# YARN-HADOOP
# (Also called Mapreduce2)

## 2CEIT702 - Big Data Analytics

**Book:**

Big Data and Analytics by Seema Acharya, Subhashini Chellappan, Paperback

**Reference book:**

https://www.uvpce.ac.in/content/syllabus-ce

# Difference between YARN and MapReduce

| MapReduce | YARN |
| --- | --- |
| Part of Hadoop 1.x | Part of Hadoop 2.x |

# Difference between YARN and MapReduce

| MapReduce | YARN |
| --- | --- |
| Part of Hadoop 1.x | Part of Hadoop 2.x |
| Single process handle (Job Tracker): Resource-allocation, Job Scheduling & process of Map-Reduce Job | One process handle (YARN): Resource management, Job scheduling & Monitoring.<br>Other process handle (MapReduce): Data processing |

# Difference between YARN and MapReduce

| MapReduce | YARN |
| --- | --- |
| Part of Hadoop 1.x | Part of Hadoop 2.x |
| Single process handle (Job Tracker): Resource-allocation, Job Scheduling & process of Map-Reduce Job | One process handle (YARN): Resource management, Job scheduling & Monitoring.<br>Other process handle (MapReduce): Data processing |
| Single Point of Failure: When Map-reduce stops working then slave nodes will stop working automatically. | Execution Continue: It has 'Active name' node and 'standby name node'. When active node stop working, passive node starts working as active node. |

# Difference between YARN and MapReduce

| MapReduce | YARN |
|---|---|
| Part of Hadoop 1.x | Part of Hadoop 2.x |
| Single process handle (Job Tracker): Resource-allocation, Job Scheduling & process of Map-Reduce Job | One process handle (YARN): Resource management, Job scheduling & Monitoring.<br>Other process handle (MapReduce): Data processing |
| Single Point of Failure: When Map-reduce stops working then slave nodes will stop working automatically. | Execution Continue: It has 'Active name' node and 'standby name node'. When active node stop working, passive node starts working as active node. |
| MapReduce has single master and multiple slave architecture, If master goes down then entire slave will stop working (Single point of failure) | It has the concept of multiple master and slave, if one master goes down then another master will resume its process and continue the execution. |

# Difference between YARN and MapReduce

| MapReduce | YARN |
|---|---|
| Part of Hadoop 1.x | Part of Hadoop 2.x |
| Single process handle (Job Tracker): Resource-allocation, Job Scheduling & process of Map-Reduce Job | One process handle (YARN): Resource management, Job scheduling & Monitoring.<br>Other process handle (MapReduce): Data processing |
| Single Point of Failure: When Map-reduce stops working then slave nodes will stop working automatically. | Execution Continue: It has 'Active name' node and 'standby name node'. When active node stop working, passive node starts working as active node. |
| MapReduce has single master and multiple slave architecture, If master goes down then entire slave will stop working (Single point of failure) | It has the concept of multiple master and slave, if one master goes down then another master will resume its process and continue the execution. |
| You can run only MapReduce jobs with Hadoop. | You can run MapReduce jobs and also run different types of distributed applications. |

