

# Cloud Databases: A Paradigm Shift in Databases

Indu Arora<sup>1</sup> and Dr. Anu Gupta<sup>2</sup>

<sup>1</sup> Department of Computer Science and Application,  
MCM DAV College for Women, Chandigarh

<sup>2</sup> Department of Computer Science and Application,  
Panjab University, Chandigarh

## Abstract

Relational databases ruled the Information Technology (IT) industry for almost 40 years. But last few years have seen sea changes in the way IT is being used and viewed. Stand alone applications have been replaced with web-based applications, dedicated servers with multiple distributed servers and dedicated storage with network storage. Cloud computing has become a reality due to its lesser cost, scalability and pay-as-you-go model. It is one of the biggest changes in IT after the rise of World Wide Web. Cloud databases such as Big Table, Sherpa and SimpleDB are becoming popular. They address the limitations of existing relational databases related to scalability, ease of use and dynamic provisioning. Cloud databases are mainly used for data-intensive applications such as data warehousing, data mining and business intelligence. These applications are read-intensive, scalable and elastic in nature. Transactional data management applications such as banking, airline reservation, online e-commerce and supply chain management applications are write-intensive. Databases supporting such applications require ACID (Atomicity, Consistency, Isolation and Durability) properties, but these databases are difficult to deploy in the cloud. The goal of this paper is to review the state of the art in the cloud databases and various architectures. It further assesses the challenges to develop cloud databases that meet the user requirements and discusses popularly used Cloud databases.

**Keywords:** *Cloud computing; Cloud Databases; Database Architectures.*

## 1. Introduction

Information Technology (IT) department of any organization is responsible for providing reliable computing, storage, backup and network facilities at the lowest feasible cost. Huge investment in IT infrastructure works as a hindrance in its adoption especially for small scale organizations. Cash-strapped organizations look for alternatives which can reduce their capital investments involved in purchasing and maintaining IT hardware and software so that they can get maximum benefits of IT. Cloud computing (CC) becomes a natural and ideal choice for such organizations and customers. Cloud computing takes benefit of many technologies such as server consolidation, huge and faster storage, grid computing,

virtualization, N-tier architecture and robust networks. It delivers highly scalable and expensive infrastructure with minimal set up and negligible maintenance cost. It provides IT-related services such as Software-as-a-Service, Development Platforms-as-a-Service and Infrastructure-as-a-Service over the network on-demand anytime from anywhere on the basis of "pay-as-you-go" model. It is a fast growing concept changing the IT related perceptions of its users. Elasticity, scalability, high availability, price-per-usage and multi-tenancy are the main features of Cloud computing. It reduces the cost of using expensive resources at the provider's end due to economies of scale. Quick provisioning and immediate deployment of latest applications at lesser cost are the benefits which force people to adopt Cloud computing.

Cloud computing has brought a paradigm shift not in the technology landscape, but also in the database landscape. With more usage of Cloud computing, demand for provisioning of database services has raised. Provisioning of Cloud databases is known as Database-as-a-Service in Cloud terminology. The main objective of the paper is to explore the trends in Cloud databases and analyze the potential challenges to develop these databases. The structure of paper has been divided into six sections. Second section describes Cloud databases. Third provides an overview of common types of databases. Section 4 discusses major challenges to develop cloud databases. Fifth summarizes existing cloud databases followed by conclusions.

## 2. Cloud Databases

Massive growth in digital data, changing data storage requirements, better broadband facilities and Cloud computing led to the emergence of cloud databases [1]. Cloud Storage, Data as a service (DaaS) and Database as a service (DBaaS) are the different terms used for data management in the Cloud. They differ on the basis of how data is stored and managed. Cloud storage is virtual storage that enables users to store documents and objects.

Dropbox, iCloud etc. are popular cloud storage services [2]. DaaS allows user to store data at a remote disk available through Internet. It is used mainly for backup purposes and basic data management. Cloud storage cannot work without basic data management services. So, these two terms are used interchangeably. DBaaS is one-step ahead. It offers complete database functionality and allows users to access and store their database at remote disks anytime from any place through Internet. Amazon's SimpleDB, Amazon RDS, Google's BigTable, Yahoo's Sherpa and Microsoft's SQL Azure Database are the commonly used databases in the Cloud [3].

