

به نام خدا

تمرین اول

Hadoop

امیرمحمد رنجبر پازکی
۸۱۰۱۹۹۳۴۰

کلان داده و تحلیل داده‌های حجمی
دکتر اسد پور

دانشکده برق و کامپیوتر دانشگاه تهران
بهار ۱۴۰۰

• گام اول - راهاندازی Hadoop Clsuter

برای راهاندازی کلaster هدوب از داکر استفاده شده است.

برای این کار ابتدا داکر و داکر کامپوز برای سیستم مورد نظر نصب شد. سپس، به پوشه docker در پوشه داده شده رفته و با استفاده از دستور زیر کلaster هدوب بالا می آید.

Docker-compose up

در فایل داکر کامپوز گره های مورد نیاز به صورت **service** تعریف شده اند که با اجرای دستور بالا این گره ها بالا می آیند. دو گره مربوط به بخش ذخیره سازی Hadoop هستند (namenode, resourceManager). سه گره مربوط به بخش مدیریتی Hadoop می شوند (datanode, historyserver, nodemanager). همچنین بخش پردازشی با استفاده از framework hive صورت می گیرد که سه گره نیز مربوط به آن می شود. پس از اجرای موفقیت آمیز دستورات بالا، با اجرای دستور زیر می توان پردازه های روی داکر را مشاهده کرد.

Docker ps

(base) iamiranbar@Amirs-MBP docker % docker ps	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
	PORTS	NAMES			
a21fdb19aea6	bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8 0.0.0.0:9820->9820/tcp, 0.0.0.0:9870->9870/tcp	namenode	"/entrypoint.sh /run..."	56 seconds ago	Up 36 seconds (healthy)
32e82d275dba	bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8 0.0.0.0:8042->8042/tcp	nodemanager	"/entrypoint.sh /run..."	56 seconds ago	Up 47 seconds (health: startin
ac802efb152f	bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8 0.0.0.0:8188->8188/tcp, 0.0.0.0:10020->10020/tcp	historyserver	"/entrypoint.sh /run..."	56 seconds ago	Up 40 seconds (healthy)
b7fb855542b9	bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8 0.0.0.0:9864->9864/tcp	datanode	"/entrypoint.sh /run..."	56 seconds ago	Up 45 seconds (healthy)
6b05d0500205	bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8 0.0.0.0:8030->8030-8032/tcp, 0.0.0.0:8088->8088/tcp	resourcemanager	"/entrypoint.sh /run..."	56 seconds ago	Up 38 seconds (healthy)
9d8f556a1b6f	bde2020/hive:2.3.2-postgresql-metastore 10000/tcp, 0.0.0.0:9083->9083/tcp, 10002/tcp	hive-metastore	"entrypoint.sh /opt/..."	56 seconds ago	Up 44 seconds
4ffb79bfe7fc	bde2020/hive-metastore-postgresql:2.3.0 0.0.0.0:5432->5432/tcp	hive-metastore-postgresql	"/docker-entrypoint..."	56 seconds ago	Up 50 seconds
5db1744b7b3c	bde2020/hive:2.3.2-postgresql-metastore 0.0.0.0:10000-10002->10000-10002/tcp	hive-server	"entrypoint.sh /bin/..."	56 seconds ago	Up 35 seconds

همانطور که در خروجی دستور قبل مشاهده می شود، کلaster با موفقیت بالا آمده است. برای چک کردن بالابودن سرویس ها می توان آدرس آن ها را در مرورگر وارد کرد و بالا بودن آن ها را دید. این کار نیز برای تمام سرویس ها انجام شد تا از بالا بودن آن ها اطمینان حاصل شود.

• گام دوم - کار با HDFS + بخش اول: دانلود فایل های متنی

برای این کار می خواهیم با استفاده از shell script متن بیست کتاب را در فرمت txt دریافت کنیم. برای این منظور از قسمت کتاب های انتشارات گوتنبرگ، آدرس txt بیست کتاب را پیدا کرده و در فایل books.txt قرار می دهیم.

سپس، با استفاده از دستور زیر آدرس hdfs /data/books را در ایجاد می کنیم.

Hadoop fs -mkdir -p /data/books

سپس، با استفاده از دستور زیر به داخل **namenode** می‌رویم و با استفاده از دستور **curl** کتاب **frankenstien** را دانلود می‌کنیم.

