Spark:

Sigle programme execution on 100 gb data

Multiple programmes on 100 gb data(faster execution with high speed)

Each programme for 10 gb

This will handle by spark

Spark works in cluser

Cluster – set of nodes

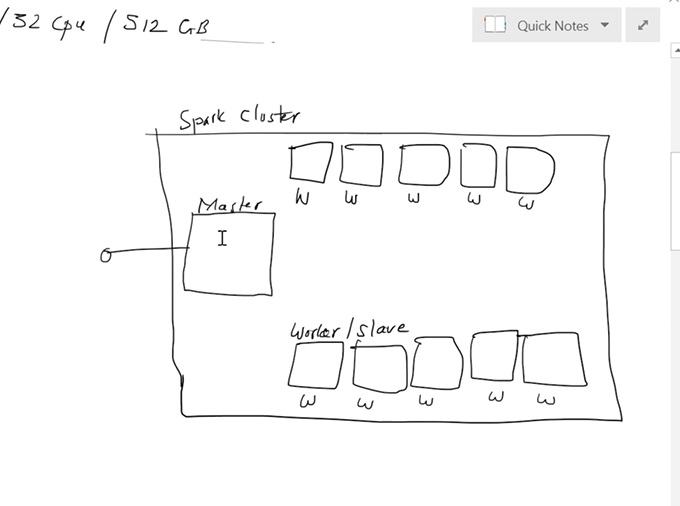
Node= physical machine

Instead of one single machine

Multiple machine (multiple nodes)

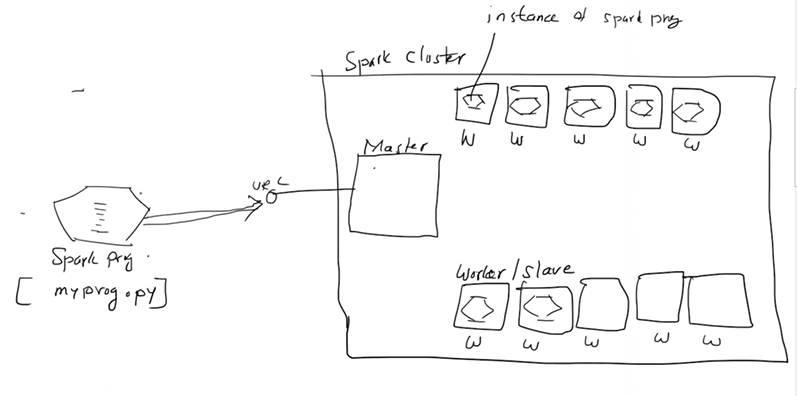
Each node have separate os,own software’s, infrastructure

Nodes or clusters may be on cloud or on-premise



Master node will be responsible to distribute work among worker nodes

Master node creates instances for progrme



The instances will be work on diff data not same data

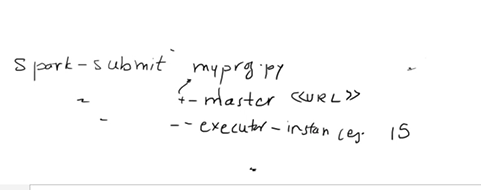
Both nodes and master will be referred by url

Whatever instance will receive master url will become master

Master decided how many instances should be created

Its not necessary now many nodes available is qual to how many instances created by master

The instances creation based on: conf of nodes and available data – master creates

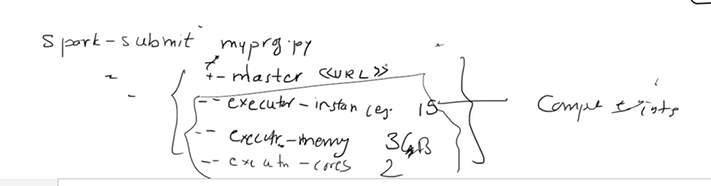


If nodes <instances

Some of nodes may have multiple instances

Executor – memory is ---- memory used in each instance

Executor-cores: 2 cores/total cores of each node assigned



If single node cluster- like local computer- where one node act as master and node

step 1: d/l java 8

java path: C:\Program Files\Java\jdk1.8.0\_351\

step 2: d/l python 3.10

Note: While Python installation check Add Python.exe to PATH

step 3: d/l spark-3.1.2-bin-hadoop2.7 (https://archive.apache.org/dist/spark/spark-3.1.2/)

unzip-->unzip to c:\

step 4: d/l hadoop 3.1.2 (https://github.com/cdarlint/winutils/tree/master/hadoop-3.1.2/bin)- winutils

create a folder in c: [c:\hadoop\bin] and keep winutils file here

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

ENVIRONMENT VARIABLES

HADOOP\_HOME: C:\hadoop

JAVA\_HOME: C:\Program Files\Java\jdk1.8.0\_351

PYSPARK\_PYTHON: C:\Users\tousifb\AppData\Local\Programs\Python\Python310\python.exe

