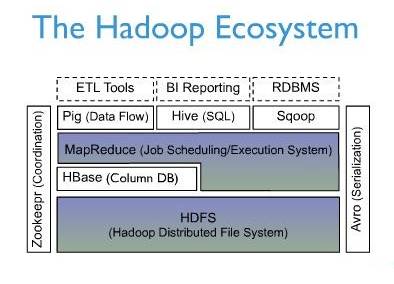
# Hadoop



## 安装

### 安装Java

Hadoop无法使用Linux自带的OpenJDK，须下载安装Oracle公司的JDK（1.6版本）

[root@localhost ~]# rpm -qa | grep jdk

java-1.6.0-openjdk-1.6.0.0-1.57.1.11.9.el6\_4.i686

[root@localhost ~]# yum -y remote java-1.6.0-openjdk-1.6.0.0-1.57.1.11.9.el6\_4.i686

[root@localhost software]# rpm -ivh jdk-7u21-linux-i586.rpm

[root@node0 ~]# java -version

//所有节点都关闭防火墙

[root@node0 conf]# service iptables stop

### localhost无密码登录

注意：以下都用hadoop用户安装和启动Hadoop，因为用root安装/启动hadoop会造成目录权限问题，无法读写HDFS

1. **为Hadoop新建用户**

[root@node0 ~]# useradd hadoop

[root@node0 ~]# passwd hadoop

1. **参考SSH章节实现hadoop用户各节点间无密码访问**
2. **hadoop用户无密码访问localhost**

[hadoop@node0 ~]# ssh-keygen -t dsa -P "" -f ~/.ssh/id\_dsa

[hadoop@node0 ~]# cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

[hadoop@node0 ~]# ssh localhost

### 配置Hadoop

[hadoop@node0 software]# tar zxvf hadoop-1.1.2.tar.gz

[hadoop@node0 software]# mv hadoop-1.1.2 hadoop

[hadoop@node0 software]# cd hadoop/conf

[hadoop@node0 conf]# vi hadoop-env.sh

export JAVA\_HOME=/usr/java/jdk1.7.0\_21

//NameNode的IP和端口

[hadoop@node0 conf]# vi core-site.xml

//fs.default.name为NameNode的URI，不能为localhost，否则window的eclipse连接将出错

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://192.168.0.59:9000</value>

</property>

</configuration>

[hadoop@node0 conf]# vi hdfs-site.xml

// dfs.replication为备份数据，默认为3，单机为1

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

[hadoop@node0 conf]# vi mapred-site.xml

// mapred.job.tracker为JobTracker的IP和端口

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>192.168.0.59:9001</value>

</property>

</configuration>

[hadoop@node0 conf]# vi masters

node0

[hadoop@node0 conf]# vi slaves

node1

[hadoop@node0 conf]# vi ~/.bash\_profile

export JAVA\_HOME=/usr/java/jdk1.7.0\_21

export CLASSPATH=$JAVA\_HOME/lib:$CLASSPATH

PATH=$PATH:$JAVA\_HOME/bin:/home/hadoop/software/hadoop/bin

[hadoop@node0 conf]# source ~/.bash\_profile

**其他各节点都如上安装jdk**

//将配置好的hadoop拷贝到其他各节点

[hadoop@node1 ~]# mkdir /home/hadoop/software/hadoop

[hadoop@node0 conf]# scp -r /home/hadoop/software/hadoop node1: /home/hadoop/software/hadoop

//格式化HDFS文件系统，默认在/tmp目录下生成名为hadoop-{username}的目录，即hadoop-hadoop目录

[hadoop@node0 conf]# hadoop namenode -format

//启动hadoop

[hadoop@node0 bin]# ./start-all.sh

//停止hadoop

[hadoop@node0 bin]# ./stop-all.sh

//查看是否成功启动

[hadoop@node0 bin]# jps

13662 JobTracker

13773 Jps

13575 SecondaryNameNode

13410 NameNode

[hadoop@node0 bin]# netstat -tunlp | grep 9000

[hadoop@node0 bin]# netstat -tunlp | grep 9001

//查看集群状态

<http://192.168.0.59:50070/dfshealth.jsp>

//查看Map/Reduce状态

<http://192.168.0.59:50030/jobtracker.jsp>

### 生成hadoop-eclipse-plugin

1. **下载稳定版源码hadoop-1.1.2.tar.gz，解压**
2. **安装ant**
3. **编辑{HADOOP\_HOME}/build.xml**

