Hadoop集群的安装和配置，主要分为两个部分：一部分是主机环境配置，主要是指Hadoop集群所依赖的操作系统及其相关软件的安装配置，包括操作系统安装、JDK安装配置、主机规划与IP地址映射配置、无密码认证会话配置；另一部分是Hadoop基本配置，主要是指Hadoop集群的各种基本组件的配置，包括HDFS的配置、MapReduce配置。

下面根据上面总结的两个部分进行详细配置实践。在配置之前，先准备一下我们使用的软件，如下所示：

* 操作系统：Ubuntu-11.10
* Sun JDK：jdk-6u30-linux-i586.bin
* Hadoop：hadoop-0.22.0.tar.gz

**主机环境配置**

* **JDK安装配置**

直接安装即可，执行如下命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. cd ~/installation
2. chmod +x jdk-6u30-linux-i586.bin
3. ./jdk-6u30-linux-i586.bin

JDK配置，需要修改环境变量文件~/.bashrc（vi ~/.bashrc）， 在~/.bashrc文件的最后面增加如下配置行，如下所示：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. export JAVA\_HOME=/home/hadoop/installation/jdk1.6.0\_30
2. export CLASSPATH=$JAVA\_HOME/lib/\*.jar:$JAVA\_HOME/jre/lib/\*.jar
3. export PATH=$PATH:$JAVA\_HOME/bin

最后，使配置生效，并验证：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. . ~/.bashrc
2. echo $JAVA\_HOME

在每台机器上的配置，都按照上面的步骤进行配置。

* **主机规划与IP地址映射配置**

**1、主结点master配置**

集群中分为主结点（Master Node）和从结点（Slave Node）两种结点。我们选择一个主结点，两个从结点进行配置。

主结点IP为192.168.0.190，修改主结点的/etc/hosts（sudo vi /etc/hosts），内容配置如下所示（不需要改动IPV6的配置内容）：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. 127.0.0.1       localhost
2. 192.168.0.190   master
3. 192.168.0.186   slave-01
4. 192.168.0.183   slave-02
5. # The following lines are desirable for IPv6 capable hosts
6. ::1     ip6-localhost ip6-loopback
7. fe00::0 ip6-localnet
8. ff00::0 ip6-mcastprefix
9. ff02::1 ip6-allnodes
10. ff02::2 ip6-allrouters

Ubuntu系统，还要修改主机名，对应的配置文件为/etc/hostname，修改该配置文件（sudo vi /etc/hostname），内容如下所示：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. master

这里，因为在集群启动以后，master需要登录到任意一个从结点上，执行相应的授权操作，所以，master的/etc/hosts中要配置上集群中全部从结点的主机名到对应IP地址的映射。

从结点有两个，分别为slave-01（192.168.0.186）、slave-02（192.168.0.183），对从结点的配置方式相同。

**2、从结点slave-01配置**

slave-01结点的/etc/hosts配置，内容如下所示：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. 127.0.0.1       localhost
2. 192.168.0.190   master
3. 192.168.0.186   slave-01
4. # The following lines are desirable for IPv6 capable hosts
5. ::1     ip6-localhost ip6-loopback
6. fe00::0 ip6-localnet
7. ff00::0 ip6-mcastprefix
8. ff02::1 ip6-allnodes
9. ff02::2 ip6-allrouters

修改192.168.0.186的主机名，对应的配置文件为/etc/hostname，修改该配置文件（sudo vi /etc/hostname），内容如下所示：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. slave-01

**3、从结点slave-02配置**

slave-02结点的/etc/hosts配置，内容如下所示：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. 127.0.0.1       localhost
2. 192.168.0.190   master
3. 192.168.0.183   slave-02
4. # The following lines are desirable for IPv6 capable hosts
5. ::1     ip6-localhost ip6-loopback
6. fe00::0 ip6-localnet
7. ff00::0 ip6-mcastprefix
8. ff02::1 ip6-allnodes
9. ff02::2 ip6-allrouters

修改192.168.0.183的主机名，对应的配置文件为/etc/hostname，修改该配置文件（sudo vi /etc/hostname），内容如下所示：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. slave-02

**4、总结说明**

**为什么建议使用主机名称，而不直接使用IP地址？**

第一，比较直观。通过看主机名称，你就能够知道集群中哪台机器是什么角色。  
第二，扩展性好。因为通过/etc/hosts对主机名和IP地址进行了映射，即使IP变了，主机名可以保持不变。在Hadoop安装的时候，需要配置master和slaves两个文件，如果这两个文件都使用IP的话，试想，一个具有200个结点的集群，因为一次企业的网络规划重新分配网段，导致IP全部变更，那么这些配置都要进行修改，工作量很大。但是，如果使用主机名的话，Hadoop的配置不需要任何改变，而对于主机名到IP地址的映射配置交给系统管理员去做好了。

