به‌نام خداوند بخشنده و مهربان

تمرین سوم: زمان اجرا

استاد: محمدعلی نعمت‌بخش

درس: تحلیل سیستم داده‌های حجیم

دستیاران: فاطمه ابراهیمی، پریسا لطیفی، امیر سرتیپی

نام و نام‌خانوادگی: عطیه نیکبخت

آدرس گیت: **https://github.com/AtiyehNikbakht/mapreduce_spark_hadoop.git**

- **لطفا پاسخ تمارین 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 HW-{homework number}-{Name Family}-{student number}**
- **تمامی فایل‌های مورد نیاز این تمرین در این لینک قابل دسترس 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.

زمان اجرا در هدوپ:

در ابتدا فایل mapper را با استفاده از دستورات زیر ایجاد می‌کنیم :

atiyeh_nikbakht@MasterPC: ~

```
atiyeh_nikbakht@MasterPC:~$ nano mapper.py
```

atiyeh_nikbakht@MasterPC: ~

```
  GNU nano 4.8
#!/usr/bin/python
import sys

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

برای ساخت reducer هم از دستورات زیر استفاده می‌کنیم:

```
atiyeh_nikbakht@MasterPC:~$ nano reducer.py
```

atiyeh_nikbakht@MasterPC: ~

```
  GNU nano 4.8
#!/usr/bin/python

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))
```

برای اجرای عملیات map-reduce از دستور زیر استفاده می‌کنیم، هم‌چنین با اضافه کردن time به ابتدای دستور می‌توان مدت زمانی که برای اجرای هر یک از فایل‌ها صرف شده را مشاهده کرد:

فایل ۱ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time hadoop jar /home/hduser/hadoop/hadoop/share/had
oop/tools/lib/hadoop-streaming-2.10.1.jar -mapper mapper.py -reducer reducer.py
-input /user/ebrahimi/hw3-data/file1G.txt -output /home/atiyeh_nikbakht/out1G
```

فایل ۵ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time hadoop jar /home/hduser/hadoop/hadoop/share/had
oop/tools/lib/hadoop-streaming-2.10.1.jar -mapper mapper.py -reducer reducer.py
-input /user/ebrahimi/hw3-data/file5G.txt -output /home/atiyeh_nikbakht/out5G
```

فایل ۱۰ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time hadoop jar /home/hduser/hadoop/hadoop/share/had
oop/tools/lib/hadoop-streaming-2.10.1.jar -mapper mapper.py -reducer reducer.py
-input /user/ebrahimi/hw3-data/file10G.txt -output /home/atiyeh_nikbakht/out10G
```

فایل ۱۲ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time hadoop jar /home/hduser/hadoop/hadoop/share/had
oop/tools/lib/hadoop-streaming-2.10.1.jar -mapper mapper.py -reducer reducer.py
-input /user/ebrahimi/hw3-data/file12G.txt -output /home/atiyeh_nikbakht/out12G
```

زمان اجرا در اسپارک:

با استفاده از دستور زیر فایل کد برای اسپارک ایجاد می‌شود:

atiyeh_nikbakht@MasterPC: ~

```
atiyeh_nikbakht@MasterPC:~$ nano wcS.py
```

atiyeh_nikbakht@MasterPC: ~

```
  GNU nano 4.8
import pyspark
from pyspark.sql import SparkSession
import time

start = time.time()
spark = SparkSession.builder.appName('SparkPractice').getOrCreate()
sc = spark.sparkContext

#Converting Text to RDD and doing map reduce function
textRdd = sc.textFile("/user/ebrahimi/hw3-data/file1G.txt")
textRddM = textRdd.flatMap(lambda x: x.split(' '))
textRddM = textRddM.map(lambda x: (x,1))
textRddM = textRddM.reduceByKey(lambda x,y: x+y)
textRddM = textRddM.collect()
end = time.time()
final = end - start

print("Time to running file1G.txt is:", final)
spark.stop()
```

کد فوق فایل داده با حجم ۱ گیگ را به‌عنوان ورودی می‌گیرد. برای فایل‌های داده دیگر به همین روش عمل می‌کنیم و مسیر مناسب را در تابع textFile() قرار می‌دهیم.

برای مشاهده زمان اجرای این فایل از دستور time.time() استفاده شده‌است. هم‌چنین می‌توان مانند آنچه پیش‌تر بیان شد در هنگام اجرای فایل اسپارک نیز از time استفاده کرد. در اینجا از هر دو روش استفاده شده است.

برای اجرای اسپارک به صورت cluster mode از دستور زیر استفاده می‌کنیم:

```
atiyeh_nikbakht@MasterPC:~$ spark-submit --deploy-mode cluster wcS.py
```

زمان اجرای فایل ۱ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time spark-submit --deploy-mode cluster  wcS.py

real    1m32.025s
user    0m8.062s
sys     0m0.855s
```

زمان اجرای فایل ۵ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time spark-submit --deploy-mode cluster  wcS.py

real    6m31.354s
user    0m10.478s
sys     0m1.477s
```

زمان اجرای فایل ۱۰ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time spark-submit --deploy-mode cluster  wcS.py

real    12m22.518s
user    0m11.598s
sys     0m1.640s
```

زمان اجرای فایل ۱۲ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time spark-submit --deploy-mode cluster  wcS.py

real    14m37.595s
user    0m12.145s
sys     0m1.649s
```

و برای اجرای اسپارک به صورت client mode می‌توان از دستور زیر استفاده کرد:

```
atiyeh_nikbakht@MasterPC:~$ spark-submit --deploy-mode client wcS.py
```

در خط اول زمان اجرای محاسبه شده توسط دستور time.time() است و خط دوم زمان اجرای به‌دست آمده از time است.

زمان اجرای فایل ۱ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time spark-submit --master yarn  --deploy-mode client wcS.py
Time to running file1G.txt is: 101.71702861785889

real    1m43.783s
user    0m20.617s
sys     0m1.952s
```

زمان اجرای فایل ۵ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time spark-submit --master yarn  --deploy-mode client wcS.py
Time to running file5G.txt is: 389.4268465042114

real    6m32.238s
user    0m24.522s
sys     0m2.402s
```

زمان اجرای فایل ۱۰ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time spark-submit --master yarn  --deploy-mode client wcS.py
Time to running file10G.txt is: 749.7422347068787

real    12m32.347s
user    0m31.324s
sys     0m3.747s
```

زمان اجرای فایل ۱۲ گیگ:

```
atiyeh_nikbakht@MasterPC:~$ time spark-submit --master yarn  --deploy-mode client wcS.py
Time to running file12G.txt is: 861.4820024967194

real    14m24.257s
user    0m32.051s
sys     0m3.373s
```

در صورتی که فایل دارای دستوری مانند print باشد، اگر اسپارک به‌صورت client mode اجرا شود می‌توان خروجی دستور print (و یا دستورات دیگر) را مشاهده کرد اما در صورت اجرای اسپارک به‌صورت cluster mode خروجی دستورات نمایش داده نمی‌شود. حالت client بیشتر برای تعامل و اشکال زدایی است.