**Nutch相关框架视频教程**

**第一讲**

1. **通过nutch，诞生了hadoop、tika、gora。**
2. **nutch通过ivy来进行依赖管理（1.2之后）。**
3. **nutch是使用svn进行源代码管理的。**
4. **lucene、nutch、hadoop，在搜索界相当有名。**
5. **ant构建之后，生成runtime文件夹，该文件夹下面有deploy和local文件夹，分别代表了nutch的两种运行方式。**
6. **nutch和hadoop是通过什么连接起来的？通过nutch脚本。通过hadoop命令把apache-nutch-1.6.job提交给hadoop的JobTracker。**
7. **nutch入门重点在于分析nutch脚本文件。**

**第二讲**

1. **git来作为分布式版本控制工具，github作为server。bitbucket.org提供免费的私有库。**
2. **nutch的提高在于研读nutch-default.xml文件中的每一个配置项的实际含义（需要结合源代码理解）。**
3. **定制开发nutch的入门方法是研读build.xml文件。**
4. **命令：**

**apt-get install subversion**

**svn co** [**https://svn.apache.org/repos/asf/nutch/tags/release-1.6/**](https://svn.apache.org/repos/asf/nutch/tags/release-1.6/)

**cd release-1.6**

**apt-get install ant**

**ant**

**cd runtime/local**

**mkdir urls**

**vi urls/url.txt 并输入http://blog.tianya.cn**

**nohup bin/nutch crawl urls -dir data -depth 3 -threads 100 &**

**vi release-1.6/conf/nutch-site.xml 增加http.agent.name配置**

**cd ../../release-1.6**

**ant**

**cd runtime/local**

**nohup bin/nutch crawl urls -dir data -depth 3 -threads 100 &**

**删除报错的文件夹**

**nohup bin/nutch crawl urls -dir data -depth 1 -threads 100 &**

**第三讲**

1. **nutch的存储文件夹data下面各个文件夹和文件里面的内容究竟是什么？**
2. **命令：**

**crawldb**

bin/nutch | grep read

bin/nutch readdb data/crawldb -stats

bin/nutch readdb data/crawldb -dump data/crawldb/crawldb\_dump

bin/nutch readdb data/crawldb -url <http://4008209999.tianyaclub.com/>

bin/nutch readdb data/crawldb -topN 10 data/crawldb/crawldb\_topN

bin/nutch readdb data/crawldb -topN 10 data/crawldb/crawldb\_topN\_m 1

**segments**

**crawl\_generate：**

bin/nutch readseg -dump data/segments/20130325042858 data/segments/20130325042858\_dump -nocontent -nofetch -noparse -noparsedata –noparsetext

**crawl\_fetch：**

bin/nutch readseg -dump data/segments/20130325042858 data/segments/20130325042858\_dump -nocontent -nogenerate -noparse -noparsedata –noparsetext

**content：**

bin/nutch readseg -dump data/segments/20130325042858 data/segments/20130325042858\_dump -nofetch -nogenerate -noparse -noparsedata –noparsetext

**crawl\_parse：**

bin/nutch readseg -dump data/segments/20130325042858 data/segments/20130325042858\_dump -nofetch -nogenerate -nocontent –noparsedata –noparsetext

**parse\_data：**

bin/nutch readseg -dump data/segments/20130325042858 data/segments/20130325042858\_dump -nofetch -nogenerate -nocontent -noparse –noparsetext

**parse\_text：**

bin/nutch readseg -dump data/segments/20130325042858 data/segments/20130325042858\_dump -nofetch -nogenerate -nocontent -noparse -noparsedata

**全部：**

bin/nutch readseg -dump data/segments/20130325042858 data/segments/20130325042858\_dump

**segments**

bin/nutch readseg -list -dir data/segments

bin/nutch readseg -list data/segments/20130325043023

bin/nutch readseg -get data/segments/20130325042858 http://blog.tianya.cn/

**linkdb**

bin/nutch readlinkdb data/linkdb -url http://4008209999.tianyaclub.com/

bin/nutch readlinkdb data/linkdb -dump data/linkdb\_dump

**第四讲**

1. **深入分析nutch的抓取周期**

inject

generate -> fetch -> parse ->updatedb

1. **3大merge和3大read命令阐释**

bin/nutch | grep merge

bin/nutch | grep read

bin/nutch mergesegs data/segments\_all -dir data/segments

1. **反转链接**

bin/nutch invertlinks data/linkdb -dir data/segments

1. **解析页面**

bin/nutch parsechecker <http://apdplat.org>

**第五讲**

1. **域统计**

bin/nutch domainstats data2/crawldb/current host host

bin/nutch domainstats data2/crawldb/current domain domain

bin/nutch domainstats data2/crawldb/current suffix suffix

bin/nutch domainstats data2/crawldb/current tld tld

**2、webgraph**

bin/nutch webgraph -segmentDir data2/segments -webgraphdb data2/webgraphdb

有相同inlinks的URL，只输出topn条

bin/nutch nodedumper -inlinks -output inlinks -topn 1 -webgraphdb data2/webgraphdb

有相同outlinks的URL，只输出topn条

bin/nutch nodedumper -outlinks -output outlinks -topn 1 -webgraphdb data2/webgraphdb