Docker exec -it namenode bash

سپس، با استفاده از دستور زیر کتاب را به پوشش فوق منقل می‌کنیم.

Hadoop fs -put Frankenstein.txt /data/books

همانطور که می‌بینید، کتاب مورد نظر در مسیر مورد نظر قرار گرفت.

Browse Directory

/data/books									Go!			
Show 25 entries									Search:			
	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name				
<input type="checkbox"/>	-rw-r--r--	root	supergroup	438.3 KB	Apr 15 14:24	1	128 MB	frenkstein.txt				
Showing 1 to 1 of 1 entries												
Previous 1 Next												

Hadoop, 2019.

این کار با دستور زیر نیز می‌توان انجام داد.

Hadoop fs -ls /data/books

```
[root@namenode:/# hadoop fs -ls /data/books
Found 1 items
-rw-r--r-- 1 root supergroup 448821 2021-04-15 09:54 /data/books/frenkstein.txt
```

حال یک **shell script** نوشته شد تا از فایل **books.txt** تک به تک آدرس کتاب‌ها را بخواند، آن‌ها را دانلود کند و به **hadoop** منتقل کند. لازم به ذکر است این **script** با استفاده از دستور زیر قابل اجرا می‌شود.

Chmod +x get_books.sh

این اسکریپت فایل **get_books.sh** است که در کنار گزارش بارگذاری شده است. لازم به ذکر است در فایل **books.txt** نام هر کتاب نیز با یک ، در کنار آدرس قرار گرفت تا فایل‌های دانلود شده هر کدام به اسم کتاب انتقال یابند. همان‌طور که در تصاویر زیر مشاهده می‌کنید تمامی کتاب‌ها با نام خودشان دانلود و به پوشش مورد نظر منتقل شدند.

```
Heart_of_Darkness.txt downloaded.
2021-04-15 10:45:06,853 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Heart_of_Darkness.txt transferred to hadoop successfully.
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload  Upload Total Spent   Left Speed
100  159k  100  159k    0     0  190k      0  --:--:-- --:--:-- --:--:-- 190k
The_Strange_Case_Of_Dr_Jekyll_And_Mr_Hyde.txt downloaded.
2021-04-15 10:45:09,639 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
The_Strange_Case_Of_Dr_Jekyll_And_Mr_Hyde.txt transferred to hadoop successfully.
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload  Upload Total Spent   Left Speed
100  860k  100  860k    0     0  678k      0  0:00:01  0:00:01  --:--:-- 678k
Dracula.txt downloaded.
2021-04-15 10:45:12,910 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Dracula.txt transferred to hadoop successfully.
Done!
```

Browse Directory

/data/books									Go!			
Show 25 entries									Search:			
	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name				
	-rw-r--r--	root	supergroup	38.89 KB	Apr 15 15:14	1	128 MB	A_Modest_Proposal.txt				
	-rw-r--r--	root	supergroup	596.06 KB	Apr 15 15:14	1	128 MB	Adventures_of_Huckleberry_Finn.txt				
	-rw-r--r--	root	supergroup	860.81 KB	Apr 15 15:15	1	128 MB	Dracula.txt				
	-rw-r--r--	root	supergroup	438.3 KB	Apr 15 15:14	1	128 MB	Frankenstein.txt				
	-rw-r--r--	root	supergroup	231.51 KB	Apr 15 15:15	1	128 MB	Heart_of_Darkness.txt				
	-rw-r--r--	root	supergroup	1.03 MB	Apr 15 15:15	1	128 MB	Jane_Eyre.txt				
	-rw-r--r--	root	supergroup	138.69 KB	Apr 15 15:14	1	128 MB	Metamorphosis.txt				
	-rw-r--r--	root	supergroup	1.22 MB	Apr 15 15:14	1	128 MB	Moby_Dick.txt				
	-rw-r--r--	root	supergroup	780.9 KB	Apr 15 15:14	1	128 MB	Pride_and_Prejudice.txt				
	-rw-r--r--	root	supergroup	593.55 KB	Apr 15 15:14	1	128 MB	Sherlock_Holmes.txt				
	-rw-r--r--	root	supergroup	299.08 KB	Apr 15 15:14	1	128 MB	The_Great_Gatsby.txt				
	-rw-r--r--	root	supergroup	138.41 KB	Apr 15 15:14	1	128 MB	The_Importance_of_Being_Earnest.txt				

+ بخش دوم: دانلود و ذخیره فایل‌های خرید و فروش روزانه بورس

برای این منظور ابتدا پوشه‌ی `hdfs /data/data_lake/stock` با استفاده از دستوری مشابه قبل ساخته شد. یک کد پایتون نوشته شد که از آدرس موردنظر داده‌های شش ماه اخیر را می‌گرفت.

