**Assignment 4 - Introduction to Amazon Web Services (and Databases)**

**Name: Shweta Pathak**

**UTA Id: 1001154572**

**Net Id: ssp4572**

**Section : 1:00 pm to 3:00 pm**

**References:**

1. <http://aws.amazon.com/developers/getting-started/python/>
2. [http://boto.readthedocs.org/en/latest/ - boto interface (python) for AWS](http://aws.amazon.com/developers/getting-started/python/)
3. [http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html](http://aws.amazon.com/developers/getting-started/python/)
4. [http://dev.mysql.com/doc/mysql-ha-scalability/en/ha-memcached-interfaces-python.html](http://aws.amazon.com/developers/getting-started/python/)
5. [http://boto.readthedocs.org/en/latest/rds\_tut.html](http://aws.amazon.com/developers/getting-started/python/)

**Code:**

# import statements

import boto

import csv

import time

import sys

import MySQLdb

import urllib2

import memcache

import hashlib

from boto.s3.key import Key

from boto.s3.connection import S3Connection

from boto.s3.connection import Location

# Upload the file to amazon S3

def put():

# Access keys

AWS\_ACCESS\_KEY\_ID='AKIAJGQE6BZY4X7LYAYA'

AWS\_SECRET\_ACCESS\_KEY='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'

# Establish connection with Amazon S3

conn = S3Connection(AWS\_ACCESS\_KEY\_ID,AWS\_SECRET\_ACCESS\_KEY,validate\_certs=False,is\_secure=False)

bucket\_name = conn.create\_bucket('shwetabucket91')

k = Key(bucket\_name)

k.key = raw\_input("Enter the file name to upload to S3: ")

start\_time = time.clock()

k.set\_contents\_from\_filename(k.key)

end\_time = time.clock()

total\_time = end\_time-start\_time

print(total\_time)

# Inserting data into the Relational database service

def insert\_data():

db = MySQLdb.connect(host= "shweta.ckcwra3d81nf.us-west-2.rds.amazonaws.com",user="root",passwd="root1234",db="earthquakes\_database")

# Access keys

AWS\_ACCESS\_KEY\_ID='AKIAJGQE6BZY4X7LYAYA'

AWS\_SECRET\_ACCESS\_KEY='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'

# Establish connection with Amazon S3

conn = S3Connection(AWS\_ACCESS\_KEY\_ID,AWS\_SECRET\_ACCESS\_KEY,validate\_certs=False,is\_secure=False)

bucket\_name = conn.get\_bucket('shwetabucket91')

k = Key(bucket\_name)

url = 'https://s3.amazonaws.com/shwetabucket91/Inpatient\_data.csv'

response = urllib2.urlopen(url)

csv\_data = csv.reader(response)

cursor = db.cursor()

cursor.execute("drop table inpatient\_data")

cursor.execute("create table inpatient\_data(DRG\_Definition varchar(255),Provider\_Id varchar(255),Provider\_Name varchar(255),Address varchar(255),City varchar(255),State varchar(255),Zip varchar(255),Region varchar(255),Total\_discharge varchar(255),Average\_Covered\_Charges varchar(255),Average\_Total\_Payments varchar(255),Average\_Medicare\_Payments varchar(255))")

start\_time = time.clock()

count = 0

for row in csv\_data:

count += 1

if count <> 0:

cursor.execute("INSERT INTO inpatient\_data(DRG\_Definition,Provider\_Id,Provider\_Name,Address,City,State,Zip,Region,Total\_discharge,Average\_Covered\_Charges,Average\_Total\_Payments,Average\_Medicare\_Payments) VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)", row)

print count

db.commit()

end\_time = time.clock()

total\_time = end\_time - start\_time

print ("Total time taken to insert data into RDS : ")

print total\_time

db.commit()

# Listing all the buckets available

def list\_objects():

# Access keys

AWS\_ACCESS\_KEY\_ID='AKIAJGQE6BZY4X7LYAYA'

AWS\_SECRET\_ACCESS\_KEY='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'

# Establish connection with Amazon S3

conn = S3Connection(AWS\_ACCESS\_KEY\_ID,AWS\_SECRET\_ACCESS\_KEY,validate\_certs=False,is\_secure=False)

bucket\_name = conn.get\_bucket('shwetabucket91')

k = Key(bucket\_name)

bucket\_list = conn.get\_all\_buckets()

print(len(bucket\_list))

for rs in bucket\_list:

print rs.name

# get all contents from the file

def list\_files():

# Access keys

AWS\_ACCESS\_KEY\_ID='AKIAJGQE6BZY4X7LYAYA'

AWS\_SECRET\_ACCESS\_KEY='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'

