Whole-Web Crawling incremental script

The following script does whole-web-crawling incrementally.

Input: a list of urls to crawl

Output: Nutch will continuously fetch $it_size urls from the input list, index and merge them with the whole-web index (so that they can be immediately searched) until all urls have been fetched.

Tested with Nutch-1.2 release (see tests output); If you don't have Nutch set up, follow this tutorial.

Script Editions:

1. Abridged using Solr (tersest)
2. Unabridged with explanations and using nutch index and local fs cmds (beginner)
3. Unabridged with explanations, using solr and Hadoop fs cmds (advanced)

Please report any bug you find on the mailing list and to me.

1. Abridged using Solr (tersest)

```
#!/bin/sh

#
# Created by Gabriele Kahlout on 27.03.11.
# The following script crawls the whole-web incrementally; Specifying a list of urls to crawl, nutch will
continuously fetch $it_size urls from a specified list of urls, index and merge them with our whole-web index,
so that they can be immediately searched, until all urls have been fetched.
# It assumes that you have setup Solr and it's running on port 8080.
#
# TO USE:
# 1. $ mv whole-web-crawling-incremental $NUTCH_HOME/whole-web-crawling-incremental
# 2. $ cd $NUTCH_HOME
# 3. $ chmod +x whole-web-crawling-incremental
# 4. $ ./whole-web-crawling-incremental

# Usage: ./whole-web-crawling-incremental [it_seedsDir-path urls-to-fetch-per-iteration depth]
# Start

rm -r crawl

seedsDir=$1
it_size=$2
depth=$3

indexedPlus1=1 #indexedPlus1 urls+1 because of tail. Never printed out
it_seedsDir="$seedsDir/it_seeds"
rm -r $it_seedsDir
mkdir $it_seedsDir

allUrls=`cat $seedsDir/*url* | wc -l | sed -e "s/^ *//"`

it_crawldb="crawl/crawldb"


while [[ $indexedPlus1 -le $allUrls ]]
do
        rm $it_seedsDir/urls
        tail -n+$indexedPlus1 $seedsDir/*url* | head -n$it_size > $it_seedsDir/urls

        bin/nutch inject $it_crawldb $it_seedsDir
        i=0

        while [[ $i -lt $depth ]]
        do
                echo
                echo "generate-fetch-updatedb-invertlinks-index-merge iteration "$i":"
                cmd="bin/nutch generate $it_crawldb crawl/segments -topN $it_size"
                output=`$cmd`
                if [[ $output == *'0 records selected for fetching'* ]]
                then
                        break;
                fi
                s1=`ls -d crawl/segments/2* | tail -1`

                bin/nutch fetch $s1

                bin/nutch updatedb $it_crawldb $s1

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

                bin/nutch solrindex http://localhost:8080/solr/ $it_crawldb crawl/linkdb crawl/segments/*

                ((i++))
                ((indexedPlus1+=$it_size))
        done
done
rm -r $it_seedsDir
```

