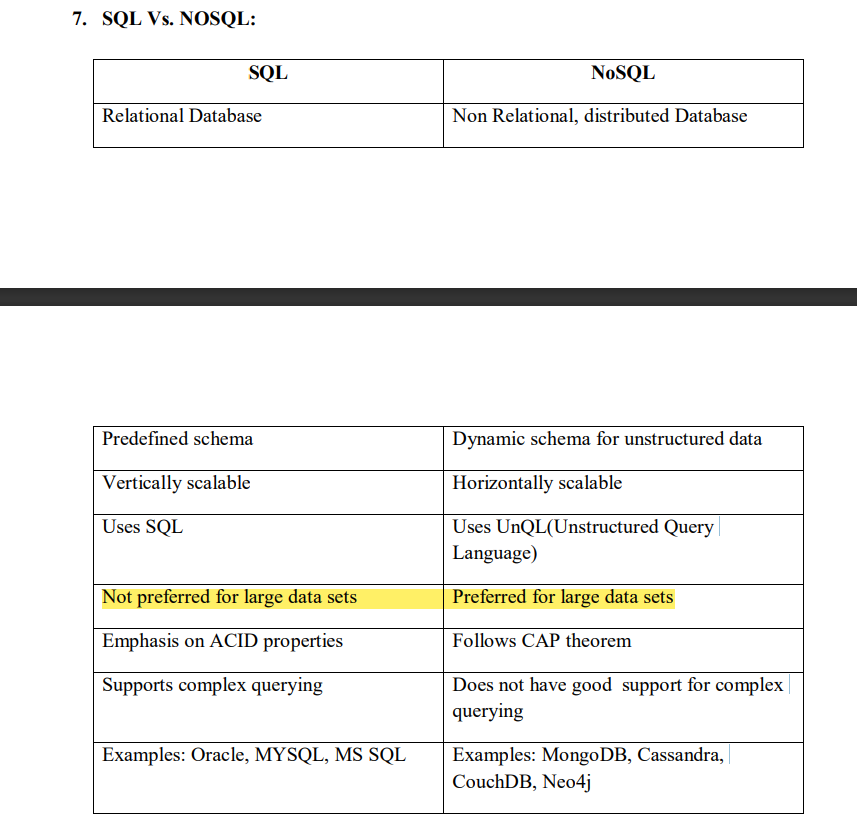
1. **Compare NOSQL databases with SQL databases**



1. **What is NOSQL? Explain briefly any two NoSQL databases**

The NOSQL database was invented by Carlo Strozzi in 1998. It provides a mechanism for retrieval and storage of data. Most of the NOSQL databases are nonrelational, distributed and do not follow ACID properties. Examples of NoSQL databases are MongoDB, Apache Cassandra and Neo4j. NOSQL is being put to use for varied reasons in industries. They are used to support analysis for applications such as web user data analysis, log analysis , sensor feed analysis, making recommendations for upsell and cross sell.

**Key-Value Databases:**

Key-value databases are the simplest form of NoSQL databases. These databases store data in the form of key-value pairs. The keys are used to identify uniquely the values stored in the database. Applications that want to store data generate unique keys and submit the key-value pairs to the database. The database uses the key to determine where the value should be stored. Most key-value databases have distributed architectures comprising of multiple storage nodes. The data is partitioned across the storage nodes by the keys. For determining the partitions for the keys, hash functions are used. The partition number for a key is obtained by applying a hash function to the key. The hash functions are chosen such that the keys are evenly distributed across the partitions.

**Column Family Databases**

In column family databases the basic unit of data storage is a column, which has a name and a value. A collection of columns makes up a row which is identified by a row-key. Columns are grouped together into columns families. Unlike, relational databases, the column family databases do not need to have fixed schemas and a fixed number of columns in each row. The number of columns in a column family database can vary across different rows. A column family can be considered as a map having key-value pairs and this map can vary across different rows. Column family databases store data in a denormalized form so that all relevant information related to an entity required by the applications can be retrieved by reading a single row. Column family databases support high-throughput reads and writes and have distributed and highly available architectures.

1. **What is Hadoop? Explain the components of Hadoop ecosystem.**

Hadoop plays an integral part in almost all big data processes. Hadoop is a framework

that allows for distributed processing of large data sets across clusters of computers using

simple programming models. It is designed to scale up from single servers to thousands of

machines, each offering local computation and storage. In other words, Hadoop is a software

library that allows its users to process large datasets across distributed clusters of computers

thereby enabling them to gather, store and analyse huge data sets. Hadoop provides various

tools and technologies, collectively known as Hadoop ecosystem, to enable development and deployment of big data solutions.