* **无密码认证会话配置**

**1、基本配置**

首先检查，你的ssh是否安装并启动，如果没有则进行安装：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. sudo apt-get install openssh-server

在master主结点上，生成密钥对：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. ssh-keygen -t rsa

一路回车下去即可，即可生成公钥（~/.ssh/id\_rsa.pub）和私钥（~/.ssh/id\_rsa）文件。

添加认证公钥，并设置权限：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys
2. chmod 644 ~/.ssh/authorized\_keys
3. 推送centos的方法ssh-copy-id -i /root/.ssh/id\_rsa.pub root@normal-slaver03,centos别模仿下面的推KEY法

验证配置，执行如下命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. ssh master

如果不需要输入密码，即可登录master（登录本机），说明配置正确。

接着，我们要进行认证公钥的分发传播。

（1）远程拷贝master的公钥到slave-01结点上

执行如下命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. scp ~/.ssh/id\_rsa.pub hadoop@slave-01:/home/hadoop/.ssh/id\_rsa.pub.master

这时，因为结点之间（master到slave-01）要进行数据交换，需要输入slave-01结点的登录密码（hadoop用户），才能执行文件的远程拷贝。这里输入密码是正常的，不要与结点之间通过ssh进行无密码公钥认证混淆。  
注意：分发公钥时，主要修改目标master公钥文件副本名称，以防覆盖掉从结点上已经存在并且正在使用的公钥文件。

在slave-01结点上，将master生成的密钥加入到认证密钥文件：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. cat ~/.ssh/id\_rsa.pub.master >> ~/.ssh/authorized\_keys
2. chmod 644 ~/.ssh/authorized\_keys

（2）远程拷贝master的公钥到slave-02结点上

类似地，执行如下命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. scp ~/.ssh/id\_rsa.pub hadoop@slave-02:/home/hadoop/.ssh/id\_rsa.pub.master

在slave-02结点上，将master生成的密钥加入到认证密钥文件，同样执行命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. cat ~/.ssh/id\_rsa.pub.master >> ~/.ssh/authorized\_keys
2. chmod 644 ~/.ssh/authorized\_keys

这时，可以验证从master结点分别登录到slave-01和slave-02结点：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. ssh slave-01
2. ssh slave-02

如果不需要输入密码，则配置成功。

**2、总结说明**

**（1）为什么要将master上的公钥分发到集群中各从结点slaves上？**

密钥是用来进行安全认证的。在两个结点之间进行认证，主要有两种方式：  
一是通过输入密码进行认证，你必须知道对方主机的登录用户名和口令，才能登录到对方主机上进行合法的授权操作；  
二是不需要密码就能够认证，这就需要用到密钥，当一个主机A访问另一个主机B时，如果对方主机B的认证密钥配置了A的公钥，那么A访问B是就能够通过配置的A的公钥进行认证，而不需要进行输入密码认证。  
Hadoop集群master分发公钥到slaves结点，并且，在各个slaves结点上配置了公钥认证，这时，当master通过ssh登录到slaves结点上以后，可以执行相应的授权操作，例如，当master要停止整个HDFS集群（namenode、datanode）时，可以在master上就能直接登录到各个slaves结点上，直接关闭datanode，从而关闭整个集群。这样的话，你就不需要分别登录每个datanode上，去执行相应的关闭操作。

**（2）使用dsa密钥还是rsa密钥？**  
DSA和RSA都可以用于认证，不过是基于不同的加密和解密算法的而已。有关DSA和RSA可以查阅相关资料。

**Hadoop集群配置**

* **Hadoop集群基本配置**

**1、解压缩**

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. cd ~/installation
2. tar -xvzf hadoop-0.22.0.tar.gz

**2、配置Hadoop环境变量**

修改vi ~/.bashrc，在文件最后面加上如下配置：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. export HADOOP\_HOME=/home/hadoop/installation/hadoop-0.22.0
2. export PATH=$PATH:$HADOOP\_HOME/bin

使配置~/.bashrc生效，执行：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. . ~/.bashrc

**4、配置master和slaves文件**

修改conf/master文件，内容如下所示：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. master

修改conf/slaves文件，内容如下所示：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. slave-01
2. slave-02