Cloud database is a database delivered to users on demand through the Internet from a cloud database provider's servers. Cloud databases provide scalability, high availability, optimized resource allocation and multi-tenancy. A cloud database can be a traditional database such as MySQL and SQL Server. These databases can be installed, configured and maintained on a Cloud server by the user himself. This option is popularly called the "Do-it-Yourself" approach (DIY). Few providers offer ready-made database services such as Xeround's MySQL [4]. In "Do-it-Yourself" approach, the developers manually ensure reliability and elasticity service. Selection of a DBaaS solution reduces the complexity and cost of running one's own database. It spares the developer from the hassles of tedious management tasks of the database. Cloud databases provide improved availability, scalability, performance and flexibility at lesser price. Conventional DBMS (Data Base Management System) deals with structured data which is held in databases along with its metadata. While Cloud databases can be used for unstructured, semi-structured data or structured data. Data stored in files of various types where the metadata was either unavailable or incomplete is called unstructured data. Cloud databases are able to support changing storage requirements of Internet-savvy users who deal more with unstructured data, user created content such as documents and photos. Shared-nothing and shared-disk are two widely-used storage architectures in database systems.

### 2.1 Shared-nothing Storage Architecture

Shared-nothing Storage architecture involves data partitioning which splits the data into independent sets. These data sets are physically located on different database servers. Each server processes and maintains its piece of the database exclusively which makes shared-nothing databases easily scalable. Due to inherent scalability, applications designed to work on shared-nothing storage architecture are suitable for Cloud. But data partitioning used in this architecture does not work well with cloud. It is very difficult to virtualize a shared-nothing database as it becomes very complex and difficult to maintain due to

data partitioning. It needs a piece of middleware to route database requests to the appropriate server. As more servers are added, data has to be repartitioned. Data partitioning should be done very carefully, otherwise data shipping (passing of the information from one machine to the other machine for processing) and joining will become difficult. More data shipping means more latency and network bandwidth bottlenecks. These issues reduce database performance badly. Shared-nothing Storage architecture is also used mainly for data-intensive workloads. IBM and Oracle released their shared-nothing implementation of DB2 in 1990 and September 2008 respectively for scalable analytical applications of data warehouses. Amazon's SimpleDB, Hadoop Distributed File System and Yahoo's PNUTS also implement shared-nothing architecture [5-7].

### 2.2 Shared-disk Database Architecture

Shared-disk Database Architecture treats the whole database as a single large piece of database stored on a Storage Area Network (SAN) or Network Attached Storage (NAS) storage that is shared and accessible through network by all nodes. It requires fewer low-cost servers. It is easy to virtualize them as each compute server is identical. It separates the compute from the storage as any number of compute instances may work on the entire data. Middleware is not required to route data requests to specific servers as each node/client has access to all of the data. Hence, it is more suitable for On-Line Transaction Processing applications. Oracle RAC, IBM DB2 pureScale, Sybase etc. support this architecture [11].

Table 1: Comparison of shared-nothing and shared disk storage architectures

<i>Architecture</i>	<i>Partitioning</i>	<i>Distributed</i>	<i>Scalability</i>	<i>ACID</i>	<i>OLTP</i>	<i>Analytical</i>	<i>Maintenance Cost</i>	<i>Useful for Cloud</i>
Shared-Nothing	Y	Y	Y	N	N	Y	High	Y
Shared-Disk	N	Y	Y	Y	Y	Y	Low	Y

Note: N-No, Y- Yes

## 3. A Comparative Study of Relational Databases and NoSQL Databases

In the earlier stages of computerization, there was more demand for transaction processing applications. As the database industry matured and people accepted computers as part and parcel of their lives, analytical applications became the focus of enterprises. Now they wanted to store

data not only for transaction processing, but to analyze consumer trends and business needs. Enterprises want to use analytical knowledge to enhance their business value. So, enterprise applications are broadly categorized into transactional and analytical applications. Relational databases played dominant role in handling transactional data. Later on, industry leaders like IBM and Oracle added analytical capabilities to their relational databases for data mining applications. In the mean time, number of databases such as Column databases, Object-oriented databases etc. came into market [12-13]. But they could not overpower the relational databases. Then Internet revolution and web 2.0 applications started producing massive sparse and unstructured data. RDBMS are not suitable for handling massive sparse data sets with loosely defined schemas. The need to store and process such big data defined the role of NoSQL databases in the database technology as Cloud databases. RDBMS and NOSQL databases are briefly discussed as follows:

