–---------------------------------------------------------------------------------------------------------------------------

Times

–---------------------------------------------------------------------------------------------------------------------------

File upload to S3

Upload time to s3 : 2.2887141704559326

File download from S3

Download time from S3 : 1.6205859184265137

Upload to RDS

Load time to RDS : 0.6053738594055176

Runtime for 2000 random queries

Time for 2000 random queries : 126.56824803352356

Runtime for 2000 limited scope (only limited set of tuples)

Time for limited queries : 95.7012619972229

Runtime for 2000 random queries with memcache:

Time for 2000 random queries using memcache : 0.3588829040527344

Runtime for 2000 random limited queries with memcache:

Time for 2000 random queries limited using memcache : 0.26987409591674805

–---------------------------------------------------------------------------------------------------------------------------

Following is the procedure to connect to S3, RDS, EC2 and run the program.

–---------------------------------------------------------------------------------------------------------------------------

sudo apt-get install mysql-dev libmysqlclient-dev python-pip python-dev mysql-server awscli

sudo pip install python-mysql MySQL-python python-memcached boto

–---------------------------------------------------------------------------------------------------------------------------

EC2

–---------------------------------------------------------------------------------------------------------------------------

user - username

password - password

sudo apt-get install awscli

AWS Access Key ID [None]: aws-access-key

AWS Secret Access Key [None]: aws-secret-key

Default region name [None]: region-name

Default output format [None]: json

aws s3 cp s3://bucketname/file

>mysql -u root -p

> load data

local infile <filename>

into table <table name>

fields terminated by ','

enclosed by '\"' lines

terminated by '\n'

ignore 1 lines;

–---------------------------------------------------------------------------------------------------------------------------

RDS

----------------------------------------------------------------------------------------------------------------------------

user - username

password – password

----------------------------------------------------------------------------------------------------------------------------

Python Code

----------------------------------------------------------------------------------------------------------------------------

'''

Created on Feb 28, 2015

@author: puneeth

'''

import os, time, MySQLdb, memcache

def s3\_stuff():

start = time.time()

os.system("aws s3 cp ./csvfiles/all\_month.csv s3://<bucket\_name>/all\_month.csv --acl public-read")

end = time.time()

runtime = end - start

print("Upload time to s3", runtime)

start = time.time()

os.system("aws s3 cp s3://<bucket\_name>/all\_month.csv all\_month.csv")

end = time.time()

s3\_time = end-start

print("Download time from S3", s3\_time)

def mysql\_stuff():

print('Connecting to RDS...')

conn = MySQLdb.connect('usaeq.cavhaazwjuoq.us-east-1.rds.amazonaws.com',

'username',

'password',

local\_infile=1)

print('Creating cursor...')

cursor = conn.cursor()

print('Cursor use')

cursor.execute('use usaeq')

print('Cursor drop')

cursor.execute("drop table if exists usaeq")

print('Cursor create')

create\_table\_string = """create table usaeq(

eq\_time timestamp,

eq\_latitude double,

eq\_longitude double,

eq\_depth double,

eq\_mag double,

eq\_magType varchar(20),

eq\_nst double,

eq\_gap double,

eq\_dmin double,

eq\_rms double,

eq\_net varchar(100),

eq\_id varchar(20),

eq\_updated timestamp,

eq\_place varchar(100),

eq\_type varchar(20)

);"""

print('Cursor execute')

cursor.execute(create\_table\_string)

start = time.time()

print('Loading data into RDS..')

cursor.execute("""load data local infile 'all\_month.csv'

into table usaeq

fields terminated by ','

enclosed by '\"'

lines terminated by '\n'

ignore 1 lines;""")

end = time.time()

mysql\_time = end - start

print("Load time to RDS", mysql\_time)

print('Cursor commit')

cursor.execute('commit')

print('Cursor select')

cursor.execute("select count(1) from usaeq")

rows\_loaded =cursor.fetchall()

print(rows\_loaded)

print('Time magnitude')

query = """select week as 'Week |', count(mag2) as '2 - 2.99 |', count(mag3) as '3 - 3.99 |', count(mag4) as '4 - 4.99 |', count(mag5) as '>= 5 |'

from

((select

case

when date(eq\_time) between cast('2015-01-20' as date) and cast('2015-01-26' as date) then 1

when date(eq\_time) between cast('2015-01-27' as date) and cast('2015-02-02' as date) then 2

when date(eq\_time) between cast('2015-02-03' as date) and cast('2015-02-09' as date) then 3

