



Yarn Document

YARN - Yet Another Resource Negotiator

1. Storage perspective - HDFS

Name Node - Master (Holds the metadata)

Data Node - Slave (the actual data in terms of blocks)

2. Processing perspective in Hadoop 1 / MR 1

In hadoop 1, the job execution was controlled by 2 processes.

master - Job tracker

Slave - task tracker



Job Tracker

used to do a lot of work in hadoop 1.

scheduling + monitoring

scheduling - deciding which job to execute first based on scheduling algo, priority of jobs, getting to know the available resources, and providing the resources to jobs.

monitoring - tracking the progress of job, if a task fails rerun the task, if the task is slow then based on speculative execution start on other machine.

Task Tracker

This task tracker tracks the tasks on each data node and informs the job tracker.

The limitations or drawbacks of MR1

1. Scalability - it was observed that when the cluster size goes beyond 4k Datanodes (yahoo and facebook) then the job tracker used to become a bottleneck.

2. Resource Utilization - in MR1

there used to be a fixed number of map and reduce slots

100 map slots and 50 reduce slots

you want to execute a map reduce job which required 150 mappers.

100 mappers at a time

50 mappers will run later

3. only map reduce jobs were supported

3 pain points in MR1

1. scalability issues with large cluster
2. less resource utilization - under utilizing the cluster resources
3. only restricted to MR jobs

YARN (yet another resource negotiator)

Resource Manager - Master

Node Manager - Slave

Application Master

major bottleneck was that Job tracker was doing a lot of work. (Scheduling + Monitoring)

Job tracker - 1

many task trackers

Monitoring aspect is taken away from your job tracker.

job tracker - (scheduling + monitoring)

Resource Manager - (scheduling)

Task Trackers

Node Manager

The resource manager creates a container on one of the task tracker

container means: memory + cpu

1 gb memory + 2 cores

inside this container the resource manager creates an application master.

This application master will take care of end to end monitoring for this application.

application master - (monitoring the job)

Job tracker - (scheduling + monitoring)

Resource manager (scheduling)

Application master (monitoring)

Application master then asks the resource manager for resources.

and it requests for requests for resources in the form of containers

how many containers - 2

2 gb + 1 core

Resource manager allocates the resources in the form of containers

and it will send the container id and the host name to the application master.

MR2 with the introduction of yarn

how limitations are handled

1. scalability -

scheduling - Resource manager

monitoring - App master

2. however in MR2 version we can have other jobs also apart from just map reduce.

Giraph, spark, tez ...

3. however in MR2 there are no longer fixed amount of map and reduce slots.

with the concept of logical containers coming in, the resource allocation is much more dynamic and we can request for any amount of cpu or memory.

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

=====

uberization

