CHAPTER -3 : ANALYSIS OF CLOUD DATABASES & RELATED TECHNOLOGIES

3.1 CLOUD DATABASE

“Pay as you go” is a trend followed in today’s cloud computing environment. It means users just have to pay only for the services used. Nowadays cloud database services are emerging as one of the most important services among cloud computing [128]. When large distance has to be covered and data has to be transferred per terabyte, it costs much. That is why cloud database become popular as it provides numerous advantages.

Cloud Database provides low and cheap cost availability to customers. Because of cloud computing characteristic features, for new applications cloud computing provides best and economical way to set up and help many small scale business to come up which were previously not able to come up due to financial problem in old traditional method [129]

3.2 CLOUD DATABASE AS A SERVICE

Database Management System is an application which provides the creation, management of the database. Professionals can enter new records, update, modify and delete data whenever required. Users can handle all tasks related to data, also insertion and modification. Reports, query, and tables are all parts of database handling software called database management software.
The concept of cloud computing in olden times was being composed of five characteristics, four deployment models and three service models but in today’s scenario, it is being extended and more categories have been added like Storage-as-a-Service, Security-as-a-Service and Database-as-a-Service [130]. In recent times, an important part of the cloud is a scalable database which comprises updating of work and decision system.

The way the data was being saved and retrieved is being changed with the emergence of cloud database. Given a cost price, servers and applications like database are provided as cloud services. In changing trend, many technical advantages like no need to spend money in physical setup of database offices and handling tasks because resources are available as services online. Now the requirement of having data centers is finished as cloud database provide service easily. Users may also make own applications which are possible because of resources being provided by PaaS model to cloud databases. As three decades have passed, research work is to deal with a cloud database management system.

### 3.3 CLOUD Database – Requirements of cloud database management in cloud

Whether an organization has been taken, a research center, or any education center it remains the primary needs to process data fast and in an efficient manner. For this purpose, institutes plan and manage database processing tasks so as to handle work of database such as installing, maintaining tasks. Database related tedious tasks are very much difficult to manage and a small problem left will turn out complicated, rather than buying hardware and products related to database, connecting the networks along with
employing experienced persons it will be better to make use of cloud database services because of inexpensiveness and easier and effective management of database related tasks.

Financial investment tends not to change as different hardware, software and networks costs may decrease but costs of people managing such complicated tasks continue to increase. It may happen that professionals costs will dominate over the solution to database handling costs [131].

For having an arrangement of data, the database needs to be restructured, makeup with some alterations, restored, take a backup, to make adjustments for spacing. There is not much development in the area when there no impact on the accessibility of database solution, and still movement from one database to next becomes possible [132]. Because moving up from one database to next makes some components unavailable to use.

Locations where computers and related components are being housed or data centers more precisely, are using database management systems. But much earlier it was up to the developers to select the type of database for the cloud and perform the installing and managing tasks. They also need to deal with the complicated administration tasks on their own [133]. Positive aspects are full control is provided as a selection of own database and its dealing can be done.

PaaS Service dealers now provide up with database services on a cloud in order to reduce the workload of cloud customers and the work is taken over by cloud provider who takes care of controlling everything starting from the handling of logs to recovering and backing up of all records. The developer needs to deal up to table and query
maintenance. Sometimes there is an organization which takes up tasks of database services.

For the creation, storage and management of database tasks there are many database service providers. Rather than using organization computing infrastructure, customers can access data using hardware and software by the service provider. Now positive aspects are many, like it may happen that at database service providers side there may be a changing in software, hardware or networking side or there may be a failure but the customers will have no effect on the problem because it will be dealt with database service providers side. Generally, the organization just takes into use system for database services whose administration is done by service providers. Now there is no need for buying, modifying, updating and performing such tasks related to the database by cloud customers.

Whether it is financial, business, or any internet related work, database is needed everywhere for some or the other purpose. The old method, which is known as a traditional database system which was much in use. Several disadvantages of this database systems are –

1) Difficult maintenance works

2) Scalability and configuration problems

3) Complication in choosing from the available systems

4) Cost price of some systems much more than expected
So these disadvantages are tackled up by the cloud-based Database as a service. In today's scenario, any DBaaS features cannot find out that is able to cope up with all the disadvantages. So there will be next boom of an evolution of database [134].

### 3.4 DEPLOYMENT OF DBaaS

So as to provide customers the advantages of database features, the Database-as-a-Service (DBaaS) is a structural and functional means to provide the capability to serve a purpose. A company’s requirements can be fulfilled with the cloud database and the functionalities that are being provided, for this there are two use-case scenarios-

1) If towards private cloud, individual databases of a big company need to be shifted.