<propertyname="version"value="1.1.3-SNAPSHOT"/>   大概30行

修改为：

<propertyname="version"value="1.1.2"/>

//注释ivy下载

<!--

<target name="ivy-download" description="To download ivy" unless="offline">

<get src="${ivy\_repo\_url}" dest="${ivy.jar}" usetimestamp="true"/>

</target>

-->

//去除对ivy-download的依赖关系

<target name="ivy-init-antlib" depends="ivy-init-dirs,ivy-probe-antlib" unless="ivy.found">

1. **编辑{HADOOP\_HOME}/src/contrib/build-contrib.xml**

<property name="eclipse.home" location="D:/eclipse"/>

<property name="version" value="1.1.2"/>

<property name="name" value="${ant.project.name}"/>

<property name="root" value="${basedir}"/>

<property name="hadoop.root" location="${root}/../../../"/>

1. **编辑${HADOOP\_HOME}/src/contrib/eclipse-plugin/build.xml**

<pathelement location="${hadoop.root}/build/classes"/>

<pathelement location="${hadoop.root}/hadoop-core-1.1.2.jar"/> //增加

<target name="jar" depends="compile" unless="skip.contrib">

<mkdir dir="${build.dir}/lib"/>

<copy file="${hadoop.root}/hadoop-core-${version}.jar" tofile="${build.dir}/lib/hadoop-core.jar" verbose="true"/>

<copy file="${hadoop.root}/lib/commons-cli-${commons-cli.version}.jar" todir="${build.dir}/lib" verbose="true"/>

<copy file="${hadoop.root}/lib/commons-configuration-1.6.jar" todir="${build.dir}/lib" verbose="true"/>

<copy file="${hadoop.root}/lib/commons-httpclient-3.0.1.jar" todir="${build.dir}/lib" verbose="true"/>

<copy file="${hadoop.root}/lib/commons-lang-2.4.jar" todir="${build.dir}/lib" verbose="true"/>

<copy file="${hadoop.root}/lib/jackson-core-asl-1.8.8.jar" todir="${build.dir}/lib" verbose="true"/>

<copy file="${hadoop.root}/lib/jackson-mapper-asl-1.8.8.jar" todir="${build.dir}/lib" verbose="true"/>

