Title: A review on cloud database service

Mr. Shardul Pramod Patil
Department of Computer Science Engineering
D. Y. Patil Agriculture and Technical University
Talsande, India
shardullpatil@dyp-atu.org

I. Abstract

In the recent era of data science, AI and machine learning there has been an exponential growth in the size of the data and growth is expected to be even more in the future. Traditionally the RDBMS were used to store the data, but with the rise of cloud computing web-based applications have replaced the stand-alone applications, distributed servers have replaced the normal dedicated servers. Cloud computing has become very efficient because of the lower cost, ability to scale up and down, flexibility, and pay-as-you-go model. This is the major change after the birth of the Internet.

Cloud computing has revolutionized the way data is stored and accessed. Cloud databases are an integral part of the cloud computing paradigm, offering scalability, flexibility, and cost-efficiency for managing large volumes of data. Cloud databases such as Amazon RDS, IBM DB2 and MongoDB are becoming popular. Cloud databases find the restrictions of older relational databases. Cloud databases are basically used for data storage, retrieval, modification, and analysis. Data intensive applications such as data warehousing, data mining and business intelligence mainly use cloud databases. As well as the transactional data management applications such as banking, airline reservation, online ecommerce and supply chain management applications also take a benefits of cloud storage.

This research paper reviews the concept of cloud databases, exploring their features, it examines the architecture of cloud databases, the comparison between traditional Relational Database Management Systems (RDBMS) and NoSQL databases, and popular implementations in the cloud environment. Additionally, the challenges faced in their development and deployment.

II. Introduction

In recent times cloud computing has grown up and is strong enough to provide their services to the user. In recent years, there has been an eye-catching change in the commercial applications. Online applications have replaced standalone applications hosted on a single server in the organization. In addition, system storage has taken the position of specialized storage. The pay-as-you-go model, flexibility, and lower cost are the main causes behind the growth of cloud computing [1].

Cloud computing is a computing technique, where a large group of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. A Cloud can be defined as a parallel and distributed system which has a number of virtualized and interconnected computers. Cloud computing takes benefit of many technologies such as server consolidation, large and quick storage, grid computing, virtualization, N-tier architecture and robust networks. It provides quick, scalable and expensive services with low set up and inconsiderable maintenance cost. There are three service models of cloud computing as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) over the internet on-demand, anytime from anywhere and on the basis of “pay-as-you-go" model. These service models provide services such as the distributed operating system, virtual hardware, virtual platform, applications, distributed database and other services[2].

A database is an organized collection of data. A Database Management System (DBMS) 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 organized collection of related data records, files and other objects. A DBMS allows various user application programs to concurrently access the database.

A Cloud Database is a database that typically runs on a Cloud Computing platform, such as Windows Azure, Amazon EC2, GoGrid and Rackspace. A Gartner report predicts [3] that the revenue from the cloud DBMS will account for 50% of total DBMS market revenue by the end of 2022, indicating that cloud databases play a crucial role in the next generation of data management systems. Cloud-based databases often referred to as the Database as a Service (DbaaS). It allows organizations to fill greater deployment flexibility, hosted hardware and software infrastructure and utility-based pricing. It provides entire database functionality and allows users to access and store their database at remote location anytime from anywhere using the Internet. Amazon's SimpleDB, Amazon RDS, Google's BigTable, Yahoo's Sherpa and Microsoft's SQL Azure Database are the commonly used databases in the Cloud [4].

This paper aims to review a comprehensive understanding of cloud databases, their features along with popular implementations of cloud database. It also overviews the architecture of cloud database and the challenges involved in their development. It also highlights a comparative study of the traditional relational database management system and cloud computing database.

