

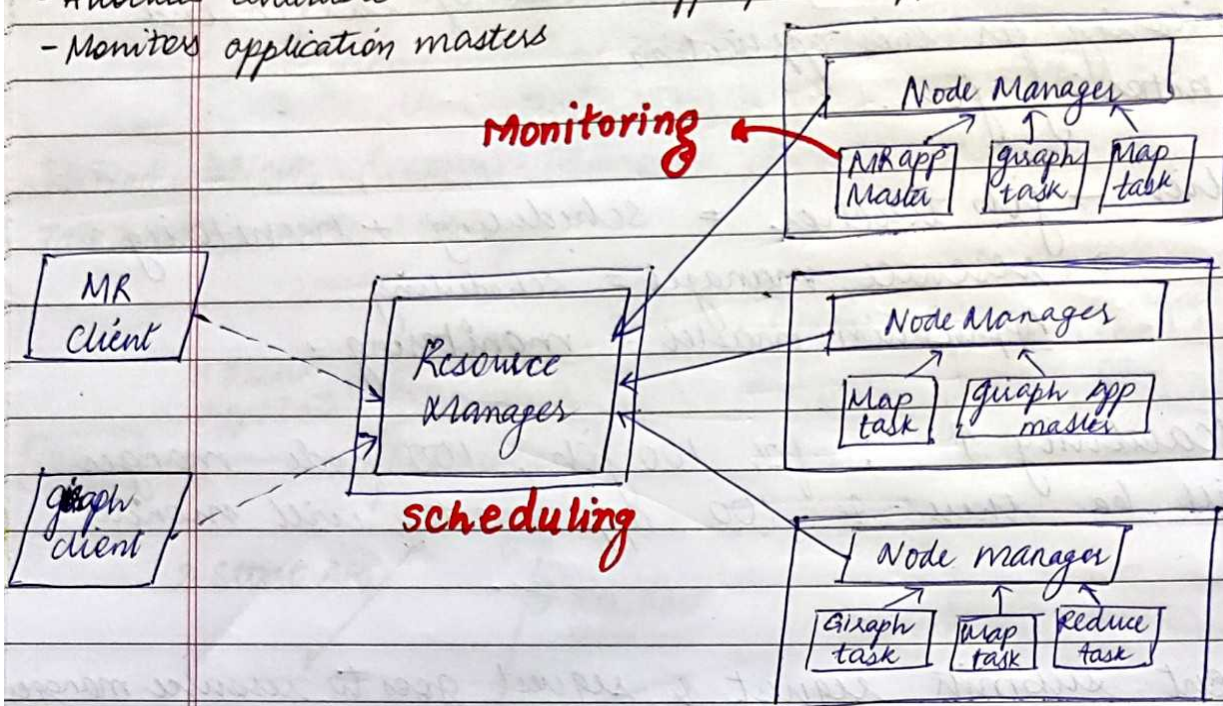
YARN Architecture:

classmate

Date _____
Page _____

Resource Manager (RM)

- Keeps track of live Node Managers & available resources
- Allocates available resources to appropriate applications and tasks
- Monitors application masters



Client

- Can submit any type of application supported by YARN.

Node Managers

- Provides computational resources in form of containers
- Managers processes running in containers

Application Master (AM)

- coordinates the execution of all tasks within its application
- Asks for appropriate resource containers to run tasks

Containers

- can run different types of tasks (also Application Masters)
- has different sizes eg. RAM, CPU.

MR2 - with introduction to YARN -
limitations are handled:

1. Scalability - scheduling → resource manager
- monitoring → App master

2. however in MR2 version we can have other job also, apart from just map reduce. eg: graph, spark, tex
3. however in MR2 there are no longer fixed amount of map & reduce slots.

With the concept of logical containers coming in, the resource allocation is much more dynamic & we can request for any amount of CPU or memory.

with this, cluster utilization is improved as the resources are not wasted

Uberization - /uber mode:

(request comes) → Resource Manager creates a container & runs application master in it. AM negotiates for resources. But when job is very small, AM can run the job in the container itself; no extra resources are required. Job can be run in resources which are already existing in container.

yarn session - 2

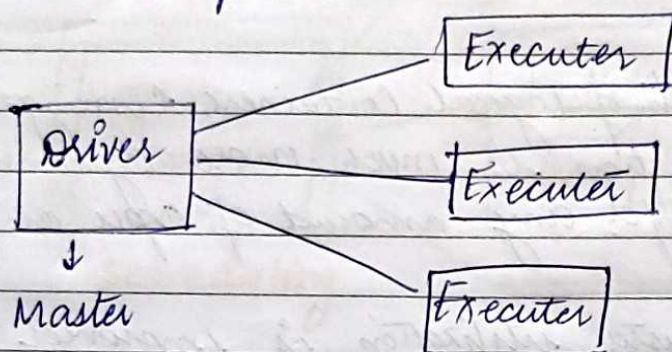
spark architecture on yarn -

1. How to execute spark programs on spark cluster -
 - interactive mode : spark-shell / pyspark / notebooks
 - submitting a job : spark-submit (bundle code & execute)
 - ↓
 - batch job ; used in prodⁿ

submit job using spark-submit utility.