### 3.1 Relational Databases

The concept of relational databases is forty years old. It worked best in the era of hardware limits such as small disk space, little memory, slow processor speed and limited networking. It has rigid database architecture based on tables, columns, indexes, relationships and schema. Data is stored in tables with predefined complex relationships. Column indexes are used for faster search. Highly skilled Developers and DBAs are required for database design and maintenance. Conventionally, they are used for transactional databases. They include details at the lowest granularity. They contain sensitive and operational data such as employee data and credit card numbers to handle critical business operations. These databases are not well suited for Cloud environment as they do not support full content data search and are difficult to scale beyond a limit [14-15].

### 3.2 NoSQL databases

NoSQL means ‘Not Only SQL’ or ‘Not Relational’. A NoSQL database is defined as a non-relational, shared-nothing, horizontally scalable database without ACID guarantees. NoSQL implementations are classified further into key/value stores, document stores, object stores, tuple stores, column stores and graph stores. They can store and retrieve unstructured, semi-structured and structured data. They are item-oriented. A domain can be compared to a table and contains items having different schemas. The items are identified by keys. All data relevant to a particular item is stored within that item. It improves scalability of these databases as complex joins are not required to regroup data from multiple tables. They have the ability to replicate and distribute data over many servers. They are dynamically provisioned on demand.

They have emerged to address the requirements of data management in the cloud as they follow BASE (Basically Available, Soft state, eventually consistent) in contrast to the ACID guarantees. So, they are not suitable for update-intensive transaction applications. They provide high availability at the cost of consistency [16-17].

Table 2: Comparison of RDBMS and NoSQL databases

<b>RDBMS</b>	<b>NoSQL Databases</b>
Data within a database is treated as a “whole”	Each entity is considered an independent unit of data and can be freely moved from one machine to the other
RDBMS support centrally managed architecture.	They follow distributed architecture.
They are statically provisioned.	They are dynamically provisioned.
It is difficult to scale them.	They are easily scalable.
They provide SQL to query data	They use API to query data (not feature rich as SQL).
ACID (Atomicity, Consistency, Isolation and Durability) Compliant; DBMS maintains Consistency.	Follow BASE (Basically Available, Soft state, Eventually consistent); The user accesses are guaranteed only at a single-key level.
They support on-line Transaction Processing applications.	They support web2.0 applications.
ORACLE, MySQL, SQL Server etc. are popular RDBMS.	Amazon SimpleDB, Yahoo’s PNUTS, CouchDB etc. are popular NoSQL Databases.

## 4. Challenges to Develop Cloud Databases

Cloud DBMSs should support features of Cloud computing as well as of traditional databases for wider acceptability, which is a Hercules’s task. The potential challenges associated with cloud databases are as follows:

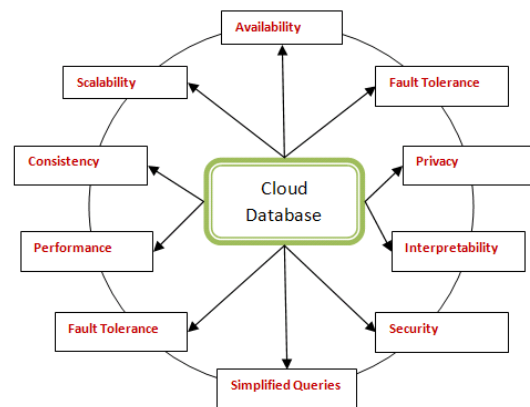


Fig. 1. Possible issues in makeup of cloud databases.

#### 4.1 Scalability

The main feature of Cloud paradigm is scalability which implies that resources can be scaled-up or scaled-down dynamically without causing any interruption in the service. It puts challenges on developers to develop databases in such a way that they can support and handle unlimited number of concurrent users and data growth. Enterprises deal with huge volumes of data. Adding additional servers on demand solve the problem of scalability, only if the process and workload are parallelizable. Scalability requirement of transactional data is lesser in comparison to analytical data.

