I.J. Modern Education and Computer Science, **2015, 6, 50-58**
Published Online June 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2015.06.08

Perspective of Database Services for Managing Large-Scale Data on the Cloud: A Comparative Study

Narinder K. Seera
Assistant Professor, Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi (INDIA)
Email: narinder.k2010@gmail.com

Vishal Jain
Assistant Professor, Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi (INDIA)
Email: vishaljain_usit@yahoo.in

Abstract—The influx of Big Data on the Internet has become a question for many businesses of how they can benefit from big data and how to use cloud computing to make it happen. The magnitude at which data is getting generated day by day is hard to believe and is beyond the scope of a human's capability to view and analyze it and hence there is an imperative need for data management and analytical tools to leverage this big data. Companies require a fine blend of technologies to collect, analyze, visualize, and process large volume of data. Big Data initiatives are driving urgent demand for algorithms to process data, accentuating challenges around data security with minimal impact on existing systems. In this paper, we present many existing cloud storage systems and query processing techniques to process the large scale data on the cloud. The paper also explores the challenges of big data management on the cloud and related factors that encourage the research work in this field.

Index Terms—Big Data, Query Processing, Cloud Computing, Distributed Storage

I. Introduction

The problem with a traditional database management system starts when the quantity of data gets beyond the storage capacity of the disk, the queries start trouncing the CPU for resources and the result sets go out of RAM. The database systems need to be re-engineered using innovative technologies to handle this growing volume of information. Rise of such problems in the organizations facilitated the emergence of cloud data stores and big data. Big data [1][2][3] refers to high volume, velocity and variety of information asset which requires new forms of processing to facilitate enhanced decision making, insight discovery and process optimization. Big data may come from a variety of sources including web logs, business information, sensors, social media, remote device and data collected through wireless sensor networks. Big data needs a cluster of servers for its processing, which cloud can readily provide. Cloud-based big data services offer significant advantages in reducing the overhead of configuring and tuning your own database servers.

As the size of data is increasing exponentially, researchers have begun to focus on how 'big data' can potentially benefit digital world of organizations [4]. Although managing the increasing complexity of data is critical to the success of the organization, new initiatives should deliberate on how to mine the information to generate high revenues from the businesses. The data driven decision making requires the use of excellent technologies to capture, store and efficiently process the big data, which is often unstructured [5]. Converting big data programs into successful activities that deliver meaningful business insight and provide sustained high-quality customer relationships can be costly, risky and sometimes unproductive.

There have been various techniques and algorithms devised for big data analytics [6]. Mining big data and applying effective algorithms to produce productive results is out of scope of this paper and can be found in the existing literature in this filed [7][8][9]. This survey intends to find out various existing systems for big data management, comparing their key features and the techniques they adhere to.

The remainder of this paper is organized as follows – We begin in Section II with the motivation behind the survey. Related work is given in Section III. A detailed discussion on the existing cloud based database systems and different query processing techniques is presented in Section IV and V respectively. We discuss the opportunities and applications of big data in various fields in Section VI. At the end we conclude the paper with the prospect of future work in this field.

II. Motivation of the Survey

In early days of computing flat file systems were used to organize the data pertaining to the organization. But due to lack of standards and decentralization of data in flat files, database management systems came into

existence. The main advantages offered by these systems are centralization of data, storing relationships between different objects or entities and easy retrieval of data from the databases. But these traditional relational database systems are not capable to processing large data sets like 10TB of data or hundreds GB of images. As a result non-relational databases were evolved that can handle large scale data and can process terabytes or petabytes of data efficiently. They are also referred to as NoSQL databases [10]. The non-relational data model is not a replacement of relational data model; they are the alternatives of each other.

The biggest difference between relational and non-relational data models is the underlying data structures used by them. The relational model stores the data in tabular form and uses SQL language to fetch the data from databases. On the other hand, non-relational databases do not rely on the concept of tables and keys; they require data manipulation techniques and processes to handle unstructured or semi-structured big data. The performance of these databases can be evaluated on three main aspects – elasticity, fault tolerance and scalability.

NoSQL (Not Only SQL) [11] is an approach to data management useful for very large sets of data which is dispersed over multiple nodes in the cloud. It encompasses a broad range of technology and architectures that seek to resolve the scalability and big data performance. A NoSQL database provides a mechanism for storage and retrieval of unstructured data that is modeled using some data structures other than the tabular structure used in relational databases. It provides finer control over availability and simplicity of design. NoSQL approach is especially useful when:

- Data is unstructured
- Data is stored remotely on multiple virtual servers in the cloud.
- Constraints and validation checks are not required.
- Dealing with growing list of elements such as twitter posts, internet server logs etc.
- Storing relationship among data elements is not important.