2. How does spark execute our programs on the cluster
 Ans - Master / Slave Architecture

: Each application has driver which is the master process
 Each application has bunch of executors which are the slave process

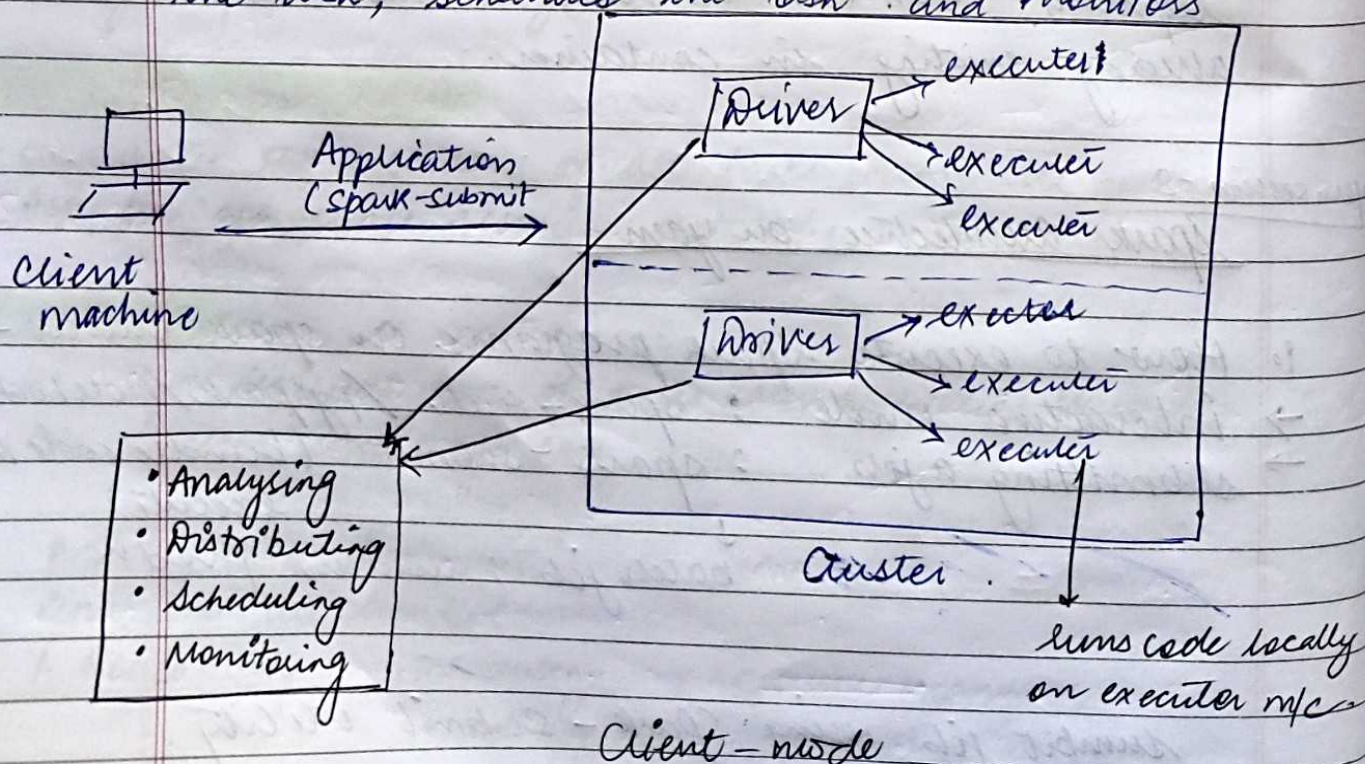


Slaves .

