



به نام خداوند بخشنده و مهربان

استاد: محمدعلی نعمت‌بخش
دستیاران: فاطمه ابراهیمی، پریسا لطیفی، امیر سرتیپی

تمرین سوم: زمان اجرا
درس: تحلیل سیستم داده‌های حجیم

نام و نام خانوادگی: فاطمه مومنی

آدرس گیت: <https://github.com/FatemehMomeni/BD3.git>

- لطفا پاسخ تمرین حتما در سامانه‌ی کوئرا ارسال شود.
- لطفا پاسخ‌های خود را در خود سند سوال نوشته و در قالب یک فایل PDF ارسال کنید.
- نام سند ارسالی {student number}-{Name Family}-{homework number}-HW
- تمامی فایل‌های مورد نیاز این تمرین در [این لینک](#) قابل دسترسی است.
- خروجی از هر مرحله‌ی تمرین را در سند خود بارگذاری کنید.

در این تمرین هدف ما مقایسه زمان اجرا در هدوپ و اسپارک است.

برای این منظور ۴ فایل داده متنی با حجم‌های ۱، ۵، ۱۰ و ۱۲ گیگابایتی در اختیار شما قرار گرفته است که انتظار می‌رود با نوشتن برنامه‌ی شمارش کلمات عملیات نگاشت-کاهش را برای داده‌ها بر روی هدوپ و اسپارک انجام دهید. نتایج را گزارش و مقایسه‌ای بین آنها انجام دهید.

آدرس فایل‌ها:

/user/ebrahimi/hw3-data

نمونه‌ی دستور اسپارک را با client mode هم امتحان کرده و تفاوت حالت cluster و client را بیان کنید.

عملیات نگاشت-کاهش بر روی هدوپ

برنامه شمارش کلمات، در شکل ۱ و شکل ۲ مشاهده می شود. این برنامه در قالب دو فایل mapper.py و reducer.py نوشته شده است. در فایل mapper.py پس از دریافت ورودی، کلمات هر خط را بر اساس فاصله بین آن ها جدا شده و در خروجی به صورت زوج (word,1) چاپ می شود. فایل reducer.py با دریافت خروجی مرحله قبل، تعداد تکرار هر کلمه را شمارش کرده و در خروجی به صورت زوج (word,count) چاپ می کند.

```
#!/usr/bin/env python

from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t', 1)
    try:
        count = int(count)
    except ValueError:
        continue
    if current_word == word:
        current_count += count
    else:
        if current_word:
            print '%s\t%s' % (current_word, current_count)
            current_count = count
            current_word = word
        if current_word == word:
            print '%s\t%s' % (current_word, current_count)
```

```
#!/usr/bin/env python

import sys

for line in sys.stdin:
    line = line.strip()
    words = line.split()
    for word in words:
        print '%s\t%s' % (word, 1)
```

شکل ۱: برنامه شمارش کلمات، عملیات نگاشت

شکل ۲: برنامه شمارش کلمات، عملیات کاهش

به منظور اجرای این برنامه، از دستور نشان داده شده در شکل ۳، استفاده شده است. در این دستور، پس از آدرس پکیج، فایل های ورودی، فایل های mapper و reducer و مسیر ذخیره خروجی، داده می شود.

```
fatemeh_momeni@MasterPC:~$ hadoop jar /home/hduser/hadoop/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.10.1.jar -files ./mapper.py,./reducer.py -mapper mapper.py -reducer reducer.py -input /user/ebrahimi/hw3-data/file1G.txt -output /user/fatemeh_momeni/hadoop_output1G
```

شکل ۳: دستور اجرای برنامه عملیات نگاشت-کاهش روی هدوپ

متأسفانه این دستور با موفقیت اجرا نشد. شکل ۴ بخش انتهایی خطای اجرای این دستور را نشان می دهد.

```

Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess fa
iled with code 1
    at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:325)
    at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:538)
    at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
    at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
    at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
    at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:459)
    at org.apache.hadoop.mapred.MapTask.run(MapTask.java:343)
    at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:177)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1926)
    at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:171)

22/04/10 17:17:22 INFO mapreduce.Job: map 100% reduce 100%
22/04/10 17:17:22 INFO mapreduce.Job: Job job_1649179500517_0538 failed with state FAILED due to: Task failed task_1649179500517_0538_m_000002
Job failed as tasks failed. failedMaps:1 failedReduces:0

22/04/10 17:17:22 INFO mapreduce.Job: Counters: 15
Job Counters
    Failed map tasks=17
    Killed map tasks=7
    Killed reduce tasks=1
    Launched map tasks=19
    Other local map tasks=13
    Data-local map tasks=5
    Rack-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=146136
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=36534
    Total vcore-milliseconds taken by all map tasks=36534
    Total megabyte-milliseconds taken by all map tasks=149643264
Map-Reduce Framework
    CPU time spent (ms)=0
    Physical memory (bytes) snapshot=0
    Virtual memory (bytes) snapshot=0
22/04/10 17:17:22 ERROR streaming.StreamJob: Job not successful!
Streaming Command Failed!

