experiment 2 : Execution times

Configuration	sf=1	sf=10	sf=100	sf=1000
SSB, normalized	11.42s	90.8s	1383s	16715s
SSB+, normalized	20.82s	208.2s	2072s	20820s
SSB+, denormalized	21.05s	217s	2135s	2864s

Time needed to generate data at different scale factors for different configurations.
the scale factor of SSB+ generates considerably more data.

Conclusion

Bilan:

Transform existing benchmark SSB into improved version adapted into two NoSQL models (columns and documents).

Futur Work:

Placing the benchemark online.

To propose Generic logical model that will determined by user.

To invetigate the new NoSQL logical models

Thank you,
Questions?