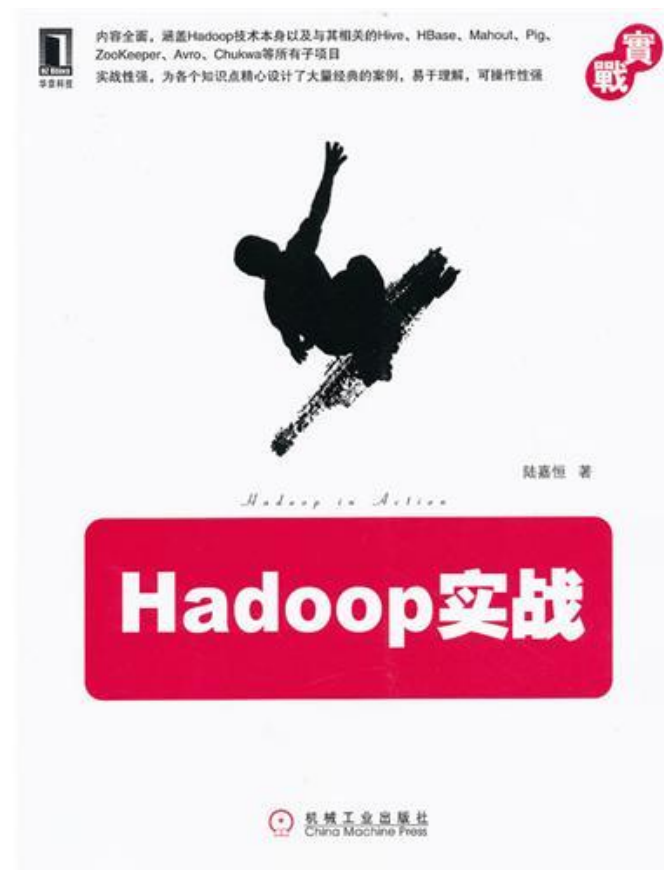
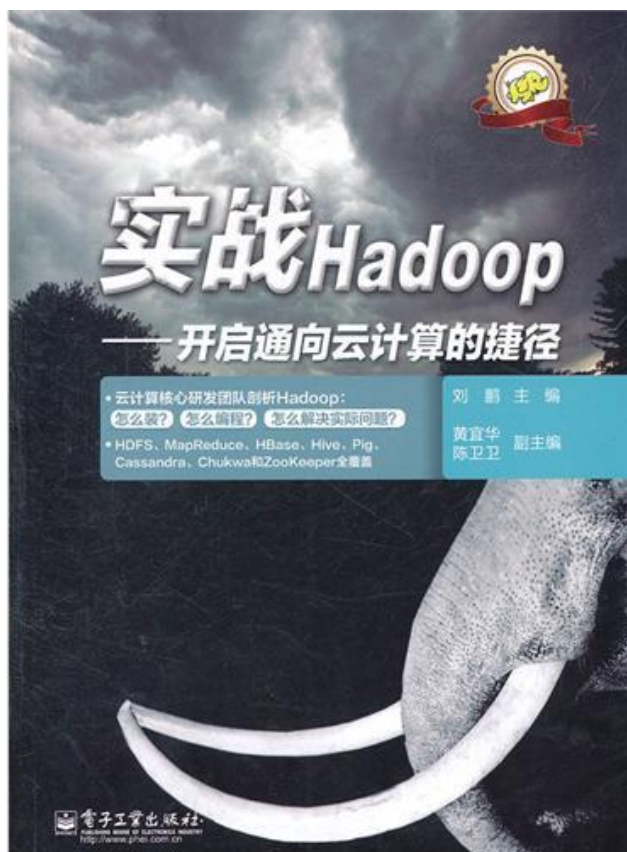


Hadoop数据分析平台 第2周

2012.8.25



2012.8.25



2012.8.25

准备与配置安装环境

- 安装虚拟机和linux，虚拟机推荐使用vmware，PC可以使用workstation，服务器可以使用ESXi，在管理上比较方便。ESXi还可以通过拷贝镜像文件复制虚拟机，复制后自动修改网卡号和ip，非常快捷。如果只是实验用途，硬盘大约预留20-30G空间。
- 以Centos为例，分区可以选择默认，安装选项选择Desktop Gnome，以及Server、Server GUI即可。其它Linux，注意选项里应包括ssh，vi（用于编辑配置文件），perl等（有些脚本里包含perl代码需要解析）
- 到Oracle官网下载java jdk安装包，并且进行安装