The NoSQL databases can be roughly categorized in four categories i.e. Column oriented, document oriented, key value based databases and graph data stores. Column oriented databases store data in columns instead of rows. The goal is to efficiently read and write data from hard disk in order to speed up the time it takes to return the result of a query. There are two main advantages of this approach First, data is highly compressed that permits aggregate functions to be performed very rapidly. Second, data is self-indexing, so it uses less space. Document oriented databases stores semi-structured data in form of records in documents in less rigid from. Documents are addressed in the database via a unique key that represents the document. This key can be used later to access the document from the database. To speed up document retrieval, the keys are further indexed. Key-Value store is a non-trivial data model where the data is represented as a collection of key-value pairs. Graph models represent the data in tree- like structures with nodes and edges connecting each other through links (or relations). Fig 1 summarizes different categories of database management systems as relational and non relational.

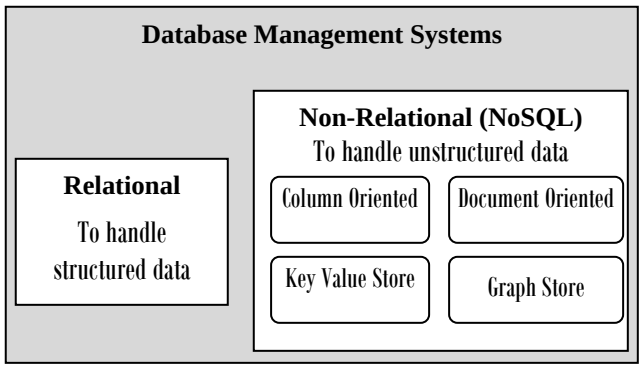

Fig. 1. Data models – Relational vs. NoSQL

Current studies reveal that many researchers have proposed different systems that provide scalable services for data management on the cloud and offer high data processing capabilities [12].

Here we present the essentials of cloud based database systems that are vital for both data storage and data processing.

Requisites of data management and data processing in Cloud

- High Performance–The workload is distributed among multiple nodes to perform high speed computing. Researchers have proved that adding more servers to process the data, linearly improves the performance.
- Scalability – The cloud systems are flexible enough to scale up and scale down to meet your demands [14].
- Availability – of data
- Resource sharing - On-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) is highly desirable with minimal management effort or service provider interaction.
- Fault Tolerance- In case of failure of any node, the other available nodes should take over the control and do not let the whole system goes down.
- Aggregation of applications and data is also an important aspect of cloud systems.
- Elasticity [15] –The cloud database systems are capable to acclimatize the changes in the workload by wisely allocating and de-allocating the available resources on demand.
- Query Processing – Although we have various providers that provide databases services on cloud but most of them do not support all the features of SQL like joins and nested or complex queries. The cloud databases should be capable of handling users' SQL queries that may require results from

different data sources.

- Representation of heterogeneous data [4] - Sheer size, huge volume and velocity are some of the terms that define Big Data. The main issue of cloud databases is to deal with such heterogeneous data – to design schemas for it and to provide services to manage and process it.
- Security and privacy [13][17] – As with other aspects of the cloud, high-security computing infrastructure is required to secure data, network and host in the cloud environment. [16] Physical security defines how one control physical access to the servers that support your infrastructure. To provide data security on someone else's server in cloud, two things can be done; either encrypt the sensitive and confidential data in the separate database or keep off-site backups. Network security can be implemented in terms of firewalls. EC2 provides security groups that define traffic rules that govern what traffic can reach virtual servers in the cloud. Host security describes how the host system is configured to handle attacks and how the effects of an attack on the overall system can be minimized. It is always good to have a complete suite of antivirus software or some other tools with no security holes.

The existence of a variety of systems and processing techniques to support big data and to analyze it; forms the foundation of this paper and motivated us to give a comprehensive overview of different kinds of cloud based databases systems with their key features.

III. Related Work

A large work in the area of database systems has been already done by various researchers in this field. The work in the existing literature spans from query processing techniques to query optimization, storage systems to data access methods and from centralized systems to distributed data processing systems etc.

Today hundreds of cloud based database systems exist which can be roughly categorized over two main dimensions – storage system aspect and query processing aspect.

On the storage system aspect, the widely accepted file storage available on the cloud is ***distributed file storage*** that allows many clients to access same data simultaneously from remote machines. Two such widely accepted distributed file storage are GFS (Google File Storage [18]) and HDFS (Hadoop Distributed File System) [19]. Google App Engine [20][21] and Amazon EC2 [22], are two widely used data stores which are used to manage data through web applications. Application developers and many technologists from different industries are taking advantage of Amazon Web Services [23] to meet the challenges of storing digital information. Amazon Web Services offer an end-to-end range of cloud computing resources to manage big data by reducing costs and gaining a competitive advantage.

On the processing aspect, some database systems provide Databases-as-a-Service (DaaS) [24] capabilities with either full SQL query support or NoSQL support. Examples of systems that fully support SQL include Amazon Simple RDS, Cloudy and Microsoft SQL Azure. Popular NoSQL databases are SimpleDB [42], MemCacheDB, Voldemort, Cassandra [46] etc. And the companies that are using NoSQL are NetFlix, LinkedIn, Twitter etc. AmazonDB does not fully support SQL, it supports only a subset of SQL queries.

