Map-Reduce With VMs and Cloud Functions

-Harsh Atha

1. Objective:
   The objective here is to deploye Map Reduce using 2 methods
   a. Using VMs for each Mapper, reducer and master in Part 1
   b. Using FaaS Google Cloud Functions for Mapper and Reducer.

2. Design Details:
   The code for both aspects is more or less similar but both have been treated as an independent assignment (A4_P1 and A4_P2 as seen in folders).

   Check the readme file of both folders to learn how to execute each part.

   A. Part 1:
   In Part 1 of VMs, we create a server instance first, upload all necessary py files, install requirements and create an image based on that instance. We use this same image to create the required mappers and reducers later in order to avoid copying of files again.

   Once this part is done, the upload files part will create the necessary buckets for this part and will upload the required text files which are input for running the map reduce job to the required bucket.

   The execute.sh file will execute the code and start the process.

   a. Mapper.py
   This part takes in the chunk txt file from the bucket, create a string word_1_file for each word in that chunk and store it as a txt file on the bucket. It then returns the name of the temporary txt file created by the mapper. The ip address of master, port are either based on config file or on dynamic ip creation of server instance which is stored in a txt file.

   b. Reducer.py
   Here it receives a key and intermediate file name as parameters. Based on the key it will access values of the interemediate json file and start combining the counts and do inverted index for those sets of words. It

will finish its job and store the data for WC and II in respective files of the final bucket. It sends "DONE_REDUCER" as output.

c. Master.py

This is the main file that will execute the entire process. It firsts initializes the cluster and goes on a listening state for incoming connections. In the map_red function, it divides data into chunks, sends the names to mapper VMs. For creating mapper VMs we use "compute instances bulk" to parallely create multiple VMs together. We then ssh into each of the Mapper VMs, run the required mapper code and wait for a response using multi threading. Once the required number of acknowledgements have been received, it goes to next step. Here we combine the data to interemediate JSON file based on the hashing of word and send key, file name to reducer. Similar to above process, we bulk create VMs and wait for output from each VM to finish. Once we receive individual reducer outputs, we combine them and store them in final bucket. Once that part is over we destroy the cluster.

B. Part 2

In this part we are mainly using FaaS feature of GCP to create 4 major functions. The start.py file will create the required bucket, upload required files to bucket and call the map_red function.

a. Map_red

This is similar to master of part 1 where instead of SSHing into VMs we simply call map_function and red_fn. We use asyncio calls to ensure parallel HTTPS requests and responses. This function is an event triggered function and will execute when a new text file is uploaded to hatha-a4-bucket1. The upload is controlled by config file and start.py but can be done manually as well provided the config file exists in respective bucket.

b. Map_function

This is exactly the same as Mapper.py except it is called by an API from map_red and the arguments format is different. The responses is temp_mapper_file_name.

c. Red_fn

This is exactly the same as Reducer.py except it is called by an API from map_red and the arguments format is different. The response is "REDUCER_DONE".

d. Ui_fn

This function is called by the front end html page in a form type request. It returns "Bucket does not exist. Run Mapreduce atleast once." if the bucket hasn't been created and we try to run the search. If we are trying to run the search while a document is being processesed it will return "Mapreduce in Progress. Please Try again". If the word is not present in any document it will return "Word not found in any document". If the word is present in any document, it will return in format {document1:count1, document2:count2}

3. Evaluation and Testing

Please find screenshots of each step of each part below

A. Part 1

a. Server instance being created

b. Mapper VMs being created:

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
hatha@server-instance:~$ python3 master.py
Input for mappers created
NAME       ZONE
mapper-1  us-west1-a
mapper-2  us-west1-a
mapper-3  us-west1-a
Bulk create request finished with status message: [VM instances created: 3, failed: 0.]
```

c. Connected to mappers, received temp file names, reducers being created, reducer job and mapreduce job done.