```

شکل ۴: بخشی از خطای اجرای ناموفق دستور راه اندازی هدوپ

عملیات نگاشت-کاهش بر روی اسپارک در cluster mode

برنامه شمارش کلمات در شکل ۵ مشاهده می شود. در این برنامه، پس از پیکربندی spark context، فایل ورودی از مسیر داده شده با تابع `textFile()` خوانده شده و توسط تابع `flatMap()` نگاشتی روی تابع درون آن که خطوط فایل ورودی را براساس فاصله بین کلمات جدا می کند، اعمال می شود. سپس، تابع `map()`، نگاشت دیگری روی خروجی به دست آمده از این مرحله اعمال می کند و زوج `(word,1)` که همان خروجی عملیات نگاشت است را به دست می دهد. در آخر، با تابع `reduceByKey()`، عملیات کاهش براساس کلید (کلمات)، انجام شده و خروجی نهایی توسط تابع `saveAsTextFile()` در مسیر مشخص شده ذخیره می شود.

```

#!/usr/bin/env python

import sys

from pyspark import SparkContext, SparkConf

if __name__ == "__main__":
    sc = SparkContext("local", "Spark Word Count")
    words = sc.textFile("/user/ebrahimi/hw3-data/file12G.txt").flatMap(lambda line: line.split(" "))
    wordCounts = words.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b)
    wordCounts.saveAsTextFile("/user/fatemeh_momeni/spark_output12G")

```

شکل ۵: برنامه شمارش کلمات با عملیات های اسپارک

برای اجرای برنامه مربوط به عملیات نگاشت-کاهش بر روی اسپارک در حالت کلاستر، دستور نشان داده شده در شکل ۶ اجرا شده است.

```
fatemeh_momeni@MasterPC:~$ spark-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode cluster /home/fatemeh_momeni/spark_mr.py
```

شکل ۶: دستور اجرای عملیات نگاشت-کاهش روی اسپارک در حالت cluster

متأسفانه این دستور پس از مدت زمان زیادی با خطای نشان داده شده در شکل ۷ متوقف شد.

```
fatemeh_momeni@MasterPC:~$ spark-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode cluster /home/fatemeh_momeni/spark_mr.py
Exception in thread "main" org.apache.spark.SparkException: Application application_1649179500517_0564 finished with failed status
    at org.apache.spark.deploy.yarn.Client.run(Client.scala:1192)
    at org.apache.spark.deploy.yarn.YarnClusterApplication.start(Client.scala:1583)
    at org.apache.spark.deploy.SparkSubmit.org$apache$spark$deploy$SparkSubmit$$runMain(SparkSubmit.scala:928)
    at org.apache.spark.deploy.SparkSubmit.doRunMain$1(SparkSubmit.scala:180)
    at org.apache.spark.deploy.SparkSubmit.submit(SparkSubmit.scala:203)
    at org.apache.spark.deploy.SparkSubmit.doSubmit(SparkSubmit.scala:90)
    at org.apache.spark.deploy.SparkSubmit$$anon$2.doSubmit(SparkSubmit.scala:1007)
    at org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:1016)
    at org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
```

شکل ۷: خطای راه اندازی اسپارک در حالت cluster

عملیات نگاشت-کاهش بر روی اسپارک در client mode

در این حالت نیز از برنامه شمارش کلمات نشان داده شده در شکل ۵ استفاده شده است. برای اجرای برنامه مربوط به عملیات نگاشت-کاهش بر روی اسپارک در حالت کلاینت، دستور نشان داده شده در شکل ۸ اجرا شده است.

```
fatemeh_momeni@MasterPC:~$ spark-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode client /home/fatemeh_momeni/spark_mr.py
```

شکل ۸: دستور اجرای عملیات نگاشت-کاهش روی اسپارک در حالت client

پس از اجرای دستور فوق، همان گونه که در شکل ۹ مشاهده می شود، به مسیری که فایل خروجی در آن ذخیره شده است رفته و با دستور h-cat محتویات آن را چاپ می کنیم.

```
fatemeh_momeni@MasterPC:~$ ls
hw3-data  mapper.py.save  reducer.py  spark_output
mapper.py  map_reduce     spark_mr.py  user_shell_exporter.sh
fatemeh_momeni@MasterPC:~$ cd spark_output/
fatemeh_momeni@MasterPC:~/spark_output$ ls
part-00000  part-00002  part-00004  part-00006  _SUCCESS
part-00001  part-00003  part-00005  part-00007
fatemeh_momeni@MasterPC:~/spark_output$ h -cat /user/fatemeh_momeni/spark_output/part-00000
```

