**Abuchi Okeke**

Big Data Training (EIT)

Interview Questions:

1. What is Big Data?

Ans: Big data is a field of technology that dealt or analyses complex problems that can’t be handled by the traditional way of data management.

1. 5V’s of Big Data?

Ans:

1. **Volume:** amount of data
2. **Velocity:** speed of accumulation of data
3. **Variety**: It refers to nature of data that is structured, semi-structured and unstructured data.
4. **Veracity:** It refers to nature of data that is structured, semi-structured and unstructured data**.**
5. **Value:** usefulness of the data (this is the most import of all the V’s)
6. Default number of mappers in Sqoop?

Ans: Fours mappers

1. What is fault tolerance in Hadoop?

Ans:

Fault tolerance is the reassignment of tasks when there is a system failure. It can be achieved by replication (creating replica of users’ data on different machines in the HDFS cluster with a replication factor of 0.45) and rack- awareness (the concept of choosing the closest data node for serving a purpose for better node communication).

1. File formats in Hadoop

Ans:

1. **CSV** : semi-structure data, comma separated, tab delimeter
2. **JSON**: semi-structured (key-value pair)
3. **XML**: parent and child tag (semi-structured)
4. **Sequence file** format (Sequence file stores data in binary format and has a similar structure to CSV file with some differences. It also doesn’t store metadata and so only schema evolution option is appending new fields, but it supports block compression.

Due to complexity, sequence files are mainly used in flight data as an intermediate storage.

1. **AVRO files:** similar to JSON format with header and body. Metadata are stored in the header while body holds binary information of the data.

It is a row major file format (RMFF)

Avro is serialized (writing data to the disk) and deserialized (reading data from the disk)

It has good compression ratio

It has schema evolution.

1. **RC:** Row columnar.

Rows are divided into groups.

Stores statistic info for each group in the dataset in the footer.

Doesn’t support schema evolution.

If you want to add anything to RC file you will have to rewrite the file. Also, it is a slower process.

1. ORC: is the compressed version of RC file and supports all the benefits of RC file with some enhancements like ORC files compress better than RC files, enabling faster queries.

Rows are divided into stripes.

Doesn’t support schema evolution.

Some benchmarks indicate that ORC files compress to be the smallest of all file formats in Hadoop.

1. PARQUET

Similar to ORC (columnar file).

Handles nested data very well.

Parquet also enjoys the features like compression and query performance benefits but is generally slower to write than non-columnar file formats.

Maintains statistic info in the footer.

1. How will the number of mappers be decided in MapReduce framework?

**Ans:**

Number of mappers is determined by number of splits which is equivalent to the number of blocks.

1. Explain ACID transaction?

Ans:

A - Atomicity: all or nothing that is the transaction should be complete else it should fail.

C - Consistency: the value should be the same and consistent.

I – Isolation: the transaction should not be visible to other session.

D – Durability: persistence (data committing).

1. Daemons available in Hadoop?

Ans:

* Name Node
* Secondary Name Node
* Data Node
* Resource Manager
* Node Manager

1. What are type of data available structure, semi-structure and unstructured data?

Ans:

Structure: two-dimensional data (tables), have schema eg. RDBMS data

Semi Structure: XML, CSV, JSON

Unstructured: audio files, video files

1. Explain Hadoop Architecture?

Ans:

Hadoop follows a Master Slave architecture for the transformation and analysis of large datasets using Hadoop MapReduce paradigm. The 3 important hadoop components that play a vital role in the Hadoop architecture are -

1. Hadoop Distributed File System (HDFS) – Patterned after the UNIX file system
2. Hadoop MapReduce
3. Yet Another Resource Negotiator (YARN)

Diagram

Description automatically generated

A screen shot of a computer

Description automatically generated

1. Explain Yarn Architecture?

Ans:

YARN stands for “Yet Another Resource Negotiator“. It was introduced in Hadoop 2.0 to remove the bottleneck on Job Tracker which was present in Hadoop 1.0. YARN was described as a “Redesigned Resource Manager” at the time of its launching, but it has now evolved to be known as large-scale distributed operating system used for Big Data processing.

YARN architecture basically separates resource management layer from the processing layer. In Hadoop 1.0 version, the responsibility of Job tracker is split between the resource manager and application manager.

