**Questions asked in interview questions-**

* **what is spark?**

Open source

Distributed

Cluster-computing

Framework

* **What is AWS?**

Amazon web service –

It is cloud

It is set of machine

Which can be access through just internet

It has applications and databases installed on it.

We don’t need to maintain it.

No physical server.

Security

Pay per use

Highly scalable

We don’t need to build cluster.

Easy to use

Higher Speed

Volume is not issue

It is useful in banking, healthcare and retail services

* **What is cloud?**

Cloud is set of remote machines which can be access through internet.

**Eg. AWS,GCP**

* **What is EC2 instance? What properties we configure in EC2 insatnce?**

Elastic Compute

It is remote machine which has software and databases.

No need to maintain physical machine,

22 regions

You can access instance using key value pair

First u need to choose region

Choose ami

Choose type of instance

Choose no of instances to create

Choose VPC

Shutdown behavior :- stop or terminate

Tenancy - shared or dedicated

Protect again accidental termination

Choose if EBS needed

Choose security ie permission granted to access each EMR components.

Eg read,write,update etc

* **What are typed of instances?**

There are 6 types instances

**Compute based –**

c series ,t2 and t3

where more CPU power is needed, eg. Batch and real time processing

**Storage based –**

D 48GB ,

H 16 GB,

I series

eg. Big data processing

**General –**

m series –

m series compute is equally distributed eg games, web servers

**Memory based –** r series –

for processing in memory data eg spark

**Instances are again divided into series**

Large , xlarge,2xlarge,4xlarge ……

* **What is cluster?**

Set of local or vitual machines.

* **What is EMR?**

Elastic map-reduce is cluster which process big data.

It is collection of EC2 machine and S3.

You can create emr

Give machine name

Choose key pair required or not

Choose which softwares are needed

Choose bucket name

And add step to execute your program

* **What is AMI?**

Amazon machine image

It provides operating system and basic softwares to EC2.

* **What is use of cloud services?**
* **How much amount of data you can upload in S3 bucket ?**

**Max 5 GB else use multiupload**

* **Tell me about Cisco company?**

Cisco developed in 1985 ,

Its head quarter is in silicon valley

it sells softwares, hardwares required for the networking.

Hence useful for global networking

It sells products which are useful for communication, data transfers

* **Tell me about Persistant company?**

Persistant developed in 1995

Its headquater is in india

It builds a software

They work with IBM, Microsoft, Agilent and Cisco clients

Works on telecom, healthcare and financial services

* **What is GCP ?**

GCP cloud platform it is networking, storing big data ,machine learning services provided by google.

* **What are components of the GCP?**

Compute services

Storage services

Networking

Big data services

Security and identity management

Cloud AI

IOT

* **Advantages of the GCP?**

Security

Better pricing

Virtual global private network even runs under ocean

Very fast

* **Different methods of authentication in GCP?**
* **How you deploy application in spark?**

So first we develop the application in scala ide,

then we compile the program

then we create jar file

We deploy jar files using spark-submit command

**Spark submit**

– **class** classname // main class name to run

– **master** yarn

– **deploy mode** client or cluster

-- **driver-mode** client

-- **driver-memory**

-- **executor-memory**

-- **num- executors**

-- **executor-cores 3**

**Tiny Executors==**

**So if we have nodes =** 9

**Number of cores per node =** 3

**Executor cores = executors per cores =** 1

**Total memory for all nodes =** 100 GB

**Memory per node =** 100GB/9 = 11.1 GB

**Num-executors = no of cores \* no of nodes in cluster**

= 3\*9

= 27

**Executor memory = memory per node/ num of executors per node**

= 11.1GB/27

= 0.40 GB

------------------------------------------------------------------------------------------------------------------------------------------

**Fat executors**

If we have nodes =9

Executor cores = **1 executor per node**

Total memory per node = 100 gb

So memory per node = 11.1 gb.

Num-executors = 9

Executor memory = memory /num of executor per node

= 11.1/9

= 1.23 GB

**Tiny as well as fat executors:**- 5 cores having 1 executor

Leave 1 core per node = so cores will be 2

Num-executor = total cores\* total nodes/ 5

= 3\*9/5

= 5.4

Executor memory = mem per node/num of executors

= 11.gb/ 5.4

= 2 gb

Off heap overhead = 7% of executor memory

= 7%\* 2

= 0.014

Actual executor memory = 2-0.014 = 1.99 GB

* **Explain executor memory ?**

It is memory allocated to executor

memory per node/ num of all executors per node

* **Explain driver memory?**

**It is by default 1 GB.**

* **Explain executor –cores?**

Very important property , executor core can be assign explicitly.

We can take tiny or fat executors

It is how many executors assigned per cores

Tiny – one executor per core ,

Fat – one executor per node

Both –one executor per 5 cores

* **How to set cores in spark?**

Spark submit -- Total-execution cores parameters

Set spark configuration Spark.cores.max parameters

------------------------------------------------------------------------------------------------------------------------------------------**Explain two deploy modes?**

**Cluster mode -**

if spark driver exist within cluster then its cluster mode

job run in application master of yarn,

data movement is fast

no network problem as spark driver and other datanodes will be near

**Client mode –** if spark driver exist outside the cluster then it is client mode, so job will run outside the cluster.

Client itself send request to container ,for scheduling and monitoring task on worker nodes.

Data movement is slow as compare to cluster mode.

Network problems can occur.

* **How to know number of partitions in RDD?**

**Rdd.partitions.size()**

* **Tell the difference between client and cluster mode**

**Volume:**

Big data can store Petabytes/Exabytes of the data.

**Velocity**:

And this data process with higher speed .

**Variety:**

And data can be structured, semi-structured and un-structured data.

**Value:-**

and providing value ie high return of investment to organization.

**Veracity:**

* **Tell the 5 properties of the Hadoop?**

Open Source

Distributed

Fault Tolerance

Reliability

High availability

Scalability

#### **What is use of apply and unapply method?**

apply method - used to map data

unapply method - used to unmap the data.

#### **How can Auxiliary Constructors and primary constructor be defined in Scala?**

What is constructor”-

Constructor is use to initialize class variables

Constructor has same name as class

There are two types of constructor

Auxiliary constructor are used for constructor overloading

It uses this function for the same

The class can have one or more auxiliary constructor.

Class student(student\_id:Int,student\_name:String,salary:Double) {

Def this(student\_id:Int,student\_name:String,salary: Double) {

this(1,”heena”,70000)

println(“three argument auxiliary constructor”)

}

Def this(student\_id:Int,student\_name:String,salary:Double){

this(2,”salim”,56l478)

println(“three argment auxiliary constructor”) }

Def this (student\_id:Int,student\_name:String )

{this(1,”shaikh”)

println(“two argument auxilory constructor”)

}

}

}

Primary constructor

Very first constructor in class is known as primary constructor

The class contains only one primary constructor

* **What is Monalds?**

It is object which wraps another object like gift box.

It uses two options:-

unit() : it is method which does not return anything

flatMap()

* **What is trait?**

It is collection of abstract as well as non abstact methods,

It is like abstract class in java,

In order to use trait we need to extend it ,

We cannot create object of trait.

