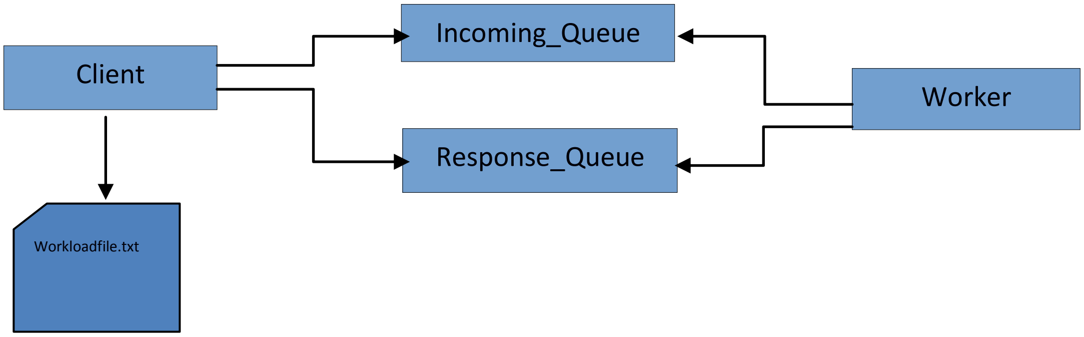
# CS 553 Cloud Computing
# Programming Assignment 3
# SujayGunjal (CWID: A20351746)
# Design

## 1) Local program design:-



1) In case of local execution Workers will start first and will continuously check Incoming_Queue for any job as soon as workers get job they will start executing it.

2) Client will read Workload.txt file and store jobs in Incoming_Queue.

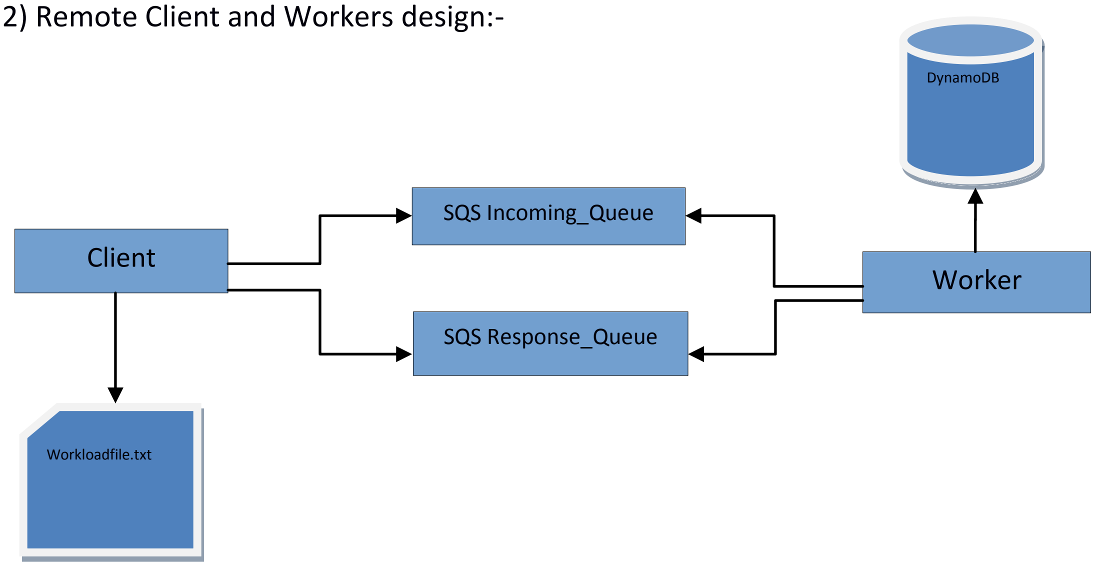
3) I have used Blocking_Queue for storing jobs and responses.

4) Workers will store 0 (success )or 1 (Failure) with ID in Response_Queue.

5) Clients other thread will continuously check for any failure and print count of success and failure on screen.

