Benchmarking Data Warehouse Systems in the Cloud

Rim Moussa
LaTICE Lab., University of Tunis
rim.moussa@esti.rnu.tn

***Abstract*—The most common benchmarks for cloud computing are the Terasort benchmark and the YCSB benchmark. Although these benchmarks are quite useful, they were not designed for data warehouse systems and related OLAP technologies. In this paper, first, we present cloud computing and data warehouse systems. Then, we argue that TPC-H benchmark -the most prominent benchmark for decision support system, mismatches cloud rationale (scalability, elasticity, pay-per-use, fault-tolerance features) and Customer Relationship Management rationale (end-user satisfaction, Quality of Service features). Finally, we present new requirements for implementing a benchmark for data warehouse systems in the cloud. The proposed requirements should allow a fair comparison of different cloud systems providers' offerings.**

Keywords*—*Data Warehouse, OLAP, Cloud, TPC-H, Benchmark.

I. Introduction

Business Intelligence aims at supportting better decision-making, through building quantitative processes for a business to arrive at optimal decisions and to perform business knowledge discovery. Business intelligence often uses data provided by Data Warehouse Systems, in order to provide historical, current and predictive views of business operations. Nevertheless, data warehousing is very expensive, since it requires experts, advanced tools as well as costly hardware. Some organizations with limited means related to each of human, software and hardware resources for data analytics, are throwing terabytes of data away. Thus, the arrival of *pay-as-you-go Cloud Computing* presents new opportunities for decision support systems.

The cloud computing market is booming, and many research groups as Forrester and Gartner, forecast a big invest in short-time on cloud technologies. Indeed, Forrester Research expects the global cloud computing market to reach $241 billion in 2020 [1], and Gartner group expects the cloud computing market will reach $US150.1 billion, with a compound annual rate of 26.5%, in 2013 [2]. Also, the Business Intelligence market continues growing and information analysts embrace well OLAP concepts and related technologies (Microsoft Analysis Services, Oracle Business Intelligence, Pentaho BI suite, SAP NetWeaver, ...). According to Gartner's latest enterprise software survey, worldwide business intelligence platform, analytic applications and performance management software revenue hit US$12.2 billion in 2011. This presents a 16.4% increase from 2010 revenue of US$10.5 billion, to take the position as the year's second-fastest growing sector in the overall worldwide enterprise software market. Gartner's view is that the market for BI platforms will remain one of the fastest growing software markets in most regions (refer to [3] for details). However, there are hurdles around dealing with Big Data. Along Ralph Kimball, *Big data is a paradigm shift in how we think about data assets, where do we collect them, how do we analyze them, and how do we monetize the insights from the analysis*. Therefore, a major reason for the growth of big data is financial and Decision Support Systems have to deal with the Big Data four V-dimensions, namely (*i*) *Volume* -challenge of management of huge volumes of data, (*ii*) *Velocity* -challenge of how fast data is analyzed, (*iii*) *Variety* -challenge of dealing with unstructured, semi-structured, relational data, and finally (*iv*) *Veracity* -challenge of semantics and variability meaning in language.

Cloud computing has gained much popularity recently, and many companies now offer a variety of public cloud computing services, based on traditional relational DBMS, extended RDBMS and NoSQL technologies. Traditional software technologies tend to get quite expensive to manage, maintain and enhance. Two architectures have emerged to address big data analytics, which are extended RDBMS and NoSQL technologies (Apache Hadoop/MapReduce framework). Architectural developments for extended RDBMS are Massively Parallel Processing (MPP) and columnar storage systems. NoSQL has emerged as an increasingly important part of Big Data trends, and several NoSQL solutions are emerging with highly variable feature sets. Cloud services differ in service models and pricing schemes, making it challenging for customers to choose the best suited cloud provider for their applications. Data Warehouse Systems place new and different demands on cloud technologies, and vis-versa. In this paper, we propose new requirements for fair benchmarking of data warehouse systems in the cloud.

The outline of this paper is the following: first, in section II, we discuss related work in order to highlight our contribution. Then, we present preliminaries related to both cloud computing and data warehouse systems. In Section III, we recall the most important characteristics of cloud computing, and that a benchmark for data warehouse systems should feature; and in section IV, we briefly overview data warehouse systems and TPC-H benchmark. The latter is the most prominent benchmark for decision support system. We argue that TPC-H current specification mismatches cloud rationale (scalability, elasticity, pay-per-use, fault-tolerance features) and Customer Relationship Management rationale (end-user satisfaction, Quality of Service features). In section V, we present new requirements for implementing a benchmark for data warehouse systems in the cloud. The proposed benchmark should allow a fair comparison of different cloud systems, as well as tuning of a cloud system for a given Cloud Service Provider (CSP) and selection of best optimizations and best cost-performance tradeoffs. Finally, we conclude the paper and present future work.

978-1-4799-0792-2/13/$31.00 ©2013 IEEE

II. Related Work

Published research focused on some interesting data warehouses inherent features. Indeed,

- *Forrester* released a *Cost Analysis Tool: Cloud versus internal file storage Excel Workbook*, as a tool for comparison of storage in-premises and in the cloud [4],
- Nguyen et al. [5] propose cost models for Views Materialization in the cloud. Proposed cost models fit into the pay-as-you-go paradigm of cloud computing. These cost models help achieve a multi-criteria optimization of the view materialization vs. CPU power consumption problem, under budget constraints.

