Cloud Database Infrastructure: Database System Transference in Cloud Computing Management and Security

Mohd Muntjir[1], Mohd. Junedul haque[2]

[1]Department of Information Technology, College of Computers and Information Technology Taif University Saudi Arabia

[2]Department of Computer Science, College of Computers and Information Technology Taif University Saudi Arabia

***Abstract* -** *Relational database management systems endure the technology of choice for most applications. RDBMS vendors have new tricks up their sleeves to boost implementation. Relational databases have enjoyed a long run as the database backbone across a wide variety of productions, and for good intentions. NoSQL database technologies are challenging the domination of the relational database management systems. There are a number of NoSQL database preferences, all developed to accomplish particular purposes that RDBMSs aren't intended to handle. Many establishments are finding new liberty in having so many choices when it comes to database constructions. However, despite their recent enterprises and competence in managing large data sets, NoSQL databases aren't the right fit for all schemes -- and they aren't likely to expel relational software from the top database roost any time soon. Contingent on your business objectives, traditional databases, NoSQL databases or a hybrid of the two may be best to dispense the most assessment. A database is manageable to clients from the cloud and conveyed to users on request via the Internet from a cloud database provider's servers. Cloud databases can use cloud computing to attain optimized scaling, high accessibility, and multi-tenancy and actual resource provision. Databases advocating such applications require ACID (Atomicity, Consistency, Isolation and Durability) exclusive but these databases are hard to organize in the cloud computing system. The persistence of this paper is to review the necessities of the art in the cloud databases and multiple frameworks. It establishes how cloud database usually runs on a cloud-computing proposal, to approach the services specified by cloud computing system.*

Keywords - *Cloud Database, Database framework, Cloud computing systems*

I. INTRODUCTION

Cloud databases are on the rise as more and more businesses look to capitalize on the advantages of cloud computing to power their business applications. Pure cloud databases actually reside in the public cloud, and are usually available on a pay-per-usage basis, usually by the megabyte. Well-known cloud databases include Microsoft Azure Database and Amazon SimpleDB. Cloud-based databases often referred to as the Database as a Service (DbaaS) "enables organizations to enjoy significant deployment flexibility, hosted hardware and software infrastructure and utility-based pricing". The rise of cloud-friendly databases brings its own share of management and skills challenges. For example, there is the risk of raised expectations. Often, such databases may even be out of the reach of IT staff, even "removed one level from the programmer and DBA," says John Goodson, senior vice president for the product development group at Progress Software. The unique features of cloud databases (namely the ability to distribute data across wide geographical areas and among different servers in one physical data centre) are based on cloud computing technology made possible by virtualization, something relational database management systems (RDBMS) were not designed for. IT Manager Daily, part of the Catalyst Media Network, provides the latest IT and business technology news for IT professionals in the trenches of small-to-medium-sized businesses. Rather than simply regurgitating the day's headlines, IT Manager Daily delivers actionable insights, helping IT execs understand what technology trends mean to their business. [5]

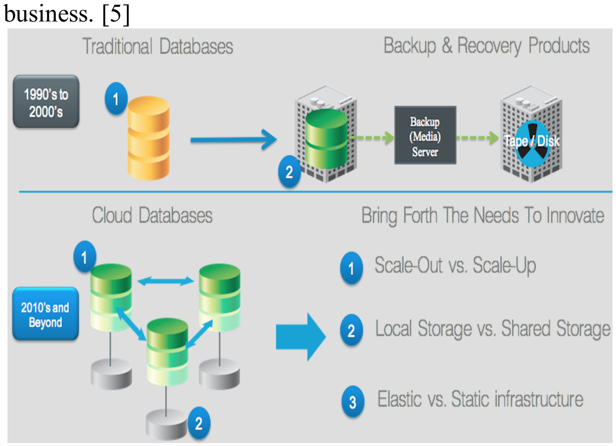

Fig. 1.Cloud Architecture Shift in Database

II. An Overview About Cloud Database

A cloud database is a database that classically runs on a cloud computing platform, get to it is provided as a overhaul. Database services take care of scalability and high availability of the database. This basically runs on a Cloud Computing platform, such as Windows Azure, Amazon EC2, GoGrid and Rackspace. There are two common deployment models: users can run databases on the cloud independently, using a virtual machine image, or they can purchase access to a database service, maintained by a Cloud Database provider. .Of the databases available on the Cloud, some are SQL-based and some use a NoSQL data model [6].Data storage is diverse at different remote locations in case of distributed databases. Dedicated servers are used to store these databases; therefore many servers are needed by companies to store their large databases. Those

Servers were static i.e. their location was fixed and the sites where data was distributed were fixed. Companies needed the infrastructure in the very beginning which costs a lot. Cloud Computing allows users to tap into a virtually unlimited pool of computing and storage resources over the Internet (the Cloud) [4]. Unlike traditional IT (Information Technology), Cloud users typically have little insight or control over the underlying infrastructure, and they must interact with the computing and storage resources via an Application Programming Interface (API) provided by the Cloud vendors. In exchange for those constraints, Cloud users benefit from utility-like costs, scalability, and reliability, as well as the ability to self-provision resources dynamically and pay only for what they use [7]. A cloud database is a database service built and accessed through a cloud platform. It serves many of the same functions as a traditional database with the added flexibility of cloud computing. Users install software on a cloud infrastructure to implement the database. A database service built and accessed through a cloud platform. Enables enterprise users to host databases without buying dedicated hardware. It can be managed by the user or offered as a service and managed by a provider. It can support SQL or NoSQL databases. Accessed through a web interface or vendor-provided API.Cloud databases can collect, deliver, replicate, and push to the edge all your data using the new hybrid cloud concept. Users no longer have to deploy the dependent middleware to deliver database requests anywhere in the world. They can connect applications directly to their database [32]. The rise of cloud computing and cloud data stores has been a precursor and facilitator to the emergence of big data. Cloud computing is the modification of computing time and data storage by means of standardized technologies. It has significant advantages over traditional physical deployments. However, cloud platforms come in several forms and sometimes have to be integrated with traditional architectures. [31] An enterprise-proven database cloud service that supports any size workload from dev/test to large scale production deployment. It is a Multi-layered, in depth security with encryption by default. Moreover a highly available and scalable service delivering speed, simplicity and flexibility for faster time to value and savings. Immense evolution in digital data, dynamic data storage needs, best broadband proficiency and Cloud computing manages to the development of cloud database systems [1]. In the Cloud terminology; Data as a service (DaaS), Cloud Storage, and Database as a service (DBaaS) are the different terms used for data executives in the Cloud Computing. They contrast on the basis of how data is stored, managed and executed.

