



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: عطیه نیکبخت

آدرس گیت: [https://github.com/AtiyehNikbakht/mapreduce\\_spark\\_hadoop.git](https://github.com/AtiyehNikbakht/mapreduce_spark_hadoop.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 HW-{homework number}-{Name Family}-{student number}
- 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.

زمان اجرا در هدوپ:

در ابتدا یک فایل برای mapper با استفاده از دستورات زیر می‌سازیم:

```
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))
```

همچنین برای reducer هم از دستورات زیر برای ساخت آن 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 mapreducer از دستور زیر استفاده می‌کنیم:

این دستور برای فایل ۱ گیگ است. برای فایل‌های دیگر نیز به همین صورت عمل می‌کنیم.

```
atiyeh_nikbakht@MasterPC:~$ hadoop jar /home/hduser/hadoop/hadoop/share/hadoop/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
packageJobJar: [/tmp/hadoop-unjar964629976360993397/] [] /tmp/streamjob1931231215021880988.jar tmpDir=null
22/04/11 11:50:42 INFO client.RMPProxy: Connecting to ResourceManager at master/172.16.20.4:8032
22/04/11 11:50:42 INFO client.RMPProxy: Connecting to ResourceManager at master/172.16.20.4:8032
22/04/11 11:50:42 INFO mapred.FileInputFormat: Total input files to process : 1
22/04/11 11:50:42 INFO mapreduce.JobSubmitter: number of splits:8
22/04/11 11:50:42 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1649179500517_0699
22/04/11 11:50:42 INFO conf.Configuration: resource-types.xml not found
22/04/11 11:50:42 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
22/04/11 11:50:42 INFO resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
22/04/11 11:50:42 INFO resource.ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
22/04/11 11:50:43 INFO impl.YarnClientImpl: Submitted application application_1649179500517_0699
22/04/11 11:50:43 INFO mapreduce.Job: The url to track the job: http://master:2104/proxy/application_1649179500517_0699/
22/04/11 11:50:43 INFO mapreduce.Job: Running job: job_1649179500517_0699
```


همانطور که در تصویر بالا مشاهده می‌کنید زمان اجرا را می‌توان از ستون مشخص شده مشاهده کرد.

همچنین زمان اجرا را می‌توان با استفاده از دستور زیر مشاهده کرد:

```
atiyeh_nikbakht@MasterPC:~$ time hadoop jar /home/hduser/hadoop/hadoop/share/hadoop/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
```

زمان اجرا در اسپارک:

با استفاده از دستورات زیر فایل کد را برای اسپارک می‌نویسیم:

 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 is:", final)
spark.stop()
```

برای مشاهده زمان اجرای این فایل از دستور `time.time()` استفاده شده است. برای فایل های داده دیگر به همین روش عمل می کنیم و مسیر مناسب را در تابع `textFile()` قرار می دهیم.

برای اجرای اسپارک به صورت `cluster mode` از دستور زیر استفاده می کنیم:

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

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
Time is: 103.60392141342163
```

همانطور که مشاهده می کنید زمان اجرا برای فایل ۱ گیگ ۱۰۳.۶۰ می باشد.