Getting Started with Nutch

Peter Lavin

2012-Oct-15

Table of Contents

About Nutch

Apache Nutch is a web crawler that is used in conjunction with Solr to index web pages. If you have the Solr server installed prior to installing Nutch, you can immediately pass the Nutch results to Solr.

Use your package manager to install Solr or follow the instructions given in Installing Solr on Red Hat-type Systems. Install Solr with the sample application file and don't worry about configuring Solr. Nutch comes with a Solr `schema.xml` file that works out of the box. Some changes may need to be made to the `solrconfig.xml` file but this is dealt with in the section called "Configuring Solr".

In this article Jetty is used as the servlet container for Solr but Tomcat will work equally well. This article describes using Nutch in the following configuration:

- Apache Solr 3.6.1
- CentOS release 6.3 (Final)
- Nutch 1.5.1

In this document Nutch and Solr are running on the same machine.

Installing and Configuring Nutch

Download the Nutch tar file and decompress it to a location of your choice for example, `/opt/apache-nutch-1.5.1`. In this case the Nutch home directory is the `/opt/apache-nutch-1.5.1` directory. You'll find the following files and directories below this directory:

```
bin  CHANGES.txt  conf  docs  lib  LICENSE.txt  logs  NOTICE.txt  plugins  README.txt
```

The **nutch** command found in the `bin` directory is typically invoked from the Nutch home directory. If you wish you can set an environment variable, `NUTCH_HOME`, but this is not a requirement.

Navigate to the Nutch home directory and create a directory named `urls` containing a file named `seed.txt`. Add a single line to this file to identify the domain that you wish to index, `http://objectorientedphp.com/`, for example.

In the `regex-urlfilter.xml` file replace the last line `+.` with a regular expression identifying the domain that you wish to crawl, for example: `+^http://([a-z0-9]*\.)*objectorientedphp.com/`.

You must also set the value of the `http.agent.name` property of the `nutch-site.xml` file before you run Nutch. If you like you can overwrite the `nutch-site.xml` file with the contents of `nutch-default.xml` file and then set `http.agent.name`—that's what `nutch-default.xml` is there for.

You must also set the `JAVA_HOME` environment variable if it is not already set. If **`which java`** returns `/usr/bin/java` on a Red Hat-type system you would add the line `export JAVA_HOME=/usr` to the `.bash_profile` file in your home directory.

Configuring Solr

To configure the Solr server to work with Nutch copy the `schema.xml` file found in the Nutch home `conf` directory to the `solr/conf` directory below the Solr home directory.

Warning

If the schema name version number is set to `1.5.1` on the line, `<schema name="nutch" version="1.5.1">`, change this to `1.5`. The Solr server will not start up until this change is made.

When initiating queries you may also need to make changes to the `solrconfig.xml` file. This is discussed in the section called "Checking Your Index".

Indexing

It is good practice to test Nutch by first searching at a reduced depth. You can do this by navigating to the Nutch home directory and issuing the command: **`bin/nutch crawl urls -dir crawl -solr http://localhost:8983/solr/ -depth 3 -topN 20`**. This command will crawl the domain defined in the `urls/seed.txt` file and at the same time create a searchable Solr index.

When Nutch has finished you should see output such as the following:

```
  ...
LinkDb: internal links will be ignored.
LinkDb: adding segment: file:/opt/apache-nutch-1.5.1/crawl-20121008104351/segments/20121008105515
LinkDb: adding segment: file:/opt/apache-nutch-1.5.1/crawl-20121008104351/segments/20121008104606
LinkDb: adding segment: file:/opt/apache-nutch-1.5.1/crawl-20121008104351/segments/20121008104454
LinkDb: adding segment: file:/opt/apache-nutch-1.5.1/crawl-20121008104351/segments/20121008104414
LinkDb: adding segment: file:/opt/apache-nutch-1.5.1/crawl-20121008104351/segments/20121008105427
LinkDb: finished at 2012-10-08 10:56:06, elapsed: 00:00:16
SolrIndexer: starting at 2012-10-08 10:56:06
Indexing 114 documents
SolrIndexer: finished at 2012-10-08 10:57:17, elapsed: 00:01:11
SolrDeleteDuplicates: starting at 2012-10-08 10:57:17
SolrDeleteDuplicates: Solr url: http://localhost:8983/solr/
SolrDeleteDuplicates: deleting 1 duplicates
SolrDeleteDuplicates: finished at 2012-10-08 10:57:23, elapsed: 00:00:06
crawl finished: crawl-20121008104351
```

The next section verifies that web pages have been indexed.