There are few papers dealing with processing and evaluating by performance measurement OLAP workloads on cloud systems. Next, we overview a bench of research projects related to experiments in the cloud,

- Floratou et al. [6] conducted series of experiments comparing cost of deployment in the cloud of different DBMSs, in order to make cloud customers aware of the high cost of using freeware software in the cloud. For instance, they run Q21 of the Wisconsin Benchmark, and compared its response time using the open-source MySQL to the commercial MS SQL Server. For the SQL Server-based service, the user has to pay an hourly license cost, while he does not need to pay any license fee for MySQL usage. MS SQL server runs Q21 in 185sec, while MySQL runs the same query in 621sec. Obviously, the end-user bill will be affected by this 3.3X performance gap,
- In order to compare, SQL technologies to NoSQL technologies, Pavlo et al. [7] compared the performance of Apache Hadoop/Hive to MS SQL Server database system using TPC-H benchmark,
- In [8], we proposed OLAP scenarios in the cloud. The proposed scenarios aim at allowing best performances, best availability and tradeoff between space, bandwidth and computing overheads. Evaluation is conducted using Apache Hadoop/Pig Latin with TPC-H benchmark, for various data volumes, workloads, and cluster' sizes.

Many cloud computing benchmarks exist, but have different objectives than data warehouse systems. For instance, the *TeraSort* [9] benchmark measures the time to sort 1 TB (10 billion 100B records) of randomly generated data. It is used to benchmark NoSQL storage systems such Hadoop and MapReduce performances. The *Yahoo Cloud Serving Benchmark - YCSB* [10] measures the scalability and performance of cloud storage systems such HBase -the column-oriented database of Hadoop project, against a standard workload. The *CloudStone* Benchmark [11] is designed to support Web 2.0 type applications and measures the performance of social-computing applications on a cloud. For data analytics, *MalStone* [12] is specifically designed to measure the performance of cloud computing middleware that supports the type of data intensive computing common when building data mining models.

In [13], Binnig et al. present initial ideas of requirements towards a web-shop benchmark (i.e. OLTP workload) in the cloud. They introduce new metrics for analyzing the scalability, the cost and the fault tolerance of cloud services. Later, in [14] they list alternative architectures to effect cloud computing for web-shop database applications and reports on the results of a comprehensive evaluation of existing commercial cloud services. They used the database and workload of the TPC-W benchmark, with which they assessed Amazon, Google, and Microsoft's offerings.

The *CloudCMP* project [15] aims at comparing the performance and the cost of various cloud service providers. It models a cloud as a combination of four standard services, namely, (1) *Elastic Computer Cluster Service*: The cluster includes an elastic number of virtual instances for a workload processing; (2) *Persistent Storage Service*: The storage service stores application data. Different types of storage services may exist: *table* (SQL and NoSQL storage are considered), *blob* (binary files) and *queue messages* (as for Windows Azur); (3) *Intra-cloud Network Service*: The network inside a cloud that connects the virtual instances of an application (4) *WAN Service*: The wide-area delivery network of a cloud delivers an application's contents to the end hosts from multiple geographically distributed data centers of the cloud. The project scope is general, it does not address benchmarking data warehouses in the cloud specifities.

Most published research focused on benchmarking through exclusively performance measurements of high level languages and platforms of cloud systems, or investigation of a cost model for a particular topic in the cloud. In this paper, we show that TPC-H benchmark -the prominent benchmark for decision support system, mismatches both (i) *cloud rationale* (scalability, elasticity, pay-per-use, fault-tolerance features) and (ii) *Customer Relationship Management rationale* (end-user satisfaction, Quality of Service features). Indeed, its metrics are not sufficient for assessing the novel cloud services. Moreover, we propose new metrics which fit to the characteristics of cloud computing and to characteristics of OLAP workloads. The proposed requirements and metrics main challenges are to make CSPs' offerings comparable from capabilities, and services perspectives.

III. Cloud Computing

The National Institute of Standards and Technology (NIST) [16] defines cloud computing as *a pay-per-use model for enabling available, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*. Hereafter, we recall the five cloud characteristics, the three cloud service models, and we overview Cloud Service Providers (CSP) pricing models.

A. Cloud Characteristics

The cloud model is composed of three characteristics of virtualized systems, namely (1) *broad network access* -cloud computing is network based, and accessible from anywhere and from any standardized platform (i.e. desktop computers, mobile devices, ...); (2) *resource pooling* -the multi-tenancy

aspect of clouds requires multiple customers with disparate requirements to be served by a single hardware infrastructure, and therefore, virtualized resources (CPUs, memory, etc.) should be sized and resized with flexibility; (3) *rapid elasticity* -cloud computing gives the illusion of infinite computing resources available on demand. In particular, it is expected that the additional resources can be (a) provisioned, possibly automatically in mere minutes, when an application load increases (scale-up) and (b) released when load decreases (scale-down). In addition to the aforementioned characteristics, the cloud model is composed of two characteristics of on-demand computing services: (4) *on-demand self-service* -consumers of cloud computing services expect on-demand, nearly instant access to resources; (5) *measured service* (a.k.a. pay as you go) -cloud services must be priced on a short term basis (e.g., by hour), allowing users to release resources as soon as they are not needed, and metering should be done accordingly for different types of service (e.g., storage, processing, and bandwidth).

