International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 2, Issue 4, July – August 2013 ISSN 2278-6856

Cloud Storage: A Survey

Amira Elzeiny[1], Ahmed Abo Elfetouh[2] ,and Alaa Riad[3]

1,2,3 Mansoura University, Faculty of Computer and Information Sciences
Information System Department, Egypt

Abstract: ***Cloud computing is a general term for anything that involves delivering hosted services over the Internet. Instead of running programs and data on an individual desktop computer, everything is hosted in the "cloud". It offers a major change in how to store information and run applications. One of the primary uses of cloud computing is for data storage. With cloud storage, data is stored on multiple third-party servers, rather than on the dedicated servers used in traditional distributed data storage. In cloud storage, the user sees a virtual server; it appears as if the data is stored in a particular place with a specific name. But that place doesn't exist in reality. It's just to reference a virtual space in the cloud. This paper gives a quick introduction to cloud storage. It discusses the need for cloud storage, the main developing requirements and the challenges to cope with the changing IT and business environment. It reviews the cloud storage general architecture. And finally discusses some of the current cloud storage service providers.***

Keywords: cloud computing, cloud storage, relational database, NOSQL database

1. INTRODUCTION

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. The use of the term "cloud" in describing these new models arose from architecture drawings that typically used a cloud as the icon for a network. The cloud represents any-to-any network connectivity in an abstract way. In this abstraction, the network connectivity in the cloud is represented without concern for how it is made to happen. Cloud computing is considered as IT revolution. It frees companies or users from large IT capital investments, and enables them to plug into extremely powerful computing resources over the network. Cloud Computing has five essential characteristics On-demand self-service, broad network access, resource pooling, rapid elasticity and Measured service [23].

Data management applications are potential candidates for deployment in the cloud to get the advantages of using the cloud. This is because the cost of developing large database systems is high in both hardware and software. For many companies especially for start-ups and medium-sized businesses, the pay-as-you-go cloud computing model, along with having someone else worrying about maintaining the hardware and managing the database is very attractive.

Traditional approach of storing data locally in the user's hard drive is not able to cope with the changing requirements of users who daily deal with massive amounts of digital data, and hence need more scalability, high availability, and optimized resources allocation. It seems to have a limitation on handling such big data volumes and modern workloads. The cloud storage provides the users with all these capabilities and more.

Cloud Computing provides an opportunity to store data in Cloud Storage instead of storing it to computer's local hard drive. Users do not need to maintain large storage infrastructures. They can store data in remote data centers, controlled and managed by big companies like Apple, Microsoft, Google, and Amazon etc. Files saved in the cloud storage can be accessed from any where with any device with an Internet connection. Cloud Storage is an important part of cloud computing. Cloud storage is an online storage available on network hosted by third party vendors. Data is stored on virtualized pools of storage. It is delivered as a service on demand in a scalable and multi-tenant way [10].

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 a virtual storage that enables users to store documents and objects. Dropbox, iCloud etc. are popular cloud storage services. DaaS allows user to store data at a remote disk available through Internet. A cloud database is a database delivered to users on demand through the Internet from a cloud database provider's servers. It 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. 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 [4]. In this paper the two terms "cloud storage" and "cloud database" are used interchangeably to refer to the database in the cloud environment.

One of the biggest challenges facing web applications is not the lack of computational power but efficiently and resiliently processing of a huge amount of database query traffic [22].

For more than 30 years, relational databases were the perfect storage solution with their impressive capabilities on transactions and queries management. However, storage requirements for the new generation of applications are huge different from legacy applications. Problems start arising once these databases have to become distributed in order to cope with the demands of new applications; they are not designed to scale.

Even though relational databases have provided the users with the best mix of simplicity, robustness, flexibility, performance, scalability and compatibility, relational databases are found to be inadequate in distributed processing involving very large number of servers and handling Big Data applications [24].

