به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

درس کلان داده

تمرین شماره یک

نام و نام خانوادگی : هاتف علی‌پور

شماره دانشجویی : ۸۱۰۱۹۷۳۲۲

اسفند ماه ۱۳۹۸

فهرست گزارش سوالات

بخش اول – عنوان سوال

برای نصب هدوپ از سندباکس Hortonworks نسخه ۲,۵ آن استفاده شده است(بجای CDH). طریقه نصب هم به این صورت می‌باشد که ابتدا ایمیج این سندباکس با استفاده از دستور زیر استخراج می‌گردد:

Docker pull start-sandbox-hdp-25.shd

سپس با استفاده از اسکریپت زیر که در گیتهاب می‌باشد، سندباکس را start می‌کنیم دلیل استفاده از اسکریپت هم این است که هر بار مجبور نباشیم هنگام استارت کردن سندباکس کلی پورت را به صورت دستی باز کنیم:

آدرس اسکریپت:https://bit.ly/2WP88IJ

طریقه فراخوانی اسکریت: به محل دانلود اسکریپت رفته و دستور زیر را بزنید:

./start-sandbox-hdp-25.sh

در شکل زیر طریقه استارت سندباکس را می‌بینیم:

```
➜  ~ ./start-sandbox-hdp-25.sh
Waiting for docker daemon to start up:
02267ab116a9        sandbox             "/usr/sbin/sshd -D"   4 months ago        Exited (255) 23 minutes ago                       sandbox
sandbox
Starting Flume                                             [  OK  ]
Starting Postgre SQL                                       [  OK  ]
Starting name node                                         [  OK  ]
Starting mysql                                             [  OK  ]
Starting Zookeeper nodes                                   [  OK  ]
Starting data node                                         [  OK  ]
Starting Ranger-admin                                      [  OK  ]
```

برای بردن فایل هم به صورت زیر عمل می‌شود که ابتدا با استفاده از scp فایل را از کامپیوتر به داخل سندباکس می‌بریم مطابق شکل زیر:

```
➜  ~ scp -P 2222 /home/hatef/Downloads/apache_log/access_log/log.txt  maria_dev@127.0.0.1:/home/maria_dev/bigdata/log.txt
maria_dev@127.0.0.1's password:
log.txt                                                                                     100%   10MB  30.6MB/s   00:00
```

سپس با استفاده از دستور زیر فایل را از داخل فایل سیستم محلی سندباکس به hdfs می‌بریم:

```
[maria_dev@sandbox bigdata]$ hadoop fs -put ~/bigdata/log.txt /user/maria_dev/bigdata1
[maria_dev@sandbox bigdata]$ hadoop fs -ls /user/maria_dev/bigdata1
Found 1 items
-rw-r--r--   1 maria_dev hdfs   10813147 2020-03-27 13:00 /user/maria_dev/bigdata1/log.txt
[maria_dev@sandbox bigdata]$
```

همچنین بدلیل ساپورت نکردن سیستم با هماهنگی تدریسیار محترم کلیه سوال‌ها با استفاده از ۱۰۰۰۰۰ لاگ اول جواب داده شده است که فایل لاگ مورد نظر هم داخل پوشه جواب می‌باشد به اسم log.txt

بخش دوم – عنوان سوال

زمان اجرا در حالتی که لوکال (خارج از هدوپ) اجرا کردیم خیلی کمتر از زمانی بود که برنامه را روی هدوپ اجرا کردیم دلیل این امر هم این است که کار مپ/ردیوس خیلی سربار دارد و در صورتی که حجم فایلی که می‌خواهیم پردازش کنیم زیاد نباشد، این سربار زمان زیادی را از ما می‌گیرد

در مود دوم اجرا که برنامه را روی هدوپ اجرا کردیم ولی آدرس‌دهی فایل به صورت لوکال بود مشاهده کردیم که فایل ابتدا در یک مکان موقت (/tmp) روی hdfs آپلود شد سپس اجرا شروع شد و پس از پایان اجرا فایل از آن مکان موقت حذف شد. زمان اجرا در این مود بیشترین بود.

