

Research Article

The Art of Handling Databases by Cloud Computing

¹R. Anandhi and ²K. Chitra

¹Department of Computer Science, SCSVMV University, Kanchipuram,

²Govt. Arts College, Melur, Madurai, Tamil Nadu, India

Abstract: It is obvious that there are tremendous inventions and technological growth in IT industry. All upcoming technologies are almost trying to deal with only one concept “DATA” -how the data can be effectively stored, easily and accurately retrieved, efficiently distributed and queried etc. Hence one of the new such computing facility is Cloud Computing-accessing various computing resources that are geographically apart. It surely gets a significant attraction over the IT industry. This study explains and analyses the capabilities of Cloud Computing while dealing with databases such as Scalability, Availability, Consistency and Elasticity.

Keywords: Cloud computing, cloud databases, database-as-a-service, elasticity, scalability

INTRODUCTION

Cloud computing provides accession to remotely available computing resources over a network (the Internet) (Maggiani, 2009). Users can buy these computing resources, such as storage and computing power, as a utility on demand. The name ‘Cloud Computing’ is derived from the cloud-shaped symbol depicting the complex infrastructure it contains in common system diagrams. Cloud computing entrusts remote services with user's data, software and computation. Cloud computing provides virtually infinite pool of computing, networking and storage resources where our applications could be scalable deployed (Anandhi and Chitra, 2012a).

The advantages of Cloud Computing (Krissi, 2008) are:

- Cost efficient
- Unlimited storage
- Backup and recovery
- Automatic software Integration
- Fast deployment

Usually cloud storage provides the required amount of storage in an immediate basis and it is also a persistent one. The storage can be accessed by two ways: either at the data center where the cloud is hosted or via internet (Logothetis and Yocum, 2008). Cloud storage will be stored at a single place in order to avoid the single point of failure and it is geographically separated to escape from failure or any natural calamities. It is a great challenge to deploy a normal

traditional database system in cloud, since traditional databases are usually optimized to all sorts of queries and they obey consistency (Anandhi and Chitra, 2012b). Unfortunately, the assumptions of availability, scalability and flexibility do not correspond to the cloud model since resources are allocated dynamically and there exist a loose coupling between data management and systems (Chitra and Jeevarani, 2013). Cloud computing is also known as “Elastic Computing”. It should be noted that the underlying database is not very elastic and scalable. The following are the usual features that are present in a cloud environment:

- Compute power is elastic, but only if workload is parallelizable.
- Data is stored at an un-trusted host.
- Data is replicated, often across large geographic distances.
- It is hard to maintain ACID guarantees in the face of data replication over large geographic distances.
- There are enormous risks in storing transactional data on an un-trusted host (Donald *et al.*, 2010).

This study presents and analyses the capabilities of Cloud Computing Database such as Scalability, Availability, Consistency and Elasticity. It discusses the data management facilities such as transaction data management and analytical data management. It elaborates about the services provided and the challenges faced by the Cloud Computing Systems. This study also deals with the NewSQL databases and Virtualization in Cloud Computing Systems.

Corresponding Author: R. Anandhi, Department of Computer Science, SCSVMV University, Kanchipuram, Tamil Nadu, India

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).



Fig. 1: Cloud computing DaaS architecture

CLOUD COMPUTING ARCHITECTURE

Cloud computing architecture partition the data for providing incremental scalability and the partitioned data is replicated to tolerate any unexpected server failures at any time. The high availability and scalability properties of cloud platform are given at a cost.

The new three aspects of Cloud Computing from the point of hardware are:

- Illusion of infinite computing resources available on-demand and hence eliminating the need for cloud computing users to plan far ahead.
- Elimination of an up-front commitment by Cloud users, thereby allowing companies to start small and increase hardware resources only when there is an increase in their needs.
- Ability to pay for use of computing resources on a short-term basis as needed and release them as needed, thereby rewarding conservation by letting machines and storage go when they are no longer useful.

Now there are various companies that come forward to act as cloud service providers. We are already aware of the services like SaaS, PaaS and IaaS. Apart from that there is also one more popular service called as DaaS (Database as a Service) (Fig. 1).

