Benchmarking criteria for a cloud data warehouse

Ahmed Alobaidi[1] and SeyedEbrahim Dashti[1]

[1]Islamic Azad University Shiraz

March 31, 2023

Abstract

The Terasort benchmark and the YCSB benchmark are the two most used cloud computing benchmarks. Despite the fact that these benchmarks are quite helpful, data warehouse systems and associated OLAP technologies were not the focus of their creation. We initially introduce cloud computing and data warehouse systems in this essay. Then, we contend that the TPC-H benchmark, which is the most well- known benchmark for decision support systems, conflicts with the justifications for cloud computing (scalability, elasticity, pay-per-use, fault-tolerance characteristics), and customer relationship management (end- user satisfaction, Quality of Service features). Finally, we propose updated specifications for a benchmark for cloud data warehouse systems. The suggested specifications ought to make it possible to fairly compare the products offered by various cloud system providers .

Introduction

By creating quantitative methods for a business to make optimal judgments and execute business knowledge discovery, business intelligence strives to enable better decision-making. Data from data warehouse systems is frequently used by business intelligence to provide historical, real-time, and forecasted perspectives of corporate activities. However, data warehousing is quite expensive because it calls for specialists, sophisticated gear, and cutting-edge software. Terabytes of data are being wasted by certain firms because they have insufficient human, software, and hardware resources for data analytics. With pay-as-you- go cloud computing now available, decision support systems have more potential than ever before.

Many research organizations, like Forrester and Gartner, predict a significant investment in cloud technology in the near future due to the rising cloud computing business. In fact, according to Forrester Research and Gartner Group, the global market for cloud computing is predicted to reach $241 billion in 2020 [2] and

$US150.1 billion in 2013, respectively [1]. Additionally, the market for business intelligence is still expanding, and information analysts are prepared with OLAP principles and related technology (Microsoft Analysis Services, Oracle Business Intelligence, Pentaho BI suite, SAP NetWeaver, . . .). Business intelligence platforms, analytical applications, and performance management software saw global revenue of US$12.2 billion in 2011, according to the most recent enterprise software report from Gartner.

In the broader global enterprise software industry, this is a 16.4% increase over 2010 revenue of US$10.5 billion, placing it as the year's second-fastest growing category. According to Gartner, the market for BI systems will continue to be one of the most rapidly expanding software sectors in most areas (refer to [3] for details). However, working with Big Data has its challenges. Ralph Kimball also described big data as a paradigm shift.

When considering data assets, we should consider where to get them, how to analyze them, and how to profit from the analysis's findings.

As a result, one of the main drivers of the growth of big data is financial, and decision support systems must address the four V-dimensions of big data: I volume, which is the challenge of managing large amounts of data; (ii) velocity, which is the challenge of how quickly data is analyzed; (iii) variety, which is the challenge of handling unstructured, semi-structured, and relational data; and (iv), veracity, which is the challenge of semantics and and variability meaning in language.

Recently, cloud computing has become quite popular, and many businesses now provide a range of public cloud computing services based on NoSQL, extended RDBMS, and standard relational DBMS technology. The cost to operate, maintain, and improve traditional software technologies can be fairly high. In order to handle large data analytics, two architectures—the extended RDBMS and the NoSQL technologies (Apache Hadoop/MapReduce framework)—have emerged. Columnar storage systems and massively parallel processing (MPP) are architectural advancements for extended RDBMS. storage devices. NoSQL has become a more significant component of Big Data trends, and a number of NoSQL solutions are surfacing with wildly varying feature sets. Customers find it difficult to select the right cloud provider for their applications since cloud providers differ in their service models and price structures. The expectations placed on cloud technologies by data warehouse systems are new and different, and vice versa. In this article, we suggest new standards for unbiased testing of cloud data warehousing systems.

The following is the paper's outline: In order to highlight our contribution, we first review related work in section II. Then, we give the preliminary results for the data warehouse and cloud computing systems. In Section III, we review the key aspects of cloud computing and the need for a benchmark for data warehouse systems; in Section IV, we provide a brief overview of data warehouse systems and the TPC-H benchmark. The latter serves as the decision support system's primary benchmark.

We contend that the existing TPC-H specification is inconsistent with the cloud logic (scalability, elasticity, pay-per-use, fault-tolerance features) (end-user satisfaction, Quality of Service features). In section V, we propose updated specifications for putting into practice a benchmark for cloud-based data warehousing systems. With the help of the suggested benchmark, it should be possible to fairly compare various cloud systems, tune a cloud system for a certain Cloud Service Provider (CSP), and choose the optimum optimizations and cost-performance tradeoffs. Finally, we wrap up the essay and discuss upcoming projects.