در مود سوم هم یک برنامه نرمال مپ/ردیوس اجرا شد که فایل در hdfs بود و برنامه با آدرس‌دهی مناسب فایل شروع به اجرا کرد.

نحوه اجرا برنامه به صورت لوکال که فقط مپ/ردیوس شبیه‌سازی می‌شود:

```
/usr/bin/python3.6 /home/hatef/PythonProject/hate/mapred.py /home/hatef/courses/term-4/hw1/2600-0.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/mapred.hatef.20200303.112948.592910
Running step 1 of 2...
Running step 2 of 2...
"the"	34725
job output is in /tmp/mapred.hatef.20200303.112948.592910/output
"and"	22307
Streaming final output from /tmp/mapred.hatef.20200303.112948.592910/output...
"to"	16755
Removing temp directory /tmp/mapred.hatef.20200303.112948.592910...
"of"	15008
"a" 10584
"he"	10007
"in"	9036
"that"	8205
"his"	7984
"was"	7361

Process finished with exit code 0
```

نتایج حاصل از مود دوم در فایل word_count_hadoop_local.txt

نتایج حاصل از اجرا مود سوم در فایل word_count_hadoop_hdfs.txt

همچنین دستور زده شده و نتایج حاصل از اجرا برنامه هم در فایل‌های ذکر شده می‌باشد.

بخش سوم – عنوان سوال

برای این سوال چند فرض گذاشتیم اول اینکه اسم تصویر برای ما مهم است ممکن است این تصویر در یک مکان دیگر در فایل سیستم باشد و یا با حروف بزرگ و کوچک نوشته شده باشد که این موارد لحاظ نشده است دلیل این امر هم این است که اگر این موارد لحاظ می‌شد نتایج را خیلی جالب نمی‌کرد مثلا گزارش اینکه pic.png ۱۰ بار در سال ۲۰۱۸ ولی Pic.png ۱۱ بار دیده شده است کمی گیج کننده است.

کد برنامه هم به این صورت است که ابتدا پارسر مربوط به لاگ آپاچی نصب شود سپس پترن مناسب را به آن می‌دهیم که این موارد در فایل برنامه به آن اشاره شده است.

ابتدا لاگ درخواست را lowercase می‌کنیم و می‌بینیم که آیا انتهای آن به png یا jpg ختم می‌شود و ابتدای آن با images/newspic شروع می‌شود. در صورت تحقق این امر سال درخواست و اسم تصویر را استخراج کرده و نگاشتی به صورت (imagename_year,1) ایجاد می‌کنیم. در مرحله‌ی کاهش هم تنها این یک‌ها را با هم جمع می‌کنیم.

در نگاشت/کاهش بعدی هم ابتدا در مرحله نگاشت سال و میزان مشاهده و اسم تصویر را استخراج می‌کنیم(تمام این اطلاعات در مرحله کاهش قبل موجود است) و خروجی‌ای به صورت (year,(imagename,count)) ایجاد می‌کنیم

و در مرحله‌ی کاهش هم تنها نتایج هر سال را مرتب می‌کنیم به صورت نزولی و ۱۰ تای اول را برمی‌گردانیم.

نتایج و نحوه اجرا در فایل most_visited_image.txt است


```
"2009"  [["lalapiposleeve_thumb.jpg", 341], ["loveexposuresleeve_thumb.jpg", 222], ["suspiriadvd_thumb.jpg", 181], ["krusty_thumb.jpg", 137],
["mike5_thumb.jpg", 122], ["191-in-the-electric-mist-stills-4-2362x1575_thumb.jpg", 53], ["thirst_tvd4018_2d_thumb.jpg", 12]]
"2010"  [["thirst_tvd4018_2d_thumb.jpg", 597], ["191-in-the-electric-mist-stills-4-2362x1575_thumb.jpg", 576], ["suspiriadvd_thumb.jpg", 573],
["loveexposuresleeve_thumb.jpg", 438], ["funnygamessleevecrop_thumb.jpg", 309], ["triffids_front_sleeve_thumb.jpg", 193], ["shouting_men_003_thumb.jpg",
170]]
```

