

## CS 553: CLOUD COMPUTING

### Evaluation for Programming Assignment #3

#### Task Execution Framework

##### In-Memory Queue

##### System Information:

AWS Instance type: t2.micro

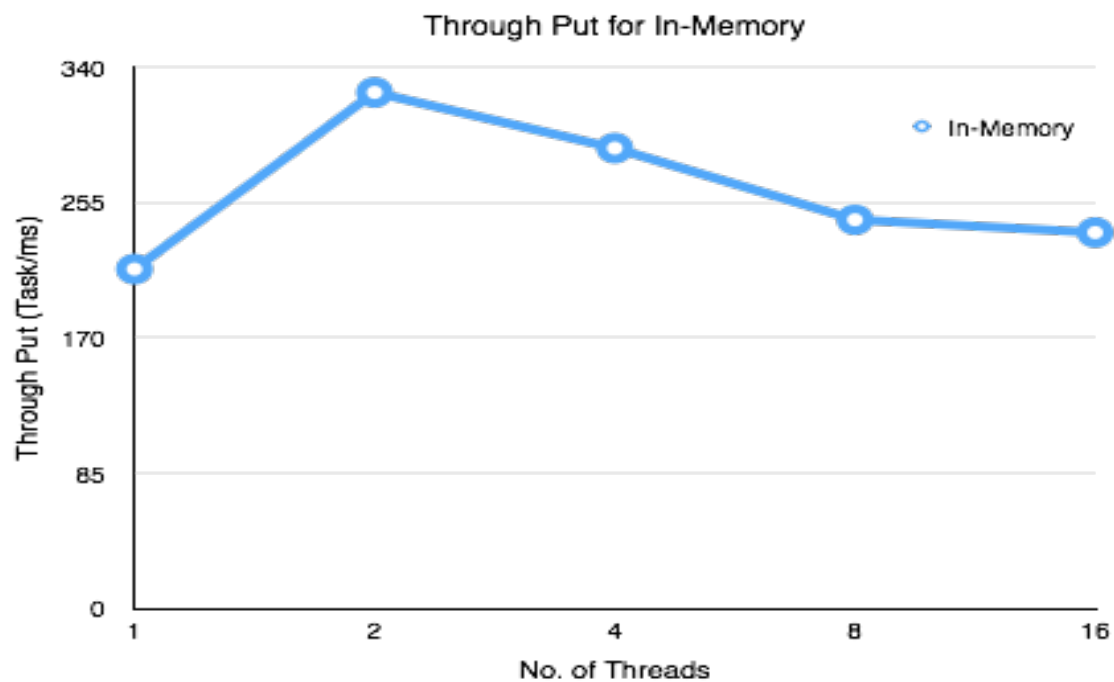
Memory: 1GB

CPU: 1

Disk: 8 GB

##### Through-put for In-Memory

No of Workers	Through Put (Task/ms)
1	213
2	324
4	289
8	244
16	236



### Graph Explanation:

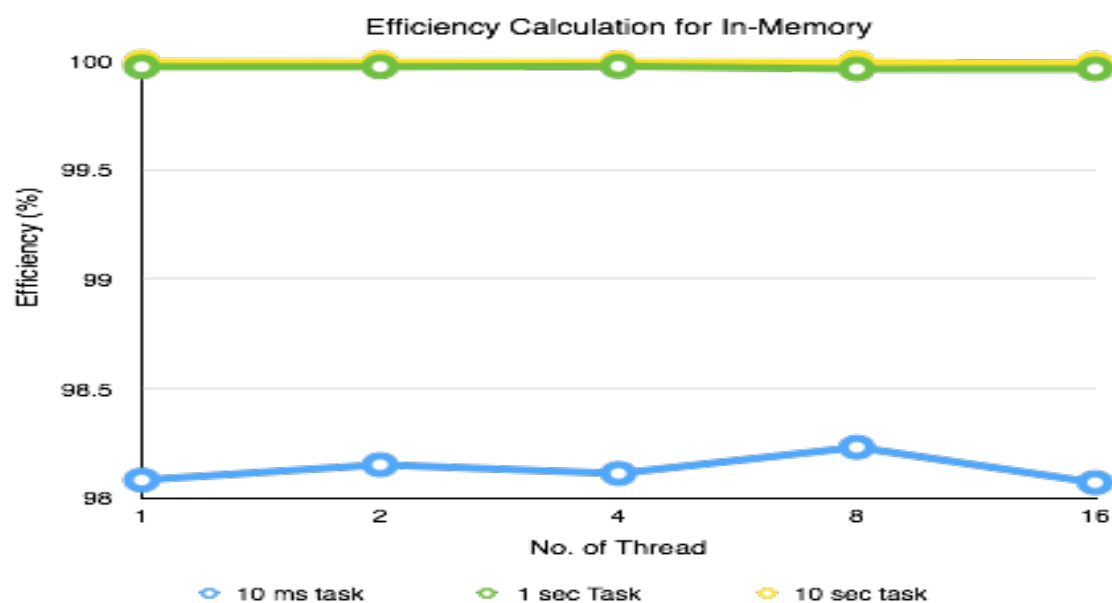
In the above we can see that maximum throughput we get for when the number of threads is 2. For 2 threads In-memory program is using efficiently CPU for task processing as well as memory for Queue.

