

General Hadoop Questions



1

What are the different vendor specific distributions of Hadoop?

cloudera



IBM InfoSphere



What are the different Hadoop configuration files?



hadoop-env.sh

mapred-site.xml

core-site.xml

yarn-site.xml

hdfs-site.xml

Master and
slaves

3

What are the 3 modes in which Hadoop can run?

1

Standalone mode

This is the default mode. It uses the local filesystem and a single Java process to run the Hadoop services

2

Pseudo-distributed mode

It uses a single node Hadoop deployment to execute all the Hadoop services

3

Fully-distributed mode

It uses separate nodes to run Hadoop master and slave services

4

What are the differences between Regular file system and HDFS?

Regular File System

1

Data is maintained in a single system

2

If the machine crashes, data recovery is very difficult due to low fault tolerance

3

Seek time is more and hence it takes more time to process the data

HDFS

1

Data is distributed and maintained on multiple systems

2

If a datanode crashes, data can still be recovered from other nodes in the cluster

3

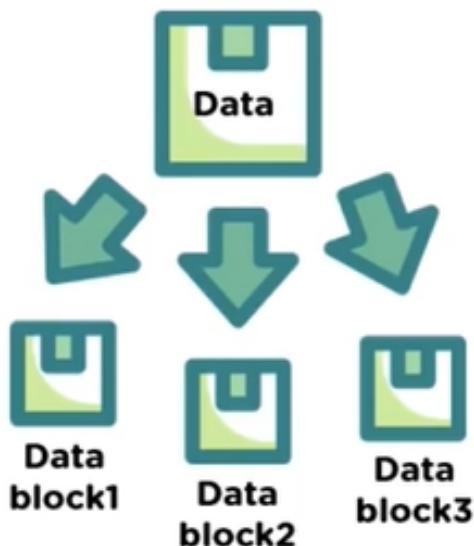
Time taken to read data is comparatively more as there is local data read to disc and coordination of data from multiple systems

HDFS Questions



Why is HDFS fault tolerant?

HDFS is fault tolerant as it replicates data on different datanodes. By default, a block of data gets replicated on 3 datanodes.

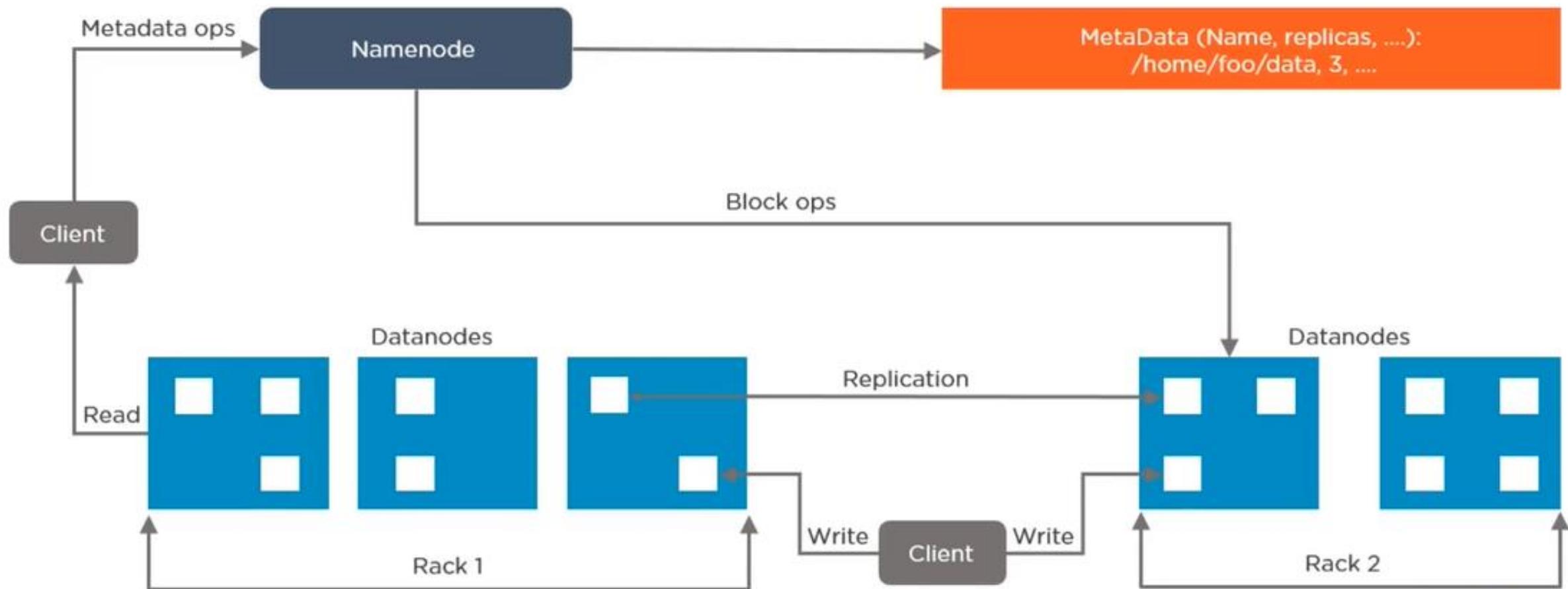


Data gets divided into **multiple blocks**

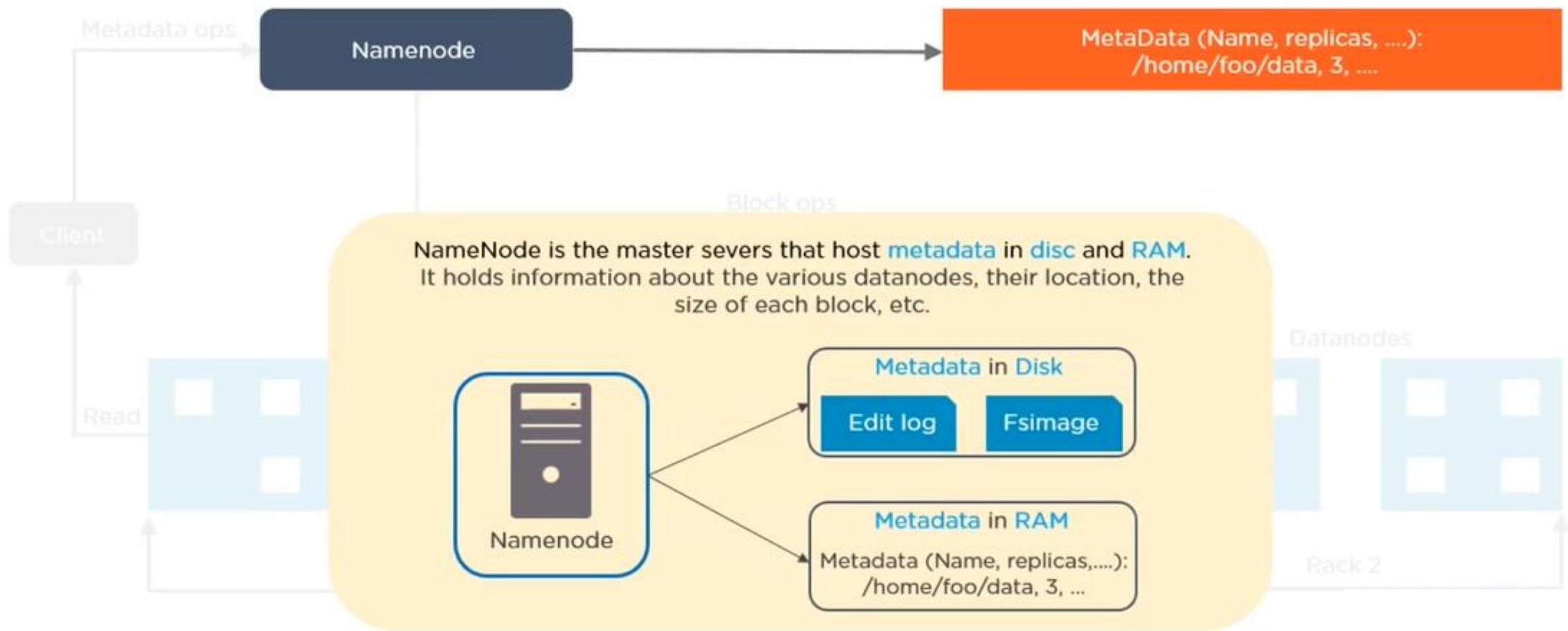


Data blocks are stored in **different datanodes**. If one node crashes, the data can still be retrieved from other datanodes. This makes HDFS fault tolerant

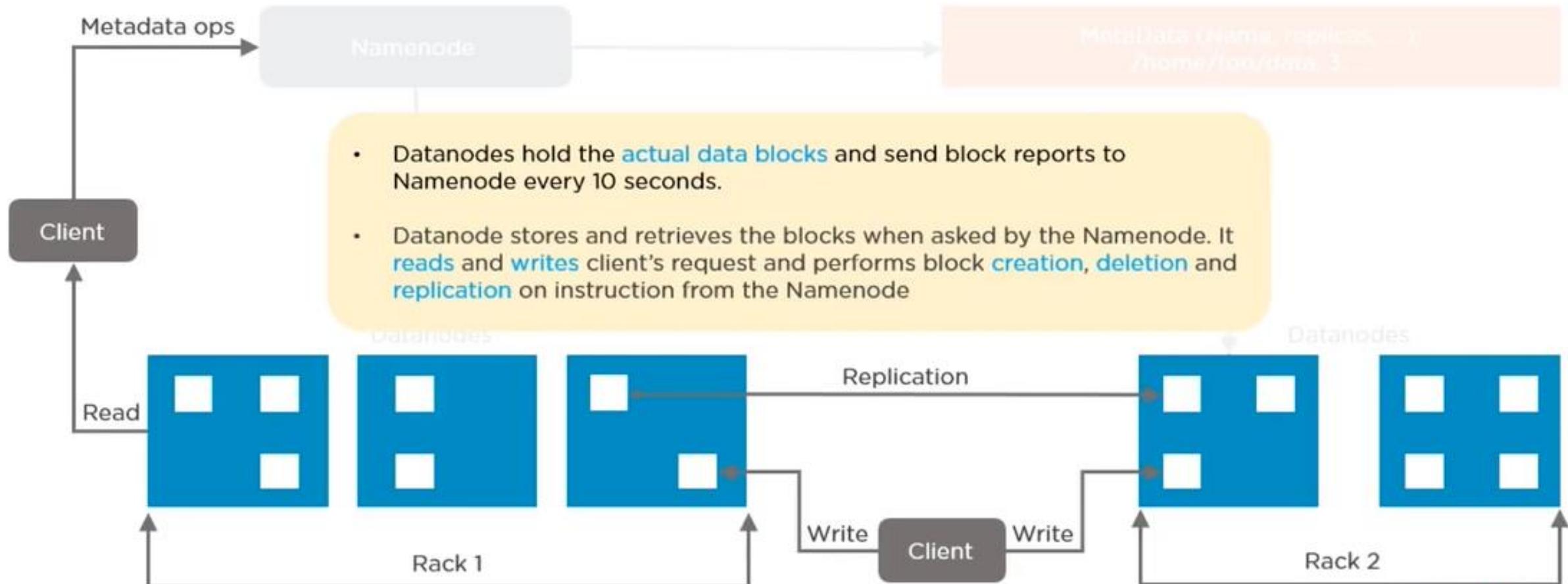
Explain the architecture of HDFS.



