# Nutch搜索引擎培训讲义

作者：杨尚川  
QQ : 281032878  
手机：18919193664   
邮箱：[yang-shangchuan@qq.com](mailto:yang-shangchuan@qq.com)  
主页1：<http://www.taskcity.com/u/Ysc>   
主页2：<http://www.csto.com/u/yangshangchuan1>  
微博：<http://t.qq.com/yang-shangchuan>  
CSDN共享资源：<http://download.csdn.net/user/yangshangchuan>  
我的开源项目：<https://github.com/ysc/APDPlat>  
我读的书：<http://book.douban.com/people/49824900/collect>  
双高证书：<http://user.qzone.qq.com/281032878/blog/1342675154>

版本变化：

V1.0 2012-9-5 杨尚川

V1.1 2012-9-23 杨尚川

V1.2 2013-1-7 杨尚川

一、nutch1.2

步骤和二大同小异，在步骤 5、配置构建路径 中需要多两个操作：在左部Package Explorer的 nutch1.2文件夹上单击右键 > Build Path > Configure Build Path... > 选中Source选项 > Default output folder:修改nutch1.2/bin为nutch1.2/\_bin，在左部Package Explorer的 nutch1.2文件夹下的bin文件夹上单击右键 > Team > 还原

二中黄色背景部分是版本号的差异，红色部分是1.2版本没有的，绿色部分是不一样的地方，如下：

1、Add JARs... > nutch1.2 > lib ，选中所有的.jar文件 > OK

2、crawl-urlfilter.txt

3、将crawl -urlfilter.txt.template改名为crawl -urlfilter.txt

4、修改crawl-urlfilter.txt，将

# accept hosts in MY.DOMAIN.NAME

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

# skip everything else

-.

5、cd /home/ysc/workspace/nutch1.2

nutch1.2是一个完整的搜索引擎，nutch1.5.1只是一个爬虫。nutch1.2可以把索引提交给SOLR，也可以直接生成LUCENE索引，nutch1.5.1则只能把索引提交给SOLR：

1、cd /home/ysc

2、wget [http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-7/v7.0.29/bin/apache-tomcat-7.0.29.tar.gz](http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-7/v7.0.29/bin/apache-tomcat-7.0.29.tar.odt)

3、tar -xvf apache-tomcat-7.0.29.tar.gz

4、在左部Package Explorer的 nutch1.2文件夹下的build.xml文件上单击右键 > Run As > Ant Build... > 选中war target > Run

5、cd /home/ysc/workspace/nutch1.2/build

6、unzip nutch-1.2.war -d nutch-1.2

7、cp -r nutch-1.2 /home/ysc/apache-tomcat-7.0.29/webapps

8、vi /home/ysc/apache-tomcat-7.0.29/webapps/nutch-1.2/WEB-INF/classes/nutch-site.xml

加入以下配置：

<property>

<name>searcher.dir</name>

<value>/home/ysc/workspace/nutch1.2/data</value>

<description>

Path to root of crawl. This directory is searched (in

order) for either the file search-servers.txt, containing a list of

distributed search servers, or the directory "index" containing

merged indexes, or the directory "segments" containing segment

indexes.

</description>

</property>

9、vi /home/ysc/apache-tomcat-7.0.29/conf/server.xml

将

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443"/>

改为

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" URIEncoding="utf-8"/>

10、cd /home/ysc/apache-tomcat-7.0.29/bin

11、./startup.sh

12、访问：<http://localhost:8080/nutch-1.2/>

关于nutch1.2更多的BUG修复及资料，请参看我在CSDN发布的资源：<http://download.csdn.net/user/yangshangchuan>

二、nutch1.5.1

1、下载并解压eclipse（集成开发环境）

下载地址：<http://www.eclipse.org/downloads/>，下载Eclipse IDE for Java EE Developers

2、安装Subclipse插件（SVN客户端）

插件地址：<http://subclipse.tigris.org/update_1.8.x>，

3、安装IvyDE插件（下载依赖Jar）

插件地址：<http://www.apache.org/dist/ant/ivyde/updatesite/>

4、签出代码

File > New > Project > SVN > 从SVN 检出项目

创建新的资源库位置 > URL：<https://svn.apache.org/repos/asf/nutch/tags/release-1.5.1/> > 选中URL > Finish

弹出New Project向导，选择Java Project > Next，输入Project name：nutch1.5.1 > Finish

5、配置构建路径

在左部Package Explorer的 nutch1.5.1文件夹上单击右键 > Build Path > Configure Build Path...

> 选中Source选项 > 选择src > Remove > Add Folder... > 选择src/bin, src/java, src/test 和 src/testresources（对于插件，需要选中src/plugin目录下的每一个插件目录下的src/java ， src/test文件夹） > OK

切换到Libraries选项 >

Add Class Folder... > 选中nutch1.5.1/conf > OK

Add JARs... > 需要选中src/plugin目录下的每一个插件目录下的lib目录下的jar文件 > OK

Add Library... > IvyDE Managed Dependencies > Next > Main > Ivy File > Browse > ivy/ivy.xml > Finish

切换到Order and Export选项>

选中conf > Top

6、执行ANT

在左部Package Explorer的 nutch1.5.1文件夹下的build.xml文件上单击右键 > Run As > Ant Build

在左部Package Explorer的 nutch1.5.1文件夹上单击右键 > Refresh

在左部Package Explorer的 nutch1.5.1文件夹上单击右键 > Build Path > Configure Build Path... > 选中Libraries选项 > Add Class Folder... > 选中build > OK

7、修改配置文件nutch-site.xml 和regex-urlfilter.txt

将nutch-site.xml.template改名为nutch-site.xml

将regex-urlfilter.txt.template改名为regex-urlfilter.txt

在左部Package Explorer的 nutch1.5.1文件夹上单击右键 > Refresh

将如下配置项加入文件nutch-site.xml：

<property>

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

<value>nutch</value>

</property>

<property>

<name>http.content.limit</name>

<value>-1</value>

</property>

修改regex-urlfilter.txt，将

# accept anything else

+.

替换为：

+^http://([a-z0-9]\*\.)\*news.163.com/

-.

8、开发调试

在左部Package Explorer的 nutch1.5.1文件夹上单击右键 > New > Folder > Folder name: urls

在刚新建的urls目录下新建一个文本文件url，文本内容为：http://news.163.com

打开src/java下的org.apache.nutch.crawl.Crawl.java类，单击右键Run As > Run Configurations > Arguments > 在Program arguments输入框中输入: urls -dir data -depth 3 > Run

在需要调试的地方打上断点Debug As > Java Applicaton

9、查看结果

查看segments目录：

打开src/java下的org.apache.nutch.segment.SegmentReader.java类

单击右键Run As > Java Applicaton，控制台会输出该命令的使用方法

单击右键Run As > Run Configurations > Arguments > 在Program arguments输入框中输入: -dump data/segments/\* data/segments/dump

用文本编辑器打开文件data/segments/dump/dump查看segments中存储的信息

查看crawldb目录：

打开src/java下的org.apache.nutch.crawl.CrawlDbReader.java类

单击右键Run As > Java Applicaton，控制台会输出该命令的使用方法

单击右键Run As > Run Configurations > Arguments > 在Program arguments输入框中输入: data/crawldb -stats

控制台会输出 crawldb统计信息

查看linkdb目录：

打开src/java下的org.apache.nutch.crawl.LinkDbReader.java类

单击右键Run As > Java Applicaton，控制台会输出该命令的使用方法

单击右键Run As > Run Configurations > Arguments > 在Program arguments输入框中输入: data/linkdb -dump data/linkdb\_dump

用文本编辑器打开文件data/linkdb\_dump/part-00000查看linkdb中存储的信息

10、全网分步骤抓取

在左部Package Explorer的 nutch1.5.1文件夹下的build.xml文件上单击右键 > Run As > Ant Build

cd /home/ysc/workspace/nutch1.5.1/runtime/local

#准备URL列表

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

gunzip content.rdf.u8.gz

mkdir dmoz

bin/nutch org.apache.nutch.tools.DmozParser content.rdf.u8 -subset 5000 > dmoz/url