This is because in a company, money has to be considerably spent on both hardware and software in order to maintain their database system. Instead of owning, installing and maintaining the database software, cloud computing vendors typically maintain little more than the hardware and give customers a set of virtual machines in which to install their own software (DeWitt and Gray, 1992). Cloud services can provide efficiencies for application providers by limiting the cost of ownership of the database system. Such services are made available in a data center, using shared commodity hardware for computation and

storage. There is a varied set of cloud services available today, including storage services (Amazon S3), data services (Amazon Simple DB, Microsoft SQL Server Data Services, Google's Data store), application services (salesforce.com), compute services (Google App Engine, Amazon EC2) etc., (Lewis, 2009; Matthias, 2008).

Three steps to map a path to cloud-based database:

- **Understand the data:** Must be aware of meaning and interrelation between the data.
- **Define objectives of data security and data governance:** Public cloud databases lack control of data i.e., absence of security and data governance. Hence increase your ability to control data.
- **Define an improved path not just migration:** Moving to cloud database means changing the way of storing and retrieving data i.e., make the data more valuable to both data owners and the data users to make strategic decisions (Maggiani, 2009).

DATA MANAGEMENT IN CLOUD

Here we are going to deal with two options of data management in cloud. They are transaction data management and analytical data management.

Transactional data management: The transaction workload is assumed to be made up of stored procedures. The next step is database partitioning to achieve better scalability since each partition independently executes transactions. But some transactions want the co-ordination of several partitions at the same time, which are often referred as multi-partition transactions (Donald *et al.*, 2010). In order to provide a consistent view across the partitions of a database, many algorithms have been proposed so far like Paros algorithm, the two-phase commit protocol, rendezvous protocols etc. They refer to the databases like reservation systems, online trading, banking etc. They should surely stick with ACID (Atomicity, Consistency, Integrity and Durability) properties and they are write-intensive (Anandhi and Chitra, 2012a). Usually these databases can't be suitable for cloud because:

- **Do not use a shared-nothing architecture:** Cloud needs a shared-nothing architecture since the data is partitioned across various sites rather than on a single node and thereby providing the concept of scalability. But the transactional data management is mostly run by Oracle, where there is no shared-nothing architecture.
- **Hard to maintain ACID:** As we have already said cloud follows CAP theorem and not ACID. CAP theorem states that a shared database system can have at most two out of three properties at a time:

Consistency, Availability and tolerance to Partitions (Anandhi and Chitra, 2012b).

- **Risk in storing on an un-trusted host:** The transactional database management system often stores sensitive information like credit card numbers, Passwords, PINs etc. Any form of hacking will lead to tremendous problems. But it is known that cloud storage is not local to premises and it may be located anywhere which is not advisable in this issue.

Analytical data management or replication management: One best approach to improve the performance of a database system is data replication. Cloud Computing has elasticity as one of its feature to have scalability in applications. There should be a well-defined standard to decide which partitions have to be replicated and the replica should appear in which partition. The most common replication approaches are either an optimistic approach or a pessimistic approach. Here the database developed for the applications for problem solving, decision support, business planning etc., is dealt. Compared to transactional data management, this analytical data management runs well on cloud environment for the following reasons:

- **Use shared-nothing architecture:** Some of the database products like Vertica, Aster Data, Teradata, Green Plum use shared-nothing architecture. It is quite surprising that Oracle also adding shared-nothing products. Here seldom writes reduces the need of strong lock and commit protocols. So the main reason for choosing this architecture is the increasing amount of data in the manipulation.
- **No need for ACID:** Rare writes in the database do not insist on ACID.
- **Leave the sensitive data:** Most sensitive data may be left before analysis and included after the encryption occurs from an analytical data store.

The Cloud Database is a database that typically runs on a cloud computing platform. There are two models of deployment:

- With the help of virtual machine image, users can run databases on the cloud independently (i.e.,) Cloud Providers allow the clients to purchase virtual machine instances for a specific time period. EG: Amazon EC2, GoGrid, Rack space.
- Users can purchase the access to a database service provided by a cloud database provider i.e., Database-as-a-Service (DaaS) without physically launching a virtual machine image, the cloud platforms offer choices for using a database. Here the database service provider takes responsibility for installing and maintaining the database and the users have to pay according to their usage. EG: Amazon Web Services provides two database

services like Simple DB (NoSQL key-value store) and Amazon Relations Database Service (SQL based services with a MySQL interface). The Database-as-a-Service (DaaS) provides many benefits as:

- **Higher availability:** The application does not have to be configured and managed individually; all can be benefited from automatic failover.
- **Economy:** By hiring all resources, the overall cost to the business can be reduced.
- **Centralized control:** By having the centralized control to the database, there is no need to appoint DBA for each development group.
- **A few risks:** The whole network is supported and managed by a team which is available on request.