Hadoop ecosystem is a framework of various types of complex tools and components. These elements may be very different from each other in terms of their architecture, but they all derive their functionality from the scalability and power of Hadoop. Hadoop ecosystem can thus be defined as a comprehensive collection of tools and technologies that can be effectively implemented and deployed to provide big data solutions in a cost-effective manner.

MapReduce and Hadoop Distributed File System (HDFS) are the two core components of Hadoop ecosystem that help to manage Big Data.

* **MapReduce**, a framework for handling vast amounts of data
* **Hadoop Distributed File System (HDFS)**, a sophisticated file-handling system.
* **YARN,** a Hadoop resource manager.

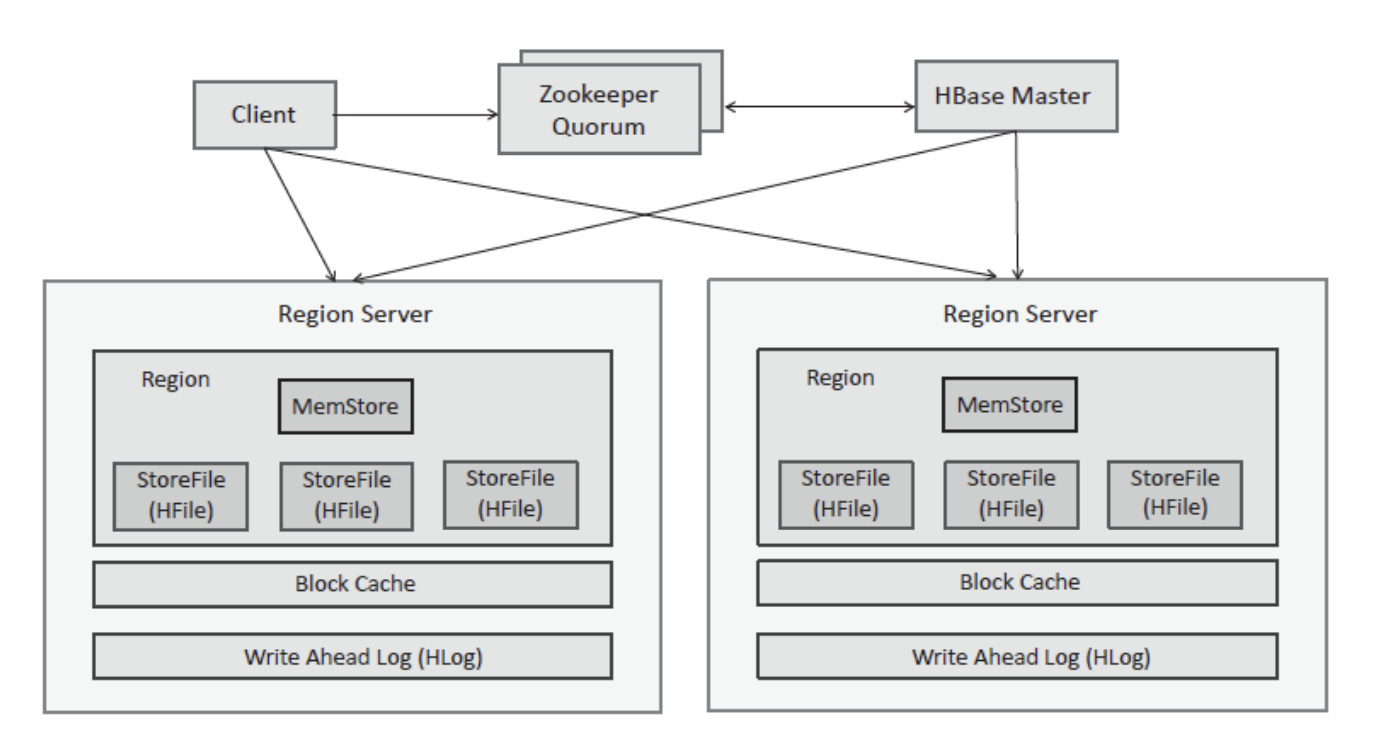
In addition to these core elements of Hadoop, Apache has also delivered other kinds of accessories or complementary tools for developers. These include

* **Apache Hive**, a data analysis tool;
* **Apache Spark**, a general engine for processing big data;
* **Apache Pig,** a data flow language;
* **HBase,** a database tool;
* **Ambari,** which can be considered as a Hadoop ecosystem manager, as it helps to administer the use of these various Apache resources together.