The web has changed the requirements of storage database systems for the next generation of applications. So there was a need for a non-relational database to store data without explicit and structured mechanisms to link data from different buckets to one another. It is called as NoSQL database. NoSQL database (not only SQL) is defined as a non-relational, shared nothing, horizontally scalable database without ACID guarantees. It can store and retrieve unstructured, semi-structured and structured data. NoSQL database has many forms (e.g. document-based, graph-based, object-based, key-value store, etc.)

On a basic level, there are three core categories of NoSQL databases [24]:

1. Key-value Stores: Data is stored as key-value pairs such that values are indexed for retrieval by keys. These systems can hold structured and unstructured data. An example is Amazon’s SimpleDB.
2. Column-oriented Databases: These types of databases contain one extendable column of closely related data rather than sets of information in a strictly structured table of columns and rows as is found in relational databases. The Column family databases stem from Google’s internally-used BigTable, Cassandra and HBase.
3. Document-based Stores: Data is stored and organized as a collection of documents. Users are allowed to add any number of fields of any length to a document. They tend to store JSON-based documents in their databases. Examples of document databases include MongoDB, Apache CouchDB.

Cloud storage has the advantages of scalability and supporting for both relational and non relational databases.

This paper introduces an overview of the cloud storage principles, it’s organized as follows; The next section discusses the need for the cloud storage. Section 3 reviews the cloud storage architecture. The key requirements in a cloud storage system are reviewed in section 4. The major challenges in developing cloud storage are discussed in section 5. Section 6 presents taxonomy of current commercial cloud storage providers, and finally section 7 concludes.

2. The Need for Cloud Storage.

Traditional approach of storing data locally in the user’s hard drive is not able to cope with the changing requirements of users, who daily deal with massive amounts of digital data.

Users want availability of data around the clock using any device from any location. Cloud storage becomes the magic solution choice for such users. The next section discusses why we need cloud storage[10],[15].

A. Growing dependency of all business operations on ICT

Integration of business and ICT is a prerequisite to success these days. Business people depend on ICT to ensure that they can respond immediately to the changes in the competitive business markets as well as to gain flexibility.

B. Explosion of digital data at an exponential rate

It has become easy to capture, alter and store data. Every company is generating massive amounts of data every day and it is growing exponentially. Companies need these amounts of data to transform it into business intelligence for making smarter decisions.

C. New set of data and applications

Users are actively creating and sharing content in the form of text, video and photo postings along with comments, tags and ratings using Blogs. So, there is a need for a new set of data and applications to deal with.

D. The consumerization of IT

Users feel that they can work faster and easier using their own devices such as smart phones, notepads, laptops and iPads. This has led to the consumerization of IT, which has created huge demand for massive and efficient storage accessible from any where with any device.

E. Lack of skilled storage professionals

Organizations are facing the requirement to store huge quantities of digital data. Storage professionals are required to design, manage and maintain the changing storage requirements. Companies are not getting skilled storage professionals due to lack of storage technology education.

F. Availability of Limited funds

Economic slow down, cut on grants and subsidies etc. have also made people to think about cost-effective alternatives of storing data. The cloud storage alternative eliminates the systems costs, the people required to maintain the systems, and at the other side provides high levels of scalability and availability for the organization.

G. Virtualization

The primary accelerator of Cloud Computing and cloud storage is virtualization [5]. Virtualization makes it possible to run multiple applications on virtual machines within the same physical server instead of running only one application on a physical server. It is used for delivering greater availability, scalability along with optimization of resources such as storage, servers etc.

3. Cloud Storage Architecture

It is essential that the cloud database be compatible with cloud computing platforms in order to deliver the benefits of cloud storage to their customers. Shared-nothing and shared-disk are two widely-used storage architectures in database systems.

