

README for Joycrawler



A spider program for hadoop applications

<http://joycrawler.googlecode.com>

Introduction

Look for a simpler web spider working on clusters?

Confused by Nutch?

Want a clear web crawler structure?

Or need to do link analysis?

The project Joycrawler is a spider program for hadoop applications. It is also a standard hadoop program. Like the most popular spiders, it downloads the internet pages on given websites, hosts or even WWW. Moreover, a pagerank calculation program is integrated and this may very helpful for some web analysis applications.

In the following sections, we will give you a brief instruction of running such a light-weight internet web spider in both stand-alone and pseudo distributed mode.

Getting Started

Stand-Alone

Prerequisites: JRE 6, Apache Ant, **Cygwin if Windows (PATH variable must be configured), Linux not required.**

1. Download latest Joycrawler release on <http://joycrawler.googlecode.com/>, and unpack it.
2. Before running Joycrawler, please specify one parameter in configuration file **conf/Joycrawler-site.xml**:

```

<property>
<name>org.joy.crawler.regEx</name>
<value>http://.*YOUR_DOMAIN_HERE.*</value>
</property>

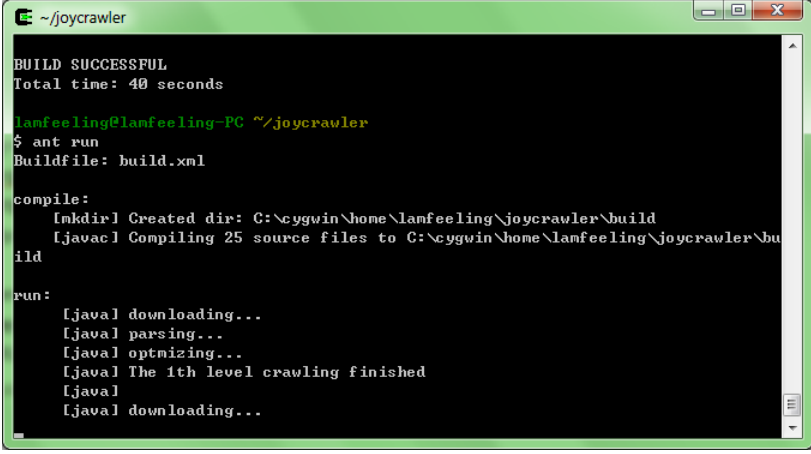
```

This property will set the host/website you want to crawl on.

NOTE: any regular expression is acceptable for this property

3. Add URL seeds in “seeds.txt” in program’s folder, one URL per line.

4. Run in command line: “ant run”, and you may see the following output:



```

~/joycrawler
BUILD SUCCESSFUL
Total time: 40 seconds

lanfeeling@lanfeeling-PC ~/joycrawler
$ ant run
Buildfile: build.xml

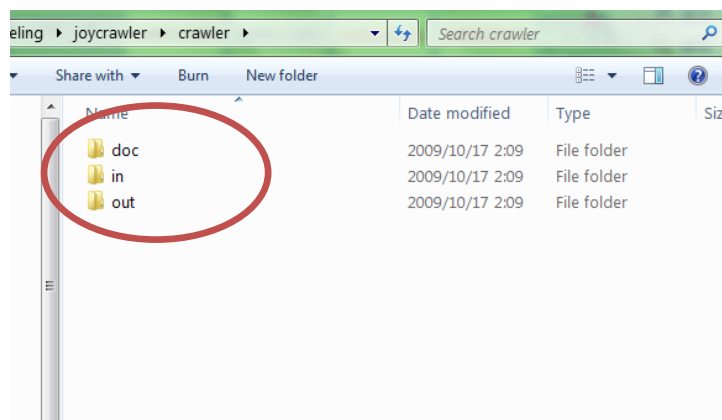
compile:
[mkdir] Created dir: C:\cygwin\home\lanfeeling\joycrawler\build
[javac] Compiling 25 source files to C:\cygwin\home\lanfeeling\joycrawler\build

run:
[java] downloading...
[java] parsing...
[java] optimizing...
[java] The 1th level crawling finished
[java]
[java] downloading...

```

After fetching all the pages or we reached the 15th iteration of fetching, the program will exit.