#### 4.2 High availability and Fault Tolerance

Availability of database implies that database is up and running 365 X 24 X 7. It becomes necessary to replicate data across large geographic distances to provide high data availability, durability and high levels of fault tolerance. Amazon's S3 cloud storage service replicates data across "regions" and "availability zones".

#### 4.3 Heterogeneous Environment

Users want to access diverse applications from different locations and devices such as mobiles, tablets, notepads and computers. Since user applications and data (structured or unstructured) vary in nature, it becomes difficult to predefine how users will use the system.

#### 4.4 Data Consistency and Integrity

Data integrity is the most critical requirement of all business applications and is maintained through database constraints. The lack of data integrity results in unexpected outputs. Cloud databases follow BASE (Basically Available, Soft state, Eventually consistent) in contrast to the ACID (Atomicity, Consistency, Isolation and Durability) guarantees. So, Cloud databases support eventual consistency due to replication of data at multiple distributed locations. It becomes difficult to maintain the consistency of a transaction in a database which changes too quickly especially in the case of transactional data. Developers need to follow BASE approach cautiously. They should not compromise data integrity in their over enthusiasm to move to cloud databases.

#### 4.5 Simplified Query Interface

Cloud Database is distributed. Querying distributed database is a major challenge that cloud developers face. A distributed query has to access multiple nodes of cloud database. There should be a simplified and standardized query interface for querying the database.

#### 4.6 Database Security and Privacy

Data physically stored in a particular country, is subject to local rules and regulations of that country. The US Patriot Act allows the government to demand access to the data stored on any computer. Amazon S3 only allows a customer to choose between US and EU data storage options. If data is encrypted using a key not located at the host, then it is little safer. Risks are involved in storing transactional data on an untrusted host. Sensitive data is encrypted before being uploaded to the cloud to prevent unauthorized access. Any application running in the cloud should not have the ability to directly decrypt the data before accessing it. Providing security and privacy to different databases on the same hardware is also a big challenge.

#### 4.7 Data Portability and Interoperability

Vendor lock-in is a key obstacle in the adoption of cloud databases. Users want the liberty to move from one vendor to another without any hassles. It can be avoided through portable and interoperable components. Data Portability is the ability to run components written for one cloud provider in another cloud provider's environment. Interoperability is the ability to write a piece of code that is flexible enough to work with multiple cloud providers, regardless of the differences between them. Currently, there are no standard API to store and access cloud databases. Legacy applications should be able to work with cloud databases. Cloud databases should also be able to interface with business intelligence tools already available in the market [18-19].

### 5. Industry Practices in Cloud Databases

Cloud databases are designed for low-cost commodity hardware. They scale out easily by distributing the database across multiple hosts/nodes as the load increases. NoSQL databases have become synonym for cloud databases. Few commonly used cloud databases in the industry are described below.

#### 5.1 Amazon Simple Storage Service (S3) and Databases

Amazon S3 is Internet based storage service. It stores objects up to 5GB in size along with 2 KB of Meta data for each object. Objects are organized by buckets. Each bucket is owned by an AWS (Amazon Web Services) account. The buckets are identified by a unique, user-assigned key. Buckets and objects are created, listed and retrieved using either a REST or SOAP interface. Amazon offers MySQL, Oracle and Microsoft SQL Server virtual instances of databases for deployment in its Amazon

Elastic Compute Cloud (EC2) cloud. Even third party management providers like Elastra and Rightscale offer MySQL images. Scaling is not easy with MySQL but it can be done. EnterpriseDB's Postgres Plus Advanced Server, a transactional database also runs in Amazon's cloud. Earlier Storage was tied to the EC2 instance. Termination of instance means loss of data associated with that instance. With Amazon's Elastic Block Store (EBS), user can choose to allocate storage volumes that persist reliably and independently from EC2 instances. Amazon Relational Database Service (RDS) is also a web service that makes it easy to set up and scale a relational database in the Cloud. It is designed for developers or businesses that require the full features and capabilities of a relational database. It gives access to the capabilities of a MySQL, Oracle or SQL Server database engines running on Amazon RDS database instance [20-21].