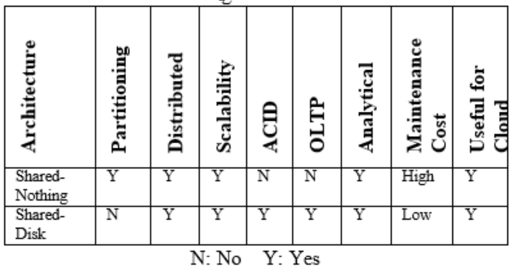
3.1 Shared-Nothing Storage Architecture:

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.[4] One might think the solution for dynamic scalability is quite easy, it's just by adding new servers, unfortunately, it isn't that simple [6]. For example, if we have two servers, each with 50% of the total data, and we add a third server, we just take a third of the data from each server and now we have three servers each owning 33% of the data. Dynamically adding another database server is not as simple as splitting the data across one more server. As more servers are added, data has to be repartitioned. Data partitioning should be done very carefully, otherwise data shipping and joining will become difficult. More data shipping means more latency and network bandwidth bottlenecks. These issues reduce database performance badly.
Amazon's SimpleDB, Hadoop Distributed File System and Yahoo's PNUTS also implement shared-nothing architecture [4].

3.2 Shared-Disk Storage Architecture:

In this architecture the whole database is treated as a single large piece of database stored on 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 pure Scale and Sybase support this architecture [4].

Table [1] A comparison between shared-disk and shared-nothing architectures



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

N: No Y: Yes

The Shared-Disk Database Architecture is Ideal for Cloud Databases

The shared-disk database architecture is ideally suited to cloud computing. It requires fewer and lower-cost servers, it provides high-availability, it reduces maintenance costs by eliminating partitioning, and it delivers dynamic scalability [6]. Table [1] shows a comparison between shared-nothing and shared disk storage architectures.
In shared-disk database architecture, all of the data is available to all of the servers; there is no partitioning of the data. As a result, if you are using two servers, and your query takes 0.5 seconds, you can dynamically add another server and the same query might now take .35 seconds. In other words, shared-disk databases support elastic scalability. There are many advantages for using the shared-disk DBMS architecture in addition to elastic scalability.
The following are some of these advantages [4]:

- ***Fewer servers required***

Since shared-nothing databases break the data into distinct pieces, it is not sufficient to have a single server for each data set, we need a back-up in case the first one fails. This is called a master-slave configuration.
Shared-disk is a master-master configuration, so each node provides fail-over for the other nodes. This reduces the number of servers required by half.

- ***Lower cost servers***

In a shared-nothing database, each server must be run at low CPU utilization in order to be able to accommodate spikes in usage for that server's data. This means that we are buying large expensive servers to handle the peaks.
Shared-disk, on the other hand, spreads these usage spikes across the entire cluster. As a result, each system can be run at a higher CPU utilization. This means that with a shared-disk database you can purchase lower-cost commodity servers instead of paying a large premium for high-end computers. This also extends the life span of existing servers, since they needn't deliver cutting-edge performance.

- ***Simplified maintenance/upgrade process***

Servers that are part of a shared-disk database can be upgraded individually, while the cluster remains online. We can selectively take nodes out of service, upgrade them, and put them back in service while the other nodes continue to operate.
In shared-nothing database this is not possible because each individual node owns a specific piece of data. Take out one server in a shared-nothing database and the entire cluster must be shut down.

- ***High-availability***

As the nodes in a shared-disk database are completely interchangeable, we can lose nodes and the performance may degrade, but the system keeps operating.
If a shared-nothing database loses a server the system goes down until a slave to the server is set up. In addition,

each time the database is (re)partitioned, the system shoud be shut down. so shared-nothing involves more scheduled and unscheduled downtime than shared-disk systems.

• ***Reduced partitioning and tuning services***

In shared-nothing cloud storage, the data must be partitioned. While it is fairly straightforward to simply split the data across servers, thoughtfully partitioning the data to minimize the traffic between nodes in the cluster (data shipping) requires a great deal of ongoing analysis and tuning.

Attempting to accomplish this in a static shared-nothing cluster is a significant challenge, but attempting to do so with a dynamically scaling database cluster is not a simple task.

• ***Reduced support costs***

One of the benefits of cloud databases is that they shift much of the low-level DBA functions to experts who are managing the databases in a centralized manner for all of the users. While tuning a shared-nothing database requires the coordinated involvement of both the DBA and the application programmer which significantly increases support costs.

Shared-disk databases cleanly separate the functions of the DBA and the application developer, which is ideal for cloud databases. It also provides seamless load-balancing, further reducing support costs in a cloud environment.