Executor - jvm process which has cpu & memory & can execute our jobs / code locally on jvm (executor machine)

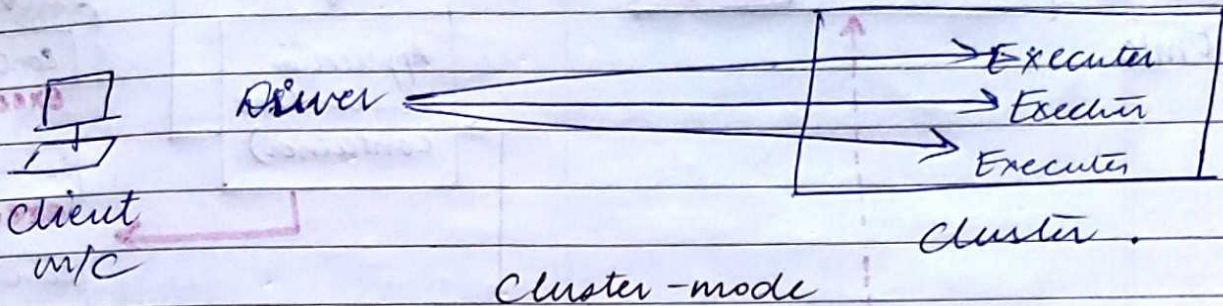
monitoring
scheduling

Driver - is responsible for analysing the work, divides the work in many tasks & distributes the task, schedules the task and monitors



3. who executes where?

- * Executors ~~cluster~~ ^{launched} always run on the cluster machines (worker nodes). However, for the driver we have flexibility to launch it on client machine or cluster machine.



Client mode) Whenever the driver runs on client m/c

Cluster mode) Whenever the driver runs on cluster or executor

Q- How Spark executes our prog on cluster?

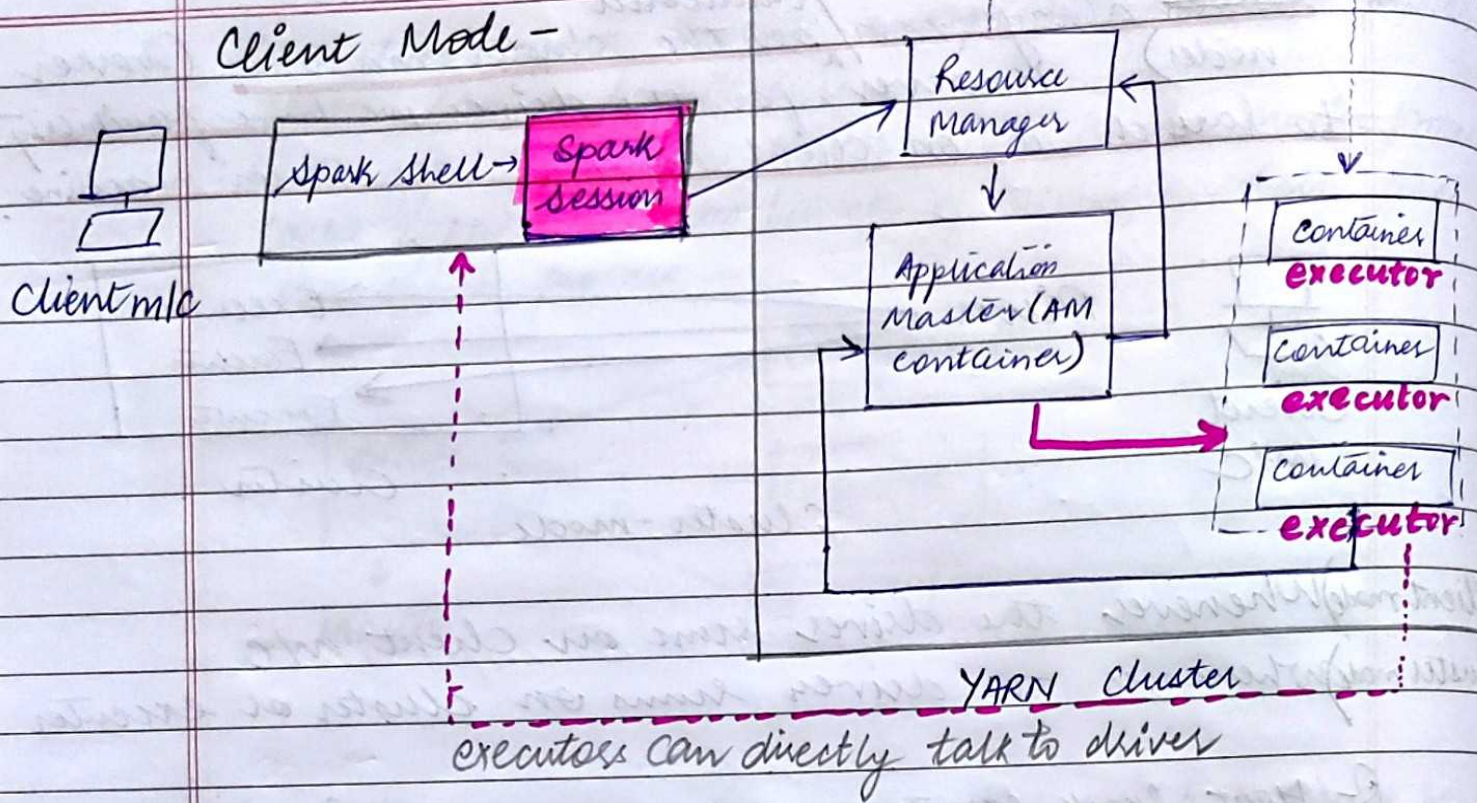
- Spark creates one driver & bunch of executors for each application
- Spark offers 2 different modes for application -
 - Client mode - driver on client m/c & executors on cluster
 - Cluster mode - Driver and executors on cluster.