With Hadoop becoming the de facto standard for data collection and becoming ubiquitous in many organizations, managers and development leaders are learning all about the Hadoop ecosystem and what kinds of things are involved in a general Hadoop setup

1. **Explain Hbase architecture.**

HBase (Hadoop Database) is a columnar database built on top of the HDFS. Being a file system, HDFS lacks the random read and write capability. It is when HBase steps in and provides fast record lookups in large tables.

****

HBase has a distributed architecture. An HBase deployment comprises multiple region servers which usually run on the same machines as the Hadoop data nodes. HBase tables are partitioned by the row key into multiple regions (HRegions). Each region server has multiple regions. HBase has a master-slave architecture with one of the nodes acting as the master node (HMaster) and other nodes are slave nodes. The HMaster is responsible for maintaining the HBase meta-data and assignment of regions to region servers. HBase uses Zookeeper for distributed state coordination.

Each Region Server stores two types of files - a store file (HFile) and a write-ahead log (HLog). The HFile contains a variable number of data blocks and the fixed blocks for file

information and trailer. Each data block contains a magic number and multiple key-value

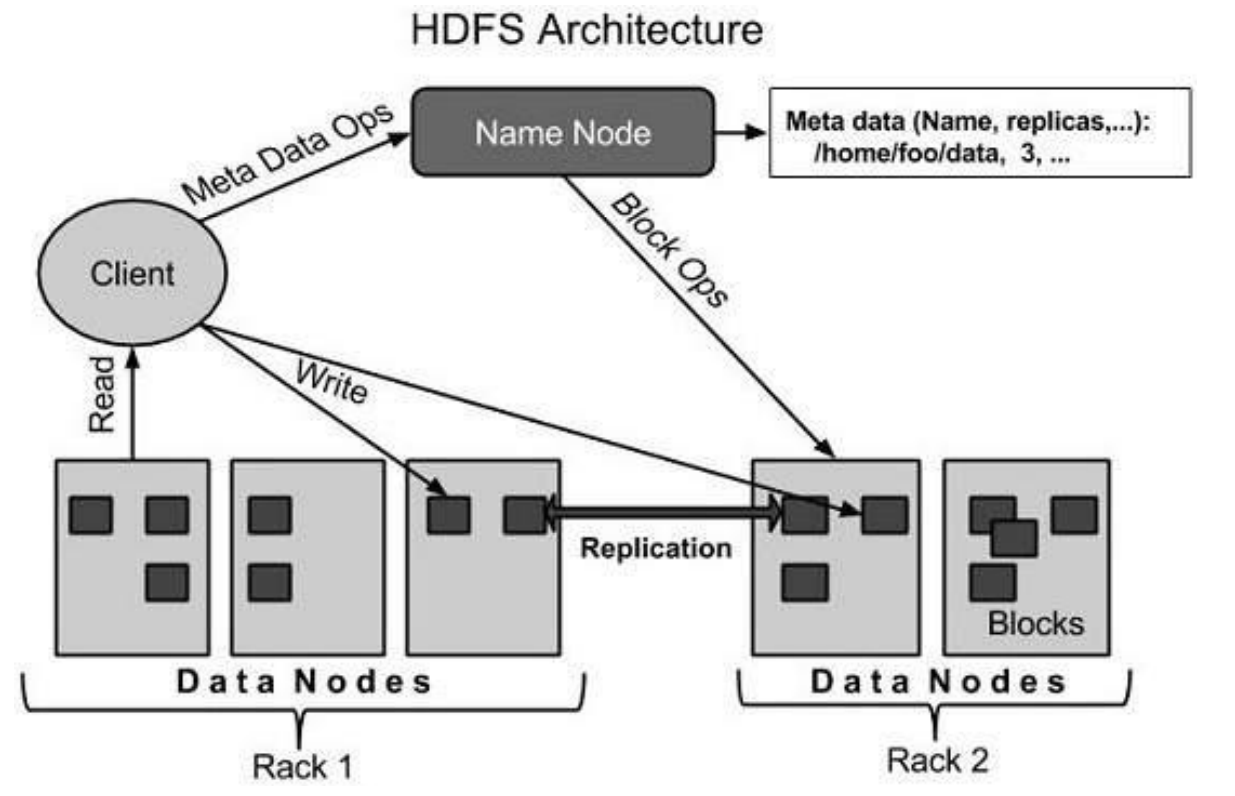
pairs. The default size of a data block is 64 KB. The index block stores the offset for the data

and the meta-blocks. The trailer stores pointers to other blocks. HFiles are persisted on