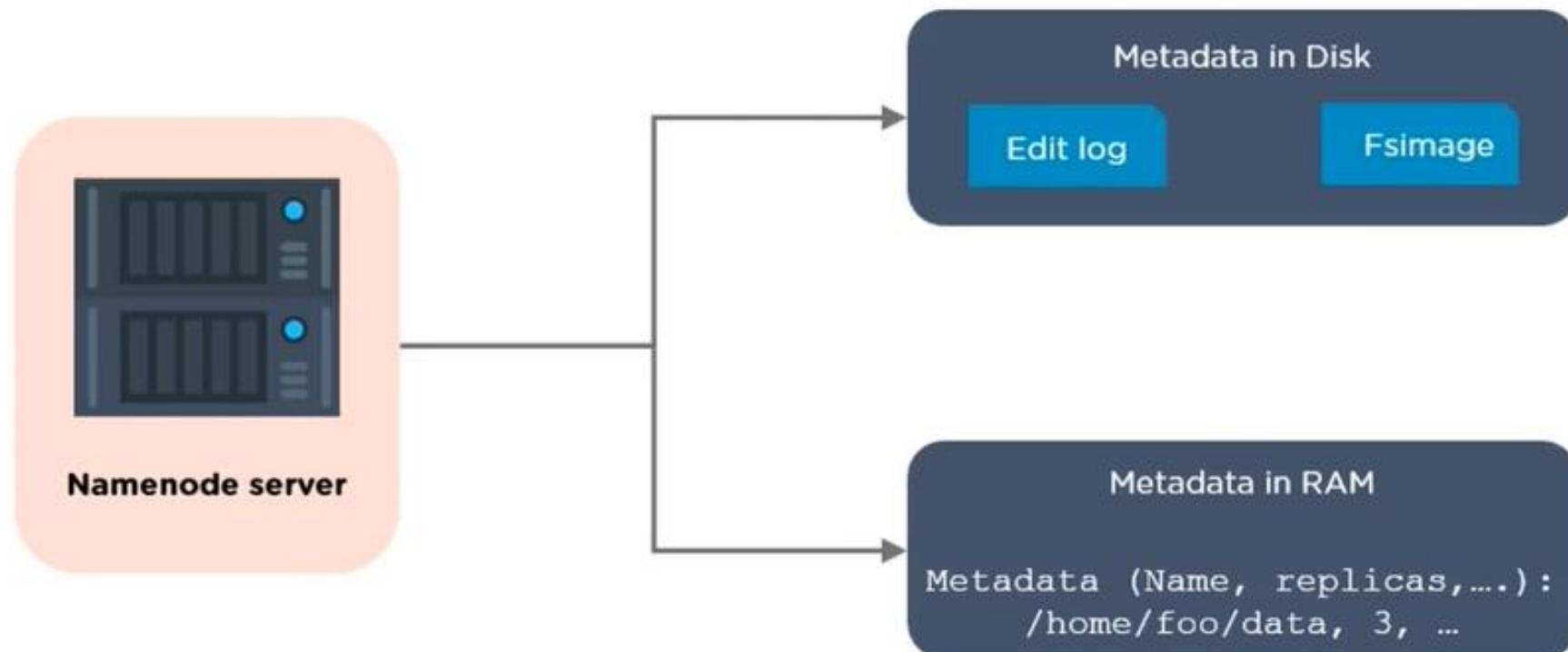
Explain the architecture of HDFS.



Explain the architecture of HDFS.



What are the 2 types of metadata a Namenode server holds?



What is difference between Federation and High Availability?

HDFS Federation

There is no limitation to the number of namenodes and the namenodes are not related to each other

All the namenodes share a pool of metadata in which each namenode will have its dedicated pool

Provides fault tolerance i.e. if one namenode goes down, that will not affect the data of the other namenode

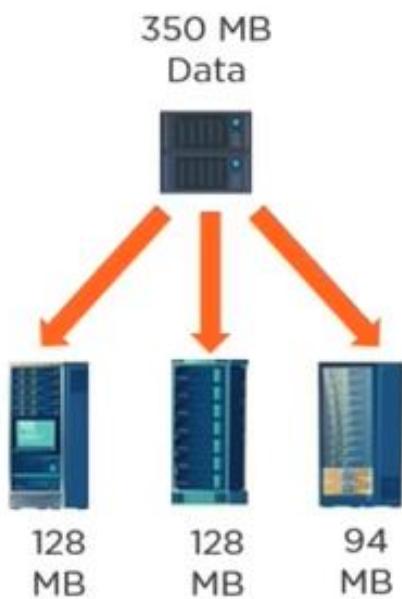
HDFS High Availability

There are 2 namenodes which are related to each other. Both active and standby namenodes work all the time

At a time, active namenode will be up and running while standby namenode will be idle and updating it's metadata once in a while

Requires two separate machines. On first, the active namenode will be configured while the secondary namenode will be configured on the other system

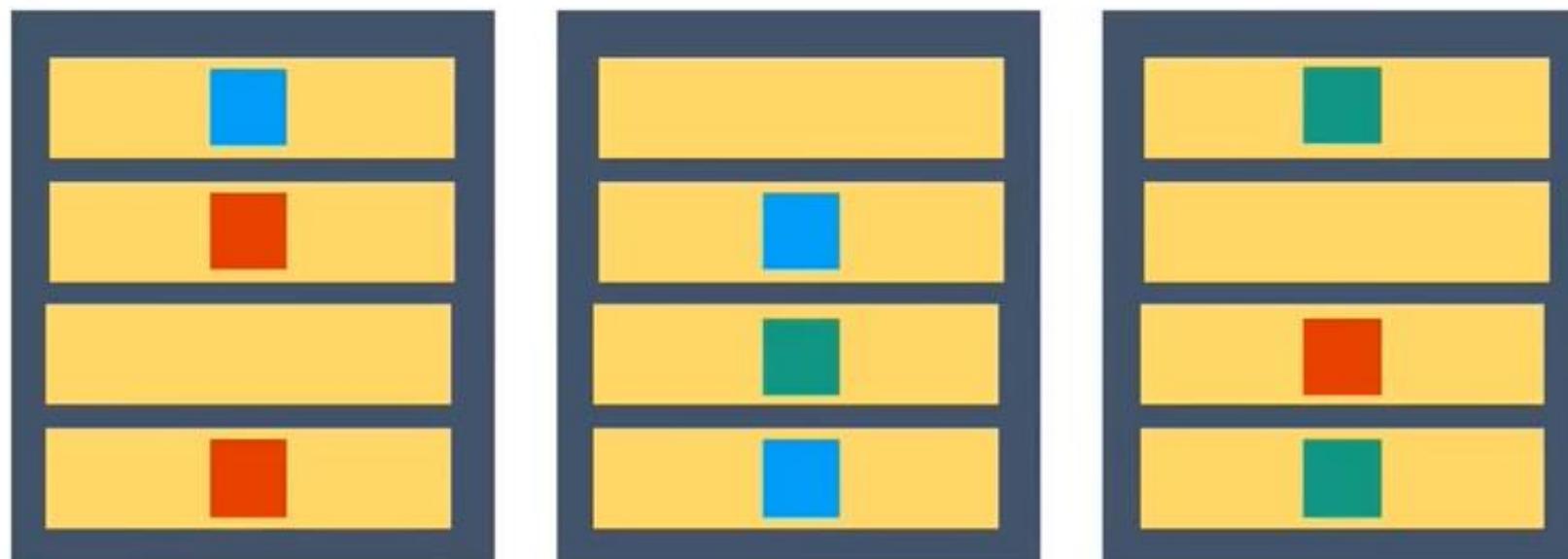
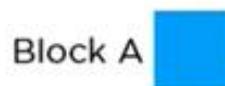
If you have an input file of 350 MB, how many input splits will be created by HDFS and what is the size of each input split?



- Each block by default is divided into **128 MB**.
- The size of all blocks except the last block will be 128 MB.
- So, there are **3 input splits** in total.
- The size of each split is **128 MB, 128 MB** and **94 MB**.

How does Rack Awareness work in HDFS?

HDFS Rack Awareness is about having knowledge of different data nodes and how it is distributed across the racks of a Hadoop Cluster



By default, each block of data gets replicated thrice on various datanodes present on different racks

11

How can you restart Namenode and all the daemons in Hadoop?

Following are the methods to do so:

1

Stop the Namenode with `./sbin/Hadoop-daemon.sh stop namenode` and then start the Namenode using `./sbin/Hadoop-daemon.sh start namenode`

2

Stop all the daemons with `./sbin/stop-all.sh` and then start the daemons using `./sbin/start-all.sh`

Which command will help you find the status of blocks and filesystem health?

`hdfs fsck <path> -files -blocks`



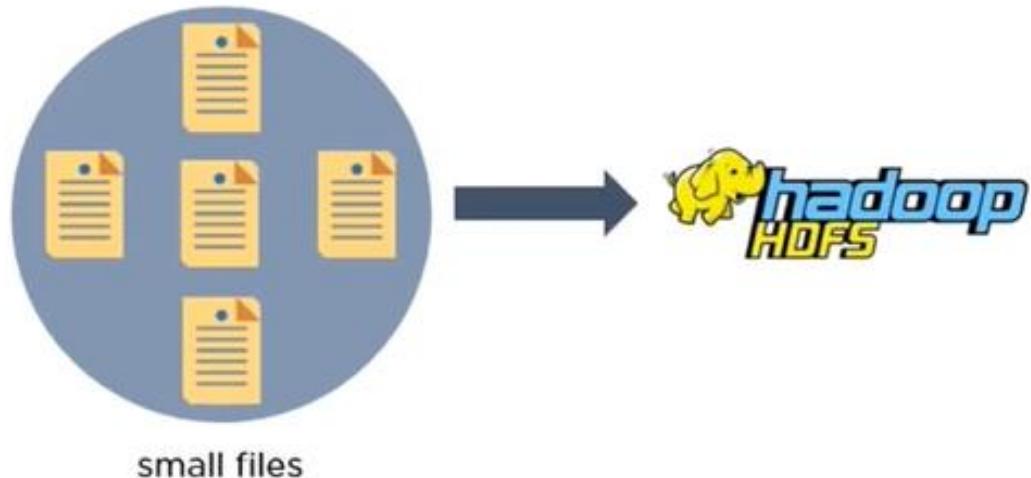
To check the status
of the blocks

`hdfs fsck / -files -blocks -locations > dfs-fsck.log`



To check the health
status of filesystem

What would happen if you store too many small files in a cluster on HDFS?



- Storing a lot of small files on HDFS generates **a lot of metadata files**
- Storing these metadata in the RAM is a challenge as each file, block or directory takes **150 bytes** just for metadata
- Thus, the cumulative size of all the metadata will be too big

How to copy data from local system on to HDFS?

Following command helps to copy data from local file system into HDFS:

```
hadoop fs -copyFromLocal [source] [destination]
```

Example: `hadoop fs -copyFromLocal /tmp/data.csv /user/test/data.csv`

When do you use `dfsadmin -refreshNodes` and `rmadmin -refreshNodes` command?

These commands are used to refresh the node information while commissioning or decommissioning of nodes is done

`dfsadmin -refreshNodes`



This is used to run HDFS client and it refreshes node configuration for the NameNode

`rmadmin -refreshNodes`



This is used to perform administrative tasks for ResourceManager

Is there anyway to change replication of files on HDFS after they are already written to HDFS?

Following are the ways to change the replication of files on HDFS:

We can change the `dfs.replication` value to a particular number in `$HADOOP_HOME/conf/hadoop-site.xml` file which will start replicating to the factor of that number for any new content that comes in

If you want to change the replication factor for a particular file or directory, then use:

`$HADOOP_HOME/bin/Hadoop dfs -setrep -w4 /path of the file`

Example: \$HADOOP_HOME/bin/Hadoop dfs -setrep -w4 /user/temp/test.csv

Who takes care of replication consistency in a Hadoop cluster and what do you mean by under/over replicated blocks?

NameNode

Namenode takes care of replication consistency in a Hadoop cluster and `fsck` command gives the information regarding over and under replicated block

Under-replicated blocks:

- These are blocks that do not meet their target replication for the file they belong to
- HDFS will automatically create new replicas of under-replicated blocks until they meet the target replication

Over-replicated blocks:

- These are blocks that exceed their target replication for the file they belong to
- Normally, over-replication is not a problem, and HDFS will automatically delete excess replicas

A professional setting with a woman in a blue blazer smiling at the camera while working on a laptop. A man in a suit stands behind her, looking thoughtful. The background is blurred.

MapReduce Questions

What is distributed cache in MapReduce?