B. Cloud Service Models

Based on user demand, cloud services include the delivery of software, infrastructure, and storage over the Internet, either as separate components or as a complete platform. Three primary cloud service models exist. The first being *Infrastructure as a Service* (IaaS) -An IaaS provider delivers computer hardware (servers, network, storage) as a service. It may also include the delivery of operating systems and virtualization technology to manage the resources. Examples of IaaS CSPs are: Amazon Elastic Computing Cloud (EC2), GoGRID. The second being *Platform as a Service* (PaaS) -a PaaS provider delivers infrastructure and an integrated set of software which provides everything a developer needs to build an application. Examples of PaaS CSPs are: Google AppEngine, Microsoft Azure Platform. The third being *Software as a Service* (SaaS) -a SaaS CSP provides business applications as a service. Examples of SaaS providers for data analytics is: Google BigQuery, and for data base as a service is: Amazon Relational Database Service.

C. CSP Pricing Models

Even though, many services look similar from the outside, the services vary when it comes to system architectures, performance, scalability, and cost. Also, CSPs have different pricing models for storage, CPU, bandwidth and software.

1) Compute Cost: There are two types of customers' charging for CPU cost,

- Instance-based: the CSP charges the customer for the number of allocated instances and how long each instance is used. This regardless of whether the instances are fully utilized or under utilized. Examples of CSPs which fall in this CPU pricing model are Amazon AWS and Windows Azure.
- CPU cycles-based: the CSP charges the customer for the number of CPU cycles a customer's application consumes. Examples of CSPs which fall in this CPU pricing model are CloudSites and Google AppEngine.

2) Storage Cost: Each storage operation induces CPU cycles. There are two types of customers' charging for storage cost,

- CPU cycles-based: the CSP charges a customer based on the CPU cycles consumed to execute each storage operation. Therefore, a complex operation can incur a higher cost than a simple one. Examples of CSPs which fall in this CPU pricing model are: Amazon Simple DB, CloudSites and Google AppEngine.
- Number of operations: the CSP charges a customer based on the number of operations, regardless of each operation complexity. Examples of CSPs which fall in this CPU pricing model: Windows Azure Table.

3) Software Licenses' Costs: The CSP may provide some software at no cost. Notice that most operating systems are charged to customers with the cost of instance, while specific software as for instance Database Management Systems or MapReduce implementations are charged on a per hour usage.

4) Intra-network cost: Most providers offer intra-cloud network bandwidth consumption at no cost. Basically, no information is available about interconnectivity of nodes within a data center. Notice that, intra-network bandwidth is very important for distributed processing of OLAP workloads, for both SQL and NoSQL solutions.

5) WAN cost: Charges for using the wide-area delivery network are based on the amount of data delivered through the cloud boundaries to the end-users. Currently, most providers have similar prices for this service, where data upload is free of charge and data download is priced.

6) SaaS Services: SaaS offers for analytics are different, than IaaS and PaaS offers. Indeed, the cost of the service is included in the price model. For instance, BigQuery [17] pricing for storage resources depend on data volume, and the pricing of workload processing depends on the number of bytes retreived for each business question.

IV. Data Warehouse Systems

Business Intelligence aims at supportting better decision-making, through building quantitative processes for a business to arrive at optimal decisions and to perform business knowledge discovery. Business intelligence often uses data provided by Data Warehouse Systems. The concept of a data warehouse first appeared in articles published in the late 1988s by Bill Inmon. A data warehouse is defined as a *collection of subject-oriented, integrated, non-volatile, and time variant data to support management's decisions*. Data warehousing definition evolved to the process of collecting, cleansing, and integrating data from a variety of operational systems and making the resultant information available for the foundation of decision support and data analysis.

A. Typical DWS Architecture

Fig. 1 illustrates a typical architecture of a data warehouse system. The latter is composed of three components: (1) Source integration system, (2) Data warehouse storage system and (3) Data analysis system.

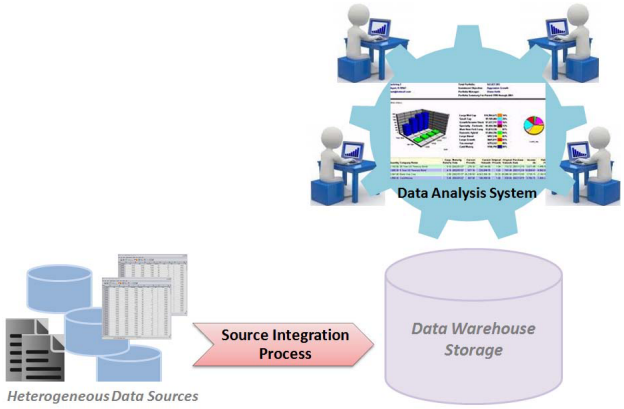

Fig. 1. Typical Data Warehouse System Architecture.

1) Source Integration System: The source integration process deals first with acquiring data from a set of relevant data sources (e.g. legacy systems, relational databases, spreadsheets, ...), then with integrating the schemas of the sources in order to obtain a global schema. For this purpose, it specifies the mapping between the global schema and the sources, and includes the specification of how to load and refresh data according to the global schema. Integration has to deal with the problem of cleaning and reconciling data coming from sources, and consequently resolving naming, structural and data conflicts.

