UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Setting-up of a HBase cluster

Lecturers: Petar Jovanovic and Sergi Nadal

November 4th, 2016

In this document, we are going to set up a HBase cluster on top of the HDFS we already have. First of all you will need to obtain HBase from the tarballs folder.

```
cp tarballs/hbase-1.1.2-bin.tar.gz ~/.
tar xf hbase-1.1.2-bin.tar.gz
```

1 Configure HBase

As we did in the previous session, this setup must be done only in the master node and then it will be replicated to other nodes.

To start with, change the configuration file *hbase-site.xml* that you can find in folder *hbase-1.1.2/conf/hbase-site.xml* to have the following configuration:

```
<configuration>
        <property>
                <name>hbase.rootdir</name>
                <value>hdfs://master:27000/hbase</value>
        </property>
        <property>
                <name>hbase.cluster.distributed</name>
                <value>true</value>
        </property>
        <property>
                <name>hbase.zookeeper.quorum</name>
                <value>slave1,slave2</value>
        </property>
        <property>
                <name>hbase.zookeeper.property.dataDir</name>
                <value>/home/bdma**/zkdata</value>
        </property>
</configuration>
```

And the meaning of the properties is as follows:

- *hbase.rootdir* defines the HDFS connection and the path where HBase is going to run on.

- *hbase.cluster.distributed* must be simply set to true if we want HBase to run in distributed mode.
- *hbase.zookeeper.quorum* is the set of machines that will work as part of the ZooKeeper cluster. Remember HBase uses ZooKeeper to keep track of some cluster information that cannot be distributed (e.g., the B+ tree root). HBase, then, relies to ZooKeeper the maintainability of such information. Here, instead of running our own ZooKeeper instance, we are going to tell HBase that we want it to run its own ZooKeeper, which will be started up and stopped altogether with the HBase cluster.
- *hbase.zookeeper.property.dataDir* is the physical path where ZooKeeper will store its data. Following the same rationale as before, we let HBase manage its own ZooKeeper instance and therefore some configurations must be defined.

Afterwards we need to make clear which nodes are going to be the slaves (the master was set in *hbase.rootdir* property). Thus, modify the file at *hbase-1.1.2/conf/regionservers*.

```
slave1
slave2
```

Finally, the file *hbase-1.1.2/conf/hbase-env.sh* should be changed so that Java is correctly configured to run:

- Firstly, you should look for the line where the variable *JAVA_HOME* is defined and change it as follows:

  ```
  export JAVA_HOME="/home/bdma**/jre1.7.0_75"
  ```

- Then, look for the *HBASE_HEAPSIZE* and assign it 2 GB of memory:

  ```
  export HBASE_HEAPSIZE=2G
  ```

- And the last one, link HBase with the native libraries of Hadoop. This is not essential at all, but it can avoid having some future headaches related with some functionalities not really working.

  ```
  export HBASE_OPTS="-XX:+UseConcMarkSweepGC
                -Djava.library.path=/home/bdma**/hadoop-2.5.1/lib/native"
  ```

2 Adjust Hadoop configuration

One of the things when running Java applications in the Hadoop ecosystem is that they must be aware of some details about the cluster elements configuration. To solve this issue in a very simple manner, add the following

lines at the very end of the *hadoop-2.5.1/etc/hadoop/hadoop-env.sh* file so that the *hadoop* runnable now knows the HBase configuration.

```
export HADOOP_CLASSPATH=${HADOOP_CLASSPATH}:"/home/bdma**/hbase-1.1.2/conf"
```

Also, we should free up some memory in favor of HBase. In the same file change the heapsize to be of 1 GB:

```
export HADOOP_HEAPSIZE=1000
```

3 Replicate the configuration

Finally, as we did in the previous session, we need to replicate all these changes to the rest of nodes. Thus, run the following commands:

```
scp hadoop-2.5.1/etc/hadoop/hadoop-env.sh bdma**@slave1:hadoop-2.5.1/etc/hadoop/.
scp hadoop-2.5.1/etc/hadoop/hadoop-env.sh bdma**@slave2:hadoop-2.5.1/etc/hadoop/.

scp -r hbase-1.1.2 bdma**@slave1:.
scp -r hbase-1.1.2 bdma**@slave2:.
```

4 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 the HBase. Note the HDFS cluster should also be started up:

```
hadoop-2.5.1/sbin/start-dfs.sh
hbase-1.1.2/bin/start-hbase.sh
```

After running those two commands, the HBase web UI should be available at *http://MASTER:16010/* (check your email for port redirection).

5 Stopping the cluster

To stop HBase and HDFS:

```
hbase-1.1.2/bin/stop-hbase.sh
hadoop-2.5.1/sbin/stop-dfs.sh
```