**Documentation Assignment-3**

**Description of functions used in the python file:**

1. **Uploading file to bucket**

#Credentials to access the AWS S3 bucket

AWS\_ACCESS\_KEY\_ID = 'AKIAIVPX4M6XPLQXKKIQ'

AWS\_SECRET\_ACCESS\_KEY = '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'

#accessing the bucket

my\_bucket = AWS\_ACCESS\_KEY\_ID.lower() + 'cloud93'

#Connect the bucket to upload the file

try:

conn = boto.connect\_s3(AWS\_ACCESS\_KEY\_ID,

AWS\_SECRET\_ACCESS\_KEY)

print "Connection successful"

except:

print "Connection failed"

#Calculate time taken to copy the fie to bucket

time1 = default\_timer();

s3=boto.connect\_s3()

bucket = conn.get\_bucket("cloud93",validate=False)

key=bucket.new\_key("UNPrecip.csv")

myFfile = "UNPrecip.csv"

print 'Copying %s file to Cloud93 bucket %s' % \

(myFfile, my\_bucket)

#Define percentage of completion

def percent\_cb(complete, total):

sys.stdout.write('.')

sys.stdout.flush()

conn.commit()

key.set\_contents\_from\_filename(myFfile,policy='public-read')

time\_taken=default\_timer() - time1

print 'Time taken to copyfile to bucket is %f' %(time\_taken)

**Description**: In the above snippet we ate copying the file and uploading the file to AWS S3 storage using “**set\_contents\_from\_filename** “ function. Time taken is instrumented and displayed.

1. **Creating table on RDS to insert data.**

#Credentials to access the bucket

AWS\_ACCESS\_KEY\_ID='AKIAIVPX4M6XPLQXKKIQ'

AWS\_SECRET\_ACCESS\_KEY = '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'

#Connect to the bucket to create table

s3 = boto.connect\_s3(AWS\_ACCESS\_KEY\_ID, AWS\_SECRET\_ACCESS\_KEY,is\_secure = False)

bucket = s3.lookup('cloud93')

key = bucket.lookup('UNPrecip.csv')

print key

#Connect to the db

try:

conn=mysql.connect(user='testdb',password='hyd\_1234',

host='testdb.cv74yfhlg7sh.us-west-2.rds.amazonaws.com', database='testdb')

except:

print "Error in connection"

cursor=conn.cursor()

time1 = int(time.time())

#Create a table to upload the all\_month.csv file to RDS

cursor.execute("""create table testdb.UNPrecip2 (Country\_or\_Territory varchar(50),Station\_Name varchar(50),WMO\_Station\_Number varchar(50),Unit varchar(10),Jan float(10,3),Feb float(10,3),Mar float(10,3),Apr float(10,3),May float(10,3),Jun float(10,3),Jul float(10,3),Aug float(10,3),Sep float(10,3),Oct float(10,3),Nov float(10,3),Decm float(10,3));""")

cursor.execute("show tables")

print cursor.fetchall()

time2 = int(time.time())

time = time2 - time1

print 'Time taken to create is '+str(totalTime)+' seconds'

**Description**: In this function we are creating a table in AWS RDS in which we will insert the data read from file on S3. Time taken is instrumented and displayed.

1. **Inserting data into the RDS database:**

#AWS Credentials to establish connection with S3 and RDS

AWS\_ACCESS\_KEY\_ID='AKIAIVPX4M6XPLQXKKIQ'

AWS\_SECRET\_ACCESS\_KEY = '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'

s3 = boto.connect\_s3(AWS\_ACCESS\_KEY\_ID, AWS\_SECRET\_ACCESS\_KEY,is\_secure = False)

#Access the file from the bucket

bucket = s3.lookup('cloud93')

key = bucket.lookup('UNPrecip3.csv')

#Establish connectio with my RDS

conn=mysql.connector.connect(user='testdb',password='hyd\_1234',

host='testdb.cv74yfhlg7sh.us-west-2.rds.amazonaws.com', database='testdb')

print 'Connected to mySQL'

cursor= conn.cursor()

opens=urllib.URLopener()

#reading data to be inserted from the bucket

link="https://s3.amazonaws.com/cloud93/UNPrecip3.csv"

f=opens.open(link)

csvfile= csv.reader(f)

#calculating time taken to insert

time1 = int(time.time())

for row in itertools.islice(csvfile, 1,None):

#executing the insert query

query\_1="""Insert into testdb.UNPrecipnw values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s);"""

cursor.execute(query\_1,row)

conn.commit()

cursor.close()

time2 = int(time.time())

time = endTime - startTime

print 'total time is '+str(time)+' seconds'

**Description:** In the above function snippet we are inserting the data afer reading it from the file on S3 and copying the data into RDS database using **“cursor.execute”** function. Time taken is instrumented and displayed.

1. **Performing 5000,10000 random queries on the data:**

#Connecting to memcache on AWS

memc = memcache.Client(["testcache.t70qbx.cfg.usw2.cache.amazonaws.com:11211"],debug=1);

conn=mysql.connect(user='testdb',password='hyd\_1234',

host='testdb.cv74yfhlg7sh.us-west-2.rds.amazonaws.com', database='testdb')

time1 = time.time()

#Searching Memcache if the query result is already present

testQuery = memc.get('runQueryfor5000')

#If the query is not found

if not testQuery:

cursor = conn.cursor()

lim= range(1,5000)

for count in lim:

query = "SELECT \* FROM earth WHERE mag like '2.%' ;"

num = cursor.execute(query)

result = cursor.fetchall()

for row in result:

print row

#Set the result as a key value pair in Memcache

memc.set('runQueryfor5000',rows,5000)

print "Memcache is updated with new data"

#If data found in memcache get data from it directly

else:

print "Data found in memcache"

for row in testQuery:

print row

print("Loaded from Memcache!")

time2 = time.time()

#Calculating time taken to get or set data from memcache

print("Total Time taken is:")

ftime = ime2-time1

print ftime

#Close connection

conn.close()

**Description:** We are performing 5000 random queries on the dataset present in the RDS. Time taken is instrumented and displayed.

1. **Performing the random queries on memcache:**

#Connecting to memcache on AWS

memc = memcache.Client(["testcache.t70qbx.cfg.usw2.cache.amazonaws.com:11211"],debug=1);

conn=mysql.connect(user='testdb',password='hyd\_1234',

host='testdb.cv74yfhlg7sh.us-west-2.rds.amazonaws.com', database='testdb')

time1 = time.time()

#Searching Memcache if the query result is already present

testQuery = memc.get('runQueryfor200')

#If the query is not found

if not testQuery:

cursor = conn.cursor()

lim= range(1,200)

for count in lim:

query = "SELECT \* FROM earth WHERE mag like '2.%' ;"

num = cursor.execute(query)

result = cursor.fetchall()

for row in result:

print row

#Set the result as a key value pair in Memcache

memc.set('runQueryfor200',rows,200)

print "Memcache is updated with new data"

#If data found in memcache get data from it directly

else:

print "Data found in memcache"

for row in testQuery:

print row

print("Loaded from Memcache!")

time2 = time.time()

#Calculating time taken to get or set data from memcache

print("Total Time taken is:")

ftime = ime2-time1

print ftime

#Close connection

conn.close()

**Description:** The random queries that were performed are now performed on Memcache . If the data is not present in memcache then we “.set” it, else if it is already present then we retrieve the data using “.get”. Time taken is instrumented and displayed.