I order to use non-abstract methods of the traits we should implement abstract methods first

The class which extends the trait but does not implement its method should declare as abstract

We can also achieve multiple inheritance using trait

Eg trait employee extends HR with Finance

**What is difference between abstract class and trait?**

Trait can not have constructor,Abstract class can have constructor

We can extend multiple traits but not multiple abstract classes at same time.

**Which data formats are supported by the Spark Dataframes? And Ways to create dataframe?Why we use those formats only.**

* **How to create dataframe?**
* Create dataframe for csv,tsv,parquet, hive tables,mysql tables,local ,HDFS and S3 file,Avro,orc

Spark.read.format(“csv/tsv/mysql/hive/parquet/json/local/hdfs/S3/avro/orc”).load(“c://heena.txt”);

Eg spark.write.format(or.apache.spark.avro/mysql/hive/s3/json).save(“dtabasefile.avro”)

Csv – comma separated file

Tsv – tab separated files

Avro – to use avro format you should import library org.apache.spark.spark\_avro\_2.12:2.4.3

**Avro–**

**Developed by apache**

for big data processing

stores data in row format

it uses json format for defining data and datastructure

and then serialize it in compact binary format, so no need of code-generator.

Mchine readable avro file format

This format can be used in c,pig,java,python,scala,r

Eg.

Val data = Employee {

City:Int ,

Name:String ,

PinCode:Long

}

Then data.binaryformat()

**JSON-**

**Spark.read.json(“”)**

Java script object notation

Json consist of object ie key value pairs ,arrays and strings

{}[] bractets are use to hold objects and arrays

Human readable format

Mostly used while serializing data to and from web applications to server,

eg twitter ,google, facebook data to our server

Eg.

Val data = Employee {

City:Int ,

Name:String ,

PinCode:Long

}

If you want to multiple information about specific employee then you can use row formats ie json or avro.

Eg if I want to know employee\_name,employee\_id,employee\_ratings,employee\_salary for employee\_id=2522 then I can use avro or json formats

**Parquet/ORC (similar):-**

**Machine readable format**

It stores data and its metadata in columns

Requires Less storage

It can be used with Spark,Hive,Pig

If you have storage cost issue we can use parquet/orc as it saves lots of space.

Read write parquet files are fast because accessing columnar data is fast

It consist of header,body (data)and footer(metadata)

Mostly useful when you need only some fields from the tables

Eg. To find count of employee having salary > 50000;

Select count(\*) ,group by salary from employee where salary>50000;

In this case we only want salary column not employee id,employee\_name,employee\_city

So it is dependent on scenarios

* **Create dataframe from rdd or dataset or raw data**

**Eg .**

Var rawdata = Array(1,3,4,6)

Rawdata.toDF()

Var rdd = sc.parrallelize(Array(6,6,7,8)).toDF()

Var dataset = Array(45,6,7).toDS().toDF()

* **Explain ofDim of scala?**

If you want create dimentional array we can use ofdim method.

You need to import Array.ofDim method for it.

Amd pass data type and size of the array

Eg .

Import Array.ofDim;

Var a = ofDim[Int](3)(3)

Array( Array(0,0,0), Array(0,0,0),Array(0,0,0))

Var I, j =0

For ( i<- 0 to 2)

For (j <- 0 to 2)

{

a[i][j] = i+1

}

* **Explain case-classes uses?( uses other than providing schema)**

**Case classes are like normal classes ,we can create object of this class.**

\*It is use to provide schema to any data,rdd,daataframes,datasets.

Eg

Case class employee(employee\_id:Int,employee\_name:String,employee\_salary:Long)

var l = list.empty()

l.::(employee(2522,”heena”,50000),employee(6789,”salim”,3000))

\*case classes can be used for pattern matching

\* it also provides toString,hashCode,equals methods to object created

\*we can create copy of object

\*we don’t need to use new keyword to create object , as it has apply or un-apply method, which creates objects automatically and destruct as well.

\* we can access case class fields without var or val

Eg. employee.employee\_id

Employee.employee\_salary;

* **What is difference between registertemtable ,and createOrreplaceview?**

**RegisterTemptable –**it is temporary table in memory of cluster, it just poimter to DAG tree

**SaveasTable-** it is permanent table in parquet format

**CreateOreplaceTempView** – lifetime is same session ends, it can not be shared by other sessions.

**createGlobalTempView** – lifetime is till spark application ends, so it can be shared between multiple sessions.

* **Explain collections**

Collections

Collection of objects

Immutable/mutable

Iterator

Set

Tuple

List

Option

}

;}

Vector

Map

|  |  |
| --- | --- |
| **List** | **Arrays** |
| Strictly Immutable | Immutable in size  Mutable for data |
| Mixed elements,  No idex | Mixed elements  Idices |
| var list2 = List("Salim","Heena","String","Shaikh")  var list3 = 1::2::3::Nil  var list4 = List.fill(6)("Heena")  var list5 = List.tabulate(6)( x => x\*2)  var list6 = "Heena".toList  var list7 = ListBuffer(5,6,7).toList | var array2 = new Array[Any](10) //when we create empt object it shouldbe initialize  array2(0) = 5;  var array3 = Array.range(20, 100)  var array4 = Array.tabulate(0)(n =>n\*n)  var array5 = Array.fill(6)("Salim")  var list0 = List.range(1,13).toArray |
| Useful to operate over sequence easily |  |
| println(list1.reverse) ;  // List(6, 2, 4, 3) println(List.concat(list1,list2))  // List(3, 4, 2, 6, Salim, Heena, String, Shaikh)  println(list1.foldRight(0)(\_-\_));  // -5 operation will end to right side it will operate only for same type of elements  println(list2.foldLeft("Name:-")(\_+\_))  // 15 operation will end to left side  println(list1.isEmpty);  //it checks list is empty or not and return true or false  println(list1.apply(0))  // select elemets of the list from particular index  println(list1.filter( x => x!=2)); //filtering particular elements  println(list1.map(x => x\*2))  println(list1.:::(list2)) //join the list  println( list1.::(999))  //add the element in list at begining  println(list1)  println(list1.drop(3)) // drop 3 from list  println(list1.dropRight(3))  println(list1.dropWhile( x => x != 3))  println(list1.equals(list2))  println( (list1 zip list2).toMap )  //Map(3 -> Salim, 4 -> Heena, 2 -> String, 6 -> Shaikh)  //zip will produce tuplesof first element of list1 and list2  //second element of list1 and list2  //eg. (3,5),(4,1)(2,"String")(6,7)    println(list1.toSeq)  println(list1.toSet) //set of unique elements  println(list1.toString)  println(list1.toArray)  println(list1.toVector)  println(list1.toBuffer) //created arraybuffer | var a = Array[Int](10)  a(6) = 6  a(2) = 3  a(1) = 1    a.distinct  a.length  a.size  a.tail  a.head  a.toBuffer  a.toList  a.toBuffer  a.last  a.reverse  a.slice(0, 5) |
| There are two types of list  ArrayList()  LinkedList() | We can create two dimentional array too using of Dim methods  val twoarray = Array.ofDim[Any](3,3)  println(twoarray)  println(twoarray(2)(0))  twoarray(2)(1) =3    for{i <- 0 until 2  j <- 0 until 2  }  {  var temp = 1  temp += 2  twoarray(i)(j) = temp  println(twoarray(i)(j))    }  println(twoarray) |
| Listbuffer()  Listbuffer are mutable and  Add,update,delete elements  As well as we can change size of listbuffer  We can change size of listbuffer  var lbuffer = ListBuffer("hena","salim",5,2,2)  lbuffer += 9  lbuffer +=8  lbuffer -= 0 | ArrayBuffer(),  Are mutable, we can add,update,delelemets ,  As well as we can change sizeof arraybuffer  No need of index  var arrbuffer = scala.collection.mutable.ArrayBuffer(9,9,5,11,13)  arrbuffer += 3  arrbuffer ++= Seq(8,3,4)  arrbuffer --= Seq(3,4)  arrbuffer -= 7  arrbuffer.remove(2)  arrbuffer.append(3) |
| Duplicate elements allowed | Duplicate elements allowed |