این کد `get_daily_stock.py` است که در کنار گزارش بارگذاری می‌شود.

پس از دانلود این فایل‌های اکسل حجم آن‌ها نیز بررسی شد و فایل‌های زیر ۱۰ کیلوبایت حذف شدند تا با استفاده از رهیافت پنکه برای بسته‌های خالی روزهای تعطیل بور که داده‌ای ندارند در

میان داده‌ها نباشند. سپس، این فایل‌های اکسل که در پوشه‌ی `temp` قرار گرفته‌بودند، با استفاده از کتابخانه `pandas` به `CSV` تبدیل شدند و همچنین، به هر کدام یک ستون برای تاریخ معاملات نیز اضافه شد تا در ادامه استفاده شود.

سپس، این فایل‌های نهایی با استفاده از یک حلقه‌ی `for` ساده در `bash` به پوشه‌ی روی `HDFS` منتقل شدند که نتایج در زیر قابل مشاهده است.

```
root@namenode:/data/stocks/results# for file in *; do hadoop fs -put $file /data/data_lake/stock/; echo $file; done
2021-04-15 13:56:53,218 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_1.csv
2021-04-15 13:56:56,883 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_10.csv
2021-04-15 13:57:00,512 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_11.csv
2021-04-15 13:57:04,325 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_14.csv
2021-04-15 13:57:08,273 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_15.csv
2021-04-15 13:57:11,812 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_16.csv
2021-04-15 13:57:16,201 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_17.csv
2021-04-15 13:57:19,874 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_18.csv
2021-04-15 13:57:23,504 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_2.csv
2021-04-15 13:57:27,563 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_21.csv
2021-04-15 13:57:31,522 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
stock_data_1399_10_22.csv
```

Browsing HDFS +

localhost:9870/explorer.html#/data/data_lake/stock

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

Browse Directory

/data/data_lake/stock

Show 25 entries Search:

<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	121.52 KB	Apr 15 18:26	1	128 MB	stock_data_1399_10_1.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	117.55 KB	Apr 15 18:26	1	128 MB	stock_data_1399_10_10.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	118.41 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_11.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	116.31 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_14.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	120.26 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_15.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	118.04 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_16.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	118.93 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_17.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	121.14 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_18.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	118.8 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_2.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	117.26 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_21.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	118.35 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_22.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	120.83 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_23.csv	<input type="checkbox"/>
<input type="checkbox"/>	-rw-r--r--	root	supergroup	118.34 KB	Apr 15 18:27	1	128 MB	stock_data_1399_10_24.csv	<input type="checkbox"/>

• گام سوم - کار با فایل‌های متنی (Word Count)

مرحله اول: کد این مرحله همان wordcount.py است که بر روی یکی از فایل‌های دریافتی اجرا شده و خروجی زیر را داده است.

```
...top/Master/Term 2/BD/HW/#1HW1-Resources/docker — docker-compose up ...
...W1-Resources/docker — com.docker.cli • docker exec -it namenode bash
...Desktop/Master/Term 2/BD/HW/#1HW1-Resources/docker/data — zsh +
```

```
root@namenode:/data# python3 wordcount.py books/Frankenstein.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory /tmp/wordcount.root.20210415.142037.845029
Running step 1 of 1...
job output is in /tmp/wordcount.root.20210415.142037.845029/output
Streaming final output from /tmp/wordcount.root.20210415.142037.845029/output...
"know," 4
"know,\u201d" 2
"know." 1
"know;" 3
"knowing" 4
"knowledge!" 1
"knowledge" 22
"knowledge," 2
"knowledge." 7
"knowledge.\u201d" 1
"known" 6
"known," 1
"known." 3
"knows" 5
"knows,\u201d" 1
"laboratory" 2
"laboratory." 4
"laboratory;" 1
"laborious" 3
"labour" 6
"labour," 5
"labour." 1
"labour;" 1
"labour\u2014but" 1
"labour\u2014the" 1
"labourers" 2
"labours" 6
"labours," 3
"labours." 5
"labours;" 3
"ladies" 1
"lady" 5
"lady," 6
"laid" 1
"lake!" 2
"lake" 12
"lake," 7
"lake." 3
```