Diagram

Description automatically generated

1. Explain Serialization and Deserialization

Ans: Serialization (writing data to the disk) and Deserialization (reading data from the disk)

1. Function of the secondary name node?

Ans:

Secondary NameNode in hadoop is a specially dedicated node in HDFS cluster whose main function is to take checkpoints of the file system metadata present on namenode.

The secondary NameNode merges the fsimage and the edits log files periodically and keeps edits log size within a limit

1. Differences between Hive and HBase?

**Ans:**

Hive (datawarehousing package) and HBase are two different Hadoop based technologies . Hive is a SQL-like engine that runs MapReduce jobs, and HBase is a NoSQL key/value database on Hadoop. Hive can be used for analytical queries (OLAP) while HBase for real-time querying. Data can even be read and written from Hive to HBase, and back again.

|  |  |
| --- | --- |
| **Hive** | **HBase** |
| Hive is a query engine | Data storage particularly for unstructured data |
| Mainly used for batch processing | Extensively used for transactional processing |
| Not a real time processing | Real-time processing |
| Only for analytical queries | Real-time querying |
| Runs on the top of Hadoop | Runs on the top of HDFS (Hadoop distributed file system) |
| Apache Hive is not a database | It support NoSQL database |
| It has schema model | It is free from schema model |
| Made for high latency operations | Made for low level latency operations |

1. Two types of partitions in Hadoop?

**Ans:**

**Static partition:** In the static partitioning mode, you can insert or input the data files individually into a partition table. You can create new partitions as needed, and define the new partitions using the ADD PARTITION clause.

**Dynamic partition:** With dynamic partitioning in hive, partitions get created automatically at load times. New partitions can be created dynamically from existing data.

Partitions are automatically created based on the value of the last column. If the partition does not already exist, it will be created.

1. Performance tuning mechanism in Hive?

Ans:

1. Changing execution engine (map reduce, spark or TEZ).
2. Partitioning and Bucketing.
3. File Formats.
4. Joins (Map-side Joins, Bucketed joins and sort merge bucketed join)
5. Types of joins in Hive?

Ans:

Outer join

* Left outer join
* Right outer join
* Full outer join

Advanced Joins:

* Map-side Joins
* Bucketed joins
* Sort merge bucketed join

1. Differences between ORC and AVRO file format?

Ans:

**AVRO:-**

* It is row major format.
* Its primary design goal was schema evolution.
* In the avro format, we store schema separately from data. Generally avro schema file (.avsc) is maintained.

**ORC**

* Column oriented storage format.
* Originally it is Hive's Row Columnar file. Now improved as Optimized RC (ORC)
* Schema is with the data, but as a part of footer.
* Data is stored as row groups and stripes.
* Each stripe maintains indexes and stats about data it stores.

1. What are the daemons needed to be up and running for Hadoop and HBase?

Ans:

* Name Node
* Secondary Name Node
* Data Node
* Resource Manager
* Node Manager
* HQuorumPeer
* HMaster
* HRegionServer

1. Explain HBase Architecture?

Ans:

Diagram

Description automatically generated

HBase architecture consists mainly of the following components

* HMaster
* HRegionserver
* HRegions
* Zookeeper
* HDFS

**HMaster:**

HMaster is the implementation of a Master server in HBase architecture. It acts as a monitoring agent to monitor all Region Server instances present in the cluster and acts as an interface for all the metadata changes. In a distributed cluster environment, Master runs on NameNode. Master runs several background threads.

The following are important roles performed by HMaster in HBase.

* Plays a vital role in terms of performance and maintaining nodes in the cluster.
* HMaster provides admin performance and distributes services to different region servers.
* HMaster assigns regions to region servers.
* HMaster has the features like controlling load balancing and failover to handle the load over nodes present in the cluster.
* When a client wants to change any schema and to change any Metadata operations, HMaster takes responsibility for these operations.

Some of the methods exposed by HMaster Interface are primarily Metadata oriented methods.

* Table (createTable, removeTable, enable, disable)
* ColumnFamily (add Column, modify Column)
* Region (move, assign)

The client communicates in a bi-directional way with both HMaster and ZooKeeper. For read and write operations, it directly contacts with HRegion servers. HMaster assigns regions to region servers and in turn, check the health status of region servers.