有相同scores的URL，只输出topn条（初始分值全为0）

bin/nutch nodedumper -scores -output scores -topn 1 -webgraphdb data2/webgraphdb

计算URL分值

bin/nutch linkrank -webgraphdb data2/webgraphdb

再次查看分值

bin/nutch nodedumper -scores -output scores –topn 1 -webgraphdb data2/webgraphdb

对结果进行分组，取最大值或是连加和（topn参数不参与）

bin/nutch nodedumper -group domain sum -inlinks -output inlinks\_group\_sum -webgraphdb data2/webgraphdb

bin/nutch nodedumper -group domain max -inlinks -output inlinks\_group\_max -webgraphdb data2/webgraphdb

对url进行分组，分组方式可选择为host或是domain，对同一组的url执行topn限制，对执行了topn限制的url集合执行max或是sum操作，max和sum所针对的排序值是3种方式之一inlinks、outlinks以及scores。

**第六讲**

**1、注入分值**

bin/nutch readdb data2/crawldb -dump crawldb\_dump

cat crawldb\_dump/\* | grep Score | sort | uniq

bin/nutch scoreupdater -crawldb data2/crawldb -webgraphdb data2/webgraphdb

**2、轻量级抓取**

bin/nutch freegen urls data3/segments

**第七讲**

**1、indexchecker**

bin/nutch indexchecker http://www.163.com

**2、安装配置SOLR3.6.2**

wget <http://mirror.bjtu.edu.cn/apache/lucene/solr/3.6.2/apache-solr-3.6.2.tgz>

tar -xzvf apache-solr-3.6.2.tgz

cd apache-solr-3.6.2/example

复制nutch的conf目录中的schema.xml文件到solr/conf目录

将solr/conf/solrconfig.xml文件里的所有<str name="df">text</str>都替换为<str name="df">content</str>

**3、运行SOLR并提交索引**

启动SOLR服务器  
 java -jar start.jar &

Web界面

[http://host2:8983](http://host2:8983/solr/admin/)

提交索引

bin/nutch solrindex http://host2:8983/solr data/crawldb -linkdb data/linkdb -dir data/segments

**4、使用LUKE工具查看索引**

**5、给SOLR3.6.2配置分词器mmseg4j**

wget <http://mmseg4j.googlecode.com/files/mmseg4j-1.8.5.zip>

unzip mmseg4j-1.8.5.zip -d  mmseg4j-1.8.5

将mmseg4j-1.8.5/mmseg4j-all-1.8.5-with-dic.jar复制到solr下的lib目录  
将schema.xml文件中所有的  
  <tokenizer class="solr.WhitespaceTokenizerFactory"/>  
  和  
  <tokenizer class="solr.StandardTokenizerFactory"/>  
  替换为  
  <tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory" mode="complex"/>

**第八讲**

**1、指定LUKE工具的分词器**

访问<https://code.google.com/p/mmseg4j/downloads/list>

下载mmseg4j-1.9.1.v20130120-SNAPSHOT.zip

将压缩包里面的dist文件夹里面的jar解压，将解压出来com和data文件夹拖到lukeall-4.0.0-ALPHA.jar里面

启动luke，在Search选项卡的Analysis里面选择com.chenlb.mmseg4j.analysis.ComplexAnalyzer

**2、安装配置SOLR4.2**

wget http://labs.mop.com/apache-mirror/lucene/solr/4.2.0/solr-4.2.0.tgz

tar -xzvf solr-4.2.0.tgz

cd solr-4.2.0/example

复制nutch的conf目录中的schema-solr4.xml文件到solr/collection1/conf目录，改名为schema.xml，覆盖原来文件

修改solr/collection1/conf/schema.xml，在<fields>下增加：<field name="\_version\_" type="long" indexed="true" stored="true"/>

**3、给SOLR4.2配置分词器mmseg4j**

wget https://mmseg4j.googlecode.com/files/mmseg4j-1.9.1.v20130120-SNAPSHOT.zip

unzip mmseg4j-1.9.1.v20130120-SNAPSHOT.zip -d  mmseg4j-1.9.1

将mmseg4j-1.9.1/dist/\*.jar复制到solr下的lib目录  
将schema.xml文件中的  
  <tokenizer class="solr.WhitespaceTokenizerFactory"/>  
  和  
  <tokenizer class="solr.StandardTokenizerFactory"/>  
  替换为  
  <tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory" mode="complex"/>

**4、运行SOLR并提交索引**

启动SOLR服务器  
 java -jar start.jar &

Web界面

[http://host2:8983](http://host2:8983/solr/admin/)

提交索引

bin/nutch solrindex http://host2:8983/solr data/crawldb -linkdb data/linkdb -dir data/segments

**第九讲**

1. **安装win上的nutch运行环境Cygwin**

cygwin路径不要有空格

把安装好的JDK拷贝到用户主目录

把nutch的文件拷贝到用户主目录

下载解压ant，加入path

1. **运行nutch**

Exception in thread "main" java.io.IOException: Failed to set permissions of path: \tmp\hadoop-ysc\mapred\staging\ysc-2036315919\.staging to 0700

<https://issues.apache.org/jira/browse/HADOOP-7682>

<http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-1.1.2/hadoop-1.1.2.tar.gz>

修改hadoop-1.1.2\src\core\org\apache\hadoop\fs\ FileUtil.java，搜索 Failed to set permissions of path，找到689行，把throw new IOException改为LOG.warn