3.3 General Cloud Storage Architecture

Cloud storage has led to the delivery of new application architectures. In these architectures, applications are fully contained on a variety of devices such as Smart phones, Tablets, PCs etc., and the backend is cloud storage accessible via web-oriented Application Programming Interfaces (APIs). Different Cloud storage providers use different storage architectures [17]. Figure [1] illustrates a generalized architecture of Cloud Storage as presented in [10].

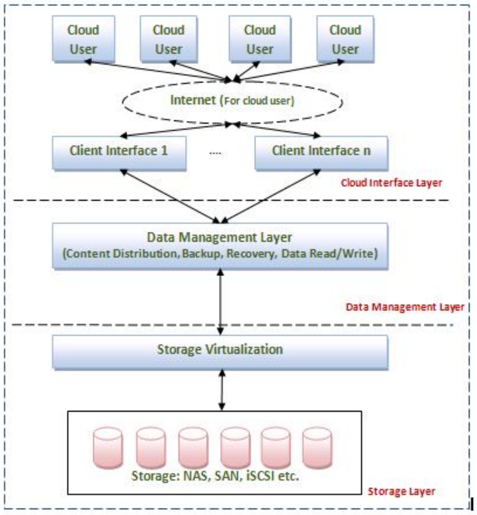


Figure [1] Generalized Architecture of Cloud Storage

This architecture includes three layers:

A. Cloud Interface Layer

It's a software layer provided by the cloud storage provider to connect cloud users to cloud storage service through Internet. This layer applies authentication and authorization techniques to authenticate the users.

B. Data Management Layer

It's a software layer used to manage data of a particular cloud client. Data management is mainly concerned with activities like data storage, content distribution across storage location, data partitioning, synchronization, maintaining consistency, replication, controlling movement of data over network, backup, data recovery, handling millions of users, maintaining metadata and catalogue etc.

C. Storage Layer

The Storage layer consists of two parts:

□ ***Virtualization*:** Storage virtualization gives illusion of unified storage. It maps distributed heterogeneous storage devices to a single continuous storage space and creates a shared dynamic platform. It is implemented by storage virtualization technology. Few virtualization technologies provide built-in availability, security and scalability to applications.

□ ***Basic storage:*** It comprises of database servers and storage devices of heterogeneous nature such as DAS, SAN, NAS etc.

4. Key Requirements in Cloud Storage Systems

The appeal of cloud storage is due to some of the same attributes that define other cloud services: pay as you go, the illusion of infinite capacity (elasticity), and the simplicity of use and management. It is therefore important that any interface for cloud storage support these attributes [15].

The main design requirements of cloud storage are scalability, availability, security, multi-tenancy, reliability, speed, control, cost, and simplicity. Storage needs can scale up or down depending on business requirements. So, Cloud storage should be scalable to meet requests from unlimited and concurrent users without affecting performance and speed.

Cloud Storage services should be available round the clock. Decentralization techniques such as replication are used for fault–tolerance and better availability of cloud services [15]. Data is replicated on different servers residing at different locations to avoid a single point of failure. Multiple nodes provide same services. If primary node fails, backup nodes take over.

Multi-tenancy means that storage is used by multiple users (tenants). Tenants should be able to gain access to their data without any disruption. So data should be stored in such a way that it is always available without any downtime. De-duplication and compression services are used to reduce storage space requirement by eliminating redundant data. They also reduce the amount of data that must be sent across a network, and reduce the amount of storage that users may consume and hence lowers their bill [19].

In general, successful cloud data management systems are designed to satisfy as much as possible from the following *wish list* [1]:

• ***Availability***
They must be always accessible even there is a network failure or a whole data center has gone offline.

• ***Scalability***
They must be able to support very large growing databases with very high request rates at very low latency.

• ***Elasticity***
They must be able to satisfy changing application requirements in both directions (scaling up or scaling down). Moreover, the system must be able to gracefully respond to these changing requirements and quickly recover to its steady state.

