Volume 8, No. 2, March 2017 (Special Issue)
International Journal of Advanced Research in Computer Science
REVIEW ARTICLE
Available Online at www.ijarcs.info

ISSN No. 0976-5697

Comparative Study of Traditional Database and Cloud Computing Database

Sapna Jain
Department of CSE, SEST
Jamia Hamdard, New Delhi, India
mailtosapnajain@rediffmail.com

M Afshar Alam
Department of CSE, SEST
Jamia Hamdard, New Delhi, India
mailtoafshar@rediffmail.com

Abstract: This paper gives a comparative analysis of latest technology called "Cloud Computing database " or "database-as-a-service"(DBaaS) called Relational Cloud which is causing a major shift in the Information Technology industry with the traditional databases Realtional Database Management System which have been used since many years .Cloud Computing is a new technology which has been developed to virtualize IT systems and to access the needed applications on the Internet, through web based applications. This means no IT costs for hardware or servers. As compared to the traditional way of building an IT environment is to buy servers, hardware, licenses and to install the software. This is a long and costly process, involving many infrastructure demands and long deployment cycles. This fully IT internal model may be commonplace, but IT as we know it today is being replaced by newer technologies.

Keywords: Cloud computing, Relational Database,SaaS, PaaS, Database management, SQL, NOSQL, DBaaS, Database.com, SQL Azure, Amazon Web Services

I. INTRODUCTION

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly. Cloud computing is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment.

A Cloud can be defined as a parallel and distributed system which has a number of virtualized and interconnected computers. These are actively provisioned and presented as single or more united computing resources depending upon the service level agreement. Cloud has three popular computing paradigms Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). These services include distributed operating system, the distributed database and other services.

The Cloud Computing database is required apace and effectively and should reduce the burdens during routing configuration. The Cloud Database is constructed by collecting a number of sites. The sites are also called as nodes which are interlinked by a communication network. Every single node is a database class. Each database class has its own database, terminals, the central processor and their individual local database management system.

A Relational Database Management System (RDBMS) is a software package with computer programs that controls the creation, maintenance, and use of a database. It allows the organizations to conveniently develop databases for various applications. A database is an integrated collection of data records, files and other objects. A DBMS allows different user application programs to concurrently access the same database. DBMSs may use a variety of database models, such as the relational model or object model to conveniently describe and support applications.

This paper highlights provides a comparative study of the traditional relational database management system and cloud computing database with their applications, advantages and disadvantages.

II. ARCHITECTURE

Traditional Relational Database Management System (RDMS): The concept of database management system is quite interesting to look at over a particular period of time. According to [27], Database Management is developed in four phases from 1970's to late 1990's. Figure [1] clearly illustrates four phases of Database Management System. In early 1970's, organizations used IBM's information management system (IMS) which stores the data using hierarchical model. But the organizations have to maintain expensive main frames in order to relay on IBM's IMS. By early 1980's, IBM's IMS is replaced by the Relational Database Management System (RDMS) such as Oracle. DBMS technology is allowed on personal computers. According to [29] Figure 2-1 shows the phases of the Relational Database Management System. This has kept growing and now this time it shifted to other dimension i.e Cloud Computing. Cloud Computing has been an interesting paradigm in the recent times due to its advantages like scalability, virtualization and pay per use.As pay per use is involved, it is important to consider the resource utilization. Cloud Computing is more helpful for IT industries to improve the management of their own resources in an easy manner. Cloud Computing provides different services such as Infrastructure-as-a-Service(SaaS), Platform-as-a-Service(PaaS) and Software-as-a-Service(SaaS). According to [33] there is an addition to this list of services, called Database-as-a-Service(DaaS). In this service, organizations host their own databases in Cloud Computing. This service provides the acess for DML(Data Manipulation Language) statement features (strore, retrieve, update and delete the data) via the internet following Figure 2.1.