修改hadoop-1.1.2\ build.xml，搜索autoreconf，移除匹配的6个executable="autoreconf"的exec配置

执行ant

用新生成的hadoop-core-1.1.3-SNAPSHOT.jar替换nutch的hadoop-core-1.0.3.jar

**第十讲**

**1、HADOOP单机本地模式**

wget <http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-1.1.2/hadoop-1.1.2.tar.gz>

tar -xzvf hadoop-1.1.2.tar.gz

export PATH=/home/ysc/hadoop-1.1.2/bin:$PATH

切换到nutch的deploy目录运行命令

**2、HADOOP单机伪分布式模式**

**新建用户和组**

addgroup hadoop

adduser --ingroup hadoop hadoop

注销root以hadoop用户登录

**配置SSH**

ssh-keygen -t rsa（密码为空，路径默认）

cp .ssh/id\_rsa.pub .ssh/authorized\_keys

**准备HADOOP运行环境**

wget <http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-1.1.2/hadoop-1.1.2.tar.gz>

tar -xzvf hadoop-1.1.2.tar.gz

在/home/hadoop/.bashrc 中追加：

export PATH=/home/hadoop/hadoop-1.1.2/bin:$PATH

重新登录就生效

ssh localhost

which hadoop

**配置HADOOP运行参数**

vi conf/core-site.xml

<property>

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

<value>hdfs://localhost:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/home/hadoop/tmp</value>

</property>

vi conf/hdfs-site.xml

<property>

<name>dfs.name.dir</name>

<value>/home/hadoop/dfs/filesystem/name</value>

</property>

<property>

<name>dfs.data.dir</name>

<value>/home/hadoop/dfs/filesystem/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

vi conf/mapred-site.xml

<property>

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

<value>localhost:9001</value>

</property>

<property>

<name>mapred.tasktracker.map.tasks.maximum</name>

<value>4</value>

</property>

<property>

<name>mapred.tasktracker.reduce.tasks.maximum</name>

<value>4</value>

</property>

<property>

<name>mapred.system.dir</name>

<value>/home/hadoop/mapreduce/system</value>

</property>

<property>

<name>mapred.local.dir</name>

<value>/home/hadoop/mapreduce/local</value>

</property>

**格式化名称节点并启动集群**

hadoop namenode -format

**启动集群并查看WEB管理界面**

start-all.sh

访问 [http://localhost:50030](http://localhost:50030/) 可以查看 JobTracker 的运行状态

访问 [http://localhost:50060](http://localhost:50060/) 可以查看 TaskTracker 的运行状态

访问 [http://localhost:50070](http://localhost:50070/) 可以查看 NameNode 以及整个分布式文件系统的状态，浏览分布式文件系统中的文件以及 log 等

**停止集群**

stop-all.sh停止集群

**3、执行Nutch的crawl命令**

**第十一讲**

**1、配置Cygwin支持无密码SSH登陆**

**安装SSH**

默认的Cygwin没有安装ssh，所以重新运行<http://www.cygwin.com/setup.exe>

在Select Packages的时候，在search输入ssh，选择openssh: The OpenSSH server and client programs

**配置SSH服务（以管理员身份运行cygwin）**

ssh-host-config

Should privilege separation be used? yes

Do you want to install sshd as a service? yes

默认确认

Do you want to use a different name? no

Create new privileged user account 'cyg\_server'? yes

输入密码

cygrunsrv -S sshd

如果需要重新安装sshd服务，可以用**cygrunsrv -R sshd**

**生成SSH Key**

ssh-keygen -t rsa（密码为空，路径默认）

cp .ssh/id\_rsa.pub .ssh/authorized\_keys

**登陆**

ssh localhost

**2、win上的HADOOP单机伪分布式**

**准备HADOOP运行环境**

下载解压并拷贝到Cygwin的用户主目录

<http://archive.apache.org/dist/hadoop/core/hadoop-0.20.2/hadoop-0.20.2.tar.gz>

1.x版本有BUG,参考：

<https://issues.apache.org/jira/browse/HADOOP-7682>  
<https://issues.apache.org/jira/browse/HADOOP-8274>

BUG修复请参考：

<http://en.wikisource.org/wiki/User:Fkorning/Code/Hadoop-on-Cygwin>

在/home/ysc/.bashrc 中追加：

export JAVA\_HOME=/home/ysc/jdk1.7.0\_17

export PATH=/home/ysc/hadoop-0.20.2/bin:$JAVA\_HOME/bin:$PATH

在hadoop-0.20.2/conf/hadoop-evn.sh中追加

export JAVA\_HOME=/home/ysc/jdk1.7.0\_17

export HADOOP\_LOG\_DIR=/tmp/logs

创建符号链接

mklink /D C:\tmp C:\cygwin\tmp

重新登录就生效

ssh localhost

which hadoop

**配置HADOOP运行参数**

vi conf/core-site.xml

<property>

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

<value>hdfs://localhost:9000</value>

</property>

vi conf/hdfs-site.xml

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

vi conf/mapred-site.xml

<property>

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

<value>localhost:9001</value>

</property>

<property>

<name>mapred.tasktracker.map.tasks.maximum</name>

<value>4</value>

</property>

<property>

<name>mapred.tasktracker.reduce.tasks.maximum</name>

<value>4</value>

</property>

**格式化名称节点并启动集群**

hadoop namenode -format

**启动集群并查看WEB管理界面**

start-all.sh

访问 [http://localhost:50030](http://localhost:50030/) 可以查看 JobTracker 的运行状态

访问 [http://localhost:50060](http://localhost:50060/) 可以查看 TaskTracker 的运行状态

访问 [http://localhost:50070](http://localhost:50070/) 可以查看 NameNode 以及整个分布式文件系统的状态，浏览分布式文件系统中的文件以及 log 等

