Experiment No: 3

Title: Hadoop Yarn

Aim: Study and demonstration of Hadoop Yarn administration commands and user command.

Theory:

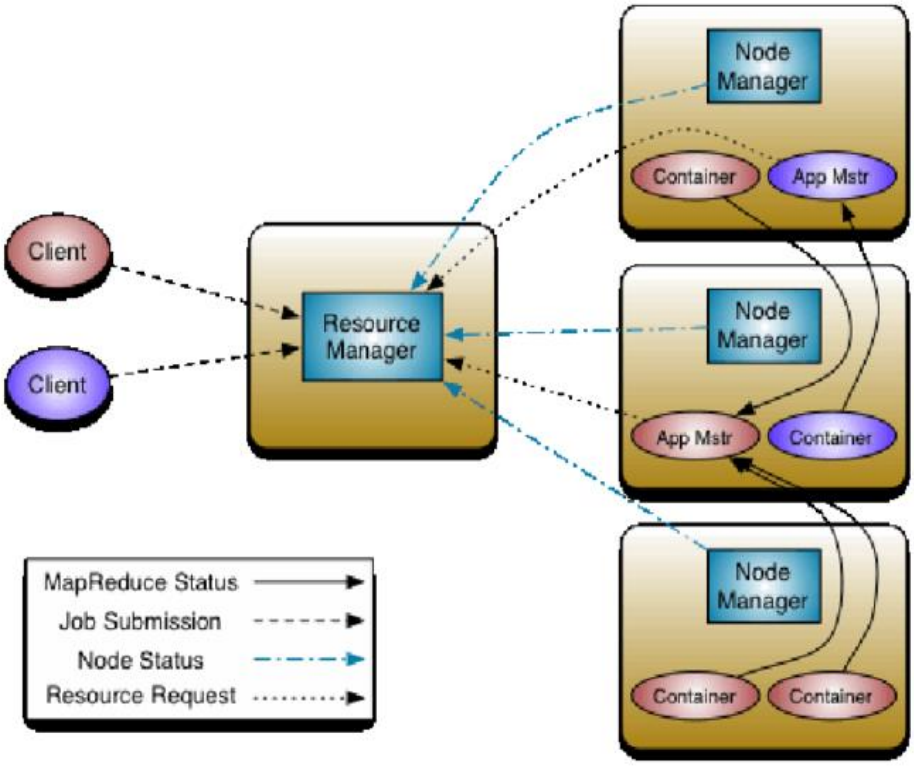
What is YARN?

YARN stands for Yet Another Resource Negotiator for Hadoop. Originally developed for Hadoop 2.0, YARN improved the MapReduce implementation and made it possible for Hadoop to handle a greater variety of data processing tasks. In simpler terms, YARN is the means by which clusters, resources, and jobs are managed in Hadoop. This means that the Hadoop can accommodate various data processing engines such as Interactive SQL, real-time streaming, and batch processing besides MapReduce thus expanding the opportunities for use of the platform**.**

YARN's three main components:

1. ResourceManager (RM): Acts as the master daemon, managing and allocating cluster resources. It comprises two main parts:
    - Scheduler: Allocates resources based on application requirements and policies.
    - ApplicationManager: Manages job submissions and coordinates with NodeManagers.
2. NodeManager: It is an application that is executed in the data nodes to coordinate the running of containers.
3. ApplicationMaster (AM): This is the component that is tasked with negotiating with the ResourceManager and either the NodeManager(s) to launch and monitor the tasks. It has a data computation framework provided by the ResourceManager and a per-node slave NodeManager. The ApplicationMaster is included in the application framework package.

Hadoop YARN Architecture



Role of YARN (Yet Another Resource Negotiator) in Hadoop

YARN (Yet Another Resource Negotiator) which is a core part of Hadoop that helps in boosting the architecture by effectively coordinating the resources and job scheduling . Here are the key roles of YARN in Hadoop:Here are the key roles of YARN in Hadoop:

Resource Management:

- Dynamic Allocation: YARN employs adequate strategies for distribution of resources the cluster CPU, memory, disk to a number of applications based on need.
- Centralized Management: It balances the resources in the Hadoop cluster and also avoids the conflict of resources within the control system.

Job Scheduling:

- Flexible Scheduling: In YARN, there are different Scheduling policies that are offered to the users such as FIFO, Capacity Scheduler, Fair Scheduler which helps in proper scheduling of the workload throughout the processing interface.
- Decoupling from MapReduce: Compared to using the concept of a job, YARN provides for more flexible use of resources by the system, as well as the separation of task scheduling from resource allocation, enabling Hadoop to work with frameworks other than MapReduce, including Apache Spark, Apache Flink, and Apache Tez.