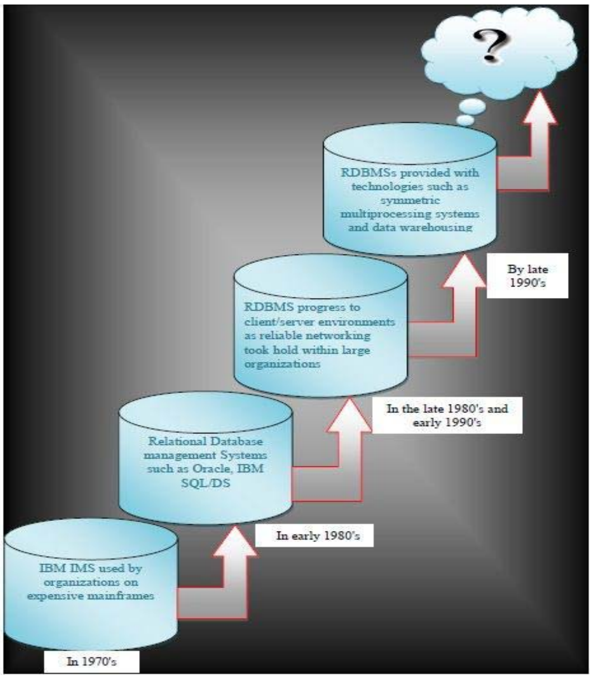

Figure 2.1: Journey of Relational Database Management System

After RDBMS progress to client /server environments and it's implemented on large organizations. In 1990's because of the fast growth of the technology symmetric multiprocessing system and data warehousing options are made available on the RDBMS.

Cloud Computing is more helpful for IT industries to improve the management of their own resources in an easy manner. Cloud Computing provides different services such as Infrastructure-as-a-Service (SaaS), Platform-as-a-Service(PaaS) and Software-as-a-Service(SaaS). According to [33] there is an addition to this list of services, called Database-as-a-Service (DaaS) as shown in figure 2.2. In this service, organizations host their own databases in Cloud Computing. This service provides the access for DML(Data Manipulation Language) statement features (store, retrieve, update and delete the data) via theinternet following [29].

According to [28], a Cloud Database is a combination of different number of nodes (or site collections) and each node has its own database, linked together in the communication network. Cloud Database system is a novel trend in the research because many organizations want to migrate their databases into Cloud to exploit the benefits Cloud Computing. Organizations look at the performance factor of the databases regardless of the paradigm, whether traditional or Cloud. In [30], authorsconducted various experiments on On-premises traditional database in terms of IBM'S DB2, Oracle database and Microsoft SQL Server. The performance of the Cloud Database is evaluated in this research and a comparison is made with that of an on premises traditional database.

