Rasesh Mori

Share what you know…..

Steps to install Hadoop 2.x release (Yarn or Next-Gen) on multi-node cluster

Posted on **October 14, 2012**

In the previous post, we saw how to setup Hadoop 2.x on single-node. Here, we will see how to set up a multi-node cluster.

Hadoop 2.x release involves many changes to Hadoop and MapReduce. The centralized JobTracker service is replaced with a ResourceManager that manages the resources in the cluster and an ApplicationManager that manages the application lifecycle. These architectural changes enable hadoop to scale to much larger clusters. For more details on architectural changes in Hadoop next-gen (a.k.a. Yarn), watch this video or visit this blog.

This post concentrates on installing Hadoop 2.x a.k.a. Yarn a.k.a. next-gen on a multi-node cluster.

Prerequisites:

- Java 6 installed
- Dedicated user for hadoop
- SSH configured

Steps to install Hadoop 2.x:

1. Download tarball

You can download tarball for hadoop 2.x from here. Extract it to a folder say, /home/hduser/yarn on master and all the slaves. We assume dedicated user for Hadoop is “hduser”.

NOTE: Master and all the slaves must have the same user and hadoop directory on same path.

```
$ cd /home/hduser/yarn
$ sudo chown -R hduser:hadoop hadoop-2.0.1-alpha
```

2. Edit /etc/hosts

Add the association between the hostnames and the ip address for the master and the slaves on all the nodes in the /etc/hosts file. Make sure that the all the nodes in the cluster are able to ping to each other.

Important Change:

```
127.0.0.1 localhost localhost.localdomain my-laptop
127.0.1.1 my-laptop
```

If you have provided alias for localhost (as done in entries above), protocol buffers will try to connect to my-laptop from other hosts while making RPC calls which will fail.

Solution:

Assuming the machine (my-laptop) has ip address “10.3.3.43”, make an entry as follows in all the other machines:

```
10.3.3.43       my-laptop
```

3. Password less SSH

Make sure that the master is able to do a password-less ssh to all the slaves.

4. Edit ~/.bashrc

```
export HADOOP_HOME=/home/hduser/yarn/hadoop-2.0.1-alpha
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

5. Edit Hadoop environment files

Add JAVA_HOME to following files

Add following line at start of script in libexec/hadoop-config.sh :

```
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-i386/
```

Add following lines at start of script in etc/hadoop/yarn-env.sh :

```
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-i386/
export HADOOP_HOME=/home/hduser/yarn/hadoop-2.0.1-alpha
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

Change the path as per your java installation.

6. Create Temp folder in HADOOP_HOME

```
$ mkdir -p $HADOOP_HOME/tmp
```

7. Add properties in configuration files

Make changes as mentioned below in all the machines:

$HADOOP_CONF_DIR/core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://master:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/home/hduser/yarn/hadoop-2.0.1-alpha/tmp</value>
  </property>
</configuration>
```

$HADOOP_CONF_DIR/hdfs-site.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
 <configuration>
   <property>
     <name>dfs.replication</name>
     <value>2</value>
   </property>
   <property>
     <name>dfs.permissions</name>
     <value>false</value>
   </property>
 </configuration>
```

$HADOOP_CONF_DIR/mapred-site.xml :

```
<?xml version="1.0"?>
<configuration>
 <property>
   <name>mapreduce.framework.name</name>
   <value>yarn</value>
 </property>
</configuration>
```

$HADOOP_CONF_DIR/yarn-site.xml :

```
<?xml version="1.0"?>
 <configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>master:8025</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>master:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>master:8040</value>
  </property>
 </configuration>
```

8. Add slaves

Add the slave entries in $HADOOP_CONF_DIR/slaves on master machine:

```
slave1
slave2
```

9. Format the namenode

```
$ bin/hadoop namenode -format
```

10. Start Hadoop Daemons

```
$ sbin/hadoop-daemon.sh start namenode
$ sbin/hadoop-daemons.sh start datanode
$ sbin/yarn-daemon.sh start resourcemanager
$ sbin/yarn-daemons.sh start nodemanager
$ sbin/mr-jobhistory-daemon.sh start historyserver
```

NOTE: For datanode and nodemanager, scripts are *-daemons.sh and not *-daemon.sh. daemon.sh does not lookup in slaves file and hence, will only start processes on master

11. Check installation

Check for jps output on slaves and master.