**停止集群**

stop-all.sh停止集群

1. **执行wordcount命令**

hadoop jar hadoop-0.20.2-examples.jar wordcount input output

**第十二讲**

1. **HADOOP多机完全分布式模式**

**三台机器**

host2(NameNode、SecondaryNameNode、JobTracker、DataNode、TaskTracker)

host6(DataNode、TaskTracker)

host8(DataNode、TaskTracker)

vi /etc/hostname（分别给每一台主机指定主机名）

vi /etc/hosts（分别给每一台主机指定主机名到IP地址的映射）

**新建用户和组**

三台机器上面都要新建用户和组

addgroup hadoop

adduser --ingroup hadoop hadoop

更改临时目录权限

chmod 777 /tmp

注销root以hadoop用户登录

**配置SSH**

在host2上面执行

ssh-keygen -t rsa（密码为空，路径默认）

该命令会在用户主目录下创建 .ssh 目录，并在其中创建两个文件：id\_rsa 私钥文件，是基于 RSA 算法创建，该私钥文件要妥善保管，不要泄漏。id\_rsa.pub 公钥文件，和 id\_rsa 文件是一对儿，该文件作为公钥文件，可以公开

cp .ssh/id\_rsa.pub .ssh/authorized\_keys

把公钥追加到其他主机的authorized\_keys 文件中

ssh-copy-id -i .ssh/id\_rsa.pub hadoop@host6

ssh-copy-id -i .ssh/id\_rsa.pub hadoop@host8

可以在host2上面通过ssh无密码登陆host6和host8

ssh host2

ssh host6

ssh host8

**准备HADOOP运行环境**

wget <http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-1.1.2/hadoop-1.1.2.tar.gz>

tar -xzvf hadoop-1.1.2.tar.gz

在/home/hadoop/.bashrc 中追加：

export PATH=/home/hadoop/hadoop-1.1.2/bin:$PATH

重新登录就生效

ssh localhost

which hadoop

**配置HADOOP运行参数**

vi conf/masters

把localhost替换为 ：host2

vi conf/slaves

删除localhost，加入两行:

host2

host6

host8

vi conf/core-site.xml

<property>

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

<value>hdfs://host2:9000</value>

</property>

vi conf/hdfs-site.xml

<property>

<name>dfs.name.dir</name>

<value>/home/hadoop/dfs/filesystem/name</value>

</property>

<property>

<name>dfs.data.dir</name>

<value>/home/hadoop/dfs/filesystem/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

vi conf/mapred-site.xml

<property>

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

<value>host2:9001</value>

</property>

<property>

<name>mapred.tasktracker.map.tasks.maximum</name>

<value>4</value>

</property>

<property>

<name>mapred.tasktracker.reduce.tasks.maximum</name>

<value>4</value>

</property>

<property>

<name>mapred.system.dir</name>

<value>/home/hadoop/mapreduce/system</value>

</property>

<property>

<name>mapred.local.dir</name>

<value>/home/hadoop/mapreduce/local</value>

</property>

**复制HADOOP文件到其他节点**

 scp -r /home/hadoop/hadoop-1.1.2 hadoop@host6:/home/hadoop/hadoop-1.1.2  
 scp -r /home/hadoop/hadoop-1.1.2 hadoop@host8:/home/hadoop/hadoop-1.1.2

**格式化名称节点并启动集群**

hadoop namenode -format

**启动集群并查看WEB管理界面**

start-all.sh

访问 [http://localhost:50030](http://localhost:50030/) 可以查看 JobTracker 的运行状态

访问 [http://localhost:50060](http://localhost:50060/) 可以查看 TaskTracker 的运行状态

访问 [http://localhost:50070](http://localhost:50070/) 可以查看 NameNode 以及整个分布式文件系统的状态，浏览分布式文件系统中的文件以及 log 等

**停止集群**

stop-all.sh停止集群

**第十三讲**

**1、改变负载**

**三台机器，改变负载**

host2(NameNode、DataNode、TaskTracker)

host6(SecondaryNameNode、DataNode、TaskTracker)

host8(JobTracker 、DataNode、TaskTracker)

**指定SecondaryNameNode为host6：**

**vi conf/masters指定host6**

scp conf/masters host6:/home/hadoop/hadoop-1.1.2/conf/masters

scp conf/masters host8:/home/hadoop/hadoop-1.1.2/conf/masters

**vi conf/hdfs-site.xml**

   <property>  
     <name>dfs.http.address</name>  
     <value>host2:50070</value>  
   </property>

