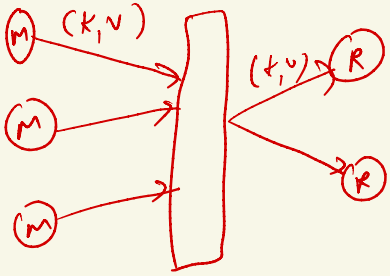
Hadoop vs Spark

Drawbacks of Hadoop

- It was made for Batch Processing
- Slow processing because not In memory Computation



Not In Memory

- Not fault tolerand In terms of Computation
- No mechanism for Caching, persistance etc.

Hadoop → HDFS
Hadoop → Map-Reduce
Hadoop → YARN

Apache Spark

- It is an distributed Computation frame-work.
- It doesn't have any seperate storage or file system.

Features of Spark

① **Speed**: Spark is 100 times faster than Hadoop

② **Powerful Caching**: Capabilities to persist data in memory

③ **Deployment**: More other resource mangers can also work with spark.

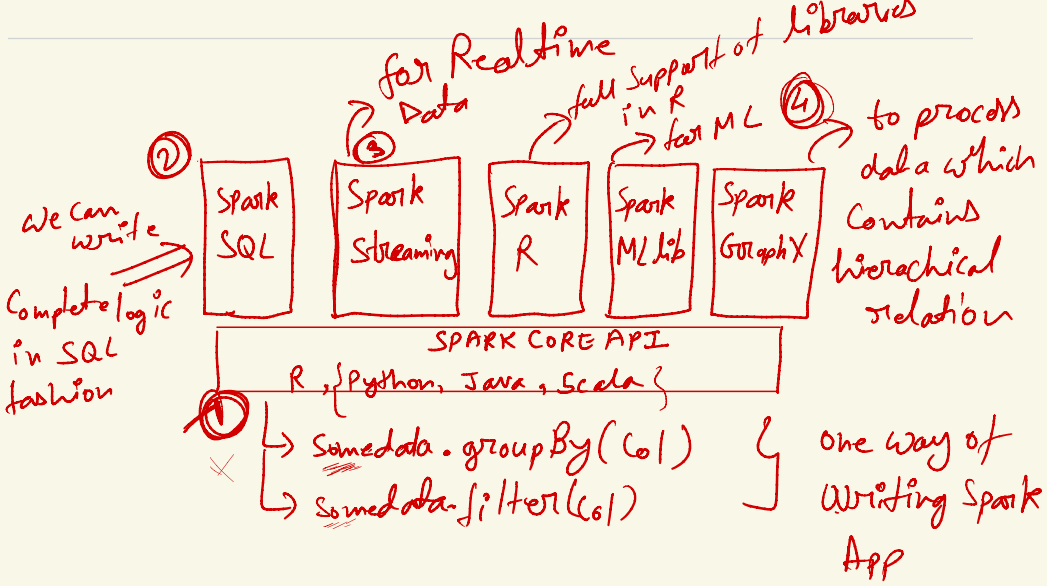
- → YARN (mostly used)
- → Mesos
- → kubernetes
- → Spark own cluster mangor

④ Good fit for Batch & Real time processing

⑤ **Polyglot** → Spark provides high lave APIs or libraries for different languages like