<jar

1. **编辑${HADOOP\_HOME}/src/contrib/eclipse-plugin/META-INF**

Bundle-ClassPath: classes/,

lib/hadoop-core.jar,

lib/commons-cli-1.2.jar,

lib/commons-configuration-1.6.jar,

lib/commons-httpclient-3.0.1.jar,

lib/commons-lang-2.4.jar,

lib/jackson-core-asl-1.8.8.jar,

lib/jackson-mapper-asl-1.8.8.jar

1. **Ant编译**

打开cmd，cd {HADOOP\_HOME}\src\contrib\eclipse-plugin\

输入ant，自动搜索当前目录下的build.xml

在{HADOOP\_HOME}\build\contrib\eclipse-plugin生成hadoop-eclipse-plugin-1.1.2.jar

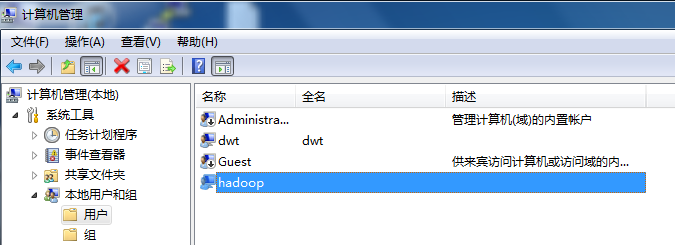
### 配置eclipse

在Linux下安装Eclipse，以hadoop用户启动eclipse，其连接执行都没问题

在Window下安装Eclipse，首先将当前用户名修改为启动Hadoop的用户名，Eclipse版本要选3.5的，太新不行

1. **修改Window用户名**

计算机右键->管理->本地用户和组，将当前用户名称改为hadoop，重启

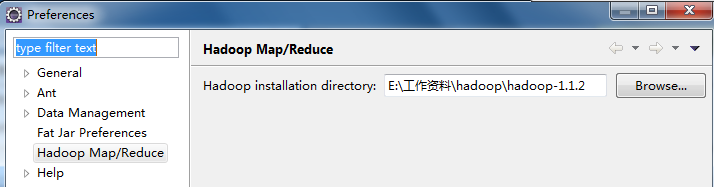


1. **配置Hadoop安装路径**

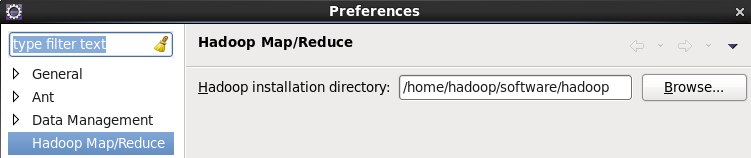
将生成的hadoop-eclipse-plugin-1.1.2.jar拷贝到eclipse的plugins目录下，重启eclipse

Window->Preferences->Hadoop Map/Reduce，选择hadoop安装目录，此处为创建map reduce 工程时导入hadoop jar包的路径，与集群中的hadoop安装目录无关

Window下（将hadoop安装包解压到某一目录即可）



Linux下（hadoop安装目录）



1. **定义Hadoop Location**

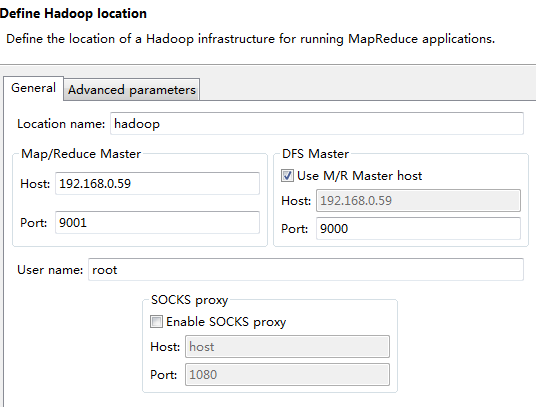
Window->Open Perspective->Map Reduce，打开Map Reduce透视图

选择下面的Map/Reduce Locations视图，右键New Hadoop Location

Map/Reduce Master为mapred.job.tracker中配置的jobtracker IP和端口；

DFS Master为fs.default.name中配置的NameData IP和端口

User Name为hadoop，与集群中启动hadoop的用户一致，且与HDFS的所有者一致



1. **设置Advanced parameters**

dfs.replication改为1，与hdfs-site.xml保持一致

hadoop.tmp.dir:默认是/tmp/hadoop-{user.name}，与core-site.xml 里hadoop.tmp.dir的设置保持一致，此处为/tmp/hadoop-hadoop

eclipse.plug-in.usr.name与安装运行hadoop的用户一致，用hadoop

**问题1：window的eclipse连不上虚拟机的hadoop**

Hadoop的配置文件用192.168.X.X这样去配置地址，不能使用localhost或者127.0.0.1.

