A Paradigmatic Innovation in Cloud Databases Organizations

Mohd Muntjir
College of Computers and Information Technology
Tail University, Taif, KSA
muntjir.m@gmail.com

***Abstract*—*Relational databases controlled the Information Technology (IT) businesses for almost 40 years. However, last few years have seen the innovations in the way Information Technology is being used and observed. Mostly, Standalone applications have been retrieved with web-based applications, dedicated storage with network storage and dedicated servers with multiple distributed servers. Cloud computing systems have developed into an authenticity due to its scalability, minimum cost, and pay-as-you-go paradigm. Cloud databases such as Sherpa, Big Table, and Simple DB are being prominent. They emphasize the limitations of actual relational databases related to scalability, dynamic appliances and ease of use facility. Cloud databases are basically used for data through applications such as business intelligence, data warehousing and data mining. These applications are scalable, read-intensive, and elastic in nature. Transferable data management applications such as airline reservation, banking, supply chain management applications and online ecommerce are composed and comprehensive. Databases advocating such applications require ACID (Atomicity, Consistency, Isolation and Durability) proprietary, but these databases are hard to deploy in the cloud systems. The main purpose of this paper is to review the requirements of the art in the cloud databases and multiple frameworks. Furthermore, it determines the contest to develop and maintains cloud databases that fulfill the user requirements and argues generally used Cloud databases.*

Keywords: Cloud Database; Cloud computing systems; Database framework.

1. INTRODUCTION

In any Organization, Information Technology (IT) department is compelled for arranging reliable and valuable backup and network facilities, computing, and storage, at thelowest feasible cost. Immense investment in Information Technologyframeworkworks as aninterruption in its approval basically for minimumscale management. Some organization as Cash-strapped organizations look forpreference which can deduct their basic investmentsinvolved in purchasing and maintaining IT hardware andsoftware so that they can get maximum profits from Information Technology.Hence, Cloud computing (CC) becomes a natural and ideal choicefor such organizations, customers and businesses. Cloud computing systemtakes benefit of various technologies such as virtualization, grid computing, serverconsolidation, big and faster storage robust networks and N-tier architectures. Itsupports IT-related services such as Software-as-a-Service (SaaS),progress Platforms-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) over the network on-demand anytime fromanywhere on the basis of “pay-as-you-go" miniature. It deducts the cost of using costlyresources at the provider’s end due to economies of scalability.Fast provisioning and extant deployment of modern applications at minimal cost are the profits, whichprovoke people to accept. Simultaneously, with usage of Cloud computing system, availability forprovisioning of database services has increased. Applianceof Cloud databases is known as Database-as-a-Service inCloud computing terminology. The importantgoal of the paper is to extend the behaviors in Cloud databases and emphasize thepotential challenges to develop and design these databases. Thearchitecture of paper has been divided into six different sections. Second section represents Cloud database systems. Third section emphasizes an overview of similar types of databases systems. Section 4reviews major challenges to develop cloud-computing databases. Furthermore, fifth represents current cloud databases succeed byconclusions.

2. CLOUD DATABASES

Immenseevolution in digital data, dynamic data storage needs, best broadband proficiency and Cloud computing manages to the development of cloud database systems [1].In the Cloud terminology;Data as a service (DaaS), Cloud Storage, and Database as a service (DBaaS) are the different terms used for dataexecutives in the Cloud Computing. They contrast on the basis of howdata is stored, managed and executed. Cloud storage is virtualstorage that permits users to store objects and documents, iCloud, Dropbox etc. are familiar cloud storage services[2]. Furthermore DaaSpermits users to store data at a remote diskaccessible through Network. The basic use of DaaSis used for backup purposes and basic data management systems. DBaaS offers whole database management and permits users to access and store their database at remotedisks anytime from any place through Network. Google’s Big Table, Amazon RDS, Amazon’sSimple DB, Microsoft’s SQL Azure and Yahoo’sSherpa Database are thecommonly used databases in the Cloud Computing [3]. In Cloud Computing systems, Cloud database is a database expressed to users on demandthrough the Internet from a server provided by cloud database providers. Cloud databases support scalability and high availability, multi-tenancy and optimized resource allocation. A cloud database can be a conventional databasesuch as SQL Server and MySQL. These traditional databases can beinstalled, configured and maintained by the user himself on a Cloud server. This preference is generally called the “Do-it-Yourself”

(DIY)accession. Some Cloud service providersprovideprefabricateddatabase services such as Xeround's MySQL [4]. In "Do-it-Yourself" (DIY) approach, the developers manually assure reliability and flexibilityservices. Election of a DBaaS solution decreases the complexity and cost ofrunning one's individual database in Cloud. Cloud databases contribute improved scalability, availability, performance and flexibility at lower prices. In a Cloud Computing,Cloud databases are able to support changing storageneeds of Internet-savvy users who deal more withuser created content, unstructured data, such as photos and documents. Shared-disk andShared-nothing are twovitalized storage frameworks in database systems.