   <property>

<name>dfs.secondary.http.address</name>

<value>host6:50090</value>

   </property>

scp conf/hdfs-site.xml host6:/home/hadoop/hadoop-1.1.2/conf/hdfs-site.xml

scp conf/hdfs-site.xml host8:/home/hadoop/hadoop-1.1.2/conf/hdfs-site.xml

**指定JobTracker为host8：**

**vi conf/mapred-site.xml**

<property>

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

<value>host8:9001</value>

</property>

scp conf/mapred-site.xml host6:/home/hadoop/hadoop-1.1.2/conf/mapred-site.xml

scp conf/mapred-site.xml host8:/home/hadoop/hadoop-1.1.2/conf/mapred-site.xml

**vi conf/core-site.xml**

<property>

<name>fs.checkpoint.dir</name>

<value>/home/hadoop/dfs/filesystem/namesecondary</value>

</property>

scp conf/core-site.xml host6:/home/hadoop/hadoop-1.1.2/conf/core-site.xml

scp conf/core-site.xml host8:/home/hadoop/hadoop-1.1.2/conf/core-site.xml

**配置host8：**

host8上的脚本start-mapred.sh会启动host2和host6上面的TaskTracker，所以需要对host8执行：

ssh-keygen -t rsa（密码为空，路径默认）

ssh-copy-id -i .ssh/id\_rsa.pub hadoop@host2

ssh-copy-id -i .ssh/id\_rsa.pub hadoop@host6

ssh-copy-id -i .ssh/id\_rsa.pub hadoop@host8

可以在host8上面通过ssh无密码登陆host2和host6

ssh host2

ssh host6

ssh host8

在/home/hadoop/.bashrc 中追加：

export PATH=/home/hadoop/hadoop-1.1.2/bin:$PATH

host2: 执行start-dfs.sh

host8: 执行start-mapred.sh

**2、SecondaryNameNode**

ssh host6

停止secondarynamenode

hadoop-1.1.2/bin/hadoop-daemon.sh stop secondarynamenode

强制合并fsimage和eidts

hadoop-1.1.2/bin/hadoop secondarynamenode -checkpoint force

启动secondarynamenode

hadoop-1.1.2/bin/hadoop-daemon.sh start secondarynamenode

**3、启用回收站**

<property>

<name>fs.trash.interval</name>

<value>10080</value>

</property>

**第十四讲**

1. **动态增加DataNode节点和TaskTracker节点**

**以host226为例**

在host226上执行：

指定主机名

vi /etc/hostname

指定主机名到IP地址的映射

vi /etc/hosts

增加用户和组

addgroup hadoop

adduser --ingroup hadoop hadoop

更改临时目录权限

chmod 777 /tmp

在host2上执行：

vi conf/slaves

增加host226

ssh-copy-id -i .ssh/id\_rsa.pub hadoop@host226

scp -r /home/hadoop/hadoop-1.1.2 hadoop@host226:/home/hadoop/hadoop-1.1.2

在host8上执行：

vi conf/slaves

增加host226

ssh-copy-id -i .ssh/id\_rsa.pub hadoop@host226

在host226上面执行：

hadoop-daemon.sh start datanode

hadoop-daemon.sh start tasktracker

**在/etc/hosts的配置文件中，localhost到ip地址的映射要放到ipv4的最后面**

**第十五讲**

**1、限制hadoop节点连接**

**NameNode:**

**vi conf/hdfs-site.xml**

<property>

<name>dfs.hosts</name>

<value>/home/hadoop/hadoop-1.1.2/conf/include</value>

</property>

<property>

<name>dfs.hosts.exclude</name>

<value>/home/hadoop/hadoop-1.1.2/conf/exclude</value>

</property>

加入集群节点

vi /home/hadoop/hadoop-1.1.2/conf/include

**JobTracker:**

**vi conf/mapred-site.xml**

<property>

<name>mapred.hosts</name>

<value>/home/hadoop/hadoop-1.1.2/conf/include</value>

</property>

<property>

<name>mapred.hosts.exclude</name>

<value>/home/hadoop/hadoop-1.1.2/conf/exclude</value>

</property>

加入集群节点

vi /home/hadoop/hadoop-1.1.2/conf/include

重启集群

**2、动态删除DataNode节点和TaskTracker节点**

vi /home/hadoop/hadoop-1.1.2/conf/exclude

增加待删除的节点host226

在NameNode上面执行：

hadoop dfsadmin -refreshNodes

vi hadoop-1.1.2/conf/slaves （去掉host226）

vi hadoop-1.1.2/conf/include （去掉host226）

hadoop dfsadmin -refreshNodes(使include的更改生效)

rm hadoop-1.1.2/conf/exclude

exclude主要是使一个datanode节点安全退役

**删除tasktracker方式一：**

vi /home/hadoop/hadoop-1.1.2/conf/exclude

增加待删除的节点host226

在JobTracker上面执行：

hadoop mradmin -refreshNodes

vi hadoop-1.1.2/conf/slaves （去掉host226）

vi hadoop-1.1.2/conf/include （去掉host226）

hadoop mradmin -refreshNodes(使include的更改生效)

rm hadoop-1.1.2/conf/exclude

**删除tasktracker方式二：**

vi /home/hadoop/hadoop-1.1.2/conf/include

删除待删除的节点host226

在JobTracker上面执行：

hadoop mradmin -refreshNodes

vi hadoop-1.1.2/conf/slaves （去掉host226）

**第十六讲**

**1、运行基准测试**

hadoop jar hadoop-test-1.1.2.jar

hadoop jar hadoop-test-1.1.2.jar DFSCIOTest -write -nrFiles 12 -fileSize 100000 -resFile test

hadoop jar hadoop-test-1.1.2.jar DFSCIOTest -read -nrFiles 12 -fileSize 100000 -resFile test

hadoop jar hadoop-test-1.1.2.jar DFSCIOTest -clear

**第十七讲**

Ganglia主要是用来监控大规模分布式系统的性能，如：cpu 、内存、硬盘、负载、网络流量等。Ganglia支持通过浏览器访问，强大的图表展示方式很容易直观地了解每个节点以及整个集群的工作状态，对集群运行参数调整、提高系统整体资源利用率起到重要作用。

1. **配置服务端**

host6作为服务端：

**创建用户和组：**

addgroup ganglia

adduser --ingroup ganglia ganglia

**安装：**

apt-get install  gmetad

apt-get install  rrdtool

apt-get install  ganglia-webfrontend

apt-get install  ganglia-monitor

**配置gmond：**

vi /etc/ganglia/gmond.conf

先找到setuid = yes,改成setuid =no;