2) Data Warehouse Storage System: Two main approaches can be distinguished for storing data within a data warehouse, namely (*i*) *MOLAP*, where data is stored directly into multidimensional data cubes. A multidimensional storage engine is used for building and storing data cubes and (*ii*) *ROLAP*, the data warehouse is physically stored using conventional Relational Database Management System and cubes are defined logically. There are also hybrid OLAP products (HOLAP), which allow both direct access to relational data for multidimensional processing, as well as having their own optimized multidimensional disk storage for aggregates and pre-calculated results.

3) Data Analysis System: The data analysis system embeds an OLAP server. The latter is a high-capacity, multi-user data manipulation engine specifically designed to operate on multidimensional data structures (or databases). Multidimensional querying implemented by OLAP clients is an exploratory process, performed by navigating along the dimensions and measures, and allowing, (*i*) increase/decrease the level of detail (respectively drill-down and roll-up OLAP operations), (*ii*) focus on specific subparts of the cube for on-screen viewing (slice and dice OLAP operations), and (*iii*) rotation of dimensions to new on-screen viewing (rotate OLAP operation).

B. Common Optimization Strategies

Data warehouse solutions and appliances achieve better performances with the following technologies,

1) Hardware Technologies: Some data warehouse appliances provide special hardware products as storage solutions on-premises. Solutions are In-memory database (DRAM) or Solid-State Drives (SSDs) to process Big Data and Parallel disk I/O. The latter allow to run a query in parallel on tens or hundreds of disk drives. Notice that these hardware-based solutions are expensive and obsolete over time.

2) Columnar Storage Technology: A column-oriented storage system stores each record's column value (or familty of columns) in different storage volumes or data blocks. This technology allows higher compression ratio and higher scan throughput than ordinary row-based storage systems.

3) Derived Data: In order to get a fast response, data warehouses use derived data such as OLAP indixes (e.g. bitmap, n-tree, ...), derived attributes and aggregate tables (a.k.a. materialized views).

C. TPC-H Benchmark

The most prominent benchmarks for evaluating decision support systems are the various benchmarks issued by the Transaction Processing Council (TPC). Next, we present TPC-H, which is the most used benchmark in the research community. The TPC-H benchmark exploits a classical product-order-supplier model. It consists of a suite of business oriented adhoc queries and concurrent data modifications. The workload is composed of twenty-two parameterized decision-support SQL queries with a high degree of complexity and two refresh functions: RF-1 *new sales* (new inserts) and RF-2 *old sales* (deletes). Scale factors used for the test database must be chosen from the set of fixed scale factors defined as follows: 1, 10, ... 100,000; resulting raw data volumes are respectively 1GB, 10GB,..., 100TB.

1) TPC-H Metrics: TPC-H benchmark reports two main metrics, (see details in Appendix A)

- *TPC-H Composite Query-per-Hour Performance Metric* (*QphH@Size*): The QphH@Size metric reflects multiple aspects of the capability of the system under test for query processing. These aspects include (*i*) the selected database size against which the queries are executed (i.e., *scale factor*), (*ii*) *power test* which is the query processing power when queries are submitted by a single stream, and (*iii*) the *throughput test*, which is the query throughput when queries are submitted by multiple concurrent users.
- *TPC-H Price-Performance Metric* ($/QphH): The $/QphH metric reflects the ratio of costs to performance. The calculation of the priced system consists of (*i*) the price of both hardware and software present in the system under test, (*ii*) the price of the communication interface supporting the required number of user interface devices, (*iii*) the price of on-line storage for the database and storage for all software, (*iv*) the price of additional products (software or hardware) required for customary operation, administration and maintenance for a period of 3 years, and finally (*v*) the price of all products required to create, execute, administer, and maintain the executable query texts or necessary to create and populate the test database.

2) Mismatching of TPC-H for Evaluation of DWS in the Cloud: The use of TPC-H for benchmarking Data Warehouse Systems in the cloud reveals the following problems,

First, considering the technical evolution of OLAP technologies in the last years, the TPC-H benchmark does not reflect modern implementations of data warehouse systems, and is not suitable for benchmarking of commercial business intelligence suites, i.e., integration services (ETL performances), OLAP engines (OLAP hypercubes building), mining structures (building data mining models), and reporting tools.

Second, the primary metric used by TPC-H -*QphH@Size*, is the number of queries processed per hour, that the system under test can handle for a fixed load. The system under test is then considered static, and this metric does not show the system scalability, i.e., system performance under variable loads and variable cluster size.

Third, the second metric used by TPC-H -*$/QphH*, is the ratio of costs to performance, such that the pricing is based on the total cost of ownership of the system under test on-premises. The ownership cost includes hardware pricing, software license costs, as well as administration and maintenance costs during 3 years. This is incompatible with the pay-as-you-go model of cloud computing, since the cloud customers are not directly exposed to the hardware, software maintenance, and administration costs of their deployment. For the cloud, different price-plans exist and the cost-performance ratio depends on data volume, workload, services, selected hardware, and the CSP pricing plan. Also, the demand for required hardware and software resources shall vary over time, and then is better formulated by the dynamic lot-size model.