#注入URL

bin/nutch inject crawl/crawldb dmoz

#生成抓取列表

bin/nutch generate crawl/crawldb crawl/segments

#第一次抓取

s1=`ls -d crawl/segments/2\* | tail -1`

echo $s1

#抓取网页

bin/nutch fetch $s1

#解析网页

bin/nutch parse $s1

#更新URL状态

bin/nutch updatedb crawl/crawldb $s1

#第二次抓取

bin/nutch generate crawl/crawldb crawl/segments -topN 1000

s2=`ls -d crawl/segments/2\* | tail -1`

echo $s2

bin/nutch fetch $s2

bin/nutch parse $s2

bin/nutch updatedb crawl/crawldb $s2

#第三次抓取

bin/nutch generate crawl/crawldb crawl/segments -topN 1000

s3=`ls -d crawl/segments/2\* | tail -1`

echo $s3

bin/nutch fetch $s3

bin/nutch parse $s3

bin/nutch updatedb crawl/crawldb $s3

#生成反向链接库

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

11、索引和搜索

cd /home/ysc/

wget <http://mirror.bjtu.edu.cn/apache/lucene/solr/3.6.1/apache-solr-3.6.1.tgz>

tar -xvf [apache-solr-3.6.1.tgz](http://mirror.bjtu.edu.cn/apache/lucene/solr/3.6.1/apache-solr-3.6.1.tgz)

cd apache-solr-3.6.1 /example

NUTCH\_RUNTIME\_HOME=/home/ysc/workspace/nutch1.5.1/runtime/local

APACHE\_SOLR\_HOME=/home/ysc/apache-solr-3.6.1

cp ${NUTCH\_RUNTIME\_HOME}/conf/schema.xml ${APACHE\_SOLR\_HOME}/example/solr/conf/

如果需要把网页内容存储到索引中，则修改 schema.xml文件中的

<field name="content" type="text" stored="false" indexed="true"/>

为

<field name="content" type="text" stored="true" indexed="true"/>

修改${APACHE\_SOLR\_HOME}/example/solr/conf/solrconfig.xml,将里面的<str name="df">text</str>都替换为<str name="df">content</str>

把${APACHE\_SOLR\_HOME}/example/solr/conf/schema.xml中的 <schema name="nutch" version="1.5.1">修改为<schema name="nutch" version="1.5">

#启动SOLR服务器

java -jar start.jar

<http://127.0.0.1:8983/solr/admin/>

<http://127.0.0.1:8983/solr/admin/stats.jsp>

cd /home/ysc/workspace/nutch1.5.1/runtime/local

#提交索引

bin/nutch solrindex http://127.0.0.1:8983/solr/ crawl/crawldb -linkdb crawl/linkdb crawl/segments/\*

执行完整crawl:

bin/nutch crawl urls -dir data -depth 2 -topN 100 -solr <http://127.0.0.1:8983/solr/>

使用以下命令分页查看所有索引的文档：

[http://127.0.0.1:8983/solr/select/?q=\*%3A\*&version=2.2&start=0&rows=10&indent=on](http://127.0.0.1:8983/solr/select/?q=*%3A*&version=2.2&start=0&rows=10&indent=on)

标题包含“网易”的文档：

[http://127.0.0.1:8983/solr/select/?q=title%3A%E7%BD%91%E6%98%93&version=2.2&start=0&rows=10&indent=on](http://127.0.0.1:8983/solr/select/?q=title%3A网易&version=2.2&start=0&rows=10&indent=on)

12、查看索引信息

cd /home/ysc/

wget http://luke.googlecode.com/files/lukeall-3.5.0.jar

java -jar lukeall-3.5.0.jar

Path: /home/ysc/apache-solr-3.6.1/example/solr/data

13、配置SOLR的中文分词

cd /home/ysc/

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

unzip mmseg4j-1.8.5.zip -d mmseg4j-1.8.5

APACHE\_SOLR\_HOME=/home/ysc/apache-solr-3.6.1

mkdir $APACHE\_SOLR\_HOME/example/solr/lib

mkdir $APACHE\_SOLR\_HOME/example/solr/dic

cp mmseg4j-1.8.5/mmseg4j-all-1.8.5.jar $APACHE\_SOLR\_HOME/example/solr/lib

cp mmseg4j-1.8.5/data/\*.dic $APACHE\_SOLR\_HOME/example/solr/dic

将${APACHE\_SOLR\_HOME}/example/solr/conf/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"* dicPath=*"/home/ysc/apache-solr-3.6.1/example/solr/dic"*/>

#重新启动SOLR服务器

java -jar start.jar

#重建索引，演示在开发环境中如何操作

打开src/java下的org.apache.nutch.indexer.solr.SolrIndexer.java类

单击右键Run As > Java Applicaton，控制台会输出该命令的使用方法

单击右键Run As > Run Configurations > Arguments > 在Program arguments输入框中输入: http://127.0.0.1:8983/solr/ data/crawldb -linkdb data/linkdb data/segments/\*

使用luke重新打开索引就会发现分词起作用了

三、nutch2.0

nutch2.0和二中的nutch1.5.1的步骤相同，但在8、开发调试之前需要做以下配置：

在左部Package Explorer的 nutch2.0文件夹上单击右键 > New > Folder > Folder name: data并指定数据存储方式，选如下之一：

1、使用mysql作为数据存储

1）、在nutch2.0/conf/nutch-site.xml中加入如下配置：

<property>

<name>storage.data.store.class</name>

<value>org.apache.gora.sql.store.SqlStore</value>

</property>

2）、将nutch2.0/conf/gora.properties文件中的

gora.sqlstore.jdbc.driver=org.hsqldb.jdbc.JDBCDriver

gora.sqlstore.jdbc.url=jdbc:hsqldb:hsql://localhost/nutchtest

gora.sqlstore.jdbc.user=sa

gora.sqlstore.jdbc.password=

修改为

gora.sqlstore.jdbc.driver=com.mysql.jdbc.Driver

gora.sqlstore.jdbc.url=jdbc:mysql://127.0.0.1:3306/nutch2

gora.sqlstore.jdbc.user=root

gora.sqlstore.jdbc.password=ROOT

3）、打开nutch2.0/ivy/ivy.xml中的*mysql-connector-java*依赖

4）、sudo apt-get install mysql-server

2、使用hbase作为数据存储

1）、在nutch2.0/conf/nutch-site.xml中加入如下配置：

<property>

<name>storage.data.store.class</name>

<value>org.apache.gora.hbase.store.HBaseStore</value>

</property>

2）、打开nutch2.0/ivy/ivy.xml中的*gora-hbase*依赖

3）*、*cd /home/ysc

4）、wget [http://mirror.bit.edu.cn/apache/hbase/hbase-0.90.5/hbase-0.90.5.tar.gz](http://mirror.bit.edu.cn/apache/hbase/hbase-0.94.1/hbase-0.94.1.tar.gz)

5）*、*tar -xvf hbase-0.90.5.tar.gz

6）*、*vi hbase-0.90.5/conf/hbase-site.xml

加入以下配置：

<property>

<name>hbase.rootdir</name>

<value>file:///home/ysc/hbase-0.90.5-database</value>

</property>

7)、hbase-0.90.5/bin/start-hbase.sh

8)、将/home/ysc/hbase-0.90.5/hbase-0.90.5.jar加入开发环境eclipse的build path

四、配置SSH

三台机器 devcluster01， devcluster02， devcluster03，分别在每一台机器上面执行如下操作：

1、sudo vi /etc/hosts

加入以下配置：

192.168.1.1 devcluster01

192.168.1.2 devcluster02

192.168.1.3 devcluster03

2、安装SSH服务：

sudo apt-get install openssh-server

3、(有提示的时候回车键确认）

ssh-keygen -t rsa

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。

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

在devcluster01上面执行时，以下两条命令的主机为02和03

在devcluster02上面执行时，以下两条命令的主机为01和03

在devcluster03上面执行时，以下两条命令的主机为01和02

5、ssh-copy-id -i .ssh/id\_rsa.pub ysc@ devcluster02

6、ssh-copy-id -i .ssh/id\_rsa.pub ysc@ devcluster03

以上两条命令实际上是将 .ssh/id\_rsa.pub 公钥文件追加到远程主机 server 的 user 主目录下的 .ssh/authorized\_keys 文件中。

五、安装Hadoop Cluster（伪分布式运行模式）并运行Nutch

步骤和四大同小异，只需要1台机器 devcluster01，所以黄色背景部分全部设置为devcluster01，不需要第11步

六、安装Hadoop Cluster（分布式运行模式）并运行Nutch

三台机器 devcluster01， devcluster02， devcluster03(vi /etc/hostname)

使用用户ysc登陆 devcluster01：

1、cd /home/ysc

2、wget <http://mirrors.tuna.tsinghua.edu.cn/apache/hadoop/common/hadoop-1.1.1/hadoop-1.1.1-bin.tar.gz>

3、tar -xvf hadoop-1.1.1-bin.tar.gz

4、cd hadoop-1.1.1

5、vi conf/masters

替换内容为 ：

devcluster01

6、vi conf/slaves

替换内容为 ：

devcluster02

devcluster03

7、vi conf/core-site.xml

加入配置：

<property>

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

<value>hdfs://devcluster01:9000</value>

<description>

Where to find the Hadoop Filesystem through the network.

