|  |  |  |  |
| --- | --- | --- | --- |
| NUTCH | Done |  | Crawler |
| HDFS | Done |  | File System |
| HBase |  |  | Key Value Stores |
| SOLR |  |  | Indexing |
| TIKA |  |  | Parsing |
| KIBANA |  |  | Analytics |

**Distributed Search Engine with Nutch, Hadoop,Hbase and Elasticsearch**

Posted on [2016年4月11日](http://www.wallen.top/index.php/2016/04/11/search-engine/) [wallen](http://www.wallen.top/index.php/author/wallena3/)Posted in [未分类](http://www.wallen.top/index.php/category/uncategorized/)

使用Nutch，Hadoop，Hbase以及ElasticSearch构建分布式搜索引擎

All in all,we use the ***Pseudo-Distributed Method*** to implement the Search Engine.By the way, this artical is for ‘apt-get’ system linux,but ‘yum’ system can also reference this.We assume you already configurate the java correctly,or you can see my article about [configutate java on linux](http://www.wallen.top/index.php/2016/02/11/linux_java/" \t "_blank).

In this project,we use:

Apache Hadoop 2.5.2

Apache Nutch 2.3.1

Apche Hbase 0.98.18-hadoop2

ElasticSearch 1.4.4

Kibana 4.0.1

You need notice that the abrove productions’s version must be adhere strictly.Because all the productions should be followed with nutch.You can see [the nutch official website](http://nutch.apache.org/).Otherwise you will get lost’s of errors.

**1:Configurate Hadoop 2.5.2**

**1.1:Change the ssh to login automatically**

if you haven’t got the ssh, use the following command to install it.

$ apt-get install ssh

then,use the following command to change the ssh to login automatically

$ ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa

$ cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

$ chmod 0600 ~/.ssh/authorized\_keys

**1.2:Edit $HADOOP\_HOME/etc/hadoop/hadoop-env**

($HADOOP\_HOME mean’s your hadoop directory’s location.Such as

/home/walle/Documents/hadoop-2.5.2

)

export JAVA\_HOME=your jdk path

**1.3:Edit $HADOOP\_HOME/etc/hadoop/core-site.xml**

<configuration>

<property>

        <name>fs.defaultFS</name>

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

    </property>

    <property>

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

<value>file:/home/walle/Documents/hadoop-2.5.2/tmp</value>

</property>

</configuration>

NOTICE:The Hadoop official said that only configurate the ‘fs.defaultFS’ is OK,but if you don’t configurate the ‘hadoop.tmp.dir’,Hadoop will use the defaul directory which is /tmp/hadoo-hadoop.And in some linux distribution,this directory will be cleaned and we must format again.So we configurate this property.

**1.4:Edit $HADOOP\_HOME/etc/hadoop/hdfs-site.xml**

<configuration>

    <property>

        <name>dfs.replication</name>

        <value>1</value>

    </property>

<property>

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

<value>file:/home/walle/Documents/hadoop-2.5.2/dfs/name</value>

</property>

<property>

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

<value>file:/home/walle/Documents/hadoop-2.5.2/dfs/data</value>

</property>

</configuration>

NOTICE:We assign the ‘dfs.namenode.name.dir’ and the ‘dfs.namenode.data.dir’ at the sametime.Otherwise in the following steps will cause some mistakes.

**1.5:Format the file system**

In the Hadoop root directory:

$ bin/hdfs namenode -format

**1.6:Start NameNode daemon and DataNode daemon**

$ sbin/start-dfs.sh

And now you can browse the web interface for the NameNode,by default it is avaliable at:http://localhost:50070

**1.7:Make the HDFS directories required to execute MapReduce jobs:**

$ bin/hdfs dfs -mkdir /user

$ bin/hdfs dfs -mkdir /user/<username>

**1.8:Edit $HADOOP\_HOME/etc/hadoop/mapred-site.xml**

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

**1.9:Edit $HADOOP\_HOME/etc/hadoop/yarn-site.xml**

<configuration>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

</configuration>

**1.10:Start ResourceManager daemon and NodeManager daemon**

$ sin/start-yarn.sh

And now you can browse the web interface for the ResourceManager,by default it is available at:http://localhost:8088