Related Work

Published studies concentrated on a few unique characteristics of data warehouses. In fact, Forrester published a Cost Analysis Tool comparing internal file storage to the cloud. Nguyen et al. [5] suggest cost models for Views Materialization in the cloud using the Excel Workbook as a tool for comparing storage on- premises and in the cloud [4]. The proposed cost models are compatible with cloud computing's pay-as- you-go methodology. Under financial restrictions, these cost models assist in achieving a multi-criteria optimization of the view materialization vs. CPU power consumption problem.

There aren't many articles discussing how to handle and evaluate OLAP workloads on cloud platforms using performance measurement. We then review a range of research initiatives pertaining to cloud experimentation,

To educate cloud users about the high cost of utilizing freeware software in the cloud, Floratou et al. [6] undertook a series of experiments comparing the cost of deployment in the cloud of various DBMSs. For instance, they ran Wisconsin Benchmark Question 21 and compared the open-source MySQL database to the for-profit MS SQL Server database's response time. The user must pay a licensing price on an hourly basis for the SQL Server-based service, but not for the use of MySQL. While MySQL executes Q21 in 621

seconds, MS SQL server does it in 185 seconds. Obviously, this 3.3X performance disparity will have an impact on the end-user cost.

Pavlo et al. [7] tested the performance of Apache Hadoop/Hive to MS SQL Server database system using TPC-H benchmark in order to compare SQL technologies to NoSQL technologies. We suggested OLAP cloud situations in [8]. The suggested scenarios seek to balance space, bandwidth, and computing overheads while allowing for best performance. The TPC-H benchmark is used to evaluate Apache Hadoop/Pig Latin across a range of data volumes, workloads, and cluster sizes.

There are several cloud computing benchmarks, however they don't have the same goals as data warehouse systems. The TeraSort [9] benchmark, for instance, calculates how long it takes to sort 1 TB (10 billion 100B records) of randomly generated data. It is used to evaluate the performance of NoSQL storage systems like Hadoop and MapReduce. The Yahoo Cloud Serving Benchmark (YCSB) [10] evaluates the speed and scalability of cloud storage solutions, including HBase, the column-oriented database of the Hadoop project.

The CloudStone Benchmark [11] evaluates social computing apps running on a cloud and is made to support Web 2.0 style applications. MalStone [12] is a performance evaluation tool for cloud computing middleware for data analytics that supports the kind of data-intensive computation that is typical when developing data mining models.

Binnig et al. outline the preliminary requirements for a web-store benchmark (i.e., OLTP workload) in [13]. They propose new measures for examining the cost, fault tolerance, and scalability of cloud services. Later, in [14], they provide a list of possible architectures to implement cloud computing for web-store database applications and present the findings of a thorough assessment of available commercial cloud services. They evaluated the products from Amazon, Google, and Microsoft using the TPC-W benchmark's database and workload.

The goal of the CloudCMP project [15] is to contrast the costs and performance of various cloud service providers. Four common services are combined to represent a cloud in its model, including Two services are available: (1) Elastic Computer Cluster Service, which uses an elastic number of virtual instances to handle workloads, and (2) Persistent Storage Service, which houses application data. Table (SQL and NoSQL storage are taken into account), blob (binary files), and queue messages (as for Windows Azur) are some examples of different types of storage services; (3) Intra-cloud Network Service: the internal cloud network that links application virtual instances (4) WAN Service: A cloud's wide-area delivery network transfers data from several geographically dispersed data centers to the end hosts where an application is running. The project's scope is broad; benchmarking data warehouses in the cloud specifically is not included.

The majority of published research concentrated on benchmarking through analysis of cost models for specific cloud topics or evaluation of high level language and platform performance measurements. We demonstrate in this work that the popular benchmark for decision support systems, TPC-H, mismatches both I the cloud reasoning (scalability, elasticity, pay-per-use, fault-tolerance characteristics) and (ii) the customer relationship management rationale (end-user satisfaction, Quality of Service features).). The novel cloud services cannot be evaluated using its measures, in fact. In addition, we provide additional measures that are appropriate for OLAP workloads and cloud computing features. Making the capabilities

and services of CSPs' offers comparable is the main difficulty of the proposed standards and measurements.