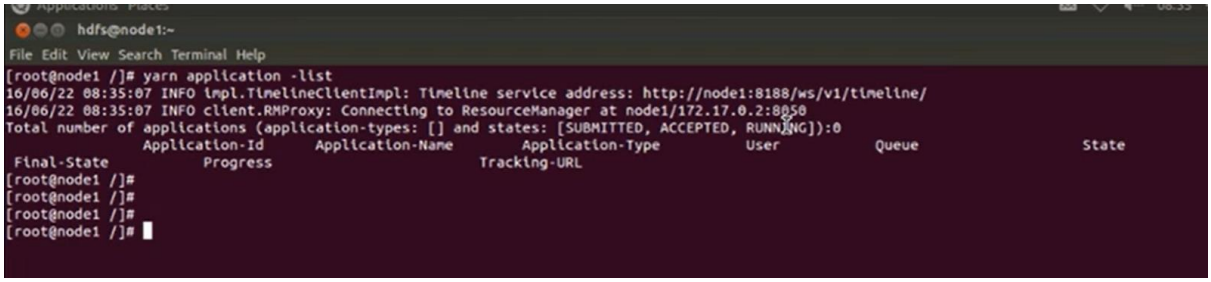
Scalability and Flexibility:

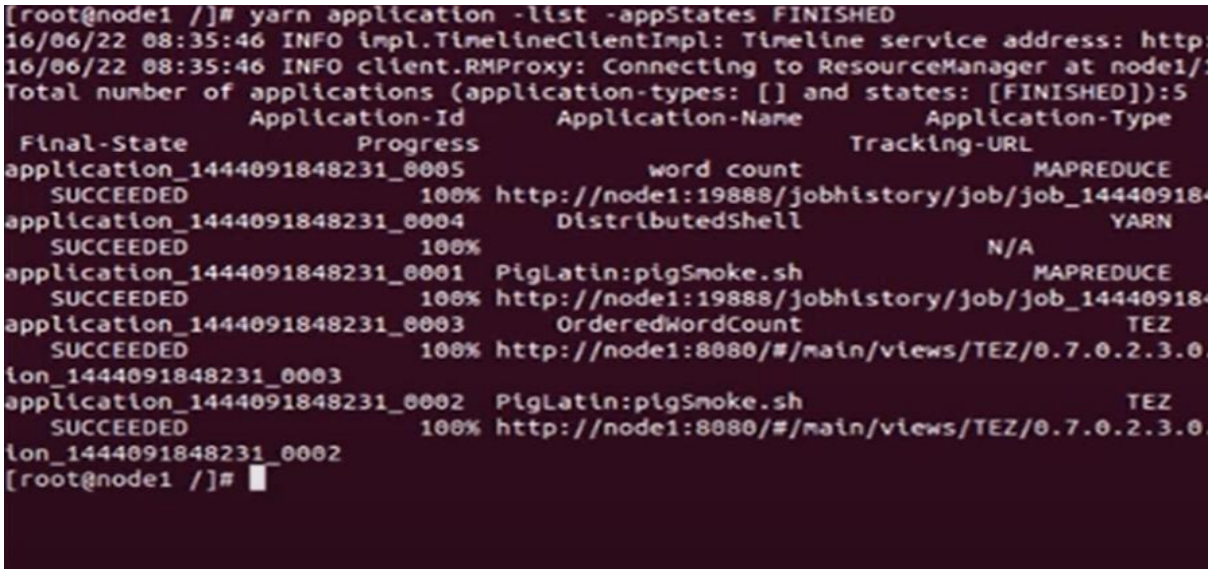
- Multi-application Support: YARN also authorizes multiple applications to run-on the same cluster and thus made Hadoop scalable and flexible.
- Efficient Cluster Utilization: With this setup, it allows the various processing engines to run thereby improving the general utilization of the cluster.

Improved Performance:

- Optimal Resource Usage: This frees up the rest of the nodes for resource provisioning to applications as needed without wasting resources hence improving the performance of the system.
- Enhanced Application Management: YARN has ApplicationMaster that is responsible for processing resource allocation and activity tracking and monitoring that apply to the particular application, which are helpful when it comes to the efficient creation and management of tasks relative to the end application.

