CS 553
CLOUD COMPUTING
PROGRAMMING ASSIGNMENT-3
CloudKon clone with Amazon EC2, S3, SQS, and DynamoDB

Design Document

SUBMITTED BY :
SACHIN KRISHNA MURTHY
CWID : A20354077

Design Document

- **Objective :** The main objective of the assignment is to implement a distributed task framework cloudkon on Amazon EC2 using SQS. This framework is divided in to two components :
- Client
- Worker

  The following tasks has been included:
- Client
- Local back-end worker
- Remote back-end worker
- Duplicate tasks.

- **Client :**

The client is the communication protocol between client and the SQS. The client is a command line tool which can submit tasks to the SQS.

Client –s Qname –w <WORKLOAD_FILE>

- **Workers :**

The workers is of two types :

- Local Back-end workers
- Remote back-endWorkers

- Local Back-endWorkers :

  Here we are running a large number of sleep tasks with different number of threads. The workload file consists of all the sleep tasks that the worker needs to execute. For local back-end workers, we have two types of queue created : Task queue and Result queue. The client file puts the sleep task into task queue based on the number of times the user wants to execute the task.

  Then, each worker will read the tasks from the task queue line by line and executes them. Finally after execution of the tasks the results are put into another queue called result queue. Also the value 0 is returned in order to indicate the successful completion of the task. The output also gives us the details about the sleep task executed and the id’s.

Here we can say that the Task queue and Result queue will act as interface between client and the worker. Also the client and the local back-end workers should run on the same virtual machine.

The local workers are started by using the following command :

```
Client –s LOCAL –t N –w <workload_file>
```

Where, N is the number of threads and –w is the number of times the tasks needs to be executed.

- **Remote Back-end workers:**

The working functionality of local back-end workers and remote back-end workers is the same. But the only difference here is remote workers run on different virtual machines than the client and other workers.

Here we have used SQS and the Dynamo DB service in order to communicate between client and the workers. The tasks are written into the SQS queue and it executed by back-end workers using different number of instances from 1 to 16.

The remote workers are started by using the following command:

```
worker –s QNAME –t N
```

➢ **Duplicate Tasks:**

In order to handle the duplicate tasks, we have used the id field for each message through which we can keep a track of the message delivered for local back-end workers.

For remote back-end workers, since it is not possible to handle duplicate tasks using SQS service, we have used Dynamo Db for this purpose.