Cloud Computing:

Cloud computing, according to the National Institute of Standards and Technology (NIST) [21], is a pay-per-use model that enables easy, on-demand network access to a pool of configurable computing resources (such as networks, servers, storage, applications, and services) that can be quickly provisioned and released with little management work or service provider involvement. We will now review the three cloud service models, the five cloud characteristics, and the pricing strategies of cloud service providers (CSPs).

Cloud Characteristics

The cloud model is made up of three virtualized system characteristics: (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 that multiple customers with disparate requirements be served by a single hardware infrastructure, and therefore, virtualized resources (CPUs, memory,. . . in particular When an application's load increases (scales up), it is anticipated that the additional resources can be (a) provisioned, possibly automatically in a matter of minutes, and (b) released when load decreases (scale- down). The cloud model is made up of two characteristics of on-demand computing services in addition to the aforementioned ones: Customers of cloud computing services anticipate on-demand, practically immediate access to resources; (4) on-demand self- help; (5) measurable service (a.k.a. pay as you go) -Cloud services must be priced on a short-term basis (for example, by the hour), allowing customers to release resources as soon as they are no longer required. Different types of service should be metered in accordance with this (e.g., storage, processing, and bandwidth).

Cloud Service Models

Internet-based software, infrastructure, and storage, either as individual parts or as a whole platform. There are three main types of cloud service models. The first is Infrastructure as a Service (IaaS), which refers to the delivery of computer hardware (servers, networks, and storage) as a service by an IaaS provider. Providing operating systems and virtualization technology to manage the resources may also be part of it. IaaS CSPs include GoGRID and Amazon Elastic Computing Cloud (EC2). The second is Platform as a Service (PaaS), which gives infrastructure and a comprehensive collection of software, giving developers everything they need to create applications. Microsoft Azure Platform and Google AppEngine are two examples of PaaS CSPs. . The third is software as a service (SaaS), in which a cloud service provider (CSP) offers commercial applications as a service. Google BigQuery and Amazon Relational Database Service are two examples of SaaS providers for data analytics and databases, respectively.

CSP Pricing Models

Despite the fact that many services appear to be similar on the surface, they differ in terms of system topologies, performance, scalability, and cost. Additionally, CSPs have various pricing tiers for software, bandwidth, CPU, and storage.

Compute Cost: There are two ways that consumers can be charged for CPU costs. Instance-based billing involves charging customers according to the number of instances allotted and the length of time that each

instance is used. This is true whether or not the examples are fully or inadequately utilized. Examples of CSPs that use this CPU pricing model are Windows Azure and Amazon AWS.

CPU cycle-based: The CSP bills the client according to how many CPU cycles the client's application uses. CloudSites and Google AppEngine are two CSPs that use this CPU pricing approach.

The cost of storage: Every storage transaction requires CPU cycles. There are two different ways that consumers are charged for storage costs. CPU cycles-based billing involves charging a client according to the number of CPU cycles needed to complete each storage activity. As a result, a complex operation may cost more than a simple one. Examples of CSPs that use this CPU pricing model are Google AppEngine, Amazon Simple DB, and CloudSites.

Number of operations: Regardless of how complex each operation is, the CSP bases its charges on the total number of operations for a customer. CSPs that fit within this CPU price model include, for example: Microsoft Azure Table .

1. Costs of Software Licenses: The CSP might offer some software for free. Observe that while specific software, like database management systems or MapReduce implementations, is billed on an hourly basis, the majority of operating systems are priced per instance and charged to customers.
2. Intra-network costs: The majority of providers offer unlimited intra-cloud network bandwidth usage. In essence, there is no information available on node interconnectivity within a data center. Be aware

that intra- network bandwidth, for both SQL and NoSQL solutions, is crucial for the distributed processing of OLAP workloads.

3. WAN cost: Fees for accessing the wide-area delivery network are determined by how much data is sent to end users through the cloud's borders. Currently, the majority of providers charge about the same amounts for this service, with data upload being free and data download being paid.
4. SaaS Services: SaaS analytics offers differ from IaaS and PaaS analytics offers. In fact, the price model takes the cost of the service into account. For example, BigQuery [17] bases the cost of storage resources on data volume, and the cost of workload processing on the quantity of bytes returned for each business question.

Data Warehouse Systems

Through the development of quantitative processes that enable a firm to make the best decisions possible and to perform business knowledge discovery, business intelligence strives to promote better decision-making. Data warehouse systems frequently leverage data that is provided by business intelligence. The idea of a data warehouse first surfaced in publications written by Bill Inmon in the late 1980s. To support management's choices, a data warehouse is referred to as a collection of subject-oriented, integrated, non-volatile, and time-variant data. The process of gathering, purifying, and integrating data from a range of operational systems and making the resulting information accessible for the underpinnings of decision support and data analysis has come to be known as data warehousing.