2. Unabridged with explanations and using nutch index and local fs cmds (beginner)

```
#!/bin/sh

#
# Created by Gabriele Kahlout on 27.03.11.
# The following script crawls the whole-web incrementally; Specifying a list of urls to crawl, nutch will
continuously fetch $it_size urls from a specified list of urls, index and merge them with our whole-web index,
so that they can be immediately searched, until all urls have been fetched.
#
# TO USE:
# 1. $ mv whole-web-crawling-incremental $NUTCH_HOME/whole-web-crawling-incremental
# 2. $ cd $NUTCH_HOME
# 3. $ chmod +x whole-web-crawling-incremental
# 4. $ ./whole-web-crawling-incremental

# Usage: ./whole-web-crawling-incremental [it_seedsDir-path urls-to-fetch-per-iteration depth]
# Start

function echoThenRun () { # echo and then run the command
  echo $1
  $1
  echo
}

echoThenRun "rm -r crawl" # fresh crawl

if [[ ! -d "build" ]]
then
        echoThenRun "ant"
fi

seedsDir="seeds"
if [[ $1 != "" ]]
then
        seedsDir=$1
fi

it_size=10
if [[ $2 != "" ]]
then
        it_size=$2
fi

indexedPlus1=1 #indexedPlus1 urls+1 because of tail. Never printed out
it_seedsDir="$seedsDir/it_seeds"
rm -r $it_seedsDir
mkdir $it_seedsDir

allUrls=`cat $seedsDir/*url* | wc -l | sed -e "s/^ *//"`
echo $allUrls" urls to crawl"

it_crawldb="crawl/crawldb"

depth=1
if [[ $3 != "" ]]
then
        depth=$3
fi

while [[ $indexedPlus1 -le $allUrls ]] #repeat generate-fetch-updatedb-invertlinks-index-merge loop until all
urls are fetched
do
        rm $it_seedsDir/urls
        tail -n+$indexedPlus1 $seedsDir/*url* | head -n$it_size > $it_seedsDir/urls
        echo
        echoThenRun "bin/nutch inject $it_crawldb $it_seedsDir"
        i=0
```

```
        while [[ $i -lt $depth ]] # depth-first
        do
                echo
                echo "generate-fetch-updatedb-invertlinks-index-merge iteration "$i":"

                echo
                cmd="bin/nutch generate $it_crawldb crawl/segments -topN $it_size"
                echo $cmd
                output=`$cmd`
                echo $output
                if [[ $output == *'0 records selected for fetching'* ]] #all the urls of this iteration have
been fetched
                then
                        break;
                fi
                s1=`ls -d crawl/segments/2* | tail -1`

                echoThenRun "bin/nutch fetch $s1"

                echoThenRun "bin/nutch updatedb $it_crawldb $s1"

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


                # echoThenRun "bin/nutch solrindex http://localhost:8080/solr/ $it_crawldb crawl/linkdb crawl
/segments/*"
                # if you have solr setup you can use it by uncommenting the above command and commenting the
following nutch index and merge step.

                # start nutch index and merge step
                new_indexes="crawl/new_indexes"
                rm -r $new_indexes $temp_indexes
                echoThenRun "bin/nutch index $new_indexes $it_crawldb crawl/linkdb crawl/segments/*"
                indexes="crawl/indexes"
                temp_indexes="crawl/temp_indexes"

                # solrindex also merged, with nutch index we've to do it:
                echoThenRun "bin/nutch merge $temp_indexes/part-1 $indexes $new_indexes" # work-around for
https://issues.apache.org/jira/browse/NUTCH-971 (Patch available)

                rm -r $indexes $new_indexes
                mv $temp_indexes $indexes

                # end nutch index and merge step

                # you can now search the index with http://localhost:8080/solr/admin/ (if setup) or http://code.
google.com/p/luke/ . The index is stored in crawl/indexes, while if Solr is used then in $NUTCH_HOME/solr/data
/index.
                ((i++))
                ((indexedPlus1+=$it_size)) # maybe should readdb crawl/crawldb -stats number of actually
fetched, but (! going to fetch a page) --> infinite loop
        done

        echoThenRun "bin/nutch readdb $it_crawldb -stats"

        allcrawldb="crawl/allcrawldb"
        temp_crawldb="crawl/temp_crawldb"
        merge_dbs="$it_crawldb $allcrawldb"

        # work-around for https://issues.apache.org/jira/browse/NUTCH-972 (Patch available)
        if [[ ! -d $allcrawldb ]]
        then
                merge_dbs="$it_crawldb"
        fi

        echoThenRun "bin/nutch mergedb $temp_crawldb $merge_dbs"

        rm -r $allcrawldb $it_crawldb crawl/segments crawl/linkdb
        mv $temp_crawldb $allcrawldb
done
```

```
echo
crawl_dump="$allcrawldb/dump"

rm -r $crawl_dump $it_seedsDir
echoThenRun "bin/nutch readdb $allcrawldb -dump $crawl_dump" # you can inspect the dump with $ vim $crawl_dump
bin/nutch readdb $allcrawldb -stats
```