Fourth, currently none of the TPC-benchmarks reports a *cost-effictiveness ratio metric*. Migration to the cloud, should help the compagny determine the best hardware configuration for managing efficiently its data and running efficiently its workload. Indeed, it does not make sense to afford an Amazon EC2 Extra Large Instance (15GB of memory and 8 EC2 compute units for $0.480 per Hour), when an Amazon EC2 Large Instance (7.5GB of memory and 4 EC2 compute units for $0.240 per Hour) satisfies the workload requirements.

Fifth, the current implementation of TPC-H assumes simultaneous run of both workload streams of queries and refresh functions. There are two types of refresh functions, namely *new data* and *old data*, where *Old data* requires deletes' processing, and most NoSQL systems (for instance Apache Hadoop), adopt the write-once strategy and are not designed to handle deletes. As a consequence, for instance for Apache Hadoop, deletes imply very costly join operations and new data files load into the system.

Sixth, the CAP theorem, also known as Brewer's theorem, states that it is impossible for a distributed computer system to simultaneously provide all three of the following guarantees, namely, (*i*) *Consistency* which guarantees that all nodes see the same data at the same time; (*ii*) *Availability* which guarantees that every request receives a response about whether it was successful or failed; and (*iii*) *Partition tolerance* which gurantees that the system continues to operate despite arbitrary message loss or failure of part of the system. Also, Brewer prooved that only two out of the three guarantees are fulfilled in a distributed system. The current TPC-H specification (ditto for TPC-DS), assumes that TPC-H deployment on a parallel machine (shared-disk or shared-memory system architecture), and not on a shared-nothing architecture. Benchmarking data warehouse systems in the cloud on a shared-nothing architecture should implement all different combinations of guarantees, namely CA, CP and AP when condidering refresh functions and high-availability.

Finally, the TPC-H benchmark lacks of adequate metrics for measuring the features of cloud systems like scalability, pay-per-use and fault-tolerance, and service level agreements. In the next section, we present requirements and new metrics for benchmarking data warehouse systems in the cloud.

V. Benchmarking Data Warehouse Systems in the Cloud

The data warehousing process is inherently complex and, as a result, is costly and time-consuming. The deployment of a data warehouse system in the cloud is very different than its deployment on-premises. Indeed, the relationship between the CSP and its customers is different than the relationship between a company and its BI department. Migration to the cloud should improve end-user satisfaction and induce greater business productivity. Thus, benchmarks designed for evaluation of data warehouse systems in the cloud should reflect end-user satisfaction, Quality of Service (QoS), as well as all inherent characteristics of cloud systems, namely scalability, pay-per-use and fault-tolerance. Next, we first present use cases of benchmarking data warehouse systems in the cloud, then we present new requirements and new metrics which aim at a fair comparison of different cloud systems providers of data warehouse systems.

A. Use Cases

Two main use cases are identified, of benchmarking data warehouse systems in the cloud. First, the *comparison of different cloud systems*, which aims to select the best CSP for final deployement of a data warehouse system. Second, the *tuning of a system*: which aims to select, for a given CSP, the capacity planning (operating system, number of instances, instance hardware configuration, ...), best optimizations, best cost-performance tradeoffs, best cost-effictiveness tradeoffs.

B. New Requirements and Metrics

Next, we detail new requirements and new metrics for benchmarking data warehouse systems in the cloud.

1) High Performance: Data warehousing is intended for decision support. The latter requires high performance for greater business productivity. Two main features of data warehousing in the cloud affect high performance, which are (*i*) data transfer to/from the CSP and (*ii*) workload processing.

First, the source integration system and the data analysis system manipulate huge data sets, and practically big data uploads on remote servers require a lot of bandwidth and perform better on local networks. So, unless creation of an expensive private link between the company and the provider, cloud computing is painful with low-speed connections and network congestion. If on-premises, companies are confronted to I/O-bound and CPU-bound applications, in the cloud they will confront to *network-bound applications*. Indeed, the bottleneck will be the network bandwidth available to perform huge data transfer to/from the CSP. Most CSPs provide data transfer

to their data centers at no cost (e.g., Data Transfer IN To Amazon EC2 From Internet costs $0.00 per GB). Nevertheless, data download is priced (e.g., Data Transfer OUT To Amazon EC2 From Internet $0.12 per GB per month for data volumes comprised between 1GB and 10TB, and it costs cheaper for higher data volumes, and it is free for lower data volumes).

Second, most OLAP engines implement intra-query parallelism to provide faster performance. Intra-query parallelism consists in breaking a complex single query into sub-queries, processing the workload over multiple processors, and finally performing post-processing for presenting the final query response. Three factors impact the final response time to the query -subject to intra-query parallelism. First, *Start-up costs*, which are related to starting up multiple processes for processing simultaneously sub-queries. The time to set-up these processes may dominate computation time if the degree of parallelism is high. Second, *Skew costs*, these costs show that in a distributed system the overall execution time is determined by the slowest of parallely executing tasks. Third, *Interference costs*, these costs relate to the time the processes are idle. Indeed, processes accessing shared resources (e.g. system bus, disks, or locks) compete with each other and spend time waiting on other processes. The conducted experiments (see details in Appendix B), measuring pig scripts response times across cluster size, show a concave curve -illustrated in Fig.2, with an optimum response time for a particular cluster size and where performance degrades from this optimum onward. For cloud computing, the slope, showing performance gain (from N to N') should be also expressed in a cost metric ($). Indeed, to obtain this improvement in response time, the system scales-out horizontally, and more instances are provisionned.