If you want to stop the hdfs system and yarn,you can use the commands below

$ sbin/stop-dfs.sh

$ sbin/stop-yarn.sh

**1.11:Some operation about hdfs system**

hadoop fs

查看Hadoop HDFS支持的所有命令

hadoop fs –ls

列出目录及文件信息

hadoop fs –lsr

循环列出目录、子目录及文件信息

hadoop fs –put test.txt /user/sunlightcs

将本地文件系统的test.txt复制到HDFS文件系统的/user/sunlightcs目录下

hadoop fs –copyFromLocal test.txt /user/sunlightcs/test.txt

从本地文件系统复制文件到HDFS文件系统，等同于put命令

hadoop fs –copyToLocal /user/sunlightcs/test.txt test.txt

从HDFS文件系统复制文件到本地文件系统，等同于get命令

**2:Configurate the Hbase-0.98.18-hadoop2**

**2.1:Edit $HBASE\_HOME/conf/hbase-env.sh**

 export JAVA\_HOME=your jdk path

**2.2:Edit $HBASE\_HOME/conf/hbase-site.xml**

<configuration>

<property>

    <name>hbase.rootdir</name>

    <!--<value>file:///home/walle/Documents/hbase</value> if you want to run hbase without hadoop use this line-->

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

  </property>

 <property>

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

    <value>/home/walle/Documents/zookeeper</value>

  </property>

<property>

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

  <value>true</value>

</property>

</configuration>

**2.3:Start the Hbase**

$ bin/start-hbase.sh

and use the following command

$ ./bin/hadoop fs -ls /hbase

you will see some information like this

Found 7 items

drwxr-xr-x - hbase users 0 2014-06-25 18:58 /hbase/.tmp

drwxr-xr-x - hbase users 0 2014-06-25 21:49 /hbase/WALs

drwxr-xr-x - hbase users 0 2014-06-25 18:48 /hbase/corrupt

drwxr-xr-x - hbase users 0 2014-06-25 18:58 /hbase/data

-rw-r--r-- 3 hbase users 42 2014-06-25 18:41 /hbase/hbase.id

-rw-r--r-- 3 hbase users 7 2014-06-25 18:41 /hbase/hbase.version

drwxr-xr-x - hbase users 0 2014-06-25 21:49 /hbase/oldWALs

finally,type command

$ jps

and you will see and then you succeed.

4507 NameNode

22581 NodeManager

4661 DataNode

23986 Jps

22255 ResourceManager

23218 HMaster

23147 HQuorumPeer

4853 SecondaryNameNode

23419 HRegionServer

**2.4:Stop the Hbase**

$ bin/stop-hbase.sh

**3:Configurate ElasticSearch 1.4.4**

As I know,Nutch 2.3.1 can work with Es 1.4.4 but can not work with Es 2.x edition.

**3.1:Start the ElasticSearch**

$ bin/elasticsearch -d

the ‘-d’ command is used to run the es in the background

**3.2:check elasticsearch is work or not**

go to http://localhost:9200 to check it.you can see the following information

{

"status" : 200,

"name" : "Margo Damian",

"cluster\_name" : "elasticsearch",

"version" : {

"number" : "1.4.4",

"build\_hash" : "c88f77ffc81301dfa9dfd81ca2232f09588bd512",

"build\_timestamp" : "2015-02-19T13:05:36Z",

"build\_snapshot" : false,

"lucene\_version" : "4.10.3"

},

"tagline" : "You Know, for Search"

}

**3.3:stop the ElasticSearch**

curl -XPOST http://localhost:9200/\_cluster/nodes/\_shutdown

**4:Configurate kibana 4.0.1**

Because we use the ElasticSearch1.4.4 so we must use kibana which edition is 4.0.1 . Kibana 4.5.0 can not work with ElasticSearch 1.4.4

**4.1:Run Kibana**

$ bin/kibana

**4.2:check kibana is work or not**

go to http://localhost:5601 to check it.