Yarn Commands:





```
[root@node1 /]# yarn logs|more
Retrieve logs for completed YARN applications.
usage: yarn logs -applicationId <application ID> [OPTIONS]

general options are:
 -am <AM Containers>              Prints the AM Container logs for this
                                  application. Specify comma-separated
                                  value to get logs for related AM
                                  Container. For example, If we specify -am
                                  1,2, we will get the logs for the first
                                  AM Container as well as the second AM
                                  Container. To get logs for all AM
                                  Containers, use -am ALL. To get logs for
                                  the latest AM Container, use -am -1. By
                                  default, it will only print out syslog.
                                  Work with -logFiles to get other logs
 -appOwner <Application Owner>    AppOwner (assumed to be current user if
                                  not specified)
 -containerId <Container ID>      ContainerId. By default, it will only
                                  print syslog if the application is
                                  runing. Work with -logFiles to get other
                                  logs.
 -help                            Displays help for all commands.
 -logFiles <Log File Name>        Work with -am/-containerId and specify
                                  comma-separated value to get specified
                                  Container log files
 -nodeAddress <Node Address>      NodeAddress in the format nodename:port
```

```
[root@node1 /]# yarn daemonlog -getlevel node1:8088 rsourcemanager
Connecting to http://node1:8088/logLevel?log=rsourcemanager
Submitted Log Name: rsourcemanager
Log Class: org.apache.commons.logging.impl.Log4JLogger
Effective level: INFO
```

```
[root@node1 /]# yarn rmadmin -getGroups hdfs
16/06/22 08:38:30 INFO client.RMProxy: Connecting to ResourceManager at node1/172.17.0.2:8141
hdfs : hadoop hdfs
```

```
[root@node1 /]# curl -X GET http://node1:8088/ws/v1/cluster/metrics
{"clusterMetrics":{"appsSubmitted":0,"appsCompleted":0,"appsPending":0,"appsRunning":0,"appsFailed":0,"appsKilled":0,"reservedMB":0,"available
MB":12288,"allocatedMB":0,"reservedVirtualCores":0,"availableVirtualCores":3,"allocatedVirtualCores":0,"containersAllocated":0,"containersRese
rved":0,"containersPending":0,"totalMB":12288,"totalVirtualCores":3,"totalNodes":1,"lostNodes":0,"unhealthyNodes":0,"decommissionedNodes":0,"r
ebootedNodes":0,"activeNodes":1}}[root@node1 /]#
[root@node1 /]#
```

```
root@node1 /]# curl -X GET http://node1:8088/ws/v1/cluster/scheduler
{"scheduler":{"schedulerInfo":{"type":"capacityScheduler","capacity":100.0,"usedCapacity":0.0,"maxCapacity":100.0,"queueName":"root","queues":
{"queue":[{"type":"capacitySchedulerLeafQueueInfo","capacity":100.0,"usedCapacity":0.0,"maxCapacity":100.0,"absoluteCapacity":100.0,"absoluteM
axCapacity":100.0,"absoluteUsedCapacity":0.0,"numApplications":0,"queueName":"default","state":"RUNNING","resourcesUsed":{"memory":0,"vCores":
0},"hideReservationQueues":false,"nodeLabels":["*"],"numActiveApplications":0,"numPendingApplications":0,"numContainers":0,"maxApplications":1
0000,"maxApplicationsPerUser":10000,"userLimit":100,"users":null,"userLimitFactor":1.0,"AMResourceLimit":{"memory":3072,"vCores":1},"usedAMRes
ource":{"memory":0,"vCores":0},"userAMResourceLimit":{"memory":3072,"vCores":1},"preemptionDisabled":true}]},"health":{"lastrun":1466599302775
,"operationsInfo":{"entry":{"key":"last-preemption","value":{"nodeId":"N/A","containerId":"N/A","queue":"N/A"}},"entry":{"key":"last-reservat
ion","value":{"nodeId":"N/A","containerId":"N/A","queue":"N/A"}},"entry":{"key":"last-allocation","value":{"nodeId":"N/A","containerId":"N/A",
"queue":"N/A"}},"entry":{"key":"last-release","value":{"nodeId":"N/A","containerId":"N/A","queue":"N/A"}}},"lastRunDetails":[{"operation":"rele
ases","count":0,"resources":{"memory":0,"vCores":0}},{"operation":"allocations","count":0,"resources":{"memory":0,"vCores":0}},{"operation":"r
eservations","count":0,"resources":{"memory":0,"vCores":0}}]}}}}[root@node1 /]#
```

```
root@node1 /]# curl -X GET http://node1:8088/ws/v1/cluster/apps
{"apps":{"app":[{"id":"application_1444091848231_0005","user":"ambari-qa","name":"word count","queue":"default","state":"FINISHED","finalStat