3. Unabridged with explanations, using solr and Hadoop fs cmds (advanced)

```
#!/bin/bash

#
# Created by Gabriele Kahlout on 27.03.11.
# The following script crawls the whole-web incrementally; Specifying a list of urls to crawl, nutch will
continuously fetch $it_size urls from a specified list of urls, index and merge them with our whole-web index,
so that they can be immediately searched, until all urls have been fetched.
#
# TO USE:
# 1. $ mv whole-web-crawling-incremental $NUTCH_HOME/whole-web-crawling-incremental
# 2. $ cd $NUTCH_HOME
# 3. $ chmod +x whole-web-crawling-incremental
# 4. $ ./whole-web-crawling-incremental

# Usage: ./whole-web-crawling-incremental [-f] [-i urls-to-fetch-per-iteration] [-d depth] [seedsDir-path]
# Start

function echoThenRun () { # echo and then run the command
  echo $1
  $1
  echo
}

it_size=1
depth=1
fresh=false
solrIndex="http://localhost:8080/solr"
seedsDir="seeds"

while getopts  "fi:d:" option
do
  case $option in
         f) fresh=true;;
                 i) it_size=$OPTARG;;
         d) depth=$OPTARG;;
   esac
done

if [[ ${@:$OPTIND} != "" ]]
then
        seedsDir=${@:$OPTIND}
fi


if $fresh ; then
        echoThenRun "bin/hadoop dfs -rmr crawl"
        echoThenRun "curl --fail $solrIndex/update?commit=true -d  '<delete><query>*:*</query></delete>'"
#empty index
fi

if [[ ! -d "build" ]]
then
        echoThenRun "ant"
fi

indexedPlus1=1 #indexedPlus1 urls+1 because of tail. Never printed out
it_seedsDir="$seedsDir/it_seeds"
```

```
bin/hadoop dfs -rmr $it_seedsDir
bin/hadoop dfs -mkdir $it_seedsDir
bin/hadoop dfs -mkdir crawl/crawldb
rm $seedsDir/urls-local-only

echoThenRun "bin/hadoop dfs -get $seedsDir/*url* $seedsDir/urls-local-only"

allUrls=`cat $seedsDir/urls-local-only | wc -l | sed -e "s/^ *//"`
echo $allUrls" urls to crawl"

j=0
while [[ $indexedPlus1 -le $allUrls ]] #repeat generate-fetch-updatedb-invertlinks-index-merge loop until all
urls are fetched
do
        bin/hadoop dfs -rm $it_seedsDir/urls

        tail -n+$indexedPlus1 $seedsDir/urls-local-only | head -n$it_size > $it_seedsDir/urls-local-only
        bin/hadoop dfs -moveFromLocal $it_seedsDir/urls-local-only $it_seedsDir/urls

        it_crawldb="crawl/crawldb/$j"
        it_segs="crawl/segments/$j"

        fCrawl=`bin/hadoop dfs -test -d $it_crawldb`
        if [[ $fCrawl == "" ]] #doesn't exist --> fresh crawl
        then
                bin/hadoop dfs -mkdir $it_crawldb
                echo
                echoThenRun "bin/nutch inject $it_crawldb $it_seedsDir"
        fi

    i=0
        while [[ $i -lt $depth ]] # depth-first
        do
                echo
                echo "generate-fetch-invertlinks-updatedb-index iteration "$i":"
                bin/hadoop dfs -rmr $it_segs
                bin/hadoop dfs -rm $it_crawldb/.locked $it_crawldb/..locked.crc
                cmd="bin/nutch generate $it_crawldb $it_segs -topN $it_size"
                echo
                echo $cmd
                output=`$cmd`
                echo $output
                echo

                if [[ $output == *'0 records selected for fetching'* ]] #all the urls of this iteration have
been fetched
                then
                        break;
                fi

                echoThenRun "bin/nutch fetch $it_segs/2*"

                echoThenRun "bin/nutch updatedb $it_crawldb $it_segs/2*"

                echoThenRun "bin/nutch invertlinks crawl/linkdb -dir $it_segs"

                echoThenRun "bin/nutch solrindex $solrIndex $it_crawldb crawl/linkdb $it_segs/*"

                # you can now search the index with http://localhost:8080/solr/admin/ (if setup) or http://code.
google.com/p/luke/ . The index is stored in crawl/indexes, while if Solr is used then in $NUTCH_HOME/solr/data
/index.

                bin/hadoop dfs -rmr $it_segs/2*
                echo
                ((i++))
        done
        ((indexedPlus1+=$it_size)) # maybe should readdb crawl/crawldb -stats number of actually fetched, but
(! going to fetch a page) --> infinite loop
        echoThenRun "bin/nutch readdb $it_crawldb -stats"
        ((j++))
done
```

```
bin/hadoop dfs -rmr $it_seedsDir
```