Cloud Computing and Cyber Security HW3

生機碩一 R09631007 吳乙澤

簡要：

我使用自己的 MAC，從 Docker Hub 拉 sequenceiq/hadoop-docker 下來。之後，撰寫 mapper.py 跟 reducer.py，並產生 map reduce application，對教授提供的 log file 作 ip 次數及時間次數分析。

步驟及討論：

在實作本次作業的過程中，我原先是按照教授 GitHub 中 sdwangntu/hadoop-cluster 的指示，安裝 Hadoop Docker Container。在將 hue 的部分註解掉後，有安裝成功。但是，該 Container 的 Hadoop 版本為 3.1.2，指令與上課講義的 2.6.0 不同。與同學討論過，並以 google 查詢後無果。最後，我打算用 sequenceiq 的 Hadoop Docker Container 來完成本次作業。以下將依序說明我完成該作業的步驟，分別是設定 Hadoop 環境、撰寫程式碼、生成 map reduce application 等三個步驟。

1. 設定 Hadoop 環境

該步驟主要藉 https://hub.docker.com/r/sequenceiq/hadoop-docker/的指示，建立 Hadoop Docker Container

(1) 從 Docker Hub 拉 docker 下來

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

(2) 執行並進入 Container。注意要加入-p，之後在 localhost 才能看到 map reduce 結果

```
docker run -it -p 8088:8088 -p 50070:50070 sequenceiq/hadoop-docker:2.7.1
/etc/bootstrap.sh -bash
```

(3) 進入適當路徑即可開始開發

```
cd $HADOOP_PREFIX
```

2. 撰寫程式碼

詳細的程式碼在此不列出，放在我自己的 GitHub 中，https://github.com/tailer954/Cloud-Computing-and-Cyber-Security/tree/main/Hadoop-Cluster。比較值得提的是，可用 pipeline 來測試程式碼

(1) 從網路抓 log file 到本機

```
curl -o logFile.txt http://hpc.ee.ntu.edu.tw/html/IntelligentClouds/
webAccessLog/access_log
```

(2) 測試，做 ip 次數分析

```
cat logfile.txt | python ./mapper_ip.py | sort | python ./reducer.py
```

(3) 測試，做時間次數分析

```
cat logfile.txt | python ./mapper_time.py | sort | python ./reducer.py
```

```
ppp2.p33.is.com.ua        3
proxy0.haifa.ac.il        19
prxint-sxb2.e-i.net       1
prxint-sxb3.e-i.net       14
px7wh.vc.shawcable.net    1
rouble.cc.strath.ac.uk    1
spica.ukc.ac.uk 2
spot.nnacorp.com          5
trrc02m01-40.bctel.ca     4
ts04-ip92.hevanet.com     28
ts05-ip44.hevanet.com     16
```

```
07/Mar/2004:20:00:00    20
07/Mar/2004:21:00:00    23
07/Mar/2004:22:00:00    29
07/Mar/2004:23:00:00    22
08/Mar/2004:00:00:00    21
08/Mar/2004:01:00:00    21
08/Mar/2004:02:00:00    27
08/Mar/2004:03:00:00    22
08/Mar/2004:04:00:00    26
08/Mar/2004:05:00:00    37
08/Mar/2004:06:00:00    17
```

左圖是 ip 分析的結果截圖、右圖則是時間分析結果截圖

3. 生成 Map Reduce Application

將前段產生的 logFile 放入 hdfs 產生的 file 中。之後以 hadoop 提供的.jar 檔案，產生 map reduce application，並在 localhost 中顯示

(1) 以 hdfs 建立 logAnalyze

```
bin/hdfs dfs -mkdir logAnalyze
```

(2) 將 logFile 放入 logAnalyze 中

```
bin/hdfs dfs -copyFromLocal ./logfile.txt logAnalyze
```

(3) 以 hadoop-streaming-2.7.0.jar 產生 map reduce application

```
bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.0.jar \ -mapper “python /usr/local/hadoop/mapper_time.py" \ -reducer “python /usr/local/hadoop/reducer.py" \ -input “logAnalyze” \ -output “logAnalyze_outdir“
```

```
sh-4.1# bin/hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.0.jar \
> -mapper "python /usr/local/hadoop/mapper_time.py" \
> -reducer "python /usr/local/hadoop/reducer.py" \
> -input "logAnalyze" \
> -output "logAnalyze_outdir"
packageJobJar: [/tmp/hadoop-unjar2290589489327632036/] [] /tmp/streamjob8702523428602299739.jar tmpDir=null
20/11/03 02:03:20 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/11/03 02:03:20 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/11/03 02:03:21 INFO mapred.FileInputFormat: Total input paths to process : 1
20/11/03 02:03:21 INFO mapreduce.JobSubmitter: number of splits:2
20/11/03 02:03:21 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1604385735249_0002
20/11/03 02:03:22 INFO impl.YarnClientImpl: Submitted application application_1604385735249_0002
20/11/03 02:03:22 INFO mapreduce.Job: The url to track the job: http://36d8f389e878:8088/proxy/application_1604385735249_0002/
20/11/03 02:03:22 INFO mapreduce.Job: Running job: job_1604385735249_0002
20/11/03 02:03:31 INFO mapreduce.Job: Job job_1604385735249_0002 running in uber mode : false
20/11/03 02:03:31 INFO mapreduce.Job:  map 0% reduce 0%
20/11/03 02:03:37 INFO mapreduce.Job:  map 100% reduce 0%
20/11/03 02:03:44 INFO mapreduce.Job:  map 100% reduce 100%
20/11/03 02:03:44 INFO mapreduce.Job: Job job_1604385735249_0002 completed successfully
```

下 hadoop-streaming-2.7.0.jar 指令後的字串截圖

(4) 從 localhost:50070 看之

Overview '36d8f389e878:9000' (active)

Started:	Tue Nov 03 01:41:58 EST 2020
Version:	2.7.0, rd4c8d4d4d203c934e8074b31289a28724c0842cf
Compiled:	2015-04-10T18:40Z by jenkins from (detached from d4c8d4d)
Cluster ID:	CID-0955d4b8-86f4-4046-a270-53006f077ee0
Block Pool ID:	BP-754408308-172.17.9.73-1431769234492

(5) 從 localhost:8088 看之

Logged in as: dr.wh

hadoop

Nodes of the cluster

- Cluster
  - About
  - Nodes
  - Node Labels
  - Applications
    - NEW
    - NEW_SAVING
    - SUBMITTED
    - ACCEPTED
    - RUNNING
    - FINISHED
    - FAILED
    - KILLED
  - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	0	1	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>

Show 20 entries Search:

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
	/default-rack	RUNNING	36d8f389e878:43521	36d8f389e878:8042	Tue Nov 03 02:12:16 -0500 2020		0	0 B	8 GB	0	8	2.7.0

Showing 1 to 1 of 1 entries First Previous 1 Next Last