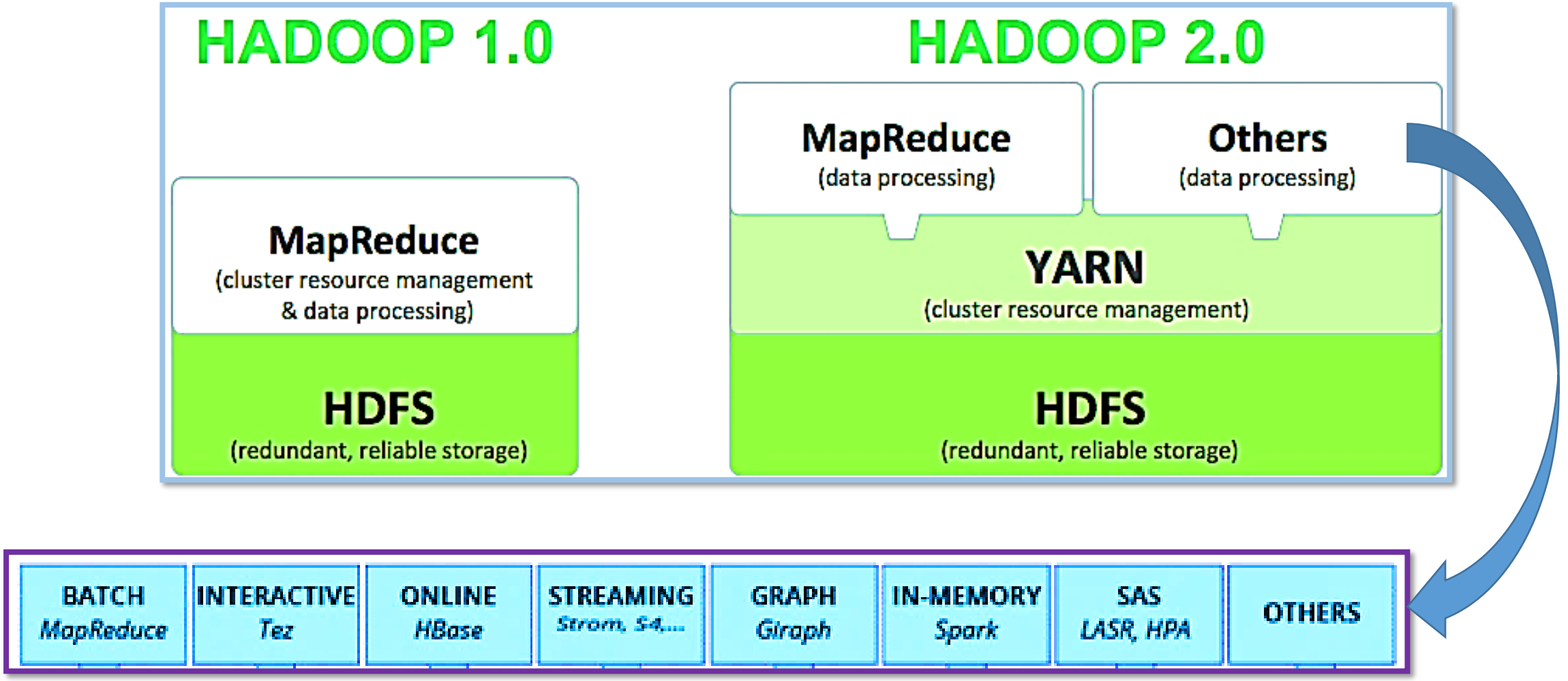
# Difference between YARN and MapReduce

| MapReduce | YARN |
|---|---|
| Part of Hadoop 1.x | Part of Hadoop 2.x |
| Single process handle (Job Tracker): Resource-allocation, Job Scheduling & process of Map-Reduce Job | One process handle (YARN): Resource management, Job scheduling & Monitoring.<br>Other process handle (MapReduce): Data processing |
| Single Point of Failure: When Map-reduce stops working then slave nodes will stop working automatically. | Execution Continue: It has 'Active name' node and 'standby name node'. When active node stop working, passive node starts working as active node. |
| MapReduce has single master and multiple slave architecture, If master goes down then entire slave will stop working (Single point of failure) | It has the concept of multiple master and slave, if one master goes down then another master will resume its process and continue the execution. |
| You can run only MapReduce jobs with Hadoop. | You can run MapReduce jobs and also run different types of distributed applications. |
| Facing issues: Delaying batch processing, Scalability | YARN solved these issues |

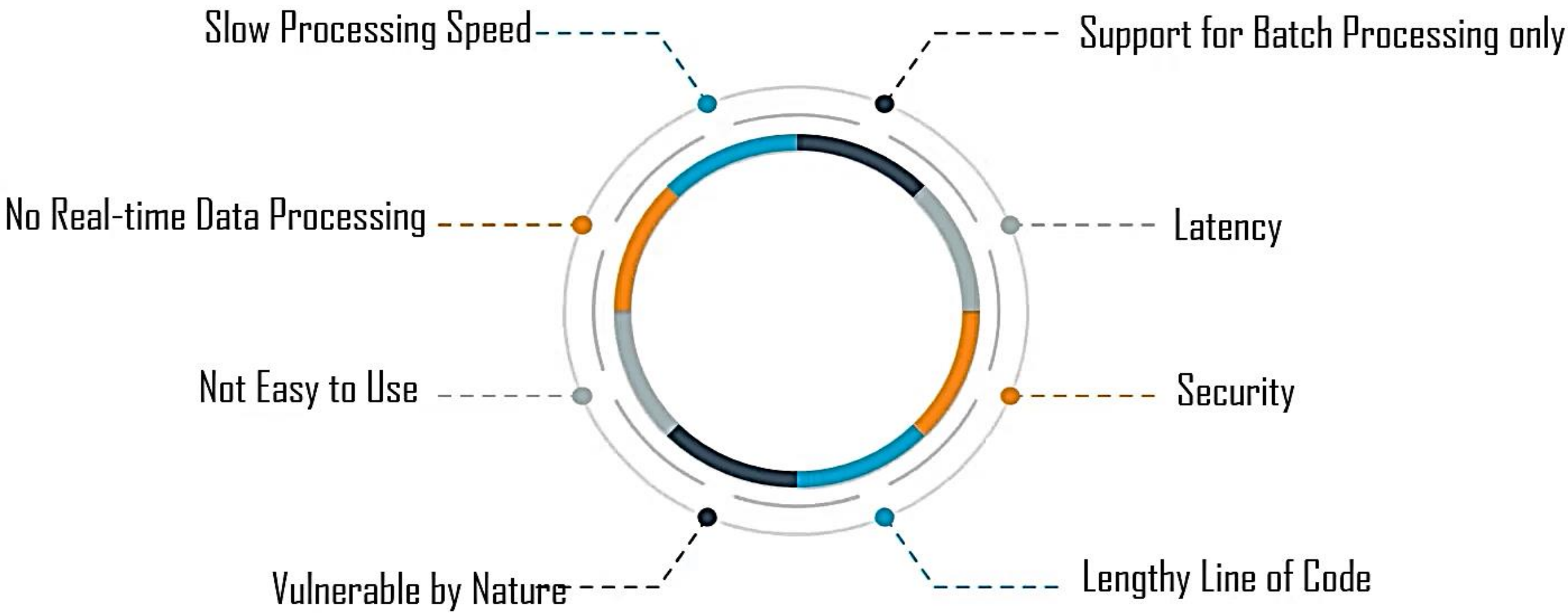
# Difference between YARN and MapReduce

| MapReduce | YARN |
|---|---|
| Part of Hadoop 1.x | Part of Hadoop 2.x |
| Single process handle (Job Tracker): Resource-allocation, Job Scheduling & process of Map-Reduce Job | One process handle (YARN): Resource management, Job scheduling & Monitoring.<br>Other process handle (MapReduce): Data processing |
| Single Point of Failure: When Map-reduce stops working then slave nodes will stop working automatically. | Execution Continue: It has 'Active name' node and 'standby name node'. When active node stop working, passive node starts working as active node. |
| MapReduce has single master and multiple slave architecture, If master goes down then entire slave will stop working (Single point of failure) | It has the concept of multiple master and slave, if one master goes down then another master will resume its process and continue the execution. |
| You can run only MapReduce jobs with Hadoop. | You can run MapReduce jobs and also run different types of distributed applications. |
| Facing issues: Delaying batch processing, Scalability | YARN solved these issues |
| Fixed slot of map & reduce tasks. so while map is running you can't use reduce slots for map tasks because of that slots go waste. | There is a concept of container. Any task can be run in it. |

