WorkLoad2:

How to run:

Run by this:

pyspark --master yarn-client --num-executors 8 assignment2-2.py

```
It treats zero value of similarity as Zero It can be changer by changing
the value of 'botheqialstoori' at line 223
Rating input write at line 220
Movie input write at line 221
15movies input write at 222
The number of input movies is flexible the number for predict rating is
2/3 of input movies.
```

Result:

The result Save in /user/wtan4210/assignment/sim

The total time of this is 220s with 8 executors.

Map-reduce design:

Step1: The program will read from 15movies and create a list called 'movielistori' then broadcast this list.

Step2: Read rating file and aggreagateByKey with 8 executors. The result of this step is like this

user_id, (movie_id, rating), (movie_id, rating), ...

Step3: map the result of last step to calculate average rating of a user then give a result the key is movie id. The value of result is (rating-average, rating_of_moive_in_list-average) of 15 movies. If the user dose not watch one of the movie in list it will return None. If the user does not see any movie in list it will be filtered. The output like this:

movie_id, [(a, b), None, …]

Then aggreagateByKey the result will be like this:

movie_id, [[(a, b), None, …], [(a, b), None, …], …]

Step4: Map the result of last step in order to calculate the similarity of each movie. The result will be like this.

movie_id, [1, 0, None, -1, …]

Step5: Use Map function to predict rating and sorted by key.

Rating, movie_id

Then collect the top 50 save as a list and broadcast it. Finally, map movie data to get movie name and output the result.

Performance analyze:

The total run time of large dataset is 220s.

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
7	saveAsTextFile at NativeMethodAccessorImpl.java:-2	2016/05/23 23:40:58	0.4 s	2/2	4/4
6	sortByKey at /home/wtan4210/clouldcomputing/clouldcomputing2/assignment2-2d.py:253	2016/05/23 23:40:58	0.2 s	1/1	2/2
5	sortByKey at /home/wtan4210/clouldcomputing/clouldcomputing2/assignment2-2d.py:253	2016/05/23 23:40:57	0.9 s	1/1	2/2
4	runJob at PythonRDD.scala:393	2016/05/23 23:40:56	1.0 s	2/2 (2 skipped)	9/9 (13 skipped)
3	sortByKey at /home/wtan4210/clouldcomputing/clouldcomputing2/assignment2-2d.py:249	2016/05/23 23:40:54	2 s	1/1 (2 skipped)	8/8 (13 skipped)
2	sortByKey at /home/wtan4210/clouldcomputing/clouldcomputing2/assignment2-2d.py:249	2016/05/23 23:37:19	3.6 min	3/3	21/21
1	collect at /home/wtan4210/clouldcomputing/clouldcomputing2/assignment2-2d.py:230	2016/05/23 23:37:19	98 ms	1/1	2/2
0	saveAsTextFile at NativeMethodAccessorImpl.java:-2	2016/05/23 23:37:17	2 s	1/1	2/2

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
4	sortByKey at /home/wtan4210/clouldcomputing/clouldcomputing2/assignment2-2d.py:249 +details	2016/05/23 23:40:52	2 s	8/8			51.1 MB	
3	aggregateByKey at /home/wtan4210/clouldcomputing/clouldcomputing2/assignment2-2d.py:243 +details	2016/05/23 23:38:06	2.8 min	8/8			289.9 MB	51.1 MB
2	aggregateByKey at /home/wtan4210/clouldcomputing/clouldcomputing2/assignment2-2d.py:239 +details	2016/05/23 23:37:19	47 s	5/5	256.0 KB			289.9 MB

Most of time was spend in aggreagateByKey. That means the shuffle time is quiet long due to the of movie_id and user_id are of a large number.

One of the reason is that in order to make the program robust some judgements are written. For instance, in the similarity calculating, if the left part and right part of the denominator are all zero it will return value of `1 or 0 depend on the value of botheqialstoori' at line 223.` Cause, when both of the similarity is zero that means the two movie is same. They are all marked as average scores.

Another is the number of movies given by user0. The length is flexible. Thus it can be calculated.