**5:Configurate Nutch 2.3.1**

**5.1???:libs preparation**

you need put the files under ‘$HADOOP\_HOME/lib/native’ to ‘$NUTCH\_HOME/lib/native’

**5.2:ant preparation**

Because the internet connection between China and maven’s repository is very slow,we need to change the $NUTCH\_HOME/ivy/ivysettings.xml

find the codes like this

<property name="repo.maven.org"

value="http://repo1.maven.org/maven2/"

override="false"/>

Change the value to

http://maven.oschina.net/content/groups/public/

This is the a chinese website oschina.They offer a mirror site of maven.

WARNING:OSCHINA IS DOWN NOW

**5.3:Add Hadoop to path**

add your HADOOP\_HOME to your /etc/profile PATH  like:

#Hadoop Enviromnet path

export HADOOP\_HOME=/home/walle/Documents/hadoop-2.5.2

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

**5.4:Edit conf/hbase-site.xml**

<configuration>

<property>

<name>hbase.rootdir</name>

<!–<value>file:///home/walle/Documents/hbase</value>–>

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

</property>

</configuration>

**5.5:Edit conf/nutch-site.xml**

<configuration>

<property>

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

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

<description>Default class for storing data</description>

</property>

<property>

<name>plugin.includes</name>

<!-- do \*\*NOT\*\* enable the parse-html plugin, if you want proper HTML parsing. Use something like parse-tika! -->

<value>protocol-httpclient|urlfilter-regex|parse-(text|tika|js)|index-(basic|anchor)|query-(basic|site|url)|response-(json|xml)|summary-basic|scoring-opic|urlnormalizer-(pass|regex|basic)|indexer-elastic</value>

</property>

<property>

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

<value>mycrawlername</value> <!-- this can be changed to something more sane if you like -->

</property>

<property>

<name>elastic.host</name>

<value>localhost</value> <!-- where is ElasticSearch listening -->

</property>

<property>

<name>elastic.cluster</name> <!-- this is the elstic clustername default is elasticsearch -->

<value>elasticsearch</value>

</property>

<property>

<name>elastic.index</name> <!--which index you want to store-->

<value>nutch</value>

</property>

<property>

<name>parser.character.encoding.default</name>

<value>utf-8</value>

</property>

</configuration>

**5.6:Ensure the Hbase gora-hbase dependency is available in $NUTCH\_HOME/ivy/ivy.xml**

<!-- Uncomment this to use HBase as Gora backend. -->

<dependency org="org.apache.gora" name="gora-hbase" rev="0.6.1" conf="\*->default" />

**5.7:Add a addition in $NUTCH\_HOME/ivy/ivy.xml**

In addition add the missing hbase-common-0.98.18-hadoop2.jar transitive dependency, this is a bug in gora-hbase 0.6.1 as described [here](https://github.com/apache/gora/pull/21). This bug is removed in current Gora development.

<dependency org="org.apache.hbase" name="hbase-common" rev="0.98.18-hadoop2" conf="\*->default" />

**5.8:Edit conf/gora.properties**

Other documentation for HBaseStore can be found [here](http://gora.apache.org/current/gora-hbase.html).

gora.datastore.default=org.apache.gora.hbase.store.HBaseStore

**5.9：ANT！**

in the nutch root directory,run

$ ant

Because of the internet cause ,the ant procedure will be about 40mins in china.

**5.10：ANT problems**

if you meet some problems when ant ,such as

resolve-default:  
[ivy:resolve] :: Apache Ivy 2.3.0 – 20130110142753 :: http://ant.apache.org/ivy/ ::  
[ivy:resolve] :: loading settings :: file = /home/walle/Documents/GD/apache-nutch-2.3.1/ivy/ivysettings.xml  
[ivy:resolve]  
[ivy:resolve] :: problems summary ::  
[ivy:resolve] :::: WARNINGS  
[ivy:resolve] ::::::::::::::::::::::::::::::::::::::::::::::  
[ivy:resolve] :: UNRESOLVED DEPENDENCIES ::  
[ivy:resolve] ::::::::::::::::::::::::::::::::::::::::::::::  
[ivy:resolve] :: org.springframework#spring-core;4.0.4.RELEASE: configuration not found in org.springframework#spring-core;4.0.4.RELEASE: ‘master’. It was required from org.apache.nutch#nutch;working@walle-X450JF master  
[ivy:resolve] ::::::::::::::::::::::::::::::::::::::::::::::  
[ivy:resolve]  
[ivy:resolve] :: USE VERBOSE OR DEBUG MESSAGE LEVEL FOR MORE DETAILS

BUILD FAILED

You can comment such as ‘spring’ events in ivy.xml ,then ant again.

