IntranetRecrawl

NOTE: the scripts listed here do not do recrawl correctly. It will add additional depth (specified by user) to a crawl. To avoid this, we need to use '-noAdditions' options to 'updatedb' command. But an annoying problem is that if you have used the 'crawl' command, newly discovered url have been added to the crawldb and will be fetched with the next 'fecth' command.

So the problem is, you will seen more pages being crawled using this recrawl script, not just the pages you have fecthed.

Here are a couple of scripts for recrawling your Intranet.

Version 0.7.2

Place in the main nutch directory and run.

Example Usage

./recrawl crawl 10 31

(with adddays being '31', all pages will be recrawled)

Script

```
#!/bin/bash

# A simple script to run a Nutch re-crawl

if [ -n "$1" ]
then
  crawl_dir=$1
else
  echo "Usage: recrawl crawl_dir [depth] [adddays]"
  exit 1
fi

if [ -n "$2" ]
then
  depth=$2
else
  depth=5
fi

if [ -n "$3" ]
then
  adddays=$3
else
  adddays=0
fi

webdb_dir=$crawl_dir/db
segments_dir=$crawl_dir/segments
index_dir=$crawl_dir/index

# The generate/fetch/update cycle
for ((i=1; i <= depth ; i++))
do
  bin/nutch generate $webdb_dir $segments_dir -adddays $adddays
  segment=`ls -d $segments_dir/* | tail -1`
  bin/nutch fetch $segment
  bin/nutch updatedb $webdb_dir $segment
done

# Update segments
mkdir tmp
bin/nutch updatesegs $webdb_dir $segments_dir tmp
rm -R tmp

# Index segments
for segment in `ls -d $segments_dir/* | tail -$depth`
do
  bin/nutch index $segment
done

# De-duplicate indexes
# "bogus" argument is ignored but needed due to
# a bug in the number of args expected
bin/nutch dedup $segments_dir bogus

# Merge indexes
ls -d $segments_dir/* | xargs bin/nutch merge $index_dir
```

Version 0.8.0 and 0.9.0

Place in the `bin` sub-directory within your Nutch install and run.

CALL THE SCRIPT USING THE FULL PATH TO THE SCRIPT OR IT WON'T WORK

Example Usage

```
./usr/local/nutch/bin/recrawl /usr/local/tomcat/webapps/ROOT /usr/local/nutch/crawl 10 31
```

Setting adddays at 31 causes all pages will to be recrawled.

Changes for 0.9.0

No changes necessary for this to run with Nutch 0.9.0.

However, if you get an error message indicating that the folder "index/merge-output" already exists, move the index/merge-output folder back into the index/ folder. For example:

```
mv $index_dir/merge-output /tmp
rm -rf $index_dir
mv /tmp/merge-output $index_dir
```