III. Cloud Database

Cloud databases have become increasingly popular as a result of the rapid growth in electronic data, evolving needs for data storage, improved internet infrastructure, and cloud computing. For the data management in Cloud Storage, there are two terminologies that are used, Data as a service (DaaS) and Database as a service (DBaaS). The difference between DaaS and DBaaS is in method of data storage and management. Cloud storage is virtual storage that allows users to store their files and objects. Dropbox, iCloud etc. are popular cloud storage services. Data as a Service (DaaS) enables customers to store data at a remote disk accessible via the Internet. Cloud storage cannot work without basic data management services. So, these two terms are used interchangeably. DBaaS is one step ahead. It provides full database functionalities and enables users to access and store their databases on remote drives whenever they want from anywhere in the world via the Internet. In DBaaS, the client has access to both data storage and querying capabilities, allowing them to change the data and get important information from the database. Amazon's SimpleDB, Amazon RDS, Google's BigTable, Yahoo's Sherpa and Microsoft's SQL Azure Database are the commonly used databases in the Cloud [4]. The term "Database-as-a-Service" (DBaaS) refers to software services that enable users to set up, operate and scale databases without the need for setting up physical hardware, installing software or configuring for performance. All of the administrative tasks and maintenance are taken care of by the service provider. Users also do not need to install software or hire staff members to maintain the database technologies. The DBaaS user has only responsibilities are using the database and controlling its content. DBaaS enables users to operate a database with a common set of abstractions (primitives) without knowing the implementations [5].

A database service created and accessible using a cloud platform is known as a cloud database. It provides many of the same functions as a conventional database with some more flexibility added by cloud computing. To implement the database, users install software on a cloud infrastructure. A database service built and accessed through a cloud platform enables enterprise customers to host databases without purchasing 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 [6]. These services can be accessible via a web interface, or an API offered by the provider. Cloud databases can collect, deliver, replicate, and push all your data to the edge using the new hybrid cloud concept. To deliver database requests anywhere in the world, users no longer need to deploy the dependent middleware.

IV. Features of Cloud Computing

There are several features of cloud computing [7] out of that some of them are as shown in fig 1:

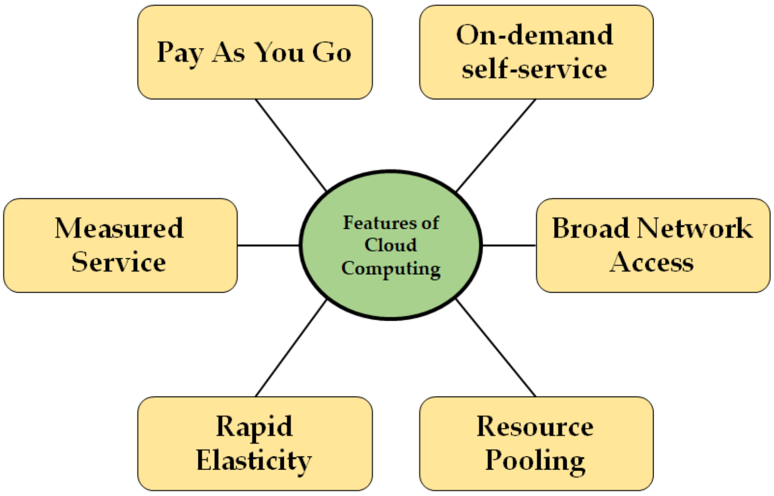

Fig 1. Features of Cloud Computing

A. **On-demand self-service:** Cloud computing delivers on-demand service. Cloud users can access their cloud accounts through a web self-service portal to view their cloud services, monitor their usage, and provision and de-provision services. They won't need to interact with the service provider [8].

B. **Broad Network Access:** Cloud users can access cloud services over the network and on portable devices like mobile phones, tablets, laptops, and desktop computers. Latency and bandwidth both play a major role in cloud computing and broad network access affect the quality of service [9].

C. **Resource Pooling:** The IT resource (e.g., networks, servers, storage, applications, and services) present are shared across multiple applications and occupant in an uncommitted manner. Multiple clients are provided service from the same physical resource. It allows customers to share the same applications or infrastructure while maintaining privacy and security [7].

D. **Rapid Elasticity:** The Computing services should have IT resources that are able to scale out and in quickly and as on needed basis. Whenever the user requires services, it is provided to him and it is scaled out as soon as its requirement gets over [7,8].

E. **Measured Service:** Cloud systems offer the metering capability to monitor, control, and optimize the usage of cloud resources. This feature can be defined as a measured service. This is done for various reasons like monitoring billing and effective use of resources. This feature enables transparency for both the provider of service and the consumer [9].

F. **Pay as you go:** In cloud computing, the user has to pay only for the service or the space they have utilized. There is no hidden or extra charge which is to be paid. The service is economical and most of the time some space is allotted for free.

