Lab 6 - Assignment

MM16B023[1]

[a]*Indian Institute of Technology Madras*

Keyword: Pub, Sub, Pull, Push, Subscriptions

Abstract: This paper presents the solutions to fourth assignment of the Big Data Laboratory course (CS4830) at *IIT Madras*. All the notations used are as according with the textbook Mining of massive data sets by Anand Rajaraman

Problem 1

Count the number of lines in a file uploaded to GCS bucket in real-time by using Google Cloud Functions and Pub/Sub

- Write a Google cloud Function which gets triggered whenever a file is added to a bucket and publishes the file name to a topic in Pub/Sub.
- Write a python file, which acts as a subscriber to this topic and prints out the number of lines in the file in real-time

Solution:

Trigger

```
def message(data, context):
    from google.cloud import pubsub_v1
    publish_client = pubsub_v1.PublisherClient()
    topic_name = 'projects/hopeful-buckeye-266720/topics/topic_lab6'
    publisher = pubsub_v1.PublisherClient()
    publisher.create_topic(topic_name)
    output = data['name']
    output = output.encode("utf-8")
    publish_client.publish(topic_name, output)
    print("The massaged has been recieved")
```

No. of lines

```
### Written by H.Vishal MM16B023 at 11:17 AM on 07/03/2020 ###

from google.cloud import pubsub_v1
from google.cloud import storage

subscriber = pubsub_v1.SubscriberClient()
topic_name = 'projects/hopeful-buckeye-266720/topics/topic_lab6'
subscription_name = 'projects/hopeful-buckeye-266720/subscriptions/subscript'
subscriber = pubsub_v1.SubscriberClient()
subscriber.create_subscription(name=subscription_name, topic=topic_name)

def callback(package):
    x = package
    print(x.data)
    print('recieved')
    with open('addresses.csv', 'r') as f:
```

```
        count = 0
        for line in f:
            count = count+1
    print('Expected line count: ' + str(count))
    package.ack()

future = subscriber.subscribe(subscription_name, callback)

try:
    future.result()
except KeyboardInterrupt:
    future.cancel()
```

```
hvishal512@cloudshell:~ (hopeful-buckeye-266720)$ python3 subscription.py
The massaged has been recieved
Expected line count: 22
hvishal512@cloudshell:~ (hopeful-buckeye-266720)$
```

Problem 2

There are two kind of subscribers - pull and push subscribers. What are the differences between the two and when would you prefer one over the other?

The choice of push/pull really depends on the situation we're dealing with.

- If everything can be managed from one location, it'd make sense to go for push as it'd be easier to apply changes to distribution agents then.
- Pull is also better when the agent downloads the data instead of sending it.
- If a subscriber is really busy or is located in some remote location, it'd be difficult to load data and this is disadvantageous in case of pull.
- The data flows from distributor to subscriber and since the agents use subscribers, pull is preferred when multiple agents are coming from one distributor.

So, the preference actually depends on the number and type of distribution agents and how fast the connectivity is to the subscribers.