在找到cluster块中的name，改成name =”hadoop-cluster”;

**配置gmetad：**

vi /etc/ganglia/gmetad.conf

在这个配置文件中增加datasource，即增加以下内容：

data\_source “hadoop-cluster” 10 host2 host6 host8

gridname "Hadoop"

**指定web文件夹：**

ln -s /usr/share/ganglia-webfrontend /var/www/ganglia

**指定主机名：**

vi /etc/apache2/apache2.conf

添加： ServerName host6

**重启服务：**

/etc/init.d/gmetad restart

/etc/init.d/ganglia-monitor restart

/etc/init.d/apache2 restart

1. **配置客户端**

在host2和host8上安装数据收集服务：

**创建用户和组：**

addgroup ganglia

adduser --ingroup ganglia ganglia

**安装：**

apt-get install  ganglia-monitor

**配置gmond：**

vi /etc/ganglia/gmond.conf

先找到setuid = yes,改成setuid =no;

在找到cluster块中的name，改成name =”hadoop-cluster”;

**重启服务：**

/etc/init.d/ganglia-monitor restart

1. **访问页面**

<http://host6/ganglia>

如果页面中的Choose a Source有unspecified, 重启gmetad即可：

/etc/init.d/gmetad restart

1. **集成hadoop**

**vi conf/hadoop-metrics2.properties**

**设置内容为：**

  # 大于0.20以后的版本用ganglia31

\*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31  
  \*.sink.ganglia.period=10  
  # default for supportsparse is false  
  \*.sink.ganglia.supportsparse=true  
 \*.sink.ganglia.slope=jvm.metrics.gcCount=zero,jvm.metrics.memHeapUsedM=both  
 \*.sink.ganglia.dmax=jvm.metrics.threadsBlocked=70,jvm.metrics.memHeapUsedM=40  
  #组播地址239.2.11.71

  namenode.sink.ganglia.servers=239.2.11.71:8649  
  datanode.sink.ganglia.servers=239.2.11.71:8649  
  jobtracker.sink.ganglia.servers=239.2.11.71:8649  
  tasktracker.sink.ganglia.servers=239.2.11.71:8649  
  maptask.sink.ganglia.servers=239.2.11.71:8649  
  reducetask.sink.ganglia.servers=239.2.11.71:8649  
  dfs.class=org.apache.hadoop.metrics.ganglia.GangliaContext31  
  dfs.period=10  
  dfs.servers=239.2.11.71:8649  
  mapred.class=org.apache.hadoop.metrics.ganglia.GangliaContext31  
  mapred.period=10  
  mapred.servers=239.2.11.71:8649  
  jvm.class=org.apache.hadoop.metrics.ganglia.GangliaContext31  
  jvm.period=10  
  jvm.servers=239.2.11.71:8649

**把配置文件复制到集群其他节点，重启集群。**

**第十八讲**

**1、准备压缩数据**

**从dmoz下载url库**

wget http://rdf.dmoz.org/rdf/content.rdf.u8.gz

gunzip content.rdf.u8.gz

**准备nutch1.6**

svn co <https://svn.apache.org/repos/asf/nutch/tags/release-1.6/>

cp release-1.6/conf/nutch-site.xml.template release-1.6/conf/nutch-site.xml

vi release-1.6/conf/nutch-site.xml

增加：

<property>

<name>http.agent.name</name>

<value>nutch</value>

</property>

cd release-1.6

ant

cd ..

**使用DmozParser把dmoz的URL库解析为文本**

release-1.6/runtime/local/bin/nutch org.apache.nutch.tools.DmozParser content.rdf.u8 > urls &

