Unit-5 (Part-2)
Cloud Database

Contents

- Cloud Database- Operational Model for Cloud Database, Types of Cloud Database

- Cloud File System- Distributed File System Basics, Concept of GFS and HDFS, Comparison of Features

Cloud Database

Cloud database is a database that runs on a cloud computing platform like Amazon EC2, Rackspace and GoGrid. There are two ways to deploy a database- users can either run the database inside a secured virtual machine (VM) or subscribe for particular database services managed by a cloud service provider. Currently, there are some SQL-based and some NoSQL-based database offerings.

Operation model for cloud database:

1. **VM Image:-** Cloud platforms allow users to purchase VM instances for a limited time. A cloud provider facilitates more security for running databases inside a VM. If users have their own VM image, then they upload it and run the database inside that or do so through preinstalled databases. Oracle, for example, provides preinstalled image with the Oracle database 11g for Amazon EC2 instances.
2. **Database as a service:-** Some cloud platform and infrastructure service providers offer database services just as other services offerings, in which case we do not need to launch any instance or individual VM for database installation. All database licensing, updating and configuration are managed by the cloud provider. Application owners have to, each month, pay-per-use of database volume. AWS provides many data-base services offering to their customers including relational database services(RDS) and NoSQL services such as Amazon RDS, Amazon DynamoDB, Amazon SimpleDB and Amazon Redshift.

   The traditional application owner prefers RDS and in RDS, users have many choices such as MySQL, Oracle, SQl Server or PostgreSQL database engines. All enterprise licensing issue and updates are taken care of by the provider.

Architectural and common Characteristics

- **Fast Deployment**: Cloud databases are the perfect choice when you urgently need a database, as they can be up and running in minutes. Cloud databases eliminate the need to purchase and install hardware and set up a network.
- **Accessibility**: Users have quick access to cloud databases remotely through the web interface.
- **Scalability**: You can expand cloud database storage capacity without disruptions and meet the requirements. Cloud database scalability is seamless due to DBaaS implementation, which is a major benefit for growing businesses with limited resources.
- **Disaster Recovery**: Data backups are regularly performed on cloud databases and kept on remote servers. These backups enable a business to stay online in cases of natural disasters, equipment failure, etc.

- **Lower Hardware Costs**: Cloud database service providers supply the infrastructure and perform database maintenance. Hence, companies invest less in hardware and have fewer IT engineers for database maintenance.
- **Value for Money**: Many DBaaS solutions are available in multiple configurations, allowing companies only to pay for what they use and turn off services when they don't need them. Cloud databases also save money by not requiring operational costs or expensive upgrades.
- **Latest Tech**: Cloud database providers upgrade infrastructure and keep it updated with new tech. This brings significant savings as companies don't have to allocate funds on new tech or staff training.
- **Security**: Most cloud database providers encrypt data and invest in the best cloud security solutions to keep the databases safe. Although there is no impenetrable security system, it is a safe way to protect data. Since cloud database providers use automation to enforce the best security practices, there is less room for human error compared to using on-premises databases.

Types of Cloud Databases

Many cloud providers offers RDS nowadays. Some popular and most adopted RDS across the globe are as follows:

1. **Amazon relational database service:** Amazon RDS is very popular and widely adopted Web service. It looks like other AWS services and provides easy management consoles for operating RDS on cloud. Amazon RDS is a highly cost-efficient and secured service. Currently it supports Oracle, SQL Server, MySQL and PostgreSQL database. Amazon RDS specifically offers two types of RDS instances.
    - On-demand instances: An on-demand instance offering is a pay-per-use instance with no long-term commitment.
    - Reserved DB instances: Reserved DB instances give the flexibility of one-time payment for the DB instance if the database usage is predictable. There also an offer of 30%-50% price cut over the on-demand price.

2. Google cloud SQL: Google cloud SQL is a MySQL database service that is managed by Google, and the entire management, data replication, encryption, security and backups are handled by Google's cloud infrastructure. Google claims maximum availability of its data because its data centers are located across every region of the world.

3. Heroku Postgres: Heroku Postgres is a relational SQL database offered by Heroku. It is accessible through all programming languages supported by Heroku. It basically provisioned as an add-on service. Heroku Postgres offers fully reliability of services, which means around 99.99% uptime and 99.999999999% durability of data. One of the advance features of Heroku Postgres is Dataclips, which enables users to send the results of the SQL query via the URL.