### 5.2 Amazon SimpleDB

It is a highly available, scalable and flexible non-relational data store. It works closely with Amazon S3 and Amazon EC2 to provide the ability to store, process and query data sets in the cloud. It is NoSQL and name/value pair data store. It offers a simple interface of Get, Post, Delete and Query to run queries on structured data. It is comprised of domains, items, attributes and values. A domain is comparable to a table or a worksheet in a spreadsheet e.g. employee table. Domains are further comprised of items (rows) and items are described by attribute-value pairs. Unlike a spreadsheet, it allows cells to contain multiple values per entry. Each item can have its own unique set of associated attributes(e.g. item "1" might have attributes "Basic" and "tax" whereas item "2" may have attributes "Basic", "tax" and "Saving". It provides scalability by allowing user to partition the workload across multiple domains. Initially, user is allocated a maximum of 250 domains. User can choose between consistency and eventual consistency. But with complex applications, it is difficult to maintain data integrity. It allows user to encrypt data before saving it. It does not decode the data but query directly on the strings stored. It automatically manages replication, indexing of data and performance tuning [22].

### 5.3 Google App's Bigtable

It is a distributed storage system based on GFS (Google File system) for structured data. It implements a replicated shared-nothing database. It has been successfully deployed in many Google products like Google app engine. It allows a more complex data store than SimpleDB. It allows entities and properties comparable to tables and columns. One can create an entity by creating a python object. The Google Datastore API also allows a get, put, delete format for accessing data. It also offers a non-SQL language

called GQL () which is not as feature rich as SQL. Select statements in GQL can be performed on one table only. GQL does not support the "Join" statement [23, 24].

### 5.4 MapReduce

It is an easy-to-use programming model that supports parallel architecture. It is very scalable and works in a distributed manner. It is useful for massive data processing, large scale search and data analysis in the cloud. It provides an abstraction by defining a "mapper" and a "reducer". The "mapper" is applied to every input key/value pair to generate an arbitrary number of intermediate key/value pairs. The "reducer" is applied to all values associated with the same intermediate key to generate output key/value pairs. It has sufficient expression capability to support many real world algorithms and tasks. It can partition the input data, schedule the execution of program across a set of machines, handle machine failures and manage the inter-machine communication. But it cannot be compared to database systems [25].

### 5.5 Hadoop

It is a programming framework for implementing MapReduce across large grid of servers. It is distributed in nature and has better scalability than relational and column store databases. It is more suitable for unstructured data. It is not for mixed workloads, complex data structures and multitasking. Hadoop is a Java based open source project. With the support from Yahoo, Hadoop has achieved great progress. It has been deployed in a large system with 4,000 nodes and is used in many large scale data processing tasks. It enables the addition of Java software Components and provides HDFS (Hadoop Distributed File System) and has been extended to include HBase, a column store database [26].

### 5.6 Windows Azure Cloud Storage

The aim of Windows Azure Storage is to let users and applications access their data efficiently from anywhere at any time using simple and familiar programming API. They can use scalable storage to store any amount of data for any length of time on pay per use basis. It supports structured as well as unstructured data, NoSQL databases and queues. It provides three data abstractions: Blobs, Tables and Queues. Blobs provide a simple interface for storing named files along with metadata for the file. Tables provide structured storage. A Table is a set of entities, which contain a set of properties. Queues provide reliable storage and delivery of messages for an application. All information held in Windows Azure storage is replicated three times which allows fault tolerance [27].

### 5.7 Microsoft SQL Server Data Services (SDDS)

It is a key/value data store, which is also called the cloud extension of Microsoft's SQL Server. It integrates with Microsoft's Sync Framework, which is a .NET library for synchronizing dissimilar data sources. It provides schema-free data storage, SOAP or REST APIs and a pay-as-you-go payment system. It has three core concepts: Entity, Container and Authority. Entity is a property bag of name and value pairs. Container is a collection of entities. Authority is collection of containers and acts as a billing unit [28].

### 5.8 Sherpa

It was popularly known as PNUTS in earlier publications. Data is organized into tables of records with attributes. Tables can be hashed or ordered. It supports blob data type along with typical data types. It is a simplified relational data model. It supports selection and projection from a single table and avoids join operation. Data is replicated asynchronously. It can operate in high availability or high consistency mode. Hadoop can use Sherpa as a data store instead of the native HDFS [29].

