CS553 Cloud Computing

Programming Assignment #3

Design

Pranitha Nagavelli(A20345406)

Following are the important modules that are implemented in this assignment:

- Client
- Local Worker
- Remote Worker
- Simple Queue Service
- DynamoDB
- Animoto

This assignment gives an overview of Amazon EC2cloud, Simple queuing service, S3 and DynamoDB.

- The framework should be separated three components, one client, Workers and Queues.
- Here client reads the task from the file and insert into the local queue or SQS depending on the user input.
- Then the workers get the tasks from this queues and perform the tasks with given number of threads.
- The other part of the assignment being Animoto where the file with url's are given to the client which gets the images using wget command and makes them into a video ffmpeg tool this url is then stored into a bucket of S3.

Client:

- Client is a command line tool that reads input from the Given input file and passes the tasks into the local queue f the operation mode is 'lw' or into SQS Queue if the operation mode is 'rw' and if the operation mode is 'animoto' the url's are sent into SQS.
- The Client interface would start on following Command:
  Client <Threads><Workload_File><OperationMode> <# of workers>
  Threads: Number of threads that are to used.
  WorkloadFile:Location of the file that contains tasks
  Operation Mode: lw for local worker, rw for remote worker, animoto
  # of workers:workers being used.(Accordingly tasks are inserted)

Local Worker:

- As the framework will only support sleep tasks, and therefore you can in run a large number of sleep tasks on the same in-memory queue, tasks are very light-weigth and doesn't require any large amount of data.
- Each task of particular same sleep time present in queue are runtogether and the result after execution of queue is stored in another queue result, and also the execution time for each task is displayed.
- Local worker in turn uses a method localThread which helps is executin different number of threads.

- Local worker is called from the client passing the queue and Number of threads as input.

Remote Worker:

- When the operation mode in the client is ‘rw’ the tasks which are also sleep tasks are inserted or sent into a AWS simpleQueueService.
- Now, the remote worker will take the tasks present in this SQS and performs the tasks deletes the task from Queue after completion sends the confirmation into another SQS named result.
- Remote Worker lso implements thread using remoteThread method and also before the execution of the tasks in the SQS, the tasks are sent into AmazonDynamoDBSample methos it checks for any duplicate tasks and also inserts the unique values into a table named ‘duplicate’.
- The remoteWorker interface would start on following Command:
  remoteWorker <Threads>
  Threads: Number of threads

Animoto

- When the operation mode in the client is ‘animoto’ ,here client will send the urls of the image into a SQS.
- Now Animoto using Animoto thread will read each url download it using wget command
- Now using ffmpeg we write a command in such a way that all urls downloaded are made it into video and store it as jpeg
- Now this video is sent into the S3 bucket, and the url is sent to the client.

SimpleQueueServiceSample.java is a sample api from aws.amazon.com. It is queue functionality that receives the tasks from the scheduler and clients receive the task from it. The tasks are pushed and pulled from the scheduler as and when needed.

DynamoDB.java has a hashmap to store key/value pairs Task and its ID. Worker checks the dynamodb table for entry of tasks. If the task is present then it is dropped hence the duplication is avoided.