Access Log - similar users

Course: 大数据软件工程

Exercise type: PySpark

Soft Due: 2015-06-16 18:00

Hard Due: 2015-06-16 18:00

Author: 王欣明(TEACHER)

Email: wangxm35@mail.sysu.edu.cn

你已经接近或者达到满分。在完成这道题后，如果愿意的话，请你在下面**Feedback**那里评价一下这道题。你的反馈将同时用邮件发送给作者。

Description:

0 Questions and 0 Answers | Ask New Question

In this final project, the data you are going to analyze is the dataset consists of requests made to the 1998 World Cup Web site.
The data format is described at http://ita.ee.lbl.gov/html/contrib/WorldCup.html. We will only use part of the full dataset.
Here are some samples:

```
34600 - - [30/Apr/1998:21:30:17 +0000] "GET /images/hm_bg.jpg HTTP/1.0" 200 24736
232 - - [30/Apr/1998:21:30:53 +0000] "GET /images/hm_bg.jpg HTTP/1.0" 200 24736
282 - - [30/Apr/1998:21:31:12 +0000] "GET /images/hm_bg.jpg HTTP/1.0" 200 24736
9719 - - [30/Apr/1998:21:31:19 +0000] "GET /french/splash_inet.html HTTP/1.0" 200 3781
9719 - - [30/Apr/1998:21:31:22 +0000] "GET /images/hm_nbg.jpg HTTP/1.0" 304 -
252 - - [30/Apr/1998:21:31:27 +0000] "GET /images/hm_bg.jpg HTTP/1.0" 200 24736
9719 - - [30/Apr/1998:21:31:29 +0000] "GET /images/hm_brdl.gif HTTP/1.0" 304 -
9719 - - [30/Apr/1998:21:31:29 +0000] "GET /images/hm_arw.gif HTTP/1.0" 304 -
9719 - - [30/Apr/1998:21:31:32 +0000] "GET /images/nav_bg_top.gif HTTP/1.0" 200 929
9719 - - [30/Apr/1998:21:31:43 +0000] "GET /french/images/nav_venue_off.gif HTTP/1.0" 304 -
9719 - - [30/Apr/1998:21:31:44 +0000] "GET /french/images/nav_hosts_off.gif HTTP/1.0" 200 1139
```

The first item in each line is the user index (identified by the IPs. The real IP is removed for privacy concern).

To assist your task, we have built a url dictionary file that maps each URL to an index. This dictionary is provided to you by url_dict in the main function. Note: due to unicode problem, not all urls in the log file will find the entry in url_dict. You can ignore such urls.

0 /images/home_intro.anim.gif
1 /images/home_bg_stars.gif
2 /images/home_fr_phrase.gif
3 /images/nav_bg_top.gif
4 /images/home_logo.gif
5 /images/logo_cfo.gif
6 /images/home_eng_phrase.gif
7 /english/index.html
8 /english/frntpage.htm
9 /images/hm_f98_top.gif
10 /images/team_hm_concacaf.gif
11 /images/team_hm_afc.gif
12 /images/team_hm_caf.gif
13 /english/playing/mascot/mascot.html
14 /images/home_tool.gif
15 /english/images/comp_bu_stage1n.gif
16 /english/images/comp_bu_stage2n_on.gif
17 /images/comp_stage2_brc_top.gif

....

Now your task is to find top-1000 pairs of users (identified by their user_index) whose visited URLs are the most similar. The similarity is measured by Jaccard coefficient. Formally, the Jaccard coefficient between a pair of users A, B is defined as:

|the set of URLs visited by both A and B| / |the set of URLs that are visited by either A or B|

The output format shall be "Jaccard-similarity-value user_idx1 user_idx2" in each line. For example:

0.90123 123 2312

0.81232 2343 45413

....

....

Requirement:

- Pairs of users who visit the same set of URLs (that is, the jaccard similarity is 1.000) are ignored. These users are likely the same.
- Only users who visit 10 or more urls are considered.
- Your answer can have a limited amount of false positives/negatives (that is, not exactly the same as the standard answer).