2) For the purpose of data management of small and medium companies the data handling charge is given to public cloud provider who takes care of many small and medium businesses.

There are many initial requirements which a DBaaS should follow, which are as follows-

a) Along with the addition of some new functions like time travel, snapshot and analytics, also the developer is made free from handling the administrative tasks related to database management and maintenance works. The performance of cloud database tasks is provided with fault tolerance, availability and high-performance skills.

b) If there is a change in workloads then the ability by which dynamically there can be adjustments according to change in workloads is called elasticity. Cloud service
providers administrative, power, infrastructure expenses should be reduced and
elasticity should be regulated to meet Service Level Agreement (SLAs).

c) The user should pay as per usage, and information security should be guaranteed.

IT professionals are able to manage database, servers and storage tasks to shared cloud
hardware and software, like this the private cloud gives database services efficiently.
Many functionalities which are given to cloud database of private clouds are pay as per
usage or metered service, features related to self-service, elasticity along with a decrease
in cost, but increase in quality of services.

Public clouds have many disadvantages because there are many risks associated with
them and service level agreements are also less available. So private clouds are
considered better than public clouds. Features like availability and performance are
offered by private clouds to the IT managers so that more policies and government
regulations can be implemented.

3.5 ARCHITECTURE MODELS FOR DATA MANAGEMENT IN CLOUD

As the clouds are much in demands so as the databases. For the complicated tasks like
data management, there have been continuous works on data sharing architectural model,
so as to suit the needs of data handling tasks which are changing day by day. So the
upcoming works on cloud data management architectures are:

a) Multiple copies of cloud database are created which are called Data Replication.
   There is one master copy and rest are simple read-only copies, the difference is
   that in a master copy updates will be done and that are transferred to multiple
copies. So the other copies serve as read-only files and the consistency is preserved.

b) Data which is regularly or most frequently accessed is added with a feature of memory caching.

c) Olden times where the traditional architectures had followed up "Shared Everything Scale-up" but the changing times have given the architectures "Shared Nothing Scale-Out". Here independent nodes serve as main blocks of building up the databases information also maintenance, data access and replication is possible in shared-nothing architecture.

3.5.1 Large Databases Based on Multitenancy

Tenants or the multiple client associations are served by the single instance of software architecture which runs on a server and the process called multi-tenancy as described by Wikipedia. According to the requirements of various customers, the need arises that different rules, interface designs and business processes should be set up as well as data of individual user should be protected and privacy and security should be maintained.

Same database tables are being shared by many tenants and here traditional database multitenancy is considered as for SaaS. But there are many cloud scenarios where different models of multitenancy are considered [135].

a) Virtual Machine Based: In virtual machine based, the same physical server is being shared by many virtual machines and as part of Virtual Assembly deployment of the database is done. The maximum level of isolation is provided
at OS level. With the help of IaaS( Infrastructure as a Service), this option is being possible.

**Shared Cluster:** In the DBaaS Self Service Portal, this can be done. Installation of database software and grid infrastructure is done previously. On top of that cloud, a database is set up.

**Shared Installation:** With the help of DBaaS Self Service Portal, the option is available. The deployment of the database is done on existing installation as a single instance of a database.

**Shared Database:** On the existing database, the database service is being deployed. The assumption which is generally taken is that the different services are being used by the consumers of the database in order to access the database and for charges also.

On the basis of subscriber model, multitenancy is being set up. So a user can take the facility according to requirements. **Multi-tenant databases** have many advantages, which are:

**Dedicated Database:** These are ‘Shared Nothing’ kind of architectures. Dedicated databases have separate databases with every tenant.

**Dedicated table & different schema:** This kind of multitenant databases have separate schema and databases are shared.

**Share table/schema:** These are ‘Share Everything’ kind of architectures. Here tables and schema remain shared or the table and databases remain same.
Though different research works are going around but still one perfect solution for cloud database management is needed to be developed.

### 3.6 PRINCIPLES OF ARCHITECTURE

For cloud DBaaS architecture, a planning to have an agile capacity should be made in organizations [136]. As per cloud Databases, there are varieties of models available but need is to select the right size model which is fit for requirements of the business. Also, the Cloud Database models should follow principles of DBaaS Architecture.

Few of DBaaS architecture principles are as follows-

**Alignment of Business:** The enterprises have business operating model and DBaaS architecture should be in alignment to that.

