scp -l 285760 -r /home/websearch root@ip:/home/websearch

传输速度大概在40MB/S

-l：表示速度，如果不加速度有可能造成stalled。

似乎也不行。。。

安装websearch的java版本最好在6或者6以下

首先在/home/websearch下建一个nutch-test文件夹

在该文件夹下建目录

dis\_search/安装Nutch目录，是为了检索

search/安装Nutch目录，为了爬取网页

filesystem/这是MapReduce文件系统目录

local/本地索引和段目录

home/这是Nutch用户根目录

tomcat/这是Tomcat安装目录

**安装Nutch和Hadoop**

1、解压Nutch1.2的包

#tar zxvf apache-nutch-1.2-src.tar.gz

2、在apache-nutch-1.2-src文件夹中创建文件 build.properties，定义一个变量dist.dir，表示Nutch安装位置

#nano build.properties

比如：dist.dir=/home/websearch/nutch-test/search

3、在该文件夹中建一个 build 文件夹

#mkdir build

4、确认Java的环境变量JAVA\_HOME路径正确

export JAVA\_HOME=/usr/java/jdk1.6.0\_45

export JRE\_HOME=$JAVA\_HOME/jre

export CLASS\_PATH=$JAVA\_HOME/lib:$JRE\_HOME/lib:$ANT\_HOME/lib:$CALSS\_PATH

export PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin:$ANT\_HOME/bin

export ANT\_HOME=/usr/ant/apache-ant-1.9.3

export HADOOP\_HOME=/home/websearch/nutch-test/search

TOMCAT\_HOME=/home/websearch/nutch-test/tomcat

FABAN\_HOME=/home/websearch/faban

5、在apache-nutch-1.2-src目录中安装Nutch，使用Ant

#ant package

**安装Hadoop**

1、修改环境变量文件/home/websearch/nutch-test/search/conf/hadoop-env.sh中的路径

export HADOOP\_HOME=/home/websearch/nutch-test/search

export JAVA\_HOME=/usr/java/jdk1.6.0\_45

export HADOOP\_LOG\_DIR=${HADOOP\_HOME}/logs

export HADOOP\_SLAVES=${HADOOP\_HOME}/conf/slaves

2、在/home/websearch/nutch-test/search/conf/slaves文件中加入“localhost”

3、在/home/websearch/nutch-test/search/conf/core-site.xml中加入：

<property>  
<name>fs.default.name</name>  
<value>hdfs://localhost:9000</value>  
<description>  
Where to find the Hadoop Filesystem through the network. Note 9000 is not the default port.  
</description>

</property>

4、在/home/websearch/nutch-test/search/conf/hdfs-site.xml中加入：

< ?xml version="1.0"?>  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
<!--Put site-specific property overrides in this file. -->  
<configuration>  
<property>  
<name>dfs.name.dir</name>  
<value>/home/websearch/nutch-test/filesystem/name</value>  
</property>  
<property>  
<name>dfs.data.dir</name>  
<value>/home/websearch/nutch-test/filesystem/data</value>  
</property>  
<property>  
<name>dfs.replication</name>  
<value>1</value>  
</property>  
</configuration>

5、先检查/home/websearch/nutch-test/filesystem/中有没有mapreduce

在/home/websearch/nutch-test/search/conf/mapred-site.xml中加入：

< ?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>  
<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. Note 9001 is not the default port. </description>  
</property>  
<property>  
<name>mapred.map.tasks</name>  
<value>4</value>  
<description>define mapred.map tasks to be number of slave hosts</description>  
</property>  
<property>  
<name>mapred.reduce.tasks</name>  
<value>4</value>  
<description>define mapred.reduce tasks to be number of slave hosts </description>  
</property>  
<property>  
<name>mapred.system.dir</name>  
<value>/home/websearch/nutch-test/filesystem/mapreduce/system</value>  
</property>  
<property>  
<name>mapred.local.dir</name>  
<value>/home/websearch/nutch-test/filesystem/mapreduce/local</value>  
</property>  
</configuration>

6、定义ssh密码

# ssh-keygen -t rsa（一直回车）

#cd /root/.ssh/

#cp id\_rsa.pub authorized\_keys

7、修改全局路径

#vim /etc/profile

修改export HADOOP\_HOME=/home/websearch/nutch-test/search

#source /etc/profile

#$HADOOP\_HOME/bin/hadoop namenode -format

8、单节点上启动Hadoop进程

#$HADOOP\_HOME/bin/start-all.sh

9、为了去测试文件系统，创建一个url的列表，在后面的crawl过程使用，并放到Hadoop分布式文件系统：

#mkdir $HADOOP\_HOME/urlsdir

#echo http://lucene.apache.org > $HADOOP\_HOME/urlsdir/urllist.txt

注意：要给urllist.txt赋予权限

#chmod 777 urllist.txt

#$HADOOP\_HOME/bin/hadoop dfs -put $HADOOP\_HOME/urlsdir urlsdir

此时有可能会遇到问题

Exception in thread "main" java.io.IOException: Not a file: hdfs://localhost.....

