CPSC 4770/6770

Distributed and Cluster Computing

Lecture 14: Debugging Hadoop MapReduce Jobs

Data: Movie Ratings and Recommendation

- An independent movie company is looking to invest in a new movie project. With limited finance, the company wants to analyze the reaction of audiences, particularly toward various movie genres, in order to identify beneficial movie project to focus on. The company relies on data collected from a publicly available recommendation service by MovieLens.
  This dataset contains **24404096** ratings and **668953** tag applications across **40110** movies. These data were created by **247753** users between January 09, 1995 and January 29, 2016. This dataset was generated on October 17, 2016.
- From this dataset, several analyses are possible, include the followings:
  - Find movies which have the highest average ratings over the years and identify the corresponding genre.
  - Find genres which have the highest average ratings over the years.
  - Find users who rate movies most frequently in order to contact them for in-depth marketing analysis.

A Glance of Data on HDFS

- hdfs dfs -ls -h intro-to-hadoop/movielens

```
Found 7 items
-rw-r--r--   3 jin6 supergroup      9.3 K 2020-09-18 09:08 intro-to-hadoop/movielens/README.txt
-rw-r--r--   3 jin6 supergroup    317.9 M 2020-09-18 09:08 intro-to-hadoop/movielens/genome-scores.csv
-rw-r--r--   3 jin6 supergroup     17.7 K 2020-09-18 09:08 intro-to-hadoop/movielens/genome-tags.csv
-rw-r--r--   3 jin6 supergroup    839.2 K 2020-09-18 09:08 intro-to-hadoop/movielens/links.csv
-rw-r--r--   3 jin6 supergroup      1.9 M 2020-09-18 09:08 intro-to-hadoop/movielens/movies.csv
-rw-r--r--   3 jin6 supergroup    632.7 M 2020-09-18 09:08 intro-to-hadoop/movielens/ratings.csv
-rw-r--r--   3 jin6 supergroup     22.9 M 2020-09-18 09:08 intro-to-hadoop/movielens/tags.csv
```

- hdfs dfs -cat intro-to-hadoop/movielens/README.txt
- hdfs dfs -cat intro-to-hadoop/movielens/links.csv 2>/dev/null | head -n 5
- hdfs dfs -cat intro-to-hadoop/movielens/movies.csv 2>/dev/null | head -n 5
- hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 5
- hdfs dfs -cat intro-to-hadoop/movielens/tags.csv 2>/dev/null | head -n 5

Find movies which have the highest average ratings over the years and report their ratings and genres

- Mapper01: Extract rating information
  - cat -n codes/avgRatingMapper01.py
  - hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 10 | python ./codes/avgRatingMapper01.py

```
%%writefile codes/avgRatingMapper01.py
#!/usr/bin/env python

import sys

for oneMovie in sys.stdin:
    oneMovie = oneMovie.strip()
    ratingInfo = oneMovie.split(",")
    movieID = ratingInfo[1]
    rating = ratingInfo[2]
    print ("%s\t%s" % (movieID, rating))
```

```
[jin6@node0397 myhadoop]$ hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 10 | python .
/codes/avgRatingMapper01.py
movieId rating
122     2.0
172     1.0
1221    5.0
1441    4.0
1609    3.0
1961    3.0
1972    1.0
441     2.0
494     2.0
```

Do we really need the headers?

```
%%writefile codes/avgRatingMapper02.py
#!/usr/bin/env python

import sys

for oneMovie in sys.stdin:
    oneMovie = oneMovie.strip()
    ratingInfo = oneMovie.split(",")
    try:
        movieID = ratingInfo[1]
        rating = float(ratingInfo[2])
        print ("%s\t%s" % (movieID, rating))
    except ValueError:
        continue
```

- Mapper02: Extract rating information without header
  - cat -n codes/avgRatingMapper02.py
  - hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 10 | python ./codes/avgRatingMapper02.py

```
[jin6@node0397 myhadoop]$ hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 10 | python .
/codes/avgRatingMapper02.py
122     2.0
172     1.0
1221    5.0
1441    4.0
1609    3.0
1961    3.0
1972    1.0
441     2.0
494     2.0
```