三种运行模式

- 单机模式：安装简单，几乎不用作任何配置，但仅限于调试用途
- 伪分布模式：在单节点上同时启动namenode、datanode、jobtracker、tasktracker、secondary namenode等5个进程，模拟分布式运行的各个节点
- 完全分布式模式：正常的Hadoop集群，由多个各司其职的节点构成

伪分布式模式的安装和配置步骤

- 下载并解压Hadoop安装包，为了和教材一致，选用了0.20.2版本
- 进入Hadoop的解压目录，编辑conf/hadoop-env.sh文件（注意0.23版后配置文件的位置有所变化）
- 编辑conf目录下core-site.xml、hdfs-site.xml和mapred-site.xml三个核心配置文件
- 配置ssh，生成密钥，使到ssh可以免密码连接localhost
- 格式化HDFS
- 使用bin/start-all.sh启动Hadoop
- 使用bin/stop-all.sh关闭Hadoop

下载Hadoop



Index of /apache/hadoop/common/hadoop-

Icon	Name	Last modified	Size	Description
[DIR]	Parent Directory		-	
[DIR]	hadoop-0.20.2/	02-Mar-2010 03:03	-	
[]	hadoop-0.20.2.tar.gz	19-Feb-2010 16:28	43M	
[TXT]	hadoop-0.20.2.tar.gz.asc	19-Feb-2010 16:28	487	

Apache/2.2.22 (Ubuntu) Server at mirror.bit.edu.cn Port 81

解压hadoop

```
[root@h1 usr]#  
[root@h1 usr]# ls /root  
anaconda-ks.cfg          hadoop-0.20.2.tar.gz    hbasedata  
Desktop                  hbase-0.90.5           install.log  
hadoop-0.20.203.0rc1.tar.gz  hbase-0.90.5.tar.gz    install.log.syslog  
[root@h1 usr]# tar xzvf /root/hadoop-0.20.2
```


Hadoop配置有关文件

表 9-1: Hadoop 配置文件

文件名	格式	描述
hadoop-env.sh	bash 脚本	在运行 Hadoop 的脚本中使用的环境变量
core-site.xml	Hadoop 配置 XML	Hadoop 核心 [®] 的配置，例如 HDFS 和 MapReduce 中很普遍的 I/O 设置
hdfs-site.xml	Hadoop 配置 XML	HDFS 后台程序设置的配置：名称节点，第二名称节点和数据节点
mapred-site.xml	Hadoop 配置 XML	MapReduce 后台程序设置的配置：jobtracker 和 tasktracker
masters	纯文本	记录运行第二名称节点的机器(一行一个)的列表
slaves	纯文本	记录运行数据节点和 tasktracker 的机器(一行一个)的列表
hadoop-metrics.properties	Java 属性	控制 Hadoop 怎么发布 metrics(参见第 10 章)的属性
log4j.properties	Java 属性	系统日志文件的属性、名称节点审计日记和 tasktracker 子进程(参见第 5 章)的日志的属性

修改hadoop-env.sh文件

```
[root@h1 conf]# pwd
/usr/hadoop-0.20.2/conf
[root@h1 conf]# ls
capacity-scheduler.xml  hadoop-env.sh          hdfs-site.xml      masters
configuration.xml      hadoop-metrics.properties  log4j.properties  slaves
core-site.xml          hadoop-policy.xml          mapred-site.xml    ssl-client.xml.example
[root@h1 conf]#
```

```
# Set HADOOP_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use. Required.
export JAVA_HOME=/usr/java/jdk1.6.0_26

# Extra Java CLASSPATH elements. Optional.
# export HADOOP_CLASSPATH=

# The maximum amount of heap to use, in MB. Default is 1000.
# export HADOOP_HEAPSIZE=2000

# Extra Java runtime options. Empty by default.
# export HADOOP_OPTS=-server
```