6) There will be one client and user can create any number of workers by varying threads.

2) Remote Client and Workers design:-



1) In case of Remote execution Workers will start first and will continuously check SQS Incoming_Queue for any job as soon as workers get job they will start executing it.

2) Client will read Workload.txt file and store jobs in Incoming_Queue.

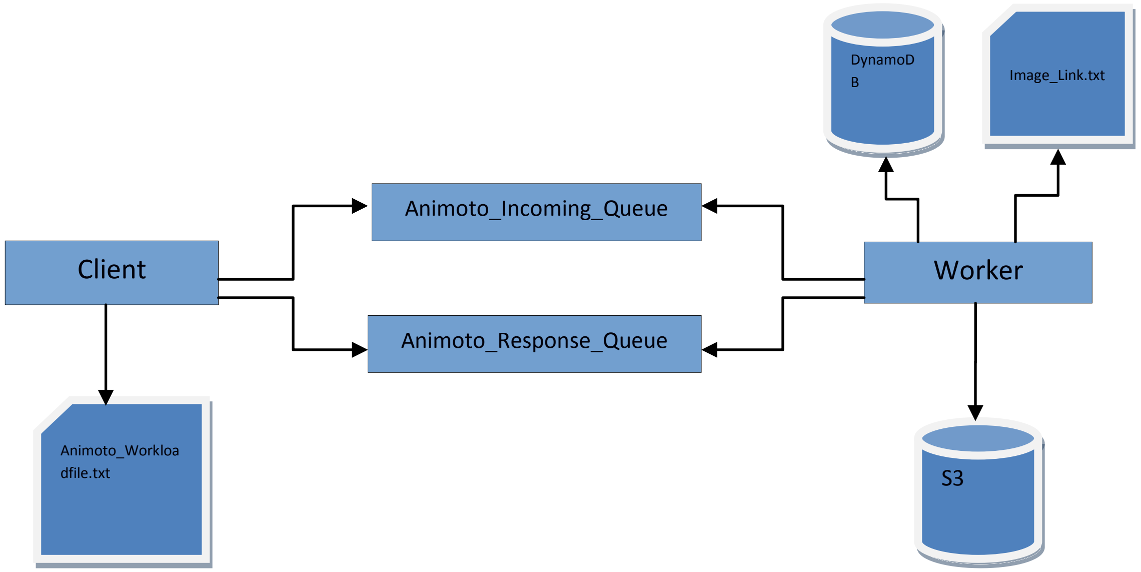
3) Once worker thread gets job it will first check job Id in DynamoDB for duplication. If it finds Id in DynamoDBthen worker thread will ignore that and will pick next job for execution.

4) If worker thread does not find ID in DynamoDB then it will insert the task ID inDynamodDB. Once job gets completed It will make the entry in Response_Queue with 0 (success)or 1 (Failure).

5) Clients other thread will check for any failure and print the time of execution screen.

6) There will be one Client and user can create any number of workers with varying threads.

3) Animoto Client and Workers design:-



1) In case of Animoto execution Workers will start first and will continuously check SQS Animoto_Incoming_Queue for any job as soon as workers get job they will start executing it.

2) Client will read Workload.txt file which has path of files which has list of 60 images and will store jobs in Animoto_Incoming_Queue.

3) All soon as workers get jobs in queue it will first check DynamoDB for duplicate task if any other thread already executing it then it will be ignored.

4) If no one has executed it yet then it will start reading file which has list of 60 images and will download one by one with wget command and will store it on local disk. Images will be renamed in sequence like Img_1.jpg, Img_2.jpg etc.

5) Once all images gets downloaded commandffmpeg -f image2 -i img%d.jpg job_1.mpg will be used to create video.

6) Once video is ready it will be upload it to s3 and will put link to the video in response queue.

7) Client thread will continuously check for number of link we get in Animoto_Response_Queue it should match with number of task submitted.

8) There will be one Client and user can create any number of workers with varying threads.