SPARK\_HOME: C:\spark-3.1.2-bin-hadoop2.7

PATH: %JAVA\_HOME%\bin;%HADOOP\_HOME%\bin;%SPARK\_HOME%\bin

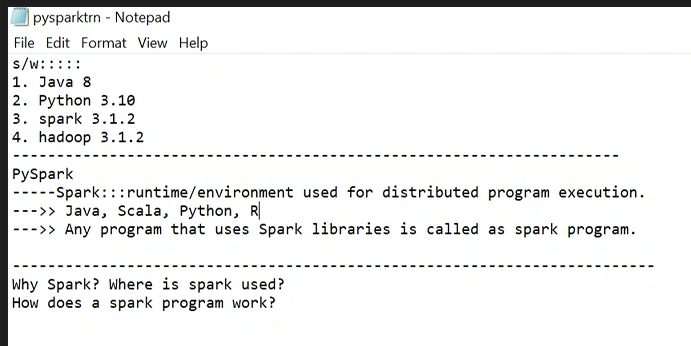
-----------------------------

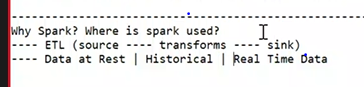
Errors that might occur

pypark module not found.

soln: pip uninstall pyspark

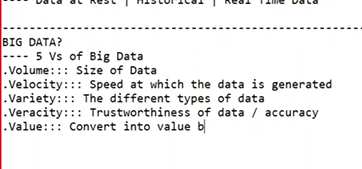
pip install pyspark==3.1.2





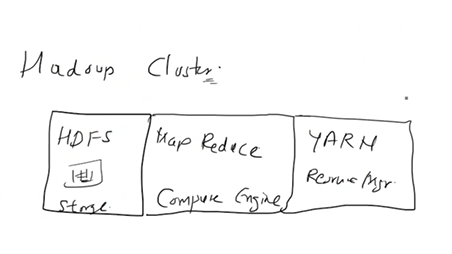
Big data: no perfect definition

A data which of machine is not capable of processing



Solution for bigdata?

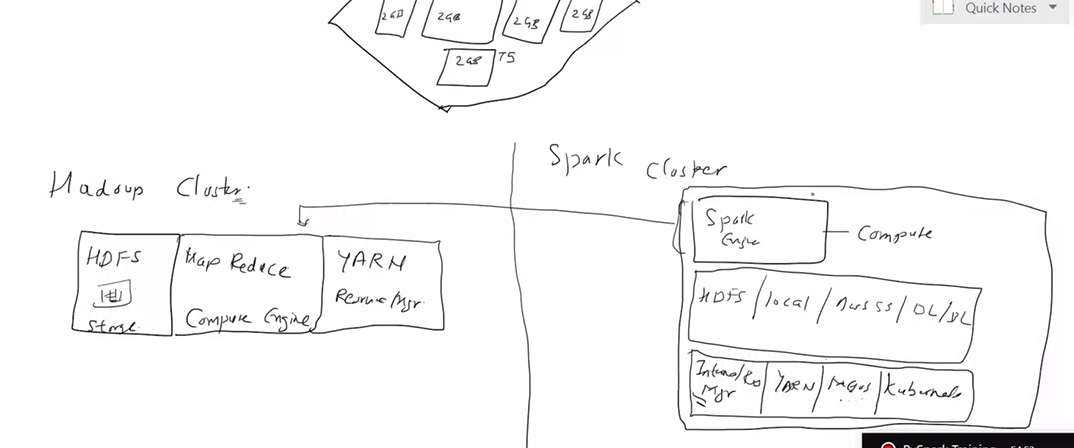






Haddop take data from hdfs

Where as spark doesn’t have its own storage system so it can use anything like hdfs,local system, s3 bucket