A. Shared-disk Database Architecture

Shared-disk Database Architecture treats the aggregate database as a single big piece of database stored on a Network Attached Storage (NAS) or Storage Area Network (SAN) storage that is shared and usable through network by all nodes in a database. It needs some low-cost servers. Basically, it is easy to visualizethem as each compute server is unique. It isolates the compute from the storage as any number of compute instances may work on the entire data in a database. Middleware is not needed to direct data requests to specific servers as each node/client has approach to all of the data. Henceforth, it is more convenient for On-Line Transaction processing operations. Oracle RAC, Sybase and IBMDB2 pureScale, etc. support this framework [8].

B. Shared-nothing Storage Architecture

Shared-nothing Storage architecture associates datapartitioning which breaches the data into independent sets or autonomousposture. These data sets are actually located on various databaseservers. Furthermore, each server processes and manages its small part ofthe database particularlywhich makes shared-nothingdatabases easily scalable and manageable. Whereasbasic scalability and applications designed to work on shared-nothing storageframework are convenient for Cloud database. Although data partitioningused in this architecture does not work well with cloud system. Itis not easy to visualize a shared-nothing database becomes very difficult and complex to manage due todata partitioning. So that, it needs a piece of middleware to routedatabase requests to the desired server. Due to adding moreservers, data has to be repartitioned. In this architecture, datapartitioning should be done very carefully, neither data neither shipping norattaching will becomedifficult and complex. More data shipping means more inactivity andnetwork bandwidth congestion. These problems reducedatabase performance carelessly. Shared-nothing Storage framework is also used basically for data-intensive workloads. Amazon's SimpleDB, Yahoo's PNUTS, and Hadoop Distributed File System implement shared nothing architecture [5-7]. Oracle and IBM released their shared-nothing implementation of DB2 in September 2008 and 1990 properly for scalable analytical applications of data warehouses.

3. A PARELLEL STUDY OF RELATIONAL DATABASES AND NOSQLDATABASES

In the earlier phase of industrialization, there were more requirements for transaction processing applications. As well as thedatabase industry matured and people take computersas part and parcel of their lives, systematic applicationsbecame the focus of businesses. Now the enterprises need to store data not only for transaction processing, although to analyzeconsumer behaviors and business requirements. Now, Enterprises want touse analytical knowledge to increase their business costs.So, business applications are widely differentiated intotransactional and analytical applications. Relationaldatabases played imperative role in controlling transactionaldata. Consequently, industryleaders like IBM and Oracle joined analytical abilities to their relational databases for data mining applications. Furthermore, number ofdatabases such as Object-oriented databases, Column databases, etc. came into market [9-10]. Still they couldnot reduce the relational databases. In turn, then Internet innovation and web 2.0 applications started developing huge sparse and unstructured data. Relational Database Management Systems (RDBMS) are not applicable for controlling massive sparse data sets with relatively defined schemas. The requirement to store and process such bigdata emphasize the role of NoSQL databases in the databasetechnology as Cloud computing. RDBMs and NOSQLdatabases are concisely discussed as given below:

A. NoSQL Databases

NoSQL database means 'Not Only SQLDatabase 'or 'Not Relational Database'. ANoSQL database is defined as a non-relational,horizontally scalable database, without ACIDguarantees, and shared nothing. NoSQL applications are defined furtherinto document stores, key/value stores, tuple stores,column stores and graph stores and object stores. They can store and retrieve structured, semi-structured and unstructured data.NoSQL Databases are item-oriented. Here, A domain can be related to astable and contains items having various schemas. In this database, the items are distinguished by keys. This technique recovers scalability of these databases as complex joins are not needed to regroup data from different tables in that schema. They are generally supplied on demand. They have developed to address the basic requirements of datamanagement in the cloud as they supervene BASE (BasicallyAvailable, Soft state, eventually consistent) in comparison tothe ACID assurance. Hence, they are not perfect for update extensivetransaction appliance. They contribute high possibility at the cost of flexibility [13-14].