V. Architecture of Cloud Database Management System

The development of cloud computing applications has numerous challenges, including service availability, data confidentiality, and shared nothing architecture. A successful cloud database management system is built to achieve the following objectives such as Accessibility, Scalability, Elasticity, Multitenancy, Load Balancing, Fault Tolerance, Ability to Operate in Multicultural Environment, and Flexible Query Interface [10]. Based on cloud DBMS design, there are five levels of object-oriented database architecture and five levels of schema architecture. At all five levels, each stage has its own role. Data will be represented at five levels in cloud computing. [11], [12], [13].

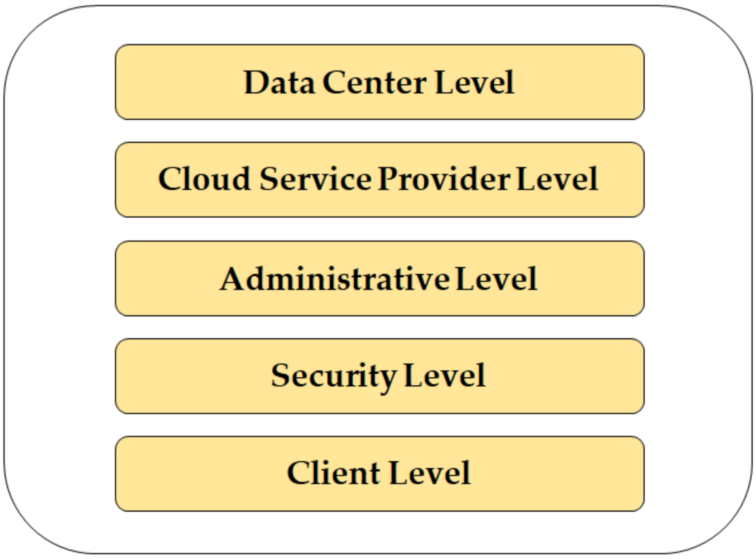

Fig 2: Layers of Cloud Architecture

A. Data Center Level:

This represents physical data storage location and contains several servers that meet the needs of cloud application users. This section can also be described as a cloud database representation. In cloud computing, data is kept as a virtualized pool of storage, i.e. storage virtualization moves data across numerous storage devices as if they were a single storage pool. Cloud providers maintain data centers based on their clients' demands and supply them with a storage pool where the client can store his data.

B. Cloud service provider Level:

As its name suggests this level provides the services to the clients that are promised to them. At this level cloud application is built and managed properly. It is a middleware level and involve of different distributed servers to meet the requirement of cloud uses. It will ensure all the facilities like availability of cloud at all times demands self-service multi tenancy elasticity and other services should be properly provided to the cloud users. It also provides to the cloud users. It additionally offers the capability of abstraction data in the cloud, i.e. it hides the information of the depository at the data center level and makes database and software basic. Users can see through it. This level is critical because it is the port that determines the appropriate access permissions of the users [11], [14], and [15].

C. Administrative Level:

This level makes use of API (Application Programming Interface) services. This application programming interface allows users and enables them to handle their database data. It creates data backups and restores databases in the instance of a failure. Databases can be created and deleted using the API. It allows for scalability based on the demands and requirements of the user.

D. Security level:

This level protects cloud users' data and guarantees the accuracy of the data without the usage of community copies. It assures end-to-end user data confidentiality and prevents data loss.

E. Client Level:

Cloud customers only see the database that they need to utilize, while other database details are hidden. At this level, operators offer a virtual illusion of resources based on their needs and impose them within their depository pool, which stores various data. The cloud user can be any computing device, such as a mobile application or a web browser [12], [16].

VI. Comparison of RDBMS and NoSQL

A greater demand for transaction processing applications existed in the early phases of computerization. Nowadays users want to store data not only for transaction processing, but also to analyze. Businesses want to increase the value of their operations by using data analysis. Therefore, transactional applications and analytical applications make up the two major categories of corporate applications. Most of the transactional data is handled by relational databases but they are not well efficient for analysis purpose [17].