```
Bulk create request finished with status message: [VM instances created: 3, failed: 0.]
waiting for a mapper connection
Warning: Permanently added 'compute.8651724206601941519' (ECDSA) to the list of known hosts.
Warning: Permanently added 'compute.2042510424211468815' (ECDSA) to the list of known hosts.
Warning: Permanently added 'compute.8814442667563299343' (ECDSA) to the list of known hosts.
connection from mapper ('10.138.0.6', 52370)
waiting for a mapper connection
connection from mapper ('10.138.0.7', 56550)
waiting for a mapper connection
connection from mapper ('10.138.0.8', 43940)
Mapper Done
temp_mapper_ip_1_large.txt
temp_mapper_ip_2_large.txt
temp_mapper_ip_3_large.txt
Intermediate mapper output created
Deleted [https://www.googleapis.com/compute/v1/projects/hatha-fa22/zones/us-west1-a/instances/mapper-1].
Deleted [https://www.googleapis.com/compute/v1/projects/hatha-fa22/zones/us-west1-a/instances/mapper-3].
NAME       ZONE
reducer-1  us-west1-a
reducer-2  us-west1-a
reducer-3  us-west1-a
Bulk create request finished with status message: [VM instances created: 3, failed: 0.]
Deleted [https://www.googleapis.com/compute/v1/projects/hatha-fa22/zones/us-west1-a/instances/mapper-2].
waiting for a reducer connection
Warning: Permanently added 'compute.1146880619098682314' (ECDSA) to the list of known hosts.
Warning: Permanently added 'compute.1639334929238602698' (ECDSA) to the list of known hosts.
Warning: Permanently added 'compute.5892123507498169290' (ECDSA) to the list of known hosts.
connection from mapper ('10.138.0.10', 51174)
waiting for a reducer connection
connection from mapper ('10.138.0.11', 56142)
waiting for a reducer connection
connection from mapper ('10.138.0.9', 56350)
Intermediate Reducer Output Created
Master performing combination of reducer outputs begins
MAP REDUCE is Done
DONE
```

d. Output of Word count stored on bucket

"the": 5548, "gutenberg": 97, "ebook": 13, "of": 2592, "doyle": 3, "anyone": 8, "in": 1707, "most": 91, "with": 817, "you": 1383, "give": 63, "under": 56, "terms": 23, "org": 9, "located": 7, "will": 203, "to": 2535, "before": 169, "using": 8, "title": 1, ...

e. Output of Inverted Index

"the": ["large", "medium"], "gutenberg": ["large", "medium"], "ebook": ["large", "medium"], "of": ["large", "medium"], "doyle": ["large"], "anyone": ["large", "medium"], "in": ["large", "medium"], "most": ["large", "medium"], "with": ["large", "medium"], "you": ["large", "medium"], "give": ["large", "medium"], "under": ["large", "medium"], "terms": ["large", "medium"], "org": ["large", "medium"], "located": ["large", "medium"], "will": ["large", "medium"], "to": ["large", "medium"], "before": ["large", "medium"], ...

f. All intermediate files being created in buckets.

	Name	Size	Type	Created	Sto
☐	mapper_ip_1_large.txt	175.2 KB	text/plain	Nov 29, 2022, 12:51:55 AM	Sta
☐	mapper_ip_1_medium.txt	23.8 KB	text/plain	Nov 29, 2022, 12:57:17 AM	Sta
☐	mapper_ip_2_large.txt	176.2 KB	text/plain	Nov 29, 2022, 12:51:57 AM	Sta