Note 9000 is not the default port.

(This is slightly changed from previous versions which didnt have "hdfs")

</description>

</property>

<property>

<name>hadoop.security.authorization</name>

<value>true</value>

</property>

编辑conf/hadoop-policy.xml

8、vi conf/hdfs-site.xml

加入配置：

<property>

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

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

</property>

<property>

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

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

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.block.size</name>

<value>671088640</value>

<description>The default block size for new files.</description>

</property>

9、vi conf/mapred-site.xml

加入配置：

<property>

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

<value>devcluster01:9001</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.reduce.tasks.speculative.execution</name>

<value>false</value>

<description>If true, then multiple instances of some reduce tasks

may be executed in parallel.</description>

</property>

<property>

<name>mapred.map.tasks.speculative.execution</name>

<value>false</value>

<description>If true, then multiple instances of some map tasks

may be executed in parallel.</description>

</property>

<property>

<name>mapred.child.java.opts</name>

<value>-Xmx2000m</value>

</property>

<property>

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

<value>4</value>

<description>

the core number of host

</description>

</property>

<property>

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

<value>4</value>

</property>

<property>

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

<value>4</value>

<description>

define mapred.map tasks to be number of slave hosts.the best number is the number of slave hosts plus the core numbers of per host

</description>

</property>

<property>

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

<value>4</value>

<description>

define mapred.reduce tasks to be number of slave hosts.the best number is the number of slave hosts plus the core numbers of per host

</description>

</property>

<property>

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

<value>BLOCK</value>

<description>If the job outputs are to compressed as SequenceFiles, how should they be compressed? Should be one of NONE, RECORD or BLOCK.

</description>

</property>

<property>

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

<value>true</value>

<description>Should the job outputs be compressed?

</description>

</property>

<property>

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

<value>true</value>

<description>Should the outputs of the maps be compressed before being sent across the network. Uses SequenceFile compression.

</description>

</property>

<property>

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

<value>/home/ysc/mapreduce/system</value>

</property>

<property>

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

<value>/home/ysc/mapreduce/local</value>

</property>

10、vi conf/hadoop-env.sh

追加：

export JAVA\_HOME=/home/ysc/jdk1.7.0\_05

export HADOOP\_HEAPSIZE=2000

#替换掉默认的垃圾回收器，因为默认的垃圾回收器在多线程环境下会有更多的wait等待

export HADOOP\_OPTS="-server -Xmn256m -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=70"

11、复制HADOOP文件

scp -r /home/ysc/hadoop-1.1.1 ysc@devcluster02:/home/ysc/hadoop-1.1.1

scp -r /home/ysc/hadoop-1.1.1 ysc@devcluster03:/home/ysc/hadoop-1.1.1

12、sudo vi /etc/profile

追加并重启系统：

export PATH=/home/ysc/hadoop-1.1.1/bin:$PATH

13、格式化名称节点并启动集群

hadoop namenode -format

start-all.sh

14、cd /home/ysc/workspace/nutch1.5.1/runtime/deploy

mkdir urls

echo [http://news.163.com](http://news.163.com/) > urls/url

hadoop dfs -put urls urls

bin/nutch crawl urls -dir data -depth 2 -topN 100

15、访问 http://localhost:50030 可以查看 JobTracker 的运行状态。访问 http://localhost:50060 可以查看 TaskTracker 的运行状态。访问 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

16、通过stop-all.sh停止集群

17、如果NameNode和SecondaryNameNode不在同一台机器上，则在SecondaryNameNode的conf/hdfs-site.xml文件中加入配置：

<property>

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

<value>namenode:50070</value>

</property>

七、配置Ganglia监控Hadoop集群和HBase集群

1、服务器端（安装到master devcluster01上）

1）、ssh devcluster01

2）、useradd ganglia -g ganglia

3）、sudo apt-get install ganglia-monitor ganglia-webfront gmetad

//补充：在Ubuntu10.04上，ganglia-webfront这个package名字叫ganglia-webfrontend

//如果install出错，则运行sudo apt-get update，如果update出错，则删除出错路径

4）、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”;

5）、sudo apt-get install rrdtool

6)、vi /etc/ganglia/gmetad.conf

在这个配置文件中增加一些datasource，即其他2个被监控的节点，增加以下内容：

data\_source “hadoop-cluster” devcluster01:8649 devcluster02:8649 devcluster03:8649

gridname "Hadoop"

2、数据源端（安装到所有slaves上）

1)、ssh devcluster02

useradd ganglia -g ganglia

sudo apt-get install ganglia-monitor

useradd ganglia -g ganglia

2)、ssh devcluster03

useradd ganglia -g ganglia

sudo apt-get install ganglia-monitor

useradd ganglia -g ganglia

3）、ssh devcluster01

scp /etc/ganglia/gmond.conf devcluster02:/etc/ganglia/gmond.conf

scp /etc/ganglia/gmond.conf devcluster03:/etc/ganglia/gmond.conf

3、配置WEB

1）、ssh devcluster01

2）、sudo ln -s /usr/share/ganglia-webfrontend /var/www/ganglia

3）、vi /etc/apache2/apache2.conf

添加：

ServerName devcluster01

4、重启服务

1）、ssh devcluster02

sudo /etc/init.d/ganglia-monitor restart

ssh devcluster03

sudo /etc/init.d/ganglia-monitor restart

2）、ssh devcluster01

sudo /etc/init.d/ganglia-monitor restart

sudo /etc/init.d/gmetad restart

sudo /etc/init.d/apache2 restart

5、访问页面

http:// devcluster01/ganglia

6、集成hadoop

1）、ssh devcluster01

2）、cd /home/ysc/hadoop-1.1.1

3）、vi conf/hadoop-metrics2.properties

# 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

#广播IP地址，这是缺省的，统一设该值(只能用组播地址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

4）、scp conf/hadoop-metrics2.properties root@devcluster02:/home/ysc/hadoop-1.1.1/conf/hadoop-metrics2.properties

5）、scp conf/hadoop-metrics2.properties [root@devcluster03:/home/ysc/hadoop-1.1.1/conf/hadoop-metrics2.properties](mailto:root@devcluster03:/home/ysc/hadoop-1.0.3/conf/hadoop-metrics2.properties)

6）、stop-all.sh

7）、start-all.sh

7、集成hbase

1）、ssh devcluster01

2）、cd /home/ysc/hbase-0.92.2

3）、vi conf/hadoop-metrics.properties(只能用组播地址239.2.11.71)

hbase.extendedperiod = 3600

hbase.class=org.apache.hadoop.metrics.ganglia.GangliaContext31

hbase.period=10

hbase.servers=239.2.11.71:8649

jvm.class=org.apache.hadoop.metrics.ganglia.GangliaContext31

jvm.period=10

jvm.servers=239.2.11.71:8649

rpc.class=org.apache.hadoop.metrics.ganglia.GangliaContext31

rpc.period=10

rpc.servers=239.2.11.71:8649

4）、scp conf/hadoop-metrics.properties root@devcluster02:/home/ysc/ hbase-0.92.2/conf/hadoop-metrics.properties