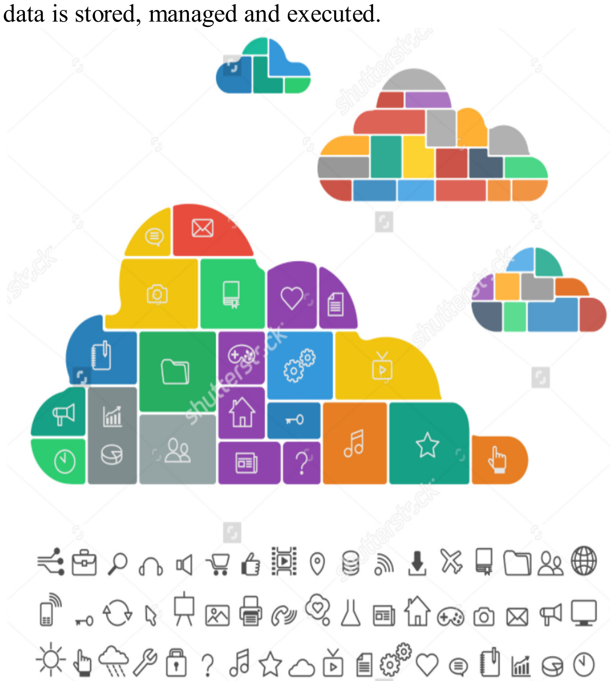

Fig. 2.Cloud Database

Cloud storage is virtual storage that permits users to store objects and documents. iCloud, Dropbox etc. are familiar cloud storage services [2]. Furthermore, DaaS permits users to store data at a remote disk accessible through Network. The basic use of DaaS is used for backup purposes and basic data management systems. DBaaS offers whole database management and permits users to access and store their database at remote disks anytime from any place through Network. Google's BigTable, Amazon RDS, Amazon's SimpleDB, Microsoft's SQL Azure and Yahoo's Sherpa Database are the commonly used databases in the Cloud Computing [3]. In Cloud Computing systems, Cloud database is a database expressed to users on demand through the Internet from a server provided by cloud database providers. Cloud databases support scalability and high availability, multi-tenancy and optimized resource allocation. A cloud database can be a conventional database such as SQL Server and MySQL. These traditional databases can be installed, configured and maintained by the user himself on a Cloud server. This preference is generally called the "Do-it-Yourself" (DIY) accession. Some Cloud service providers provide prefabricated database services such as Xeround's MySQL [4]. In "Do-it-Yourself" (DIY) approach, the developers manually assure reliability and flexibility services. Election of a DBaaS solution decreases the complexity and cost of running one's individual database in Cloud. Cloud databases contribute improved scalability, availability, performance and flexibility at lower prices. In a Cloud Computing, Cloud databases are able to support changing storage needs of Internet-savvy users who deal more with user created content, unstructured data, such as photos and documents. Shared-disk and Shared-nothing are two vitalized storage frameworks in database systems.

Cloud databases can offer important advantages over their traditional counterparts, including augmented convenience, usual failover and fast computerized recovery from failures, computerized on-the-go scaling, minimum investment and maintenance of in-house hardware, and potentially better performance. Moreover cloud databases have their share of potential drawbacks, including security and privacy issues as well as the potential loss of or inability to access critical data in the event of a disaster or bankruptcy of the cloud database service provider. Enough about what we think. Gartner's Critical Capabilities for Operational Database Management Systems evaluates vendors - from traditional relational behemoths to stylish NoSQL systems to upstart NewSQL vendors. Cloud applications connect to a database that is being run on the cloud and have unreliable degrees of efficiency. Some are physically configured, some are preconfigured, and some are inhabitant. Native cloud databases are usually better equipped and more stable that those that are customized to adjust to the cloud.

A decade ago an IT project or start-up that needed reliable and Internet connected computing resources had to rent or place physical hardware in one or several data centers. Today, anyone can rent computing time and storage of any size. The range starts with virtual machines barely powerful enough to serve web pages to the equivalent of a small supercomputer. Cloud services are mostly pay-as-you-go, which means for a few hundred dollars anyone can enjoy a few hours of supercomputer power. At the same time cloud services and resources are globally distributed. This setup ensures a high availability and durability unattainable by most but the largest organizations.

Experts suppose database-as-a-service (DBAAS), presently like all the other as-a-service options out there, to ultimately become the standard solution for all but the most highly sensitive and mission-critical data. A diversity of cloud database management systems are available to store and analyse both relational (SQL) and non-relational (NoSQL) types of data