• ***Performance***
On public cloud computing platforms, pricing is structured in a way such that one pays only for what one uses, so the vendor price increases linearly with the requisite storage, network bandwidth, and compute power. So, the system performance has a direct effect on its costs. Thus, efficient system performance is a crucial requirement to save money.

• ***Multi-tenancy***
They must be able to support many applications on the same hardware and software infrastructure. And, the performance of these applications must be isolated from each other. Also, adding new applications should require little or no effort beyond that of ensuring that enough system capacity has been provisioned for the new load.

• ***Load and Tenant Balancing***
They must be able to automatically move load between servers so that most of the hardware resources are effectively utilized and to avoid any resource overloading situations.

• ***Fault Tolerance***
For transactional workloads, a fault tolerant cloud data management system needs to be able to recover from a failure without losing any data or updates from recently committed transactions. Moreover, it needs to successfully commit transactions and make progress on a workload even in the face of worker node failures. For analytical workloads, a fault tolerant cloud data management system should not need to restart a query if one of the nodes involved in query processing fails.

• ***Ability to run in a heterogeneous environment***
There is a strong trend towards increasing the number of nodes that participate in query execution. A cloud data management system should be designed to run in a heterogeneous environment and must take appropriate measures to prevent performance degrading due to parallel processing on distributed nodes.

• ***Flexible query interface***
They should support both SQL and no-SQL interface languages (e.g. MapReduce). Moreover, they should provide mechanism for allowing the user to write user defined functions (UDFs).

According to a survey on users concerns with cloud storage services[15] , the key requirements in the provided cloud storage services include security, control, performance, support, vendor lock-in. Figure[2] shown the users concerns priorities with cloud storage services. We can see that the most priorities for the users is security, control, and performance. And hence these issues are challenging areas in cloud storage.

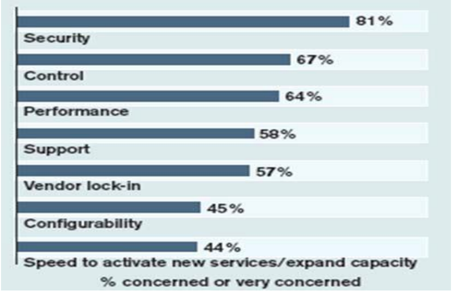


Figure [2] Survey on concerns with cloud storage services

5. Cloud Storage Challenges

Cloud DBMSs should support features of cloud computing as well as of traditional databases for wider acceptability [4].Deploying data-intensive applications on cloud environment is not a trivial or straightforward task. Figure [3] shows the challenges in developing a cloud storage system. Armbrust et al. [20] and Abadi [25] argued a list of obstacles to the growth of cloud computing applications as follows:

• ***Availability and Fault Tolerance***
Organizations worry about whether cloud computing services will have adequate availability. High availability is one of the most challenging goals because even the slightest outage can have significant financial consequences and impacts customer trust.

• ***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.

• ***Data Consistency and Integrity***
Data integrity is the most critical requirement of all business applications and is maintained through database constraints. 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.

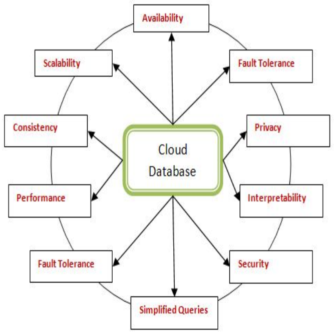


Figure [3] The Challenges in Developing Cloud Storage Systems

• ***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.

• ***Performance and Data Transfer Bottlenecks***
Cloud users and cloud providers have to think about the implications of placement and traffic at every level of the system if they want to minimize costs and enhance the system performance. More data shipping means more latency and network bandwidth bottlenecks. These issues reduce database performance badly.

• ***Data Portability and Interoperability***
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, and cloud databases should also be able to interface with business intelligence tools already available in the market [19]. This can be achieved by developing portable and interoperable components.

• ***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.

• ***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 [4]. 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 confidentiality to different databases on the same hardware is also a big challenge.