Surely database systems are critical elements for business since it requires high degree of availability and scalability at low cost. However cloud storage systems do not provide full transactional support. If we are in need of high scalability, then relaxation of consistency occurs (Anandhi and Chitra, 2012b). Reduce the interaction of transactions among replicas is a way to handle the transactions efficiently. So a solution is to partition the data and the transactions running in a single partition would not require the support with other nodes. Two types of consistency models are considered for the development of a DaaS model (Anandhi and Chitra, 2012b). They are Deferred and Immediate Consistency models:

- **Deferred consistency:** This approach may use a write back cache or write back data log. This model provides an eventual consistency.
- **Immediate consistency:** It always ensures that the data accessed by the clients will always be consistent; it is a real threat to the cloud data store. It guarantees synchronous replica across all partitions.

A general architecture will include the elements like Data Objects, Update Queue, Queue Manager, Interest Group, Transaction Manager, Master Transaction Manager, Replica Manager, Log Manager, Failure Manager etc. There are various architectures of Cloud in the view of database storage. They are:

- **Classic:** It allows using “best-of-breed” components at all layers and allows scalability and elasticity at the storage and web server layers, e.g.: AWS MySQL (Fig. 2).
- **Partitioning:** Here the database is logically partitioned and each partition is controlled by a separate database server, e.g.: Google AppEngine (Fig. 3).
- **Replication:** There are various database servers and each database server controls a copy of the whole database or partition of the database if combined with partitioning. e.g.: MySQL/R (Fig. 4).

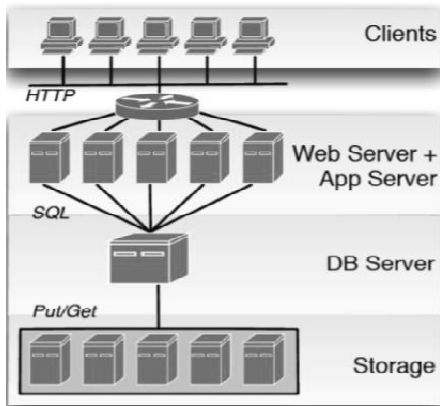


Fig. 2: Classic

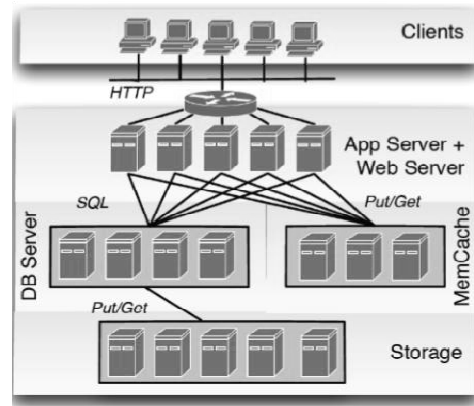


Fig. 6: Caching

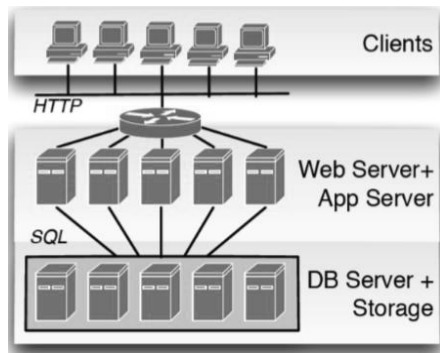


Fig. 3: Partitioning

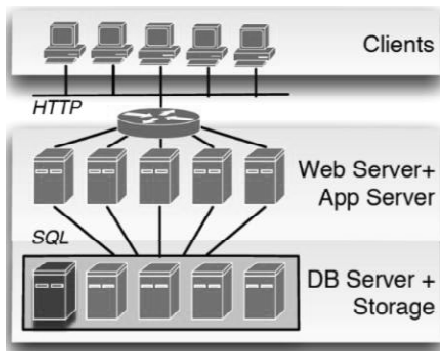


Fig. 4: Replication

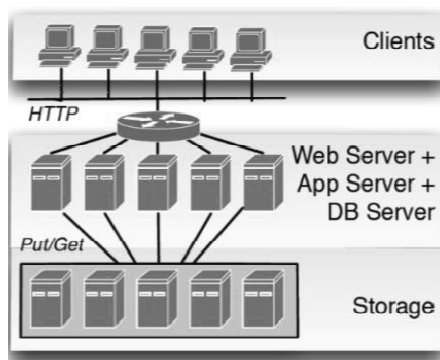


Fig. 5: Distributed control

- **Distributed control:** The database servers access concurrently and autonomously the shared data from the storage system which is separated from the database servers, e.g.: SWSS3 (Fig. 5).
- **Caching:** The results of database queries are stored by dedicated cache servers, which are referred as memcache, e.g.: Google AppEngine supports memcache (Fig. 6).