As the task are running on in-memory Queue which is on main memory of system which give very high throughput. Only things that we need to take care to efficiently manage the thread pool.

For one thread its quite obvious that number of task to is by only one thread so the throughput is less. On increasing the number of thread the throughput is decreased from 4 to 16 because increasing number of thread does not means always increase in performance because they use CPU resource.

### Efficiency for In-Memory

Number of Threads	10 ms task	1 sec Task	10 sec task
1	98.08	99.97	99.992
2	98.15	99.97	99.99
4	98.11	99.9725	99.99
8	98.23	99.96	99.99
16	98.068	99.96	99.985



### Graph Explanation

In the above graph we can see that more granular the task is less efficiency we get for those task. For task that is 1 sec and 10 sec long we are getting around 99% efficiency because processor get enough time to execute those task.

Where as when the task is of 10ms the efficiency is decreased to 98% because task is of shorter duration.

We are getting efficiency of more than 98% because the task is performed in main memory of same system.

### SQS Queue

#### System Information:

AWS Instance type: t2.micro

Memory: 1GB

CPU: 1

Disk: 8 GB

DynamoDB: read/ Write = 1000

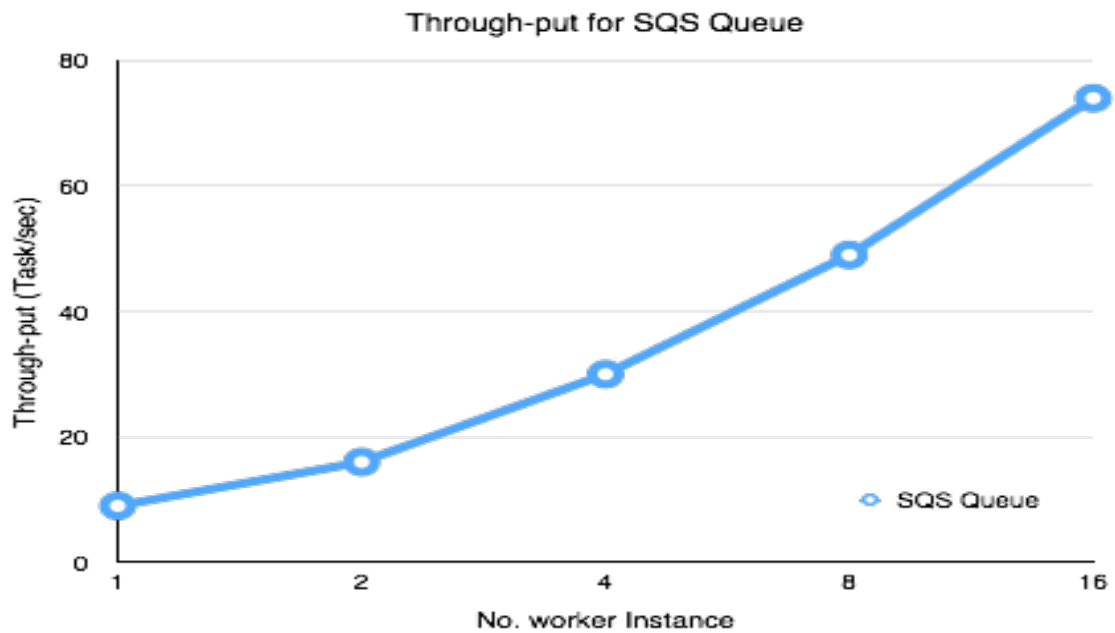
SQS Queue:

Delay: 0 sec

Pooling: 20 sec

#### Through-put for SQS Queue

No of Workers	Through Put (jobs/sec)
1	9
2	16
4	30
8	49
16	74

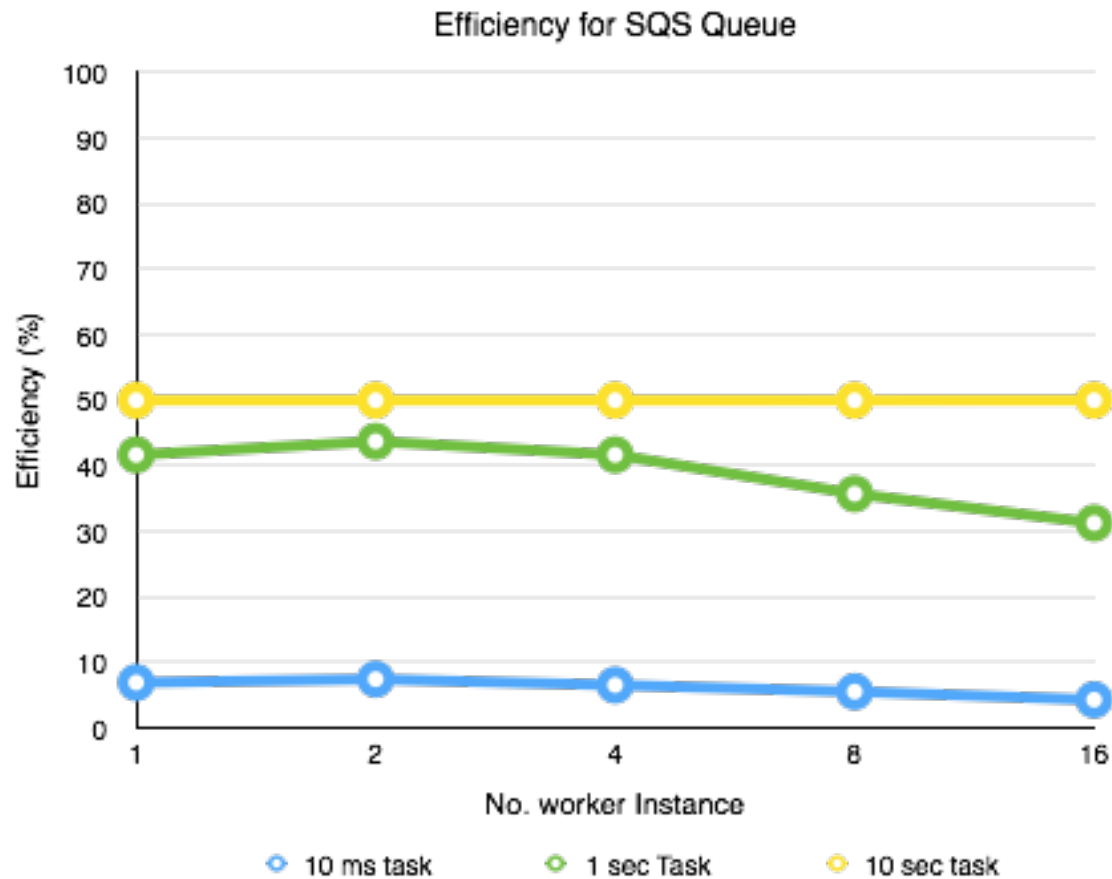


### Graph Explanation:

In the above we can see that as we increased the number of worker our throughput increased because there are more worker to pull task from queue. Max throughput we got on running 16 workers instantaneously.

### Efficiency for SQS Queue

Number of workers	10 ms task	1 sec Task	10 sec task
1	6.89	41.65	49.99
2	7.40	43.72	49.99
4	6.5	41.66	49.99
8	5.5	35.71	49.98
16	4.25	31.24	49.99



### Graph Explanation:

In the above graph we can see that more granular the task is less efficiency we get for those task.

For task that is 10 sec long we are getting around efficiency of 50% because task is of larger duration and it is of very less number leading to less number of pull request from worker.

For task is of 1s the efficiency is decreased from 50% to 30% as we increased the number of worker. It has maximum efficiency when the number of worker is 2.

For task is of 10ms the efficiency is less than 10% because task is of shorter duration and is of large number leading to lots of pull request from worker.