مرحله دوم: این پیشپردازش‌ها در تابع `preprocess_line` آمده است که هر خط را پیشپردازش می‌کند. همه کاراکترها کوچک می‌شوند و کاراکترهای بد از متن حذف می‌شوند. خروجی نیز در ادامه آمده است.

```
def preprocess_line(line):
    line = line.lower()
    return re.sub('\w+', ' ', line)
```

```
"applies"      1
"approximately" 1
"apr"          1
"are"          5
"around"        1
"as"            1
"at"            1
"auto"          1
"automated"     1
"automatically" 1
"available"     1
"be"            2
"because"       3
"been"          1
```

مرحله سوم: در این مرحله تعداد ایستوازه در فایل `stopwords.txt` قرار گرفته است که با استفاده از `pyhdfs` خوانده شده و وجود یا عدم وجود آن در تابع `mapper` چک می‌شود.

```
def read_stop_words():
    fs = pyhdfs.HdfsClient(hosts='namenode:9870')
    stopwords_file_address = "/data/stopwords.txt"
    with fs.open(stopwords_file_address) as stopwords_file:
        for line in stopwords_file:
            line = line.decode('utf-8')
            stopwords.add(line.rstrip())

    def mapper(self, _, line):
        cleaned_line = preprocess_line([line])
        for word in cleaned_line.split():
            if word not in stopwords:
                yield(word, 1)
```

خروجی این مرحله به صورت زیر است که **stopword** ها از آن حذف شده‌اند.

```
~/Desktop/Master/Term 2/BD/HW/#1/HW1-Resources/docker --zsh ... ...W1-Resources/docker — com.docker.cli • docker exec -it namenode bash ~/Desktop/Master/Term 2/BD/HW/#1/HW1-Resources/docker --zsh  
"shown" 1  
"shows" 2  
"shtml" 2  
"single" 1  
"site" 2  
"sl" 1  
"software" 1  
"sometimes" 1  
"spaces" 1  
"status" 1  
"text" 1  
"thursday" 1
```

مرحله چهارم: با افزودن یک **reducer** و تعریفتابع **steps** حذف ایستوازه‌ها یک مرحله جدا می‌شود.

```
class Count(MRJob):  
    def mapper_get_words(self, _, line):  
        cleaned_line = preprocess_line(line)  
        for word in cleaned_line.split():  
            yield(word, 1)  
  
    def reducer_stop_words_removal(self, word, counts):  
        if word not in stopwords:  
            yield(word, sum(counts))  
            # yield(None, (sum(counts), word))  
  
    def reducer_sort_by_count(self, key, values):  
        sorted_values = sorted(list(values), reverse=True)  
        return sorted_values[:10]  
  
    def steps(self):  
        return [  
            MRStep(mapper=self.mapper_get_words, reducer=self.reducer_stop_words_removal),  
            # MRStep(reducer=self.reducer_sort_by_count)  
        ]
```

```
"block" 4  
"blocked" 8  
"blocking" 1  
"blocks" 2  
"body" 2  
"br" 4  
"browser" 2  
"browsers" 1  
"cache" 1  
"cand" 2  
"case" 1  
"center" 1  
"certain" 1  
"changes" 1
```

```

Running step 1 of 2...
packageJobJar: [/tmp/hadoop-unjar5275951680447055977/] [] /tmp/streamjob4324237507261131775.jar tmpDir=null
Connecting to ResourceManager at resourcemanager/172.18.0.6:8032
Connecting to Application History server at historyserver/172.18.0.4:10200
Connecting to ResourceManager at resourcemanager/172.18.0.6:8032
Connecting to Application History server at historyserver/172.18.0.4:10200
Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1618560161341_0002
SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Total input files to process : 1
SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
number of splits:2
SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
Submitting tokens for job: job_1618560161341_0002
Executing with tokens: []
resource-types.xml not found
Unable to find 'resource-types.xml'.
Submitted application application_1618560161341_0002
The url to track the job: http://resourcemanager:8088/proxy/application_1618560161341_0002/
Running job: job_1618560161341_0002