Fig. 3.Cloud Database Migration

A. An Overview about the Shared-disk Database Architecture

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

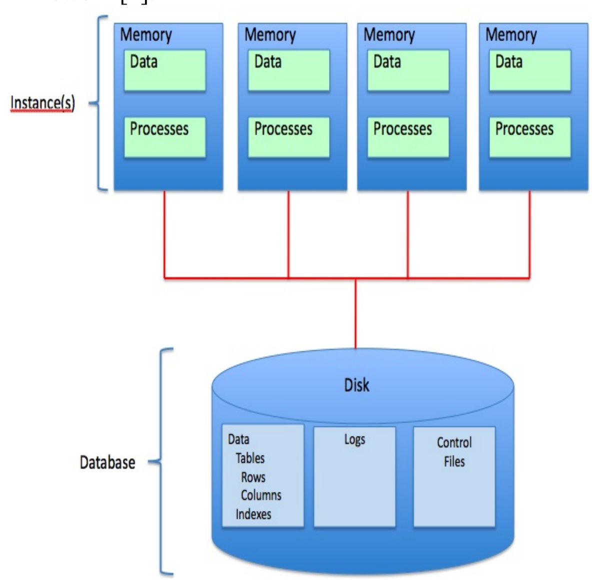

Fig. 4.Shared Disk Architecture

B. An Overview about the Shared-nothing Storage Architecture

Shared-nothing Storage architecture associates data partitioning which breaches the data into independent sets or autonomous posture. These data sets are actually located on various database servers. Furthermore, each server processes and manages its small part of the database particularly, which makes shared-nothing databases easily scalable and manageable. Whereas basic scalability and applications designed to work on shared-nothing storage framework are convenient for Cloud database. Although data partitioning used in this architecture does not work well with cloud system. It is not easy to visualize a shared-nothing database becomes very difficult and complex to manage due to data partitioning. Hence, it needs a piece of middleware to route database requests to the desired servers. Due to adding more servers, data has to be repartitioned. In this architecture, data partitioning should be done very carefully, neither data neither shipping nor attaching will become difficult and complex. More data shipping means more inactivity and network bandwidth congestion. These problems reduce database performance carelessly. Shared-nothing Storage framework is also used basically for data-intensive workloads. Amazon's SimpleDB, Yahoo's PNUTS, and Hadoop Distributed File System implement shared nothing architecture [5-7]. Oracle and IBM released their shared-nothing implementation of DB2 in September 2008 and 1990 properly for scalable analytical applications of data warehouses.

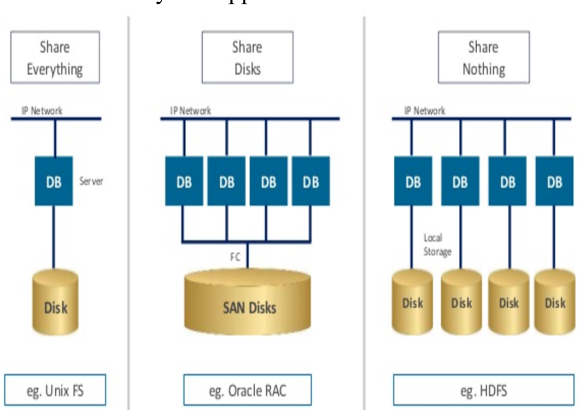

Fig. 5.Shared Nothing Architecture

III. An Augmented Analysis Of Relational Databases And NoSQL Databases

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

A. Introduction About The NoSQL Databases

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

Fig. 6. NoSQL Database

B. Introduction about the Relational Databases

The approach of relational databases is about forty years old. Relational Database worked best in the period of hardware limits such as slow processor speed and limited networking, little memory and small disk space. It has determined database framework based on relationships and schema, tables, indexes and columns.

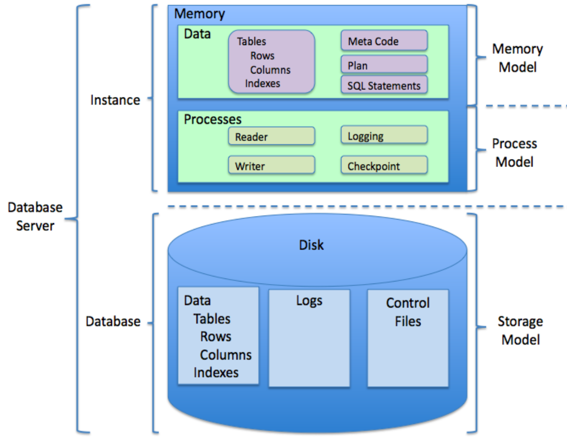

Fig. 7.RDBMS Architecture

In a Relational Database, Data is stored in tables with predefined complex relationships. Furthermore, Column indexes are used for rapid search. For this purpose, highly skilled Developers and DBAs are needed for database design maintenance and maintenance. Traditionally, they are used for transactional databases, and they integrate details at the lowest granularity and management. They incorporate sensitive and operational data such as credit card numbers to handle critical business operations, and employee data. These types of databases are not well appropriate for Cloud architecture and atmosphere as they do not support full satisfied data search and are more difficult to scale apart from limits [11-12].