It is a mechanism supported by the Hadoop MapReduce framework. The data coming from the disk can be cached and made available for all worker nodes where the map/reduce tasks are running for a given job

Once a file is cached for our job, Hadoop will make it available on each datanode where map/reduce tasks are running

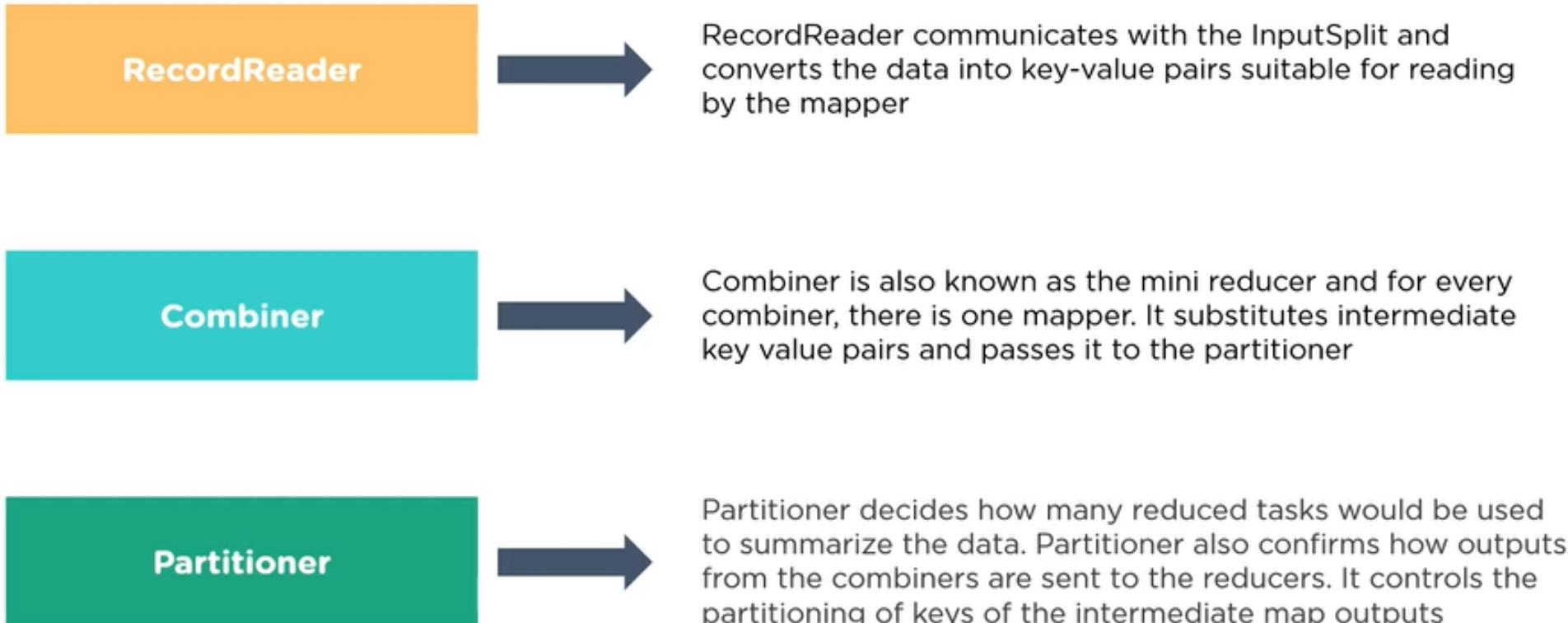
Copy the file to HDFS: \$ `hdfs dfs-put /user/Simplilearn/lib/jar_file.jar`

Setup the application's JobConf:

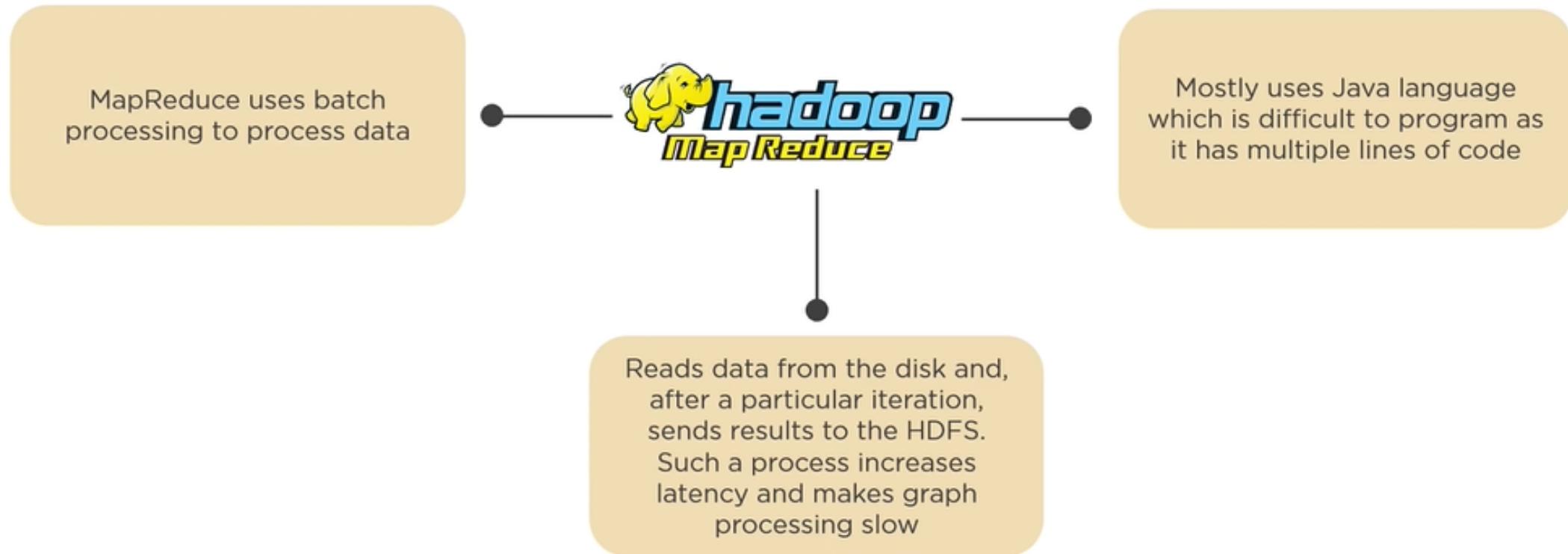
```
DistributedCache.addFileToClasspath(new  
path("/user/Simplilearn/lib/jar_file.jar"), conf)
```

Add it in Driver class

What role do RecordReader, Combiner and Partitioner play in a MapReduce operation?



Why is MapReduce slower in processing data in comparision to other processing frameworks?



For a MapReduce job, is it possible to change the number of mappers to be created?

By default, the number of mappers is always equal to the number of input splits. So, it cannot be changed

Example: If you have 1GB of file that is split into **8 blocks** (of 128MB each), so there will be only **8 mappers** running on the cluster

But, there are different ways in which you can either set a property or customize your code to change the number of mappers

Name some Hadoop specific data types that are used in a MapReduce program.

Following are some Hadoop specific data types used in a MapReduce program:

IntWritable

FloatWritable

LongWritable

DoubleWritable

BooleanWritable

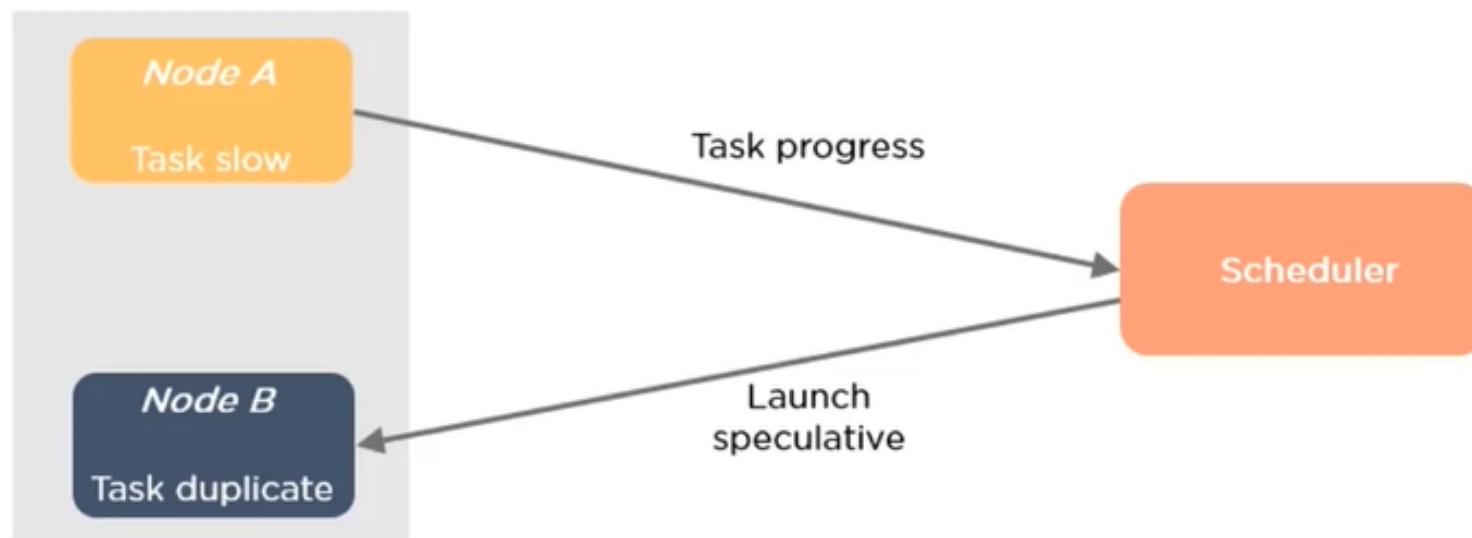
ArrayWritable

MapWritable

ObjectWritable

What is speculative execution in Hadoop?

- If a datanode is executing any task slowly, the master node can redundantly execute another instance of the same task on another node
- The task that finishes first will be accepted and the other task is killed



How is identity mapper different from chain mapper?

Identity Mapper

It is the default mapper which is chosen when no mapper is specified in MapReduce driver class

It implements identity function, which directly writes all its key-value pairs into output

It is defined in old MapReduce API (MR1) in *org.apache.Hadoop.mapred.lib.package*

Chain Mapper

This class is used to run multiple mappers in a single map task

The output of the first mapper becomes the input to the second mapper, second to third and so on

It is defined in *org.apache.Hadoop.mapreduce.lib.chain.ChainMapper* package

What are the major configuration parameters required in a MapReduce program?

- 
- 1 Input location of the job in HDFS
 - 2 Output location of the job in HDFS
 - 3 Input and output format
 - 4 Classes containing map and reduce functions
 - 5 .jar file for mapper, reducer and driver classes

What do you mean by map-side join and reduce-side join in MapReduce?

Map-side join

Here, the join is performed by the mapper

Each input data must be divided in same number of partitions

Input to each map is in the form of a structured partition and is in sorted order

Reduce-side join

Here, the join is performed by the reducer

Easier to implement than the map side join as the sorting and shuffling phase sends the values having identical keys to the same reducer

No need to have the dataset in a structured form (or partitioned)

OutputCommitter describes the commit of task output for a MapReduce job

Example:

```
org.apache.hadoop.mapreduce.OutputCommitter  
public abstract class OutputCommitter extends OutputCommitter
```

MapReduce framework relies on the OutputCommitter of the job to:

- Set's up the job initialization
- Cleanup the job after the job completion
- Setup the task temporary output
- Check whether a task needs a commit
- Commit of the task output
- Discard the task commit

Explain the process of spilling in MapReduce.

Spilling is a process of copying the data from memory buffer to disc when the content of the buffer reaches a certain threshold size

Spilling happens when there is not enough memory to fit all of the mapper output

By default, a background thread starts spilling the content from memory to disc after **80%** of the buffer size is filled

For a **100 MB** size buffer, the spilling will start after the content of the buffer reach a size of **80 MB**

29

How can you set the mappers and reducers for a MapReduce job?