|  |  |
| --- | --- |
| Map | Set |
| Immutable,mutable | Mutable,immutable |
| Collection of key value pair | Collection of mixed and unique elements |
| var simplemap = Map( 3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish") //not sorted    println(simplemap) //Map(3 -> Ajay, 1 -> Manish, 2 -> Ajinkya) | var set2 = scala.collection.immutable.TreeSet(3,4,8) |
| Useful to deal with key value pair | Useful to handle unique elements as it automatically removes the duplicate elements |
| var simplemap = Map( 3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish") //not sorted    println(simplemap) //Map(3 -> Ajay, 1 -> Manish, 2 -> Ajinkya)    var linkedhashmap = scala.collection.mutable.LinkedHashMap(3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish")  linkedhashmap.put(5,"Hannan")  //Map(3 -> Ajay, 1 -> Manish, 2 -> Ajinkya, 5 -> Hannan)  // soted based on insertion order  println(linkedhashmap)    var hashmap = scala.collection.mutable.HashMap(3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish")  println(hashmap) //Map(2 -> Ajinkya, 1 -> Manish, 3 -> Ajay)  //random order    var treemap = scala.collection.immutable.TreeMap(3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish")  println(treemap) // Map(1 -> Manish, 2 -> Ajinkya, 3 -> Ajay) //sorted based on keys    var sortedmap = scala.collection.SortedMap(3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish")  println(sortedmap) //Map(1 -> Manish, 2 -> Ajinkya, 3 -> Ajay) // sorted based on keys    simplemap += (1 -> "Vidya")  sortedmap += (1 -> "Vidya")  linkedhashmap += (1 -> "Vidya")  hashmap += (1 -> "Vidya")  treemap += (1 -> "Vidya")      // (3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish", 1 -> "Vidya")  treemap --= List( 1 ,3) //removes element based on keys only  treemap -= 1  treemap ++= List( 1 -> "Vidya",6 -> "Vidya")  println(treemap)// Map(1 -> Vidya, 2 -> Ajinkya, 6 -> Vidya)    linkedhashmap.put(2, "Vidya")  hashmap.put(2, "Vidya")    // we can do insert,delete but not update on treemap  linkedhashmap(1) = "Heena"  println(linkedhashmap)  // Map(3 -> Ajay, 1 -> Heena, 2 -> Ajinkya, 5 -> Hannan)      for{ (key,value) <- linkedhashmap }  println( key + value)    for {(key,value) <- linkedhashmap }  println( value + key)    // linkedhashmap.foreach(x => println( x.\_1)) //will print all keys of map object    println(linkedhashmap)  linkedhashmap.keysIterator.reduce( (x,y) => x + y) | var set1 = scala.collection.mutable.Set (3,4,"heena",6,"heena")  set1 += 3;  println(set1) //Set(heena, 6, 3, 4)  set1 ++= Seq(4,4)  set1 --= Seq(6,6)  set1 -= 2  println(set1) //Set(heena, 3, 4) |
| There are four types of Map  //HashMap implements Map interface , uses hash code, random order,put method works  //LinkedHashMap implements Map interface , uses double linked list , sorted based on insertion order,put method works  //TreeMap implements Map and sorted Map interface , it uses tree alorithm, sorted based on keys  //treemap is immuable in nature  //Sorted Map implements Map interface , sorted based on keys    var simplemap = Map( 3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish") //not sorted    println(simplemap) //Map(3 -> Ajay, 1 -> Manish, 2 -> Ajinkya)    var linkedhashmap = scala.collection.mutable.LinkedHashMap(3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish")  linkedhashmap.put(5,"Hannan")  //Map(3 -> Ajay, 1 -> Manish, 2 -> Ajinkya, 5 -> Hannan)  // soted based on insertion order  println(linkedhashmap)    var hashmap = scala.collection.mutable.HashMap(3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish")  println(hashmap) //Map(2 -> Ajinkya, 1 -> Manish, 3 -> Ajay)  //random order    var treemap = scala.collection.immutable.TreeMap(3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish")  println(treemap) // Map(1 -> Manish, 2 -> Ajinkya, 3 -> Ajay) //sorted based on keys    var sortedmap = scala.collection.SortedMap(3 -> "Ajay" , 1 -> "Heena", 2 ->"Ajinkya",1 -> "Manish")  println(sortedmap) //Map(1 -> Manish, 2 -> Ajinkya, 3 -> Ajay) // sorted based on keys | There are four types of Set  //sortedSet- sorted , implements Set interface TreeSet - sorted, uses treealogorithm, implemets sorted Set interface,we cannot update treeset  //HashSet - random order , uses hash coealgoritm , implements Set inteface  //linkedhashset - insertion order matters ,implements Set interface        var set1 = scala.collection.mutable.Set (3,4,"heena",6,"heena")  set1 += 3;  println(set1) //Set(heena, 6, 3, 4)  set1 ++= Seq(4,4)  set1 --= Seq(6,6)  set1 -= 2  println(set1) //Set(heena, 3, 4) |
| Duplicate values are allowed  Duplicate keys are not allowed  If it find duplicate key then it update the most recent value for the key | Duplicate elements are automatically removed |
|  |  |

|  |  |
| --- | --- |
| Tuple | Option |
| Immutable |  |
| Mixed elements | Same type of elements  Option has one type argument ,Eg Option[String]/Option[Int]  It can have some or none value |
| Var t = Tuple(4,”Heena”,50000) | val o:Option[String] = Some(“heena”) |
|  | It is useful for comparison ,pattern matching,handling return type  Eg. Def addition(x:Int,y:Int) :Option[z] {}  Def pattern\_matching(x: Option[Int]) = x match { Case Some(x) => x Case None => “?”} |
| methods | getOrElse(),isEmpty(),productArity()-size,filter,map, flatmap |
| Types | Types |
|  |  |
|  |  |

* **Explain difference between coalesce and repartition**

**Coalesce-** it is use to decrease number of partitions, it minimize shuffling

**Data can be distributed randomly**

**Coalesce are faster**

**Rdd.colesce(2);**

Eg. Array(4,2,3,8)