**6:Use Nutch to crawl**

**6.1:make the seed**

$ cd runtime/deploy

$ vim urls

and you should write the website you want to crawl write in urls,such as http://www.wallen.top

and then you should upload the seed to the hdfs system.follow the command that the terminal give you.

hadoop fs –put urls /user/walle

**6.2:inject them into Nutch by giving a file URL (!)**

$ NUTCH\_HOME/runtime/delpoy/bin/nutch inject urls

**6.3:Generate a new set of URLs to fetch**

This is is based on both the injected URLs as well as outdated URLs in the Nutch crawl db.

$NUTCH\_HOME/runtime/deploy/bin/nutch generate -topN 10

**6.4:Fetch the URLs.**

We are not clustering, so we can simply fetch all batches:

$NUTCH\_HOME/runtime/deploy/bin/nutch fetch -all

**6.5:parse all fetched pages:**

$NUTCH\_HOME/runtime/deploy/bin/nutch parse -all

**6.6:Update Nutch’s internal database:**

$NUTCH\_HOME/runtime/deploy/bin/nutch updatedb -all

**6.7:elasticsearch indexing command**

$NUTCH\_HOME/bin/nutch index -all

**7：view the data on kibana**

open localhost:5601,and you need to uncheck the button:’index contain time-based events’.

and input ‘nutch’ which is your elasticsearch’s index name in the following textarea.Then go to the discover tab,you can see the data in elasticsearch and search data.

Reference:

http://hadoop.apache.org/docs/r2.5.2/hadoop-project-dist/hadoop-common/SingleCluster.html

https://hbase.apache.org/book.html

http://www.aossama.com/search-engine-with-apache-nutch-mongodb-and-elasticsearch/

\

NUTCH SOLR integration

The first step to get started is to download the required software components, namely Apache Solr and Nutch.

1. Download Solr version 1.3.0 or LucidWorks for Solr from Download page
2. Extract Solr package
3. Download Nutch version 1.0 or later (Alternatively download the the nightly version of Nutch that contains the required functionality)
4. Extract the Nutch package

tar xzf apache-nutch-1.0.tar.gz

1. Configure Solr

For the sake of simplicity we are going to use the example configuration of Solr as a base.

a. Copy the provided Nutch schema from directory apache-nutch-1.0/conf to directory apache-solr-1.3.0/example/solr/conf (override the existing file)

We want to allow Solr to create the snippets for search results so we need to store the content in addition to indexing it:

b. Change schema.xml so that the stored attribute of field “content” is true.

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

We want to be able to tweak the relevancy of queries easily so we’ll create new dismax request handler configuration for our use case:

d. Open apache-solr-1.3.0/example/solr/conf/solrconfig.xml and paste following fragment to it

<requestHandler name="/nutch" class="solr.SearchHandler" >

<lst name="defaults">

<str name="defType">dismax</str>

<str name="echoParams">explicit</str>

<float name="tie">0.01</float>

<str name="qf">

content^0.5 anchor^1.0 title^1.2

</str>

<str name="pf">

content^0.5 anchor^1.5 title^1.2 site^1.5

</str>

<str name="fl">

url

</str>

<str name="mm">

2&lt;-1 5&lt;-2 6&lt;90%

</str>

<int name="ps">100</int>

<bool hl="true"/>

<str name="q.alt">\*:\*</str>

<str name="hl.fl">title url content</str>

<str name="f.title.hl.fragsize">0</str>

<str name="f.title.hl.alternateField">title</str>

<str name="f.url.hl.fragsize">0</str>

<str name="f.url.hl.alternateField">url</str>

<str name="f.content.hl.fragmenter">regex</str>

</lst>

</requestHandler>

1. Start Solr

cd apache-solr-1.3.0/example java -jar start.jar

1. Configure Nutch

a. Open nutch-site.xml in directory apache-nutch-1.0/conf, replace it’s contents with the following (we specify our crawler name, active plugins and limit maximum url count for single host per run to be 100) :