修改hadoop-env.sh文件

- 详细设置参见《权威指南》p269

修改core-site.xml文件

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

<!-- Put site-specific property overrides in this file. -->

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

core-site.xml中常见配置参数

- fs.default.name NameNode的IP地址和端口

修改hdfs-site.xml文件

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

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>dfs.data.dir</name>
<value>/usr/hadoop-0.20.2/data</value>
</property>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

hdfs-site.xml文件中常用配置参数

dfs.name.dir	用逗号隔开的目录名	<code>\${hadoop.tmp.dir}/dfs/name</code>	存储名称节点永久元数据的目录的列表。名称节点在列表中的每一个目录下存储着元数据的副本
dfs.data.dir	用逗号隔开的目录名	<code>\${hadoop.tmp.dir}/dfs/name</code>	数据节点存储块的目录的列表
fs.checkpoint.dir	用逗号隔开的目录名	<code>\${hadoop.tmp.dir}/dfs/namesecondary</code>	第二名称节点用来存储检查点的目录的列表。它在列表的每一个目录下存储着检查点的副本

注意：注意，HDFS 的存储目录默认在 Hadoop 的临时目录下(hadoop.tmp.dir 属性，其默认目录是/tmp/hadoop-\${user.name})。所以这些属性的设置很重要，可保证数据在系统清空临时目录时不会丢失。

修改mapred-site.xml文件

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

<!-- Put site-specific property overrides in this file. -->

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


mapred-site.xml文件中常用配置参数

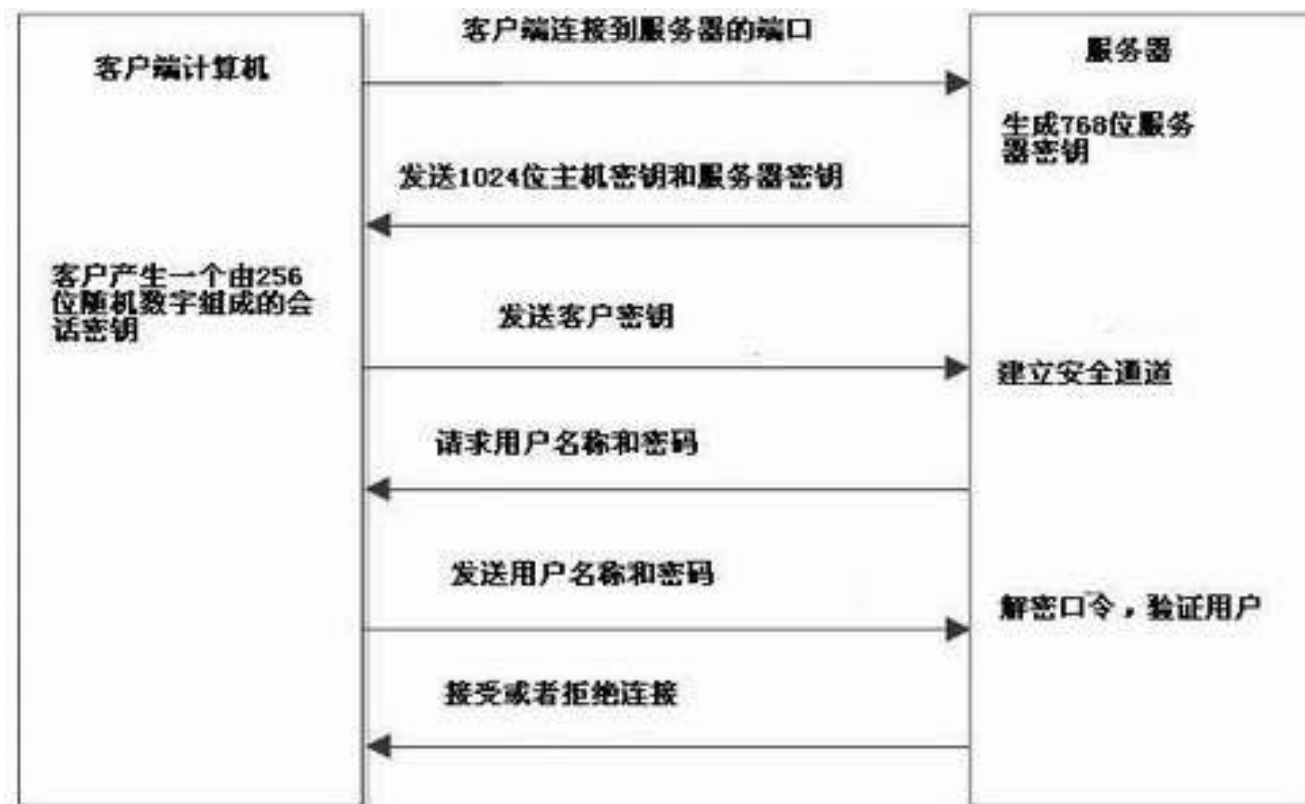
mapred.job.tracker	主机名和 local 端口		jobtracker 的 RPC 服务器运行的主机名和端口。如果设置为默认值 local, 则当运行一道作业时, jobtracker 也在同一进程内(此时, 不必启动 MapReduce 守护进程)
mapred.local.dir	用逗号分隔的目录 /mapred/local 名称		MapReduce 存储作业中间数据的目录列表。当作业结束时数据会被清空
mapred.system.dir	URI	<code>\${hadoop.tmp.dir}/mapred/system</code>	当一道作业运行时, 与存储共享文件的 fs.default.name 相关的目录
mapred.tasktracker.map.tasks.maximum	int	2	任一时刻 tasktracker 上运行 map 任务的数量
mapred.tasktracker.reduce.tasks.maximum	int	2	任一时刻在 tasktracker 上运行的 reduce 任务的数量
Mapred.child.java.opts	String	-Xmx200m	用来启动 tasktracker 子进程, 运行 map 和 reduce 任务的 JVM 选项。此属性可以在每道作业上设置, 这对为 debug 设置 JVM 属性比较有用