So coalesce will have

earlier

Node 1 – 4

Node 2 – 2

Node 3 – 3

Node 4- 8

Node1 -4,8

Node 2 – 2,3

**Repartition –** if you data is already partitioned and you partitioned it again is reparation.

It is always use to decrease the number of partitions

It do full shuffle

Data is distributed evenly

Repartition is slow

Rdd.repartition(5)

Eg. Array(4,2,3,8)

So repartition will have

Node 1 – 4

Node 2 – 2

Node 3 – 3

Node 4- 8

* **What is schemaRDD and how we can create in very detail.**

Schema rdd is objects and its schema ,

It can be created through case classes ,createSchemaRDD methods

1.using case classes

Heena.txt

1|2|heena salim

2|4|Ajinkya Ragalwar

4|99|shikh

Case class employee(employee\_id:Int,Company\_Id:Int,Salary:Int);

Eg , rdd1 = sc.textfile(“c://user/hshaik0/heena.txt).map( x=>x.split(|));.map( x=> employee(x(0).toInt,x(1).toInt,x(2))

2.Providing schema using StructType and Structfields

StructType (

Array(Structfield(“firstname” ,IntegerType),structfield(“lastname”,String)),

StructField(“Company\_id”, IntegerType),

structfield(“Salary”,LongType)

)

rdd1 = sc.textfile(“c://user/hshaik0/heena.txt).map( x=>x.split(|));.map( x=> employee(x(0).toInt,x(1).toInt,x(2));

sc.applySchema(rdd1,schema);

3. Using inferschema in case of dataframe

Spark.read.format(“csv”).schema(“inferschema”).header(“true”).load(“c://users/dowmloads/heena.txt”);

* **What is lazy evaluation**

Until and unless we ask for results they don’t work ,its called lazy evaluation

Egg. Rdd – rdd are lazily eavaluated ie.

Unless and until we invoke action on them ,they will not work.

It saves time, computations,storage.

* **Difference between Map-reduce and Yarn?**

Mapreduce =

|  |  |
| --- | --- |
| Mapreduce | Yarn(yet another resource negotiator) |
| Batch processing | Batch as well as real time processing |
| Mapper,reducers,job tracker,task tracker,combiner, | Global resourcema  nager,node manager,application master,container |
| Hadoop 1.0 | Hadoop 2.0 |
| Data processing and resource management | Resource management |
| Single point of failure | No single point of failure |
| Works with 4000 nodes | Hadoop can have more than 4000 nodes |

* **Explain cloudera versions?**

CDH5,CDH4,CDH3

* **Explain need of cloudera with Hadoop? Which platform you use to run Hadoop?**
* **Explain difference between RDD,dataframe and Dataset.**

|  |  |  |
| --- | --- | --- |
| * RDD | Dataframe | Dataset |
| Resilent distributed dataset | Resilent distributed dataset | Resilent distributed dataset |
| Array of Object | Table form | Combination of RDD and Data frame therefore any form |
| Process structured as well unstructured data | Process structured data | Process structured as well as unstructured data |
| If you want to give schema to data, you need to provide externally | Automatic schema detection is possible using inferschema | Automatic schema detection is possible using inferschema |
| We can perform transformation and actions | We can perform SQL and HQL queries | We can perform transformations , actions as well SQL and HQL queries |
| Came with Spark 1.0 | Came with Spark 1.3 | Came with Spark 1.6 |
| Compile time error checking | Runtime error checking | Compile time error checking possible |
| Need to optimize RDD externally using caching persisting  Partitioning  Broadcasting  Accumulating | Catalyst Optimizers are available  It is library of nodes and rules.  Tree:- collection of nodes  Rules :-function which can transform the trees.  It uses pattern matching to find and replace the tree .  So catalyst optimizer find the tree which need to be transform using pattern matching and skip for which pattern does not match.  Rules executes multiple times on trees  Optimizer has phases  Analysis:- analysis of logical plan  Eg. If we have sql query select name from employee  So initially we don’t know it is valid column name or not and its type.  Catalog has all the information about fields and their types  So catalyst uses rules and catalog object to track particular column exist or not and what is its type.  Then mapping input correct name  Determines which rules are creating same output and giving them unique ids  Mapping correct return type  logical optimization:- converting queries to simple forms.  physical planning:-catalyst generates multiple plan and choose best plan depending on cost.  code generation:-  it creates syntax tree then it passes to compiler which convert queries to java byte code | Catalyst Optimizers |
| Java serialization for distributing data and its schema to nodes across cluster.  And its expensive | Dataframe stores data into binary format in of-heap memory and perform its operations in off-heap memory so java serialization not needed and its cheaper | Dataframe stores data into binary format in of-heap memory and perform its operations in off-heap memory so java serialization not needed and its cheaper |
| Garbage collection handling is hard | Produce less garbage | Produce less garbage |
| Data can come by any datasources eg. Media streams | Data generally come from,  Csv,tsv,mysql,hql,avro | Data can come from any datasources |
| RDD aggregation operations are hard to use | dataframe aggregation operation is easy to use | Dataset aggregation operation is easy to use |
| Speed is slow | Speed is very high | Speed is very high. |
| Not space efficient | Space efficient | Space efficient |
| Ways to create RDD:-  Sc.parallelize()  Sc.textfile()  Spark.read.csv().rdd  Data.toDF().rdd  Data.toDS().rdd  Rdd2= Rdd1.map(x => x/2) | Ways to create dataframe:-  Data.toDF()  Spark.read.format(“csv format”).csv(//:/)  Spark.read.format(“tsv”).tsv(“//:/)  Spark.read.format(“xml format”).xml(//:/)  Spark.read.format(“hive”).option(  url= “jdbc url”,  driver= “mysql driver”,  username= “”  password=””  tablename=””  ).load()  Spark.read.format(“mysql”).option(  url=”jdc url”  driver= “”  username=””  password=””  tablenam=””  ).load()  Sc.parallelize(“//://).toDF() | Ways to create dataset:-  Data.toDS()  Spark.read.format(“csv”).csv(//:/).as[schema]  Spark.read.format(“tsv”).tsv(//:/).as[schema]  Spark.read.format(“xml”).xml(//:/).as[schema]  Spark.reas.format(“hive”).option  (url=”jdbc url”,  Driver=”hivedriver”,  Username=””,  Password=””,  Tablename=””)  .load().as[schema]  Spark.read.format(“mysql”).option  (url=”jdbc url”,  Driver=”mysql driver”,  Username=””,  Password=””,  Tablename=””)  .load().as[schema] |
| Supports java,scala,python | Supports scala ,java ,python | Supports scala and java only |

* **Difference between map and flatmap using examples?**

|  |  |
| --- | --- |
| **Map** | **FlatMap** |
| One input and one output  Same no of elements in input as well as output | One input and many outputs  More no of elements in output |
| It do only computation | It do computation and flatten results |
| Eg. a = Array(4,5,6),  a.map(x => x,x+1 )  Output – [4,5][5,6][6,7] | Eg. a = Array(4,5,6)  a.flatmap(x=> x,x+1)  Output – [4,5,5,6,6,7] |

* **Explain accumulators and broadcast variables ? How broadcast variables work in detail ?**
* **What is map join and broadcast joins?**

|  |  |  |
| --- | --- | --- |
| Accumulators | Broadcast Variable | Map Join/broadcast join are same |
| It is used to initialize variable | The variables are shared across multiple nodes | Smallest table stored in memory |
| Used with associative operations eg sum,min,max,mean,avg,sub,count  No need of for loops | Eg. if hr department and finance wants to access performance rating of the different employees,  If file is broad casted then HR members and finance members can access the files,  No need to wait for earlier action to complete | No reducer phase |
| High speed | High speed | High speed |
| Syntax –  Var ac = sc.accumelator(8)  Var x = rdd1.map( x => x+ ac ) | Syntax –  Sc.broadcast(rdd1) | select /\*+ MAPJOIN(table1)\*/\* from table1 join table2 where table1.id == table2.id |

* **What is inheritance**

Child class derived from parent class.

