

Hw3 書面報告

生機所碩一 陳玟鉸

下載 Hadoop 的 dockor image

一開始嘗試在 Windows 上 WSL 的 ubuntu 20.04 上安裝，但是花費了許多時間在解決路徑上設定及防火牆的問題後，下指令都顯示 `hdfs: commend not found`，也嘗試過自己 build dockor hadoop 的 image 但仍然沒有成功，因此最後還是選擇上 EC2 去完成這次作業。

1. 首先下 commend 來下載 image 檔

```
docker pull sequenceiq/hadoop-docker:2.7.0
```

2. 在 docker 上開啟 container

```
docker run -it sequenceiq/hadoop-docker:2.7.0 /etc/bootstrap.sh -bash
```

在 Hadoop 的 container 中創建資料夾並複製檔案

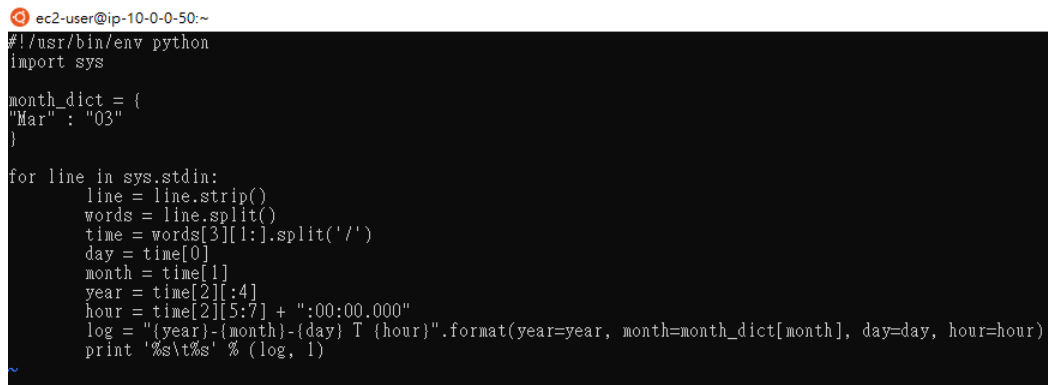
1. `hdfs dfs -mkdir wordcount`

2. `hdfs dfs -copyFromLocal access_log wordcount/access_log.txt`

修改 python 檔

1. mapper.py

修改成將字串分開讀取並存入變數中，並再依照作業要求的格式排列後輸出，因為這次的 log file 中都是 March 的資料，因此直接把月份寫死：“Mar” = “03”。下圖為 mapper.py 的程式。



```
ec2-user@ip-10-0-0-50:~  
#!/usr/bin/env python  
import sys  
  
month_dict = {  
    "Mar" : "03"  
}  
  
for line in sys.stdin:  
    line = line.strip()  
    words = line.split()  
    time = words[3][1:].split('/')  
    day = time[0]  
    month = time[1]  
    year = time[2][:4]  
    hour = time[2][5:7] + ":00:00.000"  
    log = "{year}-{month}-{day} T {hour}".format(year=year, month=month_dict[month], day=day, hour=hour)  
    print '%s\t%s' % (log, 1)
```

此輸出格式範例為：2004-03-07 T 16:00:00.000

2. reducer.py

從 `mapper.py` 中拿到輸出的字串後做計算，依照相同的時間計算該 ip 在同小時內呼叫的次數。下圖為 `reducer.py` 的程式。

```
#!/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)
```

3. 最後根據指令輸入作業給的 access_log 測試這兩個程式。

```
cat access_log | python ./mapper.py | sort | python ./reducer.py
```