So one of the limitations of the cloud computing is the transactional data management. There are various ideas to find a satisfactory way of deploying the database applications in Cloud and also support the operations for data analysis.

CHALLENGES FACED BY CLOUD WHILE HOSTING DATABASES

Vibrant environment: Since cloud environment is drastic in nature, it is necessary to have a close monitoring as server crashes, hardware failures are more common.

Service level agreement: It is very hard to setup SLAs for cloud database to run the database efficiently in the vibrant cloud environment regardless of its size.

Availability: To maintain high availability in the cloud environment, it is necessary to have the availability of “more of the same” resources which is typically more complex and expensive.

Scalability: It is very hard to scale the database tier rather than scaling an application since the database should be scaled in the parameters like throughput and performance. Elasticity of a database not only means increasing the size but also shrinking its size when under utilization.

Management overhead: The operations in cloud are heavy, tedious and complex to achieve. To achieve the features like fault tolerance, availability, scalability etc.,

constant monitoring is essential and keep multiple copies of database and moreover all copies should be synchronized. So proper decision should be taken whether the resources are to be added (Over Utilization) or removed (Under Utilization). Minimization of management overhead of keeping it all running should be kept in mind.

Developers and expertise: For effective running of cloud applications, proper support from the side of developers and expertise are needed; for that a good skill set is required which often lacks.

Distributed databases: To achieve availability, most of the databases depend upon the architecture of distributed databases. But distributed databases mostly follow stateful approach. To have this nature of databases, multiple active copies are to be maintained in several locations in spite of network problems.

Multi-tenancy: Multi-tenancy in cloud is hard to satisfy which enables a cost-effective framework for large cloud customers to run multiple databases simultaneously.

According to the place of hosting the databases, it is of three types as follows:

- **Pure cloud databases:** 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,” says Wiqar Chaudry, director of product marketing and technology evangelist for NuoDB.
- **Cloud-optimized databases:** Cloud-optimized databases can be deployed both on-premises, or within the cloud by a dedicated hosting service, public cloud, or private cloud. Examples are traditional relational databases Oracle, SQL Server, MariaDB and MySQL (which can be enhanced for the cloud by GenieDB), as well as NewSQL databases such as NuoDB, Clustrix and TransLattice and NoSQL (Not only SQL) databases such as 10 Gen, Cloudant and Infinite Graph.
- **NoSQL databases:** While some pure cloud and cloud-optimized databases are based on relational technology, NoSQL databases support unstructured or non-relational data types-the essence of “big data” now flooding organizations. A survey of 298 data managers and professionals conducted by Unisphere Research among IOUG members finds

about 11% have adopted NoSQL within enterprise settings, a number expected to grow to 15% within the year (“Big Data, Big Challenges, Big Opportunities”).

There are four major database groups that fall under the NoSQL umbrella. Key-value stores enable the storage of schema less data, aligned as a key and actual data. Column family databases store data by rows, as is the case with relational databases and instead store data within columns. Another NoSQL variation, the graph database, employs structures with nodes, edges and properties to represent and store data. For example, Objectivity's Infinite Graph database is designed to enable end users “connect the dots” on a global scale, ask deeper and more complex questions, across new or existing data stores. Another type of NoSQL database, document databases, facilitates simple storage and retrieval of document aggregates.

NewSQL databases: NewSQL is another emerging category of cloud databases. They are different flavors from Oracle and MySQL, which are traditional relational databases. NewSQL is a class of modern relational database management systems that seek to provide the same scalable performance of NoSQL systems for online transaction processing (read-write) workloads while still maintaining the ACID guarantees of a traditional single-node database system. E.g.: The Trans lattice Elastic Database (TED) is a relational database management system that provides ANSI-SQL support, the ACID transactions enterprise applications require and the ability to scale-out across wide distances using ordinary Internet connections. Some of the most emerging cloud databases are Amazon Web Services, EnterpriseDB, Garantia Data, Google Cloud SQL, Microsoft Azure, Mongo Lab, Rackspace, SAP, Storm DB and Xeround.