**Tracking of Values:** There are some prescribed key value indicators (KVIs) and they help in measuring the values of DBaaS architectures.

**Standardized Infrastructure:** Some standard technologies and assets must be taken up for database architecture.

**Ongoing Improvement in Architecture:** There must be continuous improvements done where required.

**Cost Competitiveness:** Cost of the database must not be too much high. Along with cloud providers database services charges will be in competition.

**Provider cooperation among Consumer:** With the cooperation of users, services of a database, types of attributes and the service quality is maintained.
3.7 CHALLENGES OF DBaaS

Service operators are given the work of scaling, backup, keeping privacy, configuration and controlling the access from users such that operation workload is reduced and also overall costs is being reduced. Now the cloud database providers now face challenges of privacy and security of database, elasticity and scalability, and multitenancy [137].

At cloud providers, users data needs to be placed at the premise in the service provider model. Data is the most crucial asset for any company or organization. For the requirements of privacy and security, service providers have to deal with. There are limitations with cloud database such as at the time of disaster, there may be a loss, inability to access data, also the privacy issues related to information security.

Having a user interface which is easy to use and also which is appropriate is also a challenge. The interface should be powerful, also applications building up should be easily allowed.

3.8 EVOLUTION OF DATABASE TECHNOLOGY

In olden times from flat files, new databases such as hierarchical and network evolved. After that, relational databases came into a picture where data can be stored as well as retrieved using SQL, also transactional processing is done. After the traditional relational databases (RDBMS), there was big data explosion and a need arises for more alternatives as such web features were evolving. Many RDBMS products came such as Oracle, MySQL, Microsoft SQL Server, IBM's DB2 etc.
Due to OOPs or the object oriented programming evolved, databases were linked to OOPs and the new release was Object Oriented Database management Systems (OODBMS). Here objects which are matched with programming are the persistent objects. Because of sensing data and focusing on addressing devices and OODBMS were not successful. There are many types of data with different processing requirements such as web pages, videos, messages, tweets etc [138].

Vertica is a kind of columnar database which provides state-of-art analysis abilities. Database research project named CStore is a type of column-oriented database on which Vertica is based [139]. Sometimes data is stored in form of rows on disk but here this is not followed, here data is being set up as values of columns from the same attribute. Now features that access columns set more, like a data warehouse and analytical kinds of retrieval of data, generally need more I/O but here large I/O are saved. Databases which are columnar follow ACID property, are relational and support SQL too. The different kinds of requirements of data types of data handling works cannot be performed by old traditional database management systems so need arises for new developments. Cloud Computing and Big Data have evolved out new database handling needs [140].

3.9 CDBaaS SECURITY RELATED TECHNOLOGIES AND OUTCOME OF ANALYSIS

3.9.1 Fully Homomorphic Encryption scheme

C. Gentry proposed Fully Homomorphic Encryption scheme [141]. In this scheme, there is no need for decryption in order to do any processing of any function on encrypted data, so this method allows computation over data which is in encrypted form. Before
data is stored at the server, using Fully Homomorphic Encryption, data is encrypted. Whenever there is a need for query execution, an equivalent query is developed over encrypted data. So processing work is now performed over encrypted data as any valid SQL query which is being sent to a server and processed and returns result (encrypted) to a client. Here it is decrypted back to get an answer. This is a good approach but processing work may be slow [142].

3.9.2 Another kind of method which can be used is to make the client as encrypted data storage. Data gets stored at the server but before that, it is encrypted by a client. So for query processing work, the needed database part is converted to decrypted form and shifted on the client machine. Now database which was previously encrypted is now converted to plain text database on which query can be processed upon to get output. This is useful technology but because of shifting of the needed relations in databases, there is a huge load on working as traffic gets increased. So issues related to performance are generated. Maintaining of own databases servers at client side becomes necessary here but it hides the DAAS model benefits.

There are big industries and companies where lakhs of people work and lakhs of people are associated with customers for services. There is a large amount of information related to this, which is critically necessary for the execution of all works. But leakage of this data is also possible, for this maintaining security becomes an everyday necessity. As query execution at server successfully along with maintaining privacy is becoming the real requirement. This gave a direction to our work.
Some related techniques related to cloud databases taking the above requirement can be classified into software based and hardware based techniques.

### 3.9.3 CryptDB

Popa et al. gave the approach CryptDB [143][144]. It is the software based approach. Here encryption of each attribute is performed in separation. Schemes used are as follows:

1) **RND** – This is randomized scheme for encryption which not includes ciphertext operation

2) **DET** – This is a deterministic scheme for encryption which works as equality predicates.