B. Relational Databases

The approach of relational databases is about forty years old. Relational Database worked best in the period of hardware limits such as low processor speed and limited networking, little memory and small disk space. It has determined database framework based on relationships and schema, tables, indexes and columns. In a Relational Database, Data is stored in tables with predefined complex relationships. Furthermore, Column indexes are used for rapid search. For this purpose, highly skilled Developers and DBAs are needed for database design maintenance and maintenance. Traditionally, they are used for transactional databases, and they integrate details at the lowest granularity and management. They incorporate sensitive and operational data such as credit card numbers to handle critical business operations, and employee data. These types of databases are not well appropriate for Cloud architecture and atmosphere as they do not support full satisfied data search and are more difficult to scale apart from limits [11-12].

4. Provocation To Stablish Cloud Databases

Cloud Database Management Systems should support features of Cloudcomputing as well as of conventional databases for immense approval, which is a mammoth task. The conceivable challenges integrated with cloud databases are as follows:

A. Scalability

The major characteristics of Cloud paradigm is scalability which suggests that resources can be scaled-up or scaled-downdynamically without causing any intrusion in theservices. It establishes challenges on developers to develop and design databases in such a way that they can support and control thousands numbers of concurrent users and data enhancements. Businesses deal with vast volumes of data. Furthermore, adding additional servers on demand solve the problem ofscalability, one and only if the process and workload are simultaneously parallel. Scalability need of transactional datais minimal in contrast to analytical data.

B. High availability and Fault Tolerance

Availability of database defines that database is up andrunning on the scale of 365 X 24 X 7.Further, it becomes mandatory to imitate data across large geographic distances to give durability, high data availability, and Amazon's S3 cloud storage service replicates data beyond "regions" and "availability segments".

C. Heterogeneous Environment

Users want to access distinct applications from differentlocations and devices such as computers, notepads,tablets, and mobiles. Since user applications and data diversify in quality, it becomes more difficult to pre-design how users will use the systems.

D. Data Consistency and Integrity

Data integrity is the most analytical needs of all enterprise applications and is maintained and managed through databaseconstraints. Furthermore, the lack of data integrity results in unpredictable results. Cloud databases follow BASE (Basically Available, Soft state, eventually consistent) in compare tothe ACID (Atomicity, Consistency, Isolation and Durability) guarantees. Hence, Cloud databases support dependent consistency due to replication of data at multiple shared positions. On the contrary of developers' side, developers need to follow BASE paradigm carefully.They should not adjustment data integrity in their over intensity to turn to cloud database systems.

E. Simplified Query Interface

Cloud Database is distributed and shared database. Querying shared database is a big challenge that cloud developers face in Cloud systems. Adistributed query has to approach different nodes of clouddatabase systems. There should be interpreted and standardizedquery interface for querying the database in Cloud systems.

F. Database Security and Privacy

Data physically stored in a specific country, is to follow local rules and regulations of that country in a specific manners. The US PatriotAct grants the government to demand access to the data stored and managed on any computer E.g. Amazon S3 only permits acustomer to select between US and EU data storage selections. Furthermore, if data is encrypted using a key not situated at thehost, and then it is little protected. Uncertainty is involved in storingtransactional data on an authorized host. In Cloud database, sensitive data isencrypted before being uploaded to the cloud to secure untrusted approach. Each and every application running in the cloudshould not have the capability to directly decrypt the databefore achieving it. In a Cloud database, providing safety, security and privacy to various databases on the same hardware is also a big defiance.

G. Data Portability and Interoperability

Vendor lock-in is a key obstruction in the approval of cloud database systems. Users need the liberty to move from one vendorto another without any obstacles. It can be prevented through convenient and interoperable segments. Data Portability isthe capability to run components written for one cloudprovider in another cloud provider's environment and framework. Furthermore, interoperability is the ability to write a piece of code that is more flexible to work with different cloud providers, unconcerned of the differences between them. Basically,there are no standard API to store and access clouddatabases in a cloud systems. Traditional applications should be able to work with cloud systems. Furthermore, Cloud databases should also be able to communicate with business intelligence tools already accessible in the business spots [15-16].

5. EnterpriseProceeding To Cloud Databases

Cloud databases are designed and created for low-cost assets hardware. They scale out conveniently by sharing thedatabase across various hosts/nodes as the load accession. NoSQL databases have become analogue for cloud databases. Some basically used cloud databases in the corporations are defined below.