Eg, we derived from our parents

Eg, fruits derived from tree class

Etc

There are many type of inheritance

Single inheritance- one child born from one parent class

Multiple inheritance – one child born from multiple parent class ,not supported by java

Multilevel inheritance- one child born from one parent class which is again born from grand parent

Hybrid inheritance- two childs born from one parent , and then that two child’s yields one child again

* **What is final class**
* **What is static class**
* **What is abstract class**
* **What is difference between final and static variables?**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Static | Final | Abstract |
|  | Methods/class/variable invokes very first at compile time, no need to create object to access static members | The method/class/variable which can not be changed once defined | Abstract means empty |
| Variable | Static variable = can be used by any methods | Final variable = once it is declared it cannot be changed | Abstract variable does not exist |
| Method | Static method = it can call other static method only | Final method = final method cannot be overridden | Abstract methods are empty methods defined in abstract class as well as interface |
| Class | Static class = Nested classes ,which can use outer class methods ,  Static class cannot be extended  You don’t need to create object in order to use it.  You can access static class from anywhere in application  Eg, utility\_functions | Final class = final class cannot be extended once declared | Abstract class contains abstract as well as non-abstract methods  inorder to implement non-abstract methods you should provide implementation to all abstract methods. |
| Object | We cannot create object of static class | We can create object of final class | We cannot create object of abstract class |
| Inheritance | Static class cannot be extended | Final class cannot be extended | Abstract class can be extended |
| Use case | Eg . jar class need to use multiple time so we can create it as static class , so we can access it multiple time without creating object and  No one can extend it. | Eg. Update\_fb\_password omnly respective user will have access of this class.  No other class can change it | Eg. If you want to implement withdraw money method ,you need to provide implementation to swipecard,addpin,addamounttowithdraw() method |

* **What is polymorphism, What is overriding and overloading?**

One name and many forms are know as polymorphism .

Overriding /dynamic polymorphism – runtime evaluation

We can use parent class method and provide custom implementation to that.

Overloading/compile time polymorphism 0 compile time evaluation

Class have multiple methods with same name but they have

Different number of arguments

Sometime different return type

Sometime different access specifier

Sometimes different access modifier.

* **what is constructor**

use to initialize class variables

default constructor are invoked when object of class is created ,

There are three types of constructor—

* Default constructor()- any class can have only one default constructor
* Auxiliary constructor()- parameterized constructor
* Primary constructor – class can have only one constructor ie known as primary constructor and it can have parameters , but primary constructor includes all class members.
* **What is use of withColumn?**

Withcolumn – it is used with dataframes/datasets

**to change column datatype**

withColumn($start\_date,$”start\_date”.cast(“Integer”) )

**to update column - using when and other wise**

withColumn("source\_begin\_dt",

when(!($"source\_begin\_dt".isNull),$"source\_begin\_dt").otherwise(' ')

)

withColumn($”start\_date”, when($”start\_date” < 2/12/2020 ,$”start\_date”).otherwise(‘’) )

**to update existing column**

withColumn($”salary”,salary+30000)

**to add new column**

withColumn($”end\_date”, $”startdate”+5)

**to rename columns**

withColumnRenamed ($”salary”,$”increment\_salary”)

**drop columns-**

**.**drop($“salary”)

* **Explain Different ways to work with dataframe.**

**There are two ways to work with dataframe**

1. **Using select and selctExpr,withColumn,join,groupBy and collectlist,join,filter,alias,case when**
2. **registerTemptable and sql queries**

**1.select and selectExpr ,withColumn,withColumnRenamed,groupBy,collectlist,filter,join,alias,case when**

Df1.select,avg(salary),($id,$employee\_name,$salary,$skills,$department\_name)

.withColumnRenamed($”id”,$”employee\_id”)

.withColumn($salary,$salary+30000)

.withColumn($department\_name, when($”skills” === “scala”,”BigData” ).

when($skills===Java,”Java”).when (skills===”python”,”Python”).otherwise(“Not Valid Skills”))

.filter($”salary”,$salary <=25000 ”)

.groupBy($”department\_name)

Interview questions asked in happiest mind and clairvoyant,persistant.

* Draw the star pattern in java.
* Find out the top3 salary from employee database in scala spark sql
* Find out the top 3 rd salary from employee database
* What is super class of data types in scala .(Object class
* What is companion class and what is its use ?
* How to know smallest string using lexical algorithm?
* How to do sorting of the elements ?
* Explain bubble and merge sort in detail.
* How to find palindrome of the string?
* How to reverse the string?
* How to find even, odd, prime numbers from given list of numbers?
* how to find transpose of the element?
* Difference between order by an sort by;
* Difference between parquet and Orc?
* And what is file size we can store in parquet – 254 MB
* What is spark closure?
* What is higher order functions in scala?
* What is order take?
* **How will you join two dataframes normally ?**

**var** t1 = employee.join(department,

employee("dept\_id").equalTo(department("dept\_id")),

"inner")

t1.select($"department\_name",$"employee\_name").show()

* **How will you group elements by salary?**

Select \* from table group by salary;

t1.groupBy("department\_name").agg(

collect\_list("employee\_name") as "emp\_name",

avg("salary") as "average\_salary",

count("employee\_name")

sparksession.sql("select department\_name,collect\_list(employee\_name),collect\_list(salary) from ta group by department\_name")

* **What is the command used to fetch first 5 characters of the string?**

**import** org.apache.commons.lang.StringUtils.substring;

substring("Heena",1,5);

* **How to update element of dataframe ?**

By withcolumn ,when otherwise and case when

.withColumn("$salary" , when( $"average\_salary" < 15000 , 20000 ).when($"average\_salary" <10000,10000).otherwise("average\_salary"));

Or using sparksql quries using when otherwise or case when

sparksession.sql("select emp\_name,department\_name,case when average\_salary < 15000 then 20000 else case when average\_salary < 10000 then 10000 else average\_salary end end from tc ")

* **How to order elements?**

Select \* from table order by salary;

* **How to insert elements in table?**

sparksession.sql("insert into table employee\_table select \* from (select 'A')").collect().foreach(println)

dig more

* **How to join two dataframes?**
* **How delete work?**
* [**find Second Highest Salary of Employee?(click for explaination)**](http://www.complexsql.com/query-to-find-second-highest-salary-of-employee/)

Select salary from ( select salary from employee order by salary desc ) where rownum=5; oracle

[**find duplicate rows in table?(click here for explaination )**](http://www.complexsql.com/query-to-find-duplicate-records-in-table/)

Select id,name,salary count(\*) from employee group by name having count(\*) >1 ;

**Find top 2 salaries from employee table**

Select \* from employee order by salary desc limit 3;

Display dept details if they atleast one employee within it[two tables emp and dept.]

find the average salary of employees department wise along with other details

Display the dept details if there is no employee on that.