# Difference between YARN and MapReduce

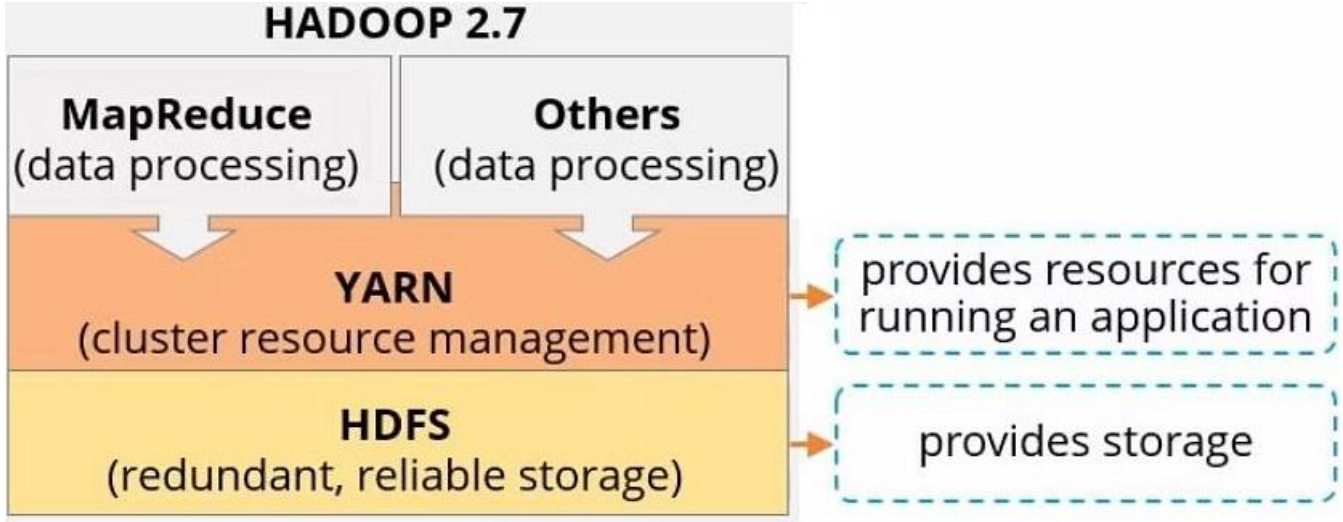


# Limitation of Apache Hadoop 1.x



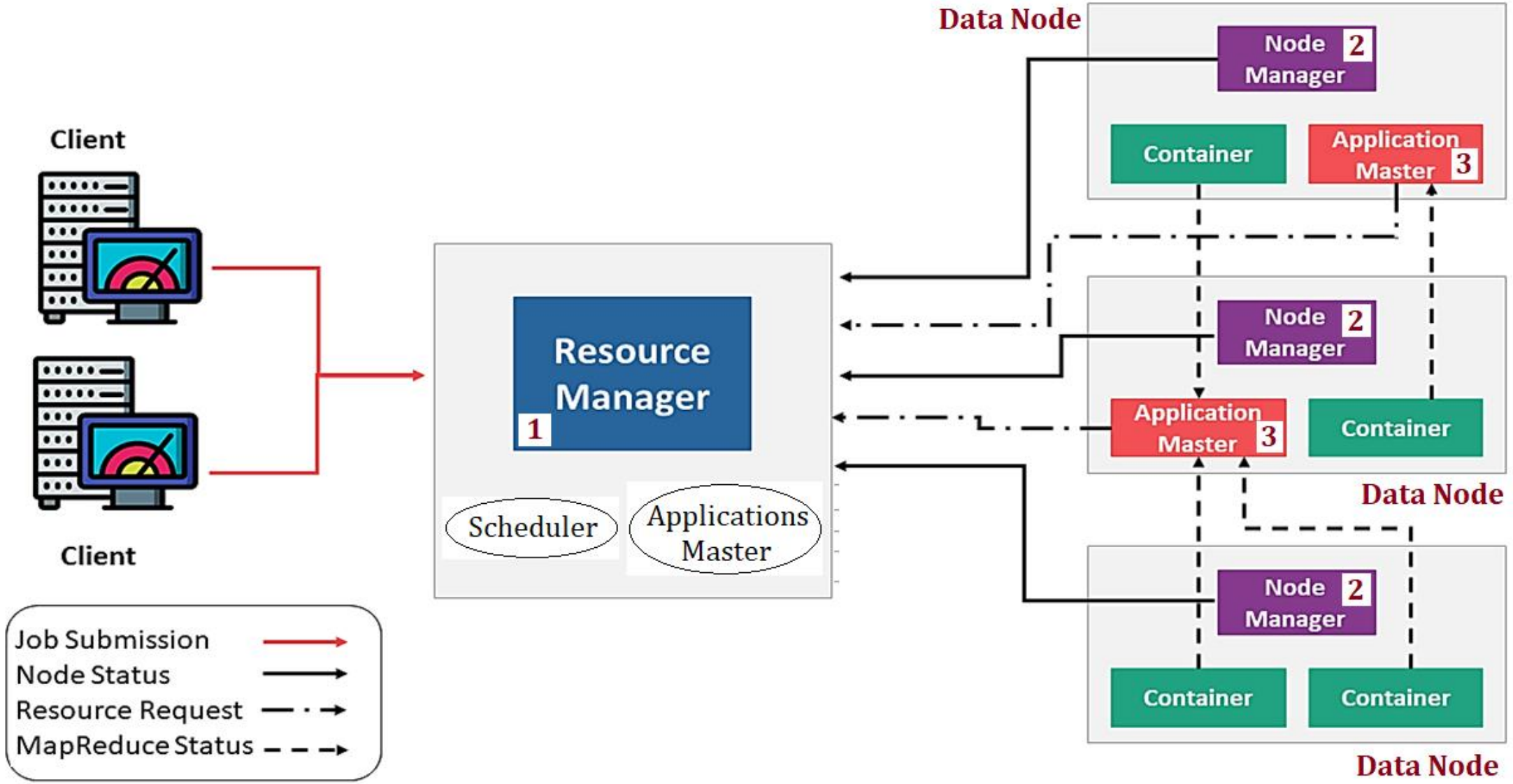
# YARN Infrastructure

- YARN infrastructure and HDFS are completely independent. YARN provides computational resources (CPUs or memory) needed for application executions while HDFS provides storage.
- The MapReduce framework is only one of the many possible frameworks that run on YARN.
- The fundamental idea of MapReduce version-2 is to split the two major functionalities of **Resource management, Job scheduling & Monitoring** and **Processing unit** into separate daemons.