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 begins with gathering data from a variety of pertinent data sources (such as legacy systems, relational databases, spreadsheets, etc.), after which the source schemas are integrated to create a single global schema. It provides the specification of how to load and refresh data in accordance with the global schema and specifies the mapping between the global schema and the sources for this purpose. In order to resolve nomenclature, structural, and data conflicts, integration must address the issue of cleansing and reconciling data from sources..
2. Data Warehouse Storage System: Two basic methods for storing data in a data warehouse may be distinguished, including I MOLAP, in which data is immediately saved into multidimensional data cubes. Data cubes are created and stored using a multidimensional storage engine, while (ii) ROLAP physically stores the data warehouse using a traditional relational database management system and defines the cubes logically. There are also hybrid OLAP solutions (HOLAP), which enable multidimensional processing with direct access to relational data as well as aggregates and pre-calculated results stored on their own multidimensional disk.
3. Data Analysis System: An OLAP server is integrated into the data analysis system. The latter is a multi- user, high-capacity data manipulation engine created primarily to work with multi-dimensional data structures (or databases). The exploratory nature of multidimensional querying used by OLAP clients enables I increase/decrease the level of detail (respectively drill-down and roll-up OLAP operations), (ii) concentrate on particular cube subparts for on-screen viewing (slice and dice OLAP operations), and (iii) rotate dimensions to new on-screen viewing (rotate OLAP operation).

Common Optimization Strategies

With the following technologies, data warehouse solutions and appliances function better:

1. Hardware Technologies: Some data warehouse applications offer specialized hardware items as on-site storage options. To process big data and parallel disk I/O, use in-memory databases (DRAM) or

solid-state drives (SSDs). The latter enable parallel query execution across dozens or hundreds of disk devices. Be aware that these hardware-based solutions becoming more and more expensive and out of date.

2. Columnar Storage Technology: In a column-oriented storage system, various storage volumes or data blocks are used to store the column value (or family of columns) of each record. Compared to standard row- based storage systems, this technology enables greater compression ratios and scan throughputs.
3. Data warehouses use derived data such as OLAP indixes (such as bitmap, n-tree,. . .), derived attributes, and aggregate tables in order to receive a quick response (a.k.a. materialized views).

TPC-H Benchmark

The many benchmarks released by the Transaction Processing Council are the most well-known standards for assessing decision support systems (TPC). We then introduce TPC- H, the most used benchmark in the research community. Utilizing the traditional product-order-supplier model is the TPC-H benchmark. It comprises of a number of concurrent data updates and business-oriented adhoc queries. Twenty-two highly sophisticated parameterized decision-support SQL queries make up the workload, together with two refresh functions called RF-1 new sales (new inserts) and RF-2 old sales (deletes). The set of fixed scale factors with the following definitions must be used to select the scale factors for the test database: 1, 10,. . . 100,000; the resulting raw data volumes are 1GB, 10GB,. . . 100TB, respectively.

TPC-H Metrics: The TPC-H benchmark provides two key metrics: (see details in Appendix A)

Query-per-Hour Performance Metric for TPC-H (QphH@Size): The QphH@Size statistic captures many facets of the query processing ability of the system under examination. These factors include I the chosen database size against which the queries are executed (also known as the scale factor), (ii) the power test, which measures the processing power of the queries when they are submitted by a single stream, and (iii) the throughput test, which measures the query throughput when it is submitted by multiple concurrent users.

Price-performance metric for TPC-H ($/QphH): The cost-to-performance ratio is represented by the $/QphH measure. The price of the priced system is determined by taking into account I the cost of the hardware and software present in the system being tested, (ii) the cost of the communication interface supporting the necessary number of user interface devices, (iii) the cost of online storage for the database and storage for all software, (iv) the cost of additional products (either software or hardware) needed for routine operation, administration, and maintenance for a period of three years, and (v) the final cost.

TPC-H mismatch for cloud-based DWS evaluation: Using TPC-H to assess cloud-based data warehouse systems highlights the following issues:

First, the TPC-H benchmark is not appropriate for evaluating commercial business intelligence suites, such as integration services (ETL performances), OLAP engines (building OLAP hypercubes), mining structures (building data mining models), and reporting tools, given the technical evolution of OLAP technologies in recent years.