The preferred approach for production is the cluster mode. Client mode is not preferred bcz if the client m/c goes down or is shut down, then the driver stops.

4. who controls the cluster & how spark gets the driver & executor?

Cluster Managers -

- YARN
- Mesos
- Kubernetes
- Spark standalone



Spark session - It is like a datastructure where driver maintains all the information including the executor location & status.

- This is the entry point for any spark application.

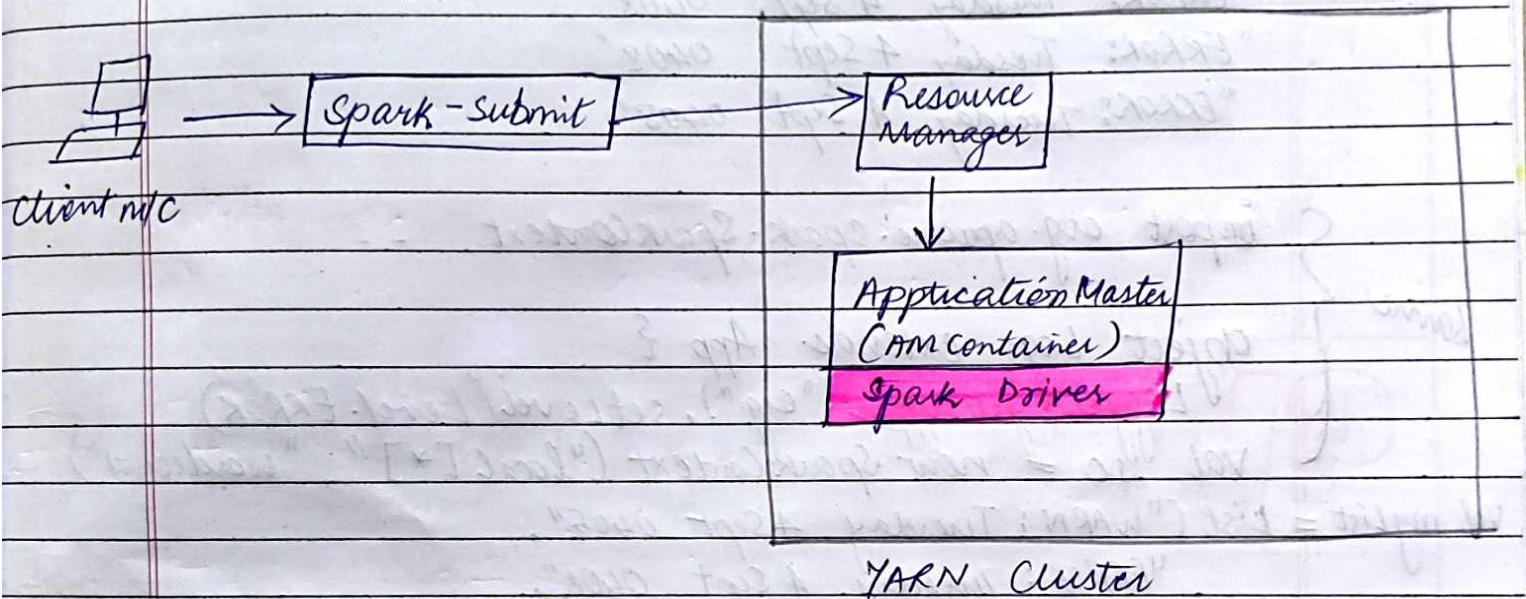
Application master launches executors in these containers. Executors can now directly talk to driver → running locally on client m/c. (Executor - driver direct connect)

Spark on yarn architecture in client mode -

1. When we launch spark-shell, automatically spark session is created
2. as soon as spark session is created request goes to Yarn resource manager
3. Yarn RM will create a container on one of the node managers & will launch an Application master for this spark application
4. This application master will negotiate for resources from the Yarn Resource manager in the ~~container~~ form of containers
5. The Yarn RM will create containers on the node managers

6. Now the application master will launch the executors in these containers
7. Now the drivers & executors can communicate directly without the involvement of containers.

spark on yarn architecture in cluster mode -



Only diff here is that Spark Driver runs on application master & it negotiates for resources.

* Even if client goes away, driver is still running in cluster, it is not impacted & thus preferred in production.

Local Mode -