Most NoSQL systems use MapReduce techniques [26] for parallel query processing [25] where the queries are divided into smaller set of operations and the answers of sub queries are joined back to get the end results. Although this reduces the query processing time by parallelizing the tasks and offer high scalability but it has certain limitations too.

[27] and [28] identified some drawbacks of Map-Reduce techniques and proposed some alternative solutions to overcome them. Comparison between map-reduce systems and parallel databases [29][30] highlights following points:

- Map-Reduce systems do not take the advantage of indexing procedures for selective access of data. It is desirable to optimize the data modeling to improve data retrieval performance.
- Parallel databases systems use relational data model to manage the data where applications can use SQL programs whereas map-reduce systems are schema free and users are independent to code their own map and reduce functions.
- Map-reduce does not have any query optimization plans to minimize data transfer across nodes. This results in lower performance and low efficiency.
- Parallel systems are designed to minimize the query processing time by carefully distributing the task into available machines. On the other hand, map reduce systems are weak in load balancing. The partitioning and distribution of input data set among n number of nodes does not always guarantee equal share of work.
- Compared to parallel systems, map reduce systems are more fault tolerant and gracefully handles the node failure by assigning the process to another available node.

Many researchers have designed cloud based database systems to overcome the limitations of traditional systems such as managing huge volume of unstructured or semi-structured data, storing data on distributed storage, processing the data present on multiple nodes (or parallel processing), providing high availability and scalability to the clients and few others. But most of them do not support the features of full relational data model and they lack the querying capabilities of powerful SQL language. In contrast, Parallel database systems are robust and high-performance data processing platforms designed to handle clusters with a static number of nodes but their main drawback is that they do not have elasticity feature

which is an essential aspect of cloud based systems.

These days distributed data processing has become a major requirement for all businesses and companies. Almost all major DBMS vendors offer distributed data processing to support various web services and new applications that run on cloud. The major reason for the exponential growth of distributed and parallel processing systems lies in the potential benefits offered by them such as managing large-scale data, providing scalable and high performance services to applications etc. Several cloud service providers provide platform-as-a-service (PaaS) [31] solutions that eliminate the need to configure databases manually, thus reducing maintenance work of the organization. The primary providers of such big data platform are Amazon, Google and Microsoft.

IV. Cloud Based Database Systems

In this section we present some cloud based database systems with their key components and their features. These features form the basis of the survey and are summarized in Table 1 given at the end of the paper.

epiC [32] is an elastic and efficient power-aware data-intensive cloud based system that supports both OLAP and OLTP jobs [33]. It has the potential of parallel systems that have processing engine as well as storage engine. It has three main components – the Query Interface, the Elastic Execution Engine (E^3) and the Elastic Storage System (ES^2).

- The main function of Query Interface is to determine the type of query fired by the user and to monitor its processing status. If the query is an analytical query then it is forwarded to OLAP controller where they are processed via parallel scans and if it is a simple select query then the OLTP controller processes it using indexing and query optimization plans. Both OLAP and OLTP controller use underlying ES^2 to provide transactional support.
- Elastic Execution Engine has a master node that takes the analytical tasks from OLAP controller and distributes the task among all the available nodes for parallel execution.
- Elastic Storage System has further sub modules for data partitioning, load adaptive replication, transaction management and managing secondary indexes.

SQLMR [34] is mainly designed to bridge the gap between traditional SQL like queries and the data processing capabilities of MapReduce. It combines the scalability and fault tolerance feature of MapReduce with easy to use SQL language. There are four main components of SQLMR –

- SQL-to-MapReduce Compiler that inputs SQL queries from the users and translates them into a sequence of MapReduce jobs.
- Query Result Manager that maintains a result cache to store the results of the queries. When a query is fired, first the cache is examined to compare the results otherwise it is parsed to generate optimized MapReduce code.
- Database Partitioning and Indexing Manager that manages data files and indexes.
- The Optimized Hadoop System framework which is responsible for distributed processing of large data sets on clusters.

Distributed Key Value Store (KVS) [29] is another distributed data store on the cloud which uses a special data structure to store the data on multiple servers. It is composed of data items and associated keys that redirects read and write requests to the appropriate servers. The advantages offered by distributed KVS include high performance and speed, aggregation processing in which multiple data are aggregated to produce results and resistance to failure as multiple copies of the same data is kept at multiple servers.

BigTable [35] is a distributed storage system to manage voluminous structured data with a simple data model. It does not support all the features of relational databases but tries to fulfill all major requirements of cloud systems such as high availability, performance and scalability. The well-known applications of BigTable are Google Earth, Google Finance, Personalized search, web indexing etc [36].

BigTable is basically a sparse and persistent multidimensional sorted map which is indexed by a row key, a column key and a timestamp. Column keys are grouped into sets called column families where each column family can have one or more named columns of same type. Each row in the table contains one or more named **column families**. Each cell in the BigTable contains multiple versions of the same data and each version is assigned a unique timestamp. BigTable uses Google's file system to store BigTable data, log and data files.