# YARN Infrastructure

- The three important elements of the YARN architecture are:



# YARN Infrastructure (Resource Manager)

- The Resource Manager (RM), which is usually one per cluster, is the master server.
- Responsible for managing several other applications and allocation of the available resources such as CPU and memory to applications. It is used for job scheduling.
- RM knows the location of the DataNode and how many resources they have. (Rack Awareness).
- It keeps all resources in the cluster in use and hence enhances system utilization.
- Resource Manager has two components:

  - **Scheduler:** Scheduler is distribute resources to the running applications only. Do not tracking and monitoring of applications.

  - **Application Manager:** It is start the application master, manages applications in the cluster and monitor it.

# YARN Infrastructure (Node Manager)

- The Node Manager works based on the instructions given by the Resource Manager.
- Node Manager is the slave daemon of YARN.
- Node Manager monitors the container's resource usage and reporting to the RM.
- The health of the node on which YARN is running is tracked by the Node Manager.
- It takes care of each node in the cluster while managing the workflow, along with user jobs on a particular node.
- It keeps the data in the Resource Manager updated
- Node Manager can also destroy or kill the container if it gets an order from the Resource Manager to do so.
- Each machine has a node manager. The node managers create containers to execute the programs.

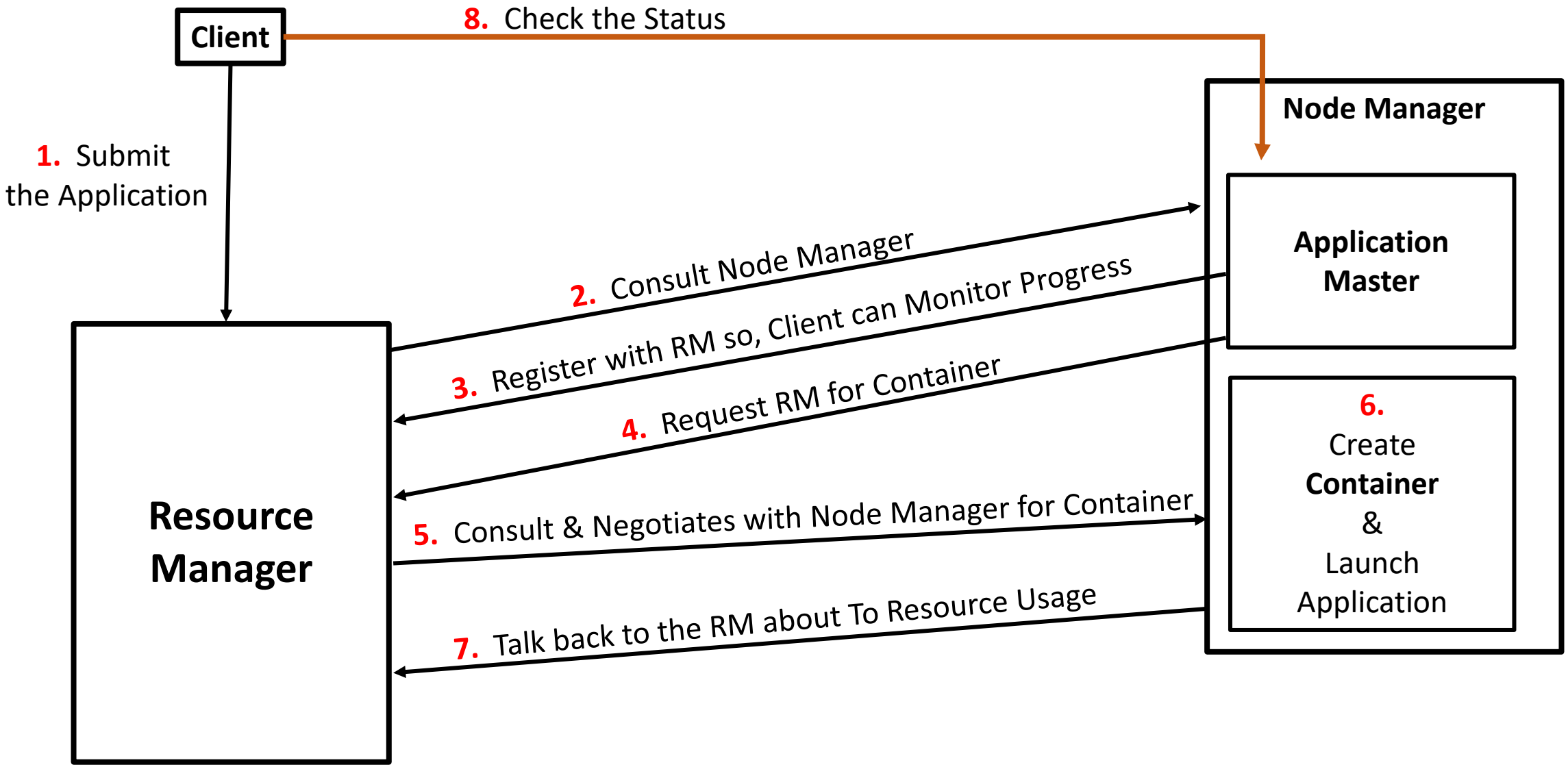
# YARN Infrastructure (Application Masters)

- Every job submitted to the framework is an application, and every application has a specific Application Master associated with it.
- It coordinates the execution of the application in the cluster, along with managing the faults.
- It negotiates resources from the Resource Manager.
- It works with the Node Manager for executing and monitoring other components' tasks.
- At regular intervals, heartbeats are sent to the Resource Manager for checking its health, along with updating records according to its resource demands.
- Application Masters are the programs per application that are executed inside the containers. Examples of Application Masters are Mapreduce-AM and Spark-AM.
- Each Application Master requests resources from the Resource Manager and then works with containers provided by Node Managers.