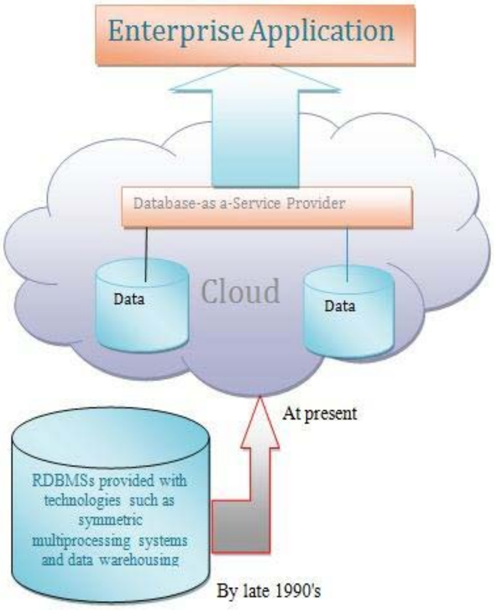

Figure 2.2: Cloud Database as a Service

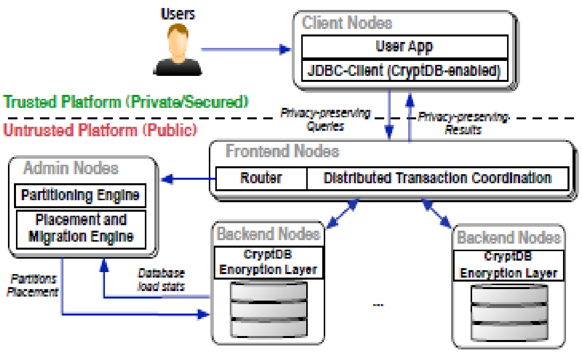

Figure2.3: Relational Cloud Architecture

Relational Cloud uses existing unmodified DBMS engines as the back-end query processing and storage nodes. Each back-end node runs a single database server. The set of back-end machines can change dynamically in response to load. Each tenant of the system which we define as a billable entity (a distinct user with a set of applications, a business unit, or a company)can load one or moredatabases. A database has one or more tables, and an associated workload, defined as the set of queries and transactions issued to it (the set may not be known until run-time). Relational Cloud does not mix the data of two different tenants into a common database or table (unlike [1]), but databases belonging to different tenant's willusually run within the same database server.

Applications communicate with Relational Cloud using a standard connectivity layer such as JDBC. They communicate with the Relational Cloud front-end using a special driver that ensures their data is kept private (e.g., cannot be read by the database administrator).When the front-end receives SQL statements from clients, it consults the router, which analyzes each SQL statement and uses its metadata to determine the execution nodes and plan. The front-end coordinates multi-node transactions, produces a distributed execution plan, and handles fail-over. It also provides a degree of performance isolation by controlling the rate at which queries from different tenants are dispatched.

The front-end monitors the access patterns induced by the workloads and the load on the database servers. Relational Cloud uses this information to periodically determine the best way to: (1) partition each database into one or more pieces, producing multiple partitions when the load on a database exceeds the capacity of a sin gle machine (x3), (2) place the database partitions on the back-end machines to both minimize the number of machines and balanceload, migrate the partitions as needed without causing downtime,and replicate the data for availability (x4), and (3) secure the data and process the queries so that they can run on untrusted back-ends over encrypted data (x5). The Relational Cloud system architecture is shown in Figure 2.3 which depicts these functions and demarcates the trusted and untrusted regions.

Cloud computing is essentially a series of levels that function together in various ways to create a system. This system is also referred to as cloud computing architecture. The cloud creates a system where resources can be pooled and partitioned as needed. Cloud architecture can couple software running on virtualized hardware in multiple locations to provide an on-demand service to user-facing hardware and software. A cloud can be created within an organization's own infrastructure or outsourced to another datacenter. Usually resources in a cloud are virtualized resources because virtualized resources are easier to modify and optimize. A compute cloud requires virtualized storage to support the staging and storage of data. From a user's perspective, it is important that the resources appear to be infinitely scalable, that the service be measurable, and that the pricing be metered in Figure 2.4 [1].

Service Class	Main Access & Management Tool	Service content
SaaS	Web Browser	**Cloud Applications** Social networks, Office suites, CRM, Video processing
PaaS	Cloud Development Environment	**Cloud Platform** Programming languages, Frameworks, Mashups editors, Structured data
IaaS	Virtual Infrastructure Manager	**Cloud Infrastructure** Compute Servers, Data Storage, Firewall, Load Balancer

Figure 2.4: Cloud Computing Stack

III. DATABASE PARTITIONING APPROACH

3.1 ***RDBMS System:*** According to [4] enterprises are becoming data-centric and increasingly producing humongous amounts of data in the form of sales, retail records and other commercial information. This data stored in the database needs to be effectively managed. Enterprises analyze these databases continuously and take informed decisions based on the analysis, so database performance plays a vital role in the overall functioning of the database. At the time of creation of database the scale of meta-data related to the database is small. As the size of the database increases, it encounters gradual deterioration in the performance. This performance degradation motivated the researchers to search for ways to improve the performance by database optimization.Database optimization can be performed at four different layers as shown in Figure 3.1.