Note:

- Items in each line are separated by "\t"
- The list is ordered by the jaccard-similarity-value, from high to low.
- For each pair, the smaller user index is listed first.
- The jaccard similarity value is accurate to five decimal places. For example: 1.00000,

Hint:

1. Don't try to do cartesian and compute jaccard similarity for every pair. This naive algorithm will certainly fail. Try to find some heuristic way to prune unlikely pairs.

2. Use broadcast variable to share data among executors. For example, put this in the main_func to create a broadcast variable for shared_dict

```
if __name__ == "__main__":
    # Only run this on the driver node
    global shared_dict
    shared_dict = sc.broadcast(url_dict)
```

3. Check http://en.wikipedia.org/wiki/MinHash

4. You can download the log file and the dictionary for local testing purpose at http://ita.ee.lbl.gov/html/contrib/WorldCup.html (the dictionary is called object_mappings.sort and can be found inside ftp://ita.ee.lbl.gov/software/WorldCup_tools.tar.gz)

Your answer:

driver.py | [*]main.py

```
1.  import
2.  from pyspark import SparkContext
3.
4.  if __name__ == "__main__":
5.        = SparkContext(       ="Co-occurrence")
6.
7.              = {}
8.      f = open("/projects/smart_programmer/smart_programmer/media/[2015] BDSE by Dr. Wang/object_mappings.sort", "r")
9.      for           :
10.         line = line.strip()
11.           =      .     (" ")
12.         url_id = int(line[:idx])
13.             =     [   +1:].       ()
14.         url_dict[url]=url_id
15.     .     ()
16.     rdd = sc.textFile("hdfs://eden/user/driver/smart_programmer/worldcup/w")
17.             (  ,    ,          )
18.
19.
```

Upload Note: The files with "[*]" are those you need to provide source code as answer.

You have reach 100 points. The standard answer is unlocked:

If you want to study the standard answer, you can add it as an optional assignment.

Add this exercise as an optional assignment

driver.py | [*]main.py

```
1.  import
2.  from pyspark import SparkContext
3.
```

```
4.  if __name__ == "__main__":
5.          = SparkContext(        ="Co-occurrence")
6.
7.                  = {}
8.      f = open("/projects/smart_programmer/smart_programmer/media/[2015] BDSE by Dr. Wang/object_mappings.sort", "r")
9.      for       in   :
10.         line = line.strip()
11.             =       .     (" ")
12.         url_id = int(line[:idx])
13.             =      [    +1:].       ()
14.         url_dict[url]=url_id
15.      .      ()
16.     rdd = sc.textFile("hdfs://eden/user/driver/smart_programmer/worldcup/w")
17.             (   ,     ,          )
18.
19.
```

Latest Submission Grade: 100 (submitted on 2015-06-04 14:34)

```
==================================[Phase1: check_plagirism_cloudera]==================================
Pass.

==================================[Phase2: cloudera_prepare]==========================================
Pass syntax checking. You got 10 points.

==================================[Phase3: cloudera_pyspark_cluster]==================================
Spent time: 78.1557919979 seconds. The standard execution time is 111 seconds.
Your spark program runs faster than the standard answer. How do you manage to achieve this? You get 100% c

Correctness points: 90
Your answer is roughly the same as the standard answer
```

Your submission record:

Submission Date	Grade	Diff
2015-06-04 14:34	100	Diff
2015-06-04 14:31	76	Diff
2015-06-04 13:50	76	Diff
2015-06-04 12:41	77	Diff
2015-06-04 12:37	76	Diff
2015-06-04 12:33	76	Diff
2015-06-04 12:31	76	Diff
2015-06-04 12:28	76	Diff
2015-06-04 12:23	46	Diff
2015-06-04 12:19	76	Diff
2015-06-04 12:15	76	Diff
2015-06-04 11:47	42	Diff
2015-06-04 11:32	12	Diff
2015-06-04 11:24	12	Diff
2015-06-04 11:19	10	Diff
2015-06-04 11:14	10	Diff
2015-06-04 11:09	12	Diff
2015-06-04 11:05	10	Diff
2015-06-04 11:01	10	Diff
2015-06-03 17:01	10	Diff
2015-06-03 16:58	10	Diff
2015-06-03 16:56	10	Diff
2015-06-03 16:44	10	Diff
2015-06-03 16:42	10	Diff