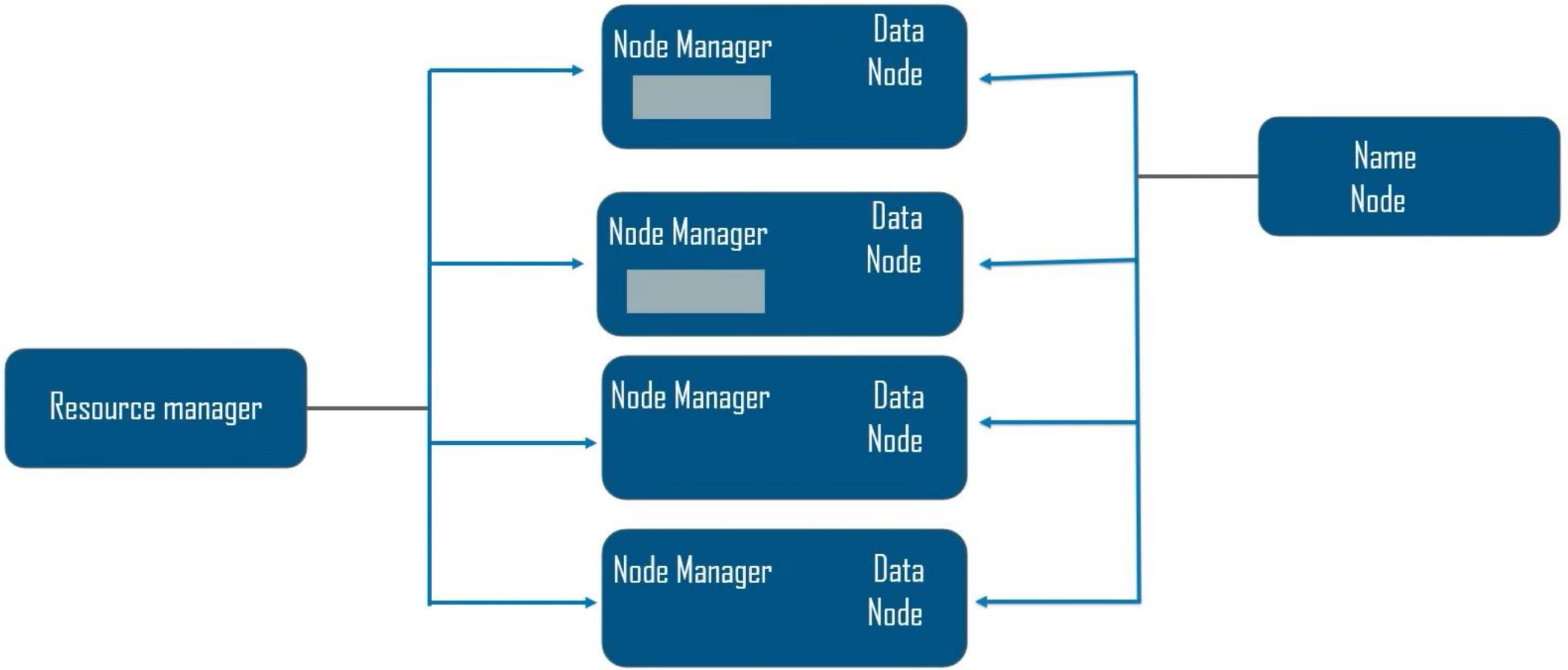
# YARN Infrastructure (Container)

- Physical unit with fixed resources of RAM, CPU cores, etc. on a single node in cluster.
- It grants rights to an application to use a specific amount of resources (memory, CPU, etc.) on a specific host.
- Each container will take care of the execution of a single entity (single unit of work) like the MapReduce. In MapReduce, a container can be said as a map or a reduce task.
- In Hadoop 1.x a slot is allocated by the JobTracker to run each MapReduce task. Then the TaskTracker generate a separate JVM for each task(unless JVM reuse is not enabled).
- In Hadoop 2.x, Container is a place where a unit of work is executed. For instance, each MapReduce task(not the entire job) runs in one container.
- An application/job will run on one or more containers.
- Each node in a Hadoop cluster can run several containers.