**5、配置hadoop-env.sh文件**

配置文件conf/hadoop-env.sh，只需要修改JAVA\_HOME配置，如下所示：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. export JAVA\_HOME=/home/hadoop/installation/jdk1.6.0\_30

其它选项，可以根据需要进行配置。

**6、配置conf/core-site.xml文件**

配置文件conf/core-site.xml的内容，如下所示：

**[html]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. **<?xml** version="1.0"**?>**
2. **<?xml-stylesheet** type="text/xsl" href="configuration.xsl"**?>**
3. **<configuration>**
4. **<property>**
5. **<name>**fs.default.name**</name>**
6. **<value>**hdfs://master:9000/**</value>**
7. **<description></description>**
8. **</property>**
9. **<property>**
10. **<name>**dfs.replication**</name>**
11. **<value>**3**</value>**
12. **</property>**
13. **<property>**
14. **<name>**fs.inmemory.size.mb**</name>**
15. **<value>**10**</value>**
16. **<description>**Larger amount of memory allocated for the in-memory file-system used to merge map-outputs at the reduces.**</description>**
17. **</property>**
18. **<property>**
19. **<name>**io.sort.factor**</name>**
20. **<value>**10**</value>**
21. **<description>**More streams merged at once while sorting files.**</description>**
22. **</property>**
23. **<property>**
24. **<name>**io.sort.mb**</name>**
25. **<value>**10**</value>**
26. **<description>**Higher memory-limit while sorting data.**</description>**
27. **</property>**
28. **<property>**
29. **<name>**io.file.buffer.size**</name>**
30. **<value>**131072**</value>**
31. **<description>**Size of read/write buffer used in SequenceFiles.**</description>**
32. **</property>**
33. **<property>**
34. **<name>**hadoop.tmp.dir**</name>**
35. **<value>**/home/hadoop/storage/tmp/hadoop-${user.name}**</value>**
36. **<description></description>**
37. **</property>**
38. **</configuration>**

上面配置内容，是与HDFS的基本属性相关的，一般在系统运行过程中比较固定的配置，都放到这里面。如果需要根据实际应用的变化，可以配置到hdfs-site.xml文件中，下面会解释。

**7、配置conf/hdfs-site.xml文件**

配置文件conf/hdfs-site.xml的内容，如下所示：

**[html]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. **<?xml** version="1.0"**?>**
2. **<?xml-stylesheet** type="text/xsl" href="configuration.xsl"**?>**
3. **<configuration>**
4. **<property>**
5. **<name>**dfs.name.dir**</name>**
6. **<value>**/home/hadoop/storage/name/a,/home/hadoop/storage/name/b**</value>**
7. **<description>**Path on the local filesystem where the NameNode stores the namespace and transactions logs persistently.**</description>**
8. **</property>**
9. **<property>**
10. **<name>**dfs.data.dir**</name>**
11. **<value>**/home/hadoop/storage/data/a,/home/hadoop/storage/data/b,/home/hadoop/storage/data/c**</value>**
12. **<description>**Comma separated list of paths on the local filesystem of a DataNode where it should store its blocks.**</description>**
13. **</property>**
14. **<property>**
15. **<name>**dfs.block.size**</name>**
16. **<value>**67108864**</value>**
17. **<description>**HDFS blocksize of 64MB for large file-systems.**</description>**
18. **</property>**
19. **<property>**
20. **<name>**dfs.namenode.handler.count**</name>**
21. **<value>**10**</value>**
22. **<description>**More NameNode server threads to handle RPCs from large number of DataNodes.**</description>**
23. **</property>**
24. **</configuration>**

该配置文件配置与HDFS相关的属性，而且这些属性可能在使用计算过程中（如进行MapReduce计算），需要变化，如数据存储目录等等。如果该配置文件hdfs-site.xml和core-site.xml中同时都配置了某个属性，则hdfs-site.xml会覆盖掉core-site.xml中配置的属性。

**8、配置conf/mapred-site.xml文件**

配置文件conf/mapred-site.xml是与MapReduce计算相关的，在实际使用中根据需要进行配置某些参数，如JVM堆内存分配大小等。该配置文件的内容，配置如下所示：