5）、scp conf/hadoop-metrics.properties [root@devcluster03:/home/ysc/ hbase-0.92.2/conf/hadoop-metrics.properties](mailto:root@devcluster03:/home/ysc/hadoop-1.0.3/conf/hadoop-metrics2.properties)

6）、stop-hbase.sh

7）、start-hbase.sh

八、Hadoop配置**Snappy**压缩

1、wget http://snappy.googlecode.com/files/snappy-1.0.5.tar.gz

2、tar -xzvf snappy-1.0.5.tar.gz

3、cd snappy-1.0.5

4、./configure

5、make

6、make install

7、scp /usr/local/lib/libsnappy\* devcluster01:/home/ysc/hadoop-1.1.1/lib/native/Linux-amd64-64/

scp /usr/local/lib/libsnappy\* devcluster02:/home/ysc/hadoop-1.1.1/lib/native/Linux-amd64-64/

scp /usr/local/lib/libsnappy\* devcluster03:/home/ysc/hadoop-1.1.1/lib/native/Linux-amd64-64/

8、vi /etc/profile

追加：

export LD\_LIBRARY\_PATH=/home/ysc/hadoop-1.1.1/lib/native/Linux-amd64-64

9、修改mapred-site.xml

<property>

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

<value>BLOCK</value>

<description>If the job outputs are to compressed as SequenceFiles, how should

they be compressed? Should be one of NONE, RECORD or BLOCK.

</description>

</property>

<property>

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

<value>true</value>

<description>Should the job outputs be compressed?

</description>

</property>

<property>

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

<value>true</value>

<description>Should the outputs of the maps be compressed before being

sent across the network. Uses SequenceFile compression.

</description>

</property>

<property>

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

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

<description>If the map outputs are compressed, how should they be

compressed?

</description>

</property>

<property>

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

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

<description>If the job outputs are compressed, how should they be compressed?

</description>

</property>

九、Hadoop配置**Lzo**压缩

1、wget http://www.oberhumer.com/opensource/lzo/download/lzo-2.06.tar.gz

2、tar -zxvf lzo-2.06.tar.gz

3、cd lzo-2.06

4、./configure --enable-shared

5、make

6、make install

7、scp /usr/local/lib/liblzo2.\* devcluster01:/lib/x86\_64-linux-gnu

scp /usr/local/lib/liblzo2.\* devcluster02:/lib/x86\_64-linux-gnu

scp /usr/local/lib/liblzo2.\* devcluster03:/lib/x86\_64-linux-gnu

8、wget <http://hadoop-gpl-compression.apache-extras.org.codespot.com/files/hadoop-gpl-compression-0.1.0-rc0.tar.gz>

9、tar -xzvf hadoop-gpl-compression-0.1.0-rc0.tar.gz

10、cd hadoop-gpl-compression-0.1.0

11、cp lib/native/Linux-amd64-64/\* /home/ysc/hadoop-1.1.1/lib/native/Linux-amd64-64/

12、cp hadoop-gpl-compression-0.1.0.jar /home/ysc/hadoop-1.1.1/lib/(这里hadoop集群的版本要和compression使用的版本一致)

13、scp -r /home/ysc/hadoop-1.1.1/lib devcluster02:/home/ysc/hadoop-1.1.1/

scp -r /home/ysc/hadoop-1.1.1/lib devcluster03:/home/ysc/hadoop-1.1.1/

14、vi /etc/profile

追加：

export LD\_LIBRARY\_PATH=/home/ysc/hadoop-1.1.1/lib/native/Linux-amd64-64

15、修改core-site.xml

<property>

<name>io.compression.codecs</name>

<value>com.hadoop.compression.lzo.LzoCodec,org.apache.hadoop.io.compress.DefaultCodec,org.apache.hadoop.io.compress.GzipCodec,org.apache.hadoop.io.compress.BZip2Codec,org.apache.hadoop.io.compress.SnappyCodec</value>

<description>A list of the compression codec classes that can be used

for compression/decompression.</description>

</property>

<property>

<name>io.compression.codec.lzo.class</name>

<value>com.hadoop.compression.lzo.LzoCodec</value>

</property>

<property>

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

<value>1440</value>

<description>Number of minutes between trash checkpoints.

If zero, the trash feature is disabled.

</description>

</property>

16、修改mapred-site.xml

<property>

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

<value>BLOCK</value>

<description>If the job outputs are to compressed as SequenceFiles, how should

they be compressed? Should be one of NONE, RECORD or BLOCK.

</description>

</property>

<property>

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

<value>true</value>

<description>Should the job outputs be compressed?

</description>

</property>

<property>

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

<value>true</value>

<description>Should the outputs of the maps be compressed before being

sent across the network. Uses SequenceFile compression.

</description>

</property>

<property>

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

<value>com.hadoop.compression.lzo.LzoCodec</value>

<description>If the map outputs are compressed, how should they be

compressed?

</description>

</property>

<property>

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

<value>com.hadoop.compression.lzo.LzoCodec</value>

<description>If the job outputs are compressed, how should they be compressed?

</description>

</property>

十、配置zookeeper集群以运行hbase

1、ssh devcluster01

2、cd /home/ysc

3、wget <http://mirror.bjtu.edu.cn/apache/zookeeper/stable/zookeeper-3.4.5.tar.gz>

4、tar -zxvf zookeeper-3.4.5.tar.gz

5、cd zookeeper-3.4.5

6、cp conf/zoo\_sample.cfg conf/zoo.cfg

7、vi conf/zoo.cfg

修改：dataDir=/home/ysc/zookeeper

添加：

server.1=devcluster01:2888:3888

server.2=devcluster02:2888:3888

server.3=devcluster03:2888:3888

maxClientCnxns=100

8、scp -r zookeeper-3.4.5 devcluster01:/home/ysc

scp -r zookeeper-3.4.5 devcluster02:/home/ysc

scp -r zookeeper-3.4.5 devcluster03:/home/ysc

9、分别在三台机器上面执行：

ssh devcluster01

mkdir /home/ysc/zookeeper（注：dataDir是zookeeper的数据目录，需要手动创建）

echo 1 > /home/ysc/zookeeper/myid

ssh devcluster02

mkdir /home/ysc/zookeeper

echo 2 > /home/ysc/zookeeper/myid

ssh devcluster03

mkdir /home/ysc/zookeeper

echo 3 > /home/ysc/zookeeper/myid

10、分别在三台机器上面执行：

cd /home/ysc/zookeeper-3.4.5

bin/zkServer.sh start

bin/zkCli.sh -server devcluster01:2181

bin/zkServer.sh status

十一、配置Hbase集群以运行nutch-2.1(Region Servers会因为内存的问题宕机)

1、nutch-2.1使用gora-0.2.1， gora-0.2.1使用hbase-0.90.4，hbase-0.90.4和hadoop-1.1.1不兼容，hbase-0.94.4和gora-0.2.1不兼容，hbase-0.92.2没问题。hbase存在系统时间同步的问题，并且误差要再30s以内。

sudo apt-get install ntp

sudo ntpdate -u 210.72.145.44

2、HBase是数据库，会在同一时间使用很多的文件句柄。大多数linux系统使用的默认值1024是不能满足的。还需要修改 hbase 用户的 nproc，在压力下，如果过低会造成 OutOfMemoryError异常。

vi /etc/security/limits.conf

添加：

ysc soft nproc 32000

ysc hard nproc 32000

ysc soft nofile 32768

ysc hard nofile 32768

vi /etc/pam.d/common-session

添加：

session required pam\_limits.so

3、登陆master，下载并解压hbase

ssh devcluster01

cd /home/ysc

wget http://apache.etoak.com/hbase/hbase-0.92.2/hbase-0.92.2.tar.gz

tar -zxvf hbase-0.92.2.tar.gz

cd hbase-0.92.2

4、修改配置文件hbase-env.sh

vi conf/hbase-env.sh

追加：

export JAVA\_HOME=/home/ysc/jdk1.7.0\_05

export HBASE\_MANAGES\_ZK=false

export HBASE\_HEAPSIZE=10000

#替换掉默认的垃圾回收器，因为默认的垃圾回收器在多线程环境下会有更多的wait等待

export HBASE\_OPTS="-server -Xmn256m -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=70"

5、修改配置文件hbase-site.xml

vi conf/hbase-site.xml

<property>

<name>hbase.rootdir</name>

<value>hdfs://devcluster01:9000/hbase</value>

</property>

<property>

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

<value>true</value>

</property>

<property>

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

<value>devcluster01,devcluster02,devcluster03</value>

</property>

<property>

<name>hfile.block.cache.size</name>

<value>0.25</value>

<description>

Percentage of maximum heap (-Xmx setting) to allocate to block cache

used by HFile/StoreFile. Default of 0.25 means allocate 25%.