4. HP cloud relational database for MySQL: HP cloud RDS automate application deployment, configuration management and patch-up task database. It currently supports command line interface (CLI). It also provides database snapshot facility in multiple availability zones for providing more reliability. It is also built atop an OpenStack-based MySQL distribution, which provides database interoperability from one cloud provider to other.

5. Microsoft Azure SQL database: Earlier it was known as SQL Azure. It is the most important component of the Microsoft Azure cloud service; however, it can be operated as a standalone cloud database also. The database can be synched easily with other SQL server databases within the cloud infrastructure of the company or organization. With Microsoft Azure SQL database, the performance of database can be predicted irrespective of whether the service chosen is basic, standard or premium.

6. Oracle database cloud service: Oracle database cloud offers two options for users: one is a single schema-based service and another is fully configured Oracle database installed virtual machine. Oracle database can be quickly provisioned, and the user can spin up a database instance with just a few clicks. It also provides flexibility in the management option: self managed service or fully managed by Oracle.

7. Rackspace cloud databases: Rackspace cloud databases are based on open standards. These Currently support MySQL, Percona and MariaDB databases. Rackspace cloud provides high database performance using container-based virtualization. It provides automated configuration, which reduces operational costs and team effort. Rackspace cloud is built on top of an open source technology like the OpenStack cloud platform.

Limitation with Existing Database

Following are some of the key limitations that became the reason behind the birth of NoSQL databases. Traditional databases are unable to:

1. Store data in TB/PB; even a good processor cannot process millions of rows.
2. Process TB of data on a single machine.

Types of NoSQL Database

1. **Key-value store:** Based on table keys and values (e.g. AWS DynamoDB).
2. **Document-based store:** Document-based database stores records that are made of tagged elements (e.g. MongoDB, CouchDB).
3. **Column-based store:** Data divided into multiple columns and every storage block contains data of each column (e.g., Apache HBase, Cassandra).
4. **Graph-based store:** A network graph storage that uses edges and nodes for storing data (e.g. Neo-4)

Distributed File System Basics

Distributed file system (DFS) is basically used for storing huge amount of data and provides accessibility of stored data to all distributed clients across the network. The objective of the DFS is to provide a system for all the geographically distributed users as a common file system for data sharing and storage.

An Internet search engine is the most common example of DFS, which is used for indexing millions of Web pages. There are a number of DFS that solve this problem in different ways. Some popular file systems are:

1. Andrew file system (AFS)
2. Network file system (NFS)
3. Microsoft distributed file system (DFS)
4. Apple filing protocol (AFP)
5. Google file system (GFS)
6. Hadoop distributed file system (HDFS)

Concept of GFS

Google invented and implemented a scalable DFS to handle their huge internal distributed data exhaustive applications and named it the Google File System. In 2002-03, Google launched its file system based on DFS architecture but added some advance features that are driven by Google's unique workload and environment.

Google File System Architecture

A cluster of a Google file system contains a single master and multiple chunk servers that are associated with many clients. The master holds the metadata of chunk servers. All the data processing happens through these chunk servers. The client first contacts the master and retrieves the metadata of the chunk server, which is then stored in the chunk servers. So the next time, client directly connects to the chunk servers.