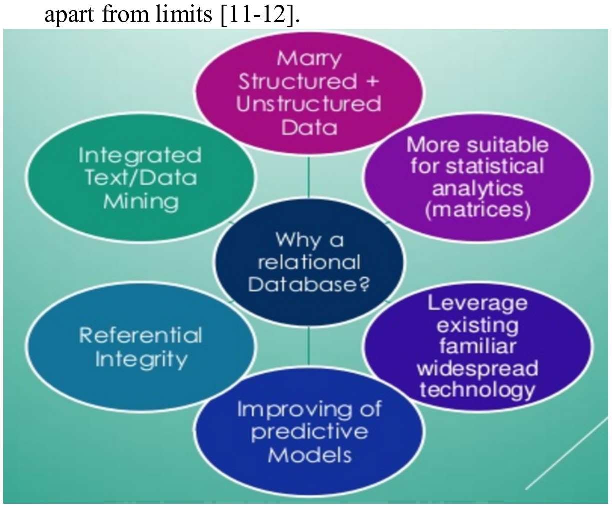

Fig. 8.Relational Database

IV. INCITEMENT TO DETERMINE CLOUD DATABASE SYSTEM

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

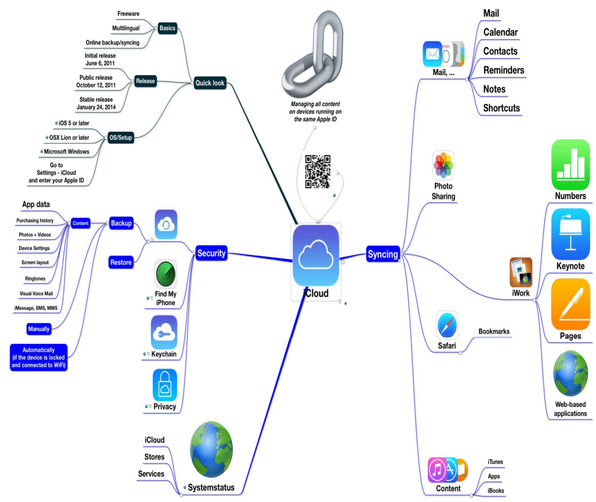

Fig. 9.Establishment of Cloud Systems

A. *Scalability in Cloud System*

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

B. *High availability and Fault Tolerance in Cloud Database*

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

C. *Heterogeneous Environment About Cloud Data base*

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

D. *Cloud Database Consistency and Integrity*

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

E. Simplified Query Interface in Cloud Database

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

F. Database Security and Privacy in Cloud Database

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

G. Data Portability and Interoperability in Cloud Database System

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

V. Innovative Progressing To Cloud Database System

Cloud databases are designed and created for low-cost assets hardware. They scale out conveniently by sharing the database across various hosts/nodes as the load accession. NoSQL databases have become analogue for cloud databases. Professional cloud storage needs to be highly available, highly durable, and has to scale from a few bytes to petabytes. Amazon's S3 cloud storage and Microsoft Azure Blob Storage are the most prominent solutions in the space. Ideally a cloud service provider offers Hadoop clusters that scale automatically with the demand of the customer. This provides maximum performance for large jobs and optimal savings when little and no processing is going on. Some basically used cloud databases in the corporations are defined below.

A. Description About Amazon Simple Storage Service (S3) and Databases

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

Fig. 10.Amazon Simple Storage Service (S3) And Databases

B. *An Overview About Amazon SimpleDB*

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

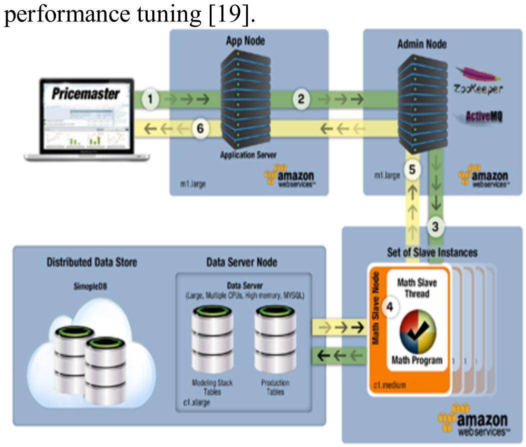

Fig. 11.Amazon SimpleDB

C. *About the Google Apps Bigtable*

It is a distributed or shared storage system based on GFS (Google File system) for structured data. BigTable was designed to support applications requiring massive scalability; from its first iteration, the technology was intended to be used with petabytes of data. Google BigTable serves as the database for applications such as the Google App Engine Data store, Google Personalized Search, Google Earth and Google Analytics. Google's thorough description of BigTable's inner workings has allowed other organizations and open source development teams to create BigTable derivatives, including the Apache HBase database. It appliance replicated shared-nothing databases. It has been clearly expanded in many Google products like Google application engine. It gives permission a more complex data store rather than SimpleDB. It allows entities and properties in compared to columns and tables. BigTable details in a technical paper presented at the USENIX Symposium on Operating Systems and Design Implementation in 2006. [34]

The Google Data store API also admits a get, put, delete format for obtaining data. It also permits a non-SQL language called GQL, which is not as constituent rich as SQL (Structured Query Language). In GQL, Select statements can be oppressed on one table only. Furthermore, GQL does not support the “Join” condition [20-21].