[hadoop@node0 conf]# vi core-site.xml

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs:// 192.168.0.59:9000</value> //此处不能为localhost

</property>

</configuration>

[hadoop@node0 conf]# vi mapred-site.xml

<configuration>

<property>

<name>mapred.job.tracker</name>

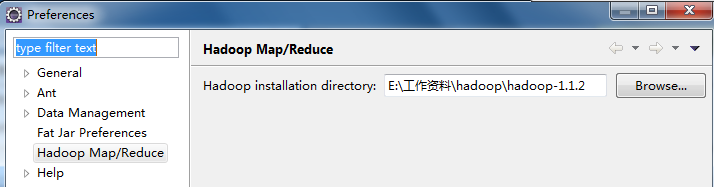
<value>192.168.0.59:9001</value>//此处不能为localhost

</property>

</configuration>

**问题2：新建hadoop工程时，Invalid Hadoop Runtime specified; please click 'Configure Hadoop install directory' or fill in library location input field**

Window->Preferences，设置hadoop安装目录

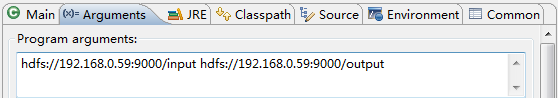


### 新建hadoop工程

1. **新建工程，Map Reduce Project**
2. **编写WordCount程序**
3. **设置参数**

首先Run on Hadoop，生成运行配置

然后Run As->Run Configuration，选中WordCount，在右边的参数设置页面，设置输入输出路径，输出路径必须保证在HDFS不存在该目录



1. **导入输入数据**

有2种方式：

方法1：Linux中新建/删除目录，上传数据到HDFS

[hadoop@node0 bin]# hadoop fs -mkdir /input

[hadoop@node0 bin]# hadoop fs -ls /

[hadoop@node0 bin]# hadoop fs -rmr /input

[hadoop@node0 bin]# vi input.txt

hello world

hello hadoop

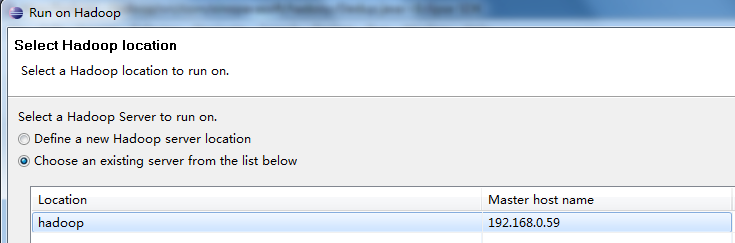
hello mapreduce

[hadoop@node0 bin]# hadoop fs -put input.txt /input/

方法2：eclipse中右键单击目录，选择Create new directory，Upload Files to DFS…

1. **执行程序**

Run on hadoop，选择Hadoop集群



**问题1：org.apache.hadoop.security.AccessControlException:Permission denied:user=Administrator,access=WRITE,inode="tmp":root:supergroup:rwxr-xr-x**

问题原因：本地用户administrator（本机windows用户）想要远程操作hadoop系统，没有权限引起的。

      解决办法：

1. 如果是测试环境，可以取消hadoop hdfs的用户权限检查。打开conf/hdfs-site.xml，找到dfs.permissions属性修改为false（默认为true）OK了。
2. 修改hadoop location参数，在advanced parameter选项卡中，找到hadoop.job.ugi项，将此项改为启动hadoop的用户名即可。（注意第一次设置的时候可能没有hadoop.job.ugi参数，报错后在去看就有了。）
3. 因为Eclipse使用hadoop插件提交作业时，会默认以 DrWho 身份去将作业写入hdfs文件系统中，对应的也就是 HDFS 上的/user/hadoop ,  由于 DrWho 用户对hadoop目录并没有写入权限，所以导致异常的发生。解决方法为：放开 hadoop 目录的权限 ， 命令如下 ：$ hadoop fs -chmod 777 /

**问题2：执行报错，ERROR security.UserGroupInformation: PriviledgedActionException**

Configuration conf = new Configuration(); //设置jobtrackerIP和端口

conf.set("mapred.job.tracker", "192.168.0.59:9001");

**问题3：java.io.IOException: Failed to set permissions of path: file:/tmp/hadoop-myname/mapred/staging**

[hadoop@node0 bin]# hadoop fs -chmod -R 777 /tmp

**问题4：/tmp/hadoop-hadoop/mapred/staging/hadoop/staging must be rwx---**

该用户的目录所有者必须为hadoop，权限必须为700

[hadoop@node0 bin]# hadoop fs -chmod -R 700 /tmp/hadoop-hadoop/mapred/staging/hadoop/