Set to 0 to disable but it's not recommended.

</description>

</property>

<property>

<name>hbase.regionserver.global.memstore.upperLimit</name>

<value>0.4</value>

<description>Maximum size of all memstores in a region server before new

updates are blocked and flushes are forced. Defaults to 40% of heap

</description>

</property>

<property>

<name>hbase.regionserver.global.memstore.lowerLimit</name>

<value>0.35</value>

<description>When memstores are being forced to flush to make room in

memory, keep flushing until we hit this mark. Defaults to 35% of heap.

This value equal to hbase.regionserver.global.memstore.upperLimit causes

the minimum possible flushing to occur when updates are blocked due to

memstore limiting.

</description>

</property>

<property>

<name>hbase.hregion.majorcompaction</name>

<value>0</value>

<description>The time (in miliseconds) between 'major' compactions of all

HStoreFiles in a region. Default: 1 day.

Set to 0 to disable automated major compactions.

</description>

</property>

6、修改配置文件regionservers

vi conf/regionservers

devcluster01

devcluster02

devcluster03

7、因为HBase建立在Hadoop之上，Hadoop使用的hadoop\*.jar和HBase使用的 必须 一致。所以要将 HBase lib 目录下的hadoop\*.jar替换成Hadoop里面的那个，防止版本冲突。

cp /home/ysc/hadoop-1.1.1/hadoop-core-1.1.1.jar /home/ysc/hbase-0.92.2/lib

rm /home/ysc/hbase-0.92.2/lib/hadoop-core-1.0.3.jar

8、复制文件到regionservers

scp -r /home/ysc/hbase-0.92.2 devcluster01:/home/ysc

scp -r /home/ysc/hbase-0.92.2 devcluster02:/home/ysc

scp -r /home/ysc/hbase-0.92.2 devcluster03:/home/ysc

9、启动hadoop并创建目录

hadoop fs -mkdir /hbase

10、管理HBase集群:

启动初始 HBase 集群：

bin/start-hbase.sh

停止HBase 集群：

bin/stop-hbase.sh

启动额外备份主服务器，可以启动到 9 个备份服务器 (总数10 个)：

bin/local-master-backup.sh start 1

bin/local-master-backup.sh start 2 3

启动更多 regionservers, 支持到 99 个额外regionservers (总100个)：

bin/local-regionservers.sh start 1

bin/local-regionservers.sh start 2 3 4 5

停止备份主服务器:

cat /tmp/hbase-ysc-1-master.pid |xargs kill -9

停止单独 regionserver：

bin/local-regionservers.sh stop 1

使用HBase命令行模式:

bin/hbase shell

11、web界面

[http://devcluster01:60010](http://221.194.43.2:60010)

[http://devcluster01:60030](http://221.194.43.2:60010)

12、如运行nutch2.1则方法一：

cp conf/hbase-site.xml /home/ysc/nutch-2.1/conf

cd /home/ysc/nutch-2.1

ant

cd runtime/deploy

unzip -d apache-nutch-2.1 apache-nutch-2.1.job

rm apache-nutch-2.1.job

cd apache-nutch-2.1

rm lib/hbase-0.90.4.jar

cp /home/ysc/hbase-0.92.2/hbase-0.92.2.jar lib

zip -r ../apache-nutch-2.1.job ./\*

cd ..

rm -r apache-nutch-2.1

13、如运行nutch2.1则方法二：

cp conf/hbase-site.xml /home/ysc/nutch-2.1/conf

cd /home/ysc/nutch-2.1

cp /home/ysc/hbase-0.92.2/hbase-0.92.2.jar lib

ant

cd runtime/deploy

zip -d apache-nutch-2.1.job lib/hbase-0.90.4.jar

启用snappy压缩：

1、vi conf/gora-hbase-mapping.xml

在family上面添加属性：compression="SNAPPY"

2、mkdir /home/ysc/hbase-0.92.2/lib/native/Linux-amd64-64

3、cp /home/ysc/hadoop-1.1.1/lib/native/Linux-amd64-64/\* /home/ysc/hbase-0.92.2/lib/native/Linux-amd64-64

4、vi /home/ysc/hbase-0.92.2/conf/hbase-site.xml

增加：

<property>

<name>hbase.regionserver.codecs</name>

<value>snappy</value>

</property>

十二、配置Accumulo集群以运行nutch-2.1(gora存在BUG)

1、wget http://apache.etoak.com/accumulo/1.4.2/accumulo-1.4.2-dist.tar.gz

2、tar -xzvf accumulo-1.4.2-dist.tar.gz

3、cd accumulo-1.4.2

4、cp conf/examples/3GB/standalone/\* conf

5、vi conf/accumulo-env.sh

export HADOOP\_HOME=/home/ysc/cluster3

export ZOOKEEPER\_HOME=/home/ysc/zookeeper-3.4.5

export JAVA\_HOME=/home/jdk1.7.0\_01

export ACCUMULO\_HOME=/home/ysc/accumulo-1.4.2

6、vi conf/slaves

devcluster01

devcluster02

devcluster03

7、vi conf/masters

devcluster01

8、vi conf/accumulo-site.xml

<property>

<name>instance.zookeeper.host</name>

<value>host6:2181,host8:2181</value>

<description>comma separated list of zookeeper servers</description>

</property>

<property>

<name>logger.dir.walog</name>

<value>walogs</value>

<description>The directory used to store write-ahead logs on the local filesystem. It is possible to specify a comma-separated list of directories.</description>

</property>

<property>

<name>instance.secret</name>

<value>ysc</value>

<description>A secret unique to a given instance that all servers must know in order to communicate with one another.

Change it before initialization. To change it later use ./bin/accumulo org.apache.accumulo.server.util.ChangeSecret [oldpasswd] [newpasswd],

and then update this file.

</description>

</property>

<property>

<name>tserver.memory.maps.max</name>

<value>3G</value>

</property>

<property>

<name>tserver.cache.data.size</name>

<value>50M</value>

</property>

<property>

<name>tserver.cache.index.size</name>

<value>512M</value>

</property>

<property>

<name>trace.password</name>

<!--

change this to the root user's password, and/or change the user below

-->

<value>ysc</value>

</property>

<property>

<name>trace.user</name>

<value>root</value>

</property>

9、bin/accumulo init

10、bin/start-all.sh

11、bin/stop-all.sh

12、web访问：<http://devcluster01:50095/>

修改nutch2.1：

1、cd /home/ysc/nutch-2.1

2、vi conf/gora.properties

增加：

gora.datastore.default=org.apache.gora.accumulo.store.AccumuloStore

gora.datastore.accumulo.mock=false

gora.datastore.accumulo.instance=accumulo

gora.datastore.accumulo.zookeepers=host6,host8

gora.datastore.accumulo.user=root

gora.datastore.accumulo.password=ysc

3、vi conf/nutch-site.xml

增加：

<property>

<name>storage.data.store.class</name>

<value>org.apache.gora.accumulo.store.AccumuloStore</value>

</property>

4、vi ivy/ivy.xml

增加：

<dependency org="org.apache.gora" name="gora-accumulo" rev="0.2.1" conf="\*->default" />