HDFS. Each Region Server has a Memstore and a Block Cache. The Memstore stores the recent edits to the data in memory and the Block Cache caches the data blocks. Each Region Server also maintains a write-ahead log (WAL) known as the Hlog which logs the writes (that are also written to Memstore). Since HLog is stored on HDFS, it ensures that even in the event of loss of Memstore (which is an in-memory buffer), the writes are never lost. Each Region Server has a Block cache, which is an in-memory store that caches the most recently used blocks for fast lookup.

1. **Explain the HDFS architecture with a neat diagram**

HDFS follows the master-slave architecture. It has a NameNode and a number of DataNodes. The NameNode is a master that manages the various DataNodes.



**NameNode:**

The NameNode is the commodity hardware that contains the GNU/Linux operating system and the NameNode software. It is a software that can be run on commodity hardware. The NameNode manages the HDFS cluster metadata. The records and directories are managed on the NameNode. Operations on them, such as modification, opening or closing are performed by NameNode . The system having the NameNode acts as the master server and it does the following tasks:

* Manages the file system namespace.
* Regulates client’s access to files.
* It also executes file system operations such as renaming, closing, and opening files and directories.

**DataNode:**

The DataNode is a commodity hardware having the GNU/Linux operating system and

DataNode software. For every node (Commodity hardware/System) in a cluster, there

will be a DataNode. These nodes manage the data storage of their system. DataNodes

perform read-write operations on the file systems, as per client request. They also

perform operations such as block creation, deletion, and replication according to the

instructions of the NameNode.

**Block:**

Generally, the user data is stored in the files of HDFS. The file in a file system will be

divided into one or more segments and/or stored in individual data nodes. These file

segments are called as blocks. In other words, the minimum amount of data that

HDFS can read or write is called a Block. The default block size is 64MB, but it can

be increased as per the need to change in HDFS configuration.

1. **List and explain the features of Hadoop**

Hadoop can handle large volumes of structured and unstructured data more efficiently

than the traditional enterprise data warehouse. The power of Hadoop lies in its distributed processing. Hadoop is open source and therefore, it can run on commodity hardware. That means the initial cost savings are dramatic with Hadoop while it can continue to grow as your organizational data grows.

**Here are a few key features of Hadoop:**

1. **Hadoop Brings Flexibility in Data Processing:**

One of the biggest challenges organizations have had in that past was the challenge of

handling unstructured data. Let’s face it, only 20% of data in any organization is

structured while the rest is all unstructured whose value has been largely ignored due

to lack of technology to analyze it. Hadoop manages data whether structured or unstructured, encoded or formatted, or any other type of data. Hadoop brings the value to the table where unstructured data can be useful in decision making process.

1. **Hadoop Is Easily Scalable**

This is a huge feature of Hadoop. It is an open-source platform and runs on industry standard hardware. That makes Hadoop extremely scalable platform, where new nodes can be easily added in the system as and data volume of processing needs grow without altering anything in the existing systems or programs.

1. **Hadoop Is Fault Tolerant**

In Hadoop, the data is stored in HDFS where data automatically gets replicated at two

other locations. So, even if one or two of the systems collapse, the file is still available

on the third system at least. This brings a high level of fault tolerance. The level of

replication is configurable and this makes Hadoop incredibly reliable data storage system. This means, even if a node gets lost or goes out of service, the system automatically reallocates work to another location of the data and continues processing as if nothing had happened.

1. **Hadoop Is Great at Faster Data Processing**

While traditional ETL and batch processes can take hours, days, or even weeks to load large amounts of data, the need to analyze that data in real-time is becoming critical day after day. Hadoop is extremely good at high-volume batch processing because of its ability to do parallel processing. Hadoop can perform batch processes 10 times faster than on a single thread server or on the mainframe.

