1. How do you submit spark job?

-spark-submit[options] <app file> [app arguments]

2. What are the options for spark job?

Master: URL of master, Num-executors, Driver-memory, Executor memory, Executor -cores, Packages, Deploy-modes, Conf.

3.Do you know about spark broadcast join? Can you give me an example?

If one of the data frame is much smaller than the other, spark broadcast join is always a good option for improved performance, It pushes the smaller data frame into each of the spark executors on each of the worker node and avoids expensive sort and shuffle. But the file should be small enough to fit into the memory of each executor. Spark SQL is smart enough to configure the broadcast join (itself).

4. Can you tell me about spark client and cluster modes?

In cluster mode, the driver runs in the spark application master. The same process is responsible for both driving the application and also requesting the resources from yarn and this process runs inside the YARN container

Whereas in client mode, the driver is launched directly within spark submit process which acts as a client to the cluster. The input and output of the application is attached to the console. So it is easy for development and debugging

5. Differences between coalesce and repartition in spark?

Coalesce uses existing partitions to minimize the amount of data that’s shuffled. So, it avoids the full shuffle and show results in partitions with different amounts of data.

Repartition creates new partitions and does a full shuffle and store the results in roughly equal sized partitions.

Coalesce may run faster than repartition. Unequal sized partitions are generally slower to work with equal sized partitions.

6. groupByKey vs reduceByKey vs aggregateByKey Vs combineByKey()

**groupByKey()** is just to group your dataset based on a key. It will result in data shuffling when RDD is not already partitioned.

When **groupByKey()** is applied on a dataset of (K, V) pairs, the data shuffle according to the key value K in another RDD. In this transformation, lots of unnecessary data transfer over the network.

Spark provides the provision to persist data to disk when there is more data shuffling onto a single executor machine than can fit in memory.

**reduceByKey**: Spark RDD reduceByKey function merges the values for each key using an associative reduce function. Basically, reduceByKey function works only for RDDs which contains key and value pairs kind of elements (i.e. RDDs having tuple or Map as a data element). It is a transformation operation which means it is lazily evaluated.

The reduceByKey works much better on a large dataset as compared to. That's because Spark knows it can combine output with a common key on each partition before shuffling the data.

**combineByKey** can be used when you are combining elements, but your return type differs from your input value type. Spark combineByKey is a transformation operation on PairRDD (i.e. RDD with key/value pair). It is a wider operation as it requires shuffle in the last stage.

**aggregateByKey** is same as combineByKey and there is slight difference in functioning and arguments. The aggregateByKey function is used to aggregate the values for each key and adds the potential to return a different value type.

The three parameters of aggregateByKey function,

zeroValue: As we are finding maximum marks out of all subjects we should use Double.MinValue (which is also known as an accumulator)

seqOp: Sequential operation is an operation of finding maximum marks (operation at each partition level data)

combOp: Combiner operation is an operation of finding maximum marks from two values (operation on aggregated data of all partitions)

7. What is Yarn?

Yet Another Resource Negotiator is the resource management layer for apache Hadoop ecosystem. It has 2 types of hosts. Master/resource manager and worker/node manager.

Resouce manager: It is the master daemon that communicates with the client, tracks resources on the cluster and orchestrates, works by assigning tasks to the node manager

Node manager: It is the worker daemon who is responsible for containers, monitoring their resource usage(cpu/memory/disk/ network) and reporting the same to the resource manager/schedules.

For each running application(one or more tasks) A special piece of code called application master helps with negotiating resources from the resource manager and works with nodemanagers to execute and monitor the tasks.

8. Do you know about small files problem in hdfs?

In hdfs, the files are split and stored in the data node to hdfs block of 128mb. If the size of the file is smaller than the block size, hdfs cannot handle it efficiently.