5、升级accumulo

cp /home/ysc/accumulo-1.4.2/lib/accumulo-core-1.4.2.jar /home/ysc/nutch-2.1/lib

cp /home/ysc/accumulo-1.4.2/lib/accumulo-start-1.4.2.jar /home/ysc/nutch-2.1/lib

cp /home/ysc/accumulo-1.4.2/lib/cloudtrace-1.4.2.jar /home/ysc/nutch-2.1/lib

6、ant

7、cd runtime/deploy

8、删除旧jar

zip -d apache-nutch-2.1.job lib/accumulo-core-1.4.0.jar

zip -d apache-nutch-2.1.job lib/accumulo-start-1.4.0.jar

zip -d apache-nutch-2.1.job lib/cloudtrace-1.4.2.jar

十三、配置Cassandra 集群以运行nutch-2.1（Cassandra 采用去中心化结构）

1、vi /etc/hosts（注意：需要登录到每一台机器上面，将localhost解析到实际地址）

192.168.1.1 localhost

2、wget http://labs.mop.com/apache-mirror/cassandra/1.2.0/apache-cassandra-1.2.0-bin.tar.gz

3、tar -xzvf apache-cassandra-1.2.0-bin.tar.gz

4、cd apache-cassandra-1.2.0

5、vi conf/cassandra-env.sh

增加：

MAX\_HEAP\_SIZE="4G"

HEAP\_NEWSIZE="800M"

6、vi conf/log4j-server.properties

修改：

log4j.appender.R.File=/home/ysc/cassandra/system.log

7、vi conf/cassandra.yaml

修改：

cluster\_name: 'Cassandra Cluster'

data\_file\_directories:

- /home/ysc/cassandra/data

commitlog\_directory: /home/ysc/cassandra/commitlog

saved\_caches\_directory: /home/ysc/cassandra/saved\_caches

- seeds: "192.168.1.1"

listen\_address: 192.168.1.1

rpc\_address: 192.168.1.1

thrift\_framed\_transport\_size\_in\_mb: 1023

thrift\_max\_message\_length\_in\_mb: 1024

8、vi bin/stop-server

增加：

user=`whoami`

pgrep -u $user -f cassandra | xargs kill -9

9、复制cassandra到其他节点：

cd ..

scp -r apache-cassandra-1.2.0 devcluster02:/home/ysc

scp -r apache-cassandra-1.2.0 devcluster03:/home/ysc

分别在devcluster02和devcluster03上面修改：

vi conf/cassandra.yaml

listen\_address: 192.168.1.2

rpc\_address: 192.168.1.2

vi conf/cassandra.yaml

listen\_address: 192.168.1.3

rpc\_address: 192.168.1.3

10、分别在3个节点上面运行

bin/cassandra

bin/cassandra -f  参数 -f 的作用是让 Cassandra 以前端程序方式运行，这样有利于调试和观察日志信息，而在实际生产环境中这个参数是不需要的（即 Cassandra 会以 daemon 方式运行）

11、bin/nodetool -host devcluster01 ring

bin/nodetool -host devcluster01 info

12、bin/stop-server

13、bin/cassandra-cli

修改nutch2.1：

1、cd /home/ysc/nutch-2.1

2、vi conf/gora.properties

增加：

gora.cassandrastore.servers=host2:9160,host6:9160,host8:9160

3、vi conf/nutch-site.xml

增加：

<property>

<name>storage.data.store.class</name>

<value>org.apache.gora.cassandra.store.CassandraStore</value>

</property>

4、vi ivy/ivy.xml

增加：

<dependency org="org.apache.gora" name="gora-cassandra" rev="0.2.1" conf="\*->default" />

5、升级cassandra

cp /home/ysc/apache-cassandra-1.2.0/lib/apache-cassandra-1.2.0.jar /home/ysc/nutch-2.1/lib

cp /home/ysc/apache-cassandra-1.2.0/lib/apache-cassandra-thrift-1.2.0.jar /home/ysc/nutch-2.1/lib

cp /home/ysc/apache-cassandra-1.2.0/lib/jline-1.0.jar /home/ysc/nutch-2.1/lib

6、ant

7、cd runtime/deploy

8、删除旧jar

zip -d apache-nutch-2.1.job lib/cassandra-thrift-1.1.2.jar

zip -d apache-nutch-2.1.job lib/jline-0.9.1.jar

十四、配置MySQL 单机服务器以运行nutch-2.1

1、apt-get install mysql-server mysql-client

2、vi /etc/mysql/my.cnf

修改：

bind-address = 221.194.43.2

在[client]下增加：

default-character-set=utf8

在[mysqld]下增加：

default-character-set=utf8

3、mysql –uroot –pysc

SHOW VARIABLES LIKE '%character%';

4、service mysql restart

5、mysql –uroot –pysc

GRANT ALL PRIVILEGES ON \*.\* TO root@"%" IDENTIFIED BY "ysc";

6、vi conf/gora-sql-mapping.xml

修改字段的长度

<primarykey column="id" length="333"/>

<field name="content" column="content" />

<field name="text" column="text" length="19892"/>

7、启动nutch之后登陆mysql

ALTER TABLE webpage MODIFY COLUMN content MEDIUMBLOB;

ALTER TABLE webpage MODIFY COLUMN text MEDIUMTEXT;

ALTER TABLE webpage MODIFY COLUMN title MEDIUMTEXT;

ALTER TABLE webpage MODIFY COLUMN reprUrl MEDIUMTEXT;

ALTER TABLE webpage MODIFY COLUMN baseUrl MEDIUMTEXT;

ALTER TABLE webpage MODIFY COLUMN typ MEDIUMTEXT;

ALTER TABLE webpage MODIFY COLUMN inlinks MEDIUMBLOB;

ALTER TABLE webpage MODIFY COLUMN outlinks MEDIUMBLOB;

修改nutch2.1：

1、cd /home/ysc/nutch-2.1

2、vi conf/gora.properties

增加：

gora.sqlstore.jdbc.driver=com.mysql.jdbc.Driver

gora.sqlstore.jdbc.url=jdbc:mysql://host2:3306/nutch?createDatabaseIfNotExist=true&useUnicode=true&characterEncoding=utf8

gora.sqlstore.jdbc.user=root

gora.sqlstore.jdbc.password=ysc

3、vi conf/nutch-site.xml

增加：

<property>

<name>storage.data.store.class</name>

<value>org.apache.gora.sql.store.SqlStore </value>

</property>

<property>

<name>encodingdetector.charset.min.confidence</name>

<value>1</value>

<description>A integer between 0-100 indicating minimum confidence value

for charset auto-detection. Any negative value disables auto-detection.

</description>

</property>

4、vi ivy/ivy.xml

增加：

<dependency org="mysql" name="mysql-connector-java" rev="5.1.18" conf="\*->default"/>

十五、nutch2.1 使用DataFileAvroStore作为数据源

1、cd /home/ysc/nutch-2.1

2、vi conf/gora.properties

增加：

gora.datafileavrostore.output.path=datafileavrostore

gora.datafileavrostore.input.path=datafileavrostore

3、vi conf/nutch-site.xml

增加：

<property>

<name>storage.data.store.class</name>

<value>org.apache.gora.avro.store.DataFileAvroStore</value>

</property>

<property>

<name>encodingdetector.charset.min.confidence</name>

<value>1</value>

<description>A integer between 0-100 indicating minimum confidence value

for charset auto-detection. Any negative value disables auto-detection.

</description>

</property>

十六、nutch2.1 使用AvroStore作为数据源

1、cd /home/ysc/nutch-2.1

2、vi conf/gora.properties

增加：

gora.avrostore.codec.type=BINARY

gora.avrostore.input.path=avrostore

gora.avrostore.output.path=avrostore

3、vi conf/nutch-site.xml

增加：

<property>

<name>storage.data.store.class</name>

<value>org.apache.gora.avro.store.AvroStore</value>

</property>

<property>

<name>encodingdetector.charset.min.confidence</name>

<value>1</value>

<description>A integer between 0-100 indicating minimum confidence value

for charset auto-detection. Any negative value disables auto-detection.

