Instalação de um cluster Hadoop no Ubuntu 12.04 LTS

Para a instalação do cluster Hadoop, é preciso fazer a primeira parte deste tutorial em todas as máquinas (masters e slaves).

Assim como a primeira parte, este post é um breve resumo do artigo Running Hadoop on Ubuntu Linux (Multi-Node Cluster), do Michael G. Noll. Anotei esses passos apenas como checklist para instalações futuras, mas resolvi compartilhar aqui no blog. Então para mais detalhes eu sugiro que leiam o artigo original.

Antes de começar, certifique-se que:

- Tenha seguido a primeira parte do tutorial em todas máquinas (masters e slaves). É recomendado manter as mesmas estruturas, caminhos de diretórios e versão do sistema operacional.
- Cada máquina tenha um IP devidamente configurado e que seja acessível pelas outras máquinas.
- Todos os serviços Hadoop das máquinas estejam parados, através do comando *stop-all.sh* (na pasta *bin* do diretório do Hadoop, caso você não tenha adicionado essa pasta no $PATH – Passo 12 da primeira parte do artigo).

1. Instalar o Hadoop no modo multi-node

1) Configuração de rede (Master e Slave)

```
$ nano /etc/hosts
```

Adicionar as seguintes linhas (com os devidos IPs da sua rede). Coloque as demais máquinas de acordo com a configuração do seu cluster.

```
192.168.2.7 master
192.168.2.8 slave
```

E retirar (ou comentar, com um '#' no início) essa linha se ela existir:

```
127.0.1.1 master
```

2) SSH (Master)

```
$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hadoop@slave
```

Para testar:

```
$ ssh master
$ ssh slave
```

3) Configuração do Hadoop

> O **master** executará o NameNode para a camada HDFS (Hadoop Distributed File System) e o JobTracker para a camada de processamento do MapReduce. Ambas as máquinas rodarão os daemons DataNode para a camada HDFS e TaskTracker para camada de processamento do MapReduce. Dessa forma, os daemons **masters** serão responsáveis pela coordenação e gerenciamento dos daemons **slaves**, enquanto estes fazem os trabalhos de processamento e armazenamento.

3.1) Master

```
$ nano /usr/local/hadoop/conf/masters

master

$ nano /usr/local/hadoop/conf/slaves

master
slave
```

3.2) Master e Slave(s)

```
$ nano /usr/local/hadoop/conf/core-site.xml

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
        <property>
          <name>hadoop.tmp.dir</name>
          <value>/home/hadoop/tmp</value>
          <description>A base for other temporary directories.</description>
        </property>

        <property>
          <name>fs.default.name</name>
          <value>hdfs://master:54310</value>
          <description>The name of the default file system.  A URI whose
          scheme and authority determine the FileSystem implementation.  The
          uri's scheme determines the config property (fs.SCHEME.impl) naming
          the FileSystem implementation class.  The uri's authority is used to
          determine the host, port, etc. for a filesystem.</description>
        </property>
</configuration>

$ nano /usr/local/hadoop/conf/mapred-site.xml

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
        <property>
          <name>mapred.job.tracker</name>
          <value>master:54311</value>
          <description>The host and port that the MapReduce job tracker runs
          at.  If "local", then jobs are run in-process as a single map
          and reduce task.
          </description>
        </property>
</configuration>

$ nano /usr/local/hadoop/conf/hdfs-site.xml

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<configuration>
        <property>
          <name>dfs.replication</name>
          <value>2</value>
```

```
          <description>Default block replication.
          The actual number of replications can be specified when the file is
created.
          The default is used if replication is not specified in create time.
          </description>
        </property>
</configuration>
```

4) Formatar o HDFS via NameNode

```
hadoop@master:~$ stop-all.sh
hadoop@master:~$ hadoop namenode -format
```

5) Iniciando o cluster

```
hadoop@master:~$ start-dfs.sh
hadoop@master:~$ start-mapred.sh
```

Para conferir:

```
hadoop@master:~$ jps

16017 Jps
14799 NameNode
15686 TaskTracker
14880 DataNode
15596 JobTracker
14977 SecondaryNameNode

hadoop@slave:~$ jps

15183 DataNode
15897 TaskTracker
16284 Jps
```

6) Parando o cluster

```
hadoop@master:~$ stop-mapred.sh
hadoop@master:~$ stop-dfs.sh
```

6) Testando

Baixar vários livros em formato texto puro para a pasta gutenberg, como mostrado anteriormente.

```
$ hadoop dfs -copyFromLocal gutenberg gutenberg
$ hadoop dfs -rmr gutenberg-output
$ cd /usr/local/hadoop
$ hadoop jar hadoop-examples-1.1.0.jar wordcount /user/hadoop/gutenberg
/user/hadoop/gutenberg-output
$ hadoop dfs -cat /user/hadoop/gutenberg-output/part-r-00000
```

Abrir os logs no master e slave(s) para conferir se o DataNode na máquina slave recebeu os blocos do master:

```
$ tail -50 /usr/local/hadoop/logs/hadoop-hadoop-datanode-slave.log
```