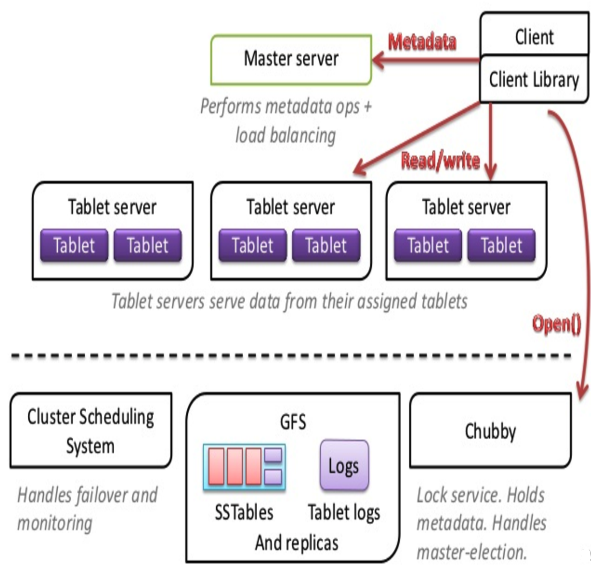

Fig. 12.Google Apps Big Table

D. *Description About the MapReduce*

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

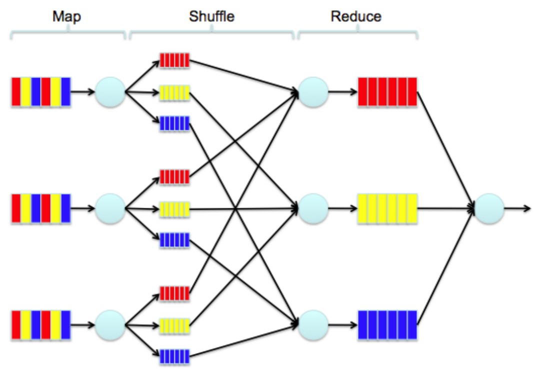

Fig. 13.MapReduce

E. Hadoop Overview

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

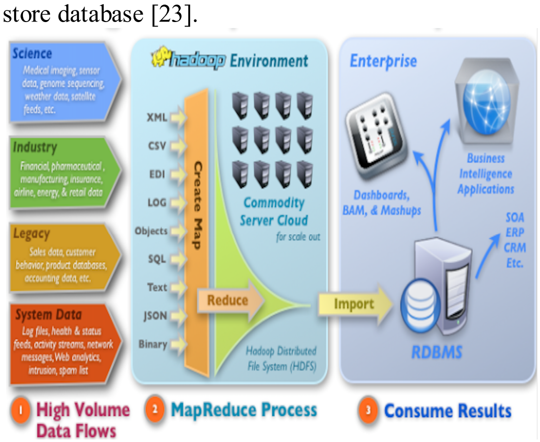

Fig. 14.Hadoop Data*base*

F. An Introduction About the Windows Azure Cloud Storage

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

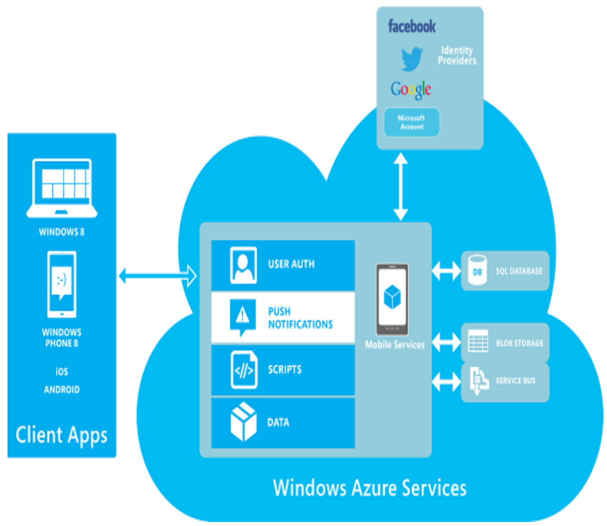

Fig. 15.Azure Architecture

G. About the Microsoft SQL Server Data Services (SDDS)

Microsoft SQL Server Data Services is a key/value data store, which is also considers the cloud extension of Microsoft's SQL Server. One of the concerns I read about is the impact on scalability. My observation is that when you look at most of the storage systems in the cloud, they don't have some magic formula for scalability, the trick is partitioning. Some systems are smarter than others in how they partition data and how dynamic the partitioning scheme is to adapt to varying system workloads. But in the end, you need to partition your data such that it's spread across a bunch of nodes; if across your system you never (or rarely) depend on cross-partition operations, then you have a sustainable scalability path. It associates with Microsoft's Sync Framework, which is a .NET library for synchronizing unique data source. It consists SOAP or REST APIs, schema free data storage, and a pay-as-you go payment systems [34]. SDDS has three core concepts: Entity, Authority and Container. The term Entity is a property bag of name and value pairs, and the term Container is a collection of entities. Further, Authority is a collection of containers and acts as a billing unit [25].

Fig. 16.Microsoft SQL Server Data Services

H. An Introduction About Sherpa

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

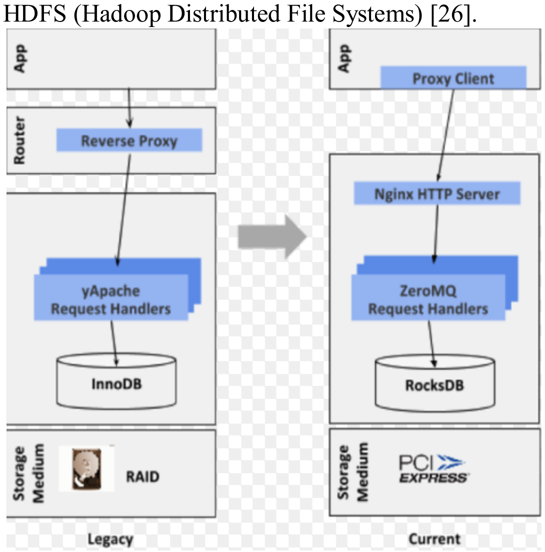

Fig. 17.Sherpa Database

I. Overview About Dynamo

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

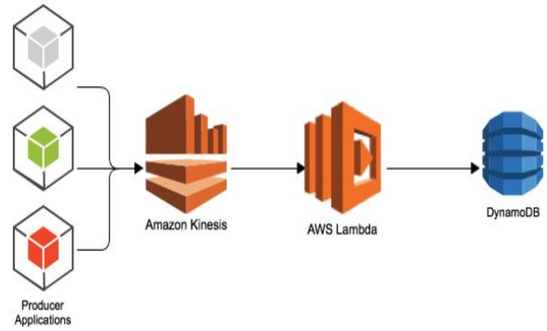

Fig. 18.Dynamo Database

J. *Description About MegaStore*

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

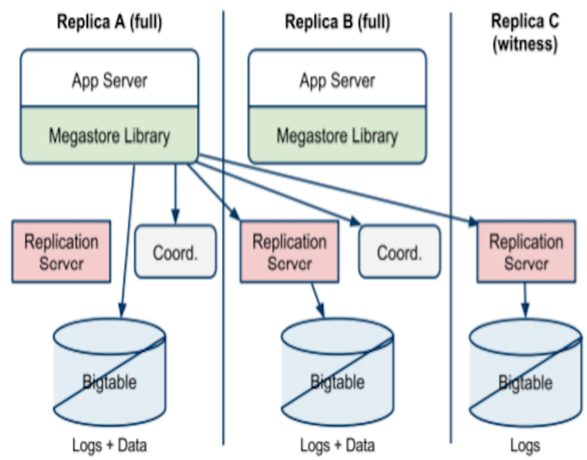

Fig. 19.MegaStore Database

K. An Overview About CouchDB

CouchDB is an open-source, free, Apache project since early 2008.CouchDB is a document-oriented database, basically written in Erlang. Further, it belongs to NoSQL generation of database. Documents (records) are stored in JSON (Java Script Object Notation) format and are controlled through an HTTP interface. It also allows "views" to be dynamically created using JavaScript. These created views map the document data onto a table-like structure that can be queried and indexed. CouchDB does not support a non-procedural query language. It accomplishes scalability through asynchronous replications. It has unique ability to serve as a self-contained application server and databases [29].