Code

```
#!/bin/bash

# Nutch recrawl script.
# Based on 0.7.2 script at http://today.java.net/pub/a/today/2006/02/16/introduction-to-nutch-2.html
#
# The script merges the new segments all into one segment to prevent redundant
# data. However, if your crawl/segments directory is becoming very large, I
# would suggest you delete it completely and generate a new crawl. This probaly
# needs to be done every 6 months.
#
# Modified by Matthew Holt
# mholt at elon dot edu

if [ -n "$1" ]
then
  tomcat_dir=$1
else
  echo "Usage: recrawl servlet_path crawl_dir depth adddays [topN]"
  echo "servlet_path - Path of the nutch servlet (full path, ie: /usr/local/tomc
at/webapps/ROOT)"
  echo "crawl_dir - Path of the directory the crawl is located in. (full path, i
e: /home/user/nutch/crawl)"
  echo "depth - The link depth from the root page that should be crawled."
  echo "adddays - Advance the clock # of days for fetchlist generation. [0 for n
one]"
  echo "[topN] - Optional: Selects the top # ranking URLS to be crawled."
  exit 1
fi

if [ -n "$2" ]
then
  crawl_dir=$2
else
  echo "Usage: recrawl servlet_path crawl_dir depth adddays [topN]"
  echo "servlet_path - Path of the nutch servlet (full path, ie: /usr/local/tomc
at/webapps/ROOT)"
  echo "crawl_dir - Path of the directory the crawl is located in. (full path, i
e: /home/user/nutch/crawl)"
  echo "depth - The link depth from the root page that should be crawled."
  echo "adddays - Advance the clock # of days for fetchlist generation. [0 for n
one]"
  echo "[topN] - Optional: Selects the top # ranking URLS to be crawled."
  exit 1
fi

if [ -n "$3" ]
then
  depth=$3
else
  echo "Usage: recrawl servlet_path crawl_dir depth adddays [topN]"
  echo "servlet_path - Path of the nutch servlet (full path, ie: /usr/local/tomc
```

```
at/webapps/ROOT)"
  echo "crawl_dir - Path of the directory the crawl is located in. (full path, i
e: /home/user/nutch/crawl)"
  echo "depth - The link depth from the root page that should be crawled."
  echo "adddays - Advance the clock # of days for fetchlist generation. [0 for n
one]"
  echo "[topN] - Optional: Selects the top # ranking URLS to be crawled."
  exit 1
fi

if [ -n "$4" ]
then
  adddays=$4
else
  echo "Usage: recrawl servlet_path crawl_dir depth adddays [topN]"
  echo "servlet_path - Path of the nutch servlet (full path, ie: /usr/local/tomcat/webapps/ROOT)"
  echo "crawl_dir - Path of the directory the crawl is located in. (full path, ie: /home/user/nutch/crawl)"
  echo "depth - The link depth from the root page that should be crawled."
  echo "adddays - Advance the clock # of days for fetchlist generation. [0 for n
one]"
  echo "[topN] - Optional: Selects the top # ranking URLS to be crawled."
  exit 1
fi

if [ -n "$5" ]
then
  topn="-topN $5"
else
  topn=""
fi

#Sets the path to bin
nutch_dir=`dirname $0`

# Only change if your crawl subdirectories are named something different
webdb_dir=$crawl_dir/crawldb
segments_dir=$crawl_dir/segments
linkdb_dir=$crawl_dir/linkdb
index_dir=$crawl_dir/index

# The generate/fetch/update cycle
for ((i=1; i <= depth ; i++))
do
  $nutch_dir/nutch generate $webdb_dir $segments_dir $topn -adddays $adddays
  segment=`ls -d $segments_dir/* | tail -1`
  $nutch_dir/nutch fetch $segment
  $nutch_dir/nutch updatedb $webdb_dir $segment
done

# Merge segments and cleanup unused segments
mergesegs_dir=$crawl_dir/mergesegs_dir
$nutch_dir/nutch mergesegs $mergesegs_dir -dir $segments_dir

for segment in `ls -d $segments_dir/* | tail -$depth`
do
  echo "Removing Temporary Segment: $segment"
  rm -rf $segment
done

cp -R $mergesegs_dir/* $segments_dir
rm -rf $mergesegs_dir

# Update segments
$nutch_dir/nutch invertlinks $linkdb_dir -dir $segments_dir

# Index segments
new_indexes=$crawl_dir/newindexes
segment=`ls -d $segments_dir/* | tail -1`
$nutch_dir/nutch index $new_indexes $webdb_dir $linkdb_dir $segment

# De-duplicate indexes
```

```
$nutch_dir/nutch dedup $new_indexes

# Merge indexes
$nutch_dir/nutch merge $index_dir $new_indexes

# Tell Tomcat to reload index
touch $tomcat_dir/WEB-INF/web.xml

# Clean up
rm -rf $new_indexes

echo "FINISHED: Recrawl completed. To conserve disk space, I would suggest"
echo " that the crawl directory be deleted once every 6 months (or more"
echo " frequent depending on disk constraints) and a new crawl generated."
```

Version 1.0

A crawl script that runs properly with bash and has been tested with Nutch 1.0 can be found here: Crawl. This script can do crawl as well as recrawl. However, not much real world recrawl has been done with this script. It might require a little bit of tweaking if you find that the script does not suit your needs.