Display the details of persons getting top 5 highest salary

Find all employees who earn more than the average salary in their department

Display details of those employees who have switched job at least twice.[job\_history columns having values of no. of switch]

* [**How to fetch  monthly Salary of Employee if annual salary is given?(click here for Explaination)**](http://www.complexsql.com/query-find-monthly-salary-employee-annual-salary-given/)

Select Employee\_name,Salary/12 as ‘Monthly Salary’ from employee;

**6.What is Query to display first 5 Records from Employee table?(90% asked Complex SQL Queries Examples)**

**Answer:**

Select \* from Employee where Rownum <= 5;

**13.Display first 50% records from Employee table?**

select rownum, e.\* from emp e where rownum<=(select count(\*)/2 from emp);

**19.Select all records from Employee table whose name is ‘Amit’ and ‘Pradnya’**

Answer:

**Select \* from Employee where Name in(‘Amit’,’Pradnya’);**

**0.How to find count of duplicate rows? (95% asked in SQL queries for Interviews )**

Answer:

Select rollno, count (rollno) from Student Group by rollno Having count (rollno)>1

Order by count (rollno) desc;

**What is datatype class which give different datatypes?**

**What is infershema**

* **Explain project you worked**
* **Tell me about yourself**
* **How to load csv file in spark?**
* **What is difference between interface and abstract class?**
* **If employee table has emp\_id,emp\_name\_dept\_id,dept\_salary**
* **What is access modifier**
* **What are access specifier**
* **What is method signature**
* **What is difference between this and super keyword?**
* **What is SparkStreaming?**
* **Which directory secondary namenode points to ?**
* **Which directory primary namenode points to ?**
* **What is java serialization and catalyst optimizer?**
* **And department table has dept\_id and dept\_name ;**
* **How will you know deptname for the employee? Do in spark**
* **How will you count employee in particular department?**
* **What is interface**
* **What is string and difference between String and StringBuffer.**
* **What is exception?**
* **Explain multithreading in java**
* **What is thread and its synchronization**
* **What is lamda function?**
* **Difference between groupByKey and reduceByKey**
* **Explain optimization techniques of spark in detail**
* **1: Hash Partitioning**

The data is distributed according to hashcode

The formulae Is ,

Partition no = HashCode(Value) % no\_of\_partitions available

So if we have 1,2,3,4,5,51 values in dataset and no\_of\_partition is 3

Then we find

Partition no = HashCode(1)% 3 = 1

Partition no= HashCode(2)%3= 2

Partition no= HashCode(3)%3=0

Partition no = Hashcode(4)%3=1

Partition no = hashcode(51)%3= 0

**And if you provide No of partitions = number of cores then it will perform better.**

**The values will go to that container**

**2: Range Partitioning**

Here the data is distributed according to range

Eg 1 to 5 data will go in first bucket

5 to 10 data will go in second bucket

* **Explain Broadcast variables**

Sharing and storing variables across executers is known as broadcast variable.

Eg suppose finance department wants employee payroll data at appraisals time.

Then in this case the employee payroll data can be broadcasted over multilple nodes and

Multiple finance department members can aceess payroll data.

It will increase speed for computation.

Eg2 – Suppose HR members want to access performance rating for particular employees.

Then employee data can be broadcasted over multiple executers.

And each HR member can accees performance data at same time from different data nodes independelty.

Syntax= sc.broadcast(“data.csv”)

**Explain accumulators**

Accumulators is like flag to initialize variables/values.

It only supports associative operations eg.

Count

Sum

Max

Min

Mean

### **Syntax**

Val accum = sc.accumulator(0)

Sc.parallelize(Array(1,2,3)).foreach(x => accum +x);

* **difference between Hadoop and Spark.**

**Speedy – 100 Times faster than Hadoop**

Achive through partitioning ,caching and Persistance ,broadcast variables,accumulators

**Deployment** – can be deploy through mesos,yarn or standalone mode

**Polyglot-** can be written in java,scala,python,R

**Streaming –** offers real time processing using sparkstreaming

**Easy** – Simple to learn

**Smaller code –** smaller code as compare to hadoop and java

**Higher functions available** – to write complex logic

**Scalable –** we can scale the cluster

* **When creating an RDD, what goes on internally?**

**Explain spark architechture.**

* **What is fault tolerance in Spark? 263. What happens to RDD when one of the nodes on which it is distributed goes down?**

Spark Achieved fault tolerance using RDD lineage .

The rdd lineage is logical execution plan having information about actions and transformation.

The lost data can be recovered from lineage graph

* **What is fault tolerance in Hadoop?**

Recovering lost data is known as fault tolerance.

In hadooop we have 3 replicas of data stored by default ,

Two replica on different datanodes of same rack and

One replica is on different rack.

* **What is difference between groupByKey and reduceByKey?**

Both are transformations and are lazily evaluated, both works with Key

**Group by key –** group the element by key

Eg . grouping employees department wise

So , it will group all the employees having same department name

(Heena,Hadoop),(Kamal,Hadoop), (Manoj,Oracle)

Heena ,Kamal = Hadoop

Manoj = Oracle

**Reduce By Key =** it will reduce elements by key .

Ie it fetch all the values for particular key and then add ,multiply, subtract values.

Eg if we have 1,3000

2 ,5000

1,3000

Then we will get (1,8000),( 2,5000)

* **How to get number of partitions running in spark?**

getNumPartitions

* **How to know datatype of already defined columns?**
* **Dataframes all the methods**
* **What is spark driver?**
* Spark driver creates the spark context -which communicates with clustermanager and launchec job through executor on worker nodes.
* It can submit or cancel job
* It initializes accumulator, broadcast variable
* **What are actions and transformation**

Transformation transforms the data eg. groupByKey,reduceByKey,sortByKey,map,flatMap,filter

Transformations are lazily evaluated

Actions are results of transformations – eg count, collect(),take(),top(),foreach.println

* **How to convert dataset and dataframe into RDD?**

We can convert rdd and dataset into dataframe– toDF()

We can convert rdd and dataframe into dataset – toDS() or as[Schema]

We can convert dataframe and dataset into rdd - .rdd /makeRDD()

### **6. What is YARN?**

### **YARN is useful for resource management and job scheduling.**

YARN :-

* yet another resource negotiater
* mapreduce 2

YARN conisit of 3 units :-

**Global resource manager:-**

Allocates resources to hadoop appliations

schedule task on datanodes uses fifo or fair scheduler(equally shares resource across multiple clusters)

**Node manager:-** its is installed on every node and manages resources available on particular node.

**Application Master:**­-works with node manager to start container -which actually launches and monitor resouces.

Advantage over mapreduce:-

Mapreduce :- Batch processing

YARN :- Real time processing

Datanode -75



Application Master

Creates container processes and use resource information reports created by node manager to help container process

