

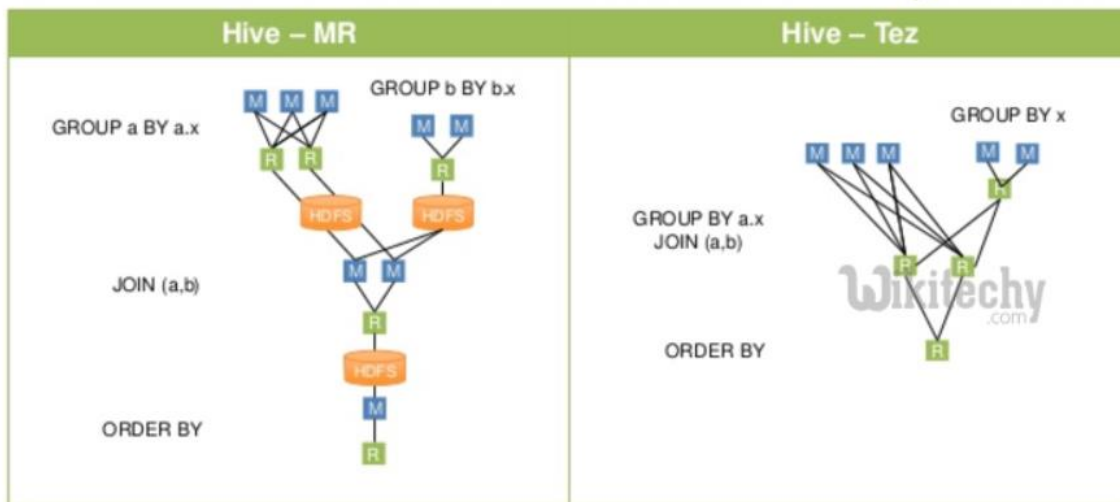
1. Hive-on-MR vs Hive-on-Tez?

Ans: See the below Query for understanding.

Hive-on-MR vs. Hive-on-Tez

```
SELECT g1.x, g1.avg, g2.cnt
FROM (SELECT a.x, AVERAGE(a.y) AS avg FROM a GROUP BY a.x) g1
JOIN (SELECT b.x, COUNT(b.y) AS cnt FROM b GROUP BY b.x) g2
ON (g1.x = g2.x)
ORDER BY avg;
```

Tez avoids unnecessary writes to HDFS



Hive-on-MR is not complete in memory computation. However, Hive-on-Tez is fully in-memory computation.

MR stores intermediate results into Hard disk and for next computation data fetched from Hard disk to RAM.

TEZ stores intermediate results into memory only, which leads less I/O and faster computation.

2. TEZ is in memory computation then why Spark came since both are fully in memory computation?

Ans: Firstly, we need to know that performance wise there is not much difference in TEZ and Spark. Both are much faster than MR.

If both are quite faster than MR then Why Spark and TEZ two separate processing engines.

Spark has few things that TEZ does have.

Like Spark has elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to datasets.

A programmer can write functionality using Scala/Python code very easily in Spark but Hive does not give programmer much flexibility.

If we talk about fault tolerance, then Spark has lineage, which allow recreating the corrupt RDD.

So TEZ and Spark both better compared to MR.

Performance in Spark and TEZ almost similar. But few additional features available in Spark.

Hence, Spark is most preferred Execution Engine in terms of speed and flexibility.

3. How do you describe I/O?

Ans: Communication between hard disk and RAM is called I/O operation in big data.

High I/O means there is more data exchange between hard disk and RAM which is less efficient.

Low I/O considered as good performance

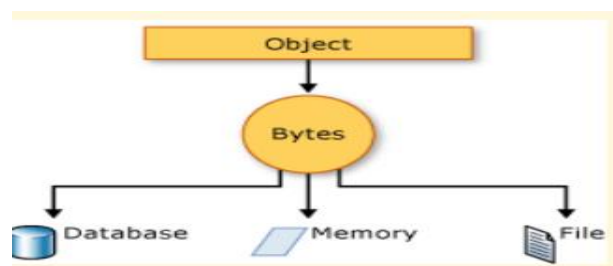
4. Describe serialization and deserialization?

Ans: Serialization is the process of turning an object in memory into a stream of bytes so you can do stuff like store it on disk or send it over the network.

Deserialization is the reverse process: turning a stream of bytes into an object in memory.

5. Why serialization and deserialization is required?

Ans: Serialization is the process of converting an object into a stream of bytes to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate it when needed. The reverse process is called deserialization.



6. Why intermediate results from Hive (MR) operation are not stored in HDFS?

Ans: Storing HDFS means replication. To avoid replication Local Directory is preferred for this.