These days, the amount of data is expanding quite rapidly. There are many new business applications that are centered around processing large amount of data. The form of data is also quite varying, for example data can now be anything like blogs, photos, tweets, messages and videos and so on. As a result, there is a extreme need for unstructured database like NoSQL databases, also in term of big data analytics [18]. NoSQL is one of the solutions that have been adopted by businesses in the IT world that are becoming increasingly interested in data-driven decision-making. The requirement to store and handle such large amounts of data defines the function of NoSQL databases in database technology such as Cloud databases. The following are some quick discussions of RDBMs and NOSQL databases:

A. RDBMS:

The concept of relational databases is the choice of the users from the last 40 years. Based on tables, columns, indexes, relationships, and schema, it features a rigorous database architecture. Tables and the predefined composite relationships are used to store data [20]. Column indices are utilized for faster search. Database design and maintenance require highly qualified developers and DBAs. RDBMS are typically used for transactional databases. SQL is used by RDBMS to conduct traditional database system queries like insert, update, delete, and modify. MySQL, MS-SQL Server, and Oracle are the most popular RDBMS. RDBMS guarantees ACID (Atomic, Consistent, Isolated, and Durable) transaction properties. Because they do not provide comprehensive content data search and are difficult to scale beyond a limit, these databases are not well suited for cloud environments.

B. NoSQL:

The phrase NoSQL (Not Only SQL) was initially introduced by Carlo Strozzi in 1998 [19]. In contrast to relational databases, which utilize mathematical relationships between tables, a non-relational approach was also used to store and retrieve unstructured data quickly. NoSQL database is referred as a non-relational, shared nothing, horizontally scalable database without ACID guarantees. Data integrity and timely completion are handled via the ACID property. While NoSQL relies on the BASE (Basically Available, Soft state, Eventually consistent) approach, which is significantly simpler. BASE model helps to provide flexibility to the NoSQL data so that the data might be structured properly. NoSQL is used by platforms like Big Data and website applications to store their data. Key/value stores, document stores, object stores, tuple stores, column stores, and graph stores are further categories for NoSQL implementations. Unstructured, semi-structured, and structured data can all be stored and retrieved using them.

VII. Challenges to Develop Cloud Databases

Cloud DBMSs should support aspects of Cloud figuring and extra databases for greater overall worth; this is Hercules' task. There are various potential issues associated with cloud databases, which are displayed in Fig. 3 [21].

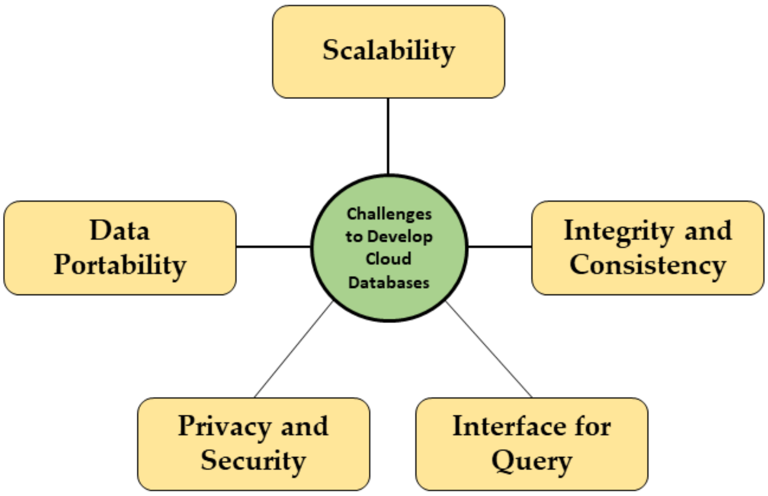

Fig 3. Challenges to develop cloud database

A. Scalability:

The rapid expansion in database size is a result of involving massive amounts of multimedia data, which necessitates the development of novel scalable systems. Because consumers expect to be able to easily scale up and down the size of data in databases to fit the needs of their business goals, cloud systems must offer scalable database services to meet their users' expectations. This is the most crucial aspect of the cloud standard. It offers services that can be scaled up or down significantly without causing any impediment in the organization. It is a significant problem in system architecture to deploy databases in the cloud to ensure that synchronous clients are supported and managed, and data can be improved.

Fault Tolerance and Reliability It is critical to replicate information across multiple geographical sites in order to provide high availability, robustness, and flexibility in adapting to internal failure. The word system availability can be broadly defined as the degree of accessibility and usage of resources for individual users or organizational staff [22]. This is one of the most critical topics that people or companies must examine before moving to a cloud database.

