Run Many Hadoop Jobs

Roy E Lowrance

September 30, 2013

1 Problem

You want to run many Hadoop jobs without writing a long script for each.

2 Solution

Use the script `map-reduce.sh` which is invoked this way:

```
$ ./map-reduce.sh INPUT JOB [READLIMIT]
```

where

- INPUT is the path to the input file in the Hadoop file system
- JOB is the name of the map-reduce job. The mapper is `JOB-map.lua`. The reducer is `JOB-reduce.lua`
- READLIMIT is ignored. It is provided for compatiblity with the `map-reduce-local.sh` script which is described in the recipe "I want to test my Hadoop program."

The `map-reduce.sh` script will run your job and copy the job's output directory to the local file system in directory `$HOME/map-reduce-output/`. You can change this directory by editing the script.

I like to display at least the first portion of the output file. I do this by writing another script `go.sh` which runs `map-reduce.sh` then prints the first output file.

3 Discussion

The script `map-reduce.sh` is just below. It is a modification of the script presented in "Using Torch on Hadoop." Refer to that recipe for documentation.

```
# run map reduce job using streaming interface
# USAGE:
# Change to the directory where your source code and scripts live. Then
# enter
#  ./map-reduce.sh INPUT JOB USERID? [READLIMIT]
# where
# INPUT     is the name of your input file in the Hadoop file system
#           If not in the Hadoop file system, must be in the local
#           file system, in which case, it is copied to the Hadoop
#           file system.
# JOB       is the name of your map reduce job. The mapper command is
#             JOB-map.lua is the mapper command
#             JOB-reduce.lua is the reducer command
# READLIMIT for compatibility with your local map-reduce job runner.
#
# The script copies the output of the job from the Hadoop file system
# to $HOME/map-reduce-output

set -x  # print each command before executing it

# 1: capture command line arguments
INPUT=$1
JOB=$2
READLIMIT=$3  # NOT USED

# 2: input and output paths
INPUT_PATH=/home/$USER/$INPUT
OUTPUT_DIR_HADOOP=$INPUT.$JOB
MAP_REDUCE_OUTPUT=map-reduce-output
OUTPUT_DIR_LOCAL=$HOME/$MAP_REDUCE_OUTPUT/$OUTPUT_DIR_HADOOP

# 3: where Hadoop lives
HADOOP_HOME=/usr/lib/hadoop
STREAMING="hadoop-streaming-1.0.3.16.jar"

# 4: copy test file to the hadoop file system if it is not already there
hadoop fs -test -e $INPUT
if [ $? -ne 0 ]
then
  echo copying input from local file system to hadoop file system
  hadoop fs -copyFromLocal $INPUT $INPUT
else
  echo input file already in the hadoop file system
fi
```

```
# 5: delete output directory from previous run if it exists
hadoop fs -test -e $OUTPUT_DIR_HADOOP
if [ $? -eq 0 ]
then
  echo deleting output directory: $OUTPUT_DIR_HADOOP
  hadoop fs -rmr $OUTPUT_DIR_HADOOP
else
  echo output directory not in the hadoop file system
fi

# 6: run the streaming job; it will create the output directory
echo starting hadoop streaming job
hadoop jar $HADOOP_HOME/contrib/streaming/$STREAMING \
 -file *.lua \
 -mapper $PWD/$JOB-map.lua \
 -reducer $PWD/$JOB-reduce.lua \
 -input $INPUT \
 -output $OUTPUT_DIR_HADOOP
echo finished hadoop job

# 7: copy output file to home directory
# delete output directory if it already exists
# We delete in case it has old content that is not overwritten by copyToLocal
echo output dir local=$OUTPUT_DIR_LOCAL
echo output dir hadoop=$OUTPUT_DIR_HADOOP
if [ -a $OUTPUT_DIR_LOCAL ]
then
  # local output directory exists, so delete it
  # run in subshell so cd has only temporary effect
  (cd $OUTPUT_DIR_LOCAL; rm -rf -- $OUTPUT_DIR_LOCAL)
fi

mkdir -p $OUTPUT_DIR_LOCAL  # copyToLocal wants the directory to already exist
echo about to copy to local from $OUTPUT_DIR_HADOOP
hadoop fs -copyToLocal $OUTPUT_DIR_HADOOP $HOME/$MAP_REDUCE_OUTPUT/
#hadoop fs -copyToLocal courant-abel-prize-winners.txt.countInput /home/rel292/map-reduce-ou

# 8: list output dir
echo LOCAL OUTPUT DIRECTORY
ls $OUTPUT_DIR_LOCAL
```

The script `go.sh` is just below. Modify it so that it uses your INPUT and JOB and prints what is relevant to your work.

```
# run map-reduce job and print output
```

```
INPUT=courant-abel-prize-winners.txt
JOB=countInput

# run the job
./map-reduce.sh $INPUT $JOB

# print the output file
cat $HOME/map-reduce-output/$INPUT.$JOB/part-00000
```

4 See also

Other relevant recipes include

- "Using Torch on Hadoop"
- "I want to test my Hadoop program"

This recipe is free documentation. You can modify it by visiting github for account "rlowrance" and forking the repo "torch-cookbook."