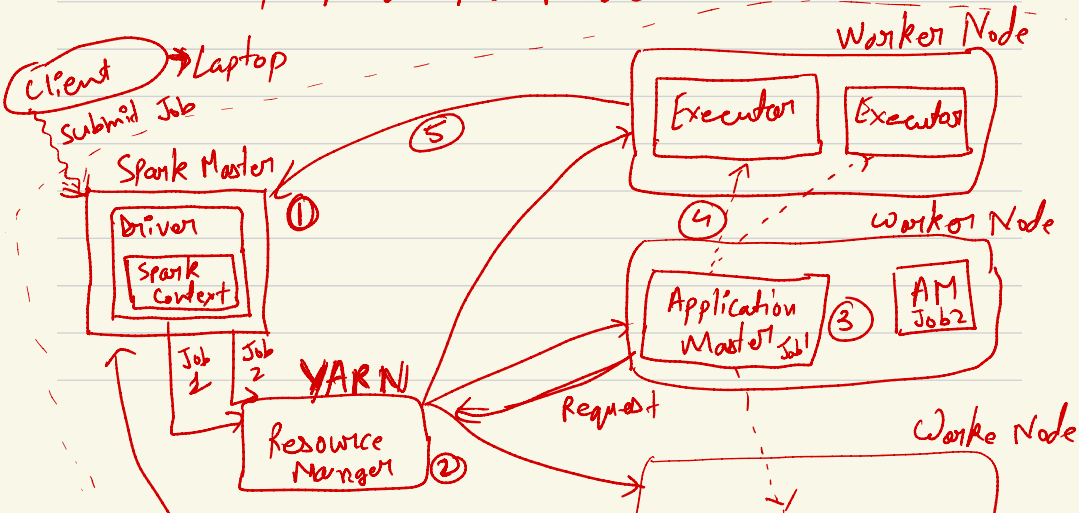
- Python (PySpark)
- Java
- Scala
- R

Apache Spark Ecosystem



Spark Architecture

- Spark follows Master-Slave Arch
- ① Spark Master
- ② Spark Worker Node

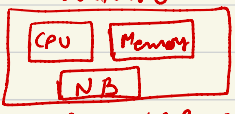


① Driver Programs
- It acts like a main method which will create the spark Context
- The first method which will be created

② Spark Context or Spark Session
- It is the entry point to spark Core of spark Application
- It will connect our spark application with execution environment or cluster
- It creates job execution graph, manages partitions, scheduling etc.

Before spark 2.00 Spark Context was used seperately
After Spark 2.0.0 + Spark Session does all the thing

③ Resource Manager
- It will create one container and start Application Master service in it
- Application Master will request to Resource Manager for Complete resource allocation or to start the required executors.
  - Virtual entity which will have CPU, memory, NB



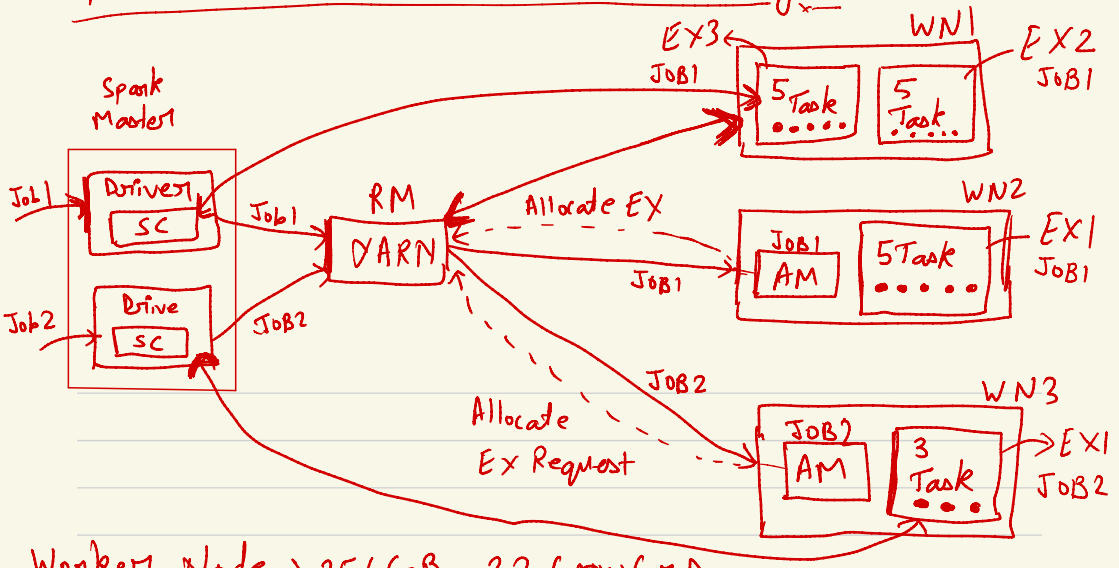
2CPU, 4GB, 3KB/Sec

as Executors does the actual computation on partitioned data

RM will Create required Requirednumber of executors for processing.