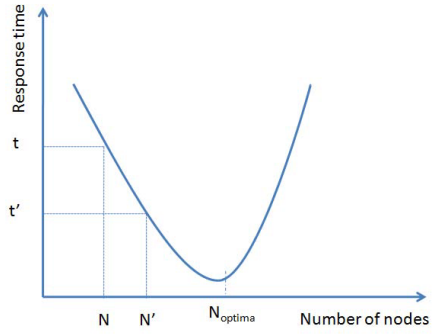

Fig. 2. Response Times of OLAP queries across Cluster Size.

2) Scalability: Scalability is the ability of a system, to increase total throughput under an increased load when hardware resources are added. Ideally, cloud services should scale linearly with a fixed cost per processed business question. Current TPC-H implementation measures the capacity of a system for a static workload. We propose that the benchmark for data warehousing should assess the system under test with an ever increasing load, and measures the throughput consequently. To quantify this requirement, we can vary the workload on a time scale basis, every 1hour for instance, and measure the number of business questions processed during the time interval. A scalable system, should maintain same number of business questions processed during a time interval, while a not scalable system records less business questions answered under a heavier load.

3) Elasticity: Elasticity adjusts the system capacity at runtime by adding and removing resources without service interruption in order to handle the workload variation. First, the metric should assess the system capacity to add/remove ressources without service interruption, and in case it does, it reports first the time required for a system to scale-down or to scale-up horizontally, i.e., *scaling latency* and second the scale-up cost, i.e. the cost of newly acquired resources (+$) or the scale-down gain, i.e. the cost of newly released resources (-$).

4) High Availability: Data distribution among multiple disks increases the distributed storage system failure likelyhood. Many approaches to build highly available distributed data storage systems have been proposed. They generally use either (*i*) replication or (*ii*) parity calculus. The latter approach uses systematic erasure-codes (e.g., Reed Solomon (RS) codes, Low-Density Parity-Check (LDPC) codes, Tornado code). With replication, data management is straightforward. However, the storage overhead with replication is always higher than it is with systematic erasure codes. When a certain level of availability is targeted the erasure codes are able to provide service with a lower storage overhead than replication techniques. For data warehousing, high availability through erasure codes saves storage costs, particularly for big data of type *write-once* (i.e., not subject to delete refreshes). Nevertheless, data recovery is more complicated than replication. Erasure codes were investigated and proved efficient for highly available distributed storage systems [18] and grid systems [19]. Fig. 3 illustrates the storage space requirements in different file high-availability schemes, namely replication and erasure codes. In our example, we show 4 blocks of a data file ($m = 4$) stored in such a way that any $(n - m) = 2$ missing blocks can be tolerated; values $n = 6$ and $m = 4$ are used as an example. With replication, k copies of the entire file are stored into separate places. The group of data blocks is 2-available through replication with a redundancy overhead of 200% versus the same group of data blocks 2-available through erasure-codes with a redundancy overhead of 50%.

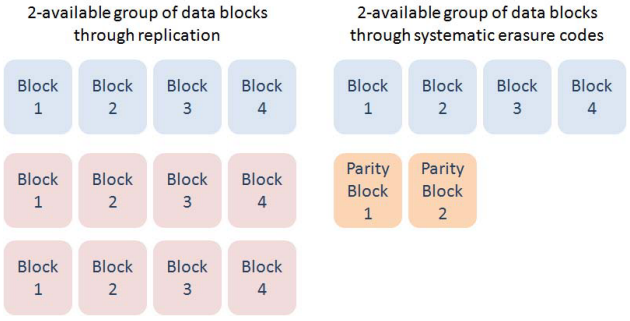

Fig. 3. Replication vs. Erasure Codes for a group of 4 data blocks.

Some CSPs implement replication for increasing the availability of stored data and preventing discontinuity of service. They also offer replicas management in data centers situated in different geographic locations. This allows disaster recovery from any whole data center outage. Nevertheless, most CSPs do not customize high availability services to their customers.

For data warehousing in the cloud, the end-user should be notified of the cost of rendering its data highly-available through different high availability strategies (i.e.,for both synchronous and asynchronous refreshes), and different levels of availability should be offered which enables customization of the recovery capacity following disasters. Consequently, the benchmark should embed metrics measuring the cost of different targeted levels of availabilities (1-available, ..., k-available, i.e. the number of failures the system can tolerate), as well as the recovery cost. We propose two metrics which denote the cost of maintaining of a k-available system $\$@k$, with k is the targeted level of availability, and a metric denoting the cost of recovery expressed in time and decreased system productivity caused by the hardware failure ($) from customer perspective. The latter should be charged to the CSP.

5) Cost-Effectiveness and Cost-Performance: The cloud-based solutions should help companies, which look to optimize costs without compromising on efficiency and quality of service. Therefore, there is an emerging need to understand, manage and proactively control costs across the cloud from two perspectives, namely *performance perspective* and *effectiveness perspective*. Indeed, instead of searching for the minimal execution time, the user may want to run his application more cost effectively, which ensures a maximal computation at minimal costs. The cost management plan should include, determination of the best hardware configuration versus performance and versus effectiveness; This assumes a systematic monitoring of resource utilization. For these purposes, we propose measuring the ratio of configuration cost ($) to performance and to resource utilization. Resource utilization is the ratio of used resources to allocated resources. Notice that used resources and allocated resources vary over time.