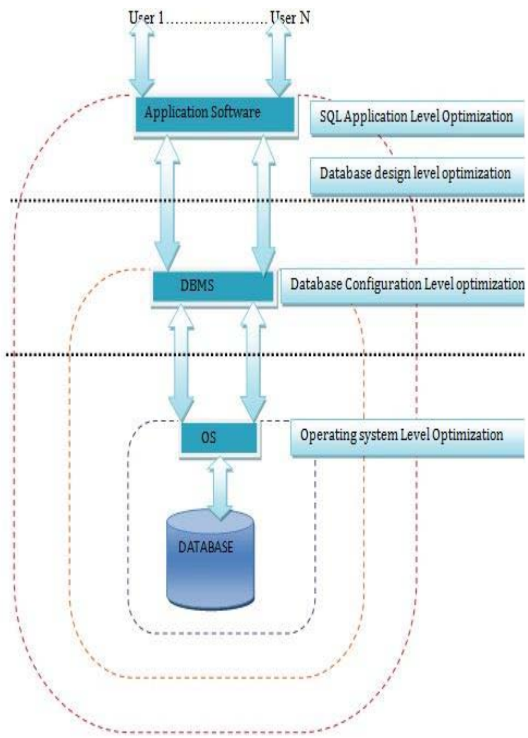

Figure 3.1: Database Performance Optimization Dependency levels.

In these four levels top most level is the SQL application level optimization. In this optimization the transaction time is reduced by indexing the database thereby leadingto improvement in the performance. The database performance translates to reduction in CPU costs in [35]. By indexing the database, the DBMS is enabled to maintain a separate database object storing the metadata related to database. These objects contained a sorted list of column values which contains row identifiers to the corresponding rows in that table as shown in [34]. Indexes are internally organized in a tree structure. According to [37] there are certain disadvantages of using the indexes to the database. Usage of the indexes results in speed up in the query execution, retrieval of data but every additional index added to the index table slows down the manipulation further. Since every INSERT/DELETE/UPDATE can be processed only after updating all the corresponding indexes it takes additional CPU cycles and time to keep the indexes synchronized with the tables. This also results in Database consuming additional space in database.

	Advantages	Disadvantages
1	Optimize the database performance	Using Index slows down manipulation further
2	Using indexes we can speed up queries	Maintenance overhead
3	Reduce CPU cost for query execution	Indexes occupy the additional space in database
4	Avoids full table scan in search queries	INSERT/DELETE/UPDATE can be processed only after updating all the corresponding indexes
5	Table data can be stored in an organized way	Need to maintain index and table synchronization every time.

Figure3.2: Advantages and Disadvantages of using Indexes

3.2 Relational Cloud Database System: It uses database partitioning for two purposes: (1)to scale a single database to multiple nodes, useful when the load exceeds the capacity of a single machine, and (2) to enable more granular placement and load balance on the back-end machinescompared to placing entire databases.

The current partitioning strategy is well-suited to OLTP and Web workloads, but the principles generalize to other workloads as well (such as OLAP). OLTP/Web workloads are characterized by short-lived transactions/queries with little internal parallelism. The way to scale these workloads is to partition the data in a way that minimizesthe number of multi-node transactions (i.e., most transactions should complete by touching data on only one node), and then place the different partitions on different nodes. The goal is to minimize the number of cross-node distributed transactions, which incur overhead both because of the extra work done on each node and because of the increase in the time spent holding locks at the back-ends.

Relational Cloud uses a workload-aware partitioning strategy.The front-end has a component that periodically analyzes query execution traces to identify sets of tuples that are accessed together within individual transactions. The algorithm represents the execution trace as a graph. Each node represents a tuple (or collection of tuples) and an edge is drawn between any two nodes whose tuples are touched within a single transaction. The weight on anedge reflects how often such pair-wise accesses occur in a workload.

Relational Cloud uses graph partitioning [13] to find ` balanced logical partitions, while minimizing the total weight of the cut edges.

This minimization corresponds to find a partitioning of the database tuples that minimizes the number of distributed transactions.