2015-06-03 16:40	10	Diff
2015-06-03 16:38	10	Diff
2015-06-03 16:16	12	Diff
2015-06-03 16:11	0	Diff
2015-06-03 16:09	11	Diff
2015-06-03 16:06	11	Diff
2015-06-03 16:03	10	Diff
2015-06-03 15:59	10	Diff
2015-06-03 15:58	10	Diff
2015-06-03 15:54	10	Diff
2015-06-03 15:52	10	Diff
2015-06-03 15:50	10	Diff
2015-06-03 15:48	10	Diff
2015-06-03 15:46	10	Diff
2015-06-03 15:44	10	Diff
2015-06-03 15:39	10	Diff
2015-06-03 15:17	10	Diff
2015-06-03 15:10	10	Diff
2015-06-03 14:30	10	Diff
2015-06-03 14:26	10	Diff
2015-06-03 12:34	10	Diff
2015-06-03 12:00	10	Diff
2015-06-03 11:59	10	Diff
2015-06-03 11:19	10	Diff
2015-06-03 11:19	10	Diff
2015-06-03 11:14	10	Diff
2015-06-03 10:51	10	Diff
2015-06-03 10:50	10	Diff
2015-06-03 10:49	10	Diff

Top performers in the class:

Name	Points	Total Trials	Last Submission
贾同学	100	2	2015-06-02 12:35
DIS	100	53	2015-06-04 14:34
乙同学	100	20	2015-06-05 10:18
［征婚广告位］	100	21	2015-06-05 17:55
完结撒花	100	43	2015-06-05 21:35
哈哈哈	100	5	2015-06-06 10:39
把我飘准的普通发凡我	100	17	2015-06-06 13:47
周德友大哥	100	4	2015-06-06 19:06
持盾卫士	100	21	2015-06-06 20:07
XiaoMoMo	100	40	2015-06-06 23:49

Feedback: (这里不是填答案的地方！)

Warning: Don't put your answer into the feedback, as they will be seen by all other students. If you have doubts on your answer, send an email to the author of this assignment instead.

为减轻服务器负担，在评论中插入大图像时建议用链接，小图像的时候建议用base64编码直接嵌入，即打开html编辑页面插入像<img src="">一样的标签 (图片转base64编码的工具)。注意，评论的大小不能超过100k字节，如果超出就会显示Invalid form data。

难度(选填): ----------

题意表达是否清楚(选填): ----------

对学习的帮助(选填): ----------

你对本题的评价(选填):

Font Family | Font Size | Paragraph

Path: p

Upload

Feedbacks from students:

Class	Comments

BDSE_2015	==========[Phase1: check_plagirism_cloudera]==========
	===
	Pass.
	
	==========[Phase2: cloudera_prepare]==========
	===
	Pass syntax checking. You got 10 points.
	
	==========[Phase3: cloudera_pyspark_cluster]==========
	===
	The spark execution fails. You get 0 out of totally 90 points.
	Failure: error code:1
	
	Output content:
	  File "/tmp/smart_programmer/assignment-4236/submission-1199423/driver.py", line 39
	    print resultimport sys
	                       ^
	SyntaxError: invalid syntax
	
	Detailed log:
	
	driver.py为什么没有from main import main_func这行?
	为什么显示driver.py错误? driver.py本身都修改不了的，还有根本没有line 39 这行 print resultimport sys