7. FS Image and Edit logs in Hadoop?

Ans: Name Node stores snapshot of metadata in hard disk as a FS Image file. However, FS image is not only the solution to store whole metadata.

Let us discuss one scenario:

We are not taking back up of metadata every moment in FSImage. We take back up after certain interval in FSImage hence if any failure happens in between we will lose data.

So, here is Edit logs, which takes backup every moment. After a certain interval, it transfer the data to FSImage file and FSImage file adds the new backup along with the old back.

8. What will happen if Name Node fails?

Ans: In the worst case, you lose the metadata of all your files, the metadata cannot be recovered, and you lose all your data.

However, in production systems, usually, high availability is configured. This means that block data is written to both the Active NameNode and the Standby NameNode. In case the Active NameNode fails, there is automatic failover provided by the Standby NameNode

9. Where does Hive stores metadata?

Ans: The Metastore is a Hive component that stores the system catalog containing metadata about Hivecreate tables, columns, and partitions. Usually, metadata is stored in the traditional RDBMS format. However, by default, Apache Hive uses the Derby database to store metadata.

10. Hive UDF VS UDAF VS UDTF?

Ans: <https://www.linkedin.com/pulse/hive-functions-udfudaf-udtf-examples-gaurav-singh/>

UDF is User defined function .Trim() is an UDF. Single row in a table and produces a single row as output. It is one to one relationship.

UDAF is User defined Aggregate function. Sum(), Avg() these are UDAF.Many rows converted to one row hence many : 1 mapping in UDAF.

UDTF is User defined Tabular function. Explode () is an UDTF. 1 : Many mapping happens here.

11. Hive Complex data type?

Ans:

Arrays: It is an ordered collection of elements. The elements in the array must be of the same type.

map: It is an unordered collection of key-value pairs. Keys must be of primitive types. Values can be of any type.

struct: It is a collection of elements of different types.

Examples: complex datatypes

ARRAY:

```
$ cat >arrayfile
1,abc,40000,a$b$c,hyd
2,def,3000,d$f,bang
```

```
hive> create table tab7(id int,name string,sal bigint,sub array<string>,city string)
> row format delimited
> fields terminated by ','
> collection items terminated by '$';
```

```
hive>select sub[2] from tab7 where id=1;
```

```
hive>select sub[0] from tab7;
```

MAP:

```
$ cat >mapfile
1,abc,40000,a$b$c,pf#500$epf#200,hyd
2,def,3000,d$f,pf#500,bang
```

```
hive>create table tab10(id int,name string,sal bigint,sub array<string>,dud
map<string,int>,city string)
row format delimited
fields terminated by ','
collection items terminated by '$'
map keys terminated by '#';
```

```
hive> load data local inpath '/home/training/mapfile' overwrite into table tab10;
```

```
hive>select dud["pf"] from tab10;
```

```
hive>select dud["pf"],dud["epf"] from tab10;
```

STRUCT:

```
cat >mapfile
1,abc,40000,a$b$c,pf#500$epf#200,hyd$ap$500001
2,def,3000,d$f,pf#500,bang$kar$600038
```

```
hive> create table tab11(id int,name string,sal bigint,sub array<string>,dud
map<string,int>,addr struct<city:string,state:string,pin:bigint>)
> row format delimited
> fields terminated by ','
> collection items terminated by '$'
> map keys terminated by '#';
```

```
hive> load data local inpath '/home/training/structfile' into table tab11;
```

```
hive>select addr.city from tab11;
```

12. Can we parse log file data and store into Hive?

Ans: Using regex serde we can parse log file data and store into Hive

13. Why Hive cannot delete data from external table?

Ans: Local Path: "D:\data files\sample.txt". Hive path: "/usr/hive/warehouse "

Generally, external table stores data in local path, which Hive does not access. Hence, data cannot be deleted. If path is given to hive then data will be deleted along with metadata.

14. Partitioning and bucketing in Hive?

Ans: Partitioning and bucketing both are optimization technique in Hive. Both escape table level scan.

If we want data is stored in equal volume then go with bucketing

But in partitioning each partition does not have same volume of data.

Bucketing rule: Column value % No of buckets

Partitioning rule: duplicate value goes to same partition. No logic applied on value.

Partitioning dir is created for partitioning but for bucketing files are created

15. How Partitioning is decided in one column?

Ans: partition decided based on cardinality

16. Explode and Lateral View?

Ans: Array containing multiple value which requires to be converted into rows then we use explode but we need lateral view with explode function when we want to view more than one column.

Amar	["Btech", "Mtech"]
Amala	["Bsc", "Msc", "Mtech"]
Akash	["Btech", "Mba"]

```
select name, myq from <table> lateral view explode(qual) q as myq;
```

17. Collect_Set in Hive?

Ans: Collect_Set is exactly opposite to explode function. Multiple rows are converted into array.