If a cloud service fails, it may influence the availability of databases, either momentarily or permanently, resulting in a significant loss of data, partially or totally. Equipment failures, security flaws, and DDoS attacks are all severe challenges to the availability of cloud database systems. In most circumstances, these kinds of failures are unexpected and can have a significant impact on the performance of companies' or individuals' activities, perhaps resulting in data corruption or the cessation of real-time services. The performance of a large number of database applications may suffer as a result of cloud service unavailability or failure.

B. Integrity and Data Consistency:

To ensure a high level of data integrity, it is critical to rigorously supervise and monitor database users, including database administrators and technical employees, who have legal permission to access the system [23].

Maintaining the consistency of a transaction in a database is also a challenging effort, made significantly more difficult if it changes rapidly, as it often does with value-based information. Designers must accurately mimic BASE (Basically Available, Soft State, Eventually Consistent) database properties. They must exercise caution to avoid losing data integrity throughout the transition to cloud databases.

C. Interface for Query:

The Cloud Database is widely used. Addressing passed on the database is a critical test that cloud planners must pass. A passed on inquiry must get to the specific focus purposes of the cloud database. For studying the database, there should be a streamlined and well-organized solicitation interface.

D. Privacy and Security of Database:

Before converting a traditional database to a database on a cloud platform, organizations should consider various security risks. Because the outcome will effect the organization's function, these security concerns are the primary and significant concern of the companies, not the cloud service provider. Specifically, if confidential data is maintained on local databases, it is critical to assure users of the security of cloud databases during the transfer process. Users should be promised confidentiality and data protection, in particular. It must be ensured that the data will not be illegally modified or stolen throughout the transfer process from the internal database to cloud storage. A secure procedure should be carefully planned and implemented to enable this safe transfer [23].

To obtain a high level of security and confidentiality, it is also necessary to encrypt the data saved on outsourced databases hosted at cloud storage.

The storing of value-based information on an insufficiently secured host poses risks. To prevent unauthorized access, sensitive data is encrypted before being stored in the cloud. The ability to decrypt data in the cloud should be limited for specific applications. It is a significant difficulty to guarantee the security and confidentiality of several databases on a single system.

E. Data Portability:

Information portability is the ability to run an application designed for one cloud provider in the settings and systems of another cloud provider. Interoperability is the capacity to give some codes that are adaptive enough to work with various cloud providers, regardless of their variances.

VIII. Popular Database Implementations in Cloud Computing

CouchDB

Apache CouchDB is an open-source project that is developed in Erlang and takes a document-oriented approach. Documents are generated in JSON and are intended to be viewed using CouchDB's JavaScript implementation of MapReduce views. This database employs a B-tree index, which is updated as data is modified. On the document level, these alterations have ACID qualities, and the usage of MVCC (Multi-Version Concurrency Control) allows readers to never block. CouchDB's document manipulation uses optimistic locks to update an append-only B-tree for data storage, which means that data must be compressed on a regular basis. Despite ensuring availability, this compression may impair performance [24]. CouchDB provides both master-slave and master-master replication systems [25], which can be utilized between separate CouchDB instances or on a single instance. Scaling in CouchDB is accomplished through data replication, which is done asynchronously. It does not support sharding/partitioning natively. Consistency is ensured through improved eventual consistency, and conflicts are resolved by picking the most recent version (the application layer can afterward try to reconcile conflicting changes back into the document, if practicable). The CouchDB programming interface is REST-based [24, 25]. Because CouchDB is largely a RAM-based database, it should be able to fit the entire dataset into the cluster's RAM.

Key features of CouchDB:

- CouchDB offers the most basic kind of replication, and no other database is as easy to replicate.
- It is a NoSQL database that uses document storage and includes values of many data kinds such as text, numeric, Boolean, lists, and so on.
- The CouchDB file layout adheres to all of the ACID characteristics.
- It provides database-level security, and permissions are separated into readers and administrators, with readers having both read and write access to the database.
- CouchDB can replicate to devices such as cellphones that have the ability to go offline and manage synchronization of data for user after the device reconnects.

