

Project id: cdsds561-project-1  
Bucket Name: hw2nirbhgsutil , gcf-check-files  
Bucket-2 Name:hw4nirbhgsutil  
Directory Name: test-dir , gcf-check-files/test-dir  
Github Link:[CDS-DS-561-hw2/hw6 at master · nirbhay221/CDS-DS-561-hw2 \(github.com\)](https://github.com/nirbhay221/CDS-DS-561-hw2)  
Topic name : projects/cdsds561-project-1/topics/hw3nirbhgsutil  
Topic Id: hw3nirbhgsutil  
Subscription Id: hw3nirbhgsutil-sub  
Service Account email for pub sub:  
pubsubserviceacc-hw3-nirbh@cdsds561-project-1.iam.gserviceaccount.com  
Service Account for pub sub: pubsubServiceAcc-hw3-nirbh  
Service Account for SQL:  
cloud-sql-authorize-vm@cdsds561-project-1.iam.gserviceaccount.com  
Reserved IP : 34.75.102.252  
Sql database instance: my-database  
Sql database: First-Trial , Second-Trial  
Sql Tables : Clients, main\_table, error\_logs.  
Vm-instance : 5 - directory - model-1 - main.py , main-1.py , 6 - main.py , main-1.py

For the following codes to run you need to install the following libraries:

```
pip install beautifulsoup4
```

```
pip install apache-beam
```

```
pip install 'apache-beam[gcp]'
```

```
pip install google-cloud-storage
```

Also for enabling the google cloud dataflow I used the following commands:

```
gcloud config set project cdsds561-project-1
```

```
for i in dataflow compute_component logging storage_component storage_api bigquery pubsub  
datastore.googleapis.com cloudresourcemanager.googleapis.com; do gcloud services enable $i; done
```

```
for i in roles/dataflow.admin roles/dataflow.worker roles/storage.objectAdmin; do gcloud projects  
add-iam-policy-binding cdsds561-project-1  
--member="serviceaccount:494559990174-compute@developer.gserviceaccount.com" $i; done
```

I have three codes for the following assignment, one is apachebeamBucketTrial-3.py which works locally and completes the parsing of the bucket files and then provides the top five files with the most outgoing links and the top five files with the most incoming links, second is apachebeamBucketTrial-5.py which works on the cloud with the dataflow runner and completes the parsing of the bucket files and then provides the top five files with the most outgoing links

and the top five files with the most incoming links, apachebeamBucketTrial-6.py which works locally with the dataflow runner and completes the parsing of the bucket files and then provides the top five files with the most outgoing links and the top five files with the most incoming links.

The apachebeamBucketTrial-3.py defines a data processing pipeline which uses apache beam for analyzing the HTML files stored in the google cloud storage bucket and the pipeline extracts the links from the HTML files, storing all the outgoing links for each file in the outgoing link dictionary and we create the incoming link dictionary from the outgoing link dictionary, it utilizes the beautiful soup library for html parsing and apache beam for parallel data processing. Using the outgoing link dictionary and the incoming link dictionary, the pipeline provides us with the top five files with the most outgoing links and the top five files with the most incoming links.

We use the following command to run it :

```
C:\Users\nirbh\AppData\Local\Google\Cloud SDK>python databeamBucketTrial-3.py --bucket  
hw2nirbhgsutil test-dir
```

Once, we run the following command, we get the following output:

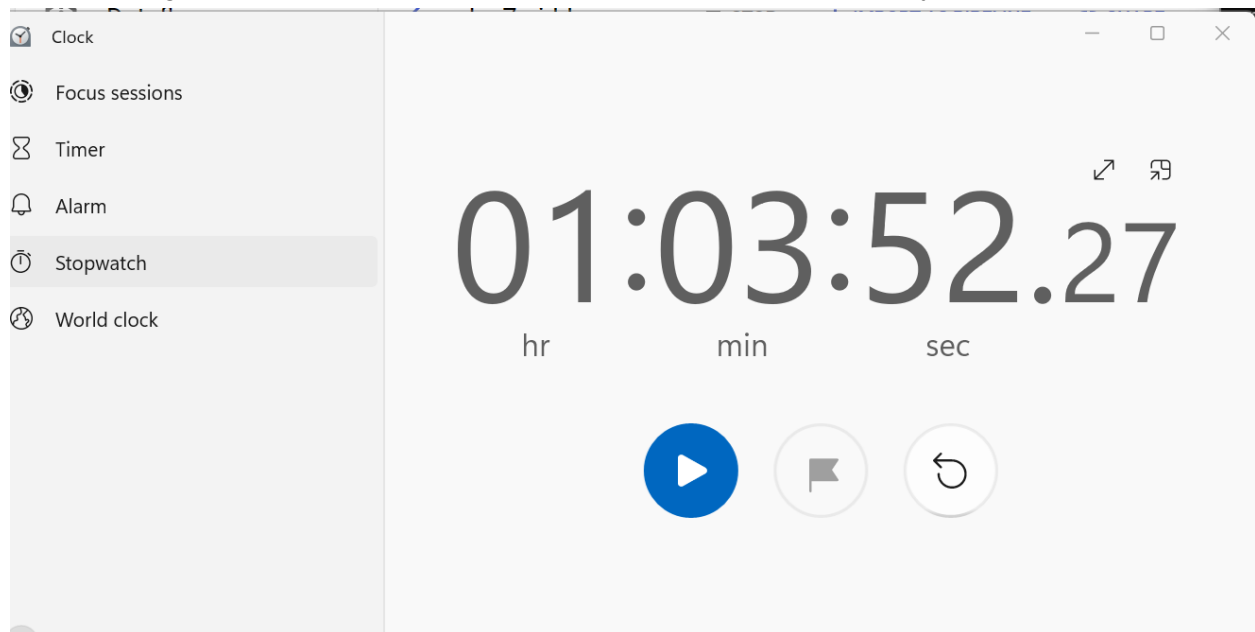
```
sed do eu'...  
Top 5 files with the largest number of incoming links:  
File: 5984.html, Incoming Links Count: 188  
File: 5789.html, Incoming Links Count: 163  
File: 1912.html, Incoming Links Count: 162  
File: 2675.html, Incoming Links Count: 160  
File: 3207.html, Incoming Links Count: 160  
  
Top 5 files with the largest number of outgoing links:  
File: 4168.html, Outgoing Links Count: 249  
File: 7642.html, Outgoing Links Count: 249  
File: 2613.html, Outgoing Links Count: 248  
File: 3641.html, Outgoing Links Count: 248  
File: 6818.html, Outgoing Links Count: 248
```

With the changes the output might look like this right now:

```
, sed do eiu'...
Top 5 files with the largest number of incoming links:
File: 5984.html, Incoming Links Count: 188
File: 5789.html, Incoming Links Count: 163
File: 1912.html, Incoming Links Count: 162
File: 2675.html, Incoming Links Count: 160
File: 3207.html, Incoming Links Count: 160
WARNING:apache_beam.options.pipeline_options:Discarding unparseable args: ['databeamBucketTrial-3.py', '-
-bucket', 'hw2nirbhgsutil', '--directory', 'test-dir']

Top 5 files with the largest number of outgoing links:
File: 4168.html, Outgoing Links Count: 249
File: 7642.html, Outgoing Links Count: 249
File: 2613.html, Outgoing Links Count: 248
File: 3641.html, Outgoing Links Count: 248
File: 6818.html, Outgoing Links Count: 248
WARNING:apache_beam.options.pipeline_options:Discarding unparseable args: ['databeamBucketTrial-3.py', '-
-bucket', 'hw2nirbhgsutil', '--directory', 'test-dir']
```

The following code completes in an hour and 3 minutes which is displayed below:



Other than this, I modified the above code to get the apachebeamBucketTrial-5.py as the code for the dataflow runner to run on the cloud. It implements a google cloud dataflow pipeline using apache beam for processing the html files stored in the google cloud storage bucket and it defines a pipeline that reads html content from the files in the bucket and extracts the outgoing links forming the outgoing link dictionary, makes the incoming link dictionary from the outgoing link dictionary, and then prints the top 5 files with the most incoming and outgoing links. The following pipeline is configured to run on the dataflow service and the code uses the apache beam and the google cloud storage for the data processing.