NN memory is consumed: Name node stores all the metadata of the blocks in it which is ususally 150b and say 10 million files are stored into hdfs, then that would be almost take upto 1.5gb of name node memory

Small files degrade map reduce performance: While processing the blocks, the mappers picks blocks randomly for processing that requires large number of disk i/o performance. Performance is always faster when processing few large sequential data than several small random data files. Also each block is picked my one map and each map is processed in one JVM. Many small files implies many blocks which means spinning up of many JVM’s. This will again need large no. of nodes. If the cluster size is small the tasks will be queued and will cause delay.

9. Hive Dynamic Partitioning?

Static partitioning columns: In DML/DDL involving multiple partitioning column’s whose values are known at compile time.

Dynamic Partitioning Columns: Dynamically determining which partitions should be created and populated while scanning the input table. In this case, columns whose values are known only at the execution time.

Usually hen loading files(big files) into the hive tables. Statics partitions are preferred which saves time in loading data compared to dynamic partition.

In dynamic partition, every row of the data is read and data is partitioned through a mapreduce job into the destination tables depending on the certain field in the file.

So usually dynamic partitions are useful when you are doing sort of an etl flow.

To enable dynamic partitions we need to set the following parameters, we need to set the following parameters in hive shell. By default it is false.

Set.hive.dynamicpartition=true

Set.hive.dynamicpartition.mode=non strict

10. Hive partitioning and bucketing?

Partitioning: Organizing tables into positions for grouping similar type of data together based on column or partition key. Each table can have one or more partition keys which determines how the data is stored which allows us to efficiently query the data.

Bucketing: Data in each partition may be further divided into buckets based o the value of each function of the column of the table. Each bucket is just a file in the partition directory. Hive determines the bucket number for a row by using aformula

Hashfunction(bucketing column) modulo(nu of buckets). Set hive.enforce.bucketing=true

Partitioned by is used to divide the table into the partition and can be divided into buckets by using the clustered by command.

11. What is scd type-2 and how do you implement it?

SCD 1 = overwrites old data with new data

Scd 2 = add new rows with version history. It allows to keep track of complete history. But table may grow without limit

Scd3 – adds new rows and maintains a history

(I will choose spark to implement scd-2 solution. As spark is a powerful data engineering framework and it delivers fast processing by in memory caching of data between iterations)

Spark integrates different files ad can easily infer schema and perform sql operation on data.

1. In spark context, Read the source file and register as temporary table – STG
2. Identify the records in the target that are not matching with STG dataset by left join TGT and STG on the keys where stg.key is null and register as t1
3. Similarly identify the target records that are matching with stg based on keys and flag it or update it with the current date and register as t2
4. Union 3 tables STG+T1+T2
5. Overwrite the target data set on S3/ File system

12. AWS EMR VS EC2

Amazon Elastic cloud compute is web service provided by amazon. It is provides secure and resizable compute capacity in the cloud. EC2 is well integrated with most aws services like s3, sqs, lambda

Emr is a collection of ec2 instances with managed Hadoop framework installed and configured on them. It is easy, fast and cost effective to process data across dynamically scalable ec2 instances. You can choose the required Hadoop tools & applications. The S/w is installed and configured by amazon. So, we don’t have to warns about infrastructute and administrative tasks

13. Do you know about aws lambda?

Lambda is a service which runs your code in response to events such as s3 object creation or modification on s3 bucket or table update in a amazon dynamo db. Or in response to a http request using amazon api gateway or api calls made using aws sdk’s

Automatically manages underlying resources. We don’t have to worry about provisioning/managing servers.

All we need to do is supply the code we want to run on an event.

But lambda must complete execution in 5 minutes

Def handler name(even, context);

Msg= ‘hello’ return some value

14. How do you copy or move s3 files of specific pattern ?

AWS s3 cp/mv s3:// src loc s3:// trgt loc --recursive --exclude “\*” – include “\*.log”

15. What is IAM and What are differences between users, groups and roles?

IAM allows you to manage users and their level of access to aws console. IAM is global. Users/groups/ roles are available globally.

Users - End user

Groups – Collection of users under one set of permissions.

