

HDFS Homework



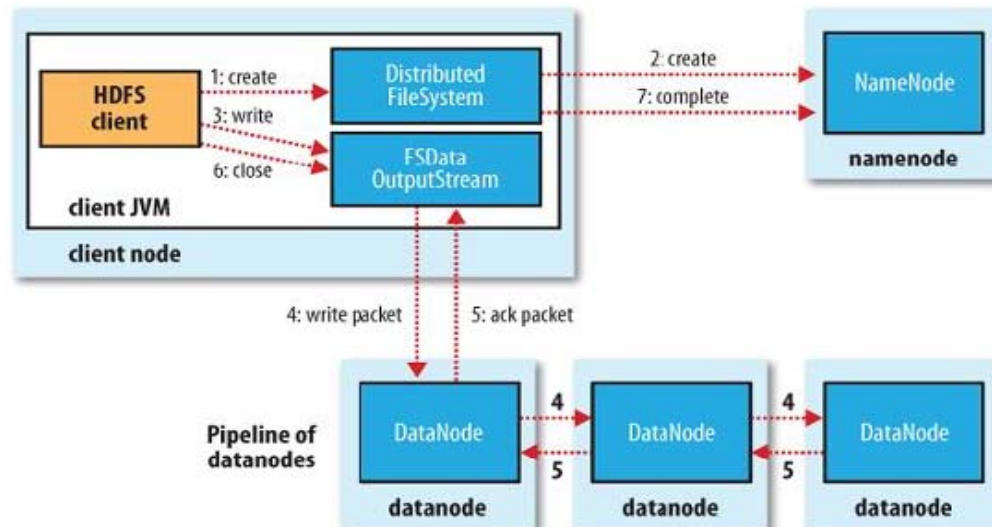
By 李小吉&朱小杰
SmartClouder.com
3/13/2012

Description

1. Use HDFS to put, get, remove data.
2. Build HDFS environment in Linux.

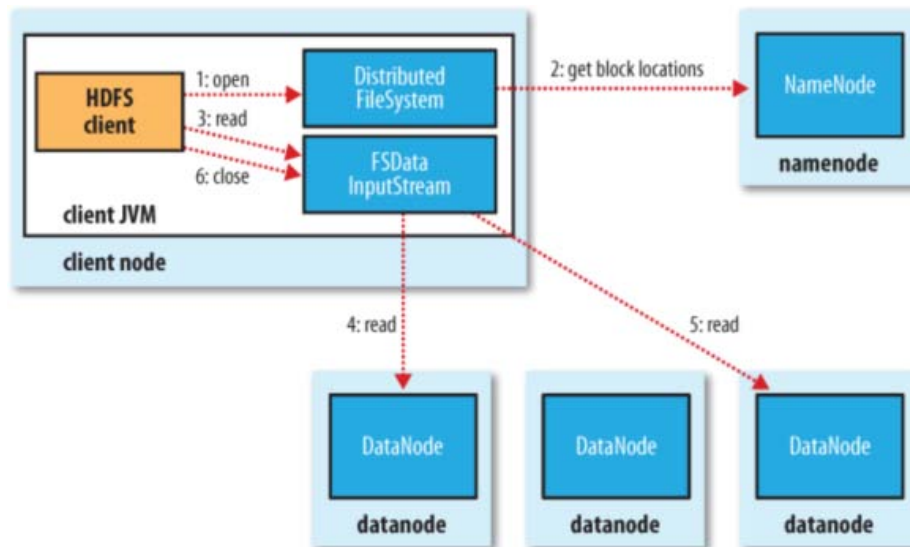
Put the data into HDFS

- `bin/hadoop fs -put my.jpg cloud/my.jpg`
 - `bin/hadoop`: the command to use hadoop.
 - `fs`: tell hadoop to run the HDFS client.
 - `-put`: the client will put the src into HDFS.
 - `my.jpg`: the source file in local.
 - `cloud/my.jpg`: the destination file in HDFS.



Get the data from HDFS

- `bin/hadoop fs -get cloud/my.jpg my2.jpg`
 - `bin/hadoop`: the command to use hadoop.
 - `fs`: tell hadoop to run the HDFS client.
 - `-get`: the client will get the destination file from HDFS.
 - `cloud/my.jpg`: the destination file path in HDFS.
 - `my2.jpg`: the local file name be stored.