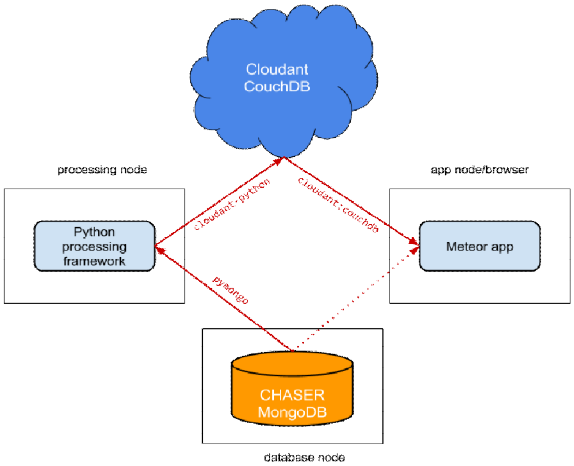

Fig. 20.CouchDB Architecture

L. An Introduction to MongoDB

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

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

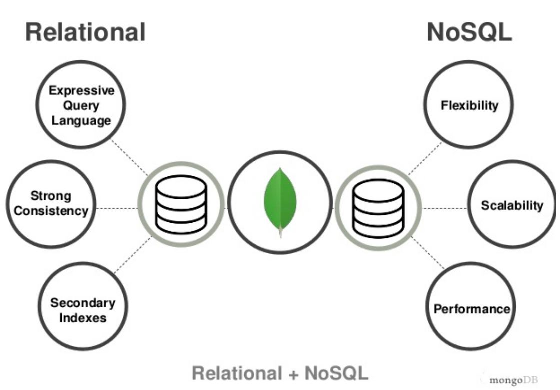

Fig. 21.MongoDB

VI. Summary About Cloud Database Security System

A. Generally approved Security Observes and Viewpoints

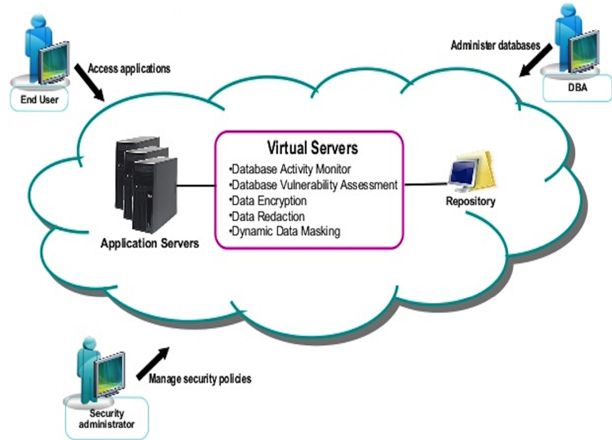

Fig. 22.Database Security

1) Fundamentals of Minimum Privileges

The principle of least privilege is the practice of limiting access to the minimal level, yet still allowing the application to function normally. For example, application owners and administrators should have access only to the data, applications, and systems and privileges necessary to perform their duties. This approach provides better stability and more predictable behaviour; e.g., unauthorized users cannot purposely or accidently remove privileged files or stop critical processes. Additionally, least privileges also improves mean time to deploy applications, since fewer privileges or roles need to be implemented. However, least privilege principle is one of the most difficult philosophies to implement. Organizations must have knowledge of the following to successfully implement least privilege principle:

1. Organization of data
2. Comprehension of where their sensitive data resides
3. Solid computerization for user access life cycle management although these philosophies apply to general application deployment, least privilege principle becomes even more important and relevant in the Cloud, as tenants will require certain level of security isolation.

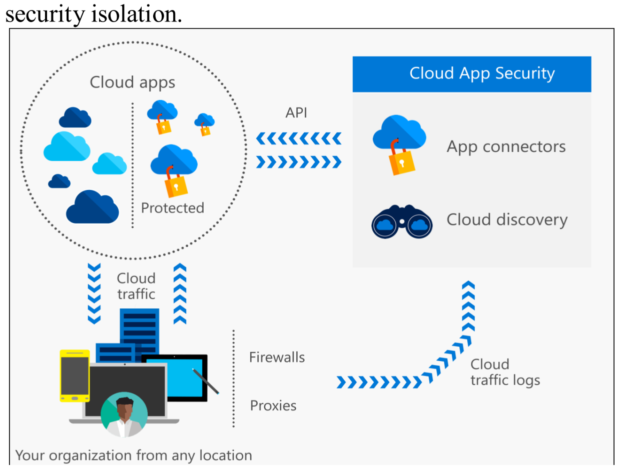

Fig. 23.Cloud Security

2) Protection in Comprehensive

Defence-in-depth is the practice of using controls at all layers of the information architecture. In Cloud architectures, the need to follow this philosophy is even more important. The areas of focus include hardware (physical access), operating system, hypervisor, virtual machines, storage, database, application servers, applications, networks, consumer portals, and the interfaces for Cloud automation and management. One of the primary architecture design goals should be to separate production from development environments. In a traditional architecture, this is done using distinct servers, storage systems, and network subnets. In Cloud architectures, using Cloud Pools is becoming an accepted method for achieving this goal. Cloud Pools can by separated by data sensitivity, line of business, and/or classification of users accessing it. A Cloud Pool is used to refer to a common set of resources shared by tenants. In this paper a Cloud Pool refers to an Oracle RACCluster. [33]

B. Private Database Cloud Security Issues and Threats

With the convergence of technology that represents current Cloud initiatives come different threats. Complicating this is the fact that traditional perimeters of the infrastructure change in a Cloud environment trust boundaries may be modified as well as the rate of change and level of scale one typically finds in a Cloud environment. Specifically, these contribute to the complexity of managing the Cloud infrastructure, which in and of itself may be considered a risk. We can state that the velocity of change in a Cloud is proportional to the velocity of attack, i.e., as the rate of change increases the potential for threats manifesting themselves also increases. Such threats are typically linked to the management of the Cloud infrastructure. Most importantly, these are all manageable with current security and governance processes and controls. The following is a sampling of potential threats that one may encounter in a Cloud environment.