**问题5: Exception: org.apache.hadoop.ipc.RemoteException: java.io.IOException: File /input/input.txt could only be replicated to 0 nodes, instead of 1**

Node1的防火墙没关,所有节点的防火墙

**问题6：java.lang.RuntimeException: java.lang.ClassNotFoundException: org.apache.hadoop.examples.WordCount$TokenizerMapper**

Eclipse版本过高，采用3.5

### 重建HDFS

[hadoop@node0 bin]# ./stop-all.sh

[hadoop@node0 bin]# rm -fr /usr/hadoop/tmp/\*

[hadoop@node0 bin]# rm -fr /tmp/\*hadoop\*

[hadoop@node0 bin]# hadoop namenode -format

Hadoop配置文件分为两类：

1. 只读默认配置文件，src/core/core-default.xml, src/hdfs/hdfs-default.xml 和 src/mapred/mapred-default.xml；
2. 指定配置文件conf/core-site.xml, conf/hdfs-site.xml and conf/mapred-site.xml

Hadoop daemons 分为 NameNode/DataNode 和 JobTracker/TaskTracker。

**问题1：Name node is in safe mode**

解除安全模式

[hadoop@node0 software]$ hadoop dfsadmin -safemode leave

# Hive

Hive是基于MapReduce的数据库，提供类SQL的HiveQL语言。

## 安装

[hadoop@node0 software]$ tar zxvf hive-0.9.0.tar.gz

[hadoop@node0 software]$ mv hive-0.9.0 hive

1. **增加Hive环境变量**

[hadoop@node0 software]$ vi ~/.bash\_profile

PATH=$PATH:$HOME/bin

export JAVA\_HOME=/usr/java/jdk1.7.0\_21

export CLASSPATH=$JAVA\_HOME/lib:$CLASSPATH

export HIVE\_HOME=/home/hadoop/software/hive

PATH=$PATH:$JAVA\_HOME/bin: /home/hadoop/software/hadoop/bin:$HIVE\_HOME/bin

export PATH

[hadoop@node0 software]$ source ~/.bash\_profile

1. **创建/tmp和/user/hive/warehouse，具有组可写权限**

[hadoop@node0 software]$ hadoop fs -mkdir /tmp

[hadoop@node0 software]$ hadoop fs -mkdir /user/hive/warehouse

[hadoop@node0 software]$ hadoop fs -chmod g+w /tmp

[hadoop@node0 software]$ hadoop fs -chmod g+w /user/hive/warehouse

1. **配置Hive**

[hadoop@node0 software]$ cd hive/conf

[hadoop@node0 conf]$ cp hive-default.xml.template hive-site.xml

[hadoop@node0 conf]$ cp hive-log4j.properties.template hive-log4j.properties

[hadoop@node0 conf]$ vi hive-log4j.properties

#log4j.appender.EventCounter=org.apache.hadoop.metrics.jvm.EventCounter

log4j.appender.EventCounter=org.apache.hadoop.log.metrics.EventCounter

否则会提示WARNING: org.apache.hadoop.metrics.jvm.EventCounter is deprecated. Please use org.apache.hadoop.log.metrics.EventCounter in all the log4j.properties files.

