

استاد: محمدعلی نعمتبخش

تمرین سوم: زمان اجرا

دستياران: فاطمه ابراهيمي، پريسا لطيفي، امير سرتيپي

درس: تحلیل سیستم دادههای حجیم

نام و نامخانوادگی: عطیه نیکبخت

https://github.com/AtiyehNikbakht/mapreduce_spark_hadoop.git

- لطفا پاسخ تمارین حتما در سامانهی کوئرا ارسال شود.
- لطفا یاسخهای خود را در خود سند سوال نوشته و در قالب یک فایل PDF ارسال کنید.
 - نام سند ارسالی {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 برنامه ی شمارش کلمات عملیات نگاشت-کاهش را برای داده ها بر روی هدوپ و اسپارک انجام دهید. نتایج را گزارش و مقایسه ای بین آنها انجام دهید.

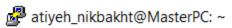
آدرس فایلها:

/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 می کنیم:



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 می کنیم:

```
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)
        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 می کنیم، هم چنین با اضافه کردن mime به ابتدای دستور می توان مدت زمانی که برای اجرای هر یک از فایل ها 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:

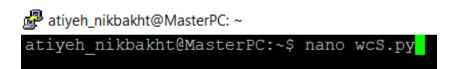
atiyeh_nikbakht@MasterPC:~\$ time hadoop jar /home/hduser/hadoop/hadoop/share/had oop/tools/lib/hadoop-streaming-2.10.1.jar -mapper mapper.py -reducer reducer.py -input /user/ebrahimi/hw3-data/file10G.txt -output /home/atiyeh_nikbakht/out10G

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/ativeh nikbakht/out12G

زمان اجرا در اسیارک:

با استفاده از دستور زیر فایل کد برای اسیارک ایجاد می شود:



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 می گیرد. برای فایلهای داده دیگر به همین روش عمل می کنیم و مسیر مناسب را در تابع ()textFile قرار می دهیم.

برای مشاهده زمان اجرای این فایل از دستور ()time.time استفاده شدهاست. همچنین می توان مانند آنچه پیش تر 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 می کنیم:

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 می توان 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 به صورت client mode اجرا شود می توان خروجی دستور print دستور print (و یا دستورات دیگر) را مشاهده کرد اما در صورت اجرای اسپارک به صورت و cluster mode خروجی دستورات نمایش داده نمی شود. حالت client بیشتر برای تعامل و اشکال زدایی است.