Also the files are stored in the gcf-check-files bucket's test-dir directory, you can locate the following files as follows:

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

TRANSFER DATA

MANAGE HOLDS

DOWNLOAD









DELETE

Filter by name prefix only

Filter

Filter objects and folders

Show deleted data

<input type="checkbox"/>	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption		
<input type="checkbox"/>	 <a href="#">top_incoming_files.txt-00000-of-0...</a>	215 B	text/plain	Nov 13, 2023, 12:23:32 AM	Standard	Nov 13, 2023, 12:23:32 AM	 Public to internet	<a href="#">Copy URL</a>	—	Google-ma	 
<input type="checkbox"/>	 <a href="#">top_outgoing_files.txt-00000-of-0...</a>	215 B	text/plain	Nov 13, 2023, 12:23:34 AM	Standard	Nov 13, 2023, 12:23:34 AM	 Public to internet	<a href="#">Copy URL</a>	—	Google-ma	 

Rows per page:

100

101 – 102 of 102

The output for the following files are as follows:

Top 5 incoming files:

```
File: 5984.html, Incoming Links Count: 188
File: 5789.html, Incoming Links Count: 163
File: 1912.html, Incoming Links Count: 162
File: 2675.html, Incoming Links Count: 160
File: 3207.html, Incoming Links Count: 160
```

Top 5 outgoing files:

```
File: 4168.html, Outgoing Links Count: 249
File: 7642.html, Outgoing Links Count: 249
File: 2613.html, Outgoing Links Count: 248
File: 3641.html, Outgoing Links Count: 248
File: 6818.html, Outgoing Links Count: 248
```

We can use the following bat file to execute the apachebeamBucketTrial-5.py using the dataflow runner as follows:

```
C:\Users\nirbh\AppData\Local\Google\Cloud SDK>cat beam-Trial.bat













python databeamBucketTrial-5.py ^
  --bucket=hw2nirbhgsutil ^
  --directory=test-dir ^
  --num_workers=1 ^
  --max_num_workers=1
```

I used one worker because due to parallelism several outgoing link dictionaries developed which caused it hard to find the top 5 files.




Now we can run the following bat file as follows:

```
C:\Users\nirbh\AppData\Local\Google\Cloud SDK>beam-Trial.bat
```

Now, can see the output on the dataflow jobs and once we select the right job we can head to the worker logs to see the following logs printed:

> 	2023-11-12 21:30:13.722 EST	Outgoing Dictionary Length:10000
> 	2023-11-12 21:30:13.733 EST	Top Outgoing Files: 7642.html, Outgoing Links Count: 249
> 	2023-11-12 21:30:13.733 EST	Top Outgoing Files: 4168.html, Outgoing Links Count: 249
> 	2023-11-12 21:30:13.733 EST	Top Outgoing Files: 3641.html, Outgoing Links Count: 248
> 	2023-11-12 21:30:13.733 EST	Top Outgoing Files: 2613.html, Outgoing Links Count: 248
> 	2023-11-12 21:30:13.733 EST	Top Outgoing Files: 6818.html, Outgoing Links Count: 248
...		
> 	2023-11-12 21:30:15.911 EST	Top Incoming Files: 5984.html, Incoming Links Count: 188
> 	2023-11-12 21:30:15.912 EST	Top Incoming Files: 5789.html, Incoming Links Count: 163
> 	2023-11-12 21:30:15.912 EST	Top Incoming Files: 1912.html, Incoming Links Count: 162
> 	2023-11-12 21:30:15.912 EST	Top Incoming Files: 2675.html, Incoming Links Count: 160
> 	2023-11-12 21:30:15.912 EST	Top Incoming Files: 3207.html, Incoming Links Count: 160
> 	2023-11-12 21:30:16.013 EST	Discarding unparseable args: [' /usr/local/lib/python3.11/site-packages/ar

And we can see the elapsed time for the following job as follows:

Name	Type	End time	Elapsed time	Start time	Status	SDK version	ID	Region	Insights 
 hw7-nirb-gs	Batch	Nov 12, 2023, 9:31:00 PM	25 min 27 sec	Nov 12, 2023, 9:05:33 PM	Succeeded	2.51.0	2023-11-12_18_05_33-14120388199646629819	us-east1	 1 INSIGHT

It took about 25 min 27 seconds for it to complete on the cloud using the dataflow runner which is far better than the local version.

The files created for storing the output are in the bucket: gcf-check-files and they look like the following:

	test-dir/	Folder									
	top_incoming_files.txt-00000-of-0...	285 B	text/plain	Nov 12, 2023, 9:30:16 PM	Standard	Nov 12, 2023, 9:30:16 PM	Public to internet	Copy URL	—		
	top_outgoing_files.txt-00000-of-0...	285 B	text/plain	Nov 12, 2023, 9:30:14 PM	Standard	Nov 12, 2023, 9:30:14 PM	Public to internet	Copy URL	—		

Other than this you can see the whole view for this as follows:

Bucket details

REFRESHLEARN

gcf-check-files

Public to internet: This bucket is publicly accessible because allUsers or allAuthenticatedUsers have one or more permissions. Remove these principals to stop public access.