Virtualization-cloud having little impact on databases: Cloud computing has taken center stage during the last 2 years as the hot enterprise technology, but in the database realm, the public cloud and off-site hosted database services still represent too many unknown factors to truly have mass adoption (Fig. 7).

According to a survey of 760 business technology professionals, only 7% are currently using a cloud provider for their primary database technology. Most of those who have adopted cloud-based database services are still managing database operations within their own staff IT departments. Of the respondents, only 2% are using a fully managed cloud-based database. Enterprises are looking to virtualization for the promised benefits of ease of deployment, reduction of hardware costs, reduction in energy costs and simplified disaster recovery. Databases benefit most from ease of deployment and simplified disaster recovery.

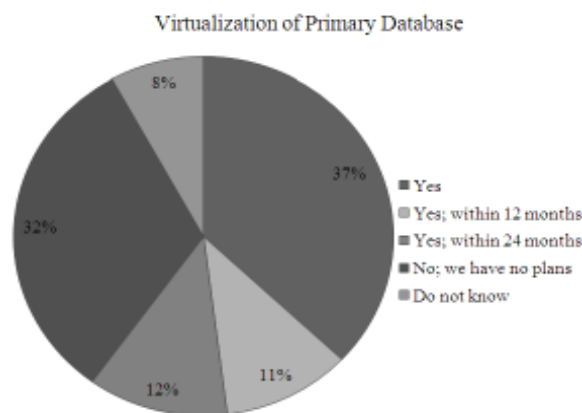


Fig. 7: Business technology professionals' survey

According to Read, database management system products almost universally make replication and recovery straightforward, which mutes that aspect of attraction to virtualization. Based on the survey results, though, some companies have adopted and realized the benefits of virtualizing their database environments, but the applicability of virtualization for database servers is specific to an IT department's strategy and operational planning.

DISADVANTAGES OF CLOUD COMPUTING

Technical issues:

- Always prone to outages and other technical issues
- Need a very fast Internet connection to the server at all times
- Invariably be stuck in case of network and connectivity problems

Security in the cloud:

- Great risk of surrendering all your company's sensitive information to a third-party cloud service provider
- Make absolutely sure that you choose the most reliable service provider, who can keep your information fully secure

CONCLUSION

Cloud computing usually involves huge amounts of data that is to be stored on countless servers and the computing power is heavily needed for read and write operations. In spite of that, the cloud computing

technology has hit the whole IT industry keeping the rule as the number of reads overrides number of writes. But as the user activities increase intensively, number of writes also increases dealing with lot of updates on the databases. This puts a black mark on the cloud databases as it already having the ACID complaint. So the main complaint with the cloud databases is that the concept of sharing is almost absent among the numerous servers and merely it is meant for data storage.

REFERENCES

- Anandhi, R. and K. Chitra, 2012a. An overview about cloud computing. *Int. J. Inform. Technol. Knowl. Manage.*, 5(1): 27-30.
- Anandhi, R. and K. Chitra, 2012b. A challenge in improving the consistency of transactions in cloud databases-scalability. *Int. J. Comput. Appl.*, 52(2): 12-14.
- Chitra, K. and B. Jeevarani, 2013. Study on basically available, scalable and eventually consistent NOSQL databases. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, 3(7).
- DeWitt, J. and J. Gray, 1992. Parallel database systems: The future of high performance database systems. *Commun. ACM*, 35(6): 85-98.
- Donald, K., K. Tim and L. Simon, 2010. An evaluation of alternative architectures for transaction. *Proceeding of Systems Group, Department of Computer Science, ETH Zurich, Switzerland*, pp: 579-590.
- Krissi, D., 2008. Distinguishing Cloud Computing from Utility Computing. *SaaS Week Blog*. Retrieved from: http://www.ebizq.net/blogs/saasweek/2008/03/distinguishing_cloud_computing/.
- Lewis, C., 2009. MySQL in Space-Amazon Relational Database Service (RDS). *CloudDB Blog*, 27.10.2009. Retrieved from: <http://clouddb.info/2009/10/27/mysql-in-spaaaaaace-amazon-relational-database-servicerds>.
- Logothetis, D. and K. Yocum, 2008. Wide-scale data stream management. *Proceeding of the Usenix Annual Technical Conference*. Boston, MA.
- Maggiani, R., 2009. Cloud computing is changing how we communicate. *Proceeding of the IEEE International Professional Communication Conference (IPCC, 2009)*, pp: 1-4.
- Matthias, B., 2008. Building a database on S3. *Proceeding of the Systems Group, ETH Zurich*, pp: 251-263.