2012.8.25

生成ssh密钥对

```
[root@h1 ~]#  
[root@h1 ~]#  
[root@h1 ~]# cd /root  
[root@h1 ~]# ssh-keygen -t rsa  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa):  
/root/.ssh/id_rsa already exists.  
Overwrite (y/n)?  
[root@h1 ~]# cd .ssh  
[root@h1 .ssh]# ls  
authorized_keys  id_dsa  id_dsa.pub  id_rsa  id_rsa.pub  known_hosts  
[root@h1 .ssh]# cp id_dsa.pub authorized_keys  
cp: overwrite `authorized_keys'? y  
[root@h1 .ssh]#
```

- 客户端向服务器端发出连接请求
- 服务器端向客户端发出自己的公钥
- 客户端使用服务器端的公钥加密通讯密钥然后发给服务器端
- 如果通讯过程被截获，由于窃听者即使获知公钥和经过公钥加密的内容，但不拥有私钥依然无法解密（RSA算法）
- 服务器端接收到密文后，用私钥解密，获知通讯密钥
- `ssh-keygen`命令给服务器端产生公私钥密钥对，`cp`命令将服务器端公钥复制到客户端（注意在伪分布模式下服务器端和客户端是同一台机器），因此客户端本身就拥有了服务器端公钥，可以直接进行免密码接入



完全分布式模式的安装和配置

- 配置hosts文件
- 建立hadoop运行账号
- 配置ssh免密码连入
- 下载并解压hadoop安装包
- 配置namenode , 修改site文件
- 配置hadoop-env.sh
- 配置masters和slaves文件
- 向各节点复制hadoop
- 格式化namenode
- 启动hadoop
- 用jps检验各后台进程是否成功启动

- 所有的节点都修改/etc/hosts，使彼此之间都能把主机名解析为ip

```
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
::1               localhost6.localdomain6 localhost6
192.168.1.102      h1
192.168.1.103      h2
192.168.1.104      h3
192.168.1.163      dog
192.168.1.162      cat
192.168.1.161      gangster
~
~
```

建立专门的运行hadoop的用户

- 在所有的节点上都建立运行hadoop的专用用户grid

```
[root@h1 ~]# passwd grid
Changing password for user grid.
New UNIX password:
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@h1 ~]#
[root@h1 ~]#
```

- 注意要以grid用户登录，在grid用户的主目录下进行操作！
- 每个节点作相同操作

```
[grid@h1 ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/grid/.ssh/id_rsa):
Created directory '/home/grid/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/grid/.ssh/id_rsa.
Your public key has been saved in /home/grid/.ssh/id_rsa.pub.
The key fingerprint is:
d0:60:e2:ee:88:23:f5:bc:9f:99:9c:a9:a5:f9:eb:9b grid@h1
```

```
[grid@h1 .ssh]$ cp id_rsa.pub authorized_keys
```