In entire architecture, we have multiple region servers. Hlog present in region servers which are going to store all the log files.

**HBase Regions Servers:**

When Region Server receives writes and read requests from the client, it assigns the request to a specific region, where the actual column family resides. However, the client can directly contact with HRegion servers, there is no need of HMaster mandatory permission to the client regarding communication with HRegion servers. The client requires HMaster help when operations related to metadata and schema changes are required.

HRegionServer is the Region Server implementation. It is responsible for serving and managing regions or data that is present in a distributed cluster. The region servers run on Data Nodes present in the Hadoop cluster.

HMaster can get into contact with multiple HRegion servers and performs the following functions.

* Hosting and managing regions
* Splitting regions automatically
* Handling read and writes requests
* Communicating with the client directly

**HBase Regions:**

HRegions are the basic building elements of HBase cluster that consists of the distribution of tables and are comprised of Column families. It contains multiple stores, one for each column family. It consists of mainly two components, which are Memstore and Hfile.

**ZooKeeper:**

In HBase, Zookeeper is a centralized monitoring server which maintains configuration information and provides distributed synchronization. Distributed synchronization is to access the distributed applications running across the cluster with the responsibility of providing coordination services between nodes. If the client wants to communicate with regions, the server's client has to approach ZooKeeper first.

It is an open source project, and it provides so many important services.

Services provided by ZooKeeper

* Maintains Configuration information
* Provides distributed synchronization
* Client Communication establishment with region servers
* Provides ephemeral nodes for which represent different region servers
* Master server’s usability of ephemeral nodes for discovering available servers in the cluster
* To track server failure and network partitions

Master and HBase slave nodes ( region servers) registered themselves with ZooKeeper. The client needs access to ZK(zookeeper) quorum configuration to connect with master and region servers.

During a failure of nodes that present in HBase cluster, ZKquoram will trigger error messages, and it starts to repair the failed nodes.

**HDFS:-**

HDFS is a Hadoop distributed file system, as the name implies it provides a distributed environment for the storage and it is a file system designed in a way to run on commodity hardware. It stores each file in multiple blocks and to maintain fault tolerance, the blocks are replicated across a Hadoop cluster.

HDFS provides a high degree of fault –tolerance and runs on cheap commodity hardware. By adding nodes to the cluster and performing processing & storing by using the cheap commodity hardware, it will give the client better results as compared to the existing one.

In here, the data stored in each block replicates into 3 nodes any in a case when any node goes down there will be no loss of data, it will have a proper backup recovery mechanism.

**HDFS**get in contact with the HBase components and stores a large amount of data in a distributed manner.

1. How will you perform incremental load in SQOOP?

Ans:

Add --incremental append

And -- check column <column name>

Properties to sqoop command

1. What are the features of Hadoop and can you explain them?

Ans.

* Horizontally Scalable
* License free
* Open source
* **Meant for Big Data Analytics**
* **Shared Nothing Architecture:**Hadoop is a shared nothing architecture**,**that means Hadoop is a cluster with independent machines. (Cluster with Nodes), that every node performs its job by using its own resources.
* Handles both storage and processing
* **Distributed File System**
* **Commodity Hardware:**Hadoop can run on commodity hardware that means Hadoop does not require a very high-end server with large memory and processing power. Hadoop runs on JBOD (just bunch of disk), so every node is independent in Hadoop.
* **Distributors (Cloudera (Sandbox), AWS (EMR), Hortonworks (AMBARI), GCP (DATA PROC), AZURE (HD Insight)**

1. Explain ORC format?

Ans:

**ORC**

* Column oriented storage format.
* Originally it is Hive's Row Columnar file. Now improved as Optimized RC (ORC)
* Schema is with the data, but as a part of footer.
* Data is stored as row groups and stripes.
* Each stripe maintains indexes and stats about data it stores.

1. What are the execution engines in Hive?

Ans:

* MapReduce
* Spark
* TEZ

1. Services Hadoop Provides?

**Ans:**

Storage and Processing

1. What is the TEZ filtering process called?

Ans:  
Predicate Pushed on