شکل ۹: دستور چاپ محتویات فایل خروجی

بخشی از خروجی مربوط به هر یک از فایل‌ها در شکل ۱۰، شکل ۱۱، شکل ۱۲ و شکل ۱۳ آورده شده است.

```
('Sesame-Soy', 2)
('Duch', 6)
('bandful', 2)
('aight...', 2)
('focus...', 18)
('squishes', 2)
('waz', 51)
('AUTOBiography', 2)
(' (Martinsburg,', 2)
('mso-hansi-font-family:Calibri;', 22)
('seemsThe', 2)
(' (dk', 2)
('92019.', 2)
('duck!', 52)
('Stumpt', 2)
('Chuffest', 4)
('Hey!', 2)
('invaluable:', 15)
('chorale's', 2)
('Oboe"', 2)
('Tahoe-LAfs', 2)
('perfect"cheap', 2)
('msite', 5)
```

```
('flickering).', 3)
('Nope:', 15)
('192.168.85.16,', 3)
('Triggerx', 3)
('Allpro', 7)
('Laudas', 3)
('Pardas', 3)
('Dosaagne', 3)
('baki', 3)
('bhoom', 7)
('Mantralay's', 3)
('payforessay.Web', 3)
('dhanga', 3)
('padi?', 3)
('GREEN3', 3)
('niche"', 44)
('=23Kg', 3)
('wonder!!!!!!', 3)
('1980s', 5)
('Ginhoven', 6)
('Carnero', 6)
('1,700-2,100', 3)
('P-Diddy', 3)
('Paris"', 3)
```

```
('Ecuaristia', 5)
('CambodiaCM', 5)
('KyrgyzstanLA', 5)
('LibyaLI', 5)
('SlovakiaSI', 5)
('WARMEST', 5)
('L1000:', 5)
('801-863-7923.', 5)
('Vortrag', 24)
('Zetzerin', 10)
('eersteHulpPosten', 5)
('iGen3', 5)
('**VKrug', 5)
('Race', 5)
('PlayMobil', 7)
('female-dominated.', 5)
('sumni', 12)
('devudeSign', 5)
('presentado', 5)
('Klausberg-Flitzer', 5)
('CERTIFICAR', 5)
('Color', 5)
('rosés.', 21)
('agony!'", 5)
```

```
('Kosovars', 2)
('Maine\t16.276', 1)
('582-6787.', 1)
('Impala', 262)
('Widler', 2)
('Zainer', 2)
('Jule', 2)
('Devorah', 18)
('Hilliard', 12)
('Gilkey', 2)
('Ambulance', 12)
('lasted.', 30)
('CHAOS', 2)
('under-performing', 26)
('Myer', 11)
('C-Suite', 38)
('Holtzer', 4)
('absolution', 50)
('vigilantism.', 4)
('Lovato', 78)
('MarieMurphy', 1)
('Resilience', 3)
('Headteacher', 24)
('direction.\x94', 1)
```

شکل ۱۳: شی از خروجی مربوط به فایل file12G.txt

شکل ۱۲: بخشی از خروجی مربوط به فایل file5G.txt

شکل ۱۱: بخشی از خروجی مربوط به فایل file10G.txt

شکل ۱۰: بخشی از خروجی مربوط به فایل file1G.txt

تفاوت حالت client و cluster در اسپارک

در حالت client، spark driver روی ماشینی که کار از آن ارسال شده‌است اجرا می‌شود. به این معنی که درایور به‌صورت محلی و با وارد کردن دستور spark-submit اجرا شده و وظایف روی ماشین‌های کارگر درون cluster اجرا می‌شوند. همچنین log‌های درایور، در ماشین محلی قابل دسترسی می‌باشند. در این حالت، اگر یک جلسه کاربر که در حال اجرای spark-submit است خاتمه یابد، برنامه نیز با شکست خاتمه می‌یابد. این حالت برای تست و debugging به‌کار می‌رود و برای کارهای تولیدی استفاده نمی‌شود. باتوجه به این که در این حالت داده‌ها باید بین ماشین درایور (ماشین محلی) و ماشین‌های کارگر جابجا شوند، بسته به تاخیر شبکه، ممکن است عملکرد برنامه کاهش یابد و زمان اجرا طولانی‌تر شود.

در حالت cluster، برنامه درایور می‌تواند روی هر یک از ماشین‌های درون cluster اسپارک اجرا شود. درواقع، روی هر یک از گره‌های موجود و در دسترس cluster می‌تواند راه‌اندازی شود. در این حالت، خاتمه یافتن یک جلسه باعث خاتمه یافتن کل برنامه نمی‌شود و برنامه می‌تواند بر روی cluster در حال اجرا بماند. باتوجه به این که در این حالت درایور و ماشین‌های کارگر درون cluster هستند، سرباره جابجایی داده بین cluster و ماشینی که کار از آن ارسال شده‌است کاهش می‌یابد.