### 5.9 Dynamo

It is a highly available, scalable and distributed key-value data-store used by Amazon's core services. It uses eventual consistency to achieve high level of availability i.e. it can write anywhere and update will eventually propagate to all replicas asynchronously. There is no record structure or indexes in Dynamo. It permits only single key updates. It makes extensive use of object versioning and application-assisted conflict resolution [30].

### 5.10 MegaStore

It blends the scalability of a NoSQL data-store and the convenience of a traditional RDBMS to meet the storage requirements of interactive Internet services such as e-mail, documents, social networking. It uses synchronous replication to achieve high availability and a consistent view of the data. It provides transactional (ACID) guarantees within an entity group. It is a flexible data model with user-defined schema, full-text indexes and queues [31].

### 5.11 CouchDB

CouchDB is a free, open-source, Apache project since early 2008. It is a document-oriented database written in Erlang. It belongs to NoSQL generation of databases. Documents (i.e. records) are stored in JSON (JavaScript Object Notation) format and are accessed through an HTTP interface. It allows "views" to be dynamically

created using JavaScript. These views map the document data onto a table-like structure that can be indexed and queried. It does not support a non-procedural query language. It achieves scalability through asynchronous replication. It has unique capability to serve as a self-contained application server and database [32].

### 5.12 MongoDB

MongoDB is a GPL (General Public License) open source document-oriented JSON database system being developed at 10gen by Geir Magnusson and Dwight Merriman. It is designed to be a true object database, rather than a pure key/value store. It stores data in JSON-like documents with dynamic schemas. It provides the speed and scalability of key-value stores and rich functionality like indexes and dynamic queries of relational databases. It provides horizontal scalability [33].

Though NoSQL databases are widely accepted as cloud databases in the database landscape, they are not a solution for all problems. They can work easily with large sparse data, but do not provide transactional integrity, flexible indexing, querying and SQL. They are not able to connect with commonly used Business Intelligence tools. It is difficult to find experienced NoSQL programmers, developers and administrators to install and maintain them. So, Cloud databases should be used with full awareness of their limitations.

## 6. Conclusions

Massive data generated by web-based applications have changed the whole database scenario. Cloud databases appear to be a good solution for handling such data. Moreover, all organizations cannot afford to set up expensive data center infrastructure for managing their own databases. The growing popularity of Cloud databases is marking the beginning of new era of databases. Though cloud databases are not ACID compliant, they are able to handle massive workloads of web-based applications, which do not require such guarantees. Different Cloud databases are available in the market. They share similar concepts and features such as schema free database, simple API, eventual/timeline consistency, scalability synchronous/asynchronous replication etc. But each has its unique API, query interface, data model and database functions. These concepts need to be standardized for their better growth. Cloud computing and Cloud databases are set to rule the next decade by overcoming the limitations they have.