**[html]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. **<?xml** version="1.0"**?>**
2. **<?xml-stylesheet** type="text/xsl" href="configuration.xsl"**?>**
3. **<configuration>**
4. **<property>**
5. **<name>**mapred.job.tracker**</name>**
6. **<value>**hdfs://master:19830/**</value>**
7. **<description>**Host or IP and port of JobTracker.**</description>**
8. **</property>**
9. **<property>**
10. **<name>**mapred.system.dir**</name>**
11. **<value>**/home/hadoop/storage/mapred/system**</value>**
12. **<description>**Path on the HDFS where where the MapReduce framework stores system files.Note: This is in the default filesystem (HDFS) and must be accessible from both the server and client machines.**</description>**
13. **</property>**
14. **<property>**
15. **<name>**mapred.local.dir**</name>**
16. **<value>**/home/hadoop/storage/mapred/local**</value>**
17. **<description>**Comma-separated list of paths on the local filesystem where temporary MapReduce data is written. Note: Multiple paths help spread disk i/o.**</description>**
18. **</property>**
19. **<property>**
20. **<name>**mapred.tasktracker.map.tasks.maximum**</name>**
21. **<value>**10**</value>**
22. **<description>**The maximum number of Map tasks, which are run simultaneously on a given TaskTracker, individually.Note: Defaults to 2 maps, but vary it depending on your hardware.**</description>**
23. **</property>**
24. **<property>**
25. **<name>**mapred.tasktracker.reduce.tasks.maximum**</name>**
26. **<value>**2**</value>**
27. **<description>**The maximum number of Reduce tasks, which are run simultaneously on a given TaskTracker, individually. Note: Defaults to 2 reduces, but vary it depending on your hardware.**</description>**
28. **</property>**
29. **<property>**
30. **<name>**mapred.reduce.parallel.copies**</name>**
31. **<value>**5**</value>**
32. **<description>**Higher number of parallel copies run by reduces to fetch outputs from very large number of maps.**</description>**
33. **</property>**
34. **<property>**
35. **<name>**mapred.map.child.java.opts**</name>**
36. **<value>**-Xmx128M**</value>**
37. **<description>**Larger heap-size for child jvms of maps.**</description>**
38. **</property>**
39. **<property>**
40. **<name>**mapred.reduce.child.java.opts**</name>**
41. **<value>**-Xms64M**</value>**
42. **<description>**Larger heap-size for child jvms of reduces.**</description>**
43. **</property>**
44. **<property>**
45. **<name>**tasktracker.http.threads**</name>**
46. **<value>**5**</value>**
47. **<description>**More worker threads for the TaskTracker's http server. The http server is used by reduces to fetch intermediate map-outputs.**</description>**
48. **</property>**
49. **<property>**
50. **<name>**mapred.queue.names**</name>**
51. **<value>**default**</value>**
52. **<description>**Comma separated list of queues to which jobs can be submitted. Note: The MapReduce system always supports atleast one queue with the name as default. Hence, this parameter's value should always contain the string default. Some job schedulers supported in Hadoop, like the Capacity Scheduler(http://hadoop.apache.org/common/docs/stable/capacity\_scheduler.html), support multiple queues. If such a scheduler is being used, the list of configured queue names must be specified here. Once queues are defined, users can submit jobs to a queue using the property name mapred.job.queue.name in the job configuration. There could be a separate configuration file for configuring properties of these queues that is managed by the scheduler. Refer to the documentation of the scheduler for information on the same.**</description>**
53. **</property>**
54. **<property>**
55. **<name>**mapred.acls.enabled**</name>**
56. **<value>**false**</value>**
57. **<description>**Boolean, specifying whether checks for queue ACLs and job ACLs are to be done for authorizing users for doing queue operations and job operations. Note: If true, queue ACLs are checked while submitting and administering jobs and job ACLs are checked for authorizing view and modification of jobs. Queue ACLs are specified using the configuration parameters of the form mapred.queue.queue-name.acl-name, defined below under mapred-queue-acls.xml. Job ACLs are described at Job Authorization(http://hadoop.apache.org/common/docs/stable/mapred\_tutorial.html#Job+Authorization).**</description>**
58. **</property>**
59. **</configuration>**

**9、安装文件远程分发**

前面，我们已经在各台机器上配置好了基础环境，同时在master上配置好了Hadoop，这时需要将Hadoop拷贝到从结点上相应的目录下。由于Hadoop发行包中存在很多源码和文档，占用了很大存储空间，可以将其删除以后，再进行远程拷贝，删除前先做好备份，我暂时删除了如下文件：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. rm -rf common/ hdfs/ contrib/ c++/ mapreduce/

接着，执行远程拷贝命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. cd ~/installation
2. scp -r ~/installation/hadoop-0.22.0/ hadoop@slave-01:/home/hadoop/installation
3. scp -r ~/installation/hadoop-0.22.0/ hadoop@slave-02:/home/hadoop/installation