☐	mapper_ip_2_medium.txt	24.3 KB	text/plain	Nov 29, 2022, 12:57:19 AM	Sta
☐	mapper_ip_3_large.txt	179.7 KB	text/plain	Nov 29, 2022, 12:51:59 AM	Sta
☐	mapper_ip_3_medium.txt	25.7 KB	text/plain	Nov 29, 2022, 12:57:21 AM	Sta

	Name	Size	Type	Created ❓	Storage class
☐	📄 intermediate_output_large.json	1.5 MB	application/json	Nov 29, 2022, 12:53:20 AM	Standard
☐	📄 intermediate_output_medium.json	225.5 KB	application/json	Nov 29, 2022, 12:58:26 AM	Standard
☐	📄 temp_mapper_ip_1_large.txt	425.4 KB	text/plain	Nov 29, 2022, 12:53:05 AM	Standard
☐	📄 temp_mapper_ip_1_medium.txt	61.3 KB	text/plain	Nov 29, 2022, 12:58:11 AM	Standard
☐	📄 temp_mapper_ip_2_large.txt	426.9 KB	text/plain	Nov 29, 2022, 12:52:59 AM	Standard
☐	📄 temp_mapper_ip_2_medium.txt	61.8 KB	text/plain	Nov 29, 2022, 12:58:11 AM	Standard
☐	📄 temp_mapper_ip_3_large.txt	430.9 KB	text/plain	Nov 29, 2022, 12:52:59 AM	Standard
☐	📄 temp_mapper_ip_3_medium.txt	64.2 KB	text/plain	Nov 29, 2022, 12:58:11 AM	Standard

	Name	Size	Type	Created ❓	Storage class	Last modified	Pul
☐	📄 ii_1intermediate_output_large.json	50.3 KB	application/json	Nov 29, 2022, 12:54:14 AM	Standard	Nov 29, 2022, 12:54:14 AM	No
☐	📄 ii_1intermediate_output_medium.j...	20.4 KB	application/json	Nov 29, 2022, 12:59:20 AM	Standard	Nov 29, 2022, 12:59:20 AM	No
☐	📄 ii_2intermediate_output_large.json	48 KB	application/json	Nov 29, 2022, 12:54:14 AM	Standard	Nov 29, 2022, 12:54:14 AM	No
☐	📄 ii_2intermediate_output_medium.j...	21.5 KB	application/json	Nov 29, 2022, 12:59:20 AM	Standard	Nov 29, 2022, 12:59:20 AM	No
☐	📄 ii_3intermediate_output_large.json	51.9 KB	application/json	Nov 29, 2022, 12:54:14 AM	Standard	Nov 29, 2022, 12:54:14 AM	No
☐	📄 ii_3intermediate_output_medium.j...	20.1 KB	application/json	Nov 29, 2022, 12:59:20 AM	Standard	Nov 29, 2022, 12:59:20 AM	No
☐	📄 wc_1intermediate_output_large.json	35.6 KB	application/json	Nov 29, 2022, 12:54:14 AM	Standard	Nov 29, 2022, 12:54:14 AM	No
☐	📄 wc_1intermediate_output_medium...	13.6 KB	application/json	Nov 29, 2022, 12:59:20 AM	Standard	Nov 29, 2022, 12:59:20 AM	No
☐	📄 wc_2intermediate_output_large.json	34 KB	application/json	Nov 29, 2022, 12:54:14 AM	Standard	Nov 29, 2022, 12:54:14 AM	No
☐	📄 wc_2intermediate_output_medium...	14.4 KB	application/json	Nov 29, 2022, 12:59:20 AM	Standard	Nov 29, 2022, 12:59:20 AM	No
☐	📄 wc_3intermediate_output_large.json	36.7 KB	application/json	Nov 29, 2022, 12:54:14 AM	Standard	Nov 29, 2022, 12:54:14 AM	No
☐	📄 wc_3intermediate_output_medium...	13.5 KB	application/json	Nov 29, 2022, 12:59:20 AM	Standard	Nov 29, 2022, 12:59:20 AM	No

	Name	Size	Type	Created ❓	Storage class
☐	📄 output_ii.json	208.5 KB	application/json	Nov 29, 2022, 12:59:36 AM	Standard
☐	📄 output_wc_large.json	106.3 KB	application/json	Nov 29, 2022, 12:54:29 AM	Standard
☐	📄 output_wc_medium.json	41.4 KB	application/json	Nov 29, 2022, 12:59:36 AM	Standard

g. Time taken to run entire Map reduce process (excluding server creation, file upload)