5. You can check the “crawler” folder for downloaded documents and links.



“Doc” folder is your webpage repository, and “out” folder contains all the link structure of the website you crawl. Both *doc* and *out* folders are standard hadoop output format. You can use `SequenceFileInputFormat` to read them into your application.

The Doc folder’s data pair is <Text, DocumentWritable>.

The Out folder’s data pair is <Text, OutlinksWritable>.

6. Pagerank result is in the *Ranking* folder, which is a plain text file, containing all the URL's rank score and links structure one record per line.

```

http://scst.suda.edu.cn/ 2.1510383844964207 http://scst.suda.edu.cn/page/2
http://scst.suda.edu.cn/article/20060004115202742.html 0.5151460260220568
http://scst.suda.edu.cn/article/20060004123156186.html 0.5151460260220568
http://scst.suda.edu.cn/article/20060004124059677.html 0.5151460260220568
http://scst.suda.edu.cn/article/20060005105614386.html 0.5141847731006361
http://scst.suda.edu.cn/article/20060126152730283.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126153558560.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126175824834.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126180421177.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126181119678.html 0.5078865732723904
http://scst.suda.edu.cn/article/2006012618145384.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126182028824.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126182706893.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126183252996.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126183434856.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126183612888.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126183956439.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126184053196.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126184222923.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126194822267.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126195324978.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060126195407879.html 0.5078865732723904
http://scst.suda.edu.cn/article/20060127111535332.html 0.5054924349575576
http://scst.suda.edu.cn/article/20060127112000917.html 0.5054924349575576

```

Distributed Mode

Before launching the Joycrawler in distributed mode, you must copy lib/neohtml-customized.jar, lib/xercesImpl.jar, lib/xercesMinimal.jar, lib/xml-apis.jar to hadoop's lib folder, and conf/Joycrawler-site.xml in hadoop's conf folder, and then start your hadoop (restart if already running).

1. Copy Joycrawler-VERSION.jar and seeds.txt to hadoop's top-level folder.

2. Run in command line:

bin/hadoop jar Joycrawler-VERSION.jar org.joy.crawler.Crawler seeds.txt 15

3. Run in command line (pagerank computing):

bin/hadoop jar Joycrawler-VERSION.jar org.joy.pagerank.RankDriver

Hadoop job_200908142354_0001 on localhost

User: song-pciadministrator
 Job Name: Download
 Job File: hdfs://localhost:9000/tmp/hadoop-Administrator/mapred/system/job_200908142354_0001/job.xml
 Job Setup: **Successful**
 Status: **Successful**
 Started at: Fri Aug 14 23:56:49 CST 2009
 Finished at: Fri Aug 14 23:57:15 CST 2009
 Finished in: 25sec
 Job Cleanup: **Successful**

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	1	0	0	1	0	0 / 0
reduce	100.00%	1	0	0	1	0	0 / 0

Counter		Map	Reduce	Total
Job Counters	Launched reduce tasks	0	0	1
	Launched map tasks	0	0	1
	Data-local map tasks	0	0	1
	FILE_BYTES_READ	0	123	123
	HDFS_BYTES_READ	25	0	25

Example

If you have trouble in how to process Joycrawler output in your own program, we provide you a very simple example Hadoop program, which located in `src/org/joy/crawler/example/ExtractorDriver.java`. This program simply reads each web text from crawler's output folder, and extracts all the context text. You can run it directly by using ***"ant example"*** after you have executed the command ***"ant run"***.



```
[java] analyzing:http://scst.suda.edu.cn/article/20071030174858399.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030175207474.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030183418991.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030183628456.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030183741584.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030185642312.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030190016501.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030190436466.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030190657450.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030191408593.html
[java] analyzing:http://scst.suda.edu.cn/article/2007103019574596.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030195845154.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030200212832.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030201410272.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030201522705.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030201628373.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030202441710.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030202558928.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030202808922.html
[java] analyzing:http://scst.suda.edu.cn/article/20071030202911621.html
```

Contact

Song Liu, Department of Computer Science, University of Bristol, UK

Email: lamfeeling2@gmail.com

MSN: lamfeeling@126.com