Getting additional file

- Mapper03: Get additional files
  - mkdir movielens
  - hdfs dfs -get intro-to-hadoop/movielens/movies.csv movielens/
  - cat –n codes/avgRatingMapper03.py
  - hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 10 | python ./codes/avgRatingMapper03.py

```
%%writefile codes/avgRatingMapper03.py
#!/usr/bin/env python

import sys
import csv

movieFile = "./movielens/movies.csv"
movieList = {}

with open(movieFile, mode = 'r') as infile:
    reader = csv.reader(infile)
    for row in reader:
        movieList[row[0]] = {}
        movieList[row[0]]["title"] = row[1]
        movieList[row[0]]["genre"] = row[2]

for oneMovie in sys.stdin:
    oneMovie = oneMovie.strip()
    ratingInfo = oneMovie.split(",")
    try:
        movieTitle = movieList[ratingInfo[1]]["title"]
        movieGenre = movieList[ratingInfo[1]]["genre"]
        rating = float(ratingInfo[2])
        print ("%s\t%s\t%s" % (movieTitle, rating, movieGenre))
    except ValueError:
        continue
```

```
[jin6@node0397 myhadoop]$ hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 10 | python .
/codes/avgRatingMapper03.py
Boomerang (1992)        2.0     Comedy|Romance
Johnny Mnemonic (1995)  1.0     Action|Sci-Fi|Thriller
Godfather: Part II, The (1974)  5.0     Crime|Drama
Benny & Joon (1993)     4.0     Comedy|Romance
187 (One Eight Seven) (1997)    3.0     Drama|Thriller
Rain Man (1988) 3.0     Drama
Nightmare on Elm Street 5: The Dream Child, A (1989)    1.0     Horror
Dazed and Confused (1993)       2.0     Comedy
Executive Decision (1996)       2.0     Action|Adventure|Thriller
```

Test reducer

```
%%writefile codes/avgRatingReducer01.py
#!/usr/bin/env python
import sys

current_movie = None
current_rating_sum = 0
current_rating_count = 0

for line in sys.stdin:
    line = line.strip()
    movie, rating, genre = line.split("\t", 2)
    try:
        rating = float(rating)
    except ValueError:
        continue

    if current_movie == movie:
        current_rating_sum += rating
        current_rating_count += 1
    else:
        if current_movie:
            rating_average = current_rating_sum / current_rating_count
            print ("%s\t%s\t%s" % (current_movie, rating_average, genre))
        current_movie = movie
        current_rating_sum = rating
        current_rating_count = 1

if current_movie == movie:
    rating_average = current_rating_sum / current_rating_count
    print ("%s\t%s\t%s" % (current_movie, rating_average, genre))
```

- Reducer01: Simple reducer
  - cat -n codes/avgRatingReducer01.py
  - hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 10 | python ./codes/avgRatingMapper03.py | sort | python ./codes/avgRatingReducer01.py

```
[jin6@node0397 myhadoop]$ hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 10 | python .
/codes/avgRatingMapper03.py | sort | python ./codes/avgRatingReducer01.py
187 (One Eight Seven) (1997)    3.0     Comedy|Romance
Benny & Joon (1993)     4.0     Comedy|Romance
Boomerang (1992)        2.0     Comedy
Dazed and Confused (1993)       2.0     Action|Adventure|Thriller
Executive Decision (1996)       2.0     Crime|Drama
Godfather: Part II, The (1974)  5.0     Action|Sci-Fi|Thriller
Johnny Mnemonic (1995)  1.0     Horror
Nightmare on Elm Street 5: The Dream Child, A (1989)    1.0     Drama
Rain Man (1988) 3.0     Drama
```

Non-HDFS correctness test

- Grep movie <<Matrix>>
- hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 2000 | python ./codes/avgRatingMapper03.py | grep Matrix

```
[jin6@node0397 myhadoop]$ hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 2000 | python
 ./codes/avgRatingMapper03.py | grep Matrix
Matrix Reloaded, The (2003)     4.0     Action|Adventure|Sci-Fi|Thriller|IMAX
Matrix, The (1999)      3.5     Action|Sci-Fi|Thriller
Matrix, The (1999)      3.5     Action|Sci-Fi|Thriller
Matrix, The (1999)      4.5     Action|Sci-Fi|Thriller
Matrix Reloaded, The (2003)     1.0     Action|Adventure|Sci-Fi|Thriller|IMAX
Matrix, The (1999)      5.0     Action|Sci-Fi|Thriller
Matrix Reloaded, The (2003)     5.0     Action|Adventure|Sci-Fi|Thriller|IMAX
Matrix Revolutions, The (2003)  2.5     Action|Adventure|Sci-Fi|Thriller|IMAX
Matrix, The (1999)      3.5     Action|Sci-Fi|Thriller
```

- hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 2000 | python ./codes/avgRatingMapper03.py | grep Matrix | sort | python ./codes/avgRatingReducer01.py

```
[jin6@node0397 myhadoop]$ hdfs dfs -cat intro-to-hadoop/movielens/ratings.csv 2>/dev/null | head -n 2000 | python
 ./codes/avgRatingMapper03.py | grep Matrix | sort | python ./codes/avgRatingReducer01.py
Matrix Reloaded, The (2003)     3.3333333333333335      Action|Adventure|Sci-Fi|Thriller|IMAX
Matrix Revolutions, The (2003)  2.5     Action|Sci-Fi|Thriller
Matrix, The (1999)      4.0     Action|Sci-Fi|Thriller
```

Full execution on HDFS

- mapred streaming -input intro-to-hadoop/movielens/ratings.csv -output intro-to-hadoop/output-movielens-01 -file ./codes/avgRatingMapper03.py -mapper avgRatingMapper03.py -file ./codes/avgRatingReducer01.py -reducer avgRatingReducer01.py

```
2020-09-18 09:15:52,232 INFO mapreduce.Job: Job job_1600434134826_0003 failed with state FAILED due to: Task fail
ed task_1600434134826_0003_m_000002
Job failed as tasks failed. failedMaps:1 failedReduces:0 killedMaps:0 killedReduces: 0

2020-09-18 09:15:52,351 INFO mapreduce.Job: Counters: 14
        Job Counters
                Failed map tasks=27
                Killed map tasks=9
                Killed reduce tasks=1
                Launched map tasks=36
                Other local map tasks=26
                Data-local map tasks=10
                Total time spent by all maps in occupied slots (ms)=7639296
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=238728
                Total vcore-milliseconds taken by all map tasks=238728
                Total megabyte-milliseconds taken by all map tasks=977829888
        Map-Reduce Framework
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
2020-09-18 09:15:52,351 ERROR streaming.StreamJob: Job not successful!
Streaming Command Failed!
```

Debugging Error

```
[jin6@node0528 codes]$ hdfs dfs -rm -R intro-to-hadoop/output-movielens-01
rm: `intro-to-hadoop/output-movielens-01': No such file or directory
[jin6@node0528 codes]$ yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar -input /repository/movielens/ratings.
csv -output intro-to-hadoop/output-movielens-01 -file avgRatingMapper03.py -mapper avgRatingMapper03.py -file avgRatingReducer01.py
 -reducer avgRatingReducer01.py

19/01/25 12:09:07 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [avgRatingMapper03.py, avgRatingReducer01.py] [/usr/hdp/2.6.5.0-292/hadoop-mapreduce/hadoop-streaming-2.7.3.2.6.5.0-
292.jar] /hadoop_java_io_tmpdir/streamjob9108723034401539927.jar tmpDir=null
19/01/25 12:09:09 INFO client.AHSProxy: Connecting to Application History server at dscim003.palmetto.clemson.edu/10.125.8.215:1020
0
19/01/25 12:09:09 INFO client.AHSProxy: Connecting to Application History server at dscim003.palmetto.clemson.edu/10.125.8.215:1020
0
19/01/25 12:09:10 INFO hdfs.DFSClient: Created HDFS_DELEGATION_TOKEN token 21561 for jin6 on ha-hdfs:dsci
19/01/25 12:09:10 INFO security.TokenCache: Got dt for hdfs://dsci; Kind: HDFS_DELEGATION_TOKEN, Service: ha-hdfs:dsci, Ident: (HDF
S_DELEGATION_TOKEN token 21561 for jin6)
19/01/25 12:09:10 INFO lzo.GPLNativeCodeLoader: Loaded native gpl library
19/01/25 12:09:10 INFO lzo.LzoCodec: Successfully loaded & initialized native-lzo library [hadoop-lzo rev 77914d73bfc2e32253ff2bb7c
61d03eaca973704]
19/01/25 12:09:10 INFO mapred.FileInputFormat: Total input paths to process : 1
19/01/25 12:09:11 INFO mapreduce.JobSubmitter: number of splits:5
19/01/25 12:09:11 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1542484792469_0142
19/01/25 12:09:11 INFO mapreduce.JobSubmitter: Kind: HDFS_DELEGATION_TOKEN, Service: ha-hdfs:dsci, Ident: (HDFS_DELEGATION_TOKEN to
ken 21561 for jin6)
19/01/25 12:09:11 INFO impl.TimelineClientImpl: Timeline service address: http://dscim003.palmetto.clemson.edu:8188/ws/v1/timeline/
19/01/25 12:09:12 INFO impl.YarnClientImpl: Submitted application application_1542484792469_0142
```

- Running the logs command of yarn with the provided application ID to access all available log information for that application:
  - yarn logs -applicationId application_ 1542484792469_0142
- Reduce log output by commenting out all non-essential lines (lines containing INFO)
  - yarn logs -applicationId application_1542484792469_0142 | grep -v INFO

Debugging Error (Cont.)