1. **Hadoop Ecosystem Is Robust:**

Hadoop has a very robust ecosystem that is well suited to meet the analytical needs of

developers and small to large organizations. Hadoop Ecosystem comes with a suite of

tools and technologies making it very much suitable to deliver to a variety of data

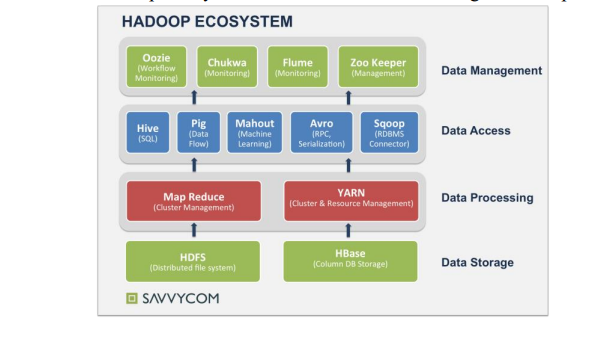
processing needs. Just to name a few, Hadoop ecosystem comes with projects such as

MapReduce, Hive, HBase, Zookeeper, HCatalog, Apache Pig etc. and many new tools and technologies are being added to the ecosystem as the market grows.

1. **Hadoop Is Very Cost Effective:**

Hadoop generates cost benefits by bringing massively parallel computing to commodity servers, resulting in a substantial reduction in the cost per terabyte of storage, which in turn makes it reasonable to model all your data. Apache Hadoop was developed to help Internet-based companies deal with prodigious volumes of data. According to some analysts, the cost of a Hadoop data management system, including hardware, software, and other expenses, comes to about $1,000 a terabyte, about one-fifth to one-twentieth the cost of other data management technologies.

1. **Explain the layers of Hadoop ecosystem with a neat diagram**



1. **Data Storage Layer:**

* **HDFS (Hadoop Distributed File System)** is the key component that makes up Hadoop. HDFS is used to store and access huge file based on client/server architecture. This system also enables the distribution and storage of data across Hadoop clusters.
* **HBase (Hadoop Database)** is a columnar database built on top of the HDFS. Being a file system, HDFS lacks the random read and write capability. It is when HBase steps in and provides fast record lookups in large tables.

1. **Data Processing Layer:**

* **MapReduce** is a parallel data processing framework over clusters. Using MapReduce can help data seeker save a lot of time, for example, if it takes a normal relational database around 20 hours to process a large data set, it might take MapReduce only around three minutes to get everything done.
* **YARN (Yet Another Resource Negotiator)** is a resource manager. It is said to be the second generation of MapReduce and also a critical advancement from Hadoop 1. YARN acts the role of an operating system; its jobs is to manage and monitor workloads, make sure it can serve multiple clients and perform security controls. In addition, YARN supports new processing models that MapReduce does not.

1. **Data Access Layer:**

* **Hive** is new kind of structured query language. It was born to help who are familiar with the traditional database and SQL to leverage Hadoop and MapReduce.
* **Pig** serves the analysis purpose for large data sets. Pig is made up of two components, firstly the platform to execute Pig programs; secondly, a powerful and simple scripting language called Pig Latin, which is used to write those programs.
* **Mahout** provides a library of the most popular machine learning algorithms written in Java that supports collaborative filtering, clustering, and classification.
* **Arvo** is a data serialization system. It uses JSON for defining data types and protocols to support data-driven applications. Arvo provides a simple integration with many different languages with the expectation to support Hadoop application to be written in other languages (e.g., Python, C++) rather than Java.
* **Sqoop (SQL + Hadoop = Sqoop)** is a command line interface application, which helps transfer data between Hadoop and relational databases (e.g., MySQL or Oracle) or mainframes.

1. **Data Management Layer:**

* **Oozie** is a workflow scheduler for Hadoop. Oozie streamlines the process of creating workflows and managing coordination jobs among Hadoop and other applications such as Map Reduce, Pig, Sqoop, Hive etc.

**The main responsibilities of Oozie are**:

* + - * + to define a sequence of actions to be executed.
        + to place triggers for those actions.
* **Chukwa** is another framework that is built on top of HDFS and Map Reduce. Its purpose is to provide a dynamic and powerful data collection system. Chukwa is capable of monitoring, analyzing and presenting the results to get the most out of collected data.
* **Flume** is also a scalable and reliable system for collecting and moving cluster logs from various sources to a centralized store like Chukwa. However, there are some differences. In Flume, chunks of data are transferred from node to node in store and forward manner; while in Chukwa, the agent of each machine will need to determine what data to be sent.
* **ZooKeeper** is a distributed coordination service for distributed system. It provides a very simple programming interface and helps reduce the management complexity by providing services such as configuration, distributed synchronization, naming, group services etc