Google BigQuery

There are few companies that must deal with a large dataset like Google [26]. It is no surprise that Google is at the forefront of the BigData scene in the twenty-first century, with many unique ideas and developments. At the turn of the century, Google discovered that "one-size-fits-all" SQL databases are insufficient for analytics workloads. They created a new database, "Dremel," for data-warehousing, or dealing with enormous amounts of

analytical data. In order to make their discoveries public, Google published a paper, "Dremel: Interactive Analysis of Web-Scale Datasets," in 2010.
In 2011, Google made public their private Dremel Database as BigQuery Database. Since then, it has become the most innovative and leading database for data warehousing and analytics load. Google Cloud Platform (GCP) has an established position in the Data Storage market, with BigQuery playing an important role.

Key Features of BigQuery:

- BigQuery is a multi-cloud data warehousing solution that is highly scalable and has independent storage, computing, and processing.
- It has a server less architecture that includes managed provisioning, maintenance, security, and scaling. It has multi-location mirrored storage and automatic high availability.
- Users can use BigQuery ML to create and run machine learning models in BigQuery using regular SQL queries.
- Its fast streaming insertion API serves as a solid base for real-time analytics.
- The BI Engine (in-memory analysis service) provides popular BI tools with sub-second query reaction time and high concurrency via conventional ODBC/JDBC.

Cassandra

Cassandra is aa open source, shared-nothing, NoSQL, column-store database that was developed and is used by Facebook [24, 27, 28]. It is built on the concepts that underpin Google BigTable and Amazon Dynamo [3,1]. In terms of data model, Cassandra is comparable to BigTable. A column is the smallest unit of storage, with rows made up of columns or super columns (nested columns). Columns are made up of the name, value, and timestamp, which are all submitted by the client. Because it is column-based, rows do not have the identical number of columns. Cassandra, along with other protocols, supports a SQL-like language called CQL [24]. Indexes and secondary indexes are supported, and atomicity is ensured down to the table row level. Logging ensures persistence. The appropriate level of consistency can be specified by the application developer, who trades off latency and consistency. Conflicts are resolved using timestamps (the most recent record is maintained). The database runs in master-master mode, with no node being distinct from the others, and combines disk persistence with in-memory caching of results, leading in significant write throughput operations [27]. The master-master architecture facilitates horizontal scaling. There are numerous partitioning mechanisms available, and the database can manage replication automatically [28].

Key Features of Cassandra:

- Cassandra may be scaled up or down without any difficulty or resistance due to its nodal architecture.
- Cassandra is designed on a peer-to-peer architecture model with all nodes are equal.
- For failure tolerance, data is automatically stored and replicated. If a node fails, it is immediately replaced.
- Cassandra's CQL (Cassandra Query Language) has been introduced. It's an easy user interface for gaining access to Cassandra.
- Cassandra offers flexibility in data distribution by duplicating data across many data centers.

Microsoft’s Azure SQL Database

Another significant player in the database industry is Microsoft. Microsoft dominated the mid-range Windows Systems commercial database market with Microsoft SQL Server. Microsoft provided a managed database service over Microsoft SQL Server when it accepted its Cloud-first strategy in the 2010s. Microsoft managed SQL Server underwent numerous revisions in the ensuing years. Currently, Azure SQL Database offers a wide range of value-added extra capabilities in addition to serving as Microsoft SQL Server's managed database-as-a-service. It is the preferred cloud database for many businesses, especially those that currently use Microsoft SQL Server because they can easily move their on-premise Microsoft SQL Server to the cloud.
Key Features of Microsoft’s Azure SQL Database:

- In addition to a standalone database, it provides adaptable Elastic pools to cost-effectively manage and scale many databases with changing loads.
- A serverless compute tier is available.
- It is a large scale SQL database with near-instantaneous backup and quick database recovery, and 99.99% availability despite infrastructure failures.
- For a particularly large-scale SQL dataset, it offers a Hyperscale service tier in addition to the normal tier.
- It offers a simple lift-and-shift transfer of on-premise Microsoft SQL databases to Azure SQL databases.

Amazon DynamoDB

Amazon discovered the hard way in December 2004 that the load generated by Web-Scale applications could not be supported by a centralized, robust, consistent RDBMS. Traditional SQL databases were unable to offer Amazon the high availability and horizontal scalability because of their tight consistency Model, relational architecture, and 2-phase commit. In 2007, the Amazon Engineering team created the DynamoDB novel NoSQL database and published the results in their Dynamo paper. A significant contribution to the later development of NoSQL databases like Cassandra and Riak was made by Amazon Dynamo paper.