Implementation

Step 1. Install

```
$ apt-get install ssh
```

```
$ apt-get install rsync
```

```
$ apt-get install openjdk-6-jdk
```

```
$ wget http://mirror.bjtu.edu.cn/apache/hadoop/core/stable/hadoop-0.20.203.0rc1.tar.gz
```

```
$ tar -xzvf hadoop-0.20.203.0rc1.tar.gz
```

Step 2. Configure

```
$vi conf/hadoop-env.sh.
```

Set JAVA_HOME as the path of the jdk.

Eg: export JAVA_HOME=/usr/lib/jvm/java-6-openjdk/jre

ssh to the localhost without a passphrase

```
$ ssh-keygen -t dsa -P "" -f ~/.ssh/id_dsa
```

```
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

Core-site.xml

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

mapred-site.xml

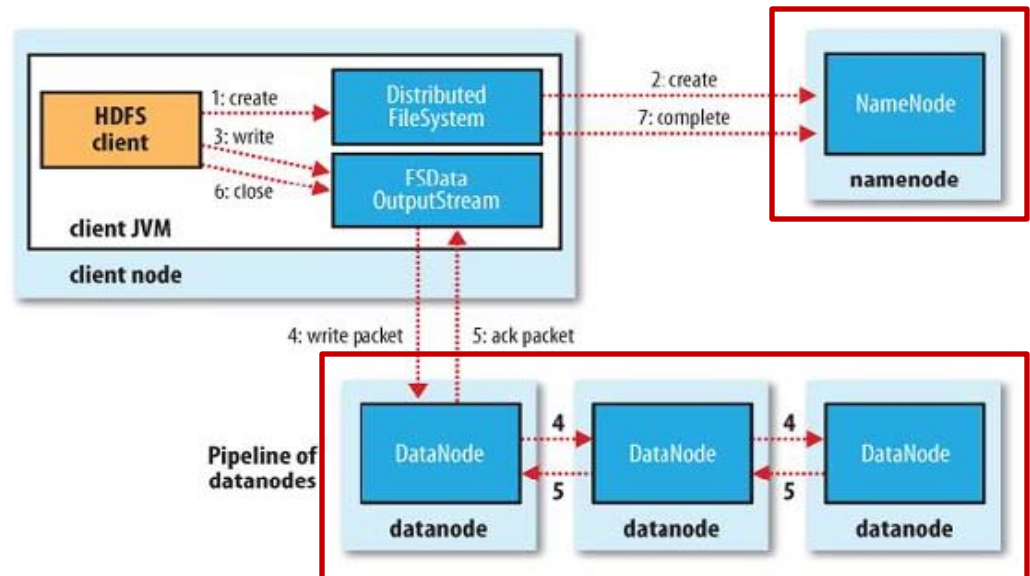
```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9001</value>
  </property>
</configuration>
```

Step 3. Startup

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

```
$ bin/start-all.sh
```

make sure namenode, datanode are running.



```
$ ps -efH
```

```
/usr/lib/jvm/java-6-openjdk/jre/bin/java -Dproc_namenode -Xmx1000m
```

```
/usr/lib/jvm/java-6-openjdk/jre/bin/java -Dproc_datanode -Xmx1000m
```


Trouble shooting

- Trouble 1: “Unrecognized option: -jvm” or “Could not create the Java”
 - solution: vi hadoop/bin/hadoop file, find “-jvm”, then delete “-jvm” , that is ok.
- Trouble 2: “java.net.UnknownHostException:”
 - Solution:
 - 1. get the hostname: run command “hostname”, linux will return hostname, e.g.: “smartclouder0”.
 - 2. vi /etc/hosts, append the string “127.0.0.1 smartclouder0” as a line to the end of the file.
- Other troubles:
 - <http://pages.cs.brandeis.edu/~cs147a/lab/hadoop-troubleshooting/>

Step 4. Test

```
$ bin/hadoop fs -put my.jpg cloud/my.jpg
```

```
$ bin/hadoop fs -ls cloud/my.jpg
```

```
$ bin/hadoop fs -get cloud/my.jpg my2.jpg
```

```
$ bin/hadoop fs -rm cloud/my.jpg
```

```
$ bin/hadoop fs -ls cloud/my.jpg
```

Q&A



SmartClouder.com