Second, the number of queries processed per hour that the system under test can manage for a fixed load is the main statistic employed by TPC-H -QphH@Size. The system under test is then regarded as static, and this metric does not demonstrate the system's ability to scale, that is, how well the system performs under varying loads and cluster sizes.

Third, the ratio of costs to performance, or the second TPC-H metric, $/QphH, determines pricing based on the total cost of ownership of the system that is being tested on-site. The ownership cost includes the cost of the hardware, the cost of the software license, as well as the costs of administration and maintenance over a three-year period. The pay-as-you-go model of cloud computing is incompatible with this since cloud users are not directly responsible for the costs of administration, maintenance, and administration of their deployment of hardware and software. The cost-performance ratio for the cloud depends on the data volume, workload, services, chosen hardware, and the CSP pricing plan. There are various price plans for the cloud.

Additionally, the dynamic lot-size model provides a more accurate representation of how the demand for necessary hardware and software resources will change over time.

Fourth, no TPC-benchmark presently gives a cost-effectiveness ratio statistic. The company should be able to select the ideal hardware configuration for maintaining its data and handling its workload efficiently with the aid of the cloud migration. When an Amazon EC2 Large Instance (7.5GB of memory and 4 EC2 compute units for $0.240 per Hour) meets the workload requirements, it is inconvenient to pay for an Amazon EC2 Extra Large Instance (15GB of memory and 8 EC2 compute units for $0.480 per Hour)) [61].

Fifth, the current TPC-H implementation assumes that both workload streams for queries and refresh functions are conducted simultaneously. Old data requires the processing of deletes, and most NoSQL systems (such as Apache Hadoop) employ the write-once technique and are not built to handle deletes. There are two sorts of refresh functions: new data and old data. As a result, deletes, for Apache Hadoop, for example, entail exceedingly expensive join procedures and the loading of fresh data files into the system.

Sixth, according to the CAP theorem, also referred to as Brewer's theorem, a distributed computer system cannot simultaneously provide all three of the following guarantees: I Consistency, which ensures that all nodes see the same data at the same time; (ii) Availability, which ensures that every request receives a response indicating whether it was successful or unsuccessful; and (iii) Partition tolerance, which ensures that the system continues to function. Additionally, Brewer demonstrated that in a distributed system, only two of the three promises are met. The existing TPC-H specification (and the same goes for TPC-DS) presupposes parallel machine deployment of TPC-H rather than shared-nothing architecture. When considering refresh functions and high-availability, benchmarking data warehousing systems in the cloud on a shared-nothing architecture should implement all possible combinations of guarantees, namely CA, CP, and AP.

Last but not least, the TPC-H benchmark does not include sufficient measures for evaluating cloud system characteristics including scalability, pay-per-use, fault tolerance, and service level agreements. The requirements and fresh metrics for evaluating data warehousing systems in the cloud are presented in the next section.

Benchmarking Data Warehouse Systems In The Cloud

Due to the process' intrinsic complexity, data warehousing is both expensive and time-consuming. A data warehousing system's cloud deployment is considerably different from its on-premises deployment. In actuality, there are differences between a company's BI department and its clients and the CSP's connection with them. The move to the cloud should increase corporate productivity and increase end-user happiness. Therefore, end-user satisfaction, Quality of Service (QoS), as well as the inherent properties of cloud systems, such as scalability, pay-per-use, and fault-tolerance, should be reflected in benchmarks established for evaluating data warehousing systems in the cloud.

Then, we propose new standards and new measures that seek to create a fair comparison between various cloud systems providers of data warehouse systems. First, we present use cases for benchmarking data warehouse systems in the cloud.

Use Cases

There are two key use cases for comparing cloud-based data warehousing systems. The first step is a comparison of several cloud systems with the goal of choosing the best CSP for the data warehousing system's eventual deployment. The second step in system tuning is to choose the appropriate optimizations, cost-performance tradeoffs, and cost-efficiency tradeoffs for a given CSP's capacity planning (operating system, number of instances, instance hardware configuration, etc.).

New Requirements and Metrics

We next go over updated specifications and metrics for comparing cloud-based data warehouse systems.

High Performance: Data warehousing is used to assist with decision-making. In order to increase corporate productivity, the latter demands good performance. High performance is impacted by two key aspects of cloud data storage: I data transport to/from the CSP, and (ii) workload processing.