The number of mappers and reducers can be set in the command line using:

```
-D mapred.map.tasks=5 -D mapred.reduce.tasks=2
```

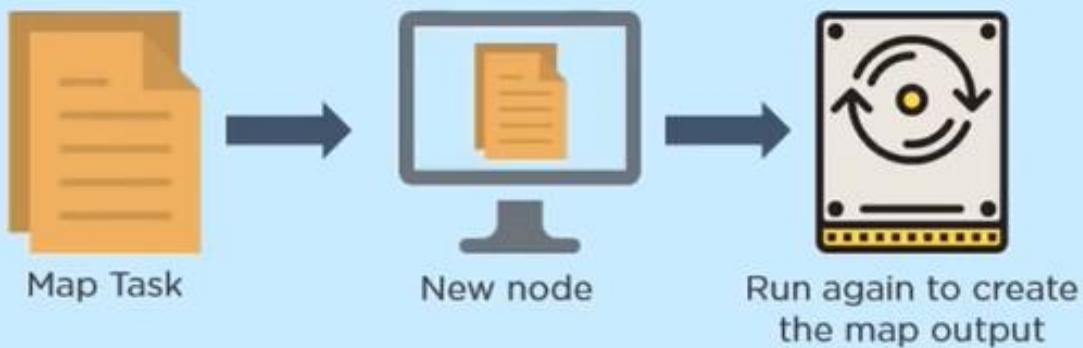
In the code, one can configure JobConf variables:

```
job.setNumMapTasks(5); // 5 mappers
```

```
job.setNumReduceTasks(2); // 2 reducers
```

What happens when a node running a map task fails before sending the output to the reducer?

If such a case happens, map tasks will be assigned to a new node and the whole task will be run again to re-create the map output



Can we write the output of MapReduce in different formats?

Yes, we can write the output of MapReduce in different formats.

Following are the examples:

TextOutputFormat

Default output format and it writes records as lines of text

SequenceFileOutputFormat

Useful to write sequence files when the output files need to be fed into another mapreduce jobs as input files

MapFileOutputFormat

Used to write output as map files

SequenceFileAsBinaryOutputFormat

Another variant of SequenceFileInputFormat. It also writes keys and values to sequence file in binary format

DBOutputFormat

Used for writing to relational databases and HBase. This format also sends the reduce output to a SQL table

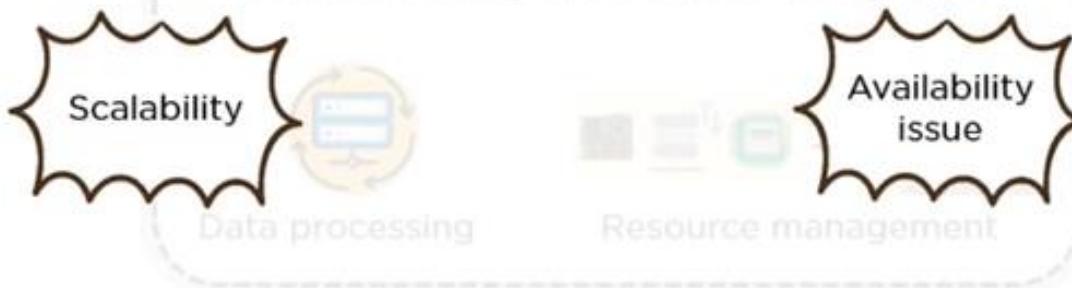
A professional setting where a woman in a blue blazer is smiling and looking towards the camera, holding a laptop. A man in a suit is visible to her right, listening attentively. The background is a modern office environment.

YARN

What benefits did YARN bring in Hadoop 2.0 and how did it solve the issues of MapReduce V1?

Following were some major issues:

data processing and resource management



Scalability



Data processing

Availability issue

Resource management

MapReduce consisted of Job Tracker and Task Tracker

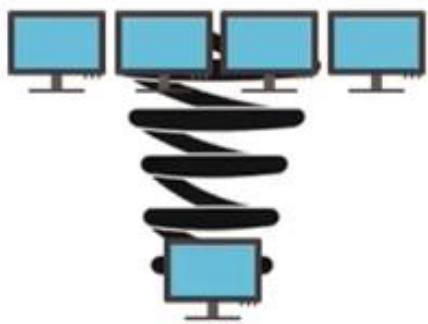
Resource utilization

Managing jobs using a single job tracker for all types of computational resources was inefficient in MR 1

Can't run non-MapReduce jobs

What benefits did YARN bring in Hadoop 2.0 and how did it solve the issues of MapReduce V1?

Scalability



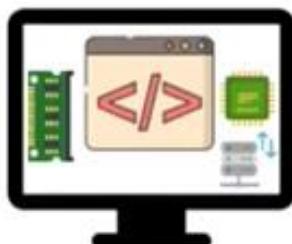
Can have a cluster size of more than 10,000 nodes and can run more than 1,00,000 concurrent tasks

Compatibility



Applications developed for Hadoop 1 runs on YARN without any disruption or availability issues

Resource utilization



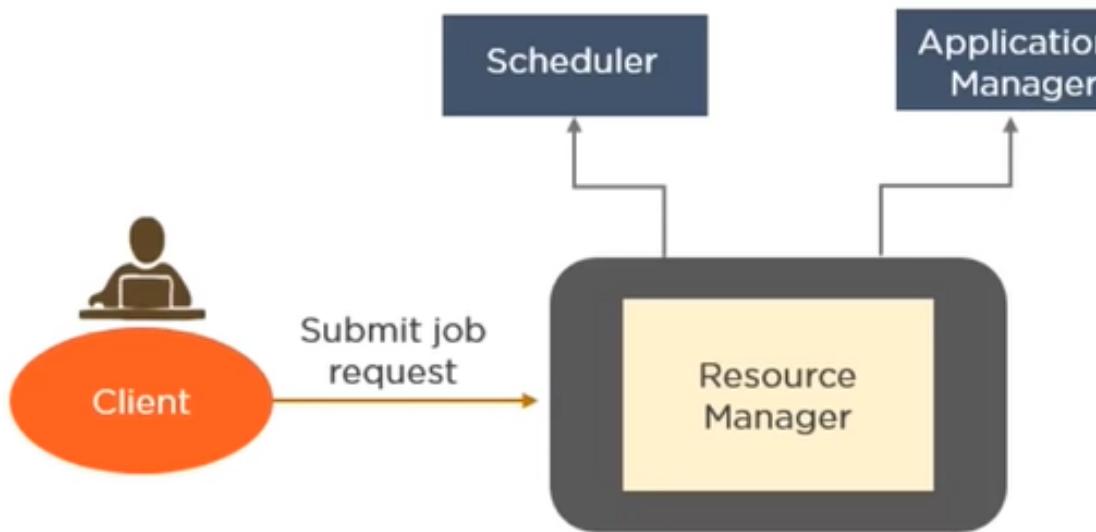
Allows dynamic allocation of cluster resources to improve resource utilization

Multitenancy



Can use open-source and proprietary data access engines and perform real-time analysis and running ad-hoc query

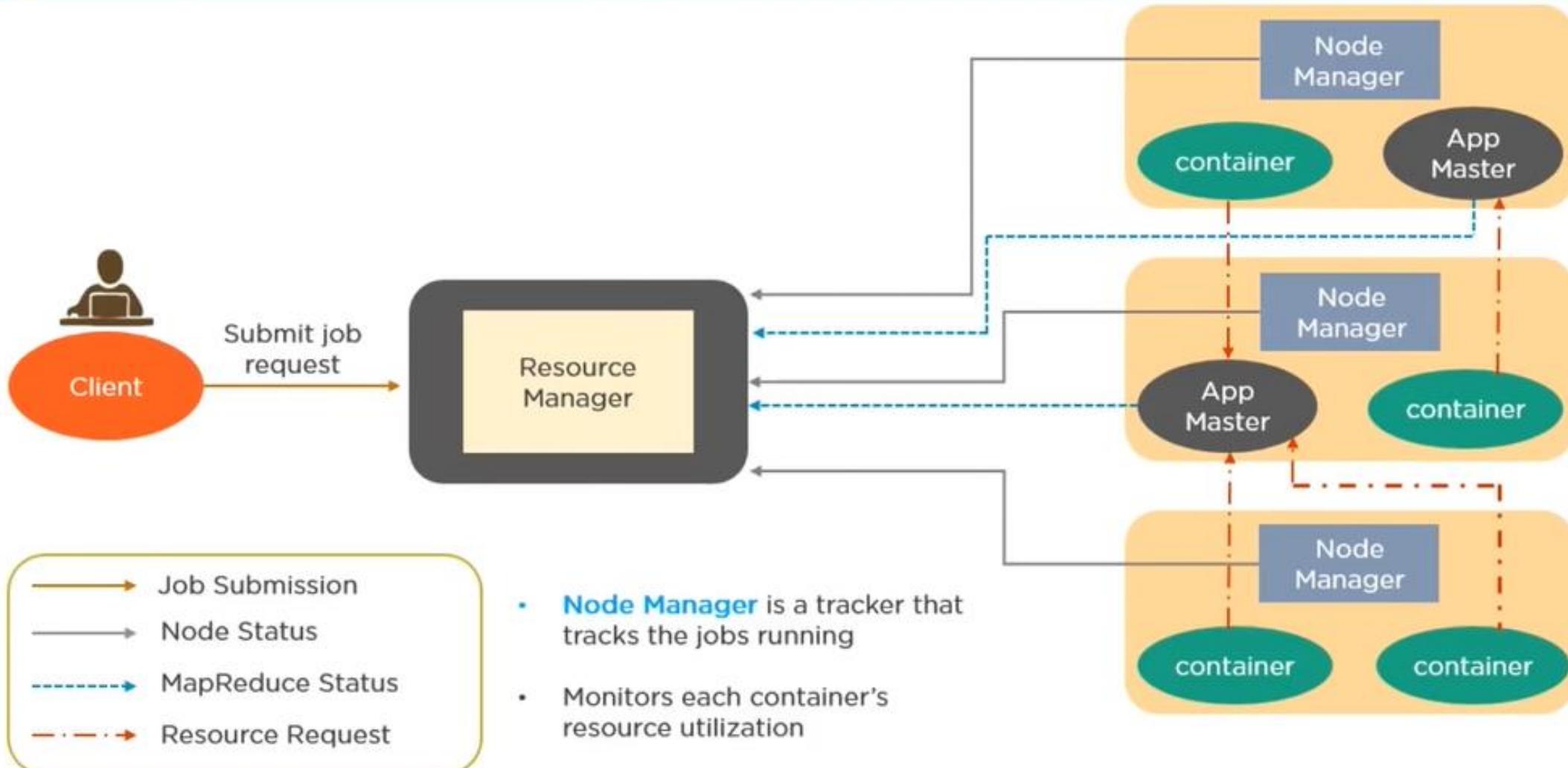
Explain how YARN allocates resources to an application with the help of its architecture.



