

YARN

Sundharakumar KB

Department of Computer Science and Engineering
School of Engineering

Shiv Nadar University Chennai

- YARN (Yet Another Resource Negotiator) is Hadoop cluster's resource management system.
- It was introduced in Hadoop v2 mainly to help in MapReduce implementation but in general it can help in any distributed programming paradigms.
- With the introduction of YARN, the roles of job tracker and task trackers were removed and instead Node manager, Application master and resource manager were introduced.

MR1	YARN
Job Tracker	Resource manager, application master
Task Tracker	Node Manager

YARN application

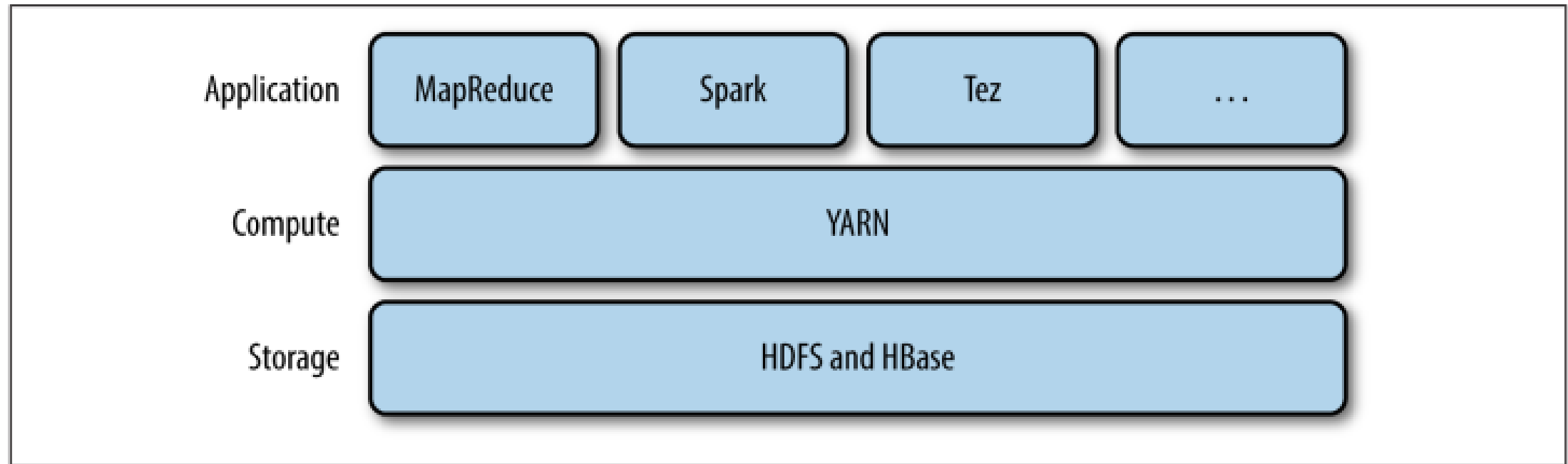


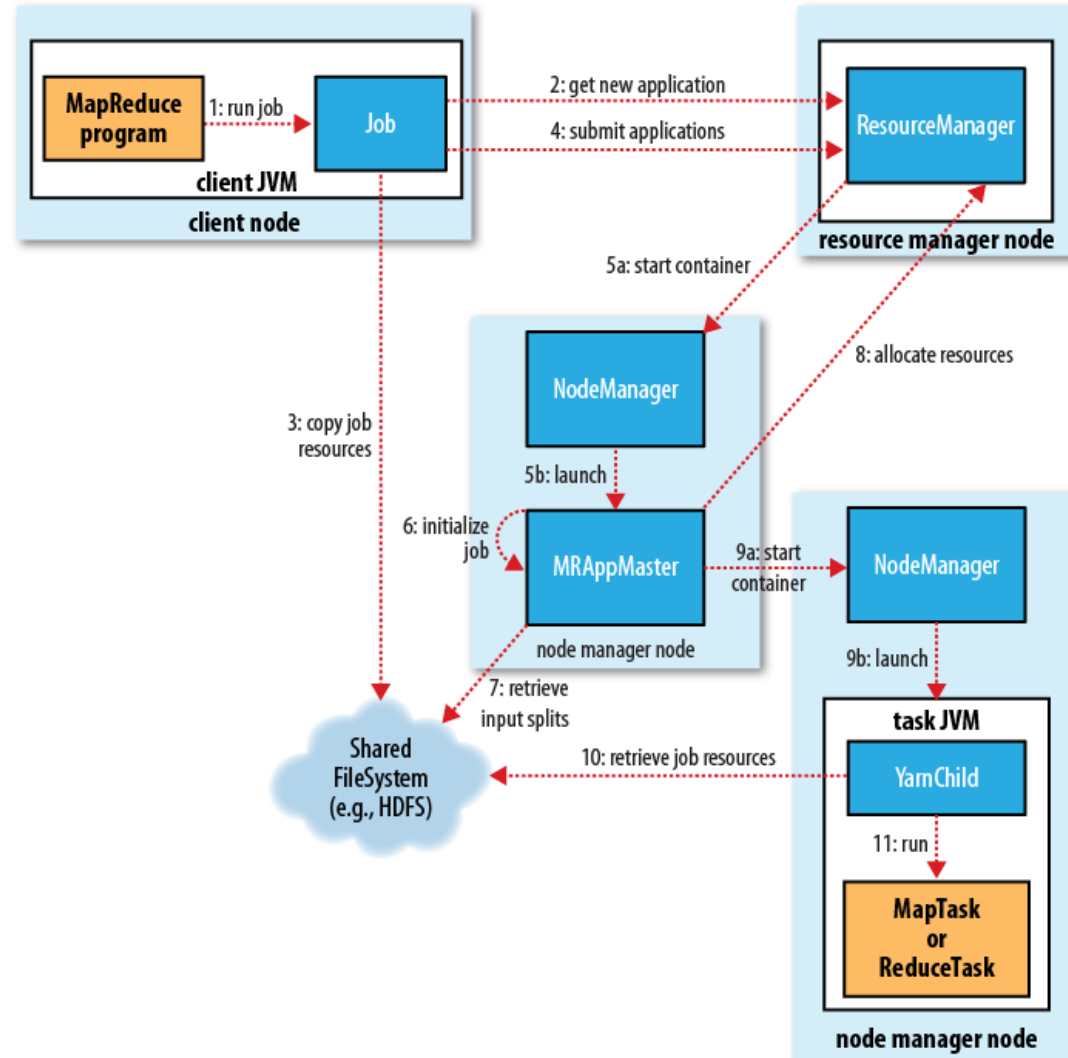
Figure 4-1. YARN applications

- **Resource Manager:** manages the resource allocation in the cluster.
- **Application Master:** handles job life cycle & talks to RM to allocate containers.
- **Node Manager:** manages the containers and status of the data nodes and sends the status to the resource manager.
- **Container :** executes application specific processes.