Container

Actually launches and monitor resouces

Node Manager= Task Tracker

Manages resources on particular datanode

By creating and sending resource information reports(active,inactive re souces) to Resouce Manager and application master

Allocates resources to hadoop appliations

schedule task on datanodes uses fifo or fair scheduler(equally shares resource across multiple clusters)

Gloabal resorce Manager

* **Explain Hadoop architechture.(HDFS and Mapreduce)**
* **What is InputSplit and RecordReader?**
* **Explain InputFormat?**
* **What is InputSplit and RecordReader?**

### **Explain about the partitioning, shuffle and sort phase in mapper and reducer? What is the purpose of “RecordReader” in Hadoop?**

* **Diffrence between inpusplit and block?**
* **What is a partitioner and how the user can control which key will go to which reducer? Whole mapreduce flow in detail?**

Hadoop has two critical components

**HDFS – data processing**

**Mapreduce – application processing**

**HDFS consist of two main units**

Data node

**Mapreduce consist of**

File

11,heena,3000

122,ajinkya,4000

34,sajiy,500

11,hasan,7899

Job –Tracker

Program

Task Tracker

Input split –logical division of data using InputFormatClass(TextInputFormat,KeyValueInputFormat,FileInputFormat)

Contains byteoffset, actual data

Byeteoffset ,(1,heena,3000 )

Byteoffset, (122,ajinkya,4000)

Byteoffset ,(34,sajiy,500)

Byteoffset,(1,hasan,7899)

B1

D

D

D

D

D

D

physsical division of the data

Record Reader-read record one by one

B2

D

D

D

D

D

D

B5

D

D

D

D

D

D

B4

D

D

D

D

D

D

B3

D

D

D

D

D

D

(1,heena,3000 ) (122,ajinkya,4000), (34,sajiy,500) ,(1,hasan,7899)

Mapper – read the data in key value format

Mapper1

D

D

D

D

D

D

Mapper4

D

D

D

D

D

D

Mapper3

D

D

D

D

D

D

Mapper2

D

D

D

D

D

D

1,(heena,3000 ) 122,(ajinkya,3400) 34,(sanjay,500) 1,(hasan,7899)

Combiner – combines the values based on keys(optional)

1,(heena,3000 ) 34,(sanjay,500) 122,(ajinkya,3400)

1,(hasan,7899

Partitioner- only decides which key,value pair will go to which reducer

1,(heena,3000 ) 34,(sanjay,500) 122,(ajinkya,3400)

1,(hasan,7899

Sorting –

1,(hasan,7899 34,(sanjay,500) 122,(ajinkya,3400)

1,(heena,3000 )

Shuffling- data is moved to respective reducers

Reducer4

D

D

D

D

D

D

Reducer3

D

D

D

D

D

D

Reducer2

D

D

D

D

D

D

Reducer1

D

D

D

D

D

D

Program- the programs sent by client are applied on the this shuffled data .

Eg. Min,max,summation,join

HDFS

Mapreduce

Jobtracker

Maintans active deactive tasktracker report and send to namenode

Namenode

* Manages datanodes by
* If datanode fail it chooses new namenode and replicate data to new namenode
* Maintains following information

File name

blocks information,

Data is store in which block,

Block is store in which datanode

Data size,

data replica information

B

B

D

B

Dat1

B

D128MB

DDD

Datanodes

Sends active ,deactive blocks reports to namenode,

Actually stores a data

Edit logs-

current system state

eg information about data ie deleted, data ie inserted , data ie updated

fsimage getloaded with edit logs

FsImage

Current Filesystem image

Checkpointing

Tasktracker

### Partitioner do partitioning of the keys of the intermediate map-outputs.

### It uses hash function for it.

### There are two partitioner program.

### HashPartitioner.

### A custom partitioner is implemented to decide which keys go to which Reducer.

### Method use to create custom partition is :-

### public class SamplePartitioner extends Partitioner {

### @Override

### public int getPartition(Text key, Text value, int numReduceTasks) {

### }}

* **How many job tracker exist per one name node?**

1

* **How many reducers run in sqoop?**

0

* **How many mappers run in sqoop by default?**

4

* **How mappers we can provide in sqoop?**

-- m 3

* **What is object?**

Instance of class,

It has identity

It has state

It has behavior

Eg . if we have chair ,

chair has identity as chair

Its state is solid

Behaviour – its movable

* **Can we extend object?**

Yes, App class can extend obejct

* **What is public static void main?**

**Public –** this members can be access from anywhere in the program

**Static –** this members invokes very first at compile time only,we don’t need to create any object inorder to invoke this method

**Void –** it does not return anything

**Main –** jvm searches for main method ;

* **How will you filter records in Hadoop?**
* **What is sqoop incremental load? Write query in sqoop for incremental load**

**Sqoop incremental load is like normal load but it has three columns**

Incremental Mode – append/last-modified

Check-column- column which will find out rows to be updated,it should not be char or varchar

Last value - it is optional we can use if we want the rows which is greater than particular value

sqoop import \

--connect jdbc:oracle:thin:@enkx3-scan:1521:dbm2 \

--username wzhou \

--password wzhou \

--table STUDENT \

--incremental append \

--check-column student\_id \

-- last-value 10 \

-m 4 \

--split-by major

**So eg. If you have check column as time , then last value will check maximum value for previous operation**

**what**

* **How will you use splity by?**

Splits data evenly based on that column eg lets say we have employee table and we are splitting data based on department name so we can use split by

* **Why we use boundry query?**

If split by used and then also if query is not giving performance then we use boundry query.

Which will divide the data into ranges , we need to specify minimum and maximum value-

sqoop import \

--connect 'jdbc:mysql://.../...' \

--direct \

--username uname --password pword \

--hive-import \

--hive-table query\_import \

--boundary-query 'SELECT 0, MAX(id) FROM a' \

--query 'SELECT a.id, a.name, b.id, b.name FROM a, b WHERE a.id = b.id AND $CONDITIONS'\

--num-mappers 3

--split-by a.id \

--target-dir /data/import \

--verbose

* **How will you perform inter-cluster datacopy?**

Hadoop distcp hdfs://trpahadap01//smith/idrp/heena.txt hdfs://trihadap01.vm.searshc.com/amith/idrp

* **Who converts inputsplit to physical unit in mapreduce?**

Inputformat

* **Which database the sqoop metastore runs on?**
* **What is sqoop-merge?**
* **What is sqoop metastore?**
* **What exceptions u faced in scala?**
* **What is the default database of Apache Sqoop? --Mysql**
* **Explain sqoop architechture**
* **Optimization techniques for hive and Pig?**
* **38. How can we do optimization in Pig?**

**1: Adding compression techniques**

set io.compression.codec.lzo.class com.hadoop.compression.lzo.LzoCodec

set pig.tmpfilecompression true

set pig.tmpfilecompression.codec lzo

**2:Setting Mappers:-**

set mapred.compress.map.output true

set mapred.min.split.size 524288;