It has three major components:

- A Master Server – to assign tablets to tablet servers. A tablet is a dynamic partition of a row called a row range. The master keeps track of the workload of tablet servers and the tablets which are unassigned.
- Tablet Servers – to manage and handle read or write request to a set of tablets assigned to it. These tablet servers can be dynamically added or removed depending upon the workload.
- A library which is linked to every client.

Big Integrator [37] facilitates the queries to access data from both relational databases and cloud databases. To achieve this functionality it has two plug-ins called absorbers and finalizers. For each kind of data source the system has separate wrapper modules that generate queries for them. For example, the RDBMS wrapper accepts the query from the system and translates it into

SQL query which is sent back to the underlying Relational database. Similarly the BigTable wrapper converts the query into corresponding GQL query for BigTable data store. The BigIntegrator wrapper module has further sub modules with defined functions – importer, absorbers, finalizers and interfaces, to carry out query processing. The results returned by different data sources are joined to get the end result.

Table 1. Characteristics of existing cloud based systems

Database Systems	Data Storage Used	Processing Engine Used	Query Language support	Data Access methodology	Query Processing Technique	Features
epiC [32]	ES2 (Elastic Storage System)	E3 (Elastic Execution Engine)	SQL like language used	Distributed secondary indexes – B+ trees & multidimensional indexes used	For OLAP queries – parallel scans OLTP queries – indexing & local query optimization	Highly scalable
SQLMR [34]	Database & Indexing Manager	SQL-to-MapReduce Compiler	SQL programs translated into mapReduce jobs	Database partitioning & indexing used	Query Result Manager used to cache results	Highly Scalable
Distributed KVS [29]	Distributed Data Manager		DISPEL used	Distributed keys are used to access values	Aggregation Processing	High Performance & availability
BigTable [35]	Distributed Storage System (GFS)	GAE (Google Application Engine)	GQL (Supports query by row key)	Row keys, column keys and timestamps used	----	High Performance, Scalability and availability
BigIntegrator [37]	Cloud databases & relational databases	BigIntegrator Query Processor	SQL / GQL	Algebraic expressions & access filters used	RDBMS wrappers and BigTable wrappers	----
Optique [38]	Shared databases	Asynchronous Execution Engine	SQL	OBDA (Ontology based data access)	Distributed query optimization & processing	Elastic & scalable query execution
Microsoft SQL Azure [41]	Cloud based Storage system	----	T-SQL	Key value access	----	High availability &performance
Others like Mongo DB, Cassandra, DynamoDB [42][46][47]	Cloud based Storage system		NoSQL	Key based access	Map and reduce techniques	Highly scalable

The BigTable wrapper has two components for server and clients. The BigTable wrapper server is a web application that accepts the client requests, translates it into GQL and returns the result to BigTable wrapper client.

Optique [38] platform has a distributed query processing engine to provide execution of complex queries on the cloud. The system is developed to deal with two different scenarios. First, to process the queries that need to access terabytes of temporal data coming from sensors with data stored in relational databases. Second, to process the queries that access data from multiple databases with different schemas. For this purpose, it uses OBDA (Ontology Based Data Access) [39][40] architecture which has four major components:

- **Query Formulation component**- to formulate queries.
- **Ontology and Mapping Management component**– to bootstrap ontologies and mappings during the installation of the system.
- **Query Transformation component**- that rewrites users' queries into queries over the underlying data sources,
- **Distributed Query Optimization and Processing component**- that optimizes and executes the queries produced by the Query Transformation component.

Microsoft Azure SQL Database [41] provides relational database-as-a-service capabilities to application developers and users. The application code and database can reside on same physical location or in distributed environment with tightly coupled servers and storage systems. To make the system work efficiently, it is divided into three layers - the client layer, the services layer and the platform layer.

The client layer is used by the application to interact directly with the database. The services layer act as a gateway between the client layer and the platform layer. It manages all the connections between the users' application and the physical servers that contain data. The platform layer is the main layer where the data actually resides. It consists of many instances of SQL Server.

Besides above mentioned cloud database systems there are several other systems that have been successfully adopted by various companies. Some of them include Amazon SimpleDB, CouchDB, MongoDB [42], Cassandra, Splunk, Apache HadoopDB etc. A precise

comparison of the features and data models deployed by theses systems is briefed in Table1.

V. Query Processing Languages And Techniques

In this section we will briefly discuss the types of query processing methods available on the cloud to manage big data.

Map-Reduce techniques [25][27][32][44] – MapReduce is a programming paradigm that processes massive amounts of unstructured data in parallel across a distributed cluster of processors. It offers no benefit over non-distributed architectures.

This technique is organized around two functions – map and reduce. The "map" function divides a large data set of input values into smaller data sets where individual elements are broken down into a number of key-value pairs. Each of these elements will then be sorted by their key and are sent to the same node, where a "reduce" function is use to merge the values (of the same key) into a single result. As shown in Figure 2, in the splitting phase, a given input is first divided into three sets by three map functions map1, map2 and map3 where they are processed. Then the sort function sorts these smaller sets of input by their keys and send them to the next phase, called the merge phase, where the reduce functions merge the values of the same key into a single output.