3) **OPE** – This is a type of encryption scheme which maintains order preserving technique.

4) **JOIN** – This encryption scheme gives support to equijoins on cipher text.

5) **SEARCH** – Word search technique is used by this search scheme.

The main theme on which this Crypt DB was built is “onions". Like an onion is made up of a number of layers similarly here for a single data value many layers of encryption scheme are made.

But this approach suffers from performance issues. As there are many layers of encryption for security but for each and every query all the layers of encryption have to
be worked upon. So the execution becomes slower. Especially when it is the first time of execution, the layer of encryption has to be removed so it takes much longer time. The next execution takes less time but first time takes the most time and reduces performance.

Though there is no need to maintain databases at client side in CryptDB but it supports very limited SQL constructs. For the practical usage in big industries there is a need for a cloud database model which can maintain privacy for many records and with very limited SQL constructs it is not possible, so it is not much in use.

3.9.4 Bucketization Approach for Databases

Hacigumus et al. [145] gave this software based approach. Here partitions are created out of the domain of column values by splitting. A unique id is given to every partition created. On the server, in the form of plain text, the partition id gets stored. Encryption is provided at the row level, which means as for the whole row, encryption is performed as a single data value.

The query processing work is as follows. First of all, the query is being submitted by the user. At the server, the first level of filtering is done at the server with the help of partition ids. The output which consists of rows which pass filter is shifted towards the client. To make a plain text database out of encrypted databases, decryption is done.

The final query processing is also done on the so-called database which is made. It depends upon partitioning that how many rows are getting transferred to the client. All
the records get transferred to the client if the worst case has to be taken. Suppose if the query taken is "Select avg(salary) from management_table". Then, in bucketization technique, for such queries, full encrypted relation needs to be brought to the client.

The need of database engine at client side is the main limitation. If database engine is maintained at server side, the output may have false positives. So client needs to do filtration of the output. So the need of database engine at client don't prove much worthy and limits the benefits of DAAS model.

3.9.5 Chip-Secured Data Access

**Bouganim and Pucheral** [146] introduced this chip based technique for secure database technique. It is hardware based approach and it uses the smart card. Here smart card is being called as tamper-proof hardware. Installation of the smart card is being done at the server by the client. Work related to data processing is done inside smart card. Both database engine and encryption engine are present in the smart card. Storage of encrypted data is being done at the server. Scalability remains the problem with this technique. This scheme faces issues due to limitations in processing power and memory.

3.9.6 GhostDB

Anciaux et al. [147] gave this technique. This is also based on the use of the smart card. Here query process execution is as follows. Storage is done at the server. Division of data into parts is being done as private and public parts. At the server, private data is
being stored in encrypted form, also in plain text form, public data is also kept. On the smart card, private data processing is performed and on the server, public data processing is being done. Again here too as the smart card is not too scalable that is why it counts here the limitations.

3.9.7 SCALEDB

SCALEDB [148] is being proposed by Ylber Januzaj, Jaumin Ajdari, Besnik Selimi. SCALEDB is a cloud DBMS model in which it serves as a storage engine and is capable of being used with application MySQL. Working process of SCALEDB is explained here. There are nodes in SCALEDB which are clustered or grouped up. The whole database can be accessed by the single node. To enhance performance or scalability, and also if the load is increased, more additional nodes can be added.

Its architecture is composed of three layers. Storage layer, Database layer, and Application layer are first, second and third layer of this architecture. First layer - The main place for storage is the first layer or cache layer which may be in the cloud or a local area. The specialty of this layer is mirroring techniques. Data can also be in the second drive if the first drive crashes.

Second layer – Also known as database layer, it contains multiple nodes, where each node can access the entire database. Also as the layer is located between storage and application, so it provides splitter between them. Sharing of information is done between nodes. Sometimes it may happen that there is an enhancement in database usage, such situations are manageable by new nodes addition.
Also known as the Application layer, the third layer may serve as application storage. In this layer, parallelization software is present. Few of the terms associated may be a load balancer, application server, or web server with this layer. Some drawbacks can be listed as it is not a type of open source application, distributed transaction and stored procedures are not supported by this technique [149].

3.9.8 Need for Hybrid Solution

First of all, small discussion about Map-Reduce Software. For the purpose of large scale data analysis work automation, many softwares like Map-Reduce have been designed. Some of its special features are: Fault Tolerance, Managing efficiently in heterogeneous environment, Good performance and Encrypted data operation.