1. **启动运行**

首先启动hadoop

1. 通过CLI交互

[hadoop@node0 conf]$ hive

hive> exit;

1. 通过脚本交互

[hadoop@node0 conf]$ hive –f script

1. 通过命令交互

[hadoop@node0 software]$ echo 'X' > /home/hadoop/software/dummy.txt

[hadoop@node0 software]$ hive -e "CREATE TABLE dummy(value STRING);LOAD DATA LOCAL INPATH '/home/hadoop/software/dummy.txt' OVERWRITE INTO TABLE dummy"

[hadoop@node0 conf]$ hive -e 'select \* from dummy'

1. 通过Web UI网络交互

[hadoop@node0 conf]$ hive --service hwi

<http://192.168.0.59:9999/hwi/>

1. 通过JDBC交互

String driverName = “org.apache.hadoop.hive.jdbc.HiveDriver”;

Class.forName(driverName);

Connection con = DriverManager.getConnection(“jdbc:hive://localhost:10000/default”,””,””);

Statement stmt = con.createStatement();

stmt.executeQuery(sql);

## 操作

### DDL

hive> CREATE TABLE pokes (foo INT, bar STRING);

hive> CREATE TABLE invites (foo INT, bar STRING) PARTITIONED BY (ds STRING);

hive> SHOW TABLES;

hive> SHOW TABLES '.\*s'; //显示以s结尾的表名

hive> DESCRIBE invites;

hive> ALTER TABLE pokes ADD COLUMNS (new\_col INT);

hive> ALTER TABLE invites ADD COLUMNS (new\_col2 INT COMMENT 'a comment');

hive> ALTER TABLE pokes RENAME TO 3koobecaf;

hive> DROP TABLE 3koobecaf;

hive> DROP TABLE invites;

### DML

hive> CREATE TABLE pokes (foo INT, bar STRING);

hive> CREATE TABLE invites (foo INT, bar STRING) PARTITIONED BY (ds STRING);

//从本地导入数据，OVERWRITE覆盖表内容

hive> LOAD DATA LOCAL INPATH '/home/hadoop/software/hive/examples/files/kv1.txt' OVERWRITE INTO TABLE pokes;

hive> LOAD DATA LOCAL INPATH '/home/hadoop/software/hive/examples/files/kv2.txt' OVERWRITE INTO TABLE invites PARTITION (ds='2008-08-15');

hive> LOAD DATA LOCAL INPATH '/home/hadoop/software/hive/examples/files/kv3.txt' OVERWRITE INTO TABLE invites PARTITION (ds='2008-08-08');

//从HDFS导入数据，没有LOCAL关键字

hive> LOAD DATA INPATH '/user/myname/kv2.txt' OVERWRITE INTO TABLE invites PARTITION (ds='2008-08-15');

hive> SELECT a.foo FROM invites a WHERE a.ds='2008-08-15';

//将invites表中所有ds='2008-08-15'的行查询写入/tmp/hdfs\_out文件

hive> INSERT OVERWRITE DIRECTORY '/tmp/hdfs\_out' SELECT a.\* FROM invites a WHERE a.ds='2008-08-15';

//写入到本地

hive> INSERT OVERWRITE LOCAL DIRECTORY '/tmp/local\_out' SELECT a.\* FROM pokes a;

hive> INSERT OVERWRITE TABLE events SELECT a.\* FROM profiles a;

hive> INSERT OVERWRITE TABLE events SELECT a.\* FROM profiles a WHERE a.key < 100;

hive> INSERT OVERWRITE LOCAL DIRECTORY '/tmp/reg\_3' SELECT a.\* FROM events a;

hive> INSERT OVERWRITE DIRECTORY '/tmp/reg\_4' select a.invites, a.pokes FROM profiles a;

hive> INSERT OVERWRITE DIRECTORY '/tmp/reg\_5' SELECT COUNT(\*) FROM invites a WHERE a.ds='2008-08-15';

hive> INSERT OVERWRITE DIRECTORY '/tmp/reg\_5' SELECT a.foo, a.bar FROM invites a;

hive> INSERT OVERWRITE LOCAL DIRECTORY '/tmp/sum' SELECT SUM(a.pc) FROM pc1 a;

hive> FROM invites a INSERT OVERWRITE TABLE events SELECT a.bar, count(\*) WHERE a.foo > 0 GROUP BY a.bar;

hive> INSERT OVERWRITE TABLE events SELECT a.bar, count(\*) FROM invites a WHERE a.foo > 0 GROUP BY a.bar;

hive> FROM pokes t1 JOIN invites t2 ON (t1.bar = t2.bar) INSERT OVERWRITE TABLE events SELECT t1.bar, t1.foo, t2.foo;

