

Big Data Lab - Lab 4

RA Keerthan

CH17B078

1.a)

Firstly, a DataProc cluster was created using the following command

```
gcloud dataproc clusters create lab4-cluster --project cobalt-pursuit-304714 --region us-central1 --single-node
```

Next we submit *count_clicks.py* to the cluster using the following command

```
gcloud dataproc jobs submit pyspark count_clicks.py --cluster lab4-cluster --region us-central1 --gs://bucket_two_2/hash_file.txt -- gs://bd_lab4/output/
```

Below is the screenshot of successful execution

```
jobUuid: e6035967-ab7c-3260-a297-2ff7e99b6bd7
placement:
  clusterName: lab4-cluster
  clusterUuid: 6b0754df-28ee-44c2-aa2a-6557de58c833
pysparkJob:
  args:
  - gs://bucket_two_2/hash_file.txt
  - gs://bd_lab4/output/
  mainPythonFileUri: gs://dataproc-staging-us-central1-1045052396568-v67orppb/google-cloud-dataproc-metainfo/6
  unt_clicks.py
reference:
  jobId: 148ab4e748814e7aade94726fc5fa609
  projectId: cobalt-pursuit-304714
status:
  state: DONE
  stateStartTime: '2021-03-01T09:15:37.022Z'
statusHistory:
- state: PENDING
  stateStartTime: '2021-03-01T09:15:05.483Z'
- state: SETUP_DONE
  stateStartTime: '2021-03-01T09:15:05.519Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2021-03-01T09:15:05.779Z'
yarnApplications:
- name: count_clicks.py
  progress: 1.0
  state: FINISHED
  trackingUrl: http://lab4-cluster-m:8088/proxy/application_1614589570753_0002/
```

1.b)

Screenshot of output directory

bd_lab4

| | | | | |
|---------|---------------|-------------|-----------|-----------|
| OBJECTS | CONFIGURATION | PERMISSIONS | RETENTION | LIFECYCLE |
|---------|---------------|-------------|-----------|-----------|

Buckets > bd_lab4 > output

UPLOAD FILES UPLOAD FOLDER CREATE FOLDER MANAGE HOLDS DOWNLOAD DELETE

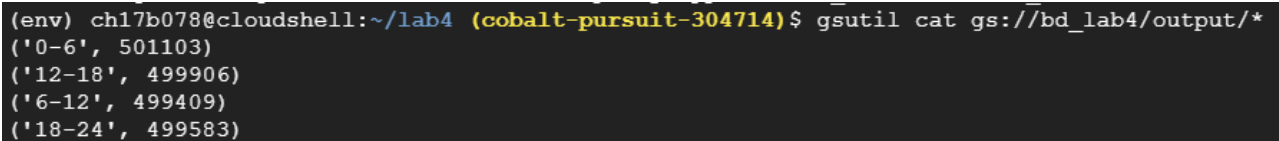
Filter by name prefix only ▾ Filter Filter objects and folders

| <input type="checkbox"/> | Name | Size | Type | Created time ? | Storage class | Last modified | Public access ? | Encryption ? | Re |
|--------------------------|---------|------|--------------------------|--------------------|---------------|---------------|-----------------|--------------------|----|
| <input type="checkbox"/> | _SUCC | 0 B | application/octet-stream | Mar 1, 2021, 2:... | Standard | Mar 1, 202... | Not public | Google-managed key | — |
| <input type="checkbox"/> | part-01 | 69 B | application/octet-stream | Mar 1, 2021, 2:... | Standard | Mar 1, 202... | Not public | Google-managed key | — |

The output of the job can be viewed through the following command

```
gsutil cat gs://bd_lab4/output/*
```

Screenshot of the output is attached below



```
(env) ch17b078@cloudshell:~/lab4 (cobalt-pursuit-304714)$ gsutil cat gs://bd_lab4/output/*
('0-6', 501103)
('12-18', 499906)
('6-12', 499409)
('18-24', 499583)
```

The output file can be found as *part-00000.txt* in my submission.

2.a)

HDFS is a network-based file system which can store very large datasets by abstracting the details of where the data is residing with the help of a mapping table. It uses a cluster (combined storage) of small storage devices to store large data. If more storage space is needed, HDFS does not require one to alter the entire hardware but rather it is enough if one adds adequate number of extra storage nodes (machines) to the cluster. To handle failure of a machine in the network, HDFS saves in total, three copies of the same machine (one main machine + two copies of it). The two copies reside in two other machines. Therefore, a machine failure will not result in data loss. HDFS achieves a high throughput whereas it is not good for interactive requirements because of its high latency.

2.b)

Hive was primarily developed to enable users to interact with data present in HDFS through SQL-like queries. To achieve this, Hive converts SQL queries into MapReduce queries which is then executed in Yarn on the data that is present in HDFS. Hive also provides functionality for analysing and summarizing data.

2.c)

Pig, like Hive, is a framework for data processing and manipulation. While Hive abstracts MapReduce and converts it to SQL, Pig abstracts MapReduce and converts it to a scripting language.

2.d)

Yarn is responsible for cluster resource management and serves as an abstraction layer for compute. It behaves like an Operating System of a cluster. Yarn also contains the information regarding number of resources available in the cluster and number of resources that have no jobs running in them. Yarn has the non-functional requirement that jobs which failed are automatically re-tried. Thereby, it has the responsibility of job completion in event of machine failures. Another non-functional responsibility of Yarn is to provide scalability i.e, for example, if more storage is required, Yarn is responsible to run the same task (with the same code) along with increasing the number of machines in the cluster.