SET mapred.max.split.size 134217728

SET pig.maxCombinedSplitSize 4000000

**3:Setting reducers**

SET mapreduce.map.java.opts: -Xmx3072m

SET mapreduce.reduce.java.opts: -Xmx6144m

* **Explain flatten ,bag,group,tuples and maps in Pig?**
* **Explain difference between group and co-group**
* **Diffrence beween dump ,describe,illustrate and explain in pig?**
* **What are different types of join in Pig?**
* **What is serde propertie s in Hive?/**
* **What is Managed table?**
* **What is external table?**
* **Tell me difference between managed table or external table**
* **How to load the data from the hdfs ?**
* **How to load the data from the local?**
* **Tell the commnd to rename the tabl column?**
* **How to add new columns into hive table ?**
* **How to change the data types of the hive table?**
* **What is clustering?**
* **What is portioning and types of the partitioning?**
* **What is custom partitioning?**
* **What is dynamic partitioning?**
* **How to do dynmic patitioning?**
* **How to add the partition to hive table?**

**alter table gold\_\_sales\_sears\_pos\_tax\_general**

**add partition (transaction\_year\_nbr='1996') location '/gold/transaction/pos/sears/tax\_aud\_s5\_general/transaction\_year\_nbr=1996';**

**hive -e "alter table gold\_\_sales\_sears\_pos\_payment add if not exists partition (transaction\_year\_nbr=$PROCESS\_YEAR);"**

* **How to drop the partition to hive table?**
* **Explain map side join in Hive?**
* **Difference between windowing ,partitioning and bucketing?**

### **How to update records in Hive?**

To update records in Hive, use the following syntax-

**UPDATE** <target **table**>

**SET** <**set** clause list>

[ **WHERE** <search condition> ]

###### ***What are the Binary Storage formats supported in Hive?***

By default Hive supports text file format, however hive also supports below binary formats.

Sence Files, Avro Data files, RCFiles, ORC files, Part files

**Sence files:** General binary format. Splittable, compressible and row oriented. a typical example can be. if we have lots of small file, we may use sence file as a container, where file name can be a key and content could stored as value. it support compression which enables huge gain in performance.

**Avro datafiles:** Same as Sence file splittable, compressible and row oriented except support of schema evolution and multilingual binding support.

**RCFiles:** Record columnar file, it’s a column oriented storage file. it breaks table in row split. in each split stores that value of first row in first column and followed sub subsently.

**ORC Files:** Optimized Record Columnar files

###### ***What is the Hive configuration precedence order?***

There is a precedence hierarchy to setting properties. In the following list, **lower numbers take precedence over higher numbers**:

The Hive **SET** command

The command line **-hiveconf** option

hive-site.xml

hive-default.xml

hadoop-site.xml (or, equivalently, core-site.xml, hdfs-site.xml, and mapred-site.xml)

hadoop-default.xml (or, equivalently, core-default.xml, hdfs-default.xml, and mapred-default.xml)

**Explain about SORT BY, ORDER BY, DISTRIBUTE BY and CLUSTER BY in Hive.**

**How Hive can improve performance with ORC format tables?**

**Difference between Pig and Hive**

* **How will you consume this CSV file into the Hive warehouse using built SerDe?**

SerDe stands for serializer/deserializer.

A SerDe allows us to convert the unstructured bytes into a record that we can process using Hive.

SerDes are implemented using Java. Hive comes with several built-in SerDes.

Hive provides a SerDe for working with CSV files. We can use issuing following commands:

CREATE EXTERNAL TABLE sample

(id int, first\_name string,

last\_name string, email string,

gender string, ip\_address string)

ROW FORMAT SERDE ‘org.apache.hadoop.hive.serde2.OpenCSVSerde’

STORED AS TEXTFILE LOCATION ‘/temp’;

Now, we can perform any query on the table ‘sample’:

SELECT first\_name FROM sample WHERE gender = ‘male’;

### **What is indexing and why do we need it?**

### **What is the default database provided by Apache Hive for metastore?**

By default, Hive provides an embedded Derby database metastore.

* **Mention what are views in Hive?**

**Can the name of a view be same as the name of a hive table?**

No. The name of a view must be unique in one database.

* **How can you stop a partition form being queried?**

By using the ENABLE OFFLINE clause with ALTER TABLE satatement.

* **Will the reducer work or not if you use “Limit 1” in any Hive query or If we use the "Limit 1" in any SQL query in Hive, will Reducer work or not.**

Ans. It will work , as hive fetch random record.

* **What are collection data types in Hive?**

**Is there a date data type in Hive?**

Yes. TIMESTAMP data stores date in java.sql.timestamp format

### **What is the purpose of storing the metadata?**

### People want to read data with a particular schema

### Eg. Manager may be interested to look attendance of the particular emplyoee

### Principle may be interested to look marks of the particular student. So to access particular field again and again and to avoid efforts, we use schema.

### **What is the purpose of Hive Driver?**

### Hive Driver is used for compiling, optimizing and then executing the Hive query.

* **Explain Hive architechture.**
* **What is Hive?**
* **What is the default value of map and reduce max attempts? 4**

### **What can be optimum value for Reducer?**

### Value of Reducers can be: 0.95

### 1.75 multiplied by ( \* < number of maximum resources per node>)

### Increasing number of reducers

### Increases the framework overhead

### Increases load balancing

### Lowers the cost of failures

### Detail description of the Reducer phases?

### Shuffle: Providing the key value pair from mapper to reducer.

### Sort: grouping and sorting the reducer inputs on the basis of the same keys.

### The shuffle and sort phases occur simultaneously;

### Reduce: reduce(WritableComparable, Iterable, Context) method is called to

### Adding the values corresponding to same keys from grouped key value pairs. The output of the reduce task is typically written using Context.write(WritableComparable, Writable).

### of the reduce task is typically written using Context.write(WritableComparable, Writable).

### How a number of partitioners and reducers are related?

### The total numbers of partitions are the same as the number of reduce tasks for the job.

### **What are the two main components of ResourceManager?**

### Scheduler

### It allocates the resources to various running applications such as memory, CPU, disk etc.

### ApplicationManager It accepts jobs, resource for executing the application specific ApplicationMaster and provides the service for restarting the ApplicationMaster container on failure.

### **What is the function of NodeManager?**

### It manages resources,

### Monitores resource usage (cpu, memory, disk, network)

### Reporting the same to the ResourceManager

### **What is the function of ApplicationMaster?**

### ApplicationMaster is per application

### Moves appropriate resources from the Scheduler

### Track resource status

### Monitor resource progress.

### **Command to format the NameNode?**

### $ hdfs namenode –format

### **How can you overwrite the replication factors in HDFS?**

### hadoop fs –setrep –w 5 /user/heena

### 

### **What is the default replication factor?**

### 3

### **What is the default location where “Hive” stores table data?**

### /user/hive/warehouse

Data warehouse location can be set using hive.metastore.warehouse.dir  property in the hive-site.xml.

* **How do “reducers” communicate with each other?**

“Reducers” do not communicate with each other, run in isolation/runs separately.

* **What is free form query in sqoop?**
* **What is windowing in hive?**
* **If I created table in hive and id column is Int , so id was 1**

**If I created table again with id column as double and loaded same data what will be id?**

File.txt

Id

1

2

3

Create table sample( id int)

Row format delimited

Location ‘/smith/idrp/file.txt’

Hive table

Id

1.0

2.0

3.0