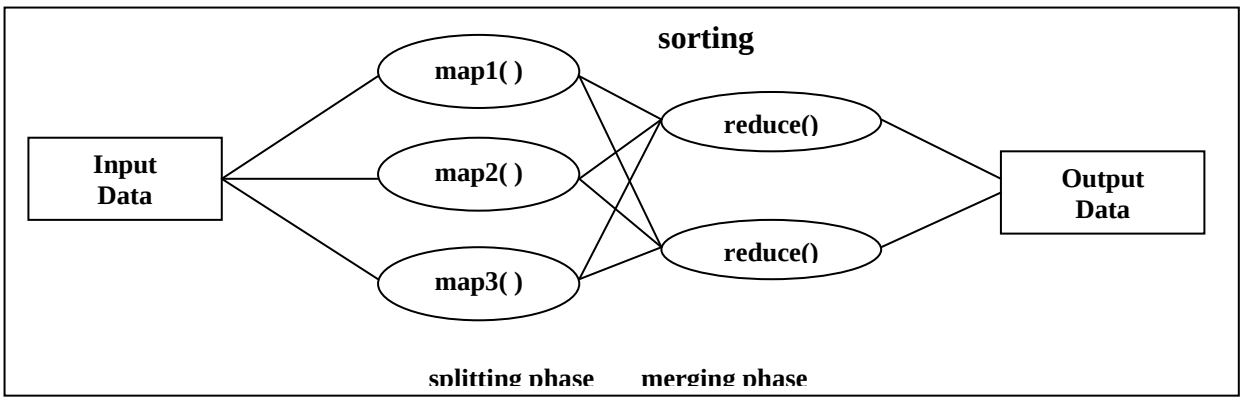

Fig. 2. Map-Reduce functionalities

SQL (Structured Query Language) –It is a most widely used and powerful query language designed to handle data in Relational Database Management Systems. It has been universally accepted by a wide range of traditional relational database systems as well as cloud based database systems. Some of the database systems that use SQL are Oracle, Microsoft SQL Server, Ingres, Simple DB etc.

Although it is originally based on the notion of relation algebra and tuple calculus, it also has procedural capabilities to write complete programs to retrieve data from databases. There are two approaches to use SQL with applications :

1. Call SQL commands from within a host language like C# or Java. For this purpose special APIs (Application Program Interface) are created such as in JDBC.
2. EEmbedded SQL - Embed SQL in host language where a preprocessor converts the SQL statements into special API calls and then a regular compiler is used to compile the program.

GQL (Google Query Language) – It is a SQL-like language which is used to fetch entities and keys from 'Google Application Engine' (GAE) [20][21] data store. It does not support 'joins' due to two main reasons:

First, it becomes inefficient when queries span more than one machine.

Second, same query can process large number of records with almost same speed.

Others - Even most database systems use SQL, most of them also have their own additional proprietary extensions that are used only on their systems. For example, Transact-SQL (T-SQL) is Microsoft's and Sybase's proprietary extension to SQL. **Transact-SQL** is an essential to Microsoft SQL Server. All applications that communicate with an instance of SQL Server do so by sending Transact-SQL statements to the server, despite of the user interface of the application. Besides SQL, T-SQL includes the concept of procedural programming, local variables, temporary objects, system and extended stored procedures, scrollable cursors, various other string handling functions, date processing and mathematical functions and changes to the delete and update statements etc.

There are some other SQL-like languages like **HiveQL** which is purposely used by Hadoop [44][19] Hive [45] database system. This language also allows traditional map - reduce programmers to plug-in their custom map and reduce functions when it is not convenient to express the logic in HiveQL. It is also a primary data processing method for Treasure data. Treasure Data is a cloud platform that permits its users to store and analyze the data on the cloud. It manages its own Hadoop cluster, which accepts queries from users and executes them using the Hadoop MapReduce framework.

VI. Opportunities And Applications Of Big Data

Today almost all organizations use Big Data for business intelligence to analyze inside and outside business data to perform risk assessment, brand management, customer acquisition and many other managerial tasks [4]. Besides all these activities Big Data has also proved its importance in various other fields.

In the field of Astronomy, with the advent of Sloan Digital Sky Survey [48], the astronomer's job is greatly influenced from taking pictures of the astronomical bodies to find and analyze interesting objects and patterns in the sky. SDSS is one of the most striving and influential surveys in the history of astronomy that obtained deep, multi-colored images of galaxies, stars, quasars etc and created 3-dimensional maps of those captured objects.

It has greatly influenced the field of Biological Sciences or bioinformatics [49]. Genetic research is now driving towards Big Data to seek solutions for acquiring and generating volume of DNA-sequence that is the basic genetic study. With suitable analytical tools, genetic research may answer everything like curing cancer, developing superior crop varieties and increased treatment efficiency and medical advancements.