- 把各个节点的authorized_keys的内容互相拷贝加入到对方的此文件中，然后就可以免密码彼此ssh连入

```
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA5QJE6021TCqmmrGoKECh2gPksQTqZ1RB3IR9B
gQXsi4BFeqHzLb1xDp+g156eBTsvyR/ZbPHAnVDNQ/ftVPP9yr0S5ZsY4QRnFx58p4bJ/Rjky
fDNmiu8HtVHp162w/YveXJ/szCJgrlykLc6a7hWLtWMMskv0Fa1RxKDSDIoSjTiyLjN6sQy51
G5w== grid@h2
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApDPNyrf1VbmwYc2Wdq7XJXw5xpD9vJ8gFdX0r
p3iuvPi19sruNgEcudfwCDK3+DMcTyogYu37tPMm2XzYA6MEqj0apRVQLWLBk0tzP6Piznhc2
yPmbziYygDdw/z/20D0301NvXjJtX69gaWJNcp2+k1pxYVZ9GsXuBN87eLKzj/8LNT9/aimqh
Enw== grid@h1
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA0IgnEQ410k2IVVPWtQRUVy18++5iLX5gugFc1
sFCNNsID4ccxikFbE50shIoe26xBLyk3Ju48YXsG9mVaLeY8K30wxwbwtqKo8jwr4KKP0JeHF
Ben++IpoxiXRJBRLG97gt8Q9dx9uUMFcKcR2oNxu56uhyry35GL2bVR2dNLihiSwnbASoqrV
+Zw== grid@h3
~
```

下载hadoop压缩包并解压

```
[grid@h1 ~]$  
[grid@h1 ~]$ su  
Password:  
[root@h1 grid]# cp hadoop-0.20.2.tar.gz /home/grid  
[root@h1 grid]# scp hadoop-0.20.2.tar.gz grid@h2:/home/grid  
[root@h1 grid]# scp hadoop-0.20.2.tar.gz grid@h3:/home/grid
```

在namenode上配置hadoop

- 按照和伪分布模式相同的方法修改core-site.xml , hdfs-site.xml , mapred-site.xml等配置文件
- 修改hadoop-env.sh文件
- 修改masters和slaves文件 , 记录集群中各个节点
- 向各节点分发hadoop
- 格式化namenode
- 在namenode上启动守护进程 , 系统会自动到各个节点启动相应的进程

修改core-site.xml、hdfs-site.xml以及mapred-site.xml (注意修改localhost)

```
[root@h1 conf]# vi core-site.xml
```

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

<!-- Put site-specific property overrides in this file. -->

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

```
[root@h1 conf]# vi mapred-site.xml
```

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

<!-- Put site-specific property overrides in this file. -->

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

```
[root@h1 conf]# vi hdfs-site.xml
```

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

<!-- Put site-specific property overrides in this file. -->

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

修改masters和slaves文件

```
[root@h1 conf]# ls
capacity-scheduler.xml  hadoop-env.sh          hdfs-site.xml          masters
configuration.xsl       hadoop-metrics.properties log4j.properties       slaves
core-site.xml           hadoop-policy.xml        mapred-site.xml         ssl-client.xml.example
[root@h1 conf]# vi masters
```

```
h1
~
~
~
~
```

```
[root@h1 conf]# vi slaves
```

```
h2
h3
~
```

修改hadoop-env.sh

```
[root@h1 conf]# ls /usr/java
jdk1.6.0_26
[root@h1 conf]# vi hadoop-env.sh

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use. Required.
export JAVA_HOME=/usr/java/jdk1.6.0_26

# Extra Java CLASSPATH elements. Optional.
# export HADOOP_CLASSPATH=

# The maximum amount of heap to use, in MB. Default is 1000.
# export HADOOP_HEAPSIZE=2000
```

向各节点复制hadoop

```
[grid@h1 ~]$ scp -r ./hadoop-0.20.2 h2:/home/grid  
[grid@h1 ~]$ scp -r ./hadoop-0.20.2 h3:/home/grid  
[grid@h1 ~]$  
[grid@h1 ~]$
```