Amazon's shopping cart application employed DynamoDB as its main database, although it wasn't made available to the public until 2012. Since then, DynmoDB has become the most widely used public cloud database and a top AWS service.

Key Features of DynmoDB:

- It is a document-based, key-value NoSQL database.
- It is a fully managed, multiple-region, multiple-master database with high availability.
- It is intended for applications at the web scale. It can manage peaks of more than 20 million requests per second and more than 10 trillion requests per day.
- A completely managed in-memory cache is provided by DynamoDB Accelerator (DAX).
- It enables single-digit millisecond response times at any scale due to its multi-region replications.

MongoDB

The 10gen firm created MongoDB, an open-source document-oriented database written in C++. Automatic sharing is primarily used by MongoDB to achieve horizontal scalability. Locks and the asynchronous master-slave model [28] both support replication. By specifying whether reads from secondary nodes are permitted and how many nodes must confirm a read, database clients can select the consistency models they prefer.

MongoDB places a lot of emphasis on document manipulation because it offers a variety of frameworks (like MapReduce and Aggregation Framework) and methods for dealing with documents. Among other activities, these can be searched, sorted, projected, and iterated using cursors. A document's modifications are ensured to be atomic. One or more fields may be subject to indexing, which is carried out via B-trees [24]. MongoDB supports a wide range of programming interfaces, and the majority of well-known programming languages have native bindings.

Key Features of MongoDB:

- Use of appropriate indexing to improve query execution.
- Replication to improve data stability and availability.
- The process of dividing larger datasets among various distributed collections is called "sharding".
- Large-scale load balancing is supported by MongoDB using technologies for horizontal scaling, such as replication and sharding.

IX. Conclusion

This paper reviews the significance of cloud databases in the era of data science, AI, and machine learning. The exponential growth of data and the rise of cloud computing have led to the development of efficient cloud databases, which offer scalability, flexibility, and cost-efficiency for managing large volumes of data.

The paper reviews the concept of cloud databases, their features, and the architecture of cloud database management systems. It also provides a comparison between traditional Relational Database Management Systems (RDBMS) and NoSQL databases, highlighting the advantages of NoSQL databases for handling unstructured data and big data analytics. However, developing cloud databases also comes with its set of challenges. These challenges include ensuring scalability, fault tolerance, data integrity, query interfaces, privacy, security, and data portability. Addressing these issues is crucial for successful cloud database deployment.

Several popular database implementations in cloud computing are discussed, including CouchDB, Google BigQuery, Cassandra, Microsoft's Azure SQL Database, Amazon DynamoDB, and MongoDB. Each of these databases offers unique features and benefits.

Overall, cloud databases have revolutionized the way data is stored, accessed, and analyzed. They provide businesses and organizations with the tools and capabilities to manage vast amounts of data efficiently, making them a vital component in modern data-driven decision-making processes.

X. Referance

Curino, C., Jones, E., Popa, R., Malviya, N., Wu, E., Madden, S., Balakrishnan, H., Zeldovich, N, "Relational Cloud: A Database Service for the Cloud", CIDR. pp. 235–240 (2011).

Sapna Jain et al, International Journal of Advanced Research in Computer Science, 8 (2), March 2017 (Special Issue),80-87

Massimo Pezzini, Donald Feinberg, Nigel Rayner, and Roxane Edjlali. 2021. Magic Quadrant for Cloud Database Management Systems. Gartner (2021, December 13) (2021), 1–37.

InduArora ,Dr.AnuGupta,"Clouddatabase:A paradigm shift in Databases"IJCI international journal,july 2012.

European Journal of Academic Essays 1(6): 12-17, 2014 Databases for Cloud Computing: Comparative Study and Review Pawan Singh1, Nirayo Hailu 2, Vinod Chandran 3

IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163 | pISSN: 2321-7308 Volume: 04 Special Issue: 06 | NCEITCS-2015 | May-2015, Available @ http://www.ijret.org 27 SURVEY ON THE DATABASES IN THE CLOUD ENVIORNMENT K. Priyatham1, Arif Mohammad Abdul2, M. Shanthi3

M. Mohammad, Z. Rafiqul. A Systematic Review on Cloud Computing. Vol.-6, Issue-11, Nov 2018.

Radack, S., Cloud computing: a review of features,benefits, and risks, and recommendations for secure, efficient implementations. National Institute of Standardsand Technology, 2012.