بخش چهارم – عنوان سوال

تعداد بازدید ماهیانه کاربر به این صورت محاسبه شده است که ابتدا با لاگ پارسر بخش‌های مختلف لاگ را استخراج کردیم خروجی نگاشت اول به صورت (ip#year#month#day, 1) می‌باشد و خروجی کاهش اول هم همان ورودی کاهش است یعنی (ip#year#month#day, 1) سپس در نگاشت دوم روز را از کلید حذف کردیم یعنی خروجی نگاشت دوم (ip#year#month, 1) است با اینکار هر ip را فارغ از اینکه چندبار به سایت سر زده است تنها یکبار حساب کردیم و در کاهش دوم هم تنها روی مقدارها عملیات جمع انجام دادیم و خروجی را به صورت (ip-year/month, totalvisit) ایجاد کردیم.

نحوه اجرا فایل و خروجی در فایل ip_visit_permonth.txt است.

```
job output is in hdfs:///user/maria_dev/tmp/mrjob/visit.maria_dev.20200327.150
Streaming final output from hdfs:///user/maria_dev/tmp/mrjob/visit.maria_dev.2
"10.1.181.142-2010/1"   1
"10.1.6.32-2010/1"  4
"10.10.116.110-2010/1"  1
"10.10.118.62-2010/1"   1
"10.10.84.48-2010/1"    1
"10.10.94.210-2010/1"   1
"10.100.194.188-2010/1" 2
"10.100.224.63-2010/1"  1
"10.102.126.202-2010/1" 5
"10.102.177.136-2010/1" 1
"10.102.40.139-2010/1"  1
"10.102.80.129-2010/1"  1
"10.104.100.229-2010/1" 1
"10.104.145.74-2010/1"  8
"10.104.151.50-2010/1"  1
"10.104.159.78-2010/1"  1
"10.104.190.155-2010/1" 1
"10.104.58.176-2010/1"  1
"10.104.62.79-2010/1"   4
"10.105.102.222-2010/1" 4
"10.105.131.222-2010/1" 1
"10.105.14.50-2010/1"   1
"10.105.153.33-2010/1"  1
"10.106.132.113-2010/1" 1
"10.106.234.115-2010/1" 5
"10.107.129.178-2010/1" 1
```

برای بخش دوم سوال هم مفهومی به اسم session_break که مقدار آن برابر با ۳۰ دقیقه است و قابل تنظیم می‌باشد معرفی شد. Session_break به این صورت است که در صورتی که بین کل درخواست‌های یک کاربر به سایت که به صورت صعودی مرتب شده‌اند یک فاصله‌ی ۳۰ دقیقه‌ای وجود داشته باشد یک

session جدید ایجاد شده است و این session از session قبلی متمایز است. به این صورت Session ها به ازای هر کاربر ساخته شد و در آخر مجموع زمان sessionها محاسبه گردید.