1) Side Channel Intrusion

The multi-tenancy or proximity of systems in conjunction with an improperly secured Cloud environment may increase the risk to data or system corruption from adjacent systems and applications. Such threats can be addressed by utilizing a strong security architecture that implements a secure by default security policy, system hardening and strong compartmentalization.

2) Data Leakage

As with existing IT systems, there is always the possibility of data leaking or being disclosed between tenants running on a Cloud system. The remedy here is to ensure that appropriate controls are put in place, such as data encryption or access control mechanisms. In a Cloud environment, we could further state that data leakage may be possible by leaving data remnants in memory or on disk after a de-provisioned database service. One approach to addressing this threat is to ensure a consistent data scrubbing process is implemented. Likewise, a Cloud environment must ensure that no data leakage can occur between different tenants. Depending upon the criticality of the data, one practical method of protecting remnant information is to physically isolate the deployments and restrict access to those deployment environments where the remnants may reside. This is no different from non-Cloud architectures.

3) Attack Policy or Threat Intensification

An improperly secured Cloud environment can be a point from which various inter/intra systems attacks can be launched to either co-tenants or external systems, thereby causing a potential cascading failure of multiple systems that are co-tenants. Threat amplification means that a problem propagates faster and farther through a Cloud environment than it would under alternate circumstances. This also has the effect of potentially reducing a timely response to and recovery from the threat. This challenge can be addressed by ensuring that comprehensive and well-managed security and governance processes are in place to detect, manage and correct threats before they go viral and spread. [33]

4) Distributed Denial of Service (DDoS) Attacks

A denial of service (DoS) attack is an incident in which a user or application becomes unavailable or non-serviceable because it is deprived of the resources necessary operate. A distributed denial-of-service (DDoS) is a large DOS attack comprised of a large number of compromised systems attack a single target. A DOS or DDoS involves a loss of service or business function, such as customer facing websites, email, or networks. DDoS attacks may be launched against a Cloud environment that can result in the loss of access to or exhaustion of some resources or services so that the systems underperform or become unusable. Proper system resource monitoring and management, replication of systems and failover processes can address such threats. A common DOS attack is the Buffer Overflow Attack. Note that DoS attacks generally do not usually results in the theft or loss of business data; however, it can cost the companies businesses and credibility.

5) Complexity

Complexity is an inherent and potential threat in any computing environment. As complexity grows, so do the security risks: more components mean more attack surfaces and more interactions among components. When a system environment includes a variety of configuration and components (e.g., multiple O/S versions to maintain, multiple vendors to track, etc.), the management of the components is more difficult.

Oracle's Private Database Cloud methodology reduces complexity by emphasizing a rationalized, standardized environment. With less variety to manage, each component can be given more detailed attention. Complexity can be further addressed through enforcement of strong security policies and procedures, along with standardized processes for provisioning users into the Cloud and decommissioning environments.

C. Tenant Security in Private Database Cloud Systems

Security plays a critical role in shaping the details of Private Database Cloud architecture. Organizations must develop a clear understanding of security requirements, trust boundaries, and threat profiles. Database security includes a comprehensive approach for how, why, and by whom data is accessed. Further, organizations must understand how trust is established and propagated as well as how threats are mitigated across a Private Database Cloud environment. Organizations need to understand where and how security policy is defined, how access is managed, and how audit and compliance requirements are met given the distinct capabilities that providers and consumers have in a Private Database Cloud. Please review the following paper for details on the different Private Database Cloud Models. Databases are at the core of Cloud environments because they generally contain the most sensitive and important information for both providers and consumers. As such, it is at the centre of the defence-in-depth approach to security. If the database is locked down so only authorized access is permitted following least privilege, the risk of data loss or compromise is significantly reduced. Although this is not specific to Cloud environments, it becomes paramount in high consolidation density configurations, which is a typical configuration for Private Database Clouds. The application's security Service Level Agreement (SLA) must be well defined before being migrated or provisioned into the Cloud. A security policy-based template can be used so that a consistent deployment strategy is used for the appropriate level of security SLA. For example, an application may have stringent security requirements, whereby its data cannot be co-mingled with other applications. Multitenant Cloud security has always been a primary concern for any Cloud deployment. There are significant challenges that involve ensuring a good balance between maintaining corporate compliance and leveraging the cost benefits of a shared infrastructure model. Creating heavily isolated configurations for specific applications to meet compliance reduces operational efficiency and diminishes the overall value of a Cloud. In order to meet this challenge, the Cloud architecture must include, at design time, security policies for database provisioning, de-provisioning, and data access control.

1) Implementing Security in a Private Database Cloud System

Cloud providers must ensure that all aspects of security, such as infrastructure, O/S and database security are in place. The following table describes security scenarios and how each impacts Private Database Cloud deployments.[33]

VII. CONCLUSION

Huge data generated by web-based applications have changed the whole database aspects. Cloud databases appear to be a good solution for controlling such data. Furthermore, all enterprises cannot afford to arrange more expensive data centre framework for managing and maintain their own databases. The emerging popularity of Cloud databases is making the beginning of new era of database systems. Although cloud databases are not ACID flexible, they are able to control huge workloads of web-based databases, which do not need such guarantees. Various Cloud databases are available in the enterprises. But each has its particular and unique API, data model and database functions, query interface. This approach required to be standardized for their better enhancements. Cloud computing and Cloud databases are set to rule the next decade by reducing the limitations they have. Furthermore, Asian nations are investing in cloud centres of economic development and are acquiring cloud services for their own businesses [34].