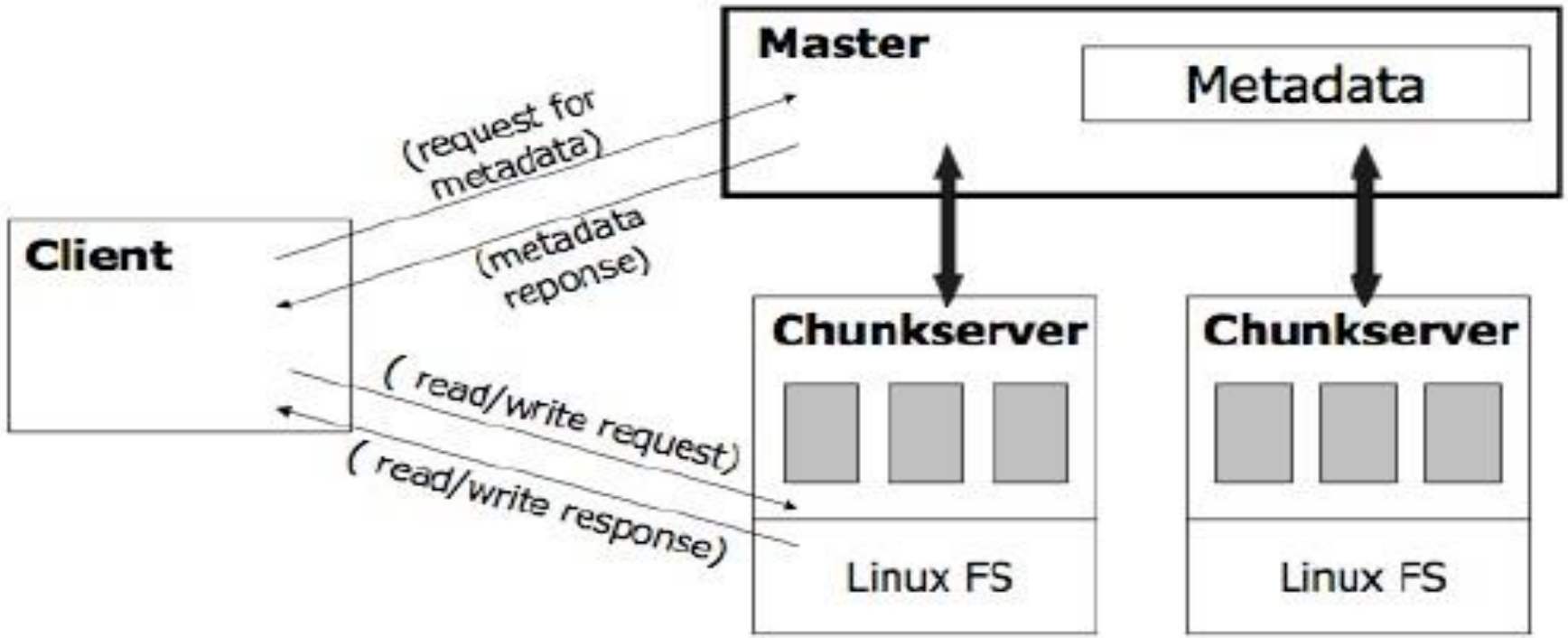

Master
Metadata
(request for metadata)
(metadata reponse)
Client
Chunkserver
Chunkserver
(read/write request)
(read/write response)
Linux FS
Linux FS

Figure 1

Following a are the details of each component of GFS:

1. Chunk: A chunk is very similar to concept of block in a file system, but chunk size is larger than the traditional file system block. The block of chunk is 64 MB. This is specifically designed for the Google environment.
2. Master: Master is a single process that runs on entirely separate machine for security purposes. It only stores metadata-related information, chunk location, file mapping information and access control information. The client first contacts the master for information about metadata and then connects to that particular chunk server.
3. Metadata: Metadata is stored in the memory of a master, therefore, master operations are much faster. Metadata contains three types of information:
   - Namespaces of file and chunk
   - Location of each chunk
   - Mapping from file to chunk

Concept of HDFS

HDFS is a DFS based on GFS that provides high throughput access to application data. It uses the commodity hardware with the expectation that failures will occur and provides portability across heterogeneous hardware and software platforms. HDFS is acquired by Hadoop Apache open source project, which is very popular these days for its ability to handle big data.

The Hadoop core consists of two modules:

1. Hadoop distributed file system: Used for storing huge amount of data.
2. MapReduce programming mode: Used for processing of large set of data.