格式化分布式文件系统

```
[grid@h1 hadoop-0.20.21$ bin/hadoop namenode -format
02/09/23 14:59:09 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = h1/192.168.1.102
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 0.20.2
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.20 -r 911707; compiled by 'chrisdo' o
n Fri Feb 19 08:07:34 UTC 2010
*****/
02/09/23 14:59:09 INFO namenode.FSNamesystem: fsOwner=grid.grid
02/09/23 14:59:09 INFO namenode.FSNamesystem: supergroup=supergroup
02/09/23 14:59:09 INFO namenode.FSNamesystem: isPermissionEnabled=true
02/09/23 14:59:09 INFO common.Storage: Image file of size 94 saved in 0 seconds.
02/09/23 14:59:09 INFO common.Storage: Storage directory /tmp/hadoop-grid/dfs/name has been successfully formatted.
02/09/23 14:59:09 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at h1/192.168.1.102
*****/
[grid@h1 hadoop-0.20.21$
```



```
[grid@h1 ~]$ cd hadoop-0.20.2
[grid@h1 hadoop-0.20.2]$ bin/start-all.sh
starting namenode, logging to /home/grid/hadoop-0.20.2/bin/../logs/hadoop-grid-namenode-h1.out
h3: starting datanode, logging to /home/grid/hadoop-0.20.2/bin/../logs/hadoop-grid-datanode-h3.out
h2: starting datanode, logging to /home/grid/hadoop-0.20.2/bin/../logs/hadoop-grid-datanode-h2.out
The authenticity of host 'h1 (192.168.1.102)' can't be established.
RSA key fingerprint is 34:c6:fb:7f:48:26:05:ca:9e:69:37:49:74:2a:8b:d0.
Are you sure you want to continue connecting (yes/no)? yes
h1: Warning: Permanently added 'h1,192.168.1.102' (RSA) to the list of known hosts.
h1: starting secondarynamenode, logging to /home/grid/hadoop-0.20.2/bin/../logs/hadoop-grid-secondarynamenode-h1.out
starting jobtracker, logging to /home/grid/hadoop-0.20.2/bin/../logs/hadoop-grid-jobtracker-h1.out
h2: starting tasktracker, logging to /home/grid/hadoop-0.20.2/bin/../logs/hadoop-grid-tasktracker-h2.out
h3: starting tasktracker, logging to /home/grid/hadoop-0.20.2/bin/../logs/hadoop-grid-tasktracker-h3.out
[grid@h1 hadoop-0.20.2]$
```

检测守护进程启动情况

```
[grid@h1 hadoop-0.20.2]$ ls /usr/java/jdk1.6.0_26/bin
appletviewer  idlj          javadoc        jconsole       jmap           jstat
apt           jar           javah          jcontrol       jps            jstatd
ControlPanel  jarsigner     javap          jdb            jrunscript     jvisualvm
extcheck      java          java-rmi.cgi   jhat           jsadebugd      keytool
HtmlConverter javac         javaws         jinfo          jstack         native2ascii

[grid@h1 hadoop-0.20.2]$ ls /usr/java/jdk1.6.0_26/bin/jps
/usr/java/jdk1.6.0_26/bin/jps
[grid@h1 hadoop-0.20.2]$ /usr/java/jdk1.6.0_26/bin/jps
5722 JobTracker
5503 NameNode
5657 SecondaryNameNode
5821 Jps
[grid@h1 hadoop-0.20.2]$
```