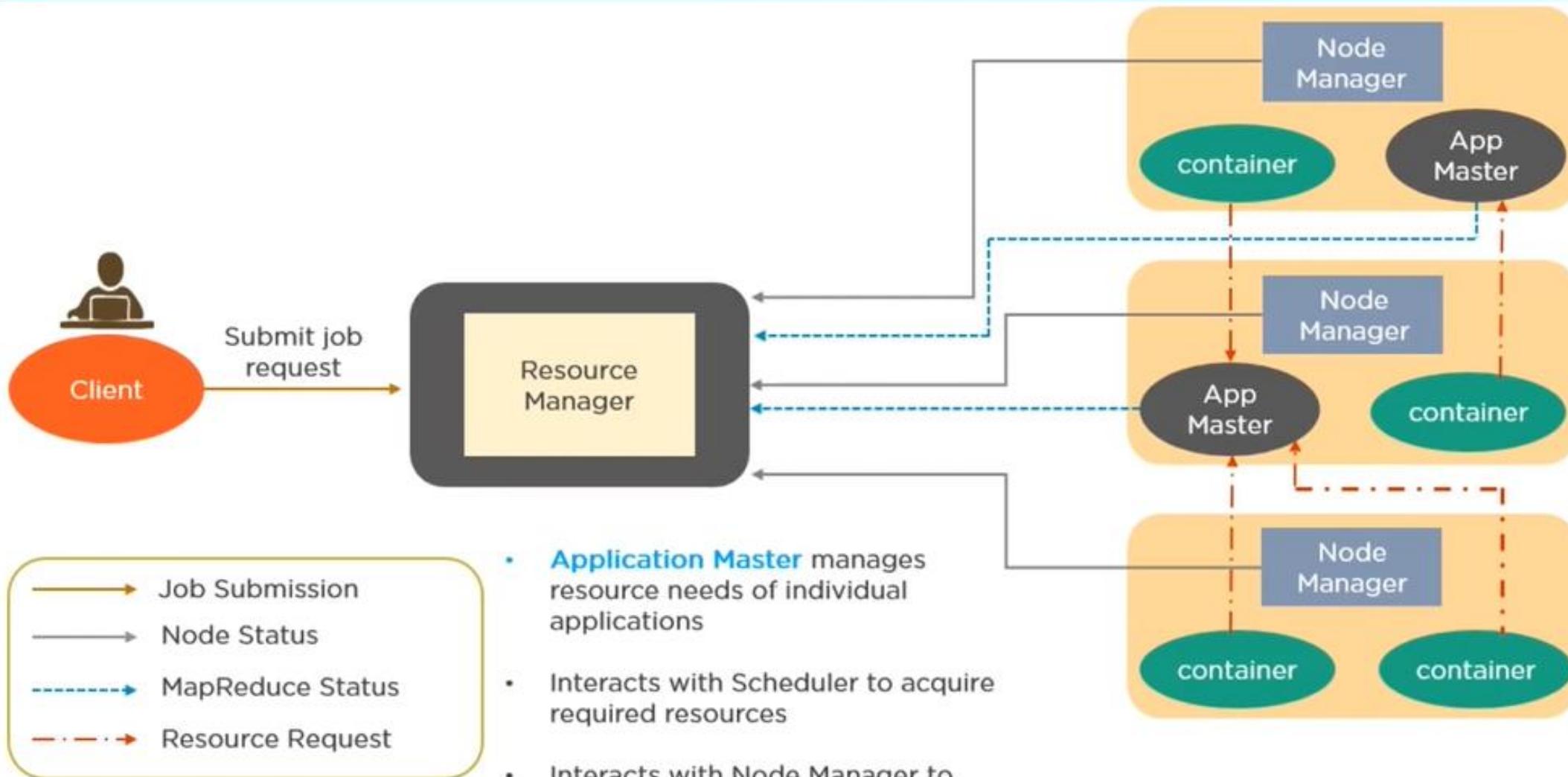
→ Job Submission

- **Scheduler** allocates resources to various running applications
- Schedules resources based on the requirements of the applications
- Does not monitor or track the status of the applications
- **Applications Manager** accepts job submissions
- Monitors and restarts application masters in case of failure

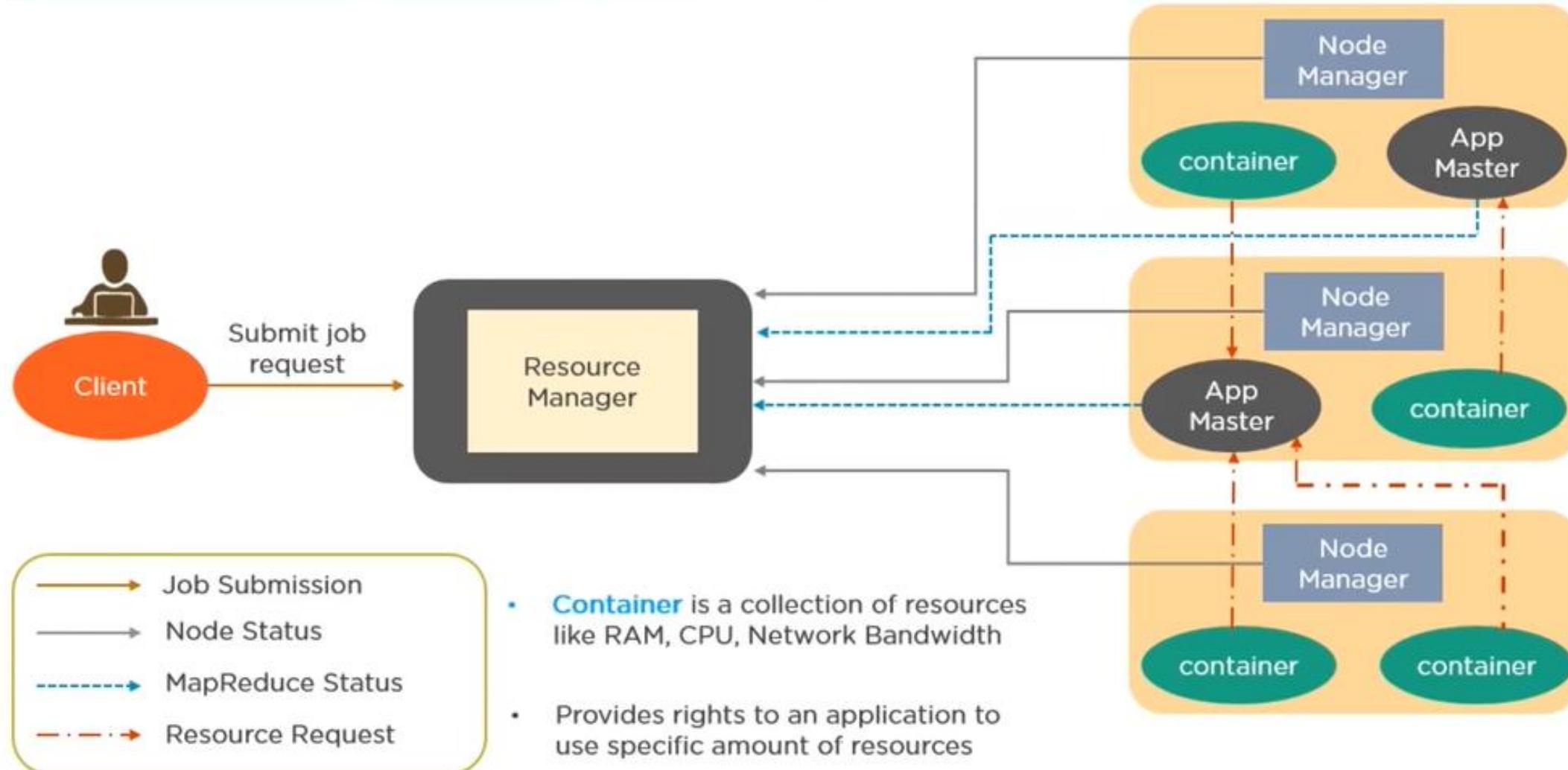
Explain how YARN allocates resources to an application with the help of its architecture.



Explain how YARN allocates resources to an application with the help of its architecture.



Explain how YARN allocates resources to an application with the help of its architecture.



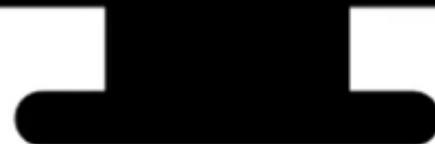
Which of the following has occupied the place of JobTracker of MapReduce V1?

(a) NodeManager

(b) ApplicationMaster

(c) ResourceManager

(d) Scheduler



Write the YARN commands to check the status of an application and kill an application.

To check the status of an application:

```
yarn application -status ApplicationID
```

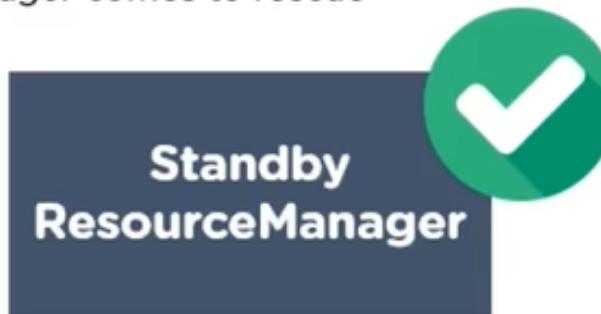
To kill or terminate an application:

```
yarn application -kill ApplicationID
```

Can we have more than 1 ResourceManager in a YARN based cluster?

Yes, there can be more than 1 ResourceManager in case of a High Availability cluster

At a particular time, there can only be [one active ResourceManager](#). In case the active ResourceManager fails, then the standby ResourceManager comes to rescue



What are the different schedulers available in YARN?.

The different schedulers available in YARN are:

1. **FIFO scheduler**: The FIFO Scheduler places applications in a queue and runs them in the order of submission (first in, first out)
2. **Capacity scheduler**: A separate dedicated queue allows the small job to start as soon as it is submitted. The large job finishes later than when using the FIFO Scheduler
3. **Fair scheduler**: There is no need to reserve a set amount of capacity, since it will dynamically balance resources between all the running jobs

What happens if a ResourceManager fails while executing an application in a high availability cluster?

1

In a high availability cluster, if one ResourceManager fails, another ResourceManager becomes active

2

If a ResourceManager fails in case of a high availability cluster, the newly active ResourceManager instructs the ApplicationsMaster to abort

3

Resource Manager recovers its running state by taking advantage of the container statuses sent from all Node Managers

39

In a cluster of 10 datanodes, each having 16 GB RAM and 10 cores, what would be the total processing capacity of the cluster?



- Every node in a Hadoop cluster have multiple processes running and these processes need RAM
- The machine which has its own processes, would also need some ram usage
- So, if you have 10 datanodes, you need to allocate at least 20-30% towards the overheads, cloudera based services, etc.
- You could have 11-12 GB and 6-7 cores available on every machine for processing. Multiply that by 10 and that's the processing capacity

What happens if requested memory or CPU cores goes beyond the size of container allocation?



If an application needs more memory and CPU cores, it cannot fit into a container allocation. So the application fails

A professional setting with a woman in a blue blouse smiling and holding a white folder. Other people are visible in the background.

Hive Questions

What are the different components of a Hive architecture?



User Interface

It calls the execute interface to the driver. This creates a session to the query. Then, it sends the query to the compiler to generate an execution plan for it

Metastore

It stores the metadata information and sends it to the compiler for execution of a query

Compiler

It generates the execution plan. It has a DAG of stages where each stage is either a metadata operation, a map or reduce job or an operation on HDFS

Execution Engine

Execution engine acts as bridge between Hive and Hadoop to process query. Execution engine communicates bidirectionally with Metastore to perform operations like create, drop tables

What is the difference between external table and managed table in Hive?

External Table

- External tables in Hive refer to the data that is at an existing location outside the warehouse directory
- Hive deletes the metadata information of a table and does not change the table data present in HDFS



Managed Table

- Also known as internal table, it manages the data and moves it into its warehouse directory by default
- If one drops a managed table, the metadata information along with the table data is deleted from the Hive warehouse directory

What is a partition in Hive and why is partitioning required in Hive?

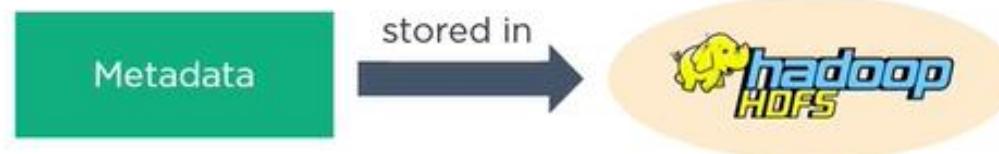
Partition is the process to group similar type of data together on the basis of column or partition key. Each table can have one or more partition keys to identify a particular partition

Partitioning provides granularity in a Hive table. It reduces the query latency by scanning only relevant partitioned data instead of the whole data set

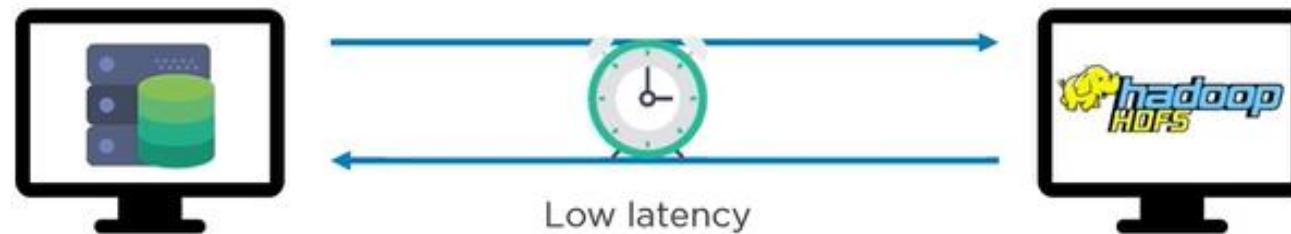


We can partition the transaction data for a bank based on month – Jan, Feb, etc. Any operation regarding a particular month, say Feb, will have to scan only the Feb partition instead of the whole table data.

