**ECE 1779 Project 1 Documentation**

**Group 3**

**Project Abstract (Edit Later)**

In this project, we will continue to develop the application of Project1. We implemented a manager app which can control the size of the memcache pool on demand. The resizing of memcache pool will be based on user input configuring with manual mode and auto‐scaling mode. More details will be presented in the next section.

**Application Components and Functions**

1.) Manager-app: controls the size of the memcache pool

A.) Memcache pool statistics chart: a chart that shows the number of nodes, miss rate, hit rate, number of items in cache, and number of requests and the chart will display the data for the last 30 minutes with 1 minute granularity.

B.) Configure Memcache attribute: User can choose the capacity of the cache and the replacement policy and it will be applied to all memcache nodes in the pool.

C.) Memcache pool resizing

1. Manual mode: Allow user to add or shrink the memcache pool manually by one with the minimum of 1 and maximum of 8.

2. Automatic mode: The Auto scaling policy configuration are based on four configuration parameters

|  |  |
| --- | --- |
| Attributes |  |
| Max Miss Rate threshold | Average for all nodes in the pool over the past 1 minute for expand |
| Min Miss Rate threshold | Average for all nodes in the pool over the past 1 minute for shrinking the pool |
| Ratio by which to expand the pool | Expand the memcache pool by ratio that getting from max miss rate. |
| Ratio by which to shrink the pool | Shirk the memcache pool by ratio that getting from min miss rate. |

D.) Delete Data: Delete all image in S3 and RDS. Also clear all data in memcache.

E.) Clearing Memcache Data: Clear the data in memcache.

**2.) Auto-Scaler Component:**

Autoscaling can ensure the application has enough capacity to handle the workload and control the usage of EC2 to lower the cost by enabling instances when needed and stopping them if not needed.

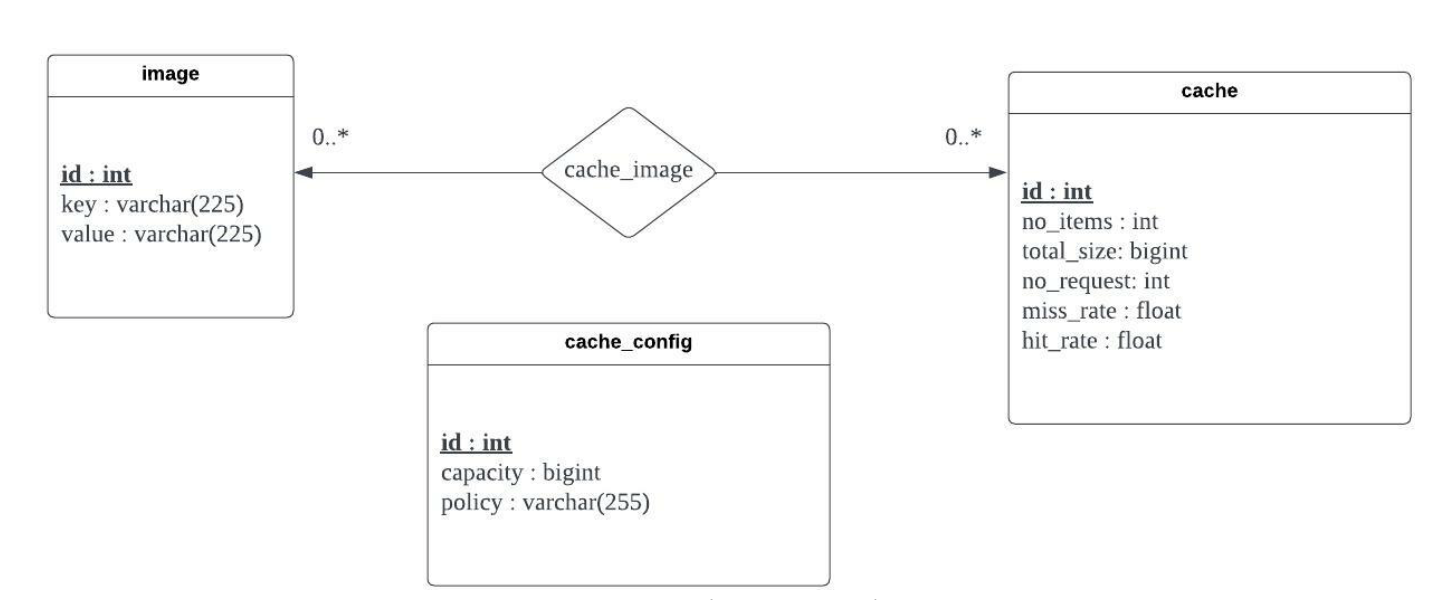
By getting the min/max missing rate of the cache pool by using AWS CloudWatch, the application will resize the memcache pool automatically with the missing rate. expand the memcache pool by max miss rate or shirk the memcache pool by the min miss rate. The cache miss rate will be updated every minute and the total pool size of the memcache pool should be within 1 to 8.

**3.) Front End:**

1. Upload: All image files should be stored in S3. The key and the value will be stored in AWS RDS
2. MD5 Hashes:

**Database Schema:**

Database Schema as the ER diagram shows, there are four tables needed in the database schema. Image, Cache, cache\_image and cache\_config. The cache\_config table stores the information of cache configuration information including the id, capacity, and policy of the cache. Therefore, the cache\_config table does not have any relationship with other tables. The image table and cache table store the information of the image and cache separately. Besides, the cache\_image is the relationship set between the image and cache sets. This relationship is set up just in case there are multiple cache instances.



**Design Decisions**:

S3: All the Image will be stored at S3.

Cloud Watch: Check the current status to increase or decrease the number of memcache size automatically.

**Database:** In this project we are using AWS

**Results**:

**Graphs for Auto-Scalar:**