The output of the practitioner is an assignment of individual tuples to logical partitions. Relational Cloud now has to come up with a succinct representation of these partitions, because the front-end's router needs a compact way to determine where to dispatch a given SQL statement. Relational Cloud solves this problem by finding a set of predicates on the tuple attributes. It is natural to formulatethis problem as a classification problem, where we are given a set of tuples (the tuple attributes are features), and a partition label for each tuple (the classification attribute). The system extracts a set of candidate attributes from the predicates used in the trace. The attribute values are fed into a decision tree algorithm together with the partitioning labels. If the decision tree successfully generalizes the partitioning with few simple predicates, a good explanation for the graph partitioning is found. If no predicate-based explanation is found (e.g., if thousands of predicates are generated), the system falls back to lookup tables to represent the partitioning scheme.

The strength of this approach is its independence from schema layout and foreign key information, which allows it to discover intrinsic correlations hidden in the data. As a consequence, this approach is effective in partitioning databases containing multiple many-to-many relationships typical in social-network scenarios and in handling skewed workloads [5].

The main practical difficulty we encountered was in scaling the graph representation. The na¨ıve approach leads to a graph with N nodes and up to N2 edges for an N-tuple database, which is untenable because existing graph partitioning implementations scale only to a few tens of millions of nodes. For this reason, we deviseda series of heuristics that effectively limit the size of the graph. The two most useful heuristics used in Relational Cloud are: (1) blanket statement removal, i.e., the exclusion from the graph occasional statements that scan large portions of the database and (2) sampling tuples and transactions.

IV. COMPARISON

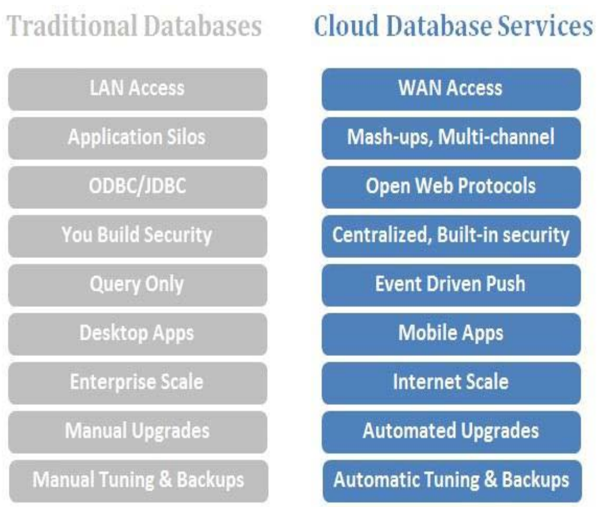

Figure 4.1: Traditional VS Cloud Data Services

The main concern is that the DBMSs and RDBMSs are not cloud-friendly because they are not as scalable as the web-servers and application servers, which can scale from a few machines to hundreds. The traditional DBMSs are not design to run on top of the shared-nothing architecture (where a set of independent machines accomplish a task with minimal resource overlap) and they do not provide the tools needed to scale-out from a few to a large number of machines.

Technology leaders such as Google, Amazon, and Microsoft have demonstrated that data centers comprising thousands to hundreds of thousands compute nodes, provide unprecedented economies-of-scale since multiple applications can share a common infrastructure. All three companies have provided frameworks such as Amazon's AWS, Google's AppEngine and Microsoft Azure for hosting third party application in their clouds (data-center infrastructures).

Because the RDBMs or "transactional data management" databases that back banking, airline reservation, online e-commerce, and supply chain management applications typically rely on the ACID (Atomicity, Consistency, Isolation, Durability) guarantees that databases provide and It is hard to maintain ACID guarantees in the face of data replication over large geographic distances1, they even have developed propriety data management technologies referred to as key-value stores or informally called NO-SQL database management systems.[6] The need for web-based application to support virtually unlimited number of users and to be able to respond to sudden load fluctuations raises the requirement to make them scalable in cloud computing platforms. There is a need that such scalability can be provisioned dynamically without causing any interruption in the service. Key-value stores and other NOSQL database solutions, such as Google Datastore offered with Google AppEngine, Amazon SimpleDB and DynamoDB, MongoDB and others, have been designed so that they can be elastic or can be dynamically provisioned in the presence of load fluctuations.