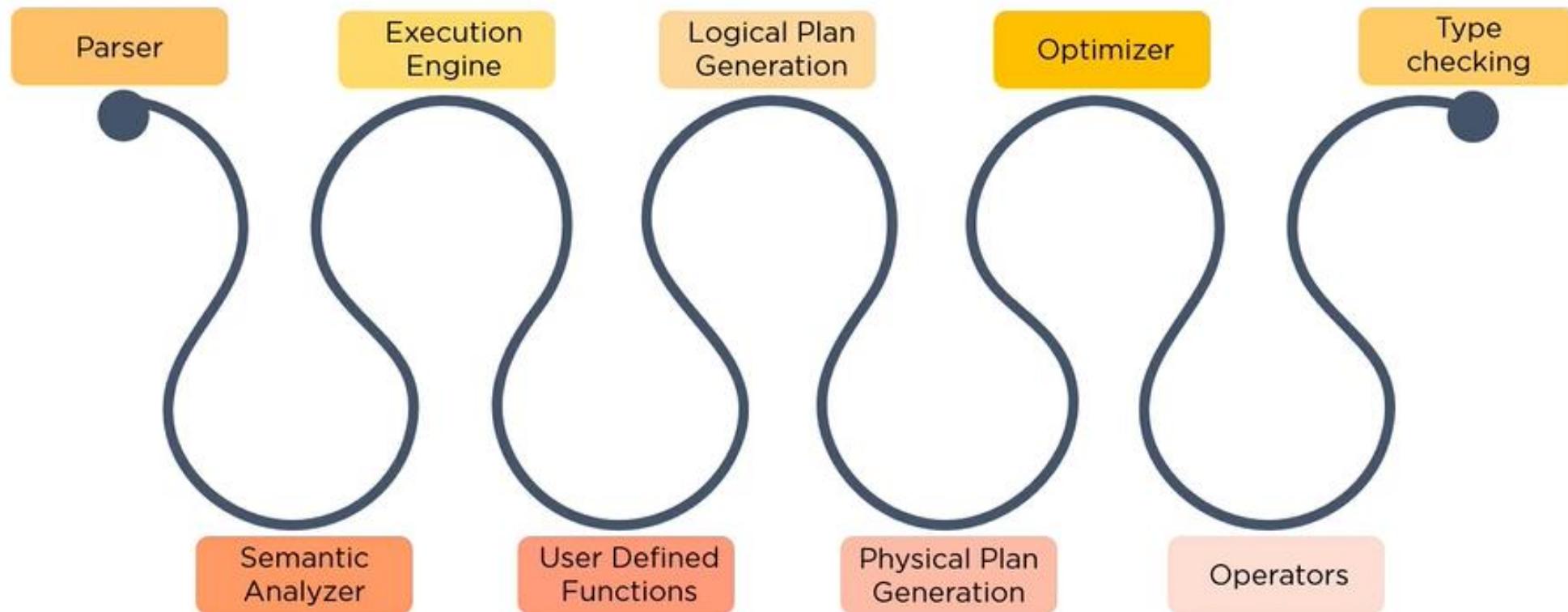
Why does Hive not store metadata information in HDFS?



HDFS read/write operations are time consuming. So, Hive stores metadata information in the metastore using RDBMS instead of HDFS. This allows to achieve low latency and is faster



What are the components used in Hive query processor?



Suppose, there are a lot of small CSV files present in /user/input directory in HDFS and you want to create a single Hive table from these files. The data in these files have fields: {registration_no, name, email, address}. What will be your approach to solve this, where will you create a single Hive table for lots of small files without degrading the performance of the system?

Using **SequenceFile format** and grouping these small files together to form a single sequence file can solve this problem. Below are the steps:

1

Create a temporary table:

```
CREATE TABLE test (registration_no INT, name STRING, email STRING, address STRING)  
ROW FORMAT FIELDS DELIMITED TERMINATED BY ',' STORED AS TEXTFILE;
```

2

Load the data into test table:

```
LOAD DATA INPATH '/user/input' INTO TABLE test;
```

3

Create a table that will store data in SequenceFile format:

```
CREATE TABLE test_seqfile (registration_no INT, name STRING, email STRING, address STRING)  
ROW FORMAT FIELDS DELIMITED TERMINATED BY ',' STORED AS SEQUENCEFILE;
```

4

Move the data from the test table into test_seqfile table:

```
INSERT OVERWRITE TABLE test_seqfile SELECT * FROM test;
```

47

Write a query to insert a new column(new_col INT) into a hive table (h_table) at a position before an existing column (x_col).

Following query will insert a new column into a hive table:

```
ALTER TABLE h_table  
CHANGE COLUMN new_col INT  
BEFORE x_col
```

What are the key differences between Hive and Pig?

HIVE

1

Uses a declarative language called HiveQL similar to SQL for reporting

2

Operates on the server side of the cluster and allows structured data

3

Does not support Avro file format by default. This can be done using:
"Org.Apache.Hadoop.Hive.serde2.Avro"

3

It was developed by Facebook and it supports partition

PIG

1

Uses a high level procedural language called Pig Latin for programming

2

Operates on the client side of the cluster and allows both structured and unstructured data

3

Supports Avro file format by default

3

It was developed by Yahoo and it does not support partition



Pig Questions

How is Apache Pig different from MapReduce?

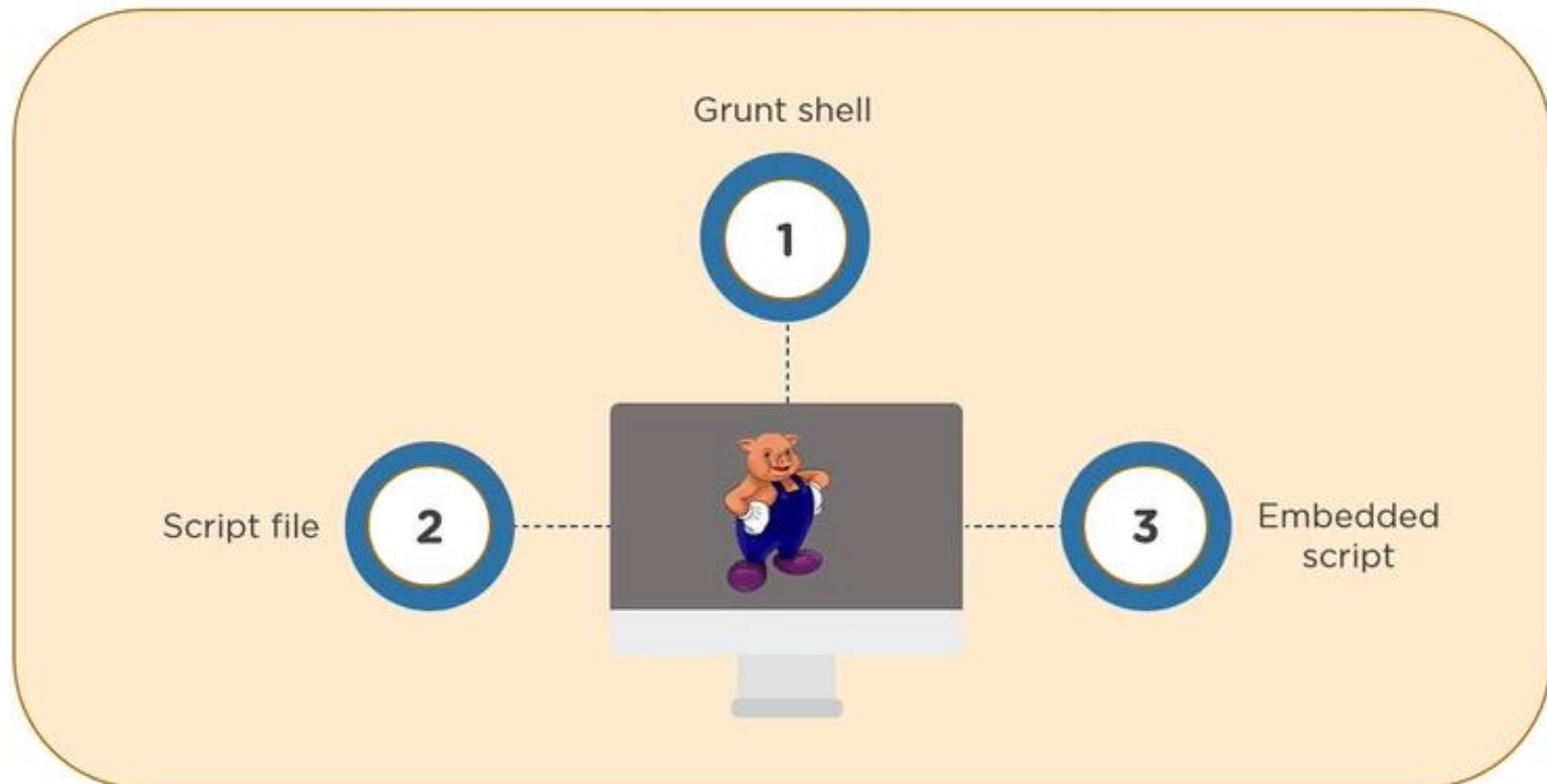
PiG

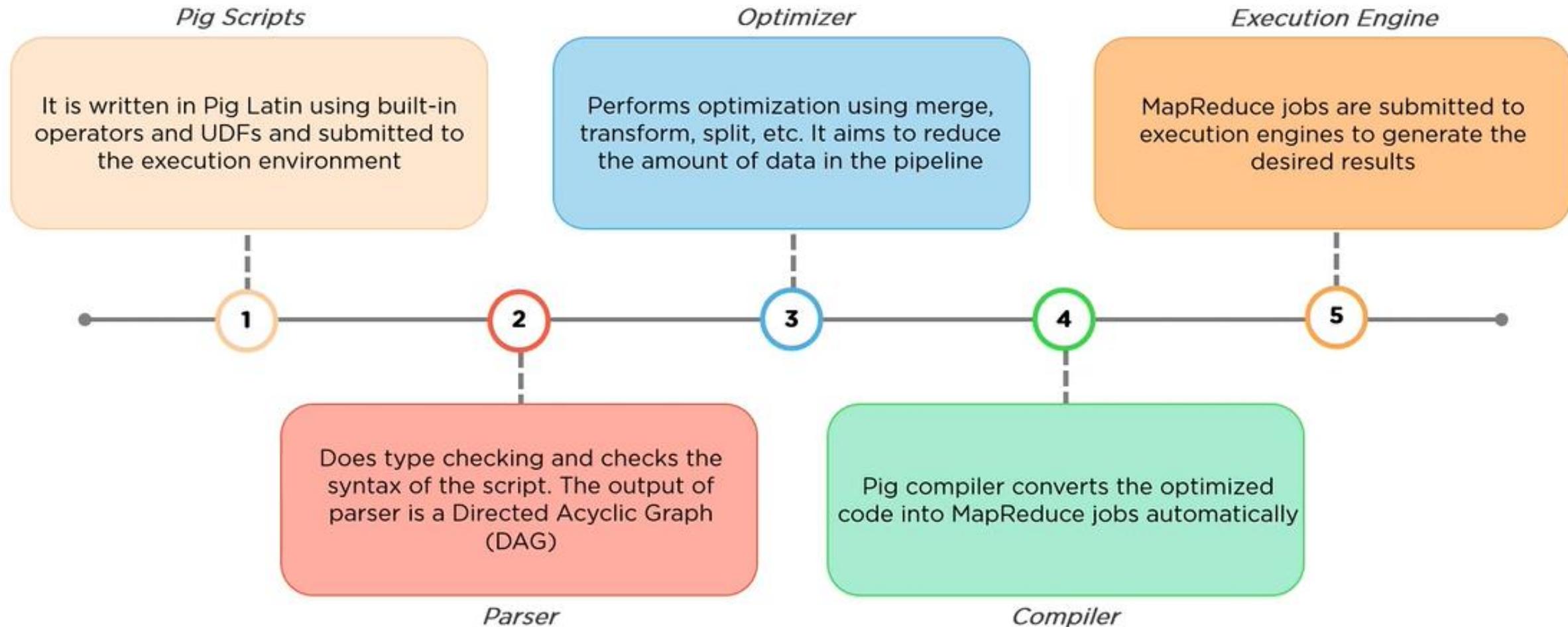
- Has less lines of code compared to MapReduce
- High level language which can easily perform join operation
- On execution, every Pig operator is converted internally into a MapReduce job
- Works with all the versions of Hadoop

MapReduce

- Has more lines of code
- Low level language which cannot perform join operation easily
- MapReduce jobs take more time to compile
- A MapReduce program written in one Hadoop version may not work with other versions

What are the different ways of executing Pig script?