• ***Application Parallelization***
Computing power is elastic but only if workload is parallelizable. Getting additional computational resources is not as simple as just upgrading to a bigger and more powerful machine on the fly. However, the additional resources are typically obtained by allocating additional server instances to a task.

6. Taxonomy of Current Commercial Cloud Storage Providers

Enterprise applications are broadly categorized into transactional and analytical applications. Relational databases played an important 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 [27]. But they could not overpower the relational databases.

Then Internet revolution and web applications started producing massive sparse and unstructured data. Relational databases has rigid architecture based on tables, columns, indexes, relationships and schema, so RDBMS are not suitable for handling such data sets. The need to store and process such big data defined the role of

NoSQL databases in the database technology as cloud databases.

NoSQL means 'Not Only SQL' or 'Not Relational' [7]. 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.

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. NoSQL cloud databases follow BASE (Basically Available, Soft state, eventually consistent) in contrast to the ACID guarantees. So, they are not suitable for transaction applications. They provide high availability at the cost of consistency [26].

The cloud databases supports both relational and NoSQL databases. Figure [4] shows taxonomy of different cloud DBMS software packages exist. The cloud DBMS offerings is divided into four categories based on whether or not they are "relational" and the degree to which they are "native" to the cloud (e.g. integrated part of a cloud service) [7].

Amazon was the first company to provide cloud services under the name Amazon Web Services (AWS) to external customers in 2006 [14].Currently, Amazon offers two structured database services; Amazon SimpleDB, and Amazon Relational Database Service (RDS). SimpleDB was the first structured database service offered by Amazon. It is a simple key-value store and thus not relational. It was optimized mainly towards simplicity and availability.

After noticing that SimpleDB was not suitable for applications that need a certain minimum of features and robustness in the DB tier, Amazon launched its second database service, called Amazon Relational Database Service (RDS) in late 2009.

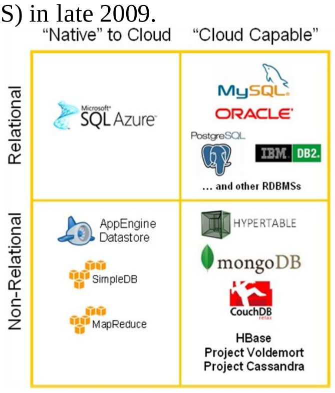


Figure [4] Taxonomy of Cloud Service Providers

The Microsoft SQL Azure Database was the first RDBMS designed for 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 [4].

The key requirement for any database to run in a cloud-based infrastructure is the ability to have administrator rights to be able to install and configure the database, and the ability to generally have persistent volumes to mount the database. Virtually any RDBMS – Oracle, IBM DB2, SQL Server, Sybase, etc. – can run in most of the cloud infrastructures [7].

An example on non-relational cloud DBMS is MongoDB. It's 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 [4].

NoSQL databases are widely accepted as cloud databases in the database environment, but they are not a solution for all problems. They can work easily with large different data, but do not provide transactional integrity, flexible indexing, querying and SQL. They are not able to connect with commonly used Business Intelligence tools. Also it is difficult to find experienced NoSQL programmers, developers and administrators to install and maintain them.

7. Conclusion

Cloud Storage is an important part of cloud computing in which storage is made available to users on-demand and pay-per-usage basis from any where through Internet. There are many open issues regarding the cloud storage. This paper presented a survey on cloud storage. It reviews the main principles of the cloud storage. It presents an overview of the cloud storage architecture and why we need cloud storage. There are still many challenges in the adaption of cloud storage. Scalability and security are the most challenges facing the cloud storage. Cloud storage has the capability to change the whole storage and data backup scenario. It is the future storage.

References

[1] Sakr, Sherif, et al. "A survey of large scale data management approaches in cloud environments." *Communications Surveys & Tutorials, IEEE* 13.3 (2011): 311-336.

[2] Rimal, Bhaskar Prasad, Eunmi Choi, and Ian Lumb. "A taxonomy and survey of cloud computing systems." *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*. *IEEE*, 2009.

[3] Kraska, Tim. *Building database applications in the cloud*. Diss. Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 18832, 2010, 2010.