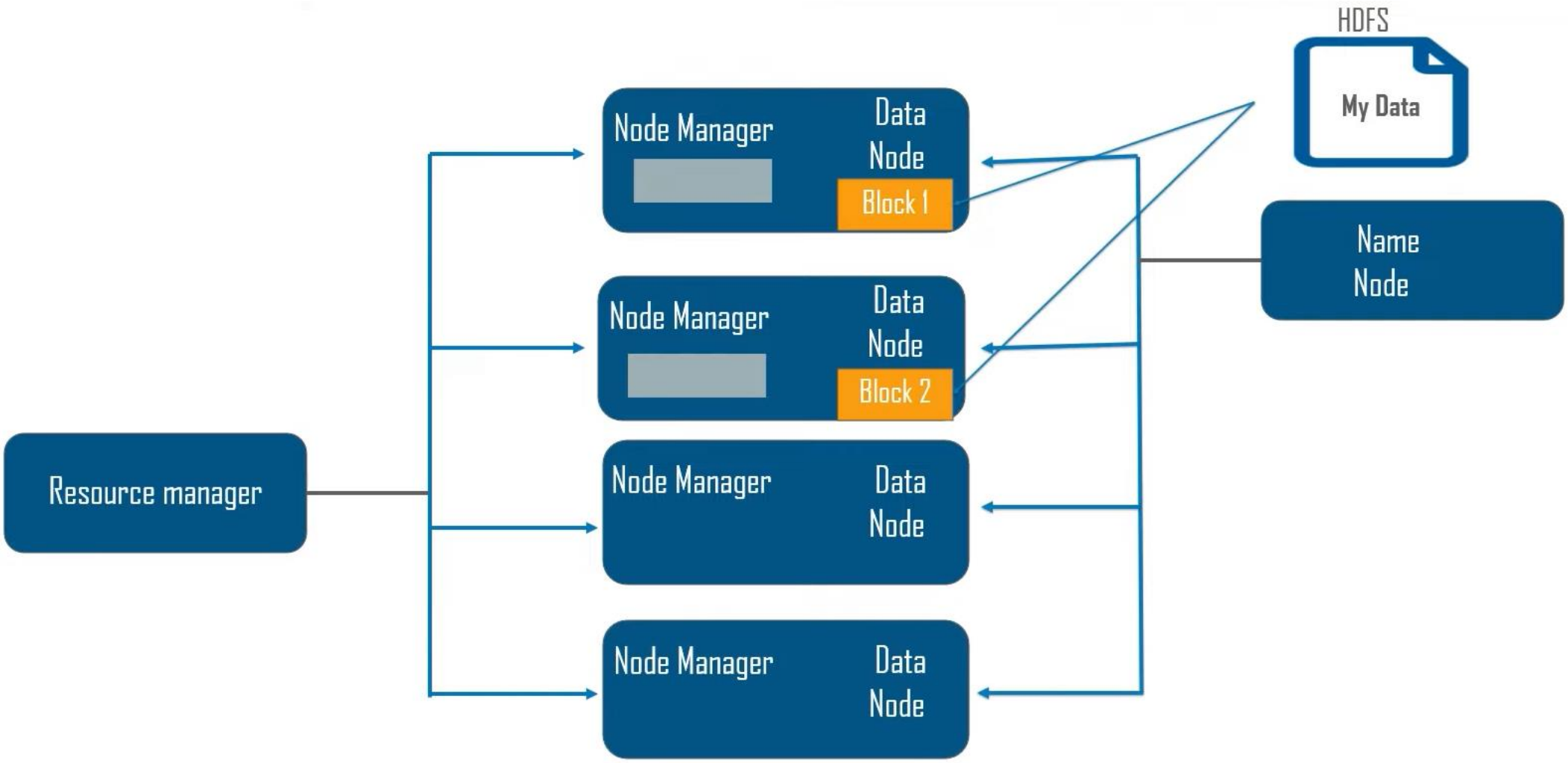
# Working steps of YARN



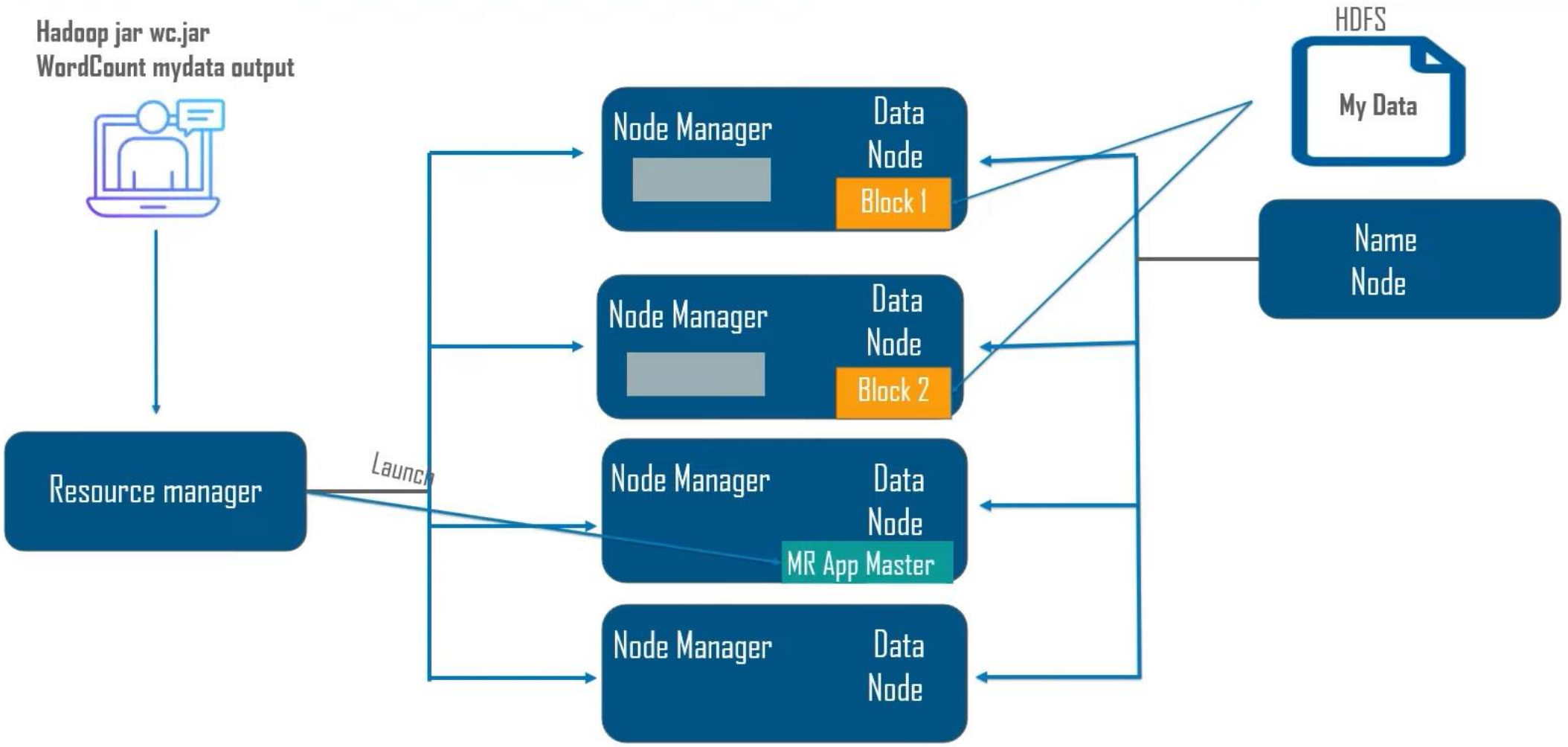
# Running Word Count Application in Map-Reduce-2