- Extract only message listing the Container IDs:
  - yarn logs -applicationId application_1542484792469_0142 | grep '^Container:'
- To request yarn to provide a more detailed log at container level, we run:
  - **yarn logs -applicationId APPLICATION_ID -containerId CONTAINER_ID --nodeAddress NODE_ADDRESS \ | grep -v INFO**
  - yarn logs -applicationId application_1542484792469_0142 \
    -containerId container_e56_1542484792469_0142_01_000012 \
    -nodeAddress dsci029.palmetto.Clemson.edu \
    | grep -v INFO

```
Container: container_e56_1542484792469_0142_01_000006 on dsci022.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000006 on dsci022.palmetto.clemson.edu_45454
19/01/25 12:32:07 INFO compress.CodecPool: Got brand-new decompressor [.deflate]
Container: container_e56_1542484792469_0142_01_000012 on dsci029.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000012 on dsci029.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000012 on dsci029.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000012 on dsci029.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000012 on dsci029.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000011 on dsci029.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000011 on dsci029.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000011 on dsci029.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000011 on dsci029.palmetto.clemson.edu_45454
Container: container_e56_1542484792469_0142_01_000011 on dsci029.palmetto.clemson.edu_45454
19/01/25 12:32:07 INFO compress.CodecPool: Got brand-new decompressor [.deflate]
Container: container_e56_1542484792469_0142_01_000014 on dsci032.palmetto.clemson.edu_45454
```

Debugging Error (Cont.)

```
Container: container_e56_1542484792469_0142_01_000012 on dsci029.palmetto.clemson.edu_45454
LogAggregationType: AGGREGATED
=======================================================================================
LogType:stderr
LogLastModifiedTime:Fri Jan 25 12:09:34 -0500 2019
LogLength:919
LogContents:
import: unable to open X server `' @ error/import.c/ImportImageCommand/369.
import: unable to open X server `' @ error/import.c/ImportImageCommand/369.
/data09/hadoop/yarn/local/usercache/jin6/appcache/application_1542484792469_0142/container_e56_1542484792469_0142_01_000012/./avgRa
tingMapper03.py: line 4: movieFile: command not found
/data09/hadoop/yarn/local/usercache/jin6/appcache/application_1542484792469_0142/container_e56_1542484792469_0142_01_000012/./avgRa
tingMapper03.py: line 5: movieList: command not found
/data09/hadoop/yarn/local/usercache/jin6/appcache/application_1542484792469_0142/container_e56_1542484792469_0142_01_000012/./avgRa
tingMapper03.py: line 7: syntax error near unexpected token `('
/data09/hadoop/yarn/local/usercache/jin6/appcache/application_1542484792469_0142/container_e56_1542484792469_0142_01_000012/./avgRa
tingMapper03.py: line 7: `with open(movieFile, mode = 'r') as infile:'

End of LogType:stderr
```

```
%%writefile codes/avgRatingMapper04.py
#!/usr/bin/env python

import sys
import csv

movieFile = "./movies.csv"
movieList = {}

with open(movieFile, mode = 'r') as infile:
    reader = csv.reader(infile)
    for row in reader:
        movieList[row[0]] = {}
        movieList[row[0]]["title"] = row[1]
        movieList[row[0]]["genre"] = row[2]

for oneMovie in sys.stdin:
    oneMovie = oneMovie.strip()
    ratingInfo = oneMovie.split(",")
    try:
        movieTitle = movieList[ratingInfo[1]]["title"]
        movieGenre = movieList[ratingInfo[1]]["genre"]
        rating = float(ratingInfo[2])
        print ("%s\t%s\t%s" % (movieTitle, rating, movieGenre))
    except ValueError:
        continue
```