```
MAP REDUCE is Done
DONE
Socket Closed

real    2m36.681s
user    0m2.338s
sys     0m0.425s
```

B. Part 2

a. Faas functions deployed on project

		Environment	Name ↑	Last deployed	Region	Trigger	Runtime	Memory allocated	Executed function	Authentication ❓	Actions
☐	✅	1st gen	map_function	Nov 27, 2022, 6:15:26 PM	us-east1	HTTP	Python 3.9	256 MB	map_function	Allow unauthenticated	⋮
☐	✅	1st gen	map_red	Dec 3, 2022, 1:51:42 PM	us-east1	Bucket: hatha-a4-bucket1	Python 3.9	256 MB	map_red		⋮
☐	✅	1st gen	red_fn	Nov 27, 2022, 6:16:02 PM	us-east1	HTTP	Python 3.9	256 MB	red_fn	Allow unauthenticated	⋮
☐	✅	1st gen	ui_fn	Dec 3, 2022, 2:02:20 PM	us-east1	HTTP	Python 3.9	256 MB	ui_fn	Allow unauthenticated	⋮

b. Front end screenshot

c. Result of queried word "the" being displayed.

d. Time taken to run entire process

```
Checking if all required buckets exist. Creating buckets if it doesn't ex
ist.
All required buckets are created
DONE
Map Reduce job for given file has been completed.

real    0m11.737s
user    0m1.049s
sys     0m0.122s
```

4. Pricing
   A. Part 1
      For an estimate of 5hrs/day for 7(assuming 1 master and 3 reducers, mappers each) e2 standard instances running all 7 days a week, it costs about 71.34$/month.

      The speed of a cold start is about 2mins 36s.

7 X		
Region: South Carolina		
1,064.583 total hours per month		
Provisioning model: Regular		
Instance type: e2-standard-2		USD 71.34
Operating System / Software: Free		
Estimated Component Cost: USD 71.34 per 1 month		

      Total Estimated Cost: USD 71.34 per 1 month
      Estimate Currency
      USD - US Dollar

      If we keep all instances running 24x7, it will cost about 342$ per month

7 X		
Region: South Carolina		
5,110 total hours per month		
Provisioning model: Regular		
Instance type: e2-standard-2		USD 342.43
Operating System / Software: Free		
Estimated Component Cost: USD 342.43 per 1 month		

      Total Estimated Cost: USD 413.77 per 1 month
      Estimate Currency
      USD - US Dollar

      Based on storage requirements, we need to add costs of buckets as well. In my case I used 5 individual buckets, but we can reduce that redundancy by using folders.

B. Part 2

a. Invocation Cost

Invocations per month	Price/million
First 2 million	Free
Beyond 2 million	$0.40

b. Compute Time costs

Memory	vCPU[1]	Price/100ms (Tier 1 Price)	Price/100ms (Tier 2 Price)
128MB	.083 vCPU	$0.000000231	$0.000000324
256MB	.167 vCPU	$0.000000463	$0.000000648
512MB	.333 vCPU	$0.000000925	$0.000001295
1024MB	.583 vCPU	$0.000001650	$0.000002310
2048MB	1 vCPU	$0.000002900	$0.000004060
4096MB	2 vCPU	$0.000005800	$0.000008120

The overall cost for this method is pretty low as the cost is based on per million invocations.

5. Comparison

In Part 1 the time consumed is pretty high for a cold start of Mappers and Reducers. Additionally the cost is pretty high as well. FaaS on the other hand has a lower cost as well as a faster run time. If our application demands keeping it alive 24*7, I am assuming the VMs with high compute powers and warm starts can give faster results. But at the current cost, I am more inclined to Cloud Functions.

6. Issues

1. In part 1 there are some times when SSH into each of VMs is not done properly and gets stuck in a loop. In order to resolve that issue, kill the run and re run the code.
2. For part 2, it is imperative that the bucket exists at the time of function being redeployed. If bucket does not exist, the event trigger Streaming Search does not work.