Executors will start Interacting with Driver Program to send the updates for Job progress and others metadata info.

Spark With YARN as Resource Manager



Worker Node → 256 GB, 32 CPU Cores

What we need for JOB1 → 3 Executors, each with 5 CPU Cores & 32 GB Memory

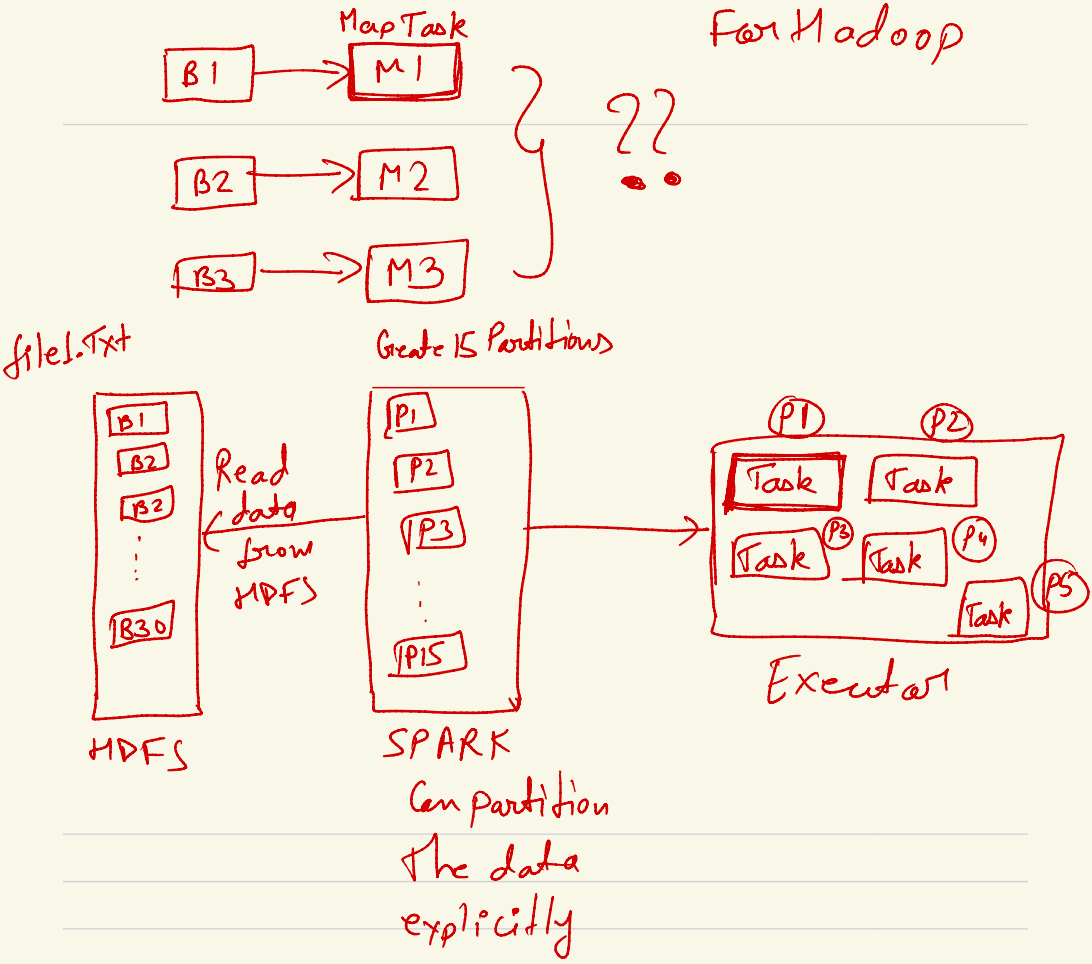
What we need for JOB2 → 1 Executor, each with 3 CPU Cores 16 GB Memory

Q.) Parameter in spark-Submit Command.

i) num of executors

ii) num of CPU cores in each executor

iii) num of memory for each executor



Task → ① Smallest unit of execution is known as Task.

② Each task will process one data partition.

③ One task will be executing on one CPU Core

④ One Executor Can run multiple tasks and equal to number of CPU cores