A. Amazon Simple Storage Service (S3) andDatabases

Amazon S3 is network based storage service. S3 stores objects or items up to 5GB in size along with 2 KB of Meta data for each item. Items are controlled by buckets and eachbucket is owned by an AWS (Amazon Web Services) account. These buckets are recognized by an exclusive, user assigned key. These Buckets and Items are designed, created, listed andretrieved using either a REST or SOAP interface in the network. Amazonoffers Oracle, MySQL, and Microsoft SQL Server virtualinstances of databases for deployment and management in its AmazonElastic Compute Cloud (EC2) cloud systems. Although third party management providers like right scale and Elastra offer MySQL images virtually. Scaling is not so easy with MySQL but itcan be scaling. Enterprise DB's Postgres Plus Advanced Server, a transactional database also works in Amazon'scloud systems. Previous Storage was fixed to the EC2 instance in Amazon. If some instance is terminated than its mean loss of data associated with that instance in EC2. With Amazon's Elastic Block Store (EBS); a user can choose to allot storage volumes that obtain reliably and automatically from EC2 instances in Amazon database. AmazonRelational Database Service (RDS) is also a web servicethat makes it easy to arrange and measure a relational databasein the Cloud database. It allows accessing the capabilities of SQL Server database engines, Oracle or MySQL,running and working on Amazon RDS database instance [17-18].

B. Amazon SimpleDB

It is scalable, flexible and a highly available non-relationaldata store. It functions closely with Amazon S3 and AmazonEC2 to afford the ability to process, store, and query datasets in the cloud system. Amazon Simple DB is NoSQL and name or value pair datastore. It proposed a simple interface ofPost, Get, Delete andQuery to run queries on structured data in a database. It is composed of domains, items, values and attributes. A domain is equivalent to a table or a worksheet in a spreadsheet e.g. customers table. Dissimilar a spreadsheet, it grants cells to contain variousvalues per entry. Further, each and every item can have its own different or unique set of correlated attributes(For example, item "1" may have attributes "Employee no" and "Employee name" whereas item "2" may have attributes "Employee no", "Employee name" and "Salary". It allots scalability byaccessing user to share and partition the workload across various domains. A user can elect between consistency and eventual consistency. It gives permissions to users to select and encrypt data before saving it. It does not decode the data basically but query directly on the strings stored. It spontaneously manages and maintains indexing of data, redundancy and performance tuning [19].

C. GoogleApp's Bigtable

It is a distributed or shared storage system based on GFS (GoogleFile system) for structured data. It appliance replicated shared-nothing databases. It has been clearly expanded in many Google products like Google application engine. It gives permission a more complex data store rather than SimpleDB. It allows entities and properties in compared to columns and tables. The GoogleData store API also admits a get, put, delete format for obtaining data. It also permits a non-SQL languagecalled GQL, which is not as constituent rich as SQL (Structured Query Language). In GQL, Select statements can be oppressed on one table only. Furthermore, GQL does not support the "Join" condition [20-21].

D. MapReduce

It is more easy-to-use programming model that approves parallel architecture. MapReduce is more scalable and works in adistributed or shared method. It is more useful for huge data processing,data analysis and large-scale search in the cloud database. It provides a consideration by defining a "mapper" and a "reducer". The term "mapper" is applied to every input key/value pair to accomplish an arbitrary number ofintermediate key/value pairs. The term "reducer" is applied toall values integrated with the same intermediate key to initiate output key/value pairs. It can separate the input data and schedule the execution of program across a set ofmachines, Furthermore, it can handle machine failures and maintain the intermachine broadcasting. But it cannot be related to database management systems [22].

E. Hadoop

It is a programming architecture for implementingMapReduce across huge grid of servers. Hadoop is distributed and shared innature and has better scalability rather thancolumn store and relational databases. It is more convenient for unstructured data. Itis not for complex data structures, multitasking and mixed workloads. It is a Java based open source project.With the help and support from Yahoo, Hadoop has attained big development. Hadoop accredits the addition of Java software Componentsand provides HDFS (Hadoop Distributed File System) andhas been expanded to include HBase, i.e. column store database [23].

F. Windows Azure Cloud Storage

The main objective of Windows Azure Storage is to let users andapplications access their data conveniently from anywhere atany time using simple and common programming API (Application Program Interface). Users can use scalable storage to store any piece of datafor any length of time on pay per use support. It supports NoSQL databases and queues, unstructured as well as structured data. It administers three data abstractions: Tables,Queues and Blobs. The term "Table" is a set of entities,which comprise a set of premises. Tables provide structured storage. Furthermore, the term "Queues" provides convenient storage and delivery of messages for one or more applications.Blobs arrange a simple interface for storing named files along with metadata for the files. Allinformation lies in Windows Azure storage is replicatedthree times which concedes fault tolerance [24].