Explain the different complex data types in Pig.

Tuple

A tuple is an ordered set of fields which can contain different datatypes for each field. It is represented by braces ().

Example: (1,3)

Bag

A bag is a set of tuples represented by curly braces {}.

Example: {(1,4), (3,5), (4,6)}



A Map is a set of key-value pairs used to represent data elements. It is represented in square brackets [].

Example: [key#value, key1#value1,...]

Map

Dump

Dump operator runs the Pig Latin scripts and displays the results on the screen

- Load the data using “`load`” operator into Pig
- Display the results using “`dump`” operator

Describe

Describe operator is used to view the schema of a relation

- Load the data using “`load`” operator into Pig
- View the schema of a relation using “`describe`” operator

Explain

Explain operator displays the physical, logical and MapReduce execution plans

- Load the data using “`load`” operator into Pig
- Display the logical, physical and MapReduce execution plans using “`explain`” operator

Illustrate

Illustrate operator gives the step-by-step execution of a sequence of statements

- Load the data using “`load`” operator into Pig
- Show the step-by-step execution of a sequence of statements using “`illustrate`” operator

State the usage of group, order by and distinct keywords in Pig scripts.

Group statement collects various records with the same key and groups the data in one or more relations.

Example: `Group_data = GROUP Relation_name BY AGE`

Order statement is used to display the contents of a relation in a sorted order based on one or more fields.

Example: `Relation_2 = ORDER Relation_name1 BY (ASC|DSC)`

Distinct statement removes duplicate records and is implemented only on entire records and not on individual records.

Example: `Relation_2 = DISTINCT Relation_name1`

What are the relational operators in Pig?

COGROUP: Joins two or more tables and then performs GROUP operation on the joined table result

CROSS: It is used to compute the cross product (cartesian product) of two or more relations

FOREACH: It will iterate through the tuples of a relation, generating a data transformation

JOIN: Used to join two or more tables in a relation

LIMIT: It will limit the number of output tuples

SPLIT: This will split the relation into two or more relations

UNION: It will merge the contents of two relations

ORDER: Used to sort a relation based on one or more fields

What is the use of having filters in Apache Pig?

Filter:

FILTER operator is used to select the required tuples from a relation based on a condition. It also allows you to remove unwanted records from the data file

"phone_sales" data

```
year, product, quantity
-----
2000, phone, 1000
2001, phone, 1500
2002, phone, 1700
2003, phone, 1200,
2004, phone, 800
2005, phone, 900
```

Filter the products whose quantity is greater than 1000

Example:

```
A = LOAD '/user/Hadoop/phone_sales' USING PigStorage(',') AS (year:int,  
product:chararray, quantity:int);
```

```
B = FILTER A BY quantity > 1000
```

Suppose there's a file called "test.txt" having 150 records in HDFS.
Write the Pig command to retrieve the first 10 records of the file.

We need to use the **limit operator** to retrieve the first 10 records from a file

Load the data in Pig:

```
test_data = LOAD "/user/test.txt" USING PigStorage(',') as (field1, field2,...);
```

Limit the data to first 10 records:

```
Limit_test_data = LIMIT test_data 10;
```

HBase Questions



What are the key components of HBase?

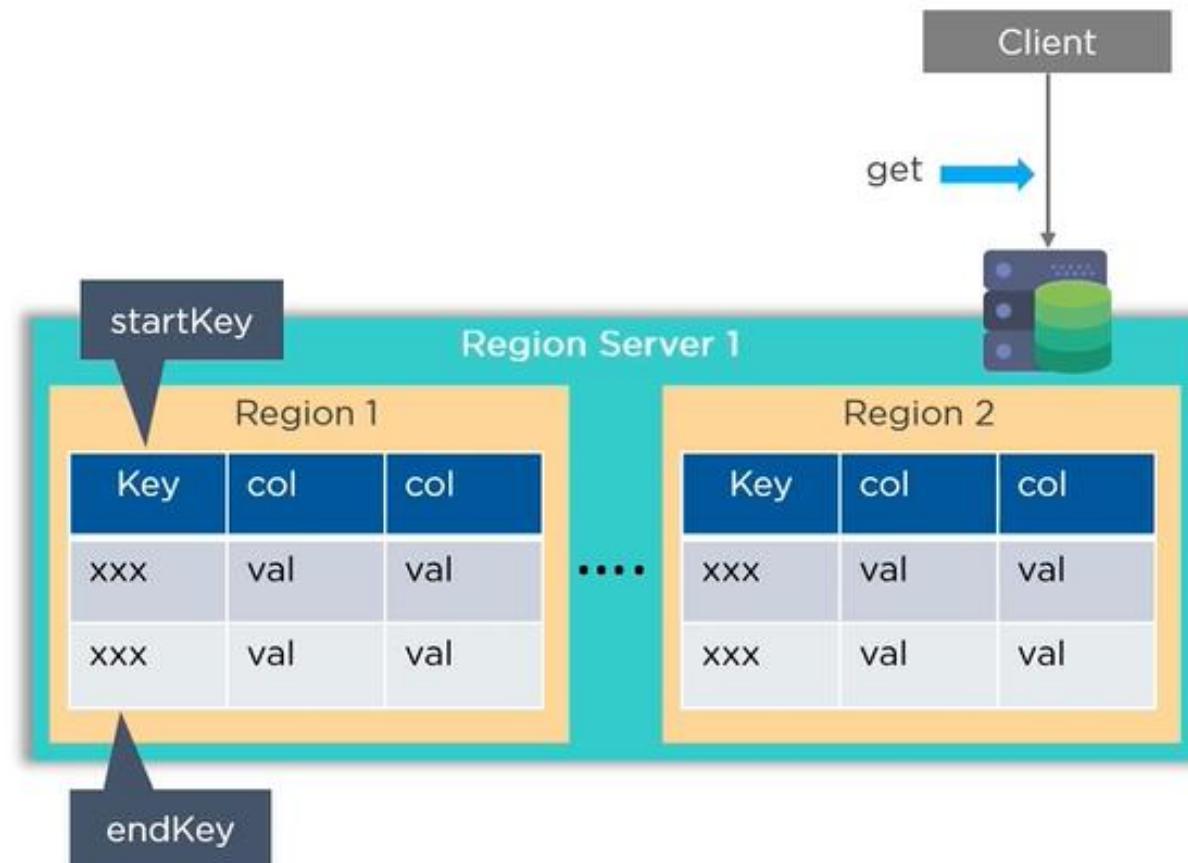
1

Region Server

Region server contains HBase tables that are divided horizontally into “Regions” based on their **Key** values

It runs on every node and decides the size of the region

Each region server is a worker node which handles **read**, **write**, **update** and **delete** request from clients



What are the key components of HBase?

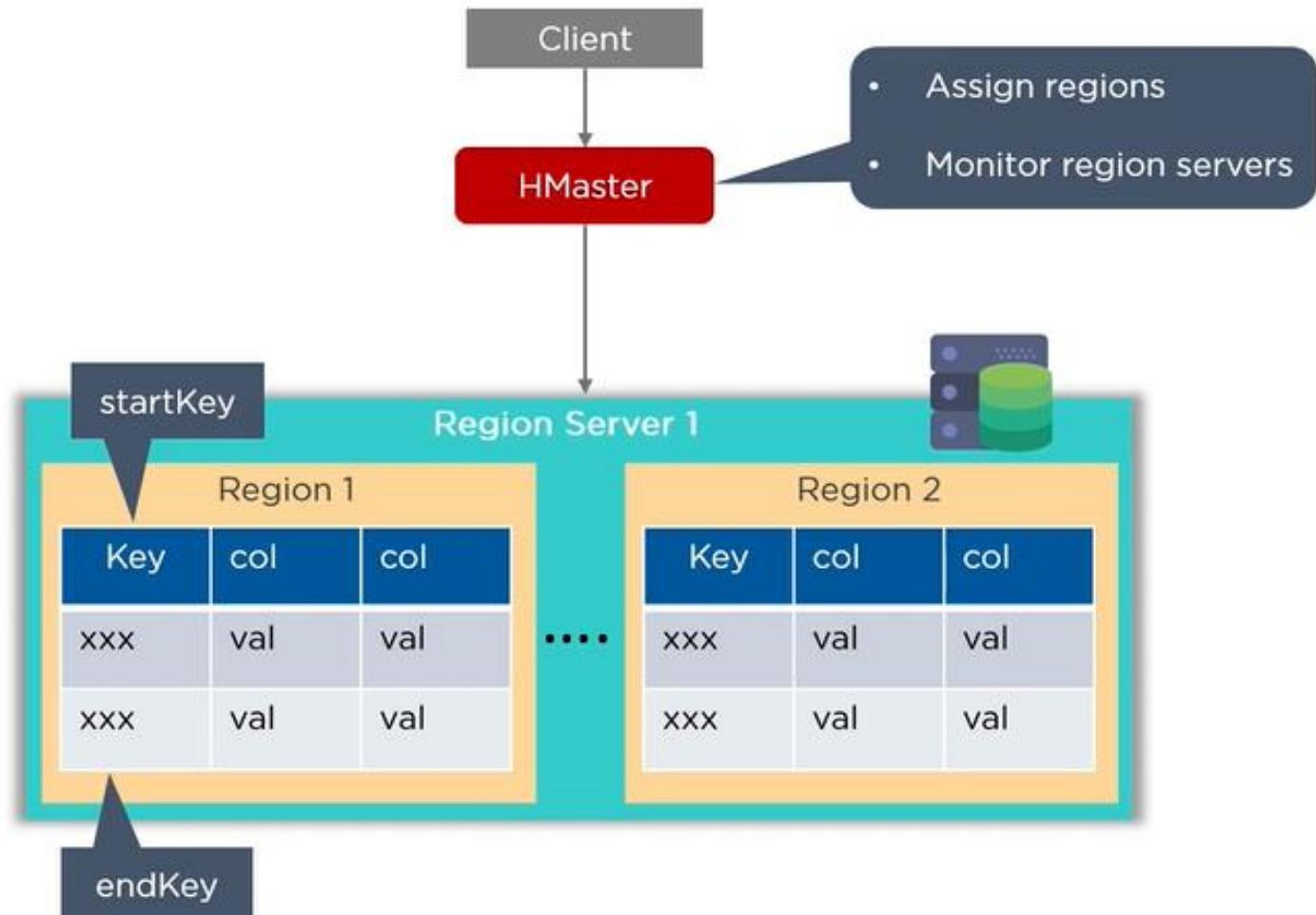
2

HMaster

Assigns regions to RegionServers for load balancing

Monitors and manages the Hadoop cluster

Whenever a client wants to change the schema and any of the metadata operations, HMaster is used



What are the key components of HBase?

