ENGG 4030 Homework_1

I declare that the assignment submitted on Elearning system is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty/ .

Signed (Student 柳一村) Date: 14/02/2018

Name LIU Yicun SID 1155092202

*** Due to the fact that I haven't been assigned a IE Cluster account, I simply borrowed the account from one of my classmates (jj015). I assure that I did not discuss any part of my solution with her and no code has been shared.**

Q1

A: Find the Top K Similar Users

Consider the similarity is counted by the common followees of users and the input data is the users and their followers list, we fist process input data to create tuples in the mappers like:

(k1, k2, k3)

which means k1 and k2 both follows k3.

```
# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    fids = line.split()
    pid, fids[0] = fids[0].split(':')
    # increase counters
    for m_index in range(0,len(fids)):
        #for s_index in range(m_index + 1,len(fids)):
        for s_index in range(0,len(fids)):
            if(m_index != s_index):
                print '%s\t%s,%s' % (fids[m_index],fids[s_index], pid)
```

Then we user k1 as the key value to assign tuples to different reducers.
In each reducer, we use k2 as secondary key to conduct secondary sort. In each reducer step, the intermediate value is stored like:

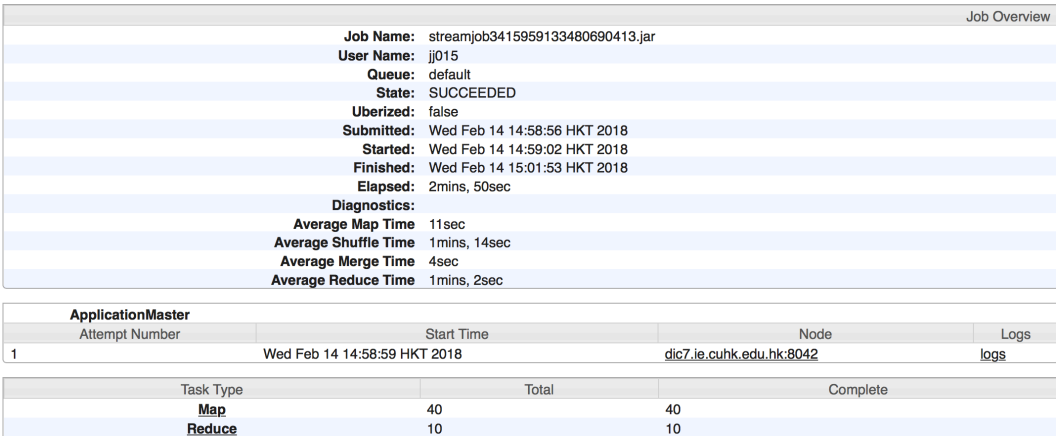
```
sim_arr = {}
id_list = {}
checksum_arr = {}
```

for which the space complexity is O(3N).

Because the output of Q1 is part of the standard output required by Q2, so we simply demonstrate the output at Q2.

Q2: Similar Users, Checksum and Common Followee IDs.

For the reducer constructed in Q1, we tested it on IE Cluster on the small dataset and they apply it on the medium dataset, with 40 mappers and 10 reducers configuration:



Job Overview

Job Name:	streamjob3415959133480690413.jar
User Name:	jj015
Queue:	default
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Feb 14 14:58:56 HKT 2018
Started:	Wed Feb 14 14:59:02 HKT 2018
Finished:	Wed Feb 14 15:01:53 HKT 2018
Elapsed:	2mins, 50sec
Diagnostics:	
Average Map Time	11sec
Average Shuffle Time	1mins, 14sec
Average Merge Time	4sec
Average Reduce Time	1mins, 2sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 14 14:58:59 HKT 2018	dic7.ie.cuhk.edu.hk:8042	logs

Task Type	Total	Complete
Map	40	40
Reduce	10	10

The total time takes about 2 min and 50 sec. And the search result for my SID is:

```
1192202:299983,{511382,536234,337641,1512706,477155},3375118
1192202:477155,{299983,337641,1591997},2229621
1192202:477157,{907962,1591997},2499959
992202:88384,{1250485,931967,193909,521780,2214256,522010,1166296,2685404,1484054},10970161
992202:1291150,{1250485,2214258,2637943,521780,2214256,522010,2685400,1484054,1166296},14696482
992202:1200402,{2637943,193909,521780,2685400,522010,2685404},9246446
1592202:377622,{3228136},3228136
892202:1060765,{2400946},2400946
892202:2835804,{2400946},2400946
892202:2388699,{2400946},2400946
```

Q3: Performance Analysis of Different Number of Mappers and Reducers

For this step, I test on the medium dataset for previous mapper and reducer, with:
Number of mapper: 10,20,40 (m=10,20,40)
Number of reducer: 1,5,10 (r=1,5,10)
Here is the performance table:

	Maximum mapper time	Minimum mapper time	Average mapper time	Maximum reducer time	Minimum reducer time	Average reducer time	Total job time
m=10, r=1	1min 29s	9s	29s	10min 34s	10min 34s	10min 34s	11min 44s
m=20, r=1	1min 14s	4s	16s	9min 17s	9min 17s	9min 17s	10min 48s
m=40, r=1	44s	4s	9s	9min 8s	9min 8s	9min 8s	9min 58s
m=10, r=5	1min 31s	7s	29s	2min 14s	1min 49s	2min 4s	4min 12s
m=20, r=5	1min 9s	4s	15s	2min 11s	1min 49s	1min 56s	3min 43s
m=40, r=5	37s	4s	11s	2min 4s	1min 21s	1min 33s	3min 12s
m=10, r=10	1min 39s	7s	34s	1min 27s	1min	1min 7s	3min 10s
m=20, r=10	1min 2s	5s	15s	1min 23s	53s	1min 8s	3min 2s
m=40, r=10	1min 15s	7s	11s	1min 20s	53s	1min 2s	2min 50s

As indicated by the complexity analysis in my implementation of mapper and reducer, most computational cost is concentrated on reducer side, with is aligned with the actual results.
From the table, we can see that there is some small performance gain when increasing the number of mappers/reducers in the task. The gain is not significant when the number of mappers and reducers is not balanced (e.g. increase mapper number when there are already 40 mappers and only 1 reducer). That is due to the bottleneck exists in the computation involved in the mappers. For mapreduce task, most part of the reducer task cannot start before their corresponding mapper task is finished. That cause the bottleneck in the mappers. For most tasks, we expect larger number in mappers than reducers to ensure the reducers start to compute tasks as soon as possible.

Q4: Large Dataset

For this part, we use the configuration of 40 mappers and 10 reducers, considering we should not take too much resources from the IE clusters.

Job Overview

Job Name:	streamjob8463216408268228116.jar
User Name:	jj015
Queue:	default
State:	SUCCEEDED
Uberized:	false
Submitted:	Wed Feb 14 11:38:41 HKT 2018
Started:	Wed Feb 14 11:38:47 HKT 2018
Finished:	Wed Feb 14 13:22:20 HKT 2018
Elapsed:	1hrs, 43mins, 33sec
Diagnostics:	
Average Map Time	8mins, 11sec
Average Shuffle Time	49mins, 14sec
Average Merge Time	6sec
Average Reduce Time	45mins, 51sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Wed Feb 14 11:38:44 HKT 2018	dicvm1.ie.cuhk.edu.hk:8042	logs

Task Type	Total	Complete
Map	40	40
Reduce	10	10

The total job takes 1 hour 43 hours with most of the time spent on the reducer. Thanks to the 0(3N) complexity in reducer, we can successfully run the large dataset without very large memory and CPU consumption.