UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Setting-up of a YARN/MapReduce cluster

Lecturers: Petar Jovanovic and Sergi Nadal

November 11th, 2016

In this document, we are going to set up the environment to work with MapReduce on top of a YARN cluster.

1 Configure YARN

The file we essentially need to change is *hadoop-2.5.1/etc/hadoop/yarn-site.xml*. Hence, we open it and write the following:

```
<configuration>
        <property>
                <name>yarn.resourcemanager.hostname</name>
                <value>master</value>
        </property>
        <property>
                <name>yarn.nodemanager.resource.memory-mb</name>
                <value>2048</value>
        </property>
        <property>
                <name>yarn.nodemanager.resource.cpu-vcores</name>
                <value>1</value>
        </property>
        <property>
                <name>yarn.nodemanager.aux-services</name>
                <value>mapreduce_shuffle</value>
        </property>
        <property>
                <name>yarn.nodemanager.vmem-check-enabled</name>
                <value>false</value>
        </property>
        <property>
                <name>yarn.nodemanager.pmem-check-enabled</name>
                <value>false</value>
        </property>
</configuration>
```

The explanations are as follows:

1. *yarn.resourcemanager.hostname* determines what machine will work as ResourceManager (i.e., the master) in the YARN cluster. This one normally is assigned the same machine as the HDFS NameNode. This way, the workers will run on top of the DataNodes so data locality is achieved when processing (remember we defined them in the *slaves* file from the Hadoop configuration.

2. *yarn.nodemanager.resource.memory-mb* is the amount of memory a NodeManager can consume. This is an upper cap, thus, if any application consumes more than this, its execution will automatically stop and will report an error (at least the opposite is specified, see item 5). But also, an application will not start running until there is this free amount of memory available in the NodeManagers.

3. *yarn.nodemanager.resource.cpu-vcores* is the number of CPU cores a NodeManager can use. The same rationale as previously also holds here.

4. *yarn.nodemanager.aux-services* is for the configuration of MapReduce. Thus, the intermediate MergeSort process any MapReduce application needs is done in YARN.

5. *yarn.nodemanager.vmem-check-enabled* and *yarn.nodemanager.pmem-check-enabled* are the variables that control whether an application is consuming more than allowed in *yarn.nodemanager.resource.memory-mb* and therefore, they trigger such error previously mentioned. In this case, we set these properties to false so that things become a little bit simplified.

1.1 Configure MapReduce

The only configuration we need to do here is to settle that MapReduce will run on top of YARN. To do so, we firstly copy the *mapred-site.xml.template* to a valid MapReduce configuration file:

```
cp hadoop-2.5.1/etc/hadoop/mapred-site.xml.template
       hadoop-2.5.1/etc/hadoop/mapred-site.xml
```

Afterwards, we edit such copy *hadoop-2.5.1/etc/hadoop/mapred-site.xml* as follows:

```
<configuration>
       <property>
              <name>mapreduce.framework.name</name>
              <value>yarn</value>
       </property>
</configuration>
```

1.2 Replicate the configuration

Finally, as we did in the previous session, we need to replicate all these changes to the rest of the nodes. Previously, however, we moved all the files but now, since just few of them have been changed, we will only send the changed ones. Thus, run the following commands:

```
cd hadoop-2.5.1/etc/hadoop

scp yarn-site.xml mapred-site.xml bdma**@slave1:hadoop-2.5.1/etc/hadoop/.
scp yarn-site.xml mapred-site.xml bdma**@slave2:hadoop-2.5.1/etc/hadoop/.

cd
```

2 Starting up the cluster

Now the cluster should be perfectly configured and, hence, it should be ready to run. The start-up process is very simple and just needs to start both the YARN and the JobHistory server. The JobHistory is just a server that will keep track of the executions a YARN cluster will have run. Thus, when an execution is finished, it is moved to the JobHistory server. Note the HDFS cluster should also be started up:

```
hadoop-2.5.1/sbin/start-yarn.sh
hadoop-2.5.1/sbin/mr-jobhistory-daemon.sh start historyserver
```

After running those two commands, the YARN web UI should be available at port 8088. Thus, the URL *http://MASTER:8088* should be available (check your email for port redirection).

3 Stopping the cluster

As we mentioned in the previous practical work, remember that, for security reasons, your cluster should only be up when working on it. When you finish working, please shut it down by running:

```
hadoop-2.5.1/sbin/stop-yarn.sh
hadoop-2.5.1/sbin/mr-jobhistory-daemon.sh stop historyserver
```

Remember to shut down your HDFS cluster as well.