G. Microsoft SQL Server Data Services (SDDS)

SDDS is a key/value data store, which is also considers the cloudextension of Microsoft's SQL Server. It associates withMicrosoft's Sync Framework, which is a .NET library forsynchronizing unique data source. It consists SOAP or REST APIs, schemafreedata storage, and a pay-as-yougopayment systems. SDDS has three core concepts: Entity,Authority and Container. The term Entity is a property bag of name and value pairs, and the term Container is a collection of entities. Further, Authority is a collection of containers and acts as a billing unit [25].

H. Sherpa

It was basically known as PNUTS in previous publications. In Sherpa, Data is classified into tables of records with its attributes. Here, tables can be ordered or smashed. Sherpa supports blob data typealong with typical data types. It is also considered as an interpreted relationaldata model. It also supports projection and selection from asingle table and prevents join operations. In this database, data is replicated nonparallel. It can control in high consistency or highavailabilitymode. In turn, Hadoopcan use Sherpa as a data store instead of the native HDFS (Hadoop Distributed File Systems) [26].

I. Dynamo

It is highly available,distributed key-value data-store, scalable and used by Amazon's core services. Dynamo uses inevitable consistency to achieve high level of availability. Its means it can write anywhere and update will consequently generate to all replicas asynchronously. In Dynamo, there is no indexes or no record structures. It only allows single key updates with features. It makes thorough use of objectversioning and application-assisted contest resolution [27].

J. MegaStore

It composites the scalability of a NoSQL data-store and the accessibility of a conventional RDBMS to meet the storage needs of common Internet services such as e-mail,social networking and documents. It uses concurrent replication to find high availability and a permanent view of the given data. It supports transactional (ACID)guarantees within an entity group. MegaStore is a convenient datamodel with schema, defined by users, queues and full-text indexes [28].

K. CouchDB

CouchDB is anopen-source,free, Apache project sinceearly 2008.CouchDB is a document-oriented database, basically written in Erlang. Further, it belongs to NoSQL generation of database. Documents (records) are stored in JSON (JavaScriptObject Notation) format and are controlled through anHTTP interface. It also allows "views" to be dynamicallycreated using JavaScript. These created views map the documentdata onto a table-like structure that can be queried and indexed. CouchDB does not support a non-procedural querylanguage. It accomplishes scalability through asynchronousreplications. It has unique ability to serve as a self-containedapplication server and databases [29].

L. MongoDB

MongoDB is a GPL (General Public License) open sourcedocument-oriented JSON database system beingdeveloped and managed at 10gen by Geir Magnusson and Dwight Merriman. MongoDB is developed to be a true object oriented database rather than a pure key/value store. It stores data in JSONlikedocuments with actual schemas. MongoDB allots thescalability and speed of key-value stores and rich performance like indexes and dynamic queries of relational database. It also provides horizontal scalability [30].

Although NoSQL databases are broadly considered as clouddatabases in the database landscape, but they are not a solutionfor all problems. They can work with large sparse data,but do not provide flexible indexing,transactional integrity,querying and SQL. Hence, Clouddatabases should be used with full recognition of theirlimitations.

6. Conclusion

Huge data generated by web-based applications have changed the whole database aspects. Cloud databases appear to be a good solution for controlling such data. Furthermore, all enterprises cannot afford to arrange more expensive data center framework for managing and maintain their own databases. The emerging popularity of Cloud databases is making the beginning of new era of database systems. Although cloud databases are not ACID flexible, they are able tocontrol huge workloads of web-based databases, which do not need such guarantees. Various Cloud databases are available in the enterprises. They share and distribute similar approach and characteristics such as simple API, schema free databases, scalability, eventual/timeline flexibility, synchronous/asynchronous replication etc. But each has its particular and unique API, data model and database functions, query interface. This approach required to be standardized for their better enhancements. Cloud computing and Cloud databases are set to rule the next decade by reducing the limitations they have. Furthermore, Asian nations are investing in cloud centers of economic development and are acquiring cloud services for their own businesses. They are very focused on the cloud circumstances with desired opportunity.

References

[1] Rajkumar,Buyya et al., "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, Vol. 25, Issue 6, June 2009, pp. 599-616.