This issue becomes especially acute in the context of pay-per-use cloud-computing platforms hosting multi-tenant applications. In this model, the service provider is interested in minimizing its operational cost by consolidating multiple ten-ants on as few machines as possible during periods of low activity and distributing these tenants on a larger number of servers during peak usage [7]. Due to the above desirable properties of key-value stores in the context of cloud computing and large-scale data-centers, they are being widely used as the data management tier for cloud-enabled Web applications. Although it is claimed that atomicity at a single key is adequate in the context of many Web-oriented applications, evidence is emerging that indicates that in many application scenarios this is not enough. In such cases, the responsibility to ensure atomicity and consistency of multiple data entities falls on the application developers. This results in the duplication of multi-entity synchronization mechanisms many times in the application software. In addition, as it is widely recognized that concurrent programs are highly vulnerable to subtle bugs and errors, this approach impacts the application reliability adversely. The realization of providing atomicity beyond single entities is widely discussed in developer blogs. Recently, this problem has also been recognized by the senior architects from Amazon and Google, leading to systems like MegaStore [10] that provide transactional guarantees on key-value stores.

V. ADVANTAGES OF CLOUD COMPUTING

5.1 Scalability

The scalability is the ability of a system to handle growing amount of work in a capable manner or its ability to improve when additional resources are added.

The scalability requirement arises due to the constant load fluctuations that are common in the context of Web-based services. In fact these load fluctuations occur at varying frequencies: daily, weekly, and over longer periods. The other source of load variation is due to unpredictable growth (or decline) in usage. The need for scalable design is to ensure that the system capacity can be augmented by adding additional hardware resources whenever warranted by load fluctuations.

Creating a cloud network that offers the maximum level of scalability potential is entirely possible if we apply a more "diagonal" solution. By incorporating the best solutions present in both vertical and horizontal scaling, it is possible to reap the benefits of both models[3]. Once the servers reach the limit of diminishing returns (no growth), we should simply start cloning them. This will allow us to keep a consistent architecture when adding new components, software, apps and users. For most individuals, problems arise from lack of resources not the inherent architecture of their cloud itself. A more diagonal approach should help the business to deal with the current and growing demands that it is facing.

5.2 Elasticity

Elasticity offers the same computing experience to which we are accustomed, with the added benefit of near limitless resources at the same time offering a way to manage the energy consumption. [1][3]

The elastic capabilities offered by cloud computing makes it perfectly suited toward handling certain activities or processes.

Establishing an "in office" communication and online networking infrastructure (for employees). Setting up a system that allows those in the organization a cleaner and more efficient system for communicating and working often leads to greatly increased profits.

Using cloud computing to handle overdrafting - high volume data transfer periods and events. Some businesses only use cloud computing when they run out of their own resources, or perhaps anticipate that they might lack needed functionalities.

This can be something that is scheduled for an annual or bi-annual basis; designed to meet a seasonal demand for a particular product for example. Assigning all customer data and transaction information to a cloud computing element. This allows an organization to keep their customer's data safe even from their own employees. Utilizing a third party to handle all customer data can also pay off in the event of a catastrophic type event. Cloud computing providers tend to keep your information more securely backed-up than most are even aware of. [3]

5.3 Lower Initial Investment

Only things needed to start using the cloud is computer and an Internet connection, it is possible take advantage of most cloud offerings without investing in any new hardware, specialized software, or adding to staff. This is one cloud computing advantage that has universal appeal regardless of the industry or the type of business. This allows organizations and especially startups to invest in new projects and ideas without risk of big loss.

5.4 Easier to manage

There are no power requirements or space considerations to think about and users do not have to understand the underlying technology in order to take advantage of it. There is no need for maintaining and updating any new hardware or software. Planning time is considerably less as well since there are fewer logistical issues.

5.5 Pay as You Go

Large upfront fees is not the norm when it comes to cloud services. Most of the cloud services as I wrote earlier in this paper are available on a month to month basis with no long term contracts. It also gives the benefit of keeping multiple projects running without enormous expenses.

5.6 Scalability

Cloud computing can be scaled to match the changing needs of the small business as it grows. Licenses, storage space, new instances and more can be added as needed.