Puthal, D., B. Sahoo, S. Mishra, and S. Swain. Cloudcomputing features, issues, and challenges: a big picture.inComputational Intelligence and Networks (CINE), 2015International Conference on. 2015. IEEE.

S. Sakr, A.Lui, Daniel M. Batista, and M. Alomari: "A Survey of Large Scale Data Management Approches in Cloud Environment". IEEE Communication Surveys and Tutorials, Volume 13(3), 2011, page 311-336.

J. M. Hellerstein, M. Stonebraker, and J. Hamilton, "Architecture of a database system," Foundations and Trends in Databases. 2007, doi: 10.1561/1900000002.

S. Kumar and R. H. Goudar, "Cloud Computing – Research Issues, Challenges, Architecture, Platforms and Applications: A Survey," Int. J. Futur. Comput. Commun., 2012, doi: 10.7763/ijfcc.2012.v1.95.

M. Singh, "Study on cloud computing and cloud database," 2015, doi:10.1109/CCAA.2015.7148468.

B. Alam, M. N. Doja, M. Alam, and S. Mongia, "5-Layered Architecture of Cloud Database Management System," AASRI Procedia, 2013, doi: 10.1016/j.aasri.2013.10.078.

M. Muntjir and M. J. haque, "Cloud Database Infrastructure: Database System Transference in Cloud Computing Management and Security," Int. J. Comput. Trends Technol., 2017, doi: 10.14445/22312803/ijctt-v47p103.

B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," Futur. Gener. Comput. Syst., 2018, doi: 10.1016/j.future.2017.09.020.

Yashraj Sharma, and Yashasvi Sharma. (2019). "CASE STUDY OF TRADITIONAL RDBMS AND NOSQL DATABASE SYSTEM." International Journal of Research - Granthaalayah, 7(7), 351-359. https://doi.org/10.5281/zenodo.3364448.

Sapna Jain et al, International Journal of Advanced Research in Computer Science, 8 (2), March 2017 (Special Issue),80-87

A Survey of Comparing Different Cloud Database Performance: SQL and NoSQLTwana Hussein Shareef1*, Karzan Hussein Shareef 2, Bilal Najmaddin Rashid3 Shareef et. al. Passer 4 (2022) 45-57
Wang, R., & Yang, Z. SQL vs NoSQL: A Performance Comparison. (2017). From: https://www.cs.rochester.edu/courses/261/fall2017/termpaper/submissions/06/Paper.pdf

102 Arora, I. and A. Gupta, Cloud databases: a paradigm shiftin databases. International J. of Computer Science Issues,2012. 9(4): p. 77-83.

Sakhi, I., Database security in the cloud. 2012.

Basar et al., International Journal of Advanced Research in Computer Science and Software Engineering7(11) ISSN(E): 2277-128X, ISSN(P): 2277-6451, pp. 150-157

Kuznetsov S, Poskonin A (2014) Nosql data management systems. Program Comput Softw 40(6):323–332

Nocu´n Ł, Nie´c M, Pikuła P, Mamla A, Turek W (2013) Car-finding system with couchdb-based sensor management platform. Comput Sci 14(3):403–422

Postgres Plus, Cloud Database: Getting started Guide. Retrieved 23rd November, 2012.

Gudivada VN, Rao D, Raghavan VV (2014) Nosql systems for big data management. In: Services (SERVICES), 2014 IEEE World Congress On. IEEE, Anchorage, AK, USA. pp 190–197

Haughian G (2014) Benchmarking replication in nosql data stores. Dissertation, Imperial College London

DeCandia G, Hastorun D, Jampani M, Kakulapati G, Lakshman A, Pilchin A, Sivasubramanian S, Vosshall P, Vogels W(2007) Dynamo: amazon's highly available key-value store. In: ACM SIGOPS Operating Systems Review. ACM,Stevenson, Washington, USA. Vol. 41. pp 205–220