EDIT ACCESSDISMISS

Location

Storage class

Public access

Protection

us-east1 (South Carolina)

Standard

Public to internet

None

OBJECTS

CONFIGURATION

PERMISSIONS

PROTECTION

LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

Buckets > gcf-check-files

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

TRANSFER DATA

MANAGE HOLDS

DOWNLOAD

DELETE

Filter by name prefix only

Filter

Filter objects and folders

Show deleted data

<input type="checkbox"/>	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	
<input type="checkbox"/>	incoming_links_output.txt	0 B	application/octet-stream	Nov 9, 2023, 3:49:23 AM	Standard	Nov 9, 2023, 3:49:23 AM	Public to internet	Copy URL	
<input type="checkbox"/>	outgoing_links_output.txt	0 B	application/octet-stream	Nov 9, 2023, 3:49:23 AM	Standard	Nov 9, 2023, 3:49:23 AM	Public to internet	Copy URL	
<input type="checkbox"/>	temp/	—	Folder	—	—	—	—	—	
<input type="checkbox"/>	test-dir/	—	Folder	—	—	—	—	—	
<input type="checkbox"/>	top_incoming_files.txt-00000-of-0...	285 B	text/plain	Nov 12, 2023, 9:30:16 PM	Standard	Nov 12, 2023, 9:30:16 PM	Public to internet	Copy URL	
<input type="checkbox"/>	top_outgoing_files.txt-00000-of-0...	285 B	text/plain	Nov 12, 2023, 9:30:14 PM	Standard	Nov 12, 2023, 9:30:14 PM	Public to internet	Copy URL	

Other than this in the gcf-check-files bucket I saved the output for the top five files with the most incoming links where gcf-check-files/top\_incoming\_files.txt-00000-of-00001 looks as follows:

```
Top Incoming Files: 5984.html, Incoming Links Count: 188
Top Incoming Files: 5789.html, Incoming Links Count: 163
Top Incoming Files: 1912.html, Incoming Links Count: 162
Top Incoming Files: 2675.html, Incoming Links Count: 160
Top Incoming Files: 3207.html, Incoming Links Count: 160
```

And in the gcf-check-files bucket I saved the output for the top five files with the most outgoing links where gcf-check-files/top\_outgoing\_files.txt-00000-of-00001 looks as follows:

```
Top Outgoing Files: 7642.html, Outgoing Links Count: 249
Top Outgoing Files: 4168.html, Outgoing Links Count: 249
Top Outgoing Files: 3641.html, Outgoing Links Count: 248
Top Outgoing Files: 2613.html, Outgoing Links Count: 248
Top Outgoing Files: 6818.html, Outgoing Links Count: 248
```

Other than this for running the local code, we can modify the above program of the apachebeamBucketTrial-5.py by using the Direct Runner in the apachebeamBucketTrial-6.py which can help in running the following dataflow pipeline locally. The following code uses apache beam for processing the HTML Files in a google cloud storage bucket and it extracts the outgoing links from each html file, forming outgoing link dictionary and then creating the incoming link dictionary from the outgoing link dictionary and then prints the top five files with the most outgoing and incoming links.

We can use the following bat file to execute the apachebeamBucketTrial-6.py using the direct runner as follows:

```
C:\Users\nirbh\AppData\Local\Google\Cloud SDK>cat beam-Trial-1.bat
python databeamBucketTrial-6.py ^
--bucket=hw2nirbhgsutil ^
--directory=test-dir ^
--num_workers=1 ^
--max_num_workers=1
```