```

مرحله پنجم: با استفاده از الگوريتم sort کتاب ده کلمه پر تكرار کتاب مورد نظر بهدست آمد.

```

class Count(MRJob):
    def mapper_get_words(self, _, line):
        cleaned_line = preprocess_line(line)
        for word in cleaned_line.split():
            yield(word, 1)

    def reducer_stop_words_removal(self, word, counts):
        if word not in stopwords:
            yield(None, (sum(counts), word))

    def reducer_sort_by_count(self, key, values):
        sorted_values = sorted(list(values), reverse=True)
        return sorted_values[:10]

    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_words, reducer=self.reducer_stop_words_removal),
            MRStep(reducer=self.reducer_sort_by_count)
        ]

```

208	"one"
198	"could"
184	"would"
152	"yet"
137	"man"
133	"father"
128	"upon"
116	"life"
114	"may"
109	"every"

مرحله ششم: با استفاده از دستور زیر کتاب به صورت موردنظر روی hdfs قرار گرفت.

```
root@namenode:/data# python3 wordcount.py -r hadoop hdfs:///data/books/Frankenstein.txt > Frankenstein.txt && hadoop fs -put Frankenstein.txt /data/books/top10
0
No configs found; falling back on auto-configuration
No configs specified for hadoop runner
Looking for hadoop binary in /opt/hadoop-3.2.1/bin...
Found hadoop binary: /opt/hadoop-3.2.1/bin/hadoop
Using Hadoop version 3.2.1
Looking for Hadoop streaming jar in /opt/hadoop-3.2.1...
Found Hadoop streaming jar: /opt/hadoop-3.2.1/share/hadoop/tools/lib/hadoop-streaming-3.2.1.jar
Creating temp directory /tmp/wordcount.root.20210416.121221.652425
uploading working dir files to hdfs:///user/root/tmp/mrjob/wordcount.root.20210416.121221.652425/files/wd...
Copying other local files to hdfs:///user/root/tmp/mrjob/wordcount.root.20210416.121221.652425/files/
```

Browse Directory

/data/books/top10								Go!									
Show	25	entries							Search:								
<input type="checkbox"/>		Permission		Owner		Group		Size		Last Modified		Replication		Block Size		Name	
<input type="checkbox"/>	-rw-r--r--		root	supergroup		100 B		Apr 16 16:44		1		1		128 MB		Frankenstein.txt	

Showing 1 to 1 of 1 entries

Previous 1 Next

Hadoop, 2019.

در آخرین مرحله، با استفاده از دستور زیر این کار بر روی تمام کتابها انجام و نتایج بر روی مسیر مورد نظر قرار گرفت.

```
root@namenode:/data# for filename in `hadoop fs -ls /data/books/ | awk '{print $NF}'` | grep .txt$; do file_name=$(echo $filename | cut -f4 -d/); python3 wordcount.py -r hadoop hdfs:///data/books/$file_name > $file_name; hadoop fs -put $file_name /data/books/top10; done
```

```
~/Desktop/Master/Term 2/BD/HW#1/HW1-Resources/docker --docker-compose up ... ~Desktop/Master/Term 2/BD/HW#1/HW1-Resources/docker --com.docker.cli - docker exec -it namenode bash +
```

```
root@namenode:/data# hadoop fs -ls /data/books/top10
Found 20 items
-rw-r--r-- 1 root supergroup 123 2021-04-16 13:42 /data/books/top10/A_Christmas_Carol.txt
-rw-r--r-- 1 root supergroup 115 2021-04-16 13:44 /data/books/top10/A_Dolls_House.txt
-rw-r--r-- 1 root supergroup 112 2021-04-16 13:45 /data/books/top10/A_Modest_Proposal.txt
-rw-r--r-- 1 root supergroup 112 2021-04-16 13:47 /data/books/top10/A_Tale_of_Two_Cities.txt
-rw-r--r-- 1 root supergroup 104 2021-04-16 13:49 /data/books/top10/Adventures_of_Huckleberry_Finn.txt
-rw-r--r-- 1 root supergroup 115 2021-04-16 13:50 /data/books/top10/Alices_Adventures_in_Wonderland.txt
-rw-r--r-- 1 root supergroup 108 2021-04-16 13:52 /data/books/top10/Dracula.txt
-rw-r--r-- 1 root supergroup 111 2021-04-16 13:54 /data/books/top10/Frankenstein.txt
-rw-r--r-- 1 root supergroup 112 2021-04-16 13:55 /data/books/top10/Heart_of_Darkness.txt
-rw-r--r-- 1 root supergroup 116 2021-04-16 13:57 /data/books/top10/Jane_Eyre.txt
-rw-r--r-- 1 root supergroup 125 2021-04-16 13:59 /data/books/top10/Metamorphosis.txt
-rw-r--r-- 1 root supergroup 106 2021-04-16 14:01 /data/books/top10/Moby_Dick.txt
-rw-r--r-- 1 root supergroup 117 2021-04-16 14:02 /data/books/top10/Pride_and_Prejudice.txt
-rw-r--r-- 1 root supergroup 111 2021-04-16 14:04 /data/books/top10/Sherlock_Holmes.txt
-rw-r--r-- 1 root supergroup 112 2021-04-16 14:06 /data/books/top10/The_Great_Gatsby.txt
-rw-r--r-- 1 root supergroup 131 2021-04-16 14:08 /data/books/top10/The_Importance_of_Being_Earnest.txt
-rw-r--r-- 1 root supergroup 112 2021-04-16 14:09 /data/books/top10/The_Picture_of_Dorian_Gray.txt
-rw-r--r-- 1 root supergroup 118 2021-04-16 14:11 /data/books/top10/The_Scarlet_Letter.txt
-rw-r--r-- 1 root supergroup 117 2021-04-16 14:13 /data/books/top10/The_Strange_Case_of_Dr_Jekyll_And_Mr_Hyde.txt
-rw-r--r-- 1 root supergroup 104 2021-04-16 14:15 /data/books/top10/The_Yellow_Wallpaper.txt
```