Though map-reduce and also parallel databases have good features, but still for cloud there is search for better solutions for data analysis. There is demand of a hybrid solution so as to meet desired properties and with efficiency. So as to have some a hybrid solution there is a need for solution which could integrate fault tolerance, many features of Map-Reduce, effective, tool plugability, and increase performance.

3.10 Cloud Architectures [150][151](Gelogo, Y.E., Lee, 2012)

3.10.1 Parallel databases can be of following types

a) Multiprocessor architecture – Three architecture provides the basis for working which are

   1. shared memory,
2. shared disk and

3. shared-nothing architectures.

b) Hybrid architecture – Basis of the working is given by two architectures which are

1. NUMA or non-uniform memory architecture

2. Cluster (shared nothing and shared disk architecture)

Now, these architectures are discussed.

**3.10.1.1 Shared Memory Architecture**

An architecture which on the cloud platform provides access to many CPUs which helps in between communication and also in removing unusable REDUNDANT copies. On single processor /multiple processors, many programs can run simultaneously. Shared memory architecture can be shown diagrammatically.
3.10.1.2 Shared Disk Architecture

As the name depicts, communication in the network is possible because of sharing of disks to all processors. The CPU controls the access of memory by the each processor in this kind of architecture. With low cost, load balancing is provided, and from centralized form easy migration is possible. These are some of advantages. Shared Disk Architecture can be shown in diagram form.
3.10.1.3 Shared Nothing architectures

Shared nothing architecture is also called “SN”. So shared nothing architecture or SN, if described in terms of "SOA" or Service Oriented Architectures, there is no sharing of disks or memory here but each service has an individual database and access to it is possible with only that associated service. But if “SN” or shared nothing architecture is described in terms of hardware then "SN" can be called as a machine by which local disk or memory are being used.
3.10.2 CDBMS in Association With Nodes

User works on cloud database from computer with the help of internet. Many nodes are used in cloud database. Here the preferred communications are peer-to-peer. A query given by user can be managed by single node. In cloud database management system, each node contains the map which relates to data stored. Node working is as follows. In order to have data, one should obtain or retrieve data directly from database, or database can be replicated.
3.10.3 Databases-as-a-Service in form of Relational Databases

Relational Cloud’s features comprises of 1) In order to achieve better efficiency in working more than present approaches this is better for multitenant databases.

2) Graph based algorithm used for partitioning in order to achieve workloads for complex issues.

3) SQL Queries can run safely and perform all operations even when encrypted data is to be handled.

DBMS engines are used as Relational Cloud in order to run query processing. Only one database server runs by back-end node. One or more databases can be load by a tenant. There are many tables in a database and also a workload associated with it along with queries and transactions.

Many tenants data is not mixed into one database or any table. A layer which is JDBC layer acts as connectivity layer through which communications of applications is done (a special driver) with relational cloud. Relational cloud works as 1) As load increases, more than a single machine can handle, then database is being partitioned into one or more partitions 2) In order to balance out load and reduce no of machines.

3.10.4 DBMS in Cloud Architecture

DBMS in Cloud Architecture is a model consisting of many layers. First Layer is the storage layer. Here data monitoring, data encryption and backups can also be done. Next layer is Database layer. The application layer is the uppermost layer. Effective data access is provided also data values better distribution is provided.
Some SQL constructs which are often used are stored so as to provide better performance and reduce time consumption. At the storage layer, when data needs to be stored it is encrypted. It may be backed up, also there will then no need for encryption or decryption. Now the application layer generates a report. The report may be detailed and may be created at each step. It is used when accessing data and also helps in increasing performance.

![Diagram: DBMS in Cloud Architecture](image)

Fig 3.4: DBMS in Cloud Architecture [152][153][154]

3.10.5 Analytical Data Management Applications in Cloud
For the purpose of decision support, problem solving and business planning there is a need to query a data store. There is a large amount of old data as well as operational data which is needed in analysis. This all refers to “Analytical data Management”. This kind of Analytical Data Management Applications can very well deploy in cloud as they suit accordingly.

There are many reasons :

1. Shared Nothing Architecture : For analytical DBMS products, shared nothing architecture is used because of increase of data day by day which is used for analysis.

2. Transactions based on ACID : In order to perform analysis, atomicity, consistency and isolation is easy to obtain as the work performed in analytical databases is performed on recent snapshot of data.

3. Sensitive data left out : many a times, some part of data is utmost sensitive and organizations fear that leaking of the data could harm organizations so that crucial and sensitive data is left out of analysis. If that data has to be included, than it should be changed after applying a function or by encryption.