**将url文本内容放到HDFS上面**

hadoop fs -put urls urls

**2、以不同压缩方法注入URL**

**进入nutch主目录**

cd release-1.6

**以未压缩的方式注入URL**

runtime/deploy/bin/nutch inject data\_no\_compress/crawldb urls

**以默认压缩的方式注入URL**

vi conf/nutch-site.xml

<property>

<name>mapred.output.compression.type</name>

<value>BLOCK</value>

</property>

<property>

<name>mapred.output.compress</name>

<value>true</value>

</property>

<property>

<name>mapred.compress.map.output</name>

<value>true</value>

</property>

<property>

<name>mapred.map.output.compression.codec</name>

<value>org.apache.hadoop.io.compress.DefaultCodec</value>

</property>

<property>

<name>mapred.output.compression.codec</name>

<value>org.apache.hadoop.io.compress.DefaultCodec</value>

</property>

ant

runtime/deploy/bin/nutch inject data\_default\_compress/crawldb urls

**以Gzip压缩的方式注入URL**

vi conf/nutch-site.xml

<property>

<name>mapred.output.compression.type</name>

<value>BLOCK</value>

</property>

<property>

<name>mapred.output.compress</name>

<value>true</value>

</property>

<property>

<name>mapred.compress.map.output</name>

<value>true</value>

</property>

<property>

<name>mapred.map.output.compression.codec</name>

<value>org.apache.hadoop.io.compress.GzipCodec</value>

</property>

<property>

<name>mapred.output.compression.codec</name>

<value>org.apache.hadoop.io.compress.GzipCodec</value>

</property>

ant

runtime/deploy/bin/nutch inject data\_gzip\_compress/crawldb urls

**以BZip2的压缩方式注入URL**

vi conf/nutch-site.xml

<property>

<name>mapred.output.compression.type</name>

<value>BLOCK</value>

</property>

<property>

<name>mapred.output.compress</name>

<value>true</value>

</property>

<property>

<name>mapred.compress.map.output</name>

<value>true</value>

</property>

<property>

<name>mapred.map.output.compression.codec</name>

<value>org.apache.hadoop.io.compress.BZip2Codec</value>

</property>

<property>

<name>mapred.output.compression.codec</name>

<value>org.apache.hadoop.io.compress.BZip2Codec</value>

</property>

ant

runtime/deploy/bin/nutch inject data\_bzip2\_compress/crawldb urls

**以Snappy的方式注入URL**

vi conf/nutch-site.xml

<property>

<name>mapred.output.compression.type</name>

<value>BLOCK</value>

</property>

<property>

<name>mapred.output.compress</name>

<value>true</value>

</property>

<property>

<name>mapred.compress.map.output</name>

<value>true</value>

</property>

<property>

<name>mapred.map.output.compression.codec</name>

<value>org.apache.hadoop.io.compress.SnappyCodec</value>

</property>

<property>

<name>mapred.output.compression.codec</name>

<value>org.apache.hadoop.io.compress.SnappyCodec</value>

</property>

ant

runtime/deploy/bin/nutch inject data\_snappy\_compress/crawldb urls

**压缩类型的影响**

**块大小的影响**

**3、Hadoop配置Snappy压缩**

**下载解压：**

wget <https://snappy.googlecode.com/files/snappy-1.1.0.tar.gz>

tar -xzvf snappy-1.1.0.tar.gz

cd snappy-1.0.5

**编译:**

./configure

make

make install

**复制库文件：**

scp /usr/local/lib/libsnappy\* host2:/home/hadoop/hadoop-1.1.2/lib/native/Linux-amd64-64/

scp /usr/local/lib/libsnappy\* host6:/home/hadoop/hadoop-1.1.2/lib/native/Linux-amd64-64/

scp /usr/local/lib/libsnappy\* host8:/home/hadoop/hadoop-1.1.2/lib/native/Linux-amd64-64/

**在每一台集群机器上面修改环境变量：**

vi /home/hadoop/.bashrc

追加：

export LD\_LIBRARY\_PATH=/home/hadoop/hadoop-1.1.2/lib/native/Linux-amd64-64

**第十九讲**

**1、Ganglia以组播方式监控同一个网段中的多个集群**

vi /etc/ganglia/gmetad.conf   
data\_source "cluster1" 10 host2   
data\_source "cluster2" 10 host6   
data\_source "cluster3" 10 host8

/etc/init.d/gmetad restart

**下面要分别指定节点使用的端口：**

cluster1：

vi /etc/ganglia/gmond.conf

**指定集群名称：**

cluster {  
  name = "cluster1"  
  owner = "unspecified"  
  latlong = "unspecified"  
  url = "unspecified"  
}

**指定端口：**

udp\_send\_channel {  
  mcast\_join = 239.2.11.71  
  port = 8661  
  ttl = 1  
}  
udp\_recv\_channel {  
  mcast\_join = 239.2.11.71  
  port = 8661  
  bind = 239.2.11.71  
}

/etc/init.d/ganglia-monitor restart

cluster2：

vi /etc/ganglia/gmond.conf

**指定集群名称：**

cluster {  
  name = "cluster2"  
  owner = "unspecified"  
  latlong = "unspecified"  
  url = "unspecified"  
}

**指定端口：**

udp\_send\_channel {  
  mcast\_join = 239.2.11.71  
  port = 8662  
  ttl = 1  
}  
udp\_recv\_channel {  
  mcast\_join = 239.2.11.71  
  port = 8662  
  bind = 239.2.11.71  
}

/etc/init.d/ganglia-monitor restart

cluster3:

vi /etc/ganglia/gmond.conf

**指定集群名称：**

cluster {  
  name = "cluster3"  
  owner = "unspecified"  
  latlong = "unspecified"  
  url = "unspecified"  
}

**指定端口：**

udp\_send\_channel {  
  mcast\_join = 239.2.11.71  
  port = 8663  
  ttl = 1  
}  
udp\_recv\_channel {  
  mcast\_join = 239.2.11.71  
  port = 8663  
  bind = 239.2.11.71  
}

/etc/init.d/ganglia-monitor restart

**2、Ganglia以单播方式监控同一个网段中的多个集群**

vi /etc/ganglia/gmetad.conf   
data\_source "cluster1" 10 host2   
data\_source "cluster2" 10 host6   
data\_source "cluster3" 10 host8

/etc/init.d/gmetad restart

cluster1：

vi /etc/ganglia/gmond.conf

**指定集群名称：**

cluster {  
  name = "cluster1"  
  owner = "unspecified"  
  latlong = "unspecified"  
  url = "unspecified"  
}

**指定接收数据的节点：**

udp\_send\_channel {

# mcast\_join = 239.2.11.71

host = host2

port = 8649

ttl = 1

}

udp\_recv\_channel {

# mcast\_join = 239.2.11.71

port = 8649

# bind = 239.2.11.71

}

/etc/init.d/ganglia-monitor restart

cluster2：

vi /etc/ganglia/gmond.conf

**指定集群名称：**

cluster {  
  name = "cluster2"  
  owner = "unspecified"  
  latlong = "unspecified"  
  url = "unspecified"  
}