خروجی به همراه کد نهایی در کنار گزارش بارگذاری شده است.

۰ گام چهارم - پردازش داده های بورس

ابتدا با استفاده از `preprocess.py` پیش پردازش گفته شد برای پالایش اسم نمادها انجام شد و با استفاده از یک `for` ساده در `bash` برای تمام فایل ها این کار انجام شد و تمامی فایل ها روی `hdfs` دوباره قرارداده شدند.

فایل `preprocess.py` در کنار گزارش بارگذاری شده است.
خروجی در زیر آمده است. لازم به ذکر است فایل های گفته شده برای هر خواسته در کنار گزارش بارگذاری شده اند.

خواسته ۱ :
گرانترین:

فایل کد: `q1_1.py`

خروجی:

تعداد ۴ :container

سحرخیز	9996
زرین	9927
وپخش	99250
صلایند	9877
سکرد	9790
ثور	9786
لوتوس	9720
ثازن	9703
شتران	9700
وساشرقی	970

ارزانترین:

فایل کد: `q1_2.py`

خروجی:

تعداد ۴ :container

کیان2	1
عیار2	1
یاقوت2	1
اطلس2	1
کمند2	1
افران2	1
گوهر2	1
نهال2	1
غشانز	10000
ثمارس	10050

خواسته ۲:

فایل کد: q2.py

خروجی:

۱۲۲ :container تعداد

خودرو ۴	22973995436.0
خودرو ۴	15042462245.0
وامین ۴	9929005292.0
شتران ۴	5271093632.0
وتجارت	4011828005.0
شپنا ۴	3962409928.0
خسپا	3768781004.0
خودرو	3695050259.0
شپنا ۴	3635097999.0
فارس ۴	3592814371.0

خواسته ۳:

فایل کد: q3.py

خروجی:

۱۲۴ :container تعداد

غالبر	9	7335.34
بوعلی	9	4067.9399999999996
نتوزیع	9	2594.14
چخر	9	2407.1
02بـ0008 پست	9	1614.4299999999998
04بـ0008 پست	9	1609.31
وکیهن	9	1363.74
09بـ0008 زعف	9	1229.029999999995
غمینوح	9	1156.8300000000002
08بـ0008 زعف	9	1136.44
جوین	8	26787.379999999997
کاندر	8	7676.650000000001
کهمداح	8	1071.59
05بـ0008 زعف	8	1059.99
10بـ0008 زعف	8	1059.9
غمهراح	8	861.48
سیح	8	828.24
سازیریح	8	572.5400000000001
تنوینح	8	357.9900000000007
رتلیح	8	227.03
وسیزد	7	27.0
صیابان	7	26.67
پا ه	7	25.71
پا ه	7	15.79
شاروم	7	15.0
ختور	7	14.040000000000001
کترام	7	10.01
پا ه	7	10.0
ختورح	7	9.99