[2] Jiyi Wu et al, "Recent Advances in Cloud Storage", in Third International Symposium on Computer Science and Computational Technology(ISCSCT '10), Jiaozuo, P. R. China, 14-15, August 2010, pp. 151-154.

[3] Database as a Service: Reference Architecture – An Overview, an Oracle White Paper on Enterprise Architecture September 2011 http://www.oracle.com/technetwork/topics/entarch/oesrefarch-dbaas-508111.pdf last accessed on May 28, 2012.

[4] http://xeround.com last accessed on May 25, 2012.

[5] Daniel J. Abadi, "Data Management in the Cloud: Limitations and Opportunities", Bulletin of the IEEEComputer Society Technical Committee on Data Engineering, 2009, 32(1):3-12.

[6] http://aws.amazon.com/simpledb/ last accessed on May 23,2012.

[7] B.F. Cooper et al., "PNUTS: Yahoo!'s Hosted Data ServingPlatform", in International Conference on Very Large Databases (VLDB), Vol. 1, no. 2, 2008, pp. 1277–1288.

[8] Donald Kossmann, Tim Kraska, Simon Loesing, "An Evaluation of Alternative Architectures for Transaction Processing in the Cloud", SIGMOD'10, June 2010.

[9] Daniel J. Abadi et al., "Column-oriented Database Systems", VLDB '09.

[10] Stonebraker, et al., "C-Store: A Column-oriented DBMS".

[11] Thakur Ramjiram Singh, "Cloud Computing: An Analysis", International Journal of Enterprise Computing and BusinessSystems", Vol. 1, issue 2, July 2011, pp. 2230-8849.

[12] Rick Cattell, "Scalable SQL and NoSQL Data Stores", ACM SIGMOD, Vol. 39, Issue 4, 2011, pp. 12-27.

[13] ArpitaMathur et al., "Cloud Based Distributed Databases: The Future Ahead", International Journal on Computer Science and Engineering (IJCSE) Vol. 3, No. 6, 2011.

[14] Bo Peng, "Implementation Issues of A Cloud Computing Platform", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering.

[15] Mihaela Ion, "Enforcing multi-user access policies to encrypted cloud databases", IEEE International Symposium on Policies for Distributed Systems and Networks, 2011, pp.175-177.

[16] Maggiani, R. "Cloud computing is changing how we communicate", IPCC 2009, 2009, pp. 1-4.

[17] http://aws.amazon.com/rds/S3 last accessed on May 24, 2012.

[18] http://aws.amazon.com/rds/ last accessed on May 24, 2012.

[19] http://aws.amazon.com/simpledb last accessed on May 25, 2012.

[20] S. Ghemawat et al., "The Google File System", inproceeding of 19th ACM Symp. Operating System Principles (SOSP 03), ACM Press, 2003, pp. 29–43.

[21] F. Chang et al., "Bigtable: A Distributed Storage System for Structured Data", in 7th UsenixSymp. Operating Systems Design and Implementation (OSDI 06), Usenix Assoc., 2006, pp. 205–218.

[22] Dawei Jiang et al., "MAP-JOIN-REDUCE: Toward Scalable and Efficient Data Analysis on Large Clusters",IEEE Transactions on Knowledge and Data Engineering,Vol. 23, No. 9, 2011.

[23] D. Borthakur, "The Hadoop Distributed File System: Architecture and Design, Apache Software Foundation", http://hadoop.apache.org/core/docs/r0.16.4/hdfs_design.html last accessed on May 27, 2012.

[24] Troy Davis, "Cloud Computing Use Cases and Considerations", http://digissance.com/ Cloud Computing Talk.pdf last accessed on June 10, 2012.

[25] www.windowsazure.com/enus/develop/net/.../c loud storage/last accessed on June 10, 2012.

[26] Brian Cooper et al., "Building a Cloud for Yahoo", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2009.

[27] Giuseppe De Candia et al., "Dynamo: Amazon's Highly Available Key-value Store", in of 21st ACM Symposium onOperating System Principles, SOSP 2007, pp 205-220.

[28] Jason Baker et al., "Megastore: Providing Scalable, Highly Available Storage for Interactive Services", in 5th Biennial Conference on Innovative Data Systems Research (CIDR '11), 2011, pp.223-234.

[29] http://www.couchbase.com/couchdb last accessed onMay31, 2012.

[30] http://www.mongodb.org last accessed on May31, 2012.