[4] Arora, Indu, and Anu Gupta. "Cloud Databases: A Paradigm Shift in Databases." *International J. of Computer Science Issues* 9.4 (2012): 77-83.

[5] TT II, C. C. H. H. A. A. R. R., and CCLL EE. "Data Storage Virtualization in Cloud Computing."

[6] Hogan, Mike. "Cloud computing & databases." *ScaleDB Inc* (2008).

[7] http://www.cloudbzz.com/cloud-dbms-databases-and-cloud-computing/

[8] Wu, Jiyi, et al. "Recent Advances in Cloud Storage." *Proceedings of the Third International Symposium on Computer Science and Computational Technology (ISCSCT'10)*. 2010.

[9] TT II, C. C. H. H. A. A. R. R., and CCLL EE. "Data Storage Virtualization in Cloud Computing."

[10] Arora, Indu, and Anu Gupta. "Opportunities, Concerns and Challenges in the Adoption of Cloud Storage."

[11] Curino, Carlo, et al. "Relational cloud: A database-as-a-service for the cloud." (2011).

[12] AL FEEL, H. A. Y. T. H. A. M., and Mohamed Khafagy. "Search content via Cloud Storage System." *International Journal of Computer Science* 8.

[13] Egger, Daniel. *SQL in the Cloud*. Diss. Master Thesis ETH Zurich, 2009, 2009.

[14] Michel, Daniel. "Databases in the Cloud." *Doktorarbeit, HSR University of Applied Science Rapperswil* (2010).

[15] Wu, Jiyi, et al. "Cloud storage as the infrastructure of cloud computing." *Intelligent Computing and Cognitive Informatics (ICICCI), 2010 International Conference on*. IEEE, 2010.

[16] Mathur, Arpita, Mridul Mathur, and Pallavi Upadhyay. "Cloud Based Distributed Databases: The Future Ahead." *International Journal on Computer Science and Engineering* 3.6 (2011): 2477-2481.

[17] Jones, M. Tim. "Anatomy of a cloud storage infrastructure." *IBM developer works (November 30, 2010).*

[18] Sasidhar, Talasila, Pavan Kumar Illa, and Subrahmanyam Kodukula. "A Generalized Cloud Storage Architecture with Backup Technology for any Cloud Storage Providers." *International journal of computer application, ISSN*: 2250-1797.

[19] Chappell, David. "A short introduction to cloud platforms." *David Chappell & Associates* (2008).

[20] Fox, Armando, et al. "Above the clouds: A Berkeley view of cloud computing." *Dept. Electrical Eng. and Compter Sciences, University of California, Berkeley, Rep. UCB/EECS* 28 (2009).

[21] Mohammad, Siba, Sebastian Breß, and Eike Schallehn. "Cloud Data Management: A Short Overview and Comparison of Current Approaches." *Grundlagen von Datenbanken*. 2012.

[22] Nikolov, Plamen, and Guillaume Pierre. *Aggregate queries in NoSQL cloud data stores*. Diss. Master Thesis, Vrije Universiteit Amsterdam, 2011.

[23] Donkena, Kaushik, and Subbarayudu Gannamani. "Performance Evaluation of Cloud Database and Traditional Database in terms of Response Time while Retrieving the Data." *Electrical Engineering* (2012).

[24] Padhy, Rabi Prasad, Manas Ranjan Patra, and Suresh Chandra Satapathy. "RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Databases." *International Journal of Advanced Engineering Science and Technologies* 11.1 (2011): 15-30.

[25] Abadi, Daniel J. "Data Management in the Cloud: Limitations and Opportunities." *IEEE Data Eng. Bull.* 32.1 (2009): 3-12.

[26] Peng, Bo, Bin Cui, and Xiaoming Li. "Implementation Issues of A Cloud Computing Platform." *IEEE Data Eng. Bull*. 32.1 (2009): 59-66.

[27] Abadi, Daniel J., Peter A. Boncz, and Stavros Harizopoulos. "Column-oriented database systems." *Proceedings of the VLDB Endowment* 2.2 (2009): 1664-1665.