5.7 Deploy Faster

Usually it is possible to get up and running significantly faster with cloud services than if there is a need to plan, buy, build, and implement in house. With many software as a service applications or other cloud offerings it is possible to start using the service within hours or days rather than weeks or months.

5.8 Location Independent

Because services are offered over the Internet, there are no limits to using cloud software or services just at work or only on one computer. Access from anywhere is a big advantage for people who travel a lot, like to be able to work from home, or whose organization is spread out across multiple locations.

5.8 Device independent

Most web-based software and cloud services are not designed specifically for any one browser or operating system. Many can be accessed via PC, Mac, on tablets and through mobile phones.

5.9 Diminished Costs

Cloud-based administrations can help institutes decrease costs and quicken the utilization of new innovations to meet developing educational needs. Students can utilize office applications without purchasing, install and stay up with the latest on their PCs. It likewise gives the instructors of Pay per use for a few applications.

5.10 Easy Access

Lesson arranges labs, grades, notes, and PowerPoint slides, pretty much anything computerized that you use in training is effectively transferred.

5.11 Security

The information, content, data, pictures – anything you store in the cloud normally requires verification (ID and secret word, for instance) – so it is not effectively available for anybody.

5.12 Shareability

Cloud computing opens up a universe of new conceivable outcomes for students, particularly the individuals who are not served well by customary training frameworks. With cloud computing, one can reach more and more diverse, students.

VI. APPLICATIONS

Microsoft's SQLAZURE

Microsoft SQL Azure Database is a cloud-based relational database service that is built on SQL Server technologies and runs in Microsoft data centers on hardware that is owned, hosted, and maintained by Microsoft.

SQL Azure is probably the most fully-featured relational database available in the cloud. It is based on the SQL Server standalone database but the way data is managed and stored in SQL Azure is significantly different.

Amazon Relational Database Service (Amazon RDS)

Amazon Relational Database Service (Amazon RDS) is a web service that can operate, and to some level scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating the administration tasks. Amazon RDS gives the users access to the capabilities of a MySQL or Oracle database running on their own Amazon RDS database instance. This gives the advantage that the code and applications that use on-premises MySQL or Oracle database can be easily migrated to Amazon RDS.

Google Cloud SQL

Google Cloud SQL is a MySQL database in the Google's cloud. It has all the capabilities and functionality of MySQL. Google Cloud SQL is currently available for Google App Engine applications that are written in Java or Python. It can also be accessed from a command-line tool.

Google Datastore

The Google App Engine Datastore is a schemaless object datastore providing robust, scalable storage mainly targeted for web application. App Engine's datastore is built on top of Bigtable. Bigtable is distributed storage system for managing structured data that is designed to scale to a very large size, petabytes of data across thousands of commodity server. Many Google project like Google Earth, Google Finance including the web indexing use Bigtable for storing data.

Database.com

Database.com is a database management system that is built for cloud computing with multitenancy inherent in its design. Traditional RDBMSs were designed to support on premises deployments for one organization. All core mechanisms such as system catalog, cashing mechanisms and query optimizer are built to support single-tenant applications and to run directly on a specifically tuned host operating system and hardware. Only possible way to build multi-tenant cloud database service with standard RDBMS is to use virtualization. Unfortunately, the extra overhead of the hypervisor typically hurts the performance of the RDBMS. Database.com combines several different persistence technologies, including a custom -designed relational database schema, which are innately designed for clouds and multitenancy no virtualization required.

VII. CONCLUSION

Database management system, for a long time has been an integral part of the computing. As the whole IT world is moving to the cloud whether you are assembling, managing or developing on a cloud computing platform, you need a cloud compatible database. In this work I gave a short overview of cloud computing and presented couple of the currently available companies that offer database as a service in the cloud. Although they differ from the most widely used “traditional” relational database systems and most of them might require revision and recoding of the existing applications, it is obvious that they bring a lot of benefits especially with the offer for fully managed and automated database administration tuning and optimization.