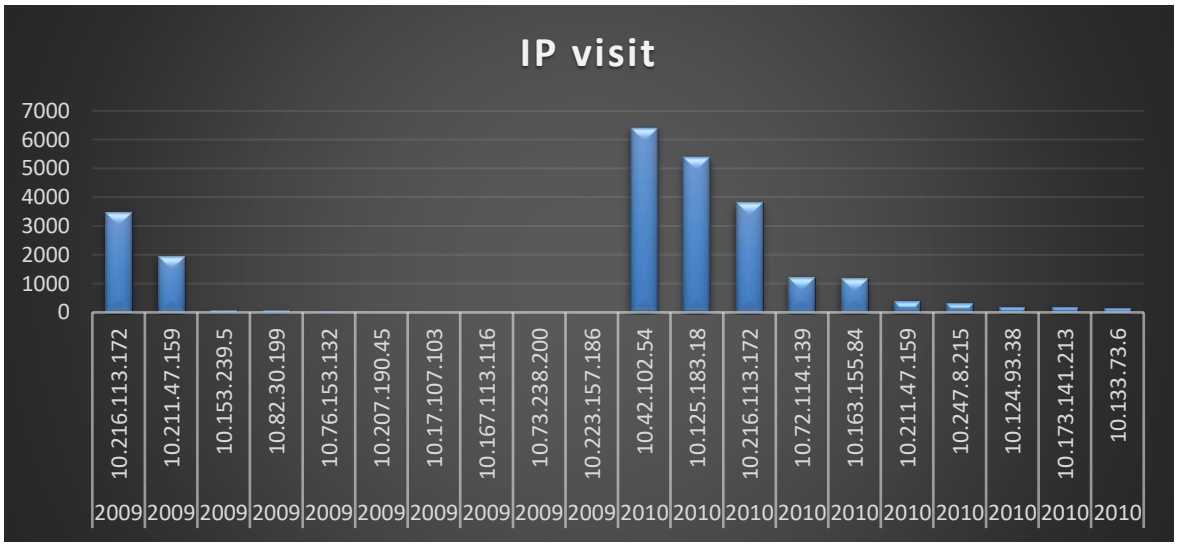
ابتدا در نگاشت اول با استفاده از لاگ پارسر بخش‌های مختلف لاگ استخراج گردید و خروجی به صورت (ip_year, datetime) تولید گردید. سپس در کاهش اول زمان سشن‌ها مطابق توضیحات ارائه شده محاسبه گردید و خروجی کاهش اول به صورت (year, ip#totaltime) تولید گردید

در مرحله‌ی دوم نگاشت نداشیم اما در مرحله‌ی کاهش تنها نتایج مرتب شدند و ۱۰ تای اول استخراج گردید.

نتایج و خروجی در فایل visittime.txt است.

اسم اسکریپت محاسبه تعداد بازدید visit.py و اسم اسکریپت محاسبه زمان بازدید VisitTime.py است.

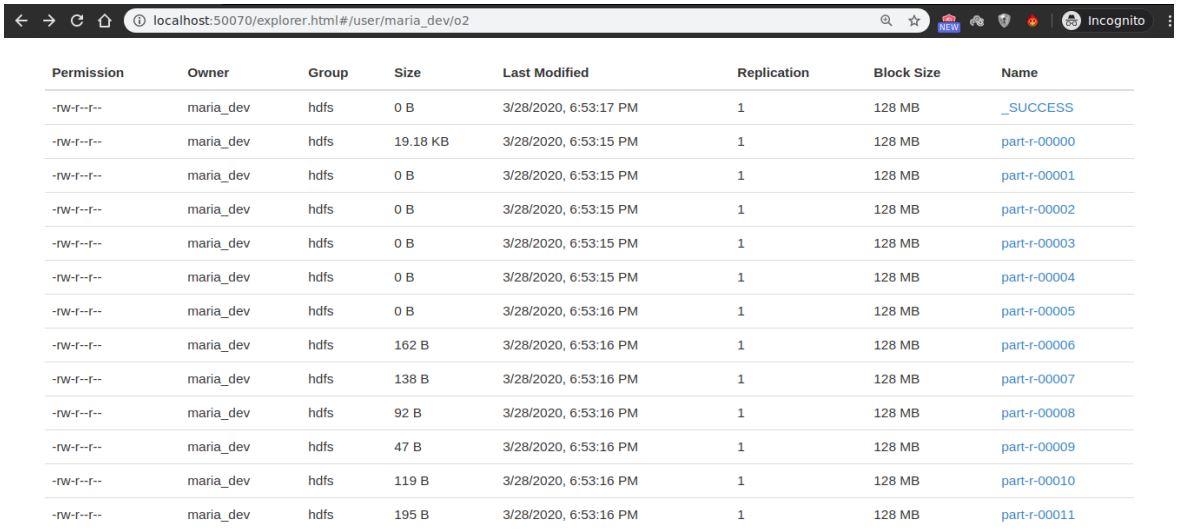
نمودار ستونی مربوطه(مدت زمان به دقیقه است)