First, the source integration system and data analysis system deal with such large data sets, transferring significant data loads to remote servers typically uses a lot of bandwidth and is more efficient when done locally. Therefore, cloud computing is difficult due to slow connections and network congestion unless an expensive private link is established between the provider and the company. Companies will encounter network-bound apps in the cloud as opposed to I/O- and CPU-bound applications on-premises. The network bandwidth that is available to handle large data transfers to and from the CSP will, in fact, be the bottleneck.

The majority of CSPs offer free data transport to their data centers (Data Transfer IN To Amazon EC2 From The Internet Costs $0.00 Per GB, for example). The cost of downloading data varies depending on the volume (e.g., Data Transfer OUT To Amazon EC2 From Internet $0.12 per GB per month for data quantities consisted of between 1GB and 10TB, while it is free for lesser data volumes) [61].

Second, to improve performance, the majority of OLAP engines use intra-query parallelism. A complex single question is divided up into smaller requests, the burden is distributed among several processors, and finally post-processing is done in order to present the final query response. Subject to intra-query parallelism, three variables have an impact on the query's final response time. First, startup costs, which are incurred when several processes are launched in order to handle multiple sub-queries simultaneously.

If there is a high level of parallelism, the setup time for these processes may take up the majority of the calculation time. Second, Skew costs demonstrate that in a distributed system, the slowest performing activities that are running in parallel decide the overall execution time. Third, interference costs, which are related to the amount of time that processes are not being used. In fact, processes that use shared resources (such as the system bus, disks, or locks) face competition from one another and must wait for other processes to complete their tasks.

Pig script reaction times were measured in trials (details in Appendix B), and the results show a concave curve (Fig. 2), with an optimal response time for each cluster size and performance degrading after this point. The performance increase slope (from N to N') for cloud computing should also be stated in a dollar amount ($). In fact, the system scales out horizontally and more instances are provisioned to achieve this improvement in response time.

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

Scalability: Scalability is a system's capacity to enhance throughput overall when faced with a heavier load and more hardware resources. Cloud services should ideally have a set cost per processed business question and linear scaling. The current TPC-H implementation gauges a system's ability to handle a static workload. We suggest that the benchmark for data warehousing evaluate the system being tested under a constant load and calculate the throughput as a result. We can change the workload on a time scale, say every hour, and count the number of business inquiries answered throughout that time period to quantify this requirement.

While a non-scalable system records fewer business questions answered under a larger load, a scalable system should keep the same number of business questions handled within a time period.

Elasticity: To adapt the system capacity at runtime to the changing workload, elasticity adds and removes resources without affecting service. First, the metric should evaluate the system's ability to add or remove resources without affecting service, and if it does, it should report both the scaling latency (the time it takes for a system to scale up or down horizontally) and the scale-up cost (the price of newly acquired resources, in dollars) or the scale-down gain (the price of newly released resources, in dollars), as appropriate.

High Availability: The likelihood of a distributed storage system failing is increased when data is scattered over several drives. There are many methods that can be used to construct highly available distributed data storage systems. They typically employ parity calculus or replication. The latter method makes use of systematic erasure-codes, such as Tornado, Low-Density Parity-Check, and Reed Solomon (RS) codes. Data management is simple with replication. However, replication always has a higher storage expense than systematic erasure codes.

Erasure codes can offer services with less storage overhead than replication methods when a specific level of availability is targeted. High availability with erasure codes reduces storage costs for data warehousing, especially for massive data of the write-once kind (i.e., not subject to delete refreshes). Data recovery, however, is trickier than replication. Erasure codes have been examined and shown to be effective for grid systems and highly available distributed storage systems, respectively [18, 19]. The storage space requirements for several file high-availability strategies are shown in Fig.3. 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%.

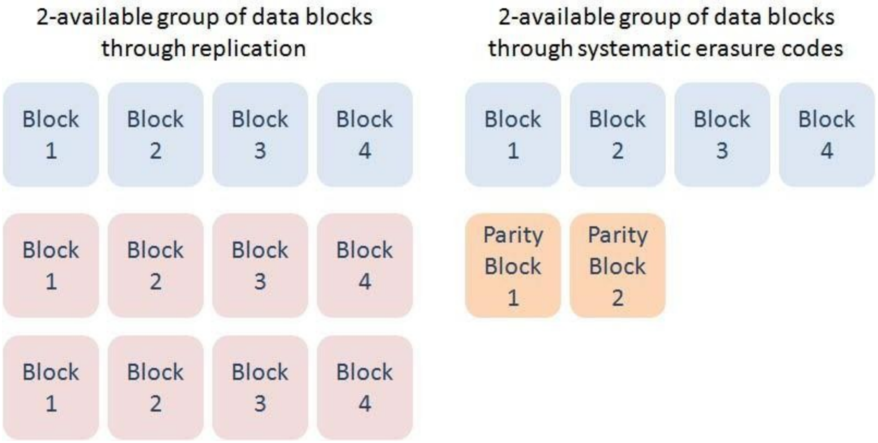

Fig. 3. Replication vs. Erasure Codes for a group of 4 data blocks.