（#hadoop dfs -mkdir urlsdir 在hdfs上创建urlsdir目录

# hadoop dfs -put $HADOOP\_HOME/urlsdir/ urlsdir）

验证urlsdir正确放到文件系统中

#$HADOOP\_HOME/bin/hadoop dfs -ls urlsdir

**单机部署Nutch**

1、修改文件

#vim $HADOOP\_HOME/conf/crawl-urlfilter.txt //计算爬取哪些网页

修改行: +^http://([a-z0-9]\*\.)\*MY.DOMAIN.NAME/

为: +^http://([a-z0-9]\*\.)\*

2、在$HADOOP\_HOME/conf/nutch-site.xml中加入：

<property>

<name>http.agent.name</name>  
<value>yourOrganization</value>  
<description>HTTP 'User-Agent' request header. MUST NOT be empty - please set this to a single word uniquely related to your organization.</description>  
</property>

3、开始爬取网页

# $HADOOP\_HOME/bin/nutch crawl urlsdir -dir crawl -depth 5 //深度越大，爬取的网页越多

4、拷贝爬取目录到本地存储(/home/websearch/nutch-test/crawl)

# $HADOOP\_HOME/bin/hadoop dfs -copyToLocal crawl /home/websearch/nutch-test

5、关闭Hadoop操作

# $HADOOP\_HOME/bin/stop-all.sh

**安装Tomcat**

1、解压Tomcat压缩包，目录为/home/websearch/nutch-test/tomcat

定义全局变量路径

#vim /etc/profile

修改TOMCAT\_HOME=/home/websearch/nutch-test/tomcat

#source /etc/profile

2、进入目录

$TOMCAT\_HOME/apache-tomcat-7.0.23/bin

3、解压commons-daemon-native包：

# tar zxvf commons-daemon-native.tar.gz

4、进入目录

$TOMCAT\_HOME/apache-tomcat-7.0.23/bin/commons-daemon-1.0.7-native-src/unix/

5、编译，执行，拷贝编译目录到$TOMCAT\_HOME/apache-tomcat-7.0.23/bin

#./configure

#make

#cp jsvc $TOMCAT\_HOME/apache-tomcat-7.0.23/bin

6、进入TOMCAT\_HOME/apache-tomcat-7.0.23目录

# bin/jsvc -cp ./bin/bootstrap.jar:./bin/tomcat-juli.jar -outfile ./logs/catalina.out -errfile ./logs/catalina.err org.apache.catalina.startup.bootstrap

7、解压缩war文件

# rm -rf $TOMCAT\_HOME/apache-tomcat-7.0.23/webapps/ROOT/\*  
# cd $TOMCAT\_HOME/apache-tomcat-7.0.23/webapps/ROOT  
# cp $HADOOP\_HOME/nutch-1.2.war ./  
# jar -xvf nutch-1.2.war

**分布式搜索**

至此不再需要Hadoop分布式文件系统

1、修改文件

#$TOMCAT\_HOME/ webapps/ROOT/WEB-INF/classes/nutch-site.xml

内容如下

<property>  
<name>fs.default.name</name>  
<value>local</value>  
</property>  
<property>  
<name>searcher.dir</name>  
<value>/home/websearch/nutch-test</value>  
</property>  
<property>  
<name>searcher.max.hits</name>  
<value>-1</value>  
</property>  
<property>  
<name>searcher.max.time.tick\_count</name>  
<value>-1</value>  
</property>  
<property>  
<name>searcher.max.time.tick\_length</name>  
<value>30</value>  
</property>

2、执行/home/websearch/nutch-test/copy\_index\_and\_segments.sh（文件需拷贝）//在search.tar解压后可以找到 将crawl下的indexes和segments文件夹拷贝到local中进行测试

3、拷贝search文件夹到dis\_search下。

4、创建一个文件conf/hadoop-site.xml在那个文件夹下，修改内容：

<?xml version="1.0"?>  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
<configuration>  
<property>  
<name>fs.default.name</name>  
<value>local</value>  
</property>  
</configuration>

5、创建一个文件/home/websearch/nutch-test/search-servers.txt修改：

localhost 8890

6、在/home/websearch/nutch-test/dis\_search/conf/nutch-default.xml中改变searcher.num.handlers比索引服务节点的核心数多一点

7、启动索引服务节点

server1: /home/websearch/nutch-test/dis\_search/bin/nutch server 8890 /home/websearch/nutch-test/local &

8、前端启动Tomcat服务器

$TOMCAT\_HOME/bin/startup.sh

/home/websearch/nutch-test/tomcat/bin/startup.sh

**设定客户端**

1、解压Faban文件在/home/websearch/faban下（$FABAN\_HOME）

2、$FABAN\_HOME/search 拷贝search-release里的search

3、启动

# $FABAN\_HOME/master/bin/startup.sh

4、进入$FABAN\_HOME/search，修改文件build.properties

faban.home=$FABAN\_HOME/可以用/home/websearch/faban/

ant.home=$ANT\_HOME/ (ant home目录)

faban.url=http://localhost:9980/

5、进入$FABAN\_HOME/search

# ant deploy

6、修改$FABAN\_HOME/search/deploy/run.xml

修改ip地址和前端端口

< serverConfig>  
< ipAddress>localhost</ipaddress>  
< portAddress>8080</portAddress>  
< /serverConfing>

修改输入条件的路径，文件名以及日志信息

< filesConfig>  
<logFile>/home/websearch/log/queries.out</logFile> //不存在 <termsFile>/home/websearch/faban/search/src/sample/searchdriver/terms\_en.out</termsFile>  
< /serverConfing>

设置客户端参数

fa:scale (客户端数)   
fa:rampUp, fa:steadyState, and fa:rampDown (the time of rump up, steady state, and rump up of the driver).   
outputDir (the path where the statistics will be written)

**运行测试集**

1、将run.sh拷贝到faban/search/下

2、运行run.sh

/home/websearch/faban/search/run.sh