* What is Big data ?
* Big data refers to the datasets that are too large or too complex. It is not possible to handle big data with traditional data-processing application software.
* The three V’s of Big data ?
* Volume: Volume refers to the unimaginable amounts of information generated every second from social media, cell phones, cars, credit cards, images, videos, etc.
* Velocity: With velocity we refer to the speed with which data is being generated. Everyday 900 million photos are uploaded on Facebook, 500 million tweets are posted on Twitter etc.
* Variety: Variety in Big Data refers to all the structured, semi-structured and unstructured data that has the possibility of getting generated either by humans or by machines.
* Difference between Hadoop and Spark.
* Hadoop’s MapReduce reads and writes from a disk. Hence the processing speed is slow. Spark reduces the number of read/write cycles to disk and stores intermediate data in memory, hence faster-processing speed.
* Hadoop is designed to handle batch processing effectively. Spark is designed to handle real-time data processing effectively.
* Hadoop uses Java or Python for MapReduce . Spark uses Java, R, Scala, Python, or Spark SQL for the APIs.
* The PageRank algorithm is used in Hadoop. Graph computation library called GraphX is used by Spark.
* Kerberos Architecture.
* Kerberos is a computer network security protocol that authenticates the service request between two or more trusted hosts. Kerberos runs as a third-party trusted server known as the Key Distribution Center(KDC).
* The main components of kerberos are:
* Authentication server(AS): The authentication server performs the initial authentication.
* Ticket Granting Server(TGS): The ticket granting server issues the ticket for the server.
* Yarn Architecture.
* YARN stands for “Yet Another Resource Negotiator“. The main components of YARN are:
* Clients: It submits the map-reduce jobs.
* Resource Manager: It is responsible for assigning and managing the resources among all the applications.Whenever it receives a processing request, it forwards it to the corresponding node manager and allocates resources for the completion of the request accordingly.
* Node Manager: It takes care of individual nodes on a cluster and manages application and workflow and that particular node.
* Application Master: An application is a single job submitted to a framework. The application master is responsible for negotiating resources with the resource manager.
* Container: It is a collection of physical resources such as RAM, CPU cores and disk on a single node.
* Logical and Physical Plan in Spark.
* Logical plan refers to an abstract of all transformation steps that need to be executed. Physical plan is an internal enhancement or optimization for spark. It is generated after creation of the optimised logical plan.
* Three API’s of Spark.
* RDD or Resilient Distribution Datasets is a collection of records with distributed computing which are fault-tolerant , immutable in nature.
* DataFrames is a distributed collection of rows under named columns. In simple terms it looks like an excel sheet with column headers or a table in relational database.
* Dataset is a strongly-typed, immutable collection of objects that are mapped to a relational schema.
* Architecture of Spark.
* Master Node: In master node, we have the driver program, which drives the application. The code we are writing behaves as a driver program or if we are using the interactive shell, the shell acts as the driver program. First thing we do is, we create a spark context. Spark context is a gateway to all the Spark functionalities.
* Cluster Manager: Spark context works with the cluster manager to manage various jobs.The driver program & Spark context takes care of the job execution within the cluster. A job is split into multiple tasks which are distributed over the worker node.
* Worker Node: Worker nodes are the slave nodes whose job is to basically execute the tasks.
* Executor Node: Executor node reserves the cpu and memory resources to the worker node.
* Different Operations in RDD.
* Transformation: A transformation is a function that produces new RDD from the existing RDDs. Some examples of transformations are map(), filter(), flatmap().
* Actions: Operations that trigger computation and return values. Some examples of actions are count, top(), savetofile().
* Different Transformation of Spark.
* Narrow Transformation: These types of transformations convert each input partition to only one output partition. There is no shuffling. These are ono-to-one mapping. Some examples of narrow transformation are map(), filter(), flatmap(), etc.
* Wide Transformation: This type of transformation will have input partitions contributing to many output partitions.Each partition at the parent RDD is used by multiple partitions of the child RDD. There is shuffling. Some examples of wide transformation are groupByKey(), reduceByKey(), foldByKey(), etc.