**指定接收数据的节点：**

udp\_send\_channel {

# mcast\_join = 239.2.11.71

host = host6

port = 8649

ttl = 1

}

udp\_recv\_channel {

# mcast\_join = 239.2.11.71

port = 8649

# bind = 239.2.11.71

}

/etc/init.d/ganglia-monitor restart

cluster3：

vi /etc/ganglia/gmond.conf

**指定集群名称：**

cluster {  
  name = "cluster3"  
  owner = "unspecified"  
  latlong = "unspecified"  
  url = "unspecified"  
}

**指定接收数据的节点：**

udp\_send\_channel {

# mcast\_join = 239.2.11.71

host = host8

port = 8649

ttl = 1

}

udp\_recv\_channel {

# mcast\_join = 239.2.11.71

port = 8649

# bind = 239.2.11.71

}

/etc/init.d/ganglia-monitor restart

**3、Ganglia监控不同网段中的多个集群**

**不同网段中的主机如果属于同一个集群，则无法使用ganglia的多播配置方法，必须使用单播。**

**下面把host226加入cluster1：**

在host226上安装数据收集服务：

**创建用户和组：**

addgroup ganglia

adduser --ingroup ganglia ganglia

**安装：**

apt-get install  ganglia-monitor

**配置gmond：**

vi /etc/ganglia/gmond.conf

先找到setuid = yes,改成setuid =no;

在找到cluster块中的name，改成name =”cluster1”;

**指定端口(要注意刚才第一步演示组播的时候已经把UDP端口改为8661)：**

udp\_send\_channel {  
  mcast\_join = 239.2.11.71  
  port = 8661  
  ttl = 1  
}  
udp\_recv\_channel {  
  mcast\_join = 239.2.11.71  
  port = 8661  
  bind = 239.2.11.71  
}

**重启服务：**

/etc/init.d/ganglia-monitor restart

**第二十讲**

**1、Ganglia以单播方式监控跨多个网段的单一集群**

vi /etc/ganglia/gmetad.conf   
data\_source "hadoop-cluster" 10 host6

/etc/init.d/gmetad restart

在集群的所有节点中指定以下配置：

vi /etc/ganglia/gmond.conf

**指定集群名称：**

cluster {  
  name = "hadoop-cluster"  
  owner = "unspecified"  
  latlong = "unspecified"  
  url = "unspecified"  
}

**指定接收数据的节点：**

udp\_send\_channel {

# mcast\_join = 239.2.11.71

host = host6

port = 8649

ttl = 1

}

udp\_recv\_channel {

# mcast\_join = 239.2.11.71

port = 8649

# bind = 239.2.11.71

}

/etc/init.d/ganglia-monitor restart

**2、配置Hadoop集群使用单播地址**

**vi conf/hadoop-metrics2.properties**

设置内容为：

  # 大于0.20以后的版本用ganglia31

\*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31  
  \*.sink.ganglia.period=10  
  # default for supportsparse is false  
  \*.sink.ganglia.supportsparse=true  
 \*.sink.ganglia.slope=jvm.metrics.gcCount=zero,jvm.metrics.memHeapUsedM=both  
 \*.sink.ganglia.dmax=jvm.metrics.threadsBlocked=70,jvm.metrics.memHeapUsedM=40

  namenode.sink.ganglia.servers=host6  
  datanode.sink.ganglia.servers= host6  
  jobtracker.sink.ganglia.servers= host6  
  tasktracker.sink.ganglia.servers= host6  
  maptask.sink.ganglia.servers= host6  
  reducetask.sink.ganglia.servers= host6  
  dfs.class=org.apache.hadoop.metrics.ganglia.GangliaContext31  
  dfs.period=10  
  dfs.servers= host6  
  mapred.class=org.apache.hadoop.metrics.ganglia.GangliaContext31  
  mapred.period=10  
  mapred.servers= host6  
  jvm.class=org.apache.hadoop.metrics.ganglia.GangliaContext31  
  jvm.period=10  
  jvm.servers= host6

**把配置文件复制到集群其他节点，重启集群。**

**3、扩展集群，节点分别位于3个不同网段**

将host226重新加入集群，并新增host138

在host6和host8的include文件中加入host226和host138

在host6和host8的slaves文件中加入host226和host138

在新增的节点host138上面执行：

**指定主机名**

vi /etc/hostname

**指定主机名到IP地址的映射**

vi /etc/hosts

**增加用户和组**

addgroup hadoop

adduser --ingroup hadoop hadoop

**更改临时目录权限**

chmod 777 /tmp

在host2和host8 上面配置对host138的SSH登陆：

ssh-copy-id -i .ssh/id\_rsa.pub hadoop@host138

在host2上将hadoop文件复制到host138：

scp -r /home/hadoop/hadoop-1.1.2 hadoop@host138:/home/hadoop/hadoop-1.1.2

如果集群已经在运行，则在host226和host138上面执行以下命令以动态增加节点：

hadoop-daemon.sh start datanode

hadoop-daemon.sh start tasktracker

**4、配置host138**

在host138上安装数据收集服务：

**创建用户和组：**

addgroup ganglia

adduser --ingroup ganglia ganglia

**安装：**

apt-get install  ganglia-monitor

**配置gmond：**

vi /etc/ganglia/gmond.conf

**指定集群名称：**

cluster {  
  name = "hadoop-cluster"  
  owner = "unspecified"  
  latlong = "unspecified"  
  url = "unspecified"  
}

**指定接收数据的节点：**

udp\_send\_channel {

# mcast\_join = 239.2.11.71

host = host6

port = 8649

ttl = 1

}

udp\_recv\_channel {

# mcast\_join = 239.2.11.71

port = 8649

# bind = 239.2.11.71

}

/etc/init.d/ganglia-monitor restart