</description>

</property>

十七、配置SOLR

配置tomcat：

1、wget http://www.fayea.com/apache-mirror/tomcat/tomcat-7/v7.0.35/bin/apache-tomcat-7.0.35.tar.gz

2、tar -xzvf apache-tomcat-7.0.35.tar.gz

3、cd apache-tomcat-7.0.35

4、vi conf/server.xml

增加URIEncoding="UTF-8"：

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" URIEncoding="UTF-8"/>

5、mkdir conf/Catalina

6、mkdir conf/Catalina/localhost

7、vi conf/Catalina/localhost/solr.xml

增加：

<Context path="/solr">

<Environment name="solr/home" type="java.lang.String" value="/home/ysc/solr/configuration/" override="false"/>

</Context>

8、cd ..

下载SOLR:

1、wget http://mirrors.tuna.tsinghua.edu.cn/apache/lucene/solr/4.1.0/solr-4.1.0.tgz

2、tar -xzvf solr-4.1.0.tgz

复制资源：

1、mkdir /home/ysc/solr

2、cp -r solr-4.1.0/example/solr /home/ysc/solr/configuration

3、unzip solr-4.1.0/example/webapps/solr.war -d /home/ysc/apache-tomcat-7.0.35/webapps/solr

配置nutch：

1、复制schema：

cp /home/ysc/nutch-1.6/conf/schema-solr4.xml /home/ysc/solr/configuration/collection1/conf/schema.xml

2、vi /home/ysc/solr/configuration/collection1/conf/schema.xml

在<fields>下增加：

<field name="\_version\_" type="long" indexed="true" stored="true"/>

配置中文分词：

1、wget http://mmseg4j.googlecode.com/files/mmseg4j-1.9.1.v20130120-SNAPSHOT.zip

2、unzip mmseg4j-1.9.1.v20130120-SNAPSHOT.zip

3、cp mmseg4j-1.9.1-SNAPSHOT/dist/\* /home/ysc/apache-tomcat-7.0.35/webapps/solr/WEB-INF/lib

4、unzip mmseg4j-1.9.1-SNAPSHOT/dist/mmseg4j-core-1.9.1-SNAPSHOT.jar -d mmseg4j-1.9.1-SNAPSHOT/dist/mmseg4j-core-1.9.1-SNAPSHOT

5、mkdir /home/ysc/dic

6、cp mmseg4j-1.9.1-SNAPSHOT/dist/mmseg4j-core-1.9.1-SNAPSHOT/data/\* /home/ysc/dic

7、vi /home/ysc/solr/configuration/collection1/conf/schema.xml

将文件中的

<tokenizer class="solr.WhitespaceTokenizerFactory"/>

和

<tokenizer class="solr.StandardTokenizerFactory"/>

替换为

<tokenizer class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory" mode="complex" dicPath="/home/ysc/dic"/>

配置tomcat本地库：

1、wget http://apache.spd.co.il/apr/apr-1.4.6.tar.gz

2、tar -xzvf apr-1.4.6.tar.gz

3、cd apr-1.4.6

4、./configure

5、make

6、make install

1、wget http://mirror.bjtu.edu.cn/apache/apr/apr-util-1.5.1.tar.gz

2、tar -xzvf apr-util-1.5.1.tar.gz

3、cd apr-util-1.5.1

4、./configure --with-apr=/usr/local/apr

5、make

6、make install

1、wget http://mirror.bjtu.edu.cn/apache//tomcat/tomcat-connectors/native/1.1.24/source/tomcat-native-1.1.24-src.tar.gz

2、tar -zxvf tomcat-native-1.1.24-src.tar.gz

3、cd tomcat-native-1.1.24-src/jni/native

4、./configure --with-apr=/usr/local/apr \

--with-java-home=/home/ysc/jdk1.7.0\_01 \

--with-ssl=no \

--prefix=/home/ysc/apache-tomcat-7.0.35

5、make

6、make install

7、vi /etc/profile

增加：

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/home/ysc/apache-tomcat-7.0.35/lib:/usr/local/apr/lib

8、source /etc/profile

启动tomcat：

cd apache-tomcat-7.0.35

bin/catalina.sh start

[http://devcluster01:8080/solr/](http://host8:8080/solr/)

十八、Nagios监控

服务端：

1、apt-get install apache2 nagios3 nagios-nrpe-plugin

输入密码：nagiosadmin

2、apt-get install nagios3-doc

3、vi /etc/nagios3/conf.d/hostgroups\_nagios2.cfg

define hostgroup {

hostgroup\_name nagios-servers

alias nagios servers

members devcluster01,devcluster02,devcluster03

}

4、cp /etc/nagios3/conf.d/localhost\_nagios2.cfg /etc/nagios3/conf.d/devcluster01\_nagios2.cfg

vi /etc/nagios3/conf.d/devcluster01\_nagios2.cfg

替换：

g/localhost/s//devcluster01/g

g/127.0.0.1/s//192.168.1.1/g

5、cp /etc/nagios3/conf.d/localhost\_nagios2.cfg /etc/nagios3/conf.d/devcluster02\_nagios2.cfg

vi /etc/nagios3/conf.d/devcluster02\_nagios2.cfg

替换：

g/localhost/s//devcluster02/g

g/127.0.0.1/s//192.168.1.2/g

6、cp /etc/nagios3/conf.d/localhost\_nagios2.cfg /etc/nagios3/conf.d/devcluster03\_nagios2.cfg

vi /etc/nagios3/conf.d/devcluster03\_nagios2.cfg

替换：

g/localhost/s//devcluster03/g

g/127.0.0.1/s//192.168.1.3/g

7、vi /etc/nagios3/conf.d/services\_nagios2.cfg

将hostgroup\_name改为nagios-servers

增加：

# check that web services are running

define service {

hostgroup\_name nagios-servers

service\_description HTTP

check\_command check\_http

use generic-service

notification\_interval 0 ; set > 0 if you want to be renotified

}

# check that ssh services are running

define service {

hostgroup\_name nagios-servers

service\_description SSH

check\_command check\_ssh

use generic-service

notification\_interval 0 ; set > 0 if you want to be renotified

}

8、vi /etc/nagios3/conf.d/extinfo\_nagios2.cfg

将hostgroup\_name改为nagios-servers

增加：

define hostextinfo{

hostgroup\_name nagios-servers

notes nagios-servers

# notes\_url http://webserver.localhost.localdomain/hostinfo.pl?host=netware1

icon\_image base/debian.png

icon\_image\_alt Debian GNU/Linux

vrml\_image debian.png

statusmap\_image base/debian.gd2

}

9、sudo /etc/init.d/nagios3 restart

10、访问http://devcluster01/nagios3/

用户名：nagiosadmin密码：nagiosadmin

监控端：

1、apt-get install nagios-nrpe-server

2、vi /etc/nagios/nrpe.cfg

替换：

g/127.0.0.1/s//192.168.1.1/g

3、sudo /etc/init.d/nagios-nrpe-server restart

十九、配置Splunk

1、wget http://download.splunk.com/releases/5.0.2/splunk/linux/splunk-5.0.2-149561-Linux-x86\_64.tgz