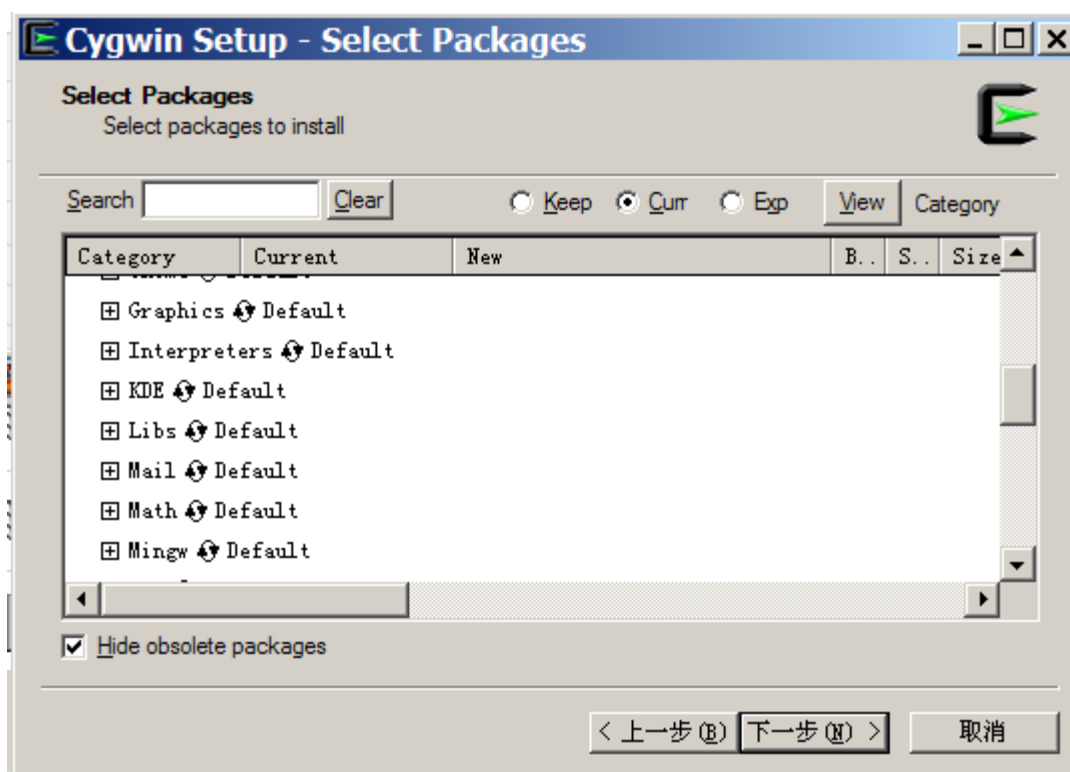
```
[grid@h2 ~]$ /usr/java/jdk1.6.0_26/bin/jps
5437 Jps
5307 DataNode
5383 TaskTracker
[grid@h2 ~]$
```

Windows下也可以装hadoop

- 下载及安装unix仿真工具cygwin
- 安装java
- 修改系统环境变量
- 启动cygwin，在仿真终端下解压hadoop
- 配置openssh
- 后续步骤与linux情形雷同