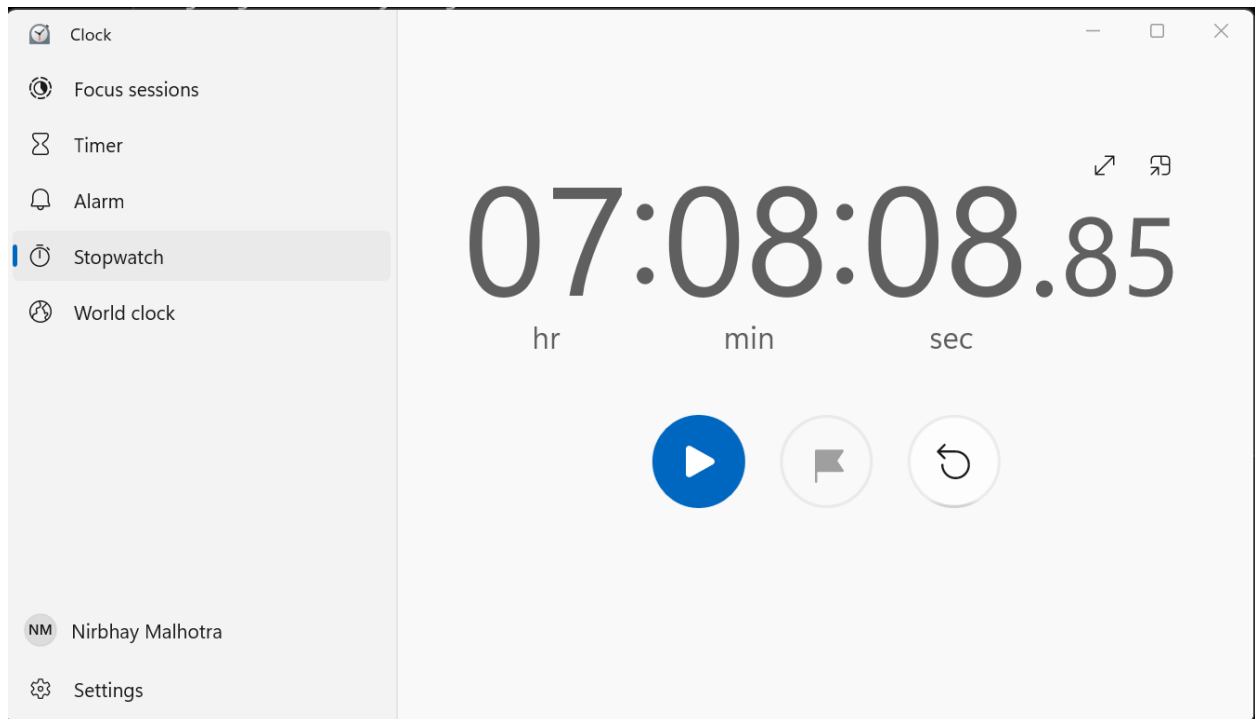
Now we can run the following bat file as follows:

```
C:\Users\nirbh\AppData\Local\Google\Cloud SDK>beam-Trial-1.bat
```

Now, can see the output on the terminal as follows:

```
Outgoing File: 4168.html, Outgoing Links Count: 249
Outgoing File: 7642.html, Outgoing Links Count: 249
Outgoing File: 2613.html, Outgoing Links Count: 248
Outgoing File: 3641.html, Outgoing Links Count: 248
Outgoing File: 6818.html, Outgoing Links Count: 248
Top Incoming File: 5984.html, Incoming Links Count: 188
Top Incoming File: 5789.html, Incoming Links Count: 163
Top Incoming File: 1912.html, Incoming Links Count: 162
Top Incoming File: 2675.html, Incoming Links Count: 160
Top Incoming File: 3207.html, Incoming Links Count: 160
```

It takes about this much time :



The following time interval for processing all the files is really high for this one with a direct runner.

This direct runner will also save the files in the gcf-check-files and overwrite them as follows :

Top outgoing files:



```
File: 4168.html, Outgoing Links Count: 249
File: 7642.html, Outgoing Links Count: 249
File: 2613.html, Outgoing Links Count: 248
File: 3641.html, Outgoing Links Count: 248
File: 6818.html, Outgoing Links Count: 248
```

Top Incoming files:

```
File: 5984.html, Incoming Links Count: 188
File: 5789.html, Incoming Links Count: 163
File: 1912.html, Incoming Links Count: 162
File: 2675.html, Incoming Links Count: 160
File: 3207.html, Incoming Links Count: 160
```

File output right now might be different as I ran the code one last time to check them.

Also, dataflow runner code was faster than the local code execution because the data flow runner takes advantage of the Google cloud dataflow's distributed processing capabilities and it can parallelize the data processing across multiple workers and scale resources based on the input data size which can reduce the overall execution time. It utilizes the google cloud infrastructure which can allocate substantial computing resources for processing. This is in contrast to local execution, where the resources are limited to the machine where the code is running. Dataflow handles data distribution and shuffling, optimizing the data processing across distributed environments. It can also dynamically scale resources based on the workload for which i faced several issues.