**10、其它配置**

因为我们在上面的配置中，使用了存储目录storage，需要预先在master和slaves结点上创建该目录，执行如下命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. mkdir ~/storage

* **Hadoop集群配置验证**

**1、启动HDFS集群**

在master上，启动HDFS集群。

格式化HDFS，执行命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. hdfs namenode -format

如果没有错误，继续执行，启动HDFS，执行命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. start-dfs.sh

此时：  
在master上，执行jps你可以看到，启动了NameNode和SecondaryNameNode；  
在slaves上，执行jps你可以看到，启动了DameNode。

你可以在各个结点上查看对应的日志，看HDFS集群启动的状况，确定启动是否正常：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. tail -500f $HADOOP\_HOME/logs/hadoop-hadoop-namenode-master.log
2. tail -500f $HADOOP\_HOME/logs/hadoop-hadoop-secondarynamenode-master.log
3. tail -500f $HADOOP\_HOME/logs/hadoop-hadoop-datanode-slave-01.log
4. tail -500f $HADOOP\_HOME/logs/hadoop-hadoop-datanode-slave-02.log

还可以通过Hadoop内置的Web Server（Jetty），通过浏览器访问监控：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. master结点：  http://master:50070 或者 http://192.168.0.190:50070
2. slave-01结点：http://slave-01:50075 或者 http://192.168.0.186:50075
3. slave-02结点：http://slave-02:50075 或者 http://192.168.0.183:50075

**2、启动MapReduce任务集群**

在master上，启动MapReduce集群，执行命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. start-mapred.sh

此时：

在master上可以看到，JobTracker已经启动；

在slaves上可以看到，TaskTracker已经启动。

也可以查看日志文件，验证启动是否正常：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. tail -500f $HADOOP\_HOME/logs/hadoop-hadoop-jobtracker-master.log
2. tail -500f $HADOOP\_HOME/logs/hadoop-hadoop-tasktracker-slave-01.log
3. tail -500f $HADOOP\_HOME/logs/hadoop-hadoop-tasktracker-slave-02.log

还可以通过Hadoop内置的Web Server（Jetty），通过浏览器访问监控：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. master结点：  http://master:50030 或者 http://192.168.0.190:50030
2. slave-01结点：http://slave-01:50060 或者 http://192.168.0.186:50060
3. slave-02结点：http://slave-02:50060 或者 http://192.168.0.183:50060

**3、上传文件到HDFS**

例如，上传一个文件到HDFS上，使用copyFromLocal命令：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. hadoop@master:~/installation/hadoop-0.22.0$ hadoop fs -lsr
2. drwxr-xr-x   - hadoop supergroup          0 2011-12-31 11:40 /user/hadoop/storage
3. hadoop@master:~/installation/hadoop-0.22.0$ hadoop fs -mkdir storage/input
4. hadoop@master:~/installation/hadoop-0.22.0$ hadoop fs -copyFromLocal ~/storage/files/extfile.txt  ./storage/input/attractions.txt
5. hadoop@master:~/installation/hadoop-0.22.0$ hadoop fs -lsr
6. drwxr-xr-x   - hadoop supergroup          0 2011-12-31 11:41 /user/hadoop/storage
7. drwxr-xr-x   - hadoop supergroup          0 2011-12-31 11:41 /user/hadoop/storage/input
8. -rw-r--r--   3 hadoop supergroup      66609 2011-12-31 11:41 /user/hadoop/storage/input/attractions.txt

将文件extfile.txt上传到HDFS，改名为attractions.txt。

**4、运行MapReduce任务**

执行如下命令，运行Hadoop自带的例子：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. hadoop jar $HADOOP\_HOME/hadoop-mapred-examples-0.22.0.jar wordcount ./storage/input/ $HADOOP\_HOME/output

**5、总结说明**

**在Hadoop集群外部为什么无法通过域名（如：http://master:50030）访问Hadoop集群结点？**

你在Hadoop集群的外部，例如，在局域网的Windows 7系统中访问master，必须能够将域名master映射到对应的IP地址，否则无法建立到目标主机的链路。在Windows 7下，可以通过设置C:\Windows\System32\drivers\etc目录下面的hosts映射文件，增加如下映射内容：

**[plain]** [view plaincopy](http://blog.csdn.net/shirdrn/article/details/7166513)

1. 192.168.0.190   master
2. 192.168.0.186   slave-01