6) Service Level Agreements: A *Service Level Agreement* (SLA) is a contract between a service provider and its customers. SLAs capture the agreed upon guarantees between a service provider and its customer. They define the characteristics of the provided service including service level objectives, as maximum response times, minimum throughput rates and data consistency, and define penalties if these objectives are not met by the service provider. The SLAs categories for the data warehousing in the cloud are scalability, elasticity, performance (throughput and response time are both considered), high-availability and independency of the CSP. For the latter, the company should be able to easily migrate to another Cloud Service Provider (CSP), and gets its data back in a standard format. This will limit losses in case the CSP requires the purchase of new software, imposes exorbitant prices, or goes bankrupt.

C. OLAP vs. OLTP Benchmarking in the Cloud

We draw a lot of inspiration from the work in [13][14]. The latter investigated OLTP benchmarking in the cloud. In Table I, we propose a comprehensive comparison of OLAP and OLTP benchmarking in the cloud,

VI. Conclusion

The rationale of migration of data warehouse systems to the cloud, are basically thrice, (*i*) reduction of capital expenditure through measured service, with infrastructure, platform, services are provided on a pay-per-use basis (*ii*) rapid elasticity and (*iii*) rapid provisioning for better cost-performance tradeoff. In this paper, we argue that TPC-H benchmark -the most prominent OLAP benchmark does not match cloud features. We also propose new requirements and metrics to be fulfilled by a benchmark for data warehouses deployement in the cloud, depicting high-performance, high-availability, cost-effectiveness, cost-performance, scalability, elasticity, as well as SLAs. In future work, we will assess most known CSPs Amazon, Google and Microsoft offers for data warehousing using a revolved TPC-H benchmark for the cloud.

References

[1] Forrester, "Sizing the cloud," http://www.forrester.com, 2011.

[2] G. group, "?" http://www.gartner.com/it/page.jsp?id=920712, ?

[3] N. Laskowski, "Business intelligence software market continues to grow," http://www.gartner.com/it/page.jsp?id=1553215, 2011.

[4] Forrester, "File storage costs less in the cloud than in-house," http://www.forrester.com, 2011.

[5] T.-V.-A. Nguyen, S. Bimonte, L. d'Orazio, and J. Darmont, "Cost models for view materialization in the cloud," in *EDBT/ICDT Workshops*, 2012, pp. 47–54.

[6] A. Floratou, J. M. Patel, W. Lang, and A. Halverson, "When free is not really free: What does it cost to run a database workload in the cloud?" in *TPCTC*, 2011, pp. 163–179.

[7] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker, "A comparison of approaches to large-scale data analysis," in *SIGMOD Conference*, 2009, pp. 165–178.

[8] R. Moussa, "Massive data analytics in the cloud: Tpc-h experience on hadoop clusters," vol. 4, no. 3, 2012, pp. 113–133.

[9] J. Gray, "Sort benchmark home page," http://research.microsoft.com/barc/SortBenchmark/, 2008.

[10] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with ycsb," in *Proceedings of the 1st ACM symposium on Cloud computing*, ser. SoCC '10, 2010, pp. 143–154.

[11] W. Sobel, S. Subramanyam, A. Sucharitakul, J. Nguyen, H. Wong, A. Klepchukov, S. Patil, A. Fox, and D. Patterson, "Cloudstone: Multi-platform, multi-language benchmark and measurement tools for web 2.0," in *Proceedings of Cloud Computing and its Applications*, 2008.

[12] C. Bennett, R. L. Grossman, D. Locke, J. Seidman, and S. Vejcik, "Malstone: towards a benchmark for analytics on large data clouds," in *KDD'10*, 2010, pp. 145–152.

[13] C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, "How is the weather tomorrow?: towards a benchmark for the cloud," in *DBTest*, 2009.

[14] D. Kossmann, T. Kraska, and S. Loesing, "An evaluation of alternative architectures for transaction processing in the cloud," in *SIGMOD Conference'10*, 2010, pp. 579–590.

[15] L. Ang, Y. Xiaowei, K. Srikanth, and Z. Ming, "Cloudcmp: Shopping for a cloud made easy," in *USENIX HotCloud*, 2010.

[16] P. Mell and T. Grance, "The nist definition of cloud computing, national institute of standards and technology," csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2011.

[17] K. Sato, "An inside look at google bigquery," https://cloud.google.com/files/BigQueryTechnicalWP.pdf , 2013.

[18] W. Litwin, R. Moussa, and T. J. E. Schwarz, "Lh*$_{RS}$ - a highly-available scalable distributed data structure," *ACM Trans. Database Syst.*, vol. 30, no. 3, pp. 769–811, 2005.

[19] M. Pitkänen, R. Moussa, D. M. Swany, and T. Niemi, "Erasure codes for increasing the availability of grid data storage," in *AICT/ICIW*, 2006, pp. 185–197.

[20] R. Moussa, "Tpc-h benchmarking of pig latin on a hadoop cluster," in *ICCIT*, 2012, pp. 96–101.