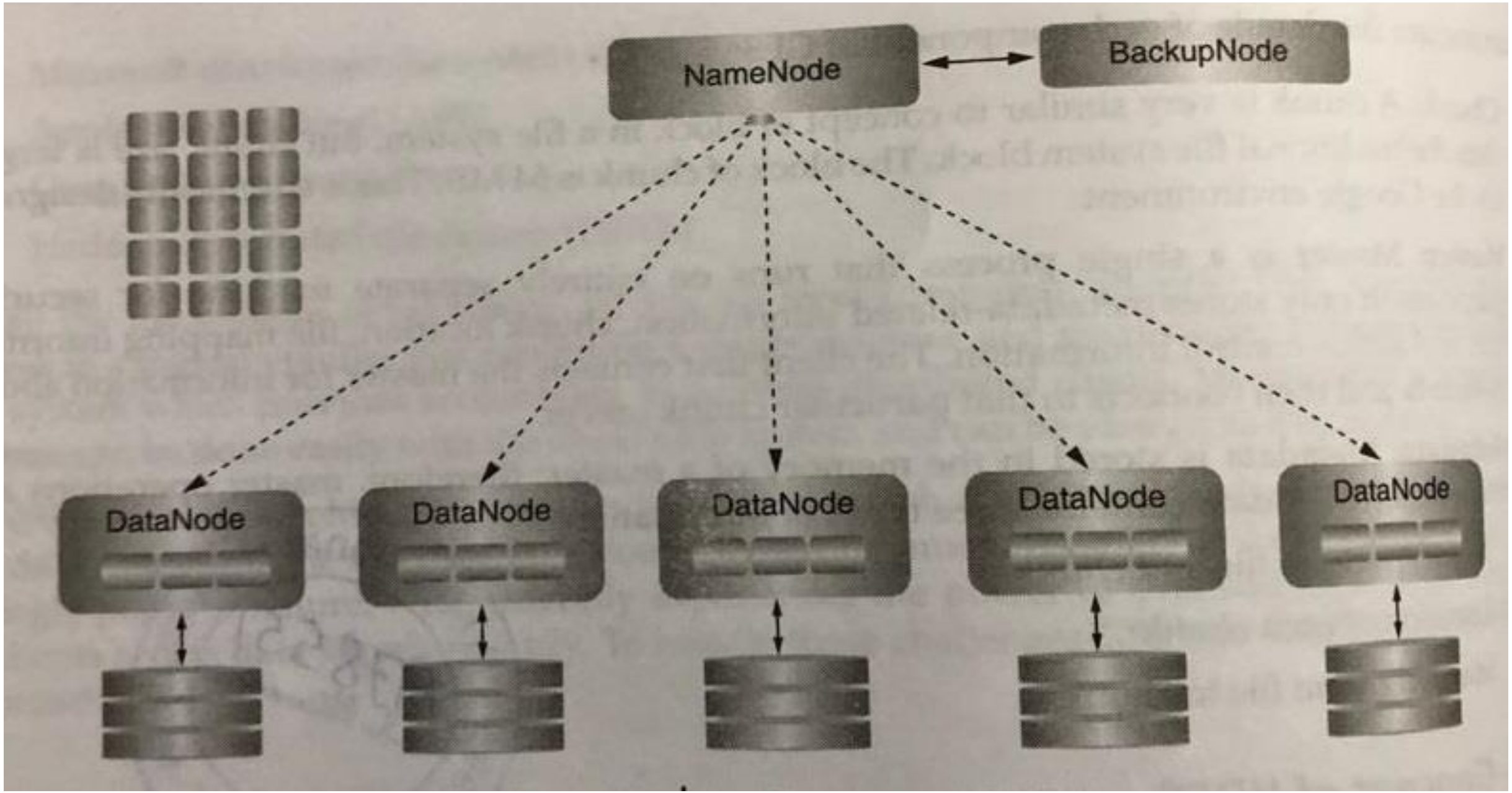
Architecture of HDFS

The working architecture of HDFS is almost similar to GFS.

An HDFS cluster consists of multiple commodity machines that can be classified into the following three types:

1. Name node (runs on master machine)
2. Secondary name node or backup node (runs on separate machine)
3. Data node (runs on slave machine)

The working of an HDFS is the same as master slave architecture. Here the master is the name node that contains the metadata of the cluster, but the processing occurs through data nodes. The client first connects to the metadata and receives information about the data node and the next time directly connects to the data node. GFS works the same way as well.


NameNode
BackupNode
DataNode
DataNode
DataNode
DataNode
DataNode

Comparison of features

Hadoop Distributed File System HDFS	Google File System GFS
Cross Platform	Linux
Developed in Java environment	Developed in C,C++ environment
Initially it was developed by Yahoo and now its an **open source Framework**	It was developed & still owned by Google
It has Name node and Data Node	It has Master-node and Chunk server
128 MB will be the default block size	**64 MB** will be the default block size
Name node receive heartbeat from Data node	Master node receive heartbeat from Chunk server
Commodity hardware are used	Commodity hardware are used
''Write Once and Read Many" times model	Multiple writer , multiple reader model
Deleted files are renamed into particular folder and then it will removed via garbage	Deleted files are not reclaimed immediately and are renamed in hidden name space and it will deleted after three days if it's not in use
Edit Log is maintained	Operational Log is maintained
Only append is possible	Random file write possible