Big Data has also marked significant benefits in the field of Research and education [50][51]. Educating individual student is one of the biggest advantages of technology and big data help teachers personalize learning. Using Big Data for data-driven teaching increases transparency and accountability in evaluating trend in education. Digital systems facilitate real time assessment for mining information. Students can be given personalized quizzes and lessons that try to find their weaknesses. The answers can be analyzed to track whether students have mastered the concepts. A pattern of wrong answers give clues on why students selected the incorrect answers. It allows teachers to mine learning patterns of the students and pinpoint the problem areas and enables them to do a better job.

The field of The Urban planning [54][55] or geographical sciences is also not untouched with the potential benefits of Big Data. Big data produced by so many places and processes (or Geographic data) contain either explicit or implicit spatial information which can be mapped and analyzed to provide new insights to urban developments. Geo-located information contains data accessed from sensors, social networking sites, online booking of various transport modes, mobile phone usage, online credit and debit card transactions etc. One of the major applications of Big Data in the field of geographical sciences is in the transport sector which aims to control congestion in road traffics so as to control pollution and accidents.

Talking about Financial systems [50][56][57], banking and finance industries are also taking a business-driven and realistic approach to big data. For Financial institutions big data is imperative due to four main reasons. First, companies require larger market data sets and fine granularity of data to make their forecasts. Secondly, to influence large amounts of consumer data across multiple service delivery channels to uncover consumer behavior patterns. Third, to improve enterprise transparency and audit-ability. And lastly, to deal with the acute stress of economic uncertainty and seeking new revenue opportunities.

VII. Conclusion & Future Scope

In this paper we compared the features of various distributed database systems that use underlying cloud storage to store data and parallel database systems which work on clusters of nodes based on shared nothing architecture. We also compared the approaches followed by various relational and NoSQL database management systems in terms of their query processing strategies and found that many systems exist that aim to fill the gap between SQL queries and map reduce paradigm by converting users' queries into map-reduce tasks. We concluded that although map reduce technique speeds up the query processing by parallelizing the tasks and provide scalability, it provides limited performance capabilities over different class of problems and offer few facilities over ad-hoc queries. It neither uses any specific data model to manage data nor does it use any indexing method to access data, therefore how to increase the efficiency of map-reduce in terms of I/O costs is still a research problem that needs to be addressed. Precisely, different systems target different design problems and opt different processing methods but realizing the potential of parallel systems with map-reduce programming is a new hope for the next decade.

We also discussed about cloud computing which is a perfect match for Big Data as it provides high availability of resources and scalability. The technology innovations in the field of data mining are remarkable and will continue to prevail in the next generation of computing. The demand is arising to devise effective algorithms to extract knowledge and interested patterns from big data. The extensive exploitation of Big Data and data mining in digital world will definitely harmonize the growth of each other to become a dominant technology of the future.

References

[1] SteveLaValle, Eric Lesser, Rebecca Shockley, Michael S. Hopkins and Nina Kruschwitz, "*Big data, Analytics and the Path from Insights to Value*", December 2010.

[2] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers, "*Big data: The next frontier for innovation, competition, and productivity*", May 2011.

[3] Divyakant Agarwal, S.Das, S.E. Abbadi, "*Big Data and Cloud Computing: Current State and Future Opportunities*" EDBT 2011, March 22–24, 2011, Uppsala, Sweden.

[4] Arup Dasgupta, "*Big Data-The future is in Analytics*" published in Geospatial World April 2013.

[5] Divyakant Agrawal, Elisa Bertino, Michael Franklin, "*Challenges and Opportunities with Big Data*".

[6] Van Renesse, R., Birman, K.P., Vogels, W.: Astrolabe: *A robust and scalable technology for distributed system*

monitoring, management, and data mining. ACM Trans. Comput. Syst. 21(2) (2003)

[7] A.N.Paidi, "*Data mining: Future trends and applications*", International Journal of Modern Engineering Research, vol 2, Issue 6, Nov-Dec 2012, pp 4657-4663.

[8] Venkatadri M., L.C.Reddy, "*A review on data mining from past to the future*", International Journal of Computer Applications (0975-8887), Volujme 15, No. 7, Feb 2011.

[9] Hans-Peter Kriegel, Karsten M. Borgwardt, Peer Kröger, Alexey Pryakhin, Matthias, Schubert, Arthur, "*Future trends in Data Mining*", Springer Science+Business Media, LLC 2007

[10] Katarina Grolinger, Wilson A Higashino, Abhinav Tiwari and Miriam AM Capretz, "*Data Management in Cloud environments: NoSQl and NewSQL data stores*" Journal of Cloud Computing: Advances, Systems and Applications 2013, pp. 2-22.

[11] Phyo Thandar Thant, "*Improving the availability of NoSQL databases for Cloud Storage*" available online at http://www.academia.edu/4112230/Improving_the_Availability_of_NoSQL_Databases_for_Cloud_Storage.

[12] A. Pavlao, E.Paulson, A. Rasin, D.Abadi, S.Madden, M.Stonebraker, "*A Comparison of approaches to large-scale data analysis*" SIGMOD'09, June 29–July 2, 2009, Providence, Rhode Island, USA.

[13] R. Gellman, "*Privacy in the clouds: Risks to privacy and confidentiality from cloud computing",* Prepared for the World Privacy Forum, online at http://www.worldprivacyforum.org/pdf/WPF Cloud Privacy Report.pdf,Feb 2009.