## 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] Jiye Wu et al., “Recent Advances in Cloud Storage”, in *Third International Symposium on Computer Science and Computational Technology (ISCST ’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/oes-refarch-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 IEEE Computer Society Technical Committee on Data Engineering*, 2009, 32(1):3-12.
- [6] <http://aws.amazon.com/simplifiedb/> last accessed on May 23, 2012.
- [7] B.F. Cooper et al., “PNUTS: Yahoo!’s Hosted Data Serving Platform”, in *International Conference on Very Large Data Bases (VLDB)*, Vol. 1, no. 2, 2008, pp. 1277–1288.
- [8] Mike Hogan, “Cloud Computing & Databases”, November 14, 2008.
- [9] Emmanuel Cecchet et al., “Dolly: Virtualization-driven Database Provisioning for the Cloud”, *UMass Technical Report UM-CS-2010-006*.
- [10] Daniel J. Abadi, “ColumnStores vs. RowStores: How Different Are They Really?” in *International Conference on Management of Data- SIGMOD’08*.
- [11] Donald Kossmann, Tim Kraska, Simon Loesing, “An Evaluation of Alternative Architectures for Transaction Processing in the Cloud”, *SIGMOD’10*, June 2010.
- [12] Daniel J. Abadi et al., “Column-oriented Database Systems”, *VLDB ’09*.
- [13] Stonebraker, et al., “C-Store: A Column-oriented DBMS”.
- [14] Thakur Ramjiram Singh, “Cloud Computing: An Analysis”, *International Journal of Enterprise Computing and Business Systems*, Vol. 1, issue 2, July 2011, pp. 2230-8849.
- [15] Rick Cattell, “Scalable SQL and NoSQL Data Stores”, *ACM SIGMOD*, Vol. 39, Issue 4, 2011, pp. 12-27.
- [16] Arpita Mathur et al., “Cloud Based Distributed Databases: The Future Ahead”, *International Journal on Computer Science and Engineering (IJCSSE)* Vol. 3, No. 6, 2011.
- [17] Bo Peng, “Implementation Issues of A Cloud Computing Platform”, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*.
- [18] 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.
- [19] Maggiani, R. “Cloud computing is changing how we communicate”, *IPCC 2009*, 2009, pp. 1-4.
- [20] <http://aws.amazon.com/rds/S3> last accessed on May 24, 2012.
- [21] <http://aws.amazon.com/rds/> last accessed on May 24, 2012.
- [22] <http://aws.amazon.com/simplifiedb> last accessed on May 25, 2012.
- [23] S. Ghemawat et al., “The Google File System”, in *proceeding of 19th ACM Symp. Operating System Principles (SOSP 03)*, ACM Press, 2003, pp. 29–43.
- [24] F. Chang et al., “Bigtable: A Distributed Storage System for Structured Data”, in *7th Usenix Symp. Operating Systems Design and Implementation (OSDI 06)*, Usenix Assoc., 2006, pp. 205–218.
- [25] 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.
- [26] 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](http://hadoop.apache.org/core/docs/r0.16.4/hdfs_design.html) last accessed on May 27, 2012.
- [27] Troy Davis, “Cloud Computing Use Cases and Considerations”, <http://digissance.com/CloudComputingTalk.pdf> last accessed on June 10, 2012
- [28] [www.windowsazure.com/en-us/develop/net.../cloud-storage/](http://www.windowsazure.com/en-us/develop/net.../cloud-storage/) last accessed on June 10, 2012
- [29] Brian Cooper et al., “Building a Cloud for Yahoo”, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2009.
- [30] Giuseppe DeCandia et al., “Dynamo: Amazon’s Highly Available Key-value Store”, in *of 21<sup>st</sup> ACM Symposium on Operating System Principles, SOSP 2007*, pp 205-220.
- [31] Jason Baker et al., “Megastore: Providing Scalable, Highly Available Storage for Interactive Services”, in *5<sup>th</sup> Biennial Conference on Innovative Data Systems Research (CIDR ’11)*, 2011, pp.223-234.
- [32] <http://www.couchbase.com/couchdb> last accessed on May31, 2012.
- [33] <http://www.mongodb.org> last accessed on May31, 2012.

**Indu Arora** obtained her MCA degree from Guru Nanak Dev University in 1992. She has been working as Assistant Professor in Computer Science & Applications at MCM DAV College, Chandigarh since 1998. She also served at BBK DAV College (Aug. 1993- Oct. 1997) and AB College, Pathankot (Aug. 1992 – Feb. 1993). She is also pursuing Doctor of Philosophy from Department of Computer Science & Applications from Panjab University, Chandigarh. Her research interests include Internet technologies, databases and Cloud Computing. She has many research papers to her credit.

**Dr. Anu Gupta** has been working as Assistant Professor in Computer Science and Applications at Panjab University, Chandigarh (India) since July 1998. She held the position of Chairperson, Department of Computer Science & Applications, Panjab University, Chandigarh from Feb. 2008 to Jan. 2011. She was awarded University medal for securing first position in M.C.A. at Punjabi University, Patiala, Punjab in the year 1997. She has the experience of working on several platforms using a variety of development tools and application packages. She obtained Doctor of Philosophy Degree from Panjab University in the area of Free/Open Source Software. Her research interests include Cloud Computing, Networking, Multimedia Technologies, E-Commerce and Software Engineering. She is a life-member of ‘Computer Society of India’ and ‘Indian Academy of Science’. She has published several research papers in various journals and conferences.