# Establish connection with Amazon S3

conn = S3Connection(AWS\_ACCESS\_KEY\_ID,AWS\_SECRET\_ACCESS\_KEY,validate\_certs=False,is\_secure=False)

bucket\_name = conn.get\_bucket('shwetabucket91')

k = Key(bucket\_name)

for key in bucket\_name.list():

print "{name}\t{size}\t{modified}".format(

name = key.name,

size = key.size,

modified = key.last\_modified,

)

# Perform 1 , 5 and 20 thousand random queries

def queries():

# Establish connection with MySQLdb

db = MySQLdb.connect(host= "shweta.ckcwra3d81nf.us-west-2.rds.amazonaws.com",user="root",passwd="root1234",db="earthquakes\_database")

cursor = db.cursor()

query = "select Provider\_Id from inpatient\_data order by "

print " \nOne thousand random queries :"

start\_time = time.clock()

for row in range(1,1001):

cursor.execute(query)

elapsed = (time.clock()-start\_time)

print "time taken to load", elapsed

query = "select Provider\_Id from inpatient\_data order by rand()"

print " \nFive thousand random queries :"

start\_time1 = time.clock()

for tuple in range(1,5001):

cursor.execute(query)

elapsed1 = (time.clock()-start\_time1)

print "time taken to load", elapsed1

query = "select Provider\_Id from inpatient\_data order by rand()"

print " \nTwenty thousand random queries :"

start\_time2 = time.clock()

for record in range(1,20001):

cursor.execute(query)

elapsed2 = (time.clock()-start\_time1)

print "time taken to load", elapsed2

db.commit()

print 'done'

def query\_tuples():

# Establish connection with MySQLdb

db = MySQLdb.connect(host= "shweta.ckcwra3d81nf.us-west-2.rds.amazonaws.com",user="root",passwd="root1234",db="earthquakes\_database")

cursor = db.cursor()

start\_time = time.clock()

print "Random queries on retrieved 200 to 800 tuples"

cursor.execute("select Provider\_Id from inpatient\_data limit 199,600")

fetch\_data = cursor.fetchall()

for row in fetch\_data:

for i in range(1,1001):

cursor.execute("select Provider\_Name from inpatient\_data where Provider\_Id = %s order by rand()", row)

elapsed = (time.clock()-start\_time)

print "time taken to load", elapsed

def mem\_query():

# Establish connection with MySQLdb

db = MySQLdb.connect(host= "shweta.ckcwra3d81nf.us-west-2.rds.amazonaws.com",user="root",passwd="root1234",db="earthquakes\_database")

memClient = memcache.Client(['shweta.rviwnp.0001.usw2.cache.amazonaws.com:11211'],debug=0)

query = "select Provider\_Id from inpatient\_data order by rand()"

hash\_key = hashlib.md5()

hash\_key.update(query)

key = hash\_key.hexdigest()

print key

cursor = db.cursor()

start\_time = time.clock()

if memClient.get(key): # If data already exists

print " Got data"

else:

cursor.execute('select Provider\_Id from inpatient\_data order by rand()')

rows = cursor.fetchall()

memClient.set(key,rows)

print "Not found"

end\_time = time.clock()

total\_time = end\_time - start\_time

print " Total time taken by memcache :"

print(total\_time)

def main():

options\_toselect = {1: put, 2: insert\_data, 3:list\_objects, 4:list\_files, 5: queries, 6:query\_tuples,7: mem\_query}

while(True):

print "\n1. Upload file on Amazon Cloud S3. \n"

print "2. Insert data into Amazon Relational Data Service. \n"

print "3. List of buckets on Amazon S3. \n"

print "4. List the files in a bucket. \n"

print "5. Time taken to execute 1 thousand , 5 thousand and 20 thousand Random queries.\n"

print "6. Random Query for 200 to 800 tuples. \n"

print "7. Elastic cache queries."

print "8. Exit \n"

option = raw\_input("Select one option : ")

if option =="1":

options\_toselect[1]()

elif option =="2":

options\_toselect[2]()

elif option =="3":

options\_toselect[3]()

elif option =="4":

options\_toselect[4]()

elif option =="5":

options\_toselect[5]()

elif option =="6":

options\_toselect[6]()

elif option =="7":

options\_toselect[7]()

elif option =="8":

sys.exit(0)

else:

print "Please select a valid choice !!!\n"

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Output:**

**Time taken to upload file on amazon S3:** 0.088118 seconds

**Time taken to insert data into database:** 12.175687 seconds

**Time taken to execute 1000 random queries:** 27.548758 seconds

**Time taken to execute 5000 random queries**: 145.914424 seconds

**Time taken to execute 20000 random queries**: 690.87653 seconds

**Time taken to execute random queries on 200 to 800 tuples:** 200.87722 seconds

**Time taken to execute random queries with memcache**:

* **If data not found :** 0.032535 seconds
* **If data found :** 0.042086 seconds

**Explanation:** Time taken to execute the random queries using memcache is very fast as compared to normal execution