BDSE_2015	。
BDSE_2015	楼上的。。。在你的main.py最后加个空行，就好了 不要问我为什么= =
BDSE_2015	楼上正解

BDSE_2015

```
import sys
#from pyspark import SparkContext
#from pyspark import SparkConf, SparkContext
#from main_func import main_func
from operator import add
import string




def url2index(line):
url_left = line.find("\"")
url_right = line.rfind("\"")
if url_left == -1 or url_right == -1:
return
url = line[url_left + 1 : url_right]
url = url.split(" ")[ 1 ]

if shared_dict.value.has_key(url) == False:
return (-1,0)
url_id = shared_dict.value[url]

index = line.find(" ")
user_id = line[:index]
return(user_id,url_id)

def map2((user_id,url_id)):
return (user_id, 1)



def map3(x):
record = {}
for items in x[ 1 ]:
record[str(items)] = 1
return (x[ 0 ], record)

def filter_1(x):
return len(x[1]) >= 10 and x[ 0 ] != -1

def map4(x):
k = []
for i in range(0,int(len(x[1]) * 0.03) + 1):
k.append((len(x[ 1 ]) - i, (x[ 0 ],x[ 1 ])))
return k

def map5(x):

interset = 0
MAP_LEFT = {}
MAP_RIGHT = {}
MAP_LEFT = MAP_LEFT.fromkeys(x[1][0][1])
MAP_RIGHT =MAP_RIGHT.fromkeys(x[1][1][1])

for item in x[1][0][1]:
if MAP_RIGHT.has_key(item):
interset += 1
#return ((interset,(x[1][0][0],x[1][1][0])))
return (float(interset) / (len(x[1][0][1]) + len(x[1][1][1]) - interset), (x[1][0][0],x[1]
[1][0]))

def myOrder(x):
return -x[0]

def filter_2(x):
return x[ 0 ] != 1.0

def map6(x):
l1 = []
l1.extend(x[ 1 ])
l = []
if len(x[ 1 ]) > 80:
i = 0
while(i < len(x[1])):
l.append(l1[i])
```

```
i += len(x[1]) / 80
else:
l = x[1]
return (( len(x[ 1 ]), (x[ 0 ],l) ))

def map7(x):
l1 = []
l1.extend(x[ 1 ])
l = []
if len(x[ 1 ]) > 80:
i = 0
while(i < len(x[1])):
l.append(l1[i])
i += (len(x[1]) / 80)
else:
l = x[1]
return ((x[ 0 ],l))

def testmap(x):
return ((x[0],(x[1][0][0],x[1][1][0])))
def testmap2(x):
return (x[0],x[1])
def main_func(sc, RDD, url_dict):
#print "!!!!!!!!!!!!!!!!!!!!!!I'm here!!!!!!!!!!!!!!!!!!!!\n\n\n\n"
if __name__ == "__main__":
# Only run this on the driver node
global shared_dict
shared_dict = sc.broadcast(url_dict)

# RDD1 = RDD.map(url2index).distinct().groupByKey().filter(filter_1)
# RDD2 = RDD1.flatMap(map4)
# RDD3 = RDD2.join(RDD1.map(map6))
# res = RDD3.map(map5).filter(filter_2).takeOrdered(1000,myOrder)
RDD1 = RDD.map(url2index).distinct().groupByKey().map(map7).filter(filter_1)
res =
RDD1.flatMap(map4).join(RDD1.map(map6)).map(map5).filter(filter_2).takeOr
dered(1000,myOrder)
isin = {}
for item in res:
a = int(item[1][0])
b = int(item[1][1])
if a < b:
if isin.has_key(item[1][0] +"-" + item[1][1]) == False:
print "%.5f %s %s" % (item[ 0 ],item[ 1 ][ 0 ],item[1][1])
isin[item[1][0]+"-"+item[1][1]] = 1
else:
if isin.has_key(item[1][1]+"-"+item[1][0]) == False:
print "%.5f %s %s" % (item[ 0 ],item[ 1 ][ 1 ],item[1][0])
isin[item[1][1]+"-"+item[1][0]] = 1
```


Recent messages:

- [06/01 14:50] From 王欣明: 最近一段时间经常发生有些同学的作业卡队列的现象。如果发现请及时通知我重启服务队列。
- [06/01 14:49] From 王欣明: 本周是大数据软件工程的期末课程项目，题目会在周二晚上6点开放，同学们可以自己在宿舍做，一周时间内完成。
- [05/31 00:38] From 谢议尊: test

Send message to an user with real name or nick name:

Name:

Message:

Send Message