下兩圖為輸出結果。

```
bash-4.1# cat access_log | python ./mapper.py | sort | python ./reducer.py
2004-03-07 T 16:00:00.000 27
2004-03-07 T 17:00:00.000 25
2004-03-07 T 18:00:00.000 24
2004-03-07 T 19:00:00.000 26
2004-03-07 T 20:00:00.000 20
2004-03-07 T 21:00:00.000 23
2004-03-07 T 22:00:00.000 29
2004-03-07 T 23:00:00.000 22
2004-03-08 T 00:00:00.000 21
2004-03-08 T 01:00:00.000 21
2004-03-08 T 02:00:00.000 27
2004-03-08 T 03:00:00.000 22
2004-03-08 T 04:00:00.000 26
2004-03-08 T 05:00:00.000 37
2004-03-08 T 06:00:00.000 17
2004-03-08 T 07:00:00.000 31
2004-03-08 T 08:00:00.000 44
2004-03-08 T 09:00:00.000 63
2004-03-08 T 10:00:00.000 39
2004-03-08 T 11:00:00.000 34
2004-03-08 T 12:00:00.000 45
2004-03-08 T 13:00:00.000 37
2004-03-08 T 14:00:00.000 23
2004-03-08 T 15:00:00.000 9
2004-03-08 T 16:00:00.000 2
2004-03-08 T 17:00:00.000 2
2004-03-08 T 18:00:00.000 9
2004-03-08 T 19:00:00.000 6
2004-03-08 T 20:00:00.000 23
2004-03-08 T 21:00:00.000 20
2004-03-08 T 22:00:00.000 1
2004-03-09 T 01:00:00.000 12
2004-03-09 T 02:00:00.000 15
2004-03-09 T 03:00:00.000 1
2004-03-09 T 04:00:00.000 24
2004-03-09 T 05:00:00.000 29
2004-03-09 T 06:00:00.000 8
2004-03-09 T 07:00:00.000 27
2004-03-09 T 08:00:00.000 2
2004-03-09 T 09:00:00.000 11
2004-03-09 T 10:00:00.000 6
2004-03-09 T 11:00:00.000 9
2004-03-09 T 12:00:00.000 8
2004-03-09 T 13:00:00.000 14
2004-03-09 T 14:00:00.000 28
2004-03-09 T 15:00:00.000 2
2004-03-09 T 16:00:00.000 8
2004-03-09 T 17:00:00.000 12
2004-03-09 T 18:00:00.000 3
2004-03-09 T 19:00:00.000 3
2004-03-09 T 20:00:00.000 5
2004-03-09 T 21:00:00.000 1
2004-03-09 T 22:00:00.000 5
2004-03-09 T 23:00:00.000 1
2004-03-10 T 00:00:00.000 6
2004-03-10 T 01:00:00.000 5
2004-03-10 T 02:00:00.000 17
2004-03-10 T 03:00:00.000 1
2004-03-10 T 04:00:00.000 1
2004-03-10 T 05:00:00.000 38
2004-03-10 T 06:00:00.000 3
2004-03-10 T 07:00:00.000 6
2004-03-10 T 08:00:00.000 29
2004-03-10 T 09:00:00.000 29
2004-03-10 T 10:00:00.000 102
2004-03-10 T 11:00:00.000 13
2004-03-10 T 12:00:00.000 3
2004-03-10 T 13:00:00.000 13
2004-03-10 T 14:00:00.000 6
2004-03-10 T 15:00:00.000 2
2004-03-10 T 16:00:00.000 2
2004-03-10 T 17:00:00.000 5
2004-03-10 T 18:00:00.000 12
2004-03-10 T 19:00:00.000 13
2004-03-10 T 20:00:00.000 9
2004-03-10 T 21:00:00.000 6
2004-03-10 T 22:00:00.000 19
2004-03-10 T 23:00:00.000 6
2004-03-11 T 00:00:00.000 1
2004-03-11 T 01:00:00.000 1
2004-03-11 T 02:00:00.000 6
2004-03-11 T 03:00:00.000 19
2004-03-11 T 04:00:00.000 6
2004-03-11 T 05:00:00.000 8
2004-03-11 T 06:00:00.000 3
2004-03-11 T 07:00:00.000 19
2004-03-11 T 08:00:00.000 17
2004-03-11 T 09:00:00.000 46
2004-03-11 T 10:00:00.000 15
2004-03-11 T 11:00:00.000 27
2004-03-11 T 12:00:00.000 6
2004-03-11 T 13:00:00.000 4
2004-03-11 T 14:00:00.000 14
2004-03-11 T 15:00:00.000 1
2004-03-11 T 16:00:00.000 1
2004-03-11 T 17:00:00.000 2
2004-03-11 T 18:00:00.000 1
2004-03-11 T 19:00:00.000 5
2004-03-11 T 20:00:00.000 11
2004-03-11 T 21:00:00.000 1
2004-03-11 T 22:00:00.000 2
2004-03-11 T 23:00:00.000 25
2004-03-12 T 00:00:00.000 22
2004-03-12 T 01:00:00.000 3
2004-03-12 T 02:00:00.000 3
2004-03-12 T 03:00:00.000 5
2004-03-12 T 04:00:00.000 11
2004-03-12 T 05:00:00.000 1
2004-03-12 T 06:00:00.000 2
2004-03-12 T 07:00:00.000 25
2004-03-12 T 08:00:00.000 22
2004-03-12 T 09:00:00.000 3
2004-03-12 T 10:00:00.000 3
2004-03-12 T 11:00:00.000 3
2004-03-12 T 12:00:00.000 3
2004-03-12 T 13:00:00.000 3
bash-4.1#
```

在 Hadoop 上再執行一次

下指令即設定路徑：

```
hadoop jar /usr/local/hadoop-2.7.0/share/hadoop/tools/lib/hadoop-streaming-2.7.0.jar -mapper "python ~/mapper.py" -reducer "python ~/reducer.py" -input "wordcount/access_log" -output "wordcount_outdir"
```

下圖為結果：

```
bash-4.1# hadoop jar /usr/local/hadoop-2.7.0/share/hadoop/tools/lib/hadoop-streaming-2.7.0.jar -mapper "python ~/mapper.py" -reducer
ducer.py" -input "access_log" -output "log_parsed"
packageJobJar: [/tmp/hadoop-unjar7307421402228841831/] [] /tmp/streamjob841662172705805298.jar tmpDir=null
20/11/03 11:38:55 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/11/03 11:38:56 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/11/03 11:38:56 INFO mapred.FileInputFormat: Total input paths to process : 1
20/11/03 11:38:56 INFO mapreduce.JobSubmitter: number of splits:2
20/11/03 11:38:57 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1604387502035_0005
20/11/03 11:38:57 INFO impl.YarnClientImpl: Submitted application application_1604387502035_0005
20/11/03 11:38:57 INFO mapreduce.Job: The url to track the job: http://63ba0d57618a:8088/proxy/application_1604387502035_0005/
20/11/03 11:38:57 INFO mapreduce.Job: Running job: job_1604387502035_0005
```

成功喚起 job_1604387502035_0005 來執行這次的 Hadoop 運算處理。