CS 553: CLOUD COMPUTING

Manual for Programming Assignment #3

Task Execution Framework

This document provides step-by-step guide to run benchmark programs.

All programs are tested in AWS micro instance with Unix operating system.

In-Memory Queue Program

Pre-requisite: Java run time is present in system

You should have workload File

Running Program for In-memory is very easy all you need to is to run the below command:

```
java –jar CloudKonClient1.0.jar –s local –t 1 –w workload.txt
```

If the client takes

–s parameter as "local" it will run the program as in-memory

-t is for number of Threads to run as worker

-w is for work load file.

SQS Queue Program

Pre-requisite: Java run time is present in system

You should have workload File

You should have two SQS Queue already created

You should have DynamoDB table named as "CloudKonTask" and column as "TaskId".

Running Program for SQS have two parts Client and worker

To run client:

```
java –jar CloudKonClient1.0.jar –s sub con –w workload.txt
```

If the client takes

–s parameter takes two Queue first for task submission and second from task completion

-w is for work load file.

-t is not present for this because we need to launch worker manually and for running performance I have created another executable.

To worker client:

```
java –jar CloudKonWorker1.0.jar –s sub com
```

If the client takes

–s parameter takes two Queue first for task submission and second from task completion

Performance Run

For performance I have created separate executable because we have different number of task as well with different number of task length.

For In–Memory

I have created a shell file that will launch the entire task on one go.

command.sh

```
#!/bin/bash

java -jar Test1.jar perf 1 &
pid1=$!
wait $pid1

java -jar Test1.jar perf 2 &
pid2=$!
wait $pid2

java -jar Test1.jar perf 4 &
pid3=$!
wait $pid3

java -jar Test1.jar perf 8 &
pid4=$!
wait $pid4

java -jar Test1.jar perf 16 &
pid5=$!
wait $pid5

java -jar Test1.jar Sleep 1 &
pid11=$!
wait $pid11

java -jar Test1.jar Sleep 2 &
pid21=$!
wait $pid21

java -jar Test1.jar Sleep 4 &
pid31=$!
wait $pid31

java -jar Test1.jar Sleep 8 &
pid41=$!
wait $pid41

java -jar Test1.jar Sleep 16 &
pid51=$!
wait $pid51

java -jar Test1.jar Sleep1 1 &
```

Output is

```
Starting ThroughPut Experiment with No of worker Threads 1
Total Time took is 469
ThroughPut  for Number Of threads 1is 213.21961620469082
Starting ThroughPut Experiment with No of worker Threads 2
Total Time took is 308
ThroughPut  for Number Of threads 2is 324.6753246753247
Starting ThroughPut Experiment with No of worker Threads 4
Total Time took is 345
ThroughPut  for Number Of threads 4is 289.8550724637681
Starting ThroughPut Experiment with No of worker Threads 8
Total Time took is 409
ThroughPut  for Number Of threads 8is 244.49877750611248
Starting ThroughPut Experiment with No of worker Threads 16
Total Time took is 422
ThroughPut  for Number Of threads 16is 236.96682464454977
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 1
Total Time took is 10195
Efficency  for Number Of threads 1is 98.0872976949485
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 2
Total Time took is 10188
Efficency  for Number Of threads 2is 196.30938358853552
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 4
Total Time took is 10192
Efficency  for Number Of threads 4is 392.4646781789639
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 8
Total Time took is 10180
Efficency  for Number Of threads 8is 785.8546168958743
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 16
Total Time took is 10197
Efficency  for Number Of threads 16is 1569.0889477297244
Starting Efficency Experiment with Sleep 1 sec No of worker Threads 1
Total Time took is 100025
Efficency  for Number Of threads 1is 99.9750062484379
Starting Efficency Experiment with Sleep 1 sec No of worker Threads 2
```

```
Total Time took is 100026

Efficency  for Number Of threads 2is 199.94801351648573

Starting Efficency Experiment with Sleep 1 sec No of worker Threads 4

Total Time took is 100027

Efficency  for Number Of threads 4is 399.89202915212894

Starting Efficency Experiment with Sleep 1 sec No of worker Threads 8

Total Time took is 100037

Efficency  for Number Of threads 8is 799.7041094794926

Starting Efficency Experiment with Sleep 1 sec No of worker Threads 16

Total Time took is 100032

Efficency  for Number Of threads 16is 1599.488163787588

Starting Efficency Experiment with Sleep 10 sec No of worker Threads 1

Total Time took is 100008

Efficency  for Number Of threads 1is 99.99200063994881

Starting Efficency Experiment with Sleep 10 sec No of worker Threads 2

Total Time took is 100009

Efficency  for Number Of threads 2is 199.98200161985423

Starting Efficency Experiment with Sleep 10 sec No of worker Threads 4

Total Time took is 100009

Efficency  for Number Of threads 4is 399.96400323970846

Starting Efficency Experiment with Sleep 10 sec No of worker Threads 8

Total Time took is 100009

Efficency  for Number Of threads 8is 799.9280064794169

Starting Efficency Experiment with Sleep 10 sec No of worker Threads 16

Total Time took is 100015

Efficency  for Number Of threads 16is 1599.7600359946007

All Process Completed
```

For SQS Queue

Client: all we to is to launch client based on number of worker

Worker: I have host.txt file that contains list of worker and I run command using pssh to execute my worker parllely.