2、tar -zxvf splunk-5.0.2-149561-Linux-x86\_64.tgz

3、cd splunk

4、bin/splunk start --answer-yes --no-prompt --accept-license

5、访问http://devcluster01:8000

用户名：admin 密码：changeme

6、添加数据 -> 从 UDP 端口 -> UDP 端口 \*: 1688 -> 来源类型 从列表 log4j -> 保存

7、配置hadoop

vi /home/ysc/hadoop-1.1.1/conf/log4j.properties

修改：

log4j.rootLogger=${hadoop.root.logger}, EventCounter, SYSLOG

增加：

log4j.appender.SYSLOG=org.apache.log4j.net.SyslogAppender

log4j.appender.SYSLOG.facility=local1

log4j.appender.SYSLOG.layout=org.apache.log4j.PatternLayout

log4j.appender.SYSLOG.layout.ConversionPattern=%p %c{2}: %m%n

log4j.appender.SYSLOG.SyslogHost=host6:1688

log4j.appender.SYSLOG.threshold=INFO

log4j.appender.SYSLOG.Header=true

log4j.appender.SYSLOG.FacilityPrinting=true

8、配置hbase

vi /home/ysc/hbase-0.92.2/conf/log4j.properties

修改：

log4j.rootLogger=${hbase.root.logger},SYSLOG

增加：

log4j.appender.SYSLOG=org.apache.log4j.net.SyslogAppender

log4j.appender.SYSLOG.facility=local1

log4j.appender.SYSLOG.layout=org.apache.log4j.PatternLayout

log4j.appender.SYSLOG.layout.ConversionPattern=%p %c{2}: %m%n

log4j.appender.SYSLOG.SyslogHost=host6:1688

log4j.appender.SYSLOG.threshold=INFO

log4j.appender.SYSLOG.Header=true

log4j.appender.SYSLOG.FacilityPrinting=true

9、配置nutch

vi /home/lanke/ysc/nutch-2.1-hbase/conf/log4j.properties

修改：

log4j.rootLogger=INFO,DRFA,SYSLOG

增加：

log4j.appender.SYSLOG=org.apache.log4j.net.SyslogAppender

log4j.appender.SYSLOG.facility=local1

log4j.appender.SYSLOG.layout=org.apache.log4j.PatternLayout

log4j.appender.SYSLOG.layout.ConversionPattern=%p %c{2}: %m%n

log4j.appender.SYSLOG.SyslogHost=host6:1688

log4j.appender.SYSLOG.threshold=INFO

log4j.appender.SYSLOG.Header=true

log4j.appender.SYSLOG.FacilityPrinting=true

10、启动hadoop和hbase

start-all.sh

start-hbase.sh

二十、配置Pig

1、wget http://labs.mop.com/apache-mirror/pig/pig-0.11.0/pig-0.11.0.tar.gz

2、tar -xzvf pig-0.11.0.tar.gz

3、cd pig-0.11.0

4、vi /etc/profile

增加：

export PIG\_HOME=/home/ysc/pig-0.11.0

export PATH=$PIG\_HOME/bin:$PATH

5、source /etc/profile

6、cp conf/log4j.properties.template conf/log4j.properties

7、vi conf/log4j.properties

8、pig

二十一、配置Hive

1、wget http://mirrors.cnnic.cn/apache/hive/hive-0.10.0/hive-0.10.0.tar.gz

2、tar -xzvf hive-0.10.0.tar.gz

3、cd hive-0.10.0

4、vi /etc/profile

增加：

export HIVE\_HOME=/home/ysc/hive-0.10.0

export PATH=$HIVE\_HOME/bin:$PATH

5、source /etc/profile

6、cp conf/hive-log4j.properties.template conf/hive-log4j.properties

7、vi conf/hive-log4j.properties

替换：

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

为：

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

二十二、配置Hadoop2.x集群

1、wget http://labs.mop.com/apache-mirror/hadoop/common/hadoop-2.0.2-alpha/hadoop-2.0.2-alpha.tar.gz

2、tar -xzvf hadoop-2.0.2-alpha.tar.gz

3、cd hadoop-2.0.2-alpha

4、vi etc/hadoop/hadoop-env.sh

追加：

export JAVA\_HOME=/home/ysc/jdk1.7.0\_05

export HADOOP\_HEAPSIZE=2000

5、vi etc/hadoop/core-site.xml

<property>

<name>fs.defaultFS</name>

<value>hdfs://devcluster01:9000</value>

<description>

Where to find the Hadoop Filesystem through the network.

Note 9000 is not the default port.

(This is slightly changed from previous versions which didnt have "hdfs")

</description>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131072</value>

<description>The size of buffer for use in sequence files.

The size of this buffer should probably be a multiple of hardware

page size (4096 on Intel x86), and it determines how much data is

buffered during read and write operations.</description>

</property>

6、vi etc/hadoop/mapred-site.xml

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapred.job.reduce.input.buffer.percent</name>

<value>1</value>

<description>The percentage of memory- relative to the maximum heap size- to

retain map outputs during the reduce. When the shuffle is concluded, any

remaining map outputs in memory must consume less than this threshold before

the reduce can begin.

</description>

</property>

<property>

<name>mapred.job.shuffle.input.buffer.percent</name>

<value>1</value>

<description>The percentage of memory to be allocated from the maximum heap

size to storing map outputs during the shuffle.

</description>

</property>

<property>

<name>mapred.inmem.merge.threshold</name>

<value>0</value>

<description>The threshold, in terms of the number of files

for the in-memory merge process. When we accumulate threshold number of files

we initiate the in-memory merge and spill to disk. A value of 0 or less than

0 indicates we want to DON'T have any threshold and instead depend only on

the ramfs's memory consumption to trigger the merge.

</description>

</property>

<property>

<name>io.sort.factor</name>

<value>100</value>

<description>The number of streams to merge at once while sorting

files. This determines the number of open file handles.</description>

</property>

<property>

<name>io.sort.mb</name>

<value>240</value>

<description>The total amount of buffer memory to use while sorting

files, in megabytes. By default, gives each merge stream 1MB, which

should minimize seeks.</description>

</property>

<property>

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

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

<description>If the map outputs are compressed, how should they be

compressed?

</description>

</property>

<property>

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

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

<description>If the job outputs are compressed, how should they be compressed?

</description>

</property>

<property>

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

<value>BLOCK</value>

<description>If the job outputs are to compressed as SequenceFiles, how should

they be compressed? Should be one of NONE, RECORD or BLOCK.

</description>

</property>

<property>

<name>mapred.child.java.opts</name>

<value>-Xmx2000m</value>

</property>

<property>

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

<value>true</value>

<description>Should the job outputs be compressed?

</description>

</property>

<property>

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

<value>true</value>

<description>Should the outputs of the maps be compressed before being

sent across the network. Uses SequenceFile compression.

</description>

</property>

<property>

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

<value>5</value>

</property>

<property>

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

<value>15</value>

</property>

<property>

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

<value>5</value>

<description>

define mapred.map tasks to be number of slave hosts.the best number is the number of slave hosts plus the core numbers of per host

</description>

</property>

<property>

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

<value>15</value>

<description>

define mapred.reduce tasks to be number of slave hosts.the best number is the number of slave hosts plus the core numbers of per host

</description>

</property>

<property>

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

<value>/home/ysc/mapreduce/system</value>

</property>

<property>

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

<value>/home/ysc/mapreduce/local</value>

</property>

<property>

<name>mapreduce.job.counters.max</name>

<value>12000</value>

<description>Limit on the number of counters allowed per job.

</description>

</property>

7、vi etc/hadoop/yarn-site.xml

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>devcluster01:8031</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>devcluster01:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>devcluster01:8030</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>devcluster01:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>devcluster01:8088</value>

</property>

<property>

<description>Classpath for typical applications.</description>

<name>yarn.application.classpath</name>

<value>

$HADOOP\_CONF\_DIR,

$HADOOP\_COMMON\_HOME/\*,$HADOOP\_COMMON\_HOME/lib/\*,

$HADOOP\_HDFS\_HOME/\*,$HADOOP\_HDFS\_HOME/lib/\*,

$HADOOP\_MAPRED\_HOME/\*,$HADOOP\_MAPRED\_HOME/lib/\*,

$YARN\_HOME/\*,$YARN\_HOME/lib/\*

</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce.shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.nodemanager.local-dirs</name> <value>/home/ysc/h2/data/1/yarn/local,/home/ysc/h2/data/2/yarn/local,/home/ysc/h2/data/3/yarn/local</value>

</property>

<property>

<name>yarn.nodemanager.log-dirs</name> <value>/home/ysc/h2/data/1/yarn/logs,/home/ysc/h2/data/2/yarn/logs,/home/ysc/h2/data/3/yarn/logs</value>

</property>

<property>

<description>Where to aggregate logs</description>

<name>yarn.nodemanager.remote-app-log-dir</name>

<value>/home/ysc/h2/var/log/hadoop-yarn/apps</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>devcluster01:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>devcluster01:19888</value>

</property>

8、vi etc/hadoop/hdfs-site.xml

<property>

<name>dfs.permissions.superusergroup</name>

<value>root</value>

</property>

<property>

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

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

</property>

<property>

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

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

</property>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<name>dfs.block.size</name>

<value>6710886400</value>

<description>The default block size for new files.</description>

</property>

9、启动hadoop

bin/hdfs namenode -format

sbin/start-dfs.sh

sbin/start-yarn.sh

10、访问管理页面

http://devcluster01:8088

http://devcluster01:50070