بخش پنجم – عنوان سوال

در این بخش عملا کد پایتون تبدیل به جاوا شده است و تعداد کاهش‌ها برابر با ۱۲ لحاظ شده است. همچنین در کد جاوا هر نگاشت/کاهش به عنوان یک جاب درنظر گرفته می‌شود این جاب از یک ورودی خوانده و نتایج را در یک خروجی می‌نویسد اگر بخواهیم یک نگاشت/کاهش دیگر داشته باشیم باید یک جاب دیگر تعریف کنیم و ورودی این جاب خروجی جاب قبلی است و این جاب تنها در صورتی مجاز به لانچ شدن است که جاب قبلی تمام شده باشد. برای بحث پارتیشن کردن هم از این موضوع استفاده شد که زمان درخواست هم در هر درخواست گنجانده شده است. با استفاده از API خود جاوا ماه را از زمان درخواست استخراج کردیم و یک عدد از ۰ تا ۱۱ براساس ماه برگرداندیم دقت شود که خروجی نگاشت اول که درواقع ورودی پارتیشنر می‌باشد به این صورت است (ip#year#month#day, 1) در حالت محاسبه تعداد بازدید و به صورت (ip_year, datetime) در هنگام محاسبه زمان بازدید می‌باشد پس می‌توان ماه را از آن استخراج کرد و یک عدد بین ۰ تا ۱۱ برگرداند.

کد در واقع همان کد پایتون است منتها یک پارتیشنر به آن اضافه شده است. در شکل زیر تمام فایل‌های تولید شده توسط این برنامه دیده می‌شود که به ازای در کاهش یک فایل ایجاد شده است.



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	maria_dev	hdfs	0 B	3/28/2020, 6:53:17 PM	1	128 MB	_SUCCESS
-rw-r--r--	maria_dev	hdfs	19.18 KB	3/28/2020, 6:53:15 PM	1	128 MB	part-r-00000
-rw-r--r--	maria_dev	hdfs	0 B	3/28/2020, 6:53:15 PM	1	128 MB	part-r-00001
-rw-r--r--	maria_dev	hdfs	0 B	3/28/2020, 6:53:15 PM	1	128 MB	part-r-00002
-rw-r--r--	maria_dev	hdfs	0 B	3/28/2020, 6:53:15 PM	1	128 MB	part-r-00003
-rw-r--r--	maria_dev	hdfs	0 B	3/28/2020, 6:53:15 PM	1	128 MB	part-r-00004
-rw-r--r--	maria_dev	hdfs	0 B	3/28/2020, 6:53:16 PM	1	128 MB	part-r-00005
-rw-r--r--	maria_dev	hdfs	162 B	3/28/2020, 6:53:16 PM	1	128 MB	part-r-00006
-rw-r--r--	maria_dev	hdfs	138 B	3/28/2020, 6:53:16 PM	1	128 MB	part-r-00007
-rw-r--r--	maria_dev	hdfs	92 B	3/28/2020, 6:53:16 PM	1	128 MB	part-r-00008
-rw-r--r--	maria_dev	hdfs	47 B	3/28/2020, 6:53:16 PM	1	128 MB	part-r-00009
-rw-r--r--	maria_dev	hdfs	119 B	3/28/2020, 6:53:16 PM	1	128 MB	part-r-00010
-rw-r--r--	maria_dev	hdfs	195 B	3/28/2020, 6:53:16 PM	1	128 MB	part-r-00011

همچنین برای پکیجینگ از maven استفاده شده است با دستور : mvn clean package

سپس فایل جار ایجاد شده به سندباکس منتقل گردید با استفاده از Scp و با استفاده از دستور اجرا گردید:

Hadoop jar filename.jar pathtomainclass

کلاس‌های ایجاد شده، فایل‌های تولید شده توسط برنامه و دیگر موارد در پوشه sec5 می‌باشد.

پوشه‌ی time of visit بخشی از برنامه هست که مدت زمان بازدید را محاسبه می‌کند کد برنامه و نحوه‌ی اجرا در پوشه است.

پوشه‌ی number of visits تعداد باری که هر آی پی به سایت سر زده است را محاسبه می‌کند.