Replication is used by CSPs to improve the availability of stored data and avoid service interruption. Additionally, they provide replica management in many geographically dispersed data centers. This makes it possible to recover from a data center disaster in its entirety. However, the majority of CSPs do not cater high availability services to their clients.

The end-user should be informed of the cost of making their data highly-available through various high availability strategies (i.e., for both synchronous and asynchronous refreshes) when using cloud data warehousing, and various levels of availability should be provided to allow for customization of the recovery capacity after disasters.

As a result, in addition to assessing the recovery cost, the benchmark should also include metrics reflecting the cost of various intended levels of availabilities (1-available,. . . , k-available, or the number of failures the system can accept). We suggest two measures that represent the cost of maintaining a k-available system ($@k), where k is the desired level of availability, and a metric that represents the customer-perceived cost of recovery represented in time and lost system productivity due to hardware breakdown ($). The CSP should be billed for the latter.

Cost-Effectiveness and Cost-Performance: Cloud-based solutions should assist businesses looking to reduce expenses without sacrificing productivity and service quality. As a result, it is becoming increasingly important to comprehend, monitor, and proactively control expenses throughout the cloud from the viewpoints of performance and effectiveness. In fact, the user may prefer to operate his application more efficiently, which ensures a maximal computation at lowest expenses, rather than focusing on the shortest possible execution time. The best hardware configuration should be determined based on performance and

effectiveness, and included in the cost management plan; This presupposes systematic resource usage monitoring. We suggest calculating the configuration cost ($) to performance and resource utilization ratio for these objectives. The ratio of used to allocated resources is known as resource utilization. Take note of how usage and allocated resources change over time.

Service Level Agreements: An SLA is a contract that a service provider and its customers enter into. SLAs record the promises that have been agreed upon between a service provider and its client. They specify the characteristics of the offered service, such as the maximum response times, minimum throughput rates, and data consistency, and they specify consequences if the service provider does not meet these goals. Scalability, elasticity, performance (throughput and reaction time are also taken into account), high- availability, and CSP independence are the SLA categories for data warehousing in the cloud.

In the latter case, the business should have no trouble switching to a different Cloud Service Provider (CSP) and receiving its data back in a common format. This will reduce losses in the event that the CSP levies exorbitant fees, demands the purchase of new software, or declares bankruptcy.

c) OLAP vs. OLTP Benchmarking in the Cloud

We are greatly influenced by the work in [13][14]. The latter looked into cloud-based OLTP benchmarking. We offer a thorough comparison of OLAP and OLTP benchmarking in the cloud in Table I.

Conclusion

The three main reasons for moving data warehouse systems to the cloud are: I lower capital costs through measured services, where infrastructure, platforms, and services are offered on a pay-per-use basis;

(ii) faster elasticity; and (iii) quicker provisioning for a better cost-performance trade-off. In this study, we contend that the most well-known OLAP benchmark, TPC-H, does not accurately reflect cloud properties. We also provide new benchmarking criteria and indicators, including high performance, high availability, cost effectiveness, cost performance, scalability, elasticity, and service level agreements (SLAs), for data warehouse deployment in the cloud. In upcoming work, we will evaluate the most well-known CSPs that Google, Amazon, and Microsoft provide for data warehousing using a cloud-based TPC-H benchmark.

TABLE I .OLAP VERSUS OLTP IN THE CLOUD.

	Data Warehouse System Deployement in the Cloud and OLAP Workload Run
Goals from Customer Perpective	Fast upload and download of huge data sets, Browse a multidimensional view of
Horizontal scale-up Added Value	More Bytes processed per hour (+BpH)
Cost	High storage cost for Data Warehouse System Complex and costly workload (CF
Metric of Interest	$/BpH (cost of Bytes processed by hour)
Recommended High-Availability Schema	Systematic Erasure Codes & Replication
Distributed Processing Features	Intra-parallelism within a business question
Risks under Peak Loads	The compagny may do not take decisions in- time ($)

The experiments that were done to compare Apache Hadoop/Pig Latin to the TPC-H benchmark are described here. [20] describes the conversion of the TPC-H workload from SQL to Pig Latin as well as an analysis of the tasks written in Pig Latin for the TPC-H workload. Borderline nodes of the GRID5000 platform are placed at the Bordeaux facility and make up the hardware system configuration for performance evaluations. Each Borderline node has 32 GB of memory and four Intel Xeon CPUs running at 2.6 GHz with two cores each. Lenny Debian Operating System is used by all nodes.