For master:

```
$ jps
6539 ResourceManager
6451 DataNode
8701 Jps
6895 JobHistoryServer
6234 NameNode
6765 NodeManager
```

For slaves:

```
$ jps
8014 NodeManager
7858 DataNode
9868 Jps
```

If these services are not up, check the logs in $HADOOP_HOME/logs directory to identify the issue.

12. Run a demo application to verify installtion

```
$ mkdir in
$ cat > in/file
This is one line
This is another one
```

Add this directory to HDFS:

```
$ bin/hadoop dfs -copyFromLocal in /in
```

Run wordcount example provided:

```
$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.*-a
```

Check the output:

```
$ bin/hadoop dfs -cat /out/*
This 2
another 1
is 2
line 1
one 2
```

13. Web interface

1. http://master:50070/dfshealth.jsp
2. http://master:8088/cluster
3. http://master:19888/jobhistory (for Job History Server)

14. Stopping the daemons

```
$ sbin/mr-jobhistory-daemon.sh stop historyserver
$ sbin/yarn-daemons.sh stop nodemanager
$ sbin/yarn-daemon.sh stop resourcemanager
$ sbin/hadoop-daemons.sh stop datanode
$ sbin/hadoop-daemon.sh stop namenode
```

15. Possible errors

If you get a exception stack trace similar to given below:

```
Container launch failed for container_1350204169962_0002_01_000004 : ja
 at org.apache.hadoop.yarn.exceptions.impl.pb.YarnRemoteExceptionPBImpl
 at org.apache.hadoop.yarn.api.impl.pb.client.ContainerManagerPBClientI
 at org.apache.hadoop.mapreduce.v2.app.launcher.ContainerLauncherImpl$C
 at org.apache.hadoop.mapreduce.v2.app.launcher.ContainerLauncherImpl$E
 at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecuto
 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecut
 at java.lang.Thread.run(Thread.java:679)
Caused by: com.google.protobuf.ServiceException: java.net.UnknownHostEx
 at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcE
 at $Proxy29.startContainer(Unknown Source)
 at org.apache.hadoop.yarn.api.impl.pb.client.ContainerManagerPBClientI
 ... 5 more
Caused by: java.net.UnknownHostException: Invalid host name: local host
 at org.apache.hadoop.net.NetUtils.wrapException(NetUtils.java:740)
 at org.apache.hadoop.ipc.Client$Connection.<init>(Client.java:248)
 at org.apache.hadoop.ipc.Client.getConnection(Client.java:1261)
 at org.apache.hadoop.ipc.Client.call(Client.java:1141)
 at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcE
 ... 7 more
```

```
Caused by: java.net.UnknownHostException
 ... 11 more
```

Solution: Check the Important Change in Step 2 and apply the necessary changes.

Happy Coding!!!

– Rasesh Mori

This entry was posted in **Hadoop** and tagged **2.0**, **2.0.1**, **2.x**, **alpha**, **cluster**, **hadoop**, **hadoop-2.0.1-alpha**, **hadoop-2.0.2-alpha**, **install**, **mapreduce**, **multi**, **nextgen**, **node**, **node cluster**, **setup**, **yarn** by **Rasesh Mori**. Bookmark the **permalink [https://raseshmori.wordpress.com/2012/10/14/install-hadoop-nextgen-yarn-multi-node-cluster/]** .

95 THOUGHTS ON "STEPS TO INSTALL HADOOP 2.X RELEASE (YARN OR NEXT-GEN) ON MULTI-NODE CLUSTER"

Big Data Events
on **January 9, 2013 at 1:02 pm** said:

Nice article, Global Big Data Conference is going to held on Jan 28, 2013, Santa Clara
Register http://bit.ly/10aSvt5

Amit Singh
on **May 24, 2013 at 7:22 pm** said:

Hi, Rashesh
Thanks for nice article

You have mentioned that :
"NOTE: Master and all the slaves must have the same user and hadoop directory on same path."

I have two slaves with same user name but i can't make same path for their hadoop installation directory. As we have different directory structures designed for difference servers (That i am trying to use as slaves) in our organization.

Thats why it is throwing me error for slaves as :
server4: bash: line 0: cd: /home/hduser/hadoop/libexec/..: No such file or directory