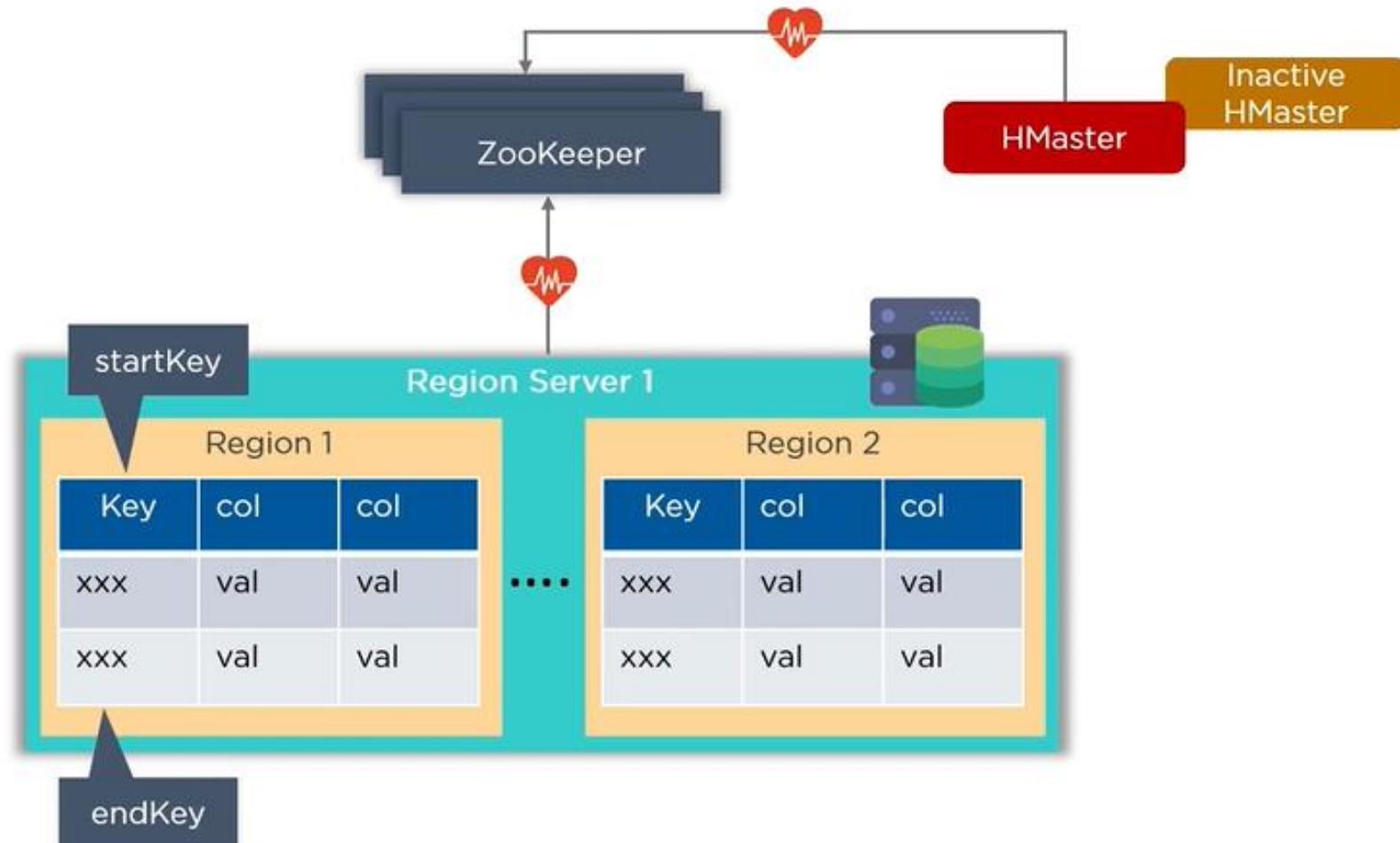
3

ZooKeeper

Provides distributed coordination service to maintain server state in the cluster

Maintains which servers are alive and available, and provides server failure notification

Region servers send their statuses to ZooKeeper indicating if they are ready for read and write operation



Explain what are row key and column families in HBase?

Row key acts as a primary key for any HBase table. It also allows logical grouping of cells and makes sure that all cells with the same row key are co-located on the same server

Column families consist of a group of columns which is defined during table creation and each column family has certain column qualifiers separated by a delimiter

		Column families				
Row Key		Personal data		Professional data		
Rowid	empid	name	city	age	designation	salary
1	1	Angela	Chicago	31	Big Data Architect	\$70,000
2	2	Dwayne	Boston	35	Web Developer	\$65,000
3	3	David	Seattle	29	Data Analyst	\$55,000

Diagram illustrating the structure of an HBase table:

- Row Key:** Indicated by red arrows pointing to the first two columns of the table.
- Column families:** Indicated by the header "Personal data" and "Professional data".
- Column Qualifiers:** Indicated by red arrows pointing to the sub-fields within the columns, such as "name", "city", "age", "designation", and "salary".
- Cells:** Indicated by red arrows pointing to the individual data values stored in the table.

The table data is as follows:

Rowid	empid	name	city	age	designation	salary
1	1	Angela	Chicago	31	Big Data Architect	\$70,000
2	2	Dwayne	Boston	35	Web Developer	\$65,000
3	3	David	Seattle	29	Data Analyst	\$55,000

Employee_table

Rowid	Personal data			Professional data	
	empid	name	city	age	designation
1	Angela	Chicago	31	Big Data Architect	\$70,000
2	Dwayne	Boston	35	Web Developer	\$65,000
3	David	Seattle	29	Data Analyst	\$55,000

HBase table is disabled to allow it to be modified or change its settings. When a table is disabled it cannot be accessed through the scan command.

To disable the employee table:

```
disable 'employee_table'
```

To check if the table is disabled:

```
is_disabled 'employee_table'
```

Write the code to open a connection in HBase.

Below code is used to open a connection in HBase:

```
Configuration myConf = HBaseConfiguration.create();  
HTableInterface usersTable = new HTable(myConf, "users");
```

What does replication mean in terms of HBase?

Replication feature in HBase provides a mechanism to copy data between clusters. This feature can be used as a disaster recovery solution that provides high availability for HBase

The following commands alter the `hbase1` table and set the `replication_scope` to 1. A `replication_scope` of 0 indicates that the table is not replicated

```
disable 'hbase1'  
alter 'hbase1', {NAME => 'family_name', REPLICATION_SCOPE => '1'}  
enable 'hbase1'
```

Can we do an import/export in HBase table?

Yes, it is possible to import and export tables from one HBase cluster to another

HBase export utility: `hbase org.apache.hadoop.hbase.mapreduce.Export "table name" "target export location"`

Example: `hbase org.apache.hadoop.hbase.mapreduce.Export "employee_table" "/export/employee_table"`

HBase import utility: `create 'emp_table_import', {NAME => 'myfam', VERSIONS => 10}`

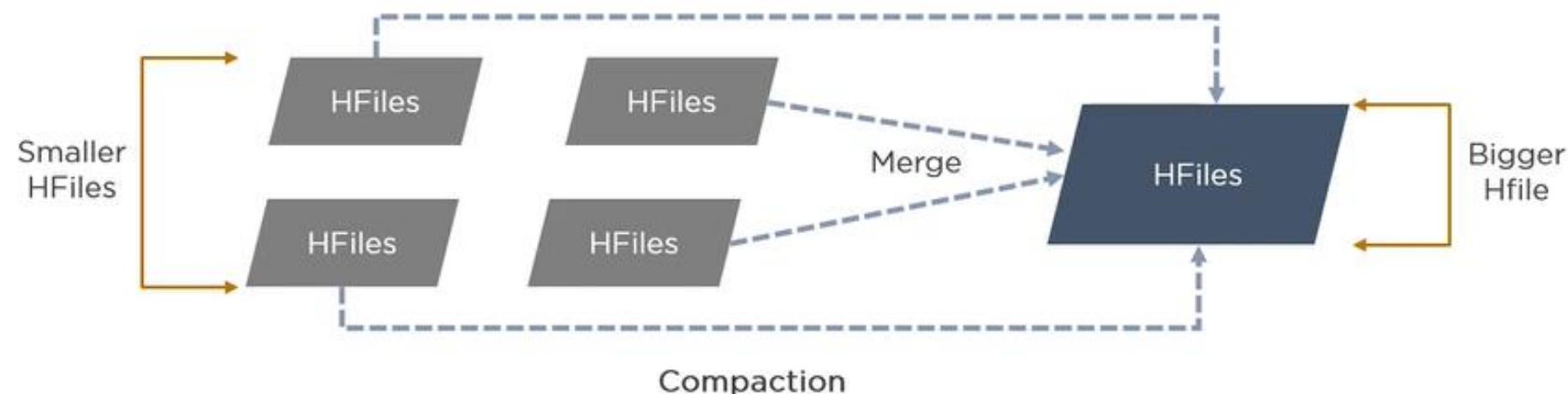
`hbase org.apache.hadoop.hbase.mapreduce.Import "table name" "target import location"`

Example: `create 'emp_table_import', {NAME => 'myfam', VERSIONS => 10}`

`hbase org.apache.hadoop.hbase.mapreduce.Import "emp_table_import" "/export/employee_table"`

What do you mean by compaction in HBase?

Compaction is the process of *merging HBase* files into a single file. This is done to *reduce the amount of memory* required to store the files and the *number of disk seeks* needed.
Once the files are merged, the original file is deleted



How does Bloom filter work?

HBase Bloom filter is a mechanism to test whether a HFile contains a specific row or row-col cell



Bloom filter is named after its creator Burton Howard Bloom. It is a data structure which predicts whether a given element is a member of a set of data



Bloom filters provide an in-memory index structure that reduces disk reads and determines the probability of finding a row in a particular file

Does HBase have any concept of namespace?

namespace

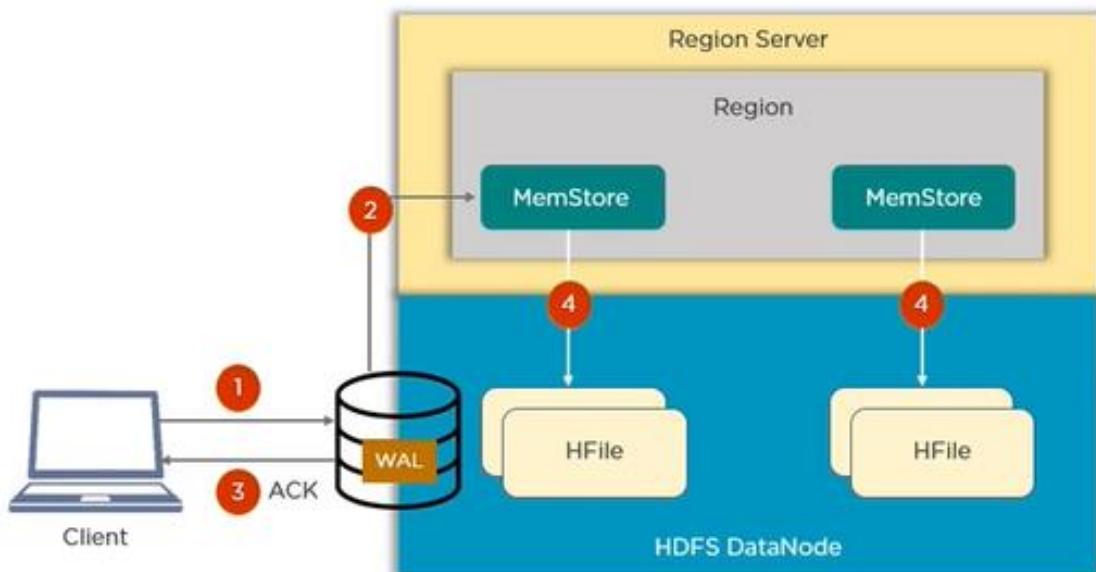
A namespace is a logical grouping of tables, analogous to a database in RDBMS. You can create the HBase namespace to the schema of the RDBMS database

To create a namespace - `create_namespace 'namespacename'`

To list all the tables that are members of the namespace - `list_namespace_tables 'default'`

To list all the namespaces - `list_namespace`

How does Write Ahead Log (WAL) help when a RegionServer crashes?



- If a RegionServer hosting a MemStore crashes, the data that in memory but not yet persisted, are lost
- HBase recovers against that by writing to the WAL before the write completes
- HBase cluster keeps a WAL to record changes as they happen
- If HBase goes down, the data that were not yet flushed from the MemStore to the HFile can be recovered by replaying the WAL

Write the HBase command to list the contents and update the column families of a table.

Below code is used to list the contents of a HBase table:

```
scan 'table_name'
```

Example: scan 'employee_table'

Update column families in the table:

```
alter 'table_name', 'column_family_name'
```

Example: alter 'employee_table', 'emp_address'

What are catalog tables in HBase?

Catalog has 2 tables: `hbase:meta` and `-ROOT-`

- The catalog table `hbase:meta` exists as an HBase table and is filtered out of the HBase shell's list command.
- It keeps a list of all the regions in the system and the location of `hbase:meta` is stored in ZooKeeper
- The `-ROOT-` table keeps track of the location of the .META table

What is Hotspotting in HBase and how to avoid Hotspotting?

- In Hbase, all read and write requests should be uniformly distributed across all of the regions in the RegionServers
- Hotspotting occurs when a given region serviced by a single RegionServer receives most or all of the read or write requests

Hotspotting can be avoided by designing the row key in such a way that data being written should go to multiple regions across the cluster.

Below are the techniques to do so:

Salting

Hashing

Reversing
the key