":"SUCCEEDED","progress":100.0,"trackingUI":"History","trackingUrl":"http://node1:8088/proxy/application_1444091848231_0005/","diagnostics":
null","clusterId":1466595992892,"applicationType":"MAPREDUCE","applicationTags":"","startedTime":1444092006441,"finishedTime":1444092023677,"
apsedTime":17236,"amContainerLogs":"http://node1:8042/node/containerlogs/container_1444091848231_0005_01_000001/ambari-qa","amHostHttpAddres
":"node1:8042","allocatedMB":-1,"allocatedVCores":-1,"runningContainers":-1,"memorySeconds":90413,"vcoreSeconds":24,"preemptedResourceMB":0,"
reemptedResourceVCores":0,"numNonAMContainerPreempted":0,"numAMContainerPreempted":0,"logAggregationStatus":"NOT_START"},{"id":"application_1
4091848231_0004","user":"ambari-qa","name":"DistributedShell","queue":"default","state":"FINISHED","finalStatus":"SUCCEEDED","progress":100.
"trackingUI":"History","trackingUrl":"http://node1:8088/proxy/application_1444091848231_0004/","diagnostics":"null","clusterId":146659599289
"applicationType":"YARN","applicationTags":"","startedTime":1444091982589,"finishedTime":1444091990261,"elapsedTime":7752,"amContainerLogs":
ttp://node1:8042/node/containerlogs/container_1444091848231_0004_01_000001/ambari-qa","amHostHttpAddress":"node1:8042","allocatedMB":-1,"all
atedVCores":-1,"runningContainers":-1,"memorySeconds":25991,"vcoreSeconds":8,"preemptedResourceMB":0,"preemptedResourceVCores":0,"numNonAMCo
ainerPreempted":0,"numAMContainerPreempted":0,"logAggregationStatus":"NOT_START"},{"id":"application_1444091848231_0001","user":"ambari-qa",
ame":"PigLatin:pigSmoke.sh","queue":"default","state":"FINISHED","finalStatus":"SUCCEEDED","progress":100.0,"trackingUI":"History","tracking
l":"http://node1:8088/proxy/application_1444091848231_0001/","diagnostics":"null","clusterId":1466595992892,"applicationType":"MAPREDUCE","a
licationTags":"","startedTime":1444091899690,"finishedTime":1444091918932,"elapsedTime":19242,"amContainerLogs":"http://node1:8042/node/cont
nerlogs/container_1444091848231_0001_01_000001/ambari-qa","amHostHttpAddress":"node1:8042","allocatedMB":-1,"allocatedVCores":-1,"runningCon
iners":-1,"memorySeconds":73288,"vcoreSeconds":23,"preemptedResourceMB":0,"preemptedResourceVCores":0,"numNonAMContainerPreempted":0,"numAMC
tainerPreempted":0,"logAggregationStatus":"NOT_START"},{"id":"application_1444091848231_0003","user":"ambari-qa","name":"OrderedWordCount","
eue":"default","state":"FINISHED","finalStatus":"SUCCEEDED","progress":100.0,"trackingUI":"History","trackingUrl":"http://node1:8088/proxy/a
lication_1444091848231_0003/","diagnostics":"null","clusterId":1466595992892,"applicationType":"TEZ","applicationTags":"","startedTime":1444
1962557,"finishedTime":1444091979751,"elapsedTime":17194,"amContainerLogs":"http://node1:8042/node/containerlogs/container_1444091848231_000
01_000001/ambari-qa","amHostHttpAddress":"node1:8042","allocatedMB":-1,"allocatedVCores":-1,"runningContainers":-1,"memorySeconds":83048,"vc
eSeconds":27,"preemptedResourceMB":0,"preemptedResourceVCores":0,"numNonAMContainerPreempted":0,"numAMContainerPreempted":0,"logAggregationS
tus":"NOT_START"},{"id":"application_1444091848231_0002","user":"ambari-qa","name":"PigLatin:pigSmoke.sh","queue":"default","state":"FINISHE
,"finalStatus":"SUCCEEDED","progress":100.0,"trackingUI":"History","trackingUrl":"http://node1:8088/proxy/application_1444091848231_0002/","
agnostics":"null","clusterId":1466595992892,"applicationType":"TEZ","applicationTags":"","startedTime":1444091939606,"finishedTime":14440919
684,"elapsedTime":19078,"amContainerLogs":"http://node1:8042/node/containerlogs/container_1444091848231_0002_01_000001/ambari-qa","amHostHtt
ddress":"node1:8042","allocatedMB":-1,"allocatedVCores":-1,"runningContainers":-1,"memorySeconds":89011,"vcoreSeconds":28,"preemptedResource
":0,"preemptedResourceVCores":0,"numNonAMContainerPreempted":0,"numAMContainerPreempted":0,"logAggregationStatus":"NOT_START"}]}}[root@node1
```

```
root@node1 /]# curl -X GET http://node1:8088/ws/v1/cluster/nodes
{"nodes":{"node":[{"rack":"/default-rack","state":"RUNNING","id":"node1:45454","nodeHostName":"node1","nodeHTTPAddress":"node1:8042","lastHeal
thUpdate":1466599338148,"version":"2.7.1.2.3.0.0-2557","healthReport":"","numContainers":0,"usedMemoryMB":0,"availMemoryMB":12288,"usedVirtual
Cores":0,"availableVirtualCores":3}]}}[root@node1 /]#
```