Cloud database system are built to use the power of the cloud, they are extremely scalable and elastic, giving the opportunity to start small and expand as you need mitigating the risk and uncertainties of investing in IT equipment and professional IT support. Cloud computing in general, with the flexible pricing models and different plans it presents the one of the best solutions for startup and small companies that are developing new products and does not have the financial power to risk and invest in uncertain projects.

VIII. REFERENCES

[1]. Cloud Computing Bible - Barrie Sosinsky, Janury 2012. ISBN: 978-0-470-90356-8

[2]. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[3]. Introduction to cloud computing - Ivanka Menken, Emereo Publishing 2011

[4]. Understanding PaaS - Michael P. McGrath, O'Reilly Media January 2012

[5]. Data Management Challenges in Cloud Computing Infrastructures - Divyakant Agrawal, A. E., University of California, Santa Barbara.

[6]. Database Scalability, Elasticity, and Autonomy in the Cloud - Divyakant Agrawal, A. E., Department of Computer Science, University of California at Santa Barbara.

[7]Cloud Computing: Principles, Systems and Applications - Gillam, N. A., Springer 2010

[8]http://relationalcloud.com/index.php?title=Database_as_a_Service

[9]. The multitenant, metadata-driven architecture of Database.com - Database.com Getting Started Series White Paper

[10] Megastore: Providing Scalable, Highly Available Storage for Interactive Services - Jason Baker, C. B.-M. http://pdos.csail.mit.edu/6.824-2012/papers/jbaker-megastore.pdf

[11] Inside SQL Azure. Microsoft TechNet. http://social.technet.microsoft.com/wiki/contents/articles/1695.inside-windows-azure-sql-database.aspx

[12] https://www.windowsazure.com/en-us/home/features/data-management/

[13] https://www.windowsazure.com/en-us/pricing/details/#storage

[14] http://aws.amazon.com/rds/

[15] https://developers.google.com/appengine/docs

[16] http://en.wikipedia.org/wiki/Paxos_algorithm

[17] Werner Vogels' weblog on building scalable and robust distributed systems http://www. allthingsdistributed.com /2007 /10/amazons_dynamo.html

[18] http://aws.amazon.com/dynamodb/

[19]http://www.databasejournal.com/features/mssql/article.php/3823471/Cloud-Computing-with-Google dataStore.htm

[20] Google AppEngine Documents https://developers.google.com /appengine/ docs/java/ overview - Product page

[21] Google AppEngine Documents https://developers .google .com / appengine/docs/phyton/overview - Product page

[22]Google AppEngine Documents //https://developers.google.com/appengine /docs /python/datastore/gqlreference

[23] MongoDB - http://www.mongodb.org/ - Product Page12

[24] MongoDB blog: http://blog.mongodb.org – Product Blog

[25] Cloudant Blog http://blog.cloudant.com/cloudant-bigcouch-is-open-source - Product Blog

[26] http://bsonspec.org/

[27] http://wiki.apache.org/couchdb/ Product wiki

[28] http://www.mongolab.com – Product page

[29] Technical Overview: Anatomy of the Cloudant Data Layer Service - 2012 Cloudant, Inc.

[30]. http://bigcouch.cloudant.com/

[31]. Building Scalable Database Solution with SQL Azure - Introducing Federation in SQL Azure. http://blogs.msdn.com

[32]. http://aws.amazon.com/solutions/case-studies/airbnb/

[33]. https://www.windowsazure.com/en-us/home/case-studies/

[34]http://aws.amazon.com/dynamodb/testimonials/#imdb

[35] http://aws.amazon.com/solutions/case-studies/alexa/

[36] White Paper - Top Ten Data Management Trends - Scalability Experts - Raj Gill, Y. B.

[37]http://nosql.mypopescu.com/post/1669537044/sql-and-nosql-in-the-cloud
[38] White Paper - NOSQL for the Enterprise - Neo Technology (2011)
[39] White Paper - Database as a Cloud Service - Scalability Experts - Wolter, R. (2011)
[40]Master Thesis Electrical Engineering December 2012.
[41] "Windows Azure" [Online]. Available: http://www.microsoft.com/windowsazure/ [Accessed:12-March-2012]