下载和安装Cygwin

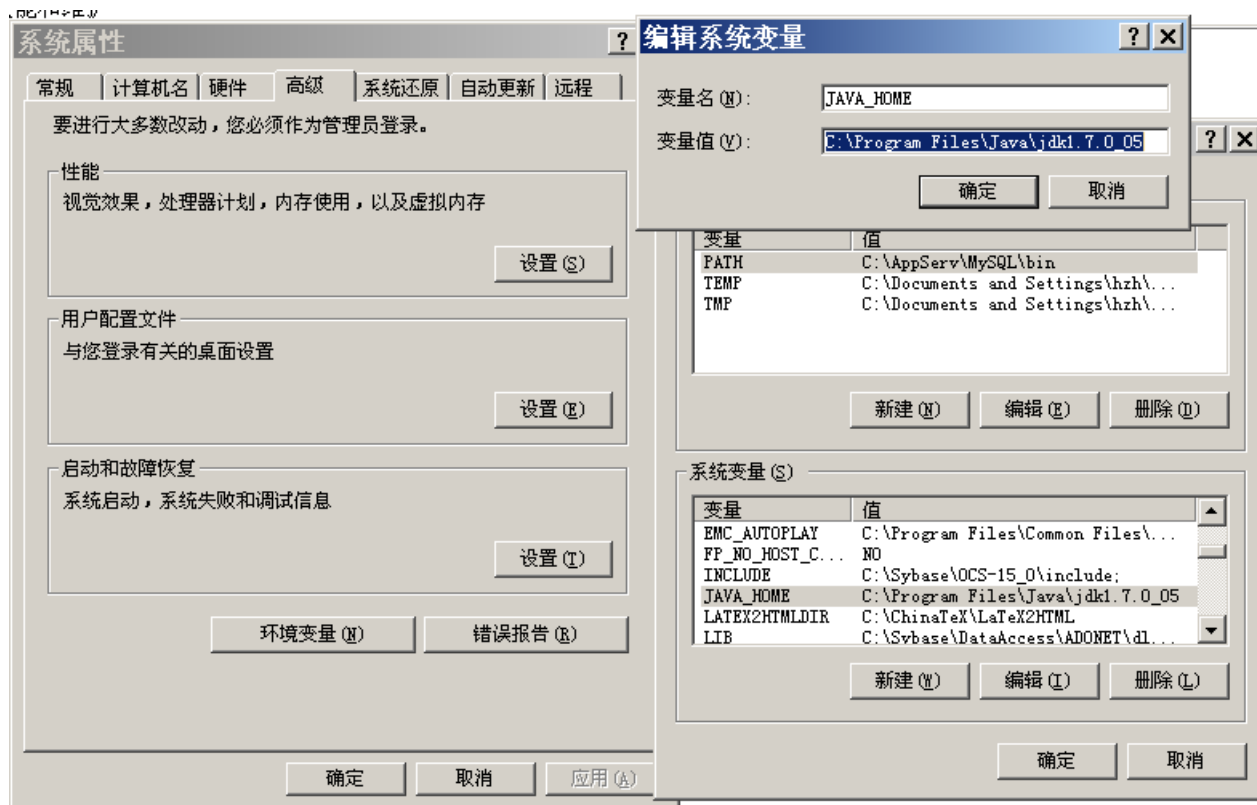
- 安装时一定要选择vi、openssh、perl等



2012.8.25

编辑系统变量

- CYGWIN=ntsec tty
- Path : 加入
c:\cygwin\bin
- 安装java , 并设置
JAVA_HOME和
CLASSPATH变量 , 把
相关bin目录加进
PATH

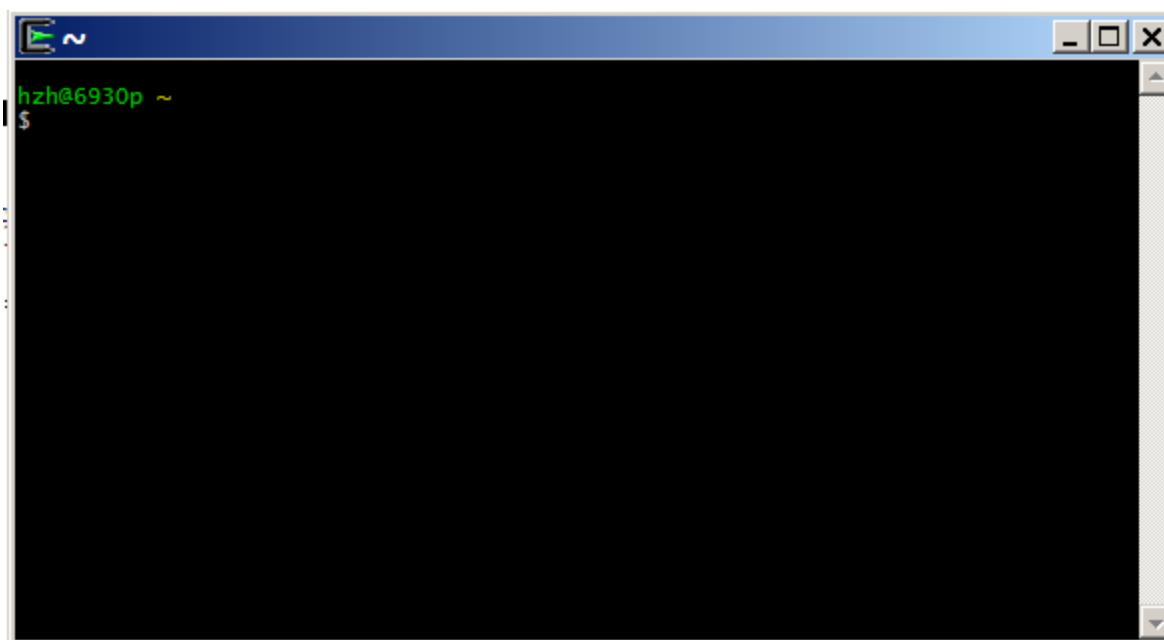


2012.8.25

伪分布模式

- 启动cygwin
- 解压hadoop包，注意映射的写法

```
tar xzvf /cygdrive/d/hadoop-0.20.2.tar.gz
```



伪分布模式

- 修改hadoop-env.sh，注意目录名称中如果有空格时的写法，例如Progra~1
- 与linux一样，修改core-site.xml等几个site文件，写法相同
- 运行ssh-host-config配置openssh，然后net start sshd启动服务
- 类似linux的方法，配置ssh免密码
- 格式化namenode
- 启动hadoop

- 与linux的情况类似



Thanks

FAQ时间