FROM src

INSERT OVERWRITE TABLE dest1 SELECT src.\* WHERE src.key < 100

INSERT OVERWRITE TABLE dest2 SELECT src.key, src.value WHERE src.key >= 100 and src.key < 200

INSERT OVERWRITE TABLE dest3 PARTITION(ds='2008-04-08', hr='12') SELECT src.key WHERE src.key >= 200 and src.key < 300

INSERT OVERWRITE LOCAL DIRECTORY '/tmp/dest4.out' SELECT src.value WHERE src.key >= 300;

hive> FROM invites a INSERT OVERWRITE TABLE events SELECT TRANSFORM(a.foo, a.bar) AS (oof, rab) USING '/bin/cat' WHERE a.ds > '2008-08-09';

# HBase

## 安装

Hadoop与HBase的版本对应表：

|  | HBase-0.92.x | HBase-0.94.x | HBase-0.96 |
| --- | --- | --- | --- |
| Hadoop-0.20.205 | S | X | X |
| Hadoop-0.22.x | S | X | X |
| Hadoop-1.0.x | S | S | S |
| Hadoop-1.1.x | NT | S | S |
| Hadoop-0.23.x | X | S | NT |
| Hadoop-2.x | X | S | S |

[hadoop@node0 software]$ tar zxvf hbase-0.94.7.tar.gz

[hadoop@node0 software]$ mv hbase-0.94.7 hbase

[hadoop@node0 software]$ cd hbase/conf

1. **hbase-site.xml**

[hadoop@node0 conf]$ vi hbase-site.xml

<configuration>

<property>

<name>hbase.rootdir</name>

<value>hdfs://192.168.0.59:9000/hbase</value>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<property>

<name>hbase.zookeeper.quorum</name>

<value>node0,node1</value>

</property>

<property>

<name>hbase.zookeeper.property.dataDir</name>

<value>/export/zookeeper</value>

</property>

</configuration>

将hbase.rootdir替换成你期望写文件的目录，默认指向 /tmp/hbase-${user.name} ，重启的时候操作系统会清理/tmp目录，即丢失数据

hbase.zookeeper.quorum为ZooKeeper运行的节点，数据保存在/export/zookeeper

1. **hbase-env.sh**

[hadoop@node0 conf]$ vi hbase-env.sh

export JAVA\_HOME=/usr/java/jdk1.7.0\_21

1. **regionservers**

[hadoop@node0 conf]$ vi regionservers

node0

node1

//其他各节点都按以上步骤部署

[hadoop@node0 conf]$ cd ../bin

[hadoop@node0 bin]$ ./start-hbase.sh

<http://192.168.0.59:60010>查看HBase的属性

//shell交互

[hadoop@node0 bin]$ ./hbase shell

//注意的是表名，行和列需要加引号

hbase(main):002:0> create 'test','cf'

hbase(main):004:0> list 'table'

//插入3行，第一个行key为row1, 列为 cf:a， 值是 value1。HBase中的列是由列族前缀和列的名字组成的，以冒号间隔。

hbase(main):005:0> put 'test','row1','cf:a','value1'

hbase(main):006:0> put 'test','row2','cf:b','value2'

hbase(main):007:0> put 'test','row3','cf:c','value3'

hbase(main):008:0> scan 'test'

hbase(main):009:0> get 'test', 'row1'

hbase(main):010:0> disable 'test'

hbase(main):011:0> drop 'test'

hbase(main):013:0> exit

[hadoop@node0 bin]$ ./stop-hbase.sh

<http://abloz.com/hbase/book.html>

# Zookeeper

为分布式应用提供协调服务

|  |
| --- |
|  |
| http://zookeeper.apache.org/doc/trunk/images/zkservice.jpg |

Server：

Client：TCP连接一个Server，发送和接收请求，心跳测试，如果Server宕，则连接到另一个