[14] Pawel Jurczyk and Li Xiong, "*Dynamic Query Processing for P2P data services in the Cloud*". Emory University, Atlanta GA 30322, USA

[15] Ioannis Konstantinou, Evangelos Angelou, Christina Boumpouka, Dimitrios Tsoumakos, Nectarios Koziris, "*On the Elasticity of NoSQL Databases over Cloud Management Platforms (extended version)*", CIKM Oct 2011, Glasgow UK.

[16] W. Itani, A. Kayssi, A. Chehab, "*Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures*," Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, Dec 2009

[17] M. Jensen, J. Schwenk, N. Gruschka, L.L. Iacono, "*On Technical Security Issues in Cloud Computing*", IEEE International Conference on Cloud Computing, (CLOUD II 2009), Banglore, India, September 2009.

[18] S. Ghemawat, H. Gobioff, and S.-T. Leung, "*The google file system*," in Proceedings of the nineteenth ACM symposium on Operating systems principles".

[19] D.Borthakur. "*The Hadoop Distributed File System: Architecture and Design",* Apache software Foundation, 2007.

[20] Google Inc. Google App Engine. [Online] 2010. [Cited: 07 17, 2010.] http://code.google.com/intl/de-DE/appengine/

[21] Severance, C. *Using Google App Engine.* Sebastopol: O'Reilly Media, 2009.

[22] Daniel J. Abadi, "*Data Management in the Cloud: Limitations and Opportunities*", IEEE 2009.

[23] Jinesh Varia, "*Cloud Architectures",* Amazon Web Services, June 2008.

[24] C.Curino, E.P.Jones, R.A.Popa, N.Malviya, E.Wu, S.Madden, H.Balakrishnan, N.Zeldovich, "*Relational Cloud: A Database-As-A-Service For The Cloud*".

[25] R. Chaiken, B. Jenkins, P.A. Larson, B. Ramsey, D. Shakib, S. Weaver, and J. Zhou, "*Easy and efficient parallel processing of massive data sets*".

[26] Jimmy Lin, "*MapReduce is good enough? If all you have is a hammer, throw away everything That's not a nail!*" arXiv:1209.2191v1 [cs.DC], Sep 2012.

[27] Christos Doulkeridis, Kjetil N., "*A Survey of large scale Analutical Query processing in MapReduce*", VLDB Journal.

[28] K. Lee, Y. Lee, H. Choi, Y. Chung, N. Moon, "*Parallel Data Processing with MapReduce: A Survey",* SIGMOD Record, Dec 2011 (Vol 40 No. 4)

[29] Patrick Valdureiz, "*Parallel database systems: Open Problems and New issues*", Kluwer Academic Publishers, Boston, 1993 pp 137-165.

[30] D.Dewitt, Jim Gray, "Parallel database systems: The future of high performance database systems", Comm of the ACM, June 1992, Vol 35 No. 6.

[31] Shyam Kotecha, "*Platform-as-a-Service*", available online at http://www.ieee.ldrp.ac.in/index.php?option=com_phocadownload&view=category&download=4:pdf&id=1:workshop&Itemid=216

[32] Chun Chen, Gang Chen, Dawei Jiang, Beng Chin Ooi, Hoang Tam Vo, Sai Wu, and Quanqing Xu, " *Providing Scalable Database Services on the Cloud*".

[33] Y. Cao, C. Chen, F. Guo, D. Jiang, Y. Lin, B. C. Ooi, H. T. Vo, S. Wu, and Q. Xu., "*A cloud data storage system for supporting both OLTP and OLAP*", Technical Report, National University of Singapore, School of Computing. TRA8/10, 2010.

[34] Meng-Ju Hsieh, Chao-Rui Chang , Li-Yung Ho , Jan-Jan Wu , Pangfeng Liu ,"*SQLMR : A Scalable Database Management System for Cloud Computing*".

[35] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "*Bigtable : A distributed storage system for structured data,"* ACM Trans Comput. Syst., vol. 26, June 2008.

[36] Xiao Chen, "*Google BigTable*", available online at http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2010-08-2/NET-2010-08-2_06.pdf.

[37] Minpeng Zhu and Tore Risch, "*Querying Combined cloud-based and Relational Databases*" in International Conference on Cloud and service computing 2011.

[38] Herald Kllapi, Dimitris Bilidas, Ian Horrocks, Yannis Ioannidis,Ernesto Jimenez-Ruiz, Evgeny Kharlamov, Manolis Koubarakis, Dmitriy Zheleznyakov, "*Distributed Query Processing on the Clou: the Optique point of View*".

[39] R. Kontchakov,C. Lutz, D. Toman, F. Wolter and M. Zakharyaschev, "*The Combined approach to Ontology based database access*".

[40] Mariano Rodruez-Muro, Roman Kontchakov and Michael Zakharyaschev, "*Ontology based database access: Ontop of databases*" available online at http://www.dcs.bbk.ac.uk/~roman/papers/ISWC13.pdf

[41] D. Campbell, G. Kakivaya and N. Ellis, "*Extreme scale with full SQL language support in Microsoft SQL Azure,*" in SIGMOD, 2010.