Checking Your Index

You can check the files that you have indexed by pointing your browser at `http://`*`solr_server`*`:8983/solr/admin/`. You should see something similar to the following:

Figure 1. Solr admin interface

If you are using the default `solrconfig.xml` and you initiate a search such as `*:*` you may see the following error:

```
problem:

Problem accessing /solr/select/. Reason:

    undefined field text
```

To remedy this, search the `solr/solrconfig.xml` file found below the Solr home directory for references to a field named `text` and replace these references with `content`. For example, the `select` request handler identifies the default field as `text`

```
 <requestHandler name="/select" class="solr.SearchHandler">
   <!-- default values for query parameters can be specified, these
        will be overridden by parameters in the request
     -->
    <lst name="defaults">
      <str name="echoParams">explicit</str>
      <int name="rows">10</int>
      <str name="df">text</str>
      ...
```

Change `text` to `content`. After making this configuration change you will have to restart Jetty. On Red Hat-type systems issue the command **`service restart jetty`**.

Removing Index Files

You may find that you want to remove the Solr index and start again from scratch especially if you have performed a test crawl. From the command line of the machine hosting the Solr server, use the following commands to remove an existing Solr index:

```
shell> curl http://localhost:8983/solr/update -H "Content-Type: text/xml" \
  --data-binary '<delete><query>*:*</query></delete>'
```

```
shell> curl http://localhost:8983/solr/update -H \
  "Content-Type: text/xml" --data-binary '<commit/>'
```

You should also remove all the Nutch files found in the `crawl` directory below the Nutch home directory.

The Final Cut

Once all configuration changes have been made, create the Solr index by navigating to the Nutch home directory and issuing the following command:

```
shell> bin/nutch crawl urls -dir crawl -solr http://localhost:8983/solr/ -depth 20 -topN 200
```

The `sites.txt` file in the `urls/` directory tells nutch which domain to crawl and the resulting files are stored in the `crawl` directory. These are used to build the Solr search index.

A complete description of the **nutch crawl** command is found at nutch crawl and reproduced below:

```
This class performs a complete crawl given a set of root urls.

Usage:

bin/nutch crawl <urlDir> [-solr <solrURL>] [-dir d] [-threads n] [-depth i] [-topN N]
<urlDir>: Contains text files with URL lists. This must be an existing directory. Example
would be ${NUTCH_HOME}/urls

[-solr <solrURL>]: Enables us to pass our Solr instance as an indexing parameter to simplify
the process of indexing with Solr.

[-dir d]: This parameter enables you to choose the directory Nutch should use when crawling.

[-threads n]: This parameter enables you to choose how many threads Nutch should use when crawling.

[-depth i]: You can tell Nutch how deep it should crawl. If you don't tell Nutch a value,
it takes 5 as his standard parameter. For example if you pass –depth 1 as the parameter,
Nutch will only index the first level. If you say –depth 2 (or more) Nutch will follow
this number of outlinks.

[-topN N]: The maximum number of outlinks Nutch will obtain from any one page.
```

Enhancements

You can easily enhance your configuration by making changes to files such as the `regex-urlfilter.txt` file. For example, if there are file types that you do not wish to index, add them to this filter.

If `parser.skip.truncated` in the `nutch-site.xml` file is set to `true` and you are using the default value for `http.content.limit` then no files larger than `65536` Kilobytes will be indexed. With this setting most PDF files will be ignored. The default setting is shown below:

```
<property>
  <name>http.content.limit</name>
  <value>65536</value>
  <description>The length limit for downloaded content using the http://
  protocol, in bytes. If this value is nonnegative (>=0), content longer
  than it will be truncated; otherwise, no truncation at all. Do not
  confuse this setting with the file.content.limit setting.
  </description>
</property>
```

Change the value of `http.content.limit` to `-1` to accept files of any size.

You will probably also want to write some code to present search results in an easily usable fashion.

Resources

Apache Nutch Wiki – the official Nutch Wiki

"Apache 3.1 Solr Cookbook" by Rafal Kuc, Packt Publishing – this book provides a recipe for getting started with Nutch

Integrating Nutch – an excerpt from LucidWorks documentation

About the Author

Peter Lavin is a technical writer who has been published in a number of print and online magazines. He is the author of Object Oriented PHP, published by No Starch Press and a contributor to PHP Hacks by O'Reilly Media.