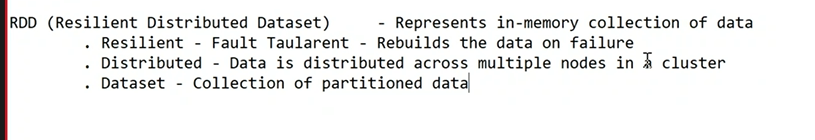
Spark has its own resourse manager: internal resourse manager by default or we can also provide like yarn, Kubernetes,

Why spark is faster:

in haddop eco system it loads data into memory then process data, once process then they send to storage again

where as in spark -in memory processing – data loaded into memory(temporary), and processed and stored in memory only , once all transformation applied it stores data to storage

rdd:



Rdd in spark- immutable

But rdd offers operations

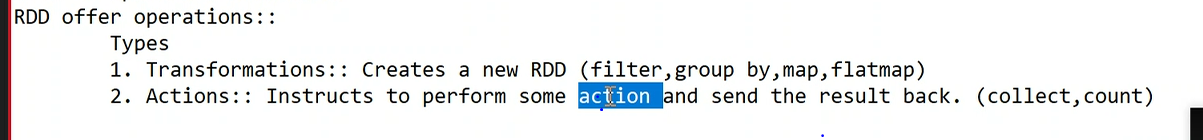
There are 2 types of operations available

1.trasformations : when we apply transformation it wont change rdd , new rdd will be created with a result of transformation

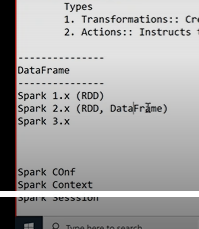


Filter -even number, employees who are manager

2. type : actions



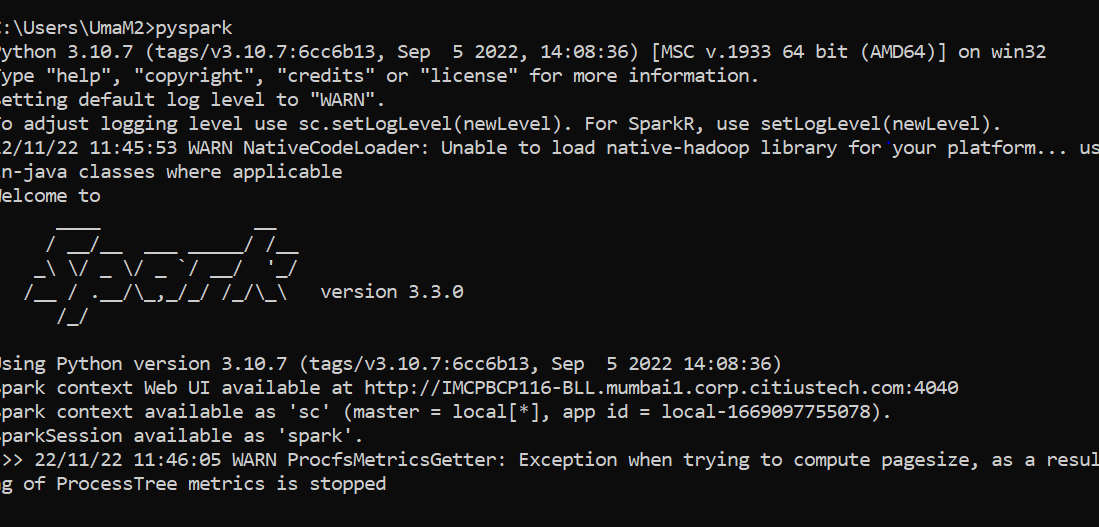
Actions gives results it wont create new rdd\ they apply some computations on data and return result back



sparkContext :

available as “sc”

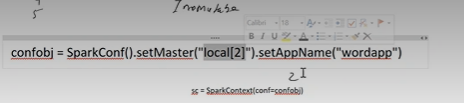
master=local(\*)

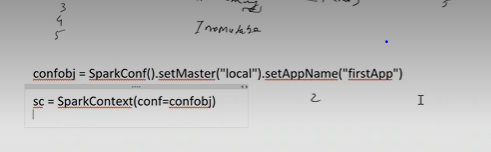


Web ui where u can see spark context

sparkConf

sparkConf is class: we need to create obj





Rdd will be created from sparkContext