when date(eq\_time) between cast('2015-02-10' as date) and cast('2015-02-16' as date) then 4

when date(eq\_time) between cast('2015-02-17' as date) and cast('2015-02-23' as date) then 5

when date(eq\_time) between cast('2015-02-24' as date) and cast('2015-02-29' as date) then 6

end week,

usaeq.eq\_id

from

usaeq) as week, (select

case

when eq\_mag between 2 and 2.99 then eq\_mag

end mag2,

usaeq.eq\_id

from

usaeq) as mag2, (select

case

when eq\_mag between 3 and 3.99 then eq\_mag

end mag3,

usaeq.eq\_id

from

usaeq) as mag3, (select

case

when eq\_mag between 4 and 4.99 then eq\_mag

end mag4,

usaeq.eq\_id

from

usaeq) as mag4, (select

case

when eq\_mag >= 5 then eq\_mag

end mag5,

usaeq.eq\_id

from

usaeq) as mag5)

where

week.eq\_id = mag2.eq\_id and

week.eq\_id = mag3.eq\_id and

week.eq\_id = mag4.eq\_id and

week.eq\_id = mag5.eq\_id

group by week;"""

start = time.time()

cursor.execute(query)

end = time.time()

time\_mag\_time = end - start

print('Time magnitude query', time\_mag\_time)

rows = cursor.fetchall()

desc = cursor.description

print('---------------------------------------------------------------')

print('Time | \t Magnitude \t \t \t \t \t |')

print('---------------------------------------------------------------')

print("%s \t %s \t %s \t %s \t %s \t " % (desc[0][0], desc[1][0], desc[2][0], desc[3][0], desc[4][0]))

for row in rows:

print('---------------------------------------------------------------')

print("%s | \t %s \t | \t %s \t | \t %s \t | \t %s | " % row)

print('---------------------------------------------------------------')

print('2000 Random queries')

start = time.time()

query = 'select eq\_time from usaeq group by rand() limit 10'

for i in range(1, 2000):

cursor.execute(query)

end = time.time()

rand\_time = end - start

print('Time for 2000 random queries', rand\_time)

print('Limited queries')

start = time.time()

restricted\_query = 'select \* from (select eq\_id from usaeq limit 2000) A group by rand() limit 10'

for i in range(1,2000):

cursor.execute(restricted\_query)

end = time.time()

lim\_time = end - start

print('Time for limited queries', lim\_time)

cursor.close()

conn.close()

def memcache\_stuff():

memc = memcache.Client(['127.0.0.1:11211'], debug=1);

try:

conn = MySQLdb.connect('usaeq.cavhaazwjuoq.us-east-1.rds.amazonaws.com',

'username',

'password',

'usaeq')

except MySQLdb.Error, e:

print "Error %d: %s" % (e.args[0], e.args[1])

sys.exit (1)

count = 0

start = time.time()

for i in range(1,2000):

usaeq = memc.get('usaeq')

if not usaeq:

cursor = conn.cursor()

cursor.execute('select \* from usaeq order by eq\_time limit 5')

rows = cursor.fetchall()

memc.set('usaeq',rows,3600)

print "Updated memcached with MySQL data"

else:

# print "Loaded data from memcached", i

count = count + 1

end = time.time()

print(count)

rand\_time = end - start

print("Time for 2000 random queries using memcache", rand\_time)

count = 0

start = time.time()

for i in range(1,2000):

usaeqrd = memc.get('usaeqrd')

if not usaeqrd:

cursor = conn.cursor()

cursor.execute('select \* from (select eq\_time from usaeq limit 2000) A group by rand() limit 10')

rows = cursor.fetchall()

memc.set('usaeqrd', rows, 3600)

print "Updated memcached with MySQL data for limited scope"

else:

# print "Loaded data from memcached limited scope", i

count = count + 1

end = time.time()

print(count)

limrand\_time = end - start

print("Time for 2000 random limited queries using memcache", limrand\_time)

s3\_stuff()

mysql\_stuff()

memcache\_stuff()

----------------------------------------------------------------------------------------------------------------------------

References

----------------------------------------------------------------------------------------------------------------------------

<https://pypi.python.org/pypi/MySQL-python>

<http://stackoverflow.com/questions/25865270/how-to-install-python-mysqldb-module-using-pip>

<http://dev.mysql.com/doc/mysql-ha-scalability/en/ha-memcached-interfaces-python.html>

<http://docs.aws.amazon.com/AWSToolkitEclipse/latest/GettingStartedGuide/tke_setup_install.html>

http://boto.readthedocs.org/en/latest/getting\_started.html