Assignment 4

**Team Members:**

Abhitej Date (1001113870)

Sagar Lakhia (1001123182)

**How to run:**

Run this code on EC2 instance:

1. **Get access to AWS (Amazon Cloud).**

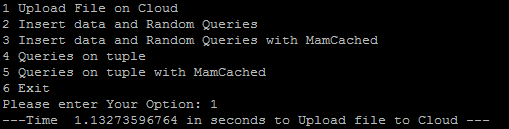
We were able to get access to Amazon Cloud and we were able to learn about different services Amazon provides.

1. **Find a large dataset (50K tuples or larger).**

We got our data from <https://www.data.gov/> . The file is about Consumer Complaints. We have attached the file in the compressed folder we provided.  
There are round about 54k tuples in it.

1. **Copy that file to AWS, and time (instrument) how much time it takes.**

We were successful in copying the file from the link to S3 and then to the RDS database. We noted the time it took to upload to S3. The timings are mentioned below in the screenshot attached:



1. **Put the data into a Relational DB (time).**

We used RDS to store our data by reading the csv file from S3 bucket and storing everything to the database. The time it took to insert all the 54k tuples in the database is given in the screenshot below:

C:\Users\Abhitej\Desktop\final\assignment4\7.PNG

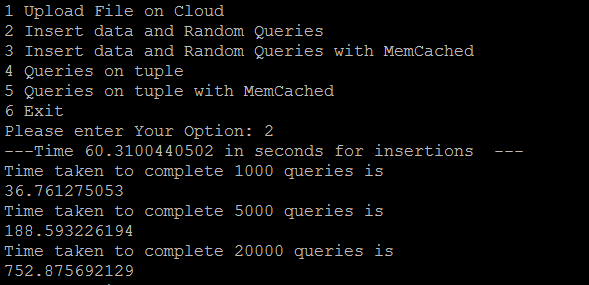
**5.** **Write the code to do one thousand, 5 thousand and 20 thousand random**

**(small) queries (time).**

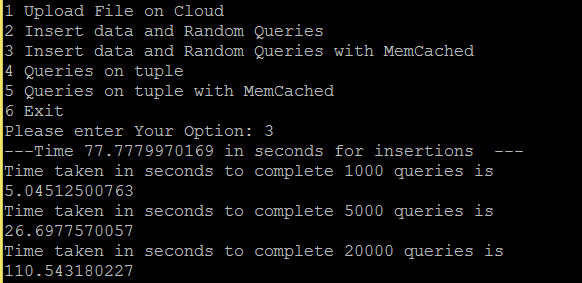
We wrote our code to perform few simple queries like Select, Distinct, Count, etc. and we looped it around for 1000, 5000 and 20,000 times (Selecting randomly each query using randint in python).

We calculated the time it took to run those queries and the results are attached in the screenshot below:

Without Memcached



With Memcached

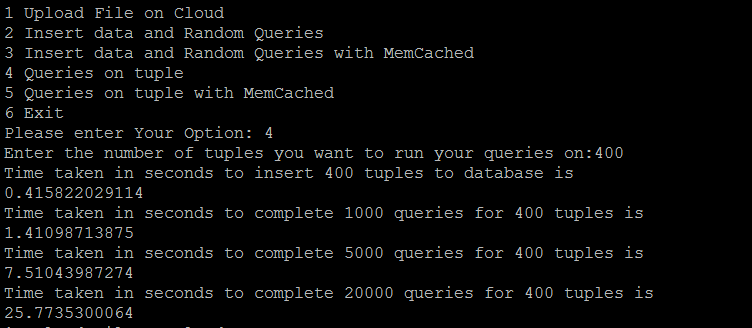


1. **Repeat using queries of only 200 to 800 tuples.**

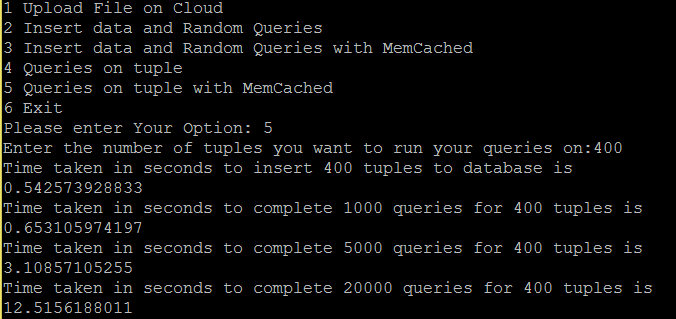
In this scenario, user will be able to select the number of tuples and we performed all the previous functions that we did for the whole dataset.

Results are in the screenshots below:

WithOut MemCached:



With MemCached:



1. **Repeat previous two steps using “Elastic” Cache (Memcache, etc.)**

Now Memcache makes our interaction with database to minimal. It saves the data to the cache server and displays it, avoiding database interactions.

There were drastic change in the time taken for processing huge number of queries. Results of those are in the screenshots below:

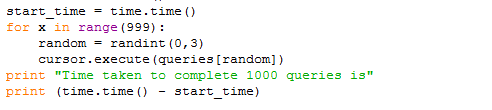
We used EC2 instance(Free tier 1 Gig Ram with Amazon AMI) to run our program.  
We created a RDS instance which interacts with S3. And with the help of Memcache we were able to generate quick results.

**Code Snippet:**

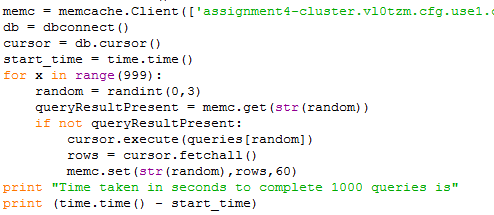
Code to calculate time taken to upload file:



Code to calculate time taken to perform 1000 queries:



Code to calculate time taken to perform 1000 queries using Memcached



**References:**

**1]** [**http://dev.mysql.com/doc/mysql-ha-scalability/en/ha-memcached-interfaces-python.html**](http://dev.mysql.com/doc/mysql-ha-scalability/en/ha-memcached-interfaces-python.html)

**2] StackOverFlow for various Questions**

**3]**[**http://docs.aws.amazon.com/AmazonElastiCache/latest/UserGuide/GettingStarted.CreateCluster.html**](http://docs.aws.amazon.com/AmazonElastiCache/latest/UserGuide/GettingStarted.CreateCluster.html)