Client:

```
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep 1
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 1
Total time took 145008
Efficency  for Number Of threads 1is 6.896171245724374
```

```
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep1 1
Starting Efficency Experiment with Sleep 1 sec No of worker Threads 1
Total time took 30005
Efficency  for Number Of threads 1is 333.27778703549404
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep10 1
Starting Efficency Experiment with Sleep 10 sec No of worker Threads 1
Total time took 25004
Efficency  for Number Of threads 1is 399.93601023836186
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep10 2
Starting Efficency Experiment with Sleep 10 sec No of worker Threads 2
^C[ec2-user@ip-172-31-22-63 ~]$ ^C
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep10 2
Starting Efficency Experiment with Sleep 10 sec No of worker Threads 2
Total time took 25004
Efficency  for Number Of threads 2is 799.8720204767237
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep1 2
Starting Efficency Experiment with Sleep 1 sec No of worker Threads 2
Total time took 30004
Efficency  for Number Of threads 2is 666.5777896280497
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep 2
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 2
Total time took 135008
Efficency  for Number Of threads 2is 14.813936951884333
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar perf
Total time took 630023
ThroughPut is 0.01587243640311544
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep10 4
Starting Efficency Experiment with Sleep 10 sec No of worker Threads 4
Total time took 25004
Efficency  for Number Of threads 4is 1599.7440409534474
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep1 4
Starting Efficency Experiment with Sleep 1 sec No of worker Threads 4
Total time took 30004
Efficency  for Number Of threads 4is 1333.1555792560994
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep 4
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 4
Total time took 150012
Efficency  for Number Of threads 4is 26.66453350398635
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar perf
Total time took 340016
ThroughPut is 0.029410380687967624
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep10 8
Starting Efficency Experiment with Sleep 10 sec No of worker Threads 8
Total time took 25004
Efficency  for Number Of threads 8is 3199.488081906895
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep1 8
Starting Efficency Experiment with Sleep 1 sec No of worker Threads 8
Total time took 35004
Efficency  for Number Of threads 8is 2285.4530910753056
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep 8
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 8
Total time took 180009
Efficency  for Number Of threads 8is 44.44222233332778
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar perf
Total time took 205010
ThroughPut is 0.04877810838495683
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep10 16
Starting Efficency Experiment with Sleep 10 sec No of worker Threads 16
Total time took 25005
Efficency  for Number Of threads 16is 6398.7202559488105
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep1 16
Starting Efficency Experiment with Sleep 1 sec No of worker Threads 16
Total time took 40006
Efficency  for Number Of threads 16is 3999.4000899865023
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar sleep 16
Starting Efficency Experiment with Sleep 10 ms No of worker Threads 16
Total time took 235018
Efficency  for Number Of threads 16is 68.07989175297212
[ec2-user@ip-172-31-22-63 ~]$ java -jar client.jar perf
Total time took 135010
ThroughPut is 0.07406858751203614
```

Worker:

```
dhcp131:EC2 Aakash$ pssh  -i  -h /Users/Aakash/Desktop/EC2/hosts.txt -t 100000000 -x
"-oStrictHostKeyChecking=no  -i Ec2s.pem" 'java -jar worker.jar'

[1] 22:41:46 [SUCCESS] ec2-user@ec2-54-187-207-233.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[2] 22:41:46 [SUCCESS] ec2-user@ec2-54-191-137-38.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[3] 22:41:46 [SUCCESS] ec2-user@ec2-54-186-31-122.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[4] 22:41:46 [SUCCESS] ec2-user@ec2-54-200-143-84.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[5] 22:41:46 [SUCCESS] ec2-user@ec2-54-200-229-16.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[6] 22:41:46 [SUCCESS] ec2-user@ec2-54-201-153-68.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[7] 22:41:46 [SUCCESS] ec2-user@ec2-54-186-190-144.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[8] 22:41:46 [SUCCESS] ec2-user@ec2-54-213-1-56.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[9] 22:41:46 [SUCCESS] ec2-user@ec2-54-186-113-28.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[10] 22:41:46 [SUCCESS] ec2-user@ec2-54-201-174-74.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[11] 22:41:46 [SUCCESS] ec2-user@ec2-54-187-238-62.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[12] 22:41:46 [SUCCESS] ec2-user@ec2-54-201-254-37.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[13] 22:41:46 [SUCCESS] ec2-user@ec2-54-191-187-100.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[14] 22:41:46 [SUCCESS] ec2-user@ec2-54-201-121-25.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[15] 22:41:46 [SUCCESS] ec2-user@ec2-54-200-128-252.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
[16] 22:41:46 [SUCCESS] ec2-user@ec2-54-200-168-15.us-west-2.compute.amazonaws.com
Connection to DynamoDB sucessful !
```

Host File:

```
ec2-user@ec2-54-187-207-233.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-200-128-252.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-201-153-68.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-201-121-25.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-191-137-38.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-186-31-122.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-200-143-84.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-186-190-144.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-200-229-16.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-201-254-37.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-213-1-56.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-186-113-28.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-191-187-100.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-200-168-15.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-201-174-74.us-west-2.compute.amazonaws.com

ec2-user@ec2-54-187-238-62.us-west-2.compute.amazonaws.com
```