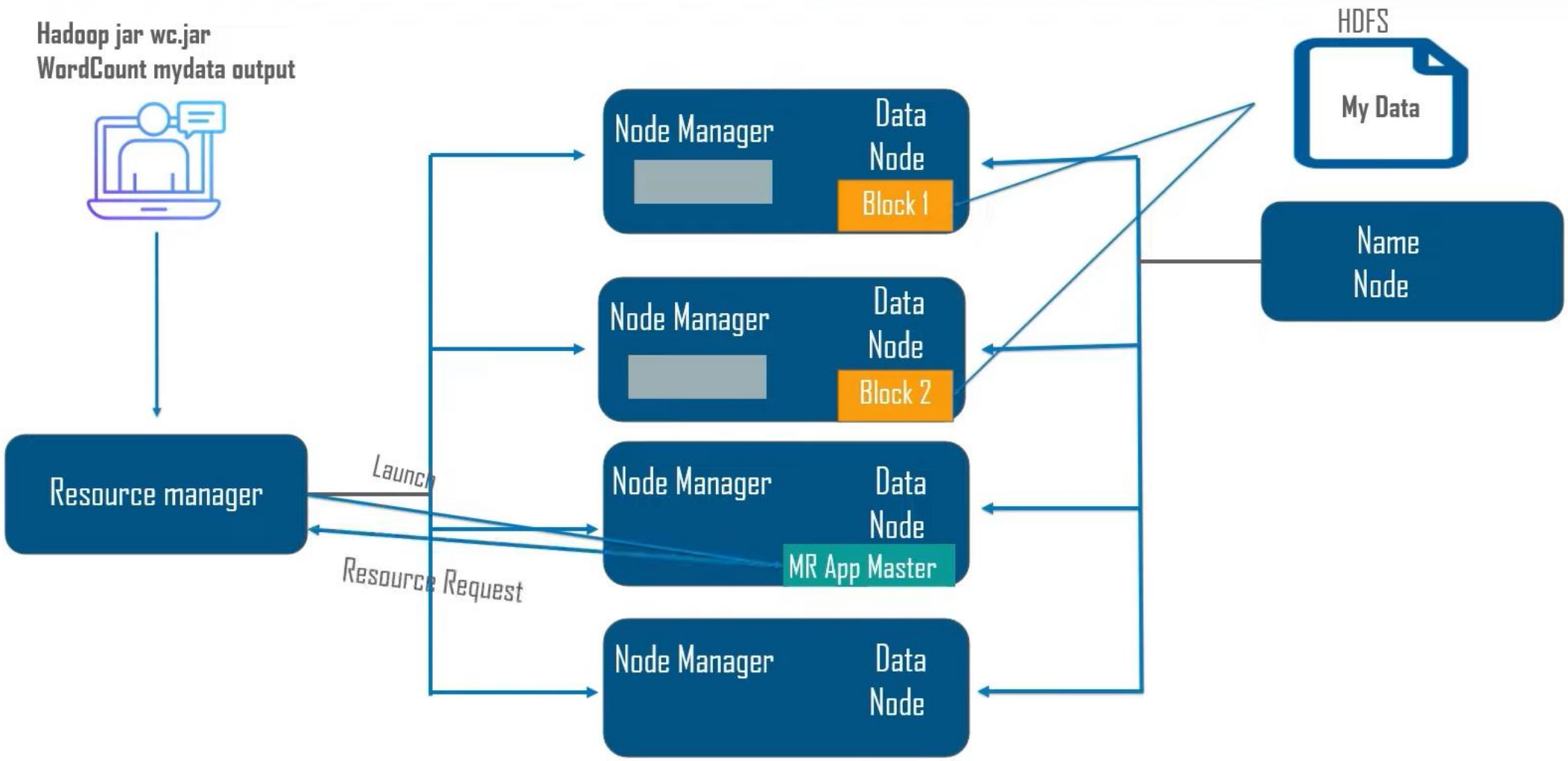
# Running Word Count Application in Map-Reduce-2



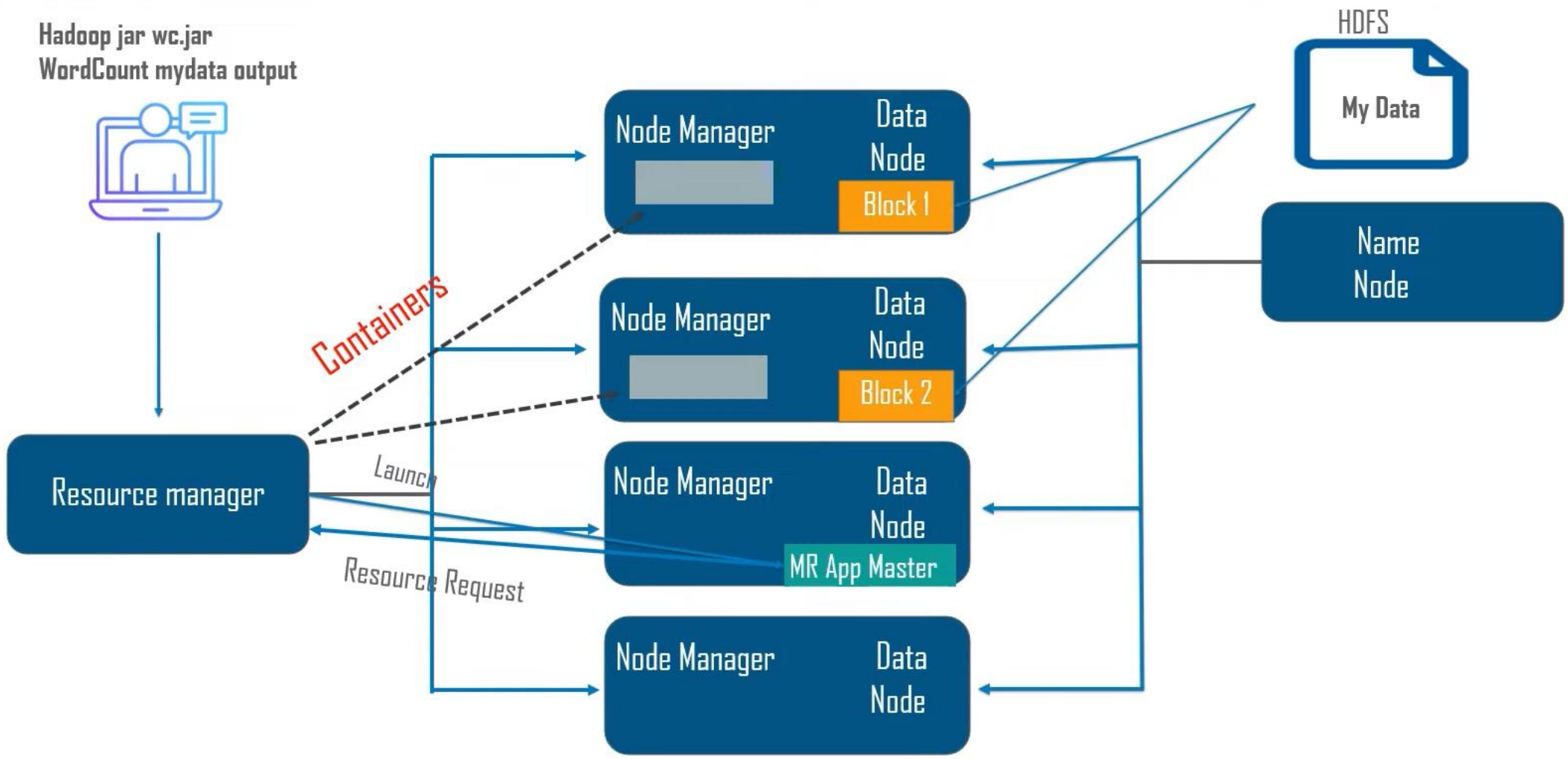
# Running Word Count Application in Map-Reduce-2