ACKNOWLEDGMENT

We would like to gratefully and sincerely thank to The Dean and Vice Dean of our College, Chairman of our Department for his guidance, understanding, patience, and most importantly, his friendly nature during this research paper. We would also like to thank my friends and colleagues, who provided us an efficient support to work on this atmosphere and good infrastructure. We would also like to thank to all the previous researchers who worked very hard and helped others to comprehend the subject of Database and Cloud Systems.

REFERENCES

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

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

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

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

[5] Database Management in the Cloud Computing Era,IT Manager Daily,http://www.itmanagerdaily.com/database-management/

[6] Kaushik Donkena Subbarayudu Gannamani," Performance Evaluation of Cloud Database and Traditional Database in terms of Response Time while Retrieving the Data ", Master Thesis Electrical Engineering December 2012, https://www.divaportal.org/smash/get/diva2:829184/FULLTEXT01.pdf
[7] Arpita Mathur, Mridul Mathur, Pallavi Upadhyay ," Cloud Based Distributed Databases: The Future Ahead ", International Journal on Computer Science and Engineering (IJCSE), http://www.enggjournals.com/ijcse/doc/IJCSE11-03-06-115.pdf
[8] Donald Kossmann, Tim Kraska, Simon Loesing, "An Evaluation of Alternative Architectures for Transaction Processing in the Cloud", SIGMOD'10, June 2010.
[9] Daniel J. Abadi et al., "Column-oriented Database Systems", VLDB '09.
[10] Stonebraker, et al., "C-Store: A Column-oriented DBMS".
[11] Thakur Ramjiram Singh, "Cloud Computing: An Analysis", International Journal of Enterprise Computing and Business Systems ", Vol. 1, issue 2, July 2011, pp. 2230-8849.
[12] Rick Cattell, "Scalable SQL and NoSQL Data Stores", ACM SIGMOD, Vol. 39, Issue 4, 2011, pp. 12-27.
[13] Arpita Mathur et al., "Cloud Based Distributed Databases: The Future Ahead", International Journal on Computer Science and Engineering (IJCSE) Vol. 3, No. 6, 2011.
[14] Bo Peng, "Implementation Issues of A Cloud Computing Platform", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering.
[15] Mihaela Ion, "Enforcing multi-user access policies to encrypted cloud databases", IEEE International Symposium on Policies for Distributed Systems and Networks, 2011, pp.175-177.
[16] Maggiani, R. "Cloud computing is changing how we communicate", IPCC 2009, 2009, pp. 1-4.
[17] http://aws.amazon.com/rds/S3 last accessed on May 24, 2012.
[18] http://aws.amazon.com/rds/ last accessed on May 24, 2012.
[19] http://aws.amazon.com/simpledb last accessed on May 25, 2012.
[20] S. Ghemawat et al., "The Google File System", inproceeding of 19th ACM Symp. Operating System Principles (SOSP 03), ACM Press, 2003, pp. 29–43.
[21] F. Chang et al., "Bigtable: A Distributed Storage System for Structured Data", in 7th UsenixSymp. Operating Systems Design and Implementation (OSDI 06), Usenix Assoc., 2006, pp. 205–218.
[22] Dawei Jiang et al., "MAP-JOIN-REDUCE: Toward Scalable and Efficient Data Analysis on Large Clusters", IEEE Transactions on Knowledge and Data Engineering,Vol. 23, No. 9, 2011.
[23] D. Borthakur, "The Hadoop Distributed File System: Architecture and Design, Apache Software Foundation", http://hadoop.apache.org/core/docs/r0.16.4/hdfs_design.html last accessed on May 27, 2012.
[24] Troy Davis, "Cloud Computing Use Cases and Considerations", http://digissance.com/ Cloud Computing Talk.pdf last accessed on June 10, 2012.
[25] Www.windowsazure.com/enus/develop/net/.../c loud storage/last accessed on June 10, 2012.
[26] Brian Cooper et al., "Building a Cloud for Yahoo", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2009.
[27] Giuseppe De Candia et al., "Dynamo: Amazon's Highly Available Key-value Store", in of 21st ACM Symposium onOperating System Principles, SOSP 2007, pp. 205-220.
[28] Jason Baker et al., "Megastore: Providing Scalable, Highly Available Storage for Interactive Services", in 5th Biennial Conference on Innovative Data Systems Research (CIDR '11), 2011, pp.223-234.
[29] http://www.couchbase.com/couchdb last accessed onMay31, 2012.
[30] http://www.mongodb.org last accessed on May31, 2012
[31] https://www.qubole.com/resources/article/big-data-cloud-database-computing/#sthash.lxMBnvaL.dpuf
[32] https://www.ibm.com/cloud-computing/learn-more/what-is-cloud-database/
[33] Security in Private Database Clouds, http://www.oracle.com/technetwork/database/database-cloud/security-in-private-db-clouds-1733933.pdf, An Oracle White Paper July 2012
[34] Mohd Muntjir, Mohd Asadullah, Shabbir Hassan, Altaf Hussain,
"Cloud database systems: A model conversion in databases", Proceedings of 2nd International Conference on Information Technology and Electronic Commerce (ICITEC), 20 December 2014,
pp- 280-285

Author's Info

Mohd Muntjir is working in College of Computers and Information Technology Taif University at Taif Saudi Arabia. He received his M.C.A. degree from H.N.B. Garhwal University Uttarakhand India and Ph.D. degree in Computer Science from OPJS University, Rajasthan India. He is a member of professional societies like ACM, IEEE, VAS, IJETAE, and CSTA. His interested are mainly in Wireless Sensor Networks, Cloud Computing Systems, and Database Management Systems. The author has published many research papers in distinctive journals, conferences and books/book Chapters.

Mohd Junedul Haque is working in College of Computers and Information Technology at Taif University Saudi Arabia. He obtained his Ph.D. degree in Computer Science and Engineering from OPJS University Rajasthan India and M. Tech. degree in Information Technology from Guru Gobind Singh Indraprastha University Delhi India. His areas of interests are Multimedia, Database and Image Processing. The author has published many research papers in distinctive journals, conferences and books/book Chapters.