خواسته ۴:

فایل کد: q4.py

خروجی:

١٢١ :container تعداد

کیان2	-11400.0
اطلس2	-10200.0
عيار2	-10100.0
پاقوت2	-9599.039999999985
آکردا2	-9400.0
كمند2	-8799.119999999986
افران2	-8499.149999999987
امین يكم2	-6899.3099999999895
آگاس2	-6200.0
سردو2	-5800.0

خواسته ۵:

فایل کد: q5.py

خروجی:

١٢٢ :container تعداد

لازم به ذکر است که اینجا چندین نماد
خرنگی داده شده است چرا که بسیاری
از این نمادها موقتاً تستند و با نگاه
کردن با این لیست می‌توان دید که
اولین نماد حقیقی که بسته بوده است
پایابان است که یک روز یا زیبوده است.

گکوثر	2	1
چکارن	4	1
چدن	2	1
چسبیا	4	1
پیزد	2	1
پلاسک	2	1
پکور	2	1
پکرمان	4	1
پکرمان	2	1
پرفروردی		1
پدرخشن	4	1
پترول	2	1
پتایر	2	1
پاکشو	4	1
پاسا	4	1
پاسا	2	1
پارسیانح	2	1
پارسان	4	1
پارس	4	1
پالبان		1
پا	9	1
پا	8	1
پا	7	1
پا	6	1
پا	2	1
پا	10	1
وگستر	4	1
ویویا	2	1
ویست	2	1

همانطور که مشاهده می‌کنید خواسته سوم منابع بیشتری به خود اختصاص داده است چرا که سنگین‌تر است.

۰ گام پنجم - پردازش خواسته‌ها با هایو

پس از نصب DBeaver و ساخت جدول، خواسته‌ها با استفاده از کوئری‌هایی اجرا شدند. کوئری‌ها به همراه خروجی و زمان گرفته شده در زیر به ترتیب آمده‌اند. لازم به ذکر است به دلیل ناسازگاری فرمت تاریخ با استفاده از دستور زیر مجبور به ساخت یک موقت برای اصلاح این مورد شدم.

```
CREATE EXTERNAL TABLE IF NOT EXISTS Stock_Exchange_Daily(
symbol STRING,
full_name STRING,
quantity BIGINT,
volume BIGINT,
value BIGINT,
yesterday_qnt BIGINT,
first_order_value INT,
last_order_value INT,
last_order_value_change FLOAT,
last_order_value_change_percent FLOAT,
close_price INT,
close_price_change FLOAT,
close_price_change_percent FLOAT,
min_price INT,
max_price INT,
symbol_date DATE
)
STORED AS TEXTFILE LOCATION '/data/main_table';
```

```
INSERT OVERWRITE TABLE default.stock_exchange_daily
Select symbol, full_name, quantity, volume, value, yesterday_qnt, first_order_value, last_order_value, last_order_value_change,
last_order_value_change_percent, close_price, close_price_change, close_price_change_percent, min_price, max_price,
from_unixtime(unix_timestamp(symbol_date , 'yyyy/MM/dd'), 'yyyy-MM-dd') FROM default.stock_exchange_daily_tmp ;
```

خواسته ۱:

گرانترین:

زمان: ۳۲.۴ ثانیه

کوئری در کنار خروجی:

```
SELECT symbol, close_price
FROM default.stock_exchange_daily
WHERE symbol_date == "1400-01-26"
ORDER BY close_price DESC
LIMIT 10;
```

symbol	close_price
پست	1,668,809
پست	1,534,235
پست	1,513,048
پست	1,401,001
اروند	947,200
زیر	529,500
زیر	526,000
زیر	522,267
فینتا	408,780
غدام	394,670

ارزانترین:
زمان: ۳۵.۲ ثانیه
کوئری در کنار خروجی:

```
SELECT symbol, close_price
FROM default.stock_exchange_daily
WHERE symbol_date == "1400-01-26"
ORDER BY close_price ASC
LIMIT 10;
```

A screenshot of a database query results grid. The grid has two columns: 'symbol' and 'close_price'. The data shows 10 rows of results, with the 9th row being highlighted. The 9th row contains the symbol 'صیارادیب' with a value of 501. The 10th row contains the symbol 'وسپیستا' with a value of 760. The grid includes standard SQL navigation buttons at the bottom.