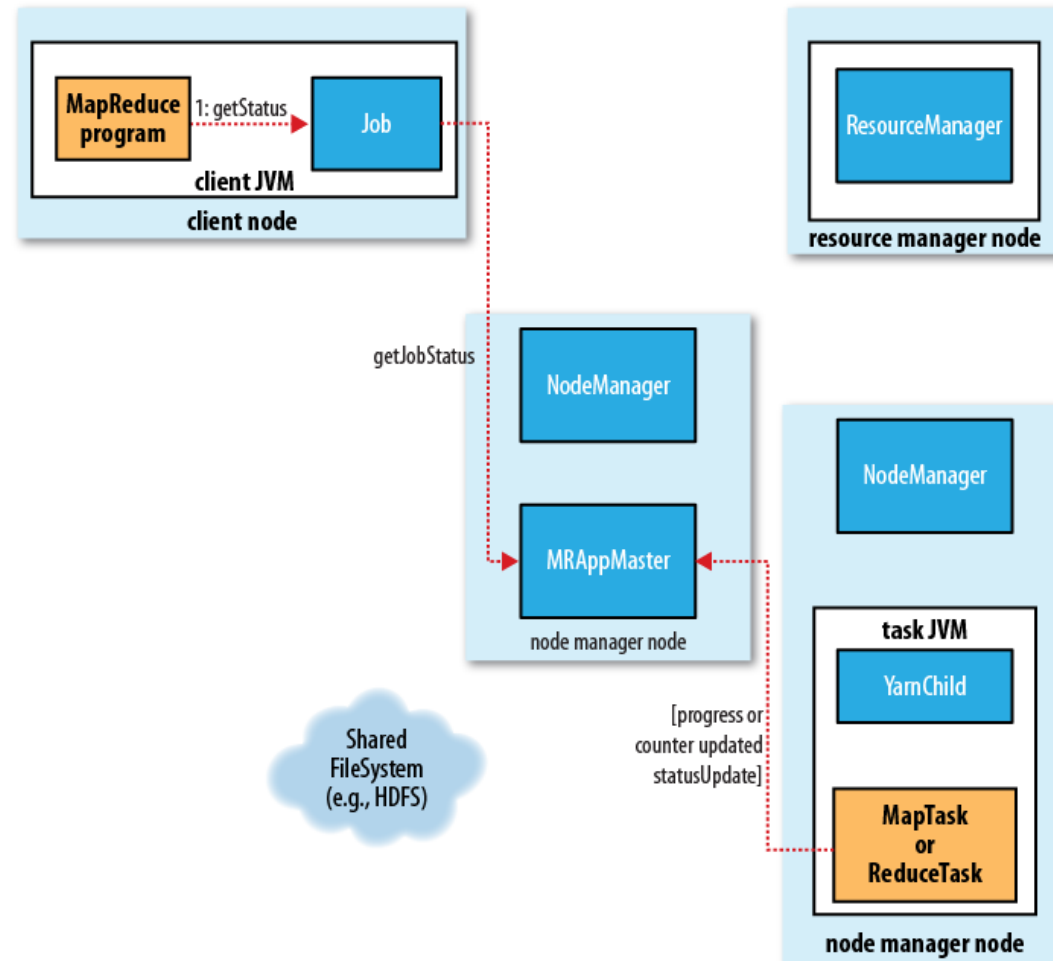
Anatomy of Job run in YARN

- Client submits an application to the resource manager
- RM allocates a container to start the application master
- Application master registers itself with the resource manager.
- Application master negotiates containers from the RM.
- Application master notifies the NM to launch the containers
- Application code is executed in the container.
- Client contacts the RM/appmaster to monitor the status.
- Once the process is complete, application master will un-register with the RM.

Anatomy of Job run in YARN



Job completion in YARN



Hadoop Streaming

- A utility that allows one to run a map/reduce jobs on the cluster with any **executable**; eg: shell scripts, Perl, python, etc.
- Executable reads input from STDIN and write to STDOUT.
- Maps input line data from STDIN to (key,val) pairs in STDOUT.

Hadoop Streaming - example

```
$HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/hadoop-streaming.jar \  
-input myInputDirs \  
-output myOutputDir \  
-mapper /bin/cat \  
-reducer /bin/wc
```

Mapper executable

- When an executable is specified for mappers, each mapper task will launch the executable as a separate process when the mapper is initialized.
- As the mapper task runs, it converts the inputs into lines and feed the lines to the **stdin** of the process.
- Meanwhile, the mapper collects the line-oriented outputs from the **stdout** of the process and converts each line into a key/value pair, which is collected as the output of the mapper.
- By default, the prefix of a line up to the first tab character is the key and the rest of the line (excluding the tab character) will be the value. If there is no tab character in the line, then entire line is considered as key and the value is null. However, this can be customized.

Reducer executable

- When an executable is specified for reducers, each reducer task will launch the executable as a separate process when the reducer is initialized.
- As the reducer task runs, it converts its input key/values pairs into lines and feeds the lines to the **stdin** of the process.
- In the meantime, the reducer collects the line oriented outputs from the **stdout** of the process, converts each line into a key/value pair, which is collected as the output of the reducer.
- By default, the prefix of a line up to the first tab character is the key and the rest of the line (excluding the tab character) is the value. However, this can be customized.

Providing python code as the mapper & reducer file

```
$HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/hadoop-streaming.jar \  
-input myInputDirs \  
-output myOutputDir \  
-mapper myPythonScript.py \  
-reducer /bin/wc \  
-file myPythonScript.py \  
-file myDictionary.txt
```

Customizing map and reduce splits

```
$HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/hadoop-streaming.jar \  
-D stream.map.output.field.separator=. \  
-D stream.num.map.output.key.fields=4 \  
-input myInputDirs \  
-output myOutputDir \  
-mapper org.apache.hadoop.mapred.lib.IdentityMapper \  
-reducer org.apache.hadoop.mapred.lib.IdentityReducer
```

Hadoop Streaming