TABLE I. OLAP VERSUS OLTP IN THE CLOUD.

	Data Warehouse System Deployement in the Cloud and OLAP Workload Run	**Operational System Deployement in the cloud and Web-shop/OLTP alike Workload Run**
Goals from Customer Perpective	• Fast upload and download of huge data sets, • Browse a multidimensional view of data in mere seconds	Good Responsivity to the user's interactions (Transaction branches) and validation of the interaction (transaction)
Horizontal scale-up Added Value	More Bytes processed per hour ($+BpH$)	More Web interactions processed per second ($+wips$)
Cost	• High storage cost for Data Warehouse System • Complex and costly workload (CPU, I/O, Network-bound applications)	• Storage Cost required by the Operational System • Simple workload
Metric of Interest	$/BpH (cost of Bytes processed by hour)	$/wips (cost of web interactions processed per second)
Recommended High-Availability Schema	Systematic Erasure Codes & Replication	Replication
Distributed Processing Features	Intra-parallelism within a business question	Inter-parallelism among interactions
Risks under Peak Loads	The compagny may do not take decisions in-time ($)	Unexpected performance problems and loss of responsivity to end-users. ($)

APPENDIX A
DETAILED TPC-H METRICS

- *Qph@SF Metric*

$$power\_test@SF = \frac{3600 \times SF}{\sqrt[24]{\prod_{i=1}^{i=22} QI(i,0) \times \prod_{j=1}^{j=2} RI(j,0)}}$$

$$throughput\_test@SF = \frac{S \times 22 \times 3600}{T \times SF}$$

with
- *3600* : is 1 hour duration in seconds
- *SF* : is the scale factor.
- *QI(i,0)* : is the timing interval, in seconds, of query Q_i within the single query stream of the power test,
- *RI(j,0)* : is the timing interval, in seconds, of refresh function RF_j within the single query stream of the power test,
- *S* : is the number of query streams, such that a stream is composed of different 22 queries.
- *T* : is the measurement interval,

$$QphH@SF = \sqrt{power\_test@SF \times throughput\_test@SF}$$

- *$/QphH Metric* $\$/QphH = \frac{PricedSystem}{QphH@Size}$

APPENDIX B
TPC-H BENCHMARKING OF PIG LATIN ON A HADOOP CLUSTER

Hereafter, we describe experiments carried out in order to benchmark Apache Hadoop/Pig Latin using TPC-H benchmark. In [20], we detail the translation of TPC-H workload from SQL into Pig Latin, as well as analysis of Pig Latin jobs for TPC-H workload. The hardware system configuration used for performance measurements are Borderline nodes located at Bordeaux site of GRID5000 platform. Each Borderline node has 32 GB of memory and its CPUs are Intel Xeon, 2.6 GHz, with 4 CPUs per node and 2 cores per CPU. All nodes run Lenny Debian Operating System. We conduct experiments of evaluating Pig for various cluster size and various data sizes. Thus, the following figures show Pig performance results for N=3, 5, 8 nodes (respectively 2, 4 and 7 Hadoop Task Trackers/ data nodes or workers and one Hadoop master). We generated TPC-H data files for SF=1, 10 resulting respectively in 1.1GB and 11GB. Fig. 4 presents TPC-H workload response times for 1.1GB of data. Notice that pig scripts execution times are not improved when the Hadoop cluster size doubles in size. Business questions which execution times improve when cluster size increases correspond to business questions which do not perform join operation, as Q1 and Q6. Fig. 5 presents TPC-H workload response times for 11GB of data. In general, the cluster size improves response times, as opposed to results corresponding to a volume of 1GB. Complex business questions as Q2, Q11, Q13 and so on are not affected by the cluster size. In comparison with results illustrated in Figure 4 for 1.1GB of data, Pig presents good performance face to a 10 fold data size. Indeed, elapsed times for responding to all queries whether is the cluster size (N=3, 5, 8) for a 11GB TPC-H warehouse are at maximum 5 times and in average twice elapsed times for a 1.1GB warehouse. We conclude that the cluster size is important, and it does not improve necessary the workload perormances.

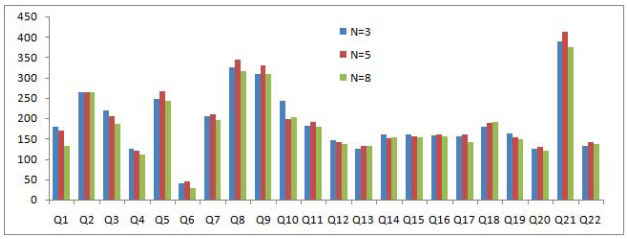

Fig. 4. Pig performances (sec) for 1.1GB of TPC-H data (SF=1).

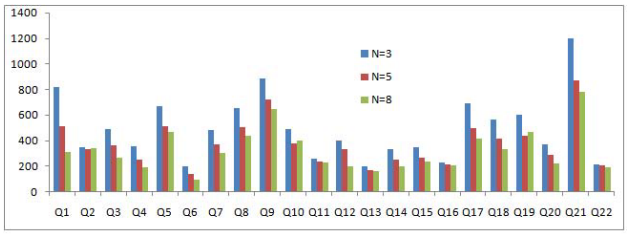

Fig. 5. Pig performances (sec) for 11GB of TPC-H data (SF=10).