<?xml version="1.0"?>

<configuration>

<property>

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

<value>nutch-solr-integration</value>

</property>

<property>

<name>generate.max.per.host</name>

<value>100</value>

</property>

<property>

<name>plugin.includes</name>

<value>protocol-http|urlfilter-regex|parse-html|index-(basic|anchor)|query-(basic|site|url)|response-(json|xml)|summary-basic|scoring-opic|urlnormalizer-(pass|regex|basic)</value>

</property>

</configuration>

b. Open regex-urlfilter.txt in directory apache-nutch-1.0/conf, replace it’s content with following:

-^(https|telnet|file|ftp|mailto):

**skip some suffixes**

-.(swf|SWF|doc|DOC|mp3|MP3|WMV|wmv|txt|TXT|rtf|RTF|avi|AVI|m3u|M3U|flv|FLV|WAV|wav|mp4|MP4|avi|AVI|rss|RSS|xml|XML|pdf|PDF|js|JS|gif|GIF|jpg|JPG|png|PNG|ico|ICO|css|sit|eps|wmf|zip|ppt|mpg|xls|gz|rpm|tgz|mov|MOV|exe|jpeg|JPEG|bmp|BMP)$

**skip URLs containing certain characters as probable queries, etc.**

-[?\*!@=]

**allow urls in foofactory.fi domain**

+^http://([a-z0-9-A-Z]\*.)\*lucidimagination.com/

**deny anything else**

-. 8. Create a seed list (the initial urls to fetch)

mkdir urls

echo "http://www.lucidimagination.com/" > urls/seed.txt

1. Inject seed url(s) to nutch crawldb (execute in nutch directory)

bin/nutch inject crawl/crawldb urls

1. Generate fetch list, fetch and parse content

bin/nutch generate crawl/crawldb crawl/segments

The above command will generate a new segment directory under crawl/segments that at this point contains files that store the url(s) to be fetched. In the following commands we need the latest segment dir as parameter so we’ll store it in an environment variable:

export SEGMENT=crawl/segments/`ls -tr crawl/segments|tail -1`

Now I launch the fetcher that actually goes to get the content:

bin/nutch fetch $SEGMENT -noParsing

Next I parse the content:

bin/nutch parse $SEGMENT

Then I update the Nutch crawldb. The updatedb command wil store all new urls discovered during the fetch and parse of the previous segment into Nutch database so they can be fetched later. Nutch also stores information about the pages that were fetched so the same urls won’t be fetched again and again.

bin/nutch updatedb crawl/crawldb $SEGMENT -filter -normalize

Now a full Fetch cycle is completed. Next you can repeat step 10 couple of more times to get some more content.

1. Create linkdb

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

1. Finally index all content from all segments to Solr

bin/nutch solrindex <http://127.0.0.1:8983/solr/> crawl/crawldb crawl/linkdb crawl/segments/\* Now the indexed content is available through Solr. You can try to execute searches from the Solr admin ui from

http://127.0.0.1:8983/solr/admin

, or directly with url like

http://127.0.0.1:8983/solr/nutch/?q=solr&version=2.2&start=0&rows=10&indent=on&wt=json

|  |
| --- |
| [share](http://stackoverflow.com/a/34714888/1638675)[edit](http://stackoverflow.com/posts/34714888/edit)[flag](http://stackoverflow.com/questions/29768111/how-can-i-integrate-solr-and-nutch-in-cloudera) |

<http://stackoverflow.com/questions/26364057/exception-in-thread-main-java-lang-noclassdeffounderror-org-apache-hadoop-hba>

<http://stackoverflow.com/questions/37458637/error-while-integrating-apache-nutch-2-3-with-hbase-0-94-14-and-solr-5-2-1>

<http://www.santy.io/nutch-hbase>

<http://stackoverflow.com/questions/16401667/java-lang-classnotfoundexception-org-apache-gora-hbase-store-hbasestore>