[42] Chad DeLoatch and Scott Blindt, "*NoSQL databases: Scalable Cloud and Enterprise Solutions"*, Aug 2012.

[43] Christos Doulkeridis, Kjetil Nervag, " *A Survey of Large-Scale Analytical Query Processing in MapReduce*".

[44] J. Dittrich and J.A. Quian, "*Efficient Big Data processing in Hadoop MapReduce*", Proceedings of the VLDB Endowment.

[45] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff and Raghotham Murthy, "*Hive - A Warehousing Solution Over a Map-Reduce Framework*", VLDB'09, August 24-28, 2009, Lyon, France.
[46] "*Evaluating Apache Cassandra as a Cloud database"*, White Paper by Datastax Corporation, Oct 2013.
[47] Kristóf Kovács, "*Cassandra vs MongoDB vs CouchDBvs Redis vs Riak vs HBase comparison*", available online at http://kkovacs.eu/cassandra vs mongodb vs couchdb vs redis.
[48] http://www.sdss.org
[49] Eve S. McCulloch, "*Harnessing the Power of Big Data in Biological Research*", AIBS Washington Watch, September 2013.
[50] Spotfire Blogging Team, "*10 trends shaping big Data in financial services*", January 2014.
[51] Richard Winter, "*Big Data : Business Opportunities, Requirements and Oracle's Approach"*, December 2011.
[52] Lisa Fleisher, "Big Data Enters the Classroom: Technological Advances and Privacy Concerns Clash".
[53] Darrell M. West, "Big Data for Education: Data Mining, Data Analytics, and Web Dashboards", Governance Studies at Brookings.
[54] Taylor Shelton and Mark Graham, "Geography and the future of Big Data, Big Data and the future of Geography", December 2013.
[55] Joan Serras, Melanie Bosredon, Ricardo Herranz & Michael Batty, "Urban Planning and Big Data – Taking LUTi Models to the Next Level?" Nordregio News Issue 1, 2014
[56] An Executive Report by IBM Institute for Business Value "Analytics: The real world use of Big Data in financial services".
[57] A Deloitte Analytics paper, "*Big Data – Time for a lean approach in fininacial services*", online available at http://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Technology\ 2012_big_data_deloitte_ireland.pdf
[58] A. Abouzeid, K. Bajda-Pawlikowski, D. Abadi, A. Silberschatz and A. Rasin, "*Hadoopdb: an architectural hybrid of mapreduce and dbms technologies for analytical workloads*," Proc. VLDB Endow., vol. 2, August 2009.
[59] N.Samatha1, K.Vijay Chandu, P.Raja Sekhar Redd," *Query Optimization Issues for Data Retrieval in Cloud Computing*".
[60] M. Tamer Oezsu, Patrick Valduriez ``*Principles of Distributed Database Systems, Second Edition''* Prentice Hall, ISBN 0-13-659707-6, 1999
[61] W. Itani, A. Kayssi, A. Chehab, "*Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures*," Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, Dec 2009
[62] L. Haas, D. Kossmann, E.Wimmers, and J. Yang, "*Optimizing queries across diverse data source*," in Proc. VLDB 1997, Athens, Greece.
[63] Edd Dumbill, "*Big Data in the Cloud*", Feb 2011 available online at http://www.o'reilly.com
[64] Vishal Jain, Dr. Mayank Singh, "Ontology Development and Query Retrieval using Protégé Tool", International Journal of Intelligent Systems and Applications (IJISA), Hongkong, Vol. 5, No. 9, August 2013, page no. 67-75, having ISSN No. 2074-9058, DOI: 10.5815/ijisa.2013.09.08 and index with Thomson Reuters (Web of Science), EBSCO, Proquest, DOAJ, Index Copernicus.
[65] Vishal Jain, Dr. Mayank Singh, "Ontology Based Information Retrieval in Semantic Web: A Survey", International Journal of Information Technology and Computer Science (IJITCS), Hongkong, Vol. 5, No. 10, September 2013, page no. 62-69, having ISSN No. 2074-9015, DOI: 10.5815/ijitcs.2013.10.06 and index with Thomson Reuters (Web of Science), EBSCO, Proquest, DOAJ, Index Copernicus.

Authors' Profiles

N.K.Seera is working as an Assistant Professor in Bharati Vidyapeeth College in the department of Computer Applications and Management, New Delhi. She is doing her Ph.D on *Big Data on Cloud*. Her areas of interest are Databases, Query Processing and Cloud Computing. She has presented many papers in National and International conferences.

V.Jain is working as an Assistant Professor in Bharati Vidyapeeth College in the department of Computer Applications and Management, New Delhi. He is doing his Ph.D from Lingaya's University, Faridabad. His areas of interests are Information retrieval, ontology and semantic web. He is a lifetime member of CSI (Computer Society of India) and ISTE (Indian Society of Technical Education).