Collect_Set discards duplicate entry but Collect_List keeps Duplicate entry.

18. If N jobs submitted then how many Application Master will launch?

Ans: N

19. Application Master and Node Manager which one is Master and which one is Slave?

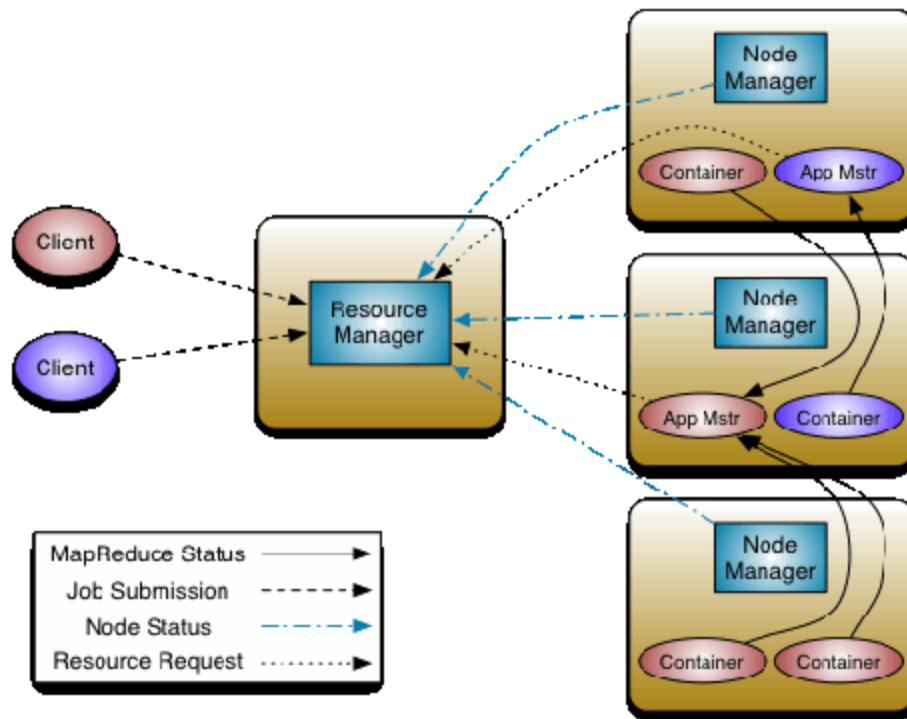
Ans: Application Master -> Master

Node Manager-> Slave

20. Explain lifecycle of a YARN application/job?

Ans:

Now that we know a bit of basics about all of the YARN components – RM, NM, AM, AsM and Containers, let's see how they participate in lifecycle of a YARN application/job using below image from Apache Hadoop site.



Client submits a new application/job request to RM

RM (AsM) accepts the job request. Based on the cluster wide resource stats RM (AsM) finds a slave node that is not too busy. RM (AsM) instructs the NM on this node to launch a container to be used as AM for the given application/job. AM can be thought of as a specialized container, which will manage/monitor application/job tasks.

AM sends request to RM, asking for resources required to run the application/job.

RM responds back with a list of containers along with list of slave nodes that they can be spawned on.

AM starts the containers on each of specified slave nodes; it is also responsible for providing progress status to client whenever requested.

21. Hadoop we know there is Secondary Name Node then why we need Standby Name Node?

Ans: Standby Name Node has come in second version of Hadoop. It prevents SPOF (Single Point of Failure).

Secondary Name Node is a helper of Name Node. It is not replica of Name Node.

But, Standby Name Node is a replica of Name Node. Zookeeper interacts between Secondary Name Node and Standby Name Node.

22. What is hivevar and hiveconf?

Ans: The --hivevar variables are accessed using \${var-name}, while the --hiveconf are accessed \${hiveconf: var-name} inside hive.

E.g. Below examples access variable and print it's value in hive.

hivevar:

```
hive --hivevar a='this is a' -e '!echo ${a};'
```

output: this is a

hiveconf:

```
hive --hiveconf a='this is a' -e '!echo ${hiveconf: a};'
```

output: this is a

The hiveconf namespace and --hiveconf should be used to set Hive configuration values.

The hivevar namespace and --hivevar should be used to define user variables.

Setting user variables under the hiveconf namespace probably won't break anything, but isn't recommended.

23. Msck repair table?

Ans: Msck repair table is used to add partitions that exist in HDFS but not in the hive Metastore.

To update metadata we generally use Msck repair table.

24. What is SCD (Slowly Changing Dimension) and write a query to achieve SCD2?