

CloudKon clone with Amazon EC2, S3, SQS, and DynamoDB

i. Design

In this assignment the distributed task distribution framework which is being implemented is divided into two parts

Client

Workers

The communication between the Client and the Worker is implemented two ways i.e. **Local** and **Remote**.

Local Task Execution

For the local implementation, client is run with the interface client -s LOCAL -t N -w <Path-to-WorkloadFile> which checks for the number of arguments and validates the arguments.

Since the argument 2 is LOCAL the code executes the local implementation logic.

Local Implementation Logic

- A queue is created.
- Client reads the workload file passed through the argument 6 which is the path to the file.
- Client writes the values read to the queue by writing only the sleep time mentioned in the file by splitting content by space.
- The sleep time in the queue is accessed by the Local Worker which in case of LOCAL is the same system and hence representing the threads in the Thread Pool.
- Once the threadpool is created the workers execute the task mentioned in the workload file using the executor API of java.
- The time taken from creation of workers (Threadpool) till after execution of the task is measured.

Remote Task Execution

Here the worker interfaces and client interface are different as the client and the worker runs on different VMs.

To run the remote workers, the interface used is

Worker -s QNAME -t n

The interface to run the remote client is

Client -s QNAME -w <Path-to-WorkloadFile>

The arguments are validated before running the remote client or the remote worker.

Remote Logic Implementation

Remote Client :

- The remote client just adds task to the Amazon's SQS and wait for completion of the execution of the tasks by the workers.
- SQS provide APIs for common developers to explore, and in this assignment the Queue is created using them.
- The message read from the file is sent to the SQS queue using the queue URL obtained after creation of the queue.

Remote Worker :

- Worker is started on separate machine and connected using the interface.
- The remote worker receives message from the created source queue (Client) using URL of the same.
- Calls the function which executes the task on the queue as a thread in the pool but since the message received is one at the time the worker executes only one task at the time.
- Time taken to receive the message and execution is calculated.

ii. Manual

The SourceCode folder contains following java files.

Framework.java

RClient.java

RWorkers.java

RemoteWorkers.java

RQueueTask.java

The Framework.java is the main command line interface implementation code where the client and the workers are started as per the command line arguments specified.

The executable jar (Example - Framework.jar) must be created in order to run the framework and the jar must be moved to the folder where the files are present in the EC2 instance by using the **cd** commands. Before running the Framework.jar the .aws folder must be moved to the folder where the programs are run.

For Local Task Execution Implementation

Run the executable jar by giving the following command example

COMMAND : Java -jar Framework.jar client -s LOCAL -t n -w <path -to-the-workloadfile>

Where n -> 0 to 16 in powers of 2 and < > must contain the path to the workload file.

For Remote Task Execution Implementation

Run the executable jar by giving the following command example

Step 1: Run the client

COMMAND : `Java -jar Framework.jar client -s SQSqueue -w <path -to-the-workloadfile>`

Where < > must contain the path to the workload file and SQSqueue is the source queue name. Must be the name of the queue which is created in SQS.

Step 2 : Run the workers

Example command

COMMAND : `Java -jar Framework.jar worker -s SQSqueue -t n`

Where n -> 0 to 16 in powers of 2 and SQSqueue is the source queue name. Must be the name of the queue which is created in SQS.

The execution of remote workloads needs to be done with the .aws folder where the jar being executed is. The .aws folder gives the credentials to connect to the AWS account.

For the performance analysis

The throughput and efficiency for the local worker execution and remote worker execution is calculated as given below

Throughput = $\text{Number of Tasks} / \text{Time Taken to execute the tasks in seconds}$

Efficiency = $\text{Ideal time taken (Number of tasks * time for each task execution)}$