- cat -n codes/avgRatingMapper04.py
- mapred streaming \
  -input intro-to-hadoop/movielens/ratings.csv \
  -output intro-to-hadoop/output-movielens-01 \
  -file ./codes/avgRatingMapper04.py \
  -mapper avgRatingMapper04.py \
  -file ./codes/avgRatingReducer01.py \
  -reducer avgRatingReducer01.py \
  -file ./movielens/movies.csv

Fixing the Error

- mapred streaming \
  -input intro-to-hadoop/movielens/ratings.csv \
  -output intro-to-hadoop/output-movielens-02 \
  -file ./codes/avgRatingMapper04.py \
  -mapper avgRatingMapper04.py \
  -file ./codes/avgRatingReducer01.py \
  -reducer avgRatingReducer01.py \
  -file ./movielens/movies.csv
- hdfs dfs -ls intro-to-hadoop/output-movielens-02
- hdfs dfs -cat intro-to-hadoop/output-movielens-02/part-00000 2>/dev/null | head -n 20

```
[jin6@node0397 myhadoop]$ hdfs dfs -ls intro-to-hadoop/output-movielens-02
2020-09-18 09:38:51,058 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... usi
ng builtin-java classes where applicable
Found 2 items
-rw-r--r--   3 jin6 supergroup          0 2020-09-18 09:38 intro-to-hadoop/output-movielens-02/_SUCCESS
-rw-r--r--   3 jin6 supergroup    2139648 2020-09-18 09:38 intro-to-hadoop/output-movielens-02/part-00000
[jin6@node0397 myhadoop]$ hdfs dfs -cat intro-to-hadoop/output-movielens-02/part-00000 2>/dev/null | head -n 20
"Great Performances" Cats (1998)        2.7819905213270144      Comedy|Drama
#1 Cheerleader Camp (2010)      2.75    Drama|Horror|Mystery|Thriller
#Horror (2015)  2.2222222222222223      Documentary
#chicagoGirl: The Social Network Takes on a Dictator (2013)     3.6666666666666665      Comedy|Crime|Drama
$ (Dollars) (1971)      2.75    Western
$1,000 on the Black (1966)      3.0     Drama|Western
$100,000 for Ringo (1965)       2.5     Comedy|Drama
$5 a Day (2008) 2.9716981132075473      Drama
$50K and a Call Girl: A Love Story (2014)       3.75    Animation
$9.99 (2008)    3.1384615384615384      Documentary
$ellebrity (Sellebrity) (2012)  2.25    Comedy|Western
'49-'17 (1917)  2.5     Action|Drama|Thriller|War
'71 (2014)      3.6968911917098444      Action|Adventure|Comedy|Documentary|Fantasy
'Hellboy': The Seeds of Creation (2004) 3.059090909090909       Drama|Thriller
'Human' Factor, The (Human Factor, The) (1975)  2.25    Drama
'Master Harold'... and the Boys (1985)  3.5     Western
'Neath the Arizona Skies (1934) 2.2916666666666665      Action
'Pimpernel' Smith (1941)        3.0     Crime|Drama
'R Xmas (2001)  2.75    Drama|Musical
'Round Midnight (1986)  3.6159420289855073      Drama|Horror|Mystery|Thriller
```

Challenge

- Modify *avgRatingReducer02.py* so that only movies with averaged ratings higher than 3.75 are collected
- Further enhance your modification so that not only movies with averaged ratings higher than 3.75 are collected but these movies also need to be rated at least 5000 times.

```
%%writefile codes/avgRatingMapper04challenge.py
#!/usr/bin/env python

import sys
import csv

movieFile = "./movies.csv"
movieList = {}


with open(movieFile, mode = 'r') as infile:
    reader = csv.reader(infile)
    for row in reader:
        movieList[row[0]] = {}
        movieList[row[0]]["title"] = row[1]
        movieList[row[0]]["genre"] = row[2]

for oneMovie in sys.stdin:
    oneMovie = oneMovie.strip()
    ratingInfo = oneMovie.split(",")
    try:
        movieTitle = movieList[ratingInfo[1]]["title"]
        movieGenre = movieList[ratingInfo[1]]["genre"]
        rating = float(ratingInfo[2])
        if _________:
            print ("%s\t%s\t%s" % (movieTitle, rating, movieGenre))
    except ValueError:
        continue
```

```
yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar \
    -input /repository/movielens/ratings.csv \
    -output intro-to-hadoop/output-movielens-challenge \
    -file _____________ \
    -mapper _____________  \
    -file avgRatingReducer02.py \
    -reducer avgRatingReducer02.py \
    -file ./movielens/movies.csv
```