Roles – You can create roles and can assign them to aws resources

\*\* for an ec2 instance , we assign a role to access s3 and then it doesn’t require any password or secret access key

16. Security Groups.

Security groups are just like firewall.

We can add rules to each security group that will allow traffic to or from the instance that is inbound and outbound traffic. And security groups allow outbound traffic by default.

17. Do you know about AWS SQS?

It’s a simple Queue Service where you can send, store and receive messages.

There are 2 types of message queues

1. Standard message which ensures atleast one-time delivery and order of message is not guaranteed
2. FIFO Queue which process atlease once and make sure the messages are delivered in the order they are sent.

18. Union vs Union All vs Join?

Union operator is used to combined result set of 2 or more select statements. It removes duplicates. Each select statement within the union must have the same no. of fields with similar data type.

Union all shows duplicates also in result set.

Join clause combines data from 2 or more tables based on a commnn relational column between them

19. Different joins in sql

Joins are used to combine 2 or more tables based on related columns between them

Inner join or join: This is the default join which shows the records that there have atleast one matching values in both tables.

Left Join: Returns rows from the left table and the matched rows from the right table. The results contain the rows from left table even if the there is no matching value in right table and returns null in the columns that doesn’t have a matching value

Right join: Returns rows from right table and the matched rows from the left table. The results contain the rows from the right table even if there is no matching value in the left table and returns null in the columns that doesn’t have a matching value

Full outer join: Returns all rows for which there is a matching in either tables. A full join combines the effect of applying both left and right joins. The result will be equivalent to performing the union of left and right outer queries

Cross Join: It doesnot have a where clause and returns the cartesian product of tables involved in the join. Cartesian product result set is the number of rows in the first table multiplied by the number of rows in second table. Cross join combined with a where clause functions like inner join.

20. How do you find the duplicated in a Table?

Select Column(s), Count(\*) from table

Group by columns having count(\*)>1

These can be deleted uusing cte and row number feature

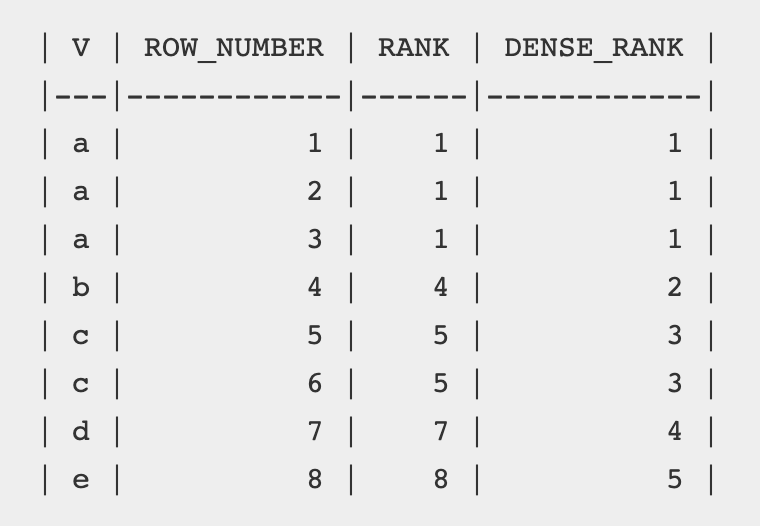
21. Rank Vs Dense\_Rank Vs Row\_number

**ROW\_NUMBER:** Returns the sequence and unique number for each group based on the fields applied in PARTITION BY clause. If PARTITION BY is not specified, the function treats all rows of the query result set as a single group.

**RANK:** This function will assign a unique value to each distinct Row, but it leaves a group between the groups.

Gap represents number of occurrence example - EmpName="atul" is repeated 3 times and has rank "1", the next rank will be 1+3=4 and same with the next value.

**DENSE\_RANK:** Dense\_Rank() Function is similar to Rank with only difference, this will not leave gaps between groups.



22. Masking the column:

<https://tunetotech.com/Post.aspx?post=9>