We test Pig's performance with a range of cluster sizes and data volumes. The performance results for Pig for N=3, 5, 8 nodes (corresponding to 2, 4, and 7 Hadoop Task Trackers/data nodes or workers and one Hadoop master) are shown in the following figures. We produced TPC-H data files with SF=1, 10, and 1.1 and 11 GB, respectively. TPC-H workload response times for 1.1GB of data are shown in Fig. 4. It is important to note that doubling the size of the Hadoop cluster has no impact on how quickly pig scripts execute. Business questions that do not require join operations, such as Q1 and Q6, have faster execution times as cluster size grows. TPC-H workload response times for 11GB of data are shown in Fig. 5. Contrary to results corresponding to a volume of 1GB, reaction times are generally improved with increasing cluster size. Business-related complex questions like Q2, Q11, Q13, and so forth are unaffected by cluster size. Pig performs admirably when compared to results for 1.1GB of data shown in Figure 4 when the data amount is multiplied by 10. In fact, the average response time for a 1.1GB warehouse is two times slower than the average response time for an 11GB TPCH warehouse, regardless of the cluster size (N=3, 5, or 8). We come to the conclusion that while cluster size is significant, workload perormances are not always improved by it.

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

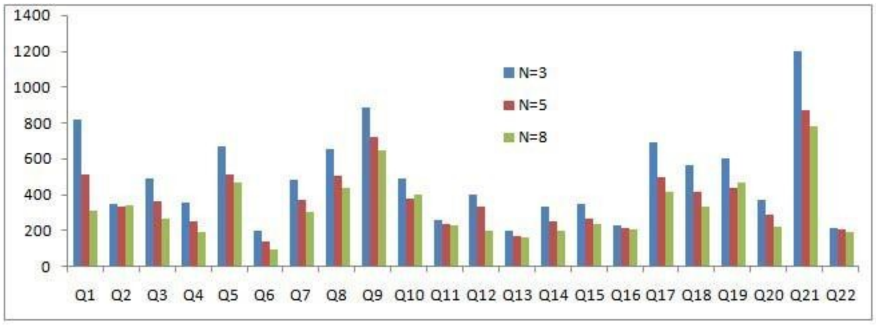

Fig. 5. Pig performances (sec) for 11GB of TPC-H data (SF=10).

References

1. G. group, –?—— http://www.gartner.com/it/page.jsp?id=920712, ?.
2. Forrester, –Sizing the cloud,—— http://www.forrester.com, 2011.
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.

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.

1. 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.
2. R. Moussa, –Massive data analytics in the cloud: Tpc-h experience on hadoop clusters,—— vol. 4, no. 3, 2012, pp. 113–133.
3. J. Gray, –Sort benchmark home page,—— http://research.microsoft.com/barc/SortBenchmark/, 2008.
4. 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.

W. Sobel, S. Subramanyam, A. Sucharitakul, J. Nguyen, H. Wong,

A. Klepchukov, S. Patil, A. Fox, and D. Patterson, –Cloudstone: Multiplatform, multi-language benchmark and measurement tools for web

2.0,—— in Proceedings of Cloud Computing and its Applications, 2008.

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.

1. C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, –How is the weather tomorrow?: towards a benchmark for the cloud,—— in DBTest, 2009.
2. 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.
3. L. Ang, Y. Xiaowei, K. Srikanth, and Z. Ming, –Cloudcmp: Shopping for a cloud made easy,—— in USENIX HotCloud, 2010.
4. Chao Xue, –Scalability Analysis of Request Scheduling in Cloud Computing—— in TSINGHUA SCIENCE AND TECHNOLOGY, 2019
5. K. Sato, –An inside look at google bigquery,—— https://cloud.google.com/files/BigQueryTechnicalWP.pdf , 2013.
6. 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.

1. M. Pitk¨anen, 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.
2. R. Moussa, –Tpc-h benchmarking of pig latin on a hadoop cluster,—— in ICCIT, 2012, pp. 96–101.
3. 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.