	symbol	close_price
1	افران	1
2	کمند	1
3	اطلس	1
4	یاقوت	1
5	نهال	1
6	گهر	1
7	عيار	1
8	کیان	1
9	صیارادیب	501
10	وسپیستا	760

خواسته ۲:
زمان: ۳۲.۹ ثانیه
کوئری در کنار خروجی:

```
SELECT symbol, volume
FROM default.stock_exchange_daily
ORDER BY volume DESC
LIMIT 1;
```

A screenshot of a database query results grid. The grid has two columns: 'symbol' and 'volume'. The single row of data shows the symbol 'فارس' with a volume of 34,152,999,908. The grid includes standard SQL navigation buttons at the bottom.

	symbol	volume
1	فارس	34,152,999,908

خواسته ۳:
زمان: ۷۱ ثانیه
کوئری در کنار خروجی:

```
SELECT *
FROM (
    SELECT symbol, symbol_month, month_change,
    row_number() over (partition by symbol_month order by month_change desc) as symbol_rank
    FROM default.stock_exchange_daily
    GROUP BY symbol, MONTH(symbol_date)
    ) AS symbol_month_change
) AS ranked_symbols
WHERE symbol_rank <= 10;
```

Enter a SQL expression to filter results (use Ctrl+Space)

	symbol	symbol_month	month_change	symbol_rank
1	آرمانی		331,000	1
2	09 پست		270,351	2
3	01 پر3		173,926	3
4	08 پست		161,656	4
5	04 پ0008		150,057	5
6	سنت		106,724	6
7	040003		96,909	7
8	01 پ0003		77,545	8
9	02 پ0008		51,001	9
10	02 پ0003		46,625	10
11	09 پ0008		104,377	1
12	عفولاد		50,310	2
13	اجاد		42,157	3
14	شسینا		38,750	4
15	سلامت		30,000	5

Rows: 1 70 row(s) fetched - 1m 11s (+16ms)

خواسته ۴:
زمان: ۶۴ ثانیه
کوئری در کنار خروجی:

```
SELECT symbol, SUM(close_price_change) as period_change
FROM default.stock_exchange_daily
GROUP BY symbol
ORDER BY period_change ASC
LIMIT 10;
```

	symbol	period_change
1	اطلس	-20,430,101
2	فیروزه	-4,911,790
3	آگاس	-4,199,610
4	سرور	-3,835,742
5	گهر	-3,126,574
6	کیان	-3,043,310
7	عيار	-2,925,066
8	آکورد	-2,832,955
9	طلای	-2,142,873
10	اعتماد	-1,341,460

Rows: 1 10 row(s) fetched - 1m 4s (+18ms)

خواسته ۵:
زمان: ۶۶ ثانیه
کوئری در کنار خروجی:

```
SELECT symbol, COUNT(*) as cnt
FROM default.stock_exchange_daily
GROUP BY symbol
ORDER BY cnt ASC;
```

The screenshot shows a database query results interface. At the top, there is a search bar with the placeholder "Enter a SQL expression to filter results (use Ctrl+Space)". Below the search bar is a table with two columns: "symbol" and "cnt". The table contains 18 rows of data, each representing a stock symbol and its count. The symbols listed are: خکار (4), تابان (01), خسپا (4), خاهن (2), دتولید (2), توربل (4), حریل (4), خاور (4), دالبر (4), خفناور (2), البرز (2), البرز (02), تممسکن (2), اکلا (1), and آندر (1). The counts for all symbols are 1. On the right side of the interface, there is a "Value" input field with the value "برکت" and a "2" button. At the bottom of the interface, there are buttons for "Save", "Cancel", "Script", and various navigation and filtering icons. The status bar at the bottom right indicates "100 row(s) fetched - 1m 6s (+22ms)".

	symbol	cnt
4	خکار	1
5	تابان	1
6	خسپا	1
7	خاهن	1
8	دتولید	1
9	توربل	1
10	حریل	1
11	خاور	1
12	دالبر	1
13	خفناور	1
14	البرز	1
15	البرز	1
16	تممسکن	1
17	اکلا	1
18	آندر	1

در اینجا نیز تعداد بیشتر به همان دلیل قبل خروجی داده شده است.
همانطور که مشاهده کردید در اینجا نیز خواسته سوم زمان بیشتری به خود اختصاص داد.