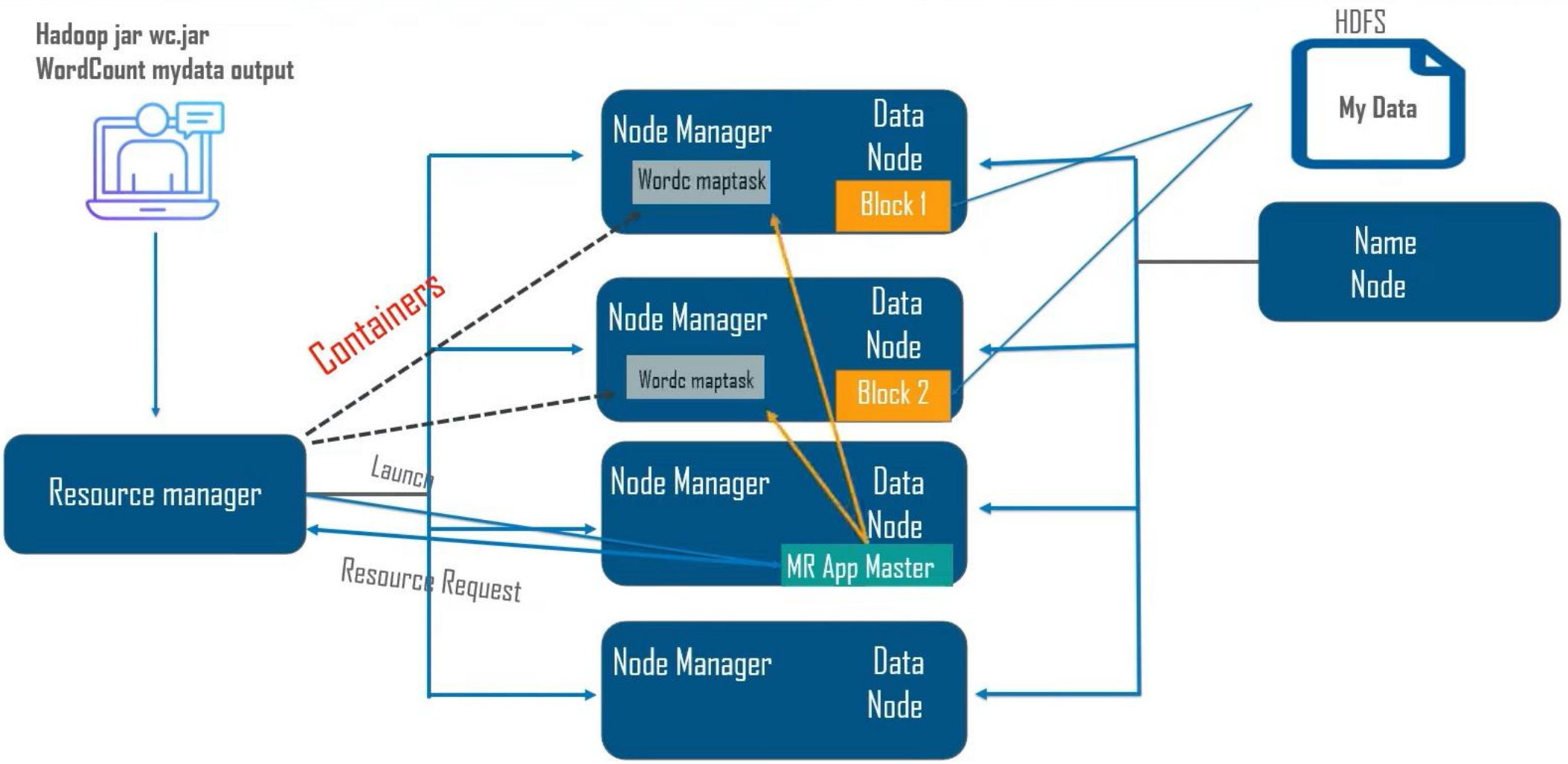
# Running Word Count Application in Map-Reduce-2



# Running Word Count Application in Map-Reduce-2



# Running Word Count Application in Map-Reduce-2




THANK YOU