

SCREEN SHOTS:

• Local Worker:

1. Throughput Experiment

ubuntu@ip-172-31-60-91: ~/CS553_AS3
priyankpshah@priyankpshah-Lenovo-Y40-80: ~
ubuntu@ip-172-31-60-91: ~/PycharmProjects/CS553 x
ubuntu@ip-172-31-60-91: ~/CS553_AS3 185x49

```
Model:          62
Stepping:       4
CPU MHz:      2500.00
Bogomips:     5000.20
Virtualization vendor: Xen
L1d cache:    32K
L1i cache:    32K
L2 cache:     256K
L3 cache:    25600K
NUMA node0 CPU(s): 0
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 4 -w Worker1/10_ms
0 thread0
1 thread1
2 thread2
3 thread3
Execution Time: 10.48 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi Client.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 8 -w Worker1/10_ms
0 thread0
1 thread1
2 thread2
3 thread3
4 thread4
5 thread5
6 thread6
7 thread7
Execution Time: 11.533 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 16 -w Worker1/10_ms
0 thread0
1 thread1
2 thread2
3 thread3
4 thread4
5 thread5
6 thread6
7 thread7
8 thread8
9 thread9
10 thread10
11 thread11
12 thread12
13 thread13
14 thread14
15 thread15
Execution Time: 13.213 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 1 -w Worker1/10_ms
0 thread0
Execution Time: 10.774 Sec
```

2. 10 ms Experiment

ubuntu@ip-172-31-60-91: ~/CS553_AS3
priyankpshah@priyankpshah-Lenovo-Y40-80: ~
ubuntu@ip-172-31-60-91: ~/PycharmProjects/CS553 x
ubuntu@ip-172-31-60-91: ~/CS553_AS3 185x49

```
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 2 -w Worker1/10_ms
0 thread0
1 thread1
Execution Time: 10.162 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 2 -w Worker1/10_ms
0 thread0
1 thread1
Execution Time: 10.165 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 4 -w Worker1/10_ms
0 thread0
1 thread1
2 thread2
3 thread3
Execution Time: 10.169 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 8 -w Worker1/10_ms
0 thread0
1 thread1
2 thread2
3 thread3
4 thread4
5 thread5
6 thread6
7 thread7
Execution Time: 10.174 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 16 -w Worker1/10_ms
0 thread0
1 thread1
2 thread2
3 thread3
4 thread4
5 thread5
6 thread6
7 thread7
8 thread8
9 thread9
10 thread10
11 thread11
12 thread12
13 thread13
14 thread14
15 thread15
```

3. 1 Second Experiment:

```
ubuntu@ip-172-31-60-91:~/CS553_AS3
priyankpshah@priyankpshah-Lenovo-Y40-80:~ x priyankpshah@priyankpshah-Lenovo-Y40-80:~/PycharmProjects/CS553 x
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 2 -w Worker1/1000_ms
0 thread0
1 thread1
Execution Time: 100.119 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 4 -w Worker1/1000_ms
0 thread0
1 thread1
2 thread2
3 thread3
Execution Time: 100.122 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 8 -w Worker1/1000_ms
0 thread0
1 thread1
2 thread2
3 thread3
4 thread4
5 thread5
6 thread6
7 thread7
Execution Time: 100.121 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 16 -w Worker1/1000_ms
0 thread0
1 thread1
2 thread2
3 thread3
4 thread4
5 thread5
6 thread6
7 thread7
8 thread8
9 thread9
10 thread10
11 thread11
12 thread12
13 thread13
14 thread14
15 thread15
Execution Time: 100.123 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$
```

```
ubuntu@ip-172-31-60-91:~/PycharmProjects/CS553 x
priyankpshah@priyankpshah-Lenovo-Y40-80:~ x priyankpshah@priyankpshah-Lenovo-Y40-80:~/PycharmProjects/CS553 x
ubuntu@ip-172-31-60-91:~/PycharmProjects/CS553 x
```

4. 10 Second Experiment:

```
ubuntu@ip-172-31-60-91:~/CS553_AS3
priyankpshah@priyankpshah-Lenovo-Y40-80:~ x priyankpshah@priyankpshah-Lenovo-Y40-80:~/PycharmProjects/CS553 x
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 1 -w Worker1/10_s
0 thread0
1 thread1
[1]+ Stopped python Client.py -s LOCAL -t 1 -w Worker1/10_s
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 2 -w Worker1/10_s
0 thread0
1 thread1
2 thread2
3 thread3
Execution Time: 100.102 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 4 -w Worker1/10_s
0 thread0
1 thread1
2 thread2
3 thread3
4 thread4
5 thread5
6 thread6
7 thread7
Execution Time: 100.102 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 8 -w Worker1/10_s
0 thread0
1 thread1
2 thread2
3 thread3
4 thread4
5 thread5
6 thread6
7 thread7
8 thread8
9 thread9
10 thread10
11 thread11
12 thread12
13 thread13
14 thread14
15 thread15
Execution Time: 100.101 Sec
ubuntu@ip-172-31-60-91:~/CS553_AS3$ vi writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python writer.py
ubuntu@ip-172-31-60-91:~/CS553_AS3$ python Client.py -s LOCAL -t 16 -w Worker1/10_s
0 thread0
1 thread1
2 thread2
3 thread3
4 thread4
5 thread5
6 thread6
7 thread7
8 thread8
9 thread9
10 thread10
11 thread11
12 thread12
13 thread13
14 thread14
15 thread15
```

```
ubuntu@ip-172-31-60-91:~/PycharmProjects/CS553 x
priyankpshah@priyankpshah-Lenovo-Y40-80:~ x priyankpshah@priyankpshah-Lenovo-Y40-80:~/PycharmProjects/CS553 x
ubuntu@ip-172-31-60-91:~/PycharmProjects/CS553 x
```

- **Remote Experiment:**

1. **SQS Screen shot:**

The screenshot shows the AWS SQS Management Console. At the top, there's a navigation bar with tabs like 'AWS', 'Services', and 'Edit'. Below it, a search bar says 'Create New Queue' and 'Queue Actions'. A table lists one item: 'task_table'. The table has columns for 'Name' and 'QNAME', with values 'task_table' and 'task_table' respectively. It also shows 'Messages Available' (2,000), 'Messages in Flight' (0), and 'Created' (2016-05-02 04:29:32 GMT-05:00). The bottom of the screen shows a toolbar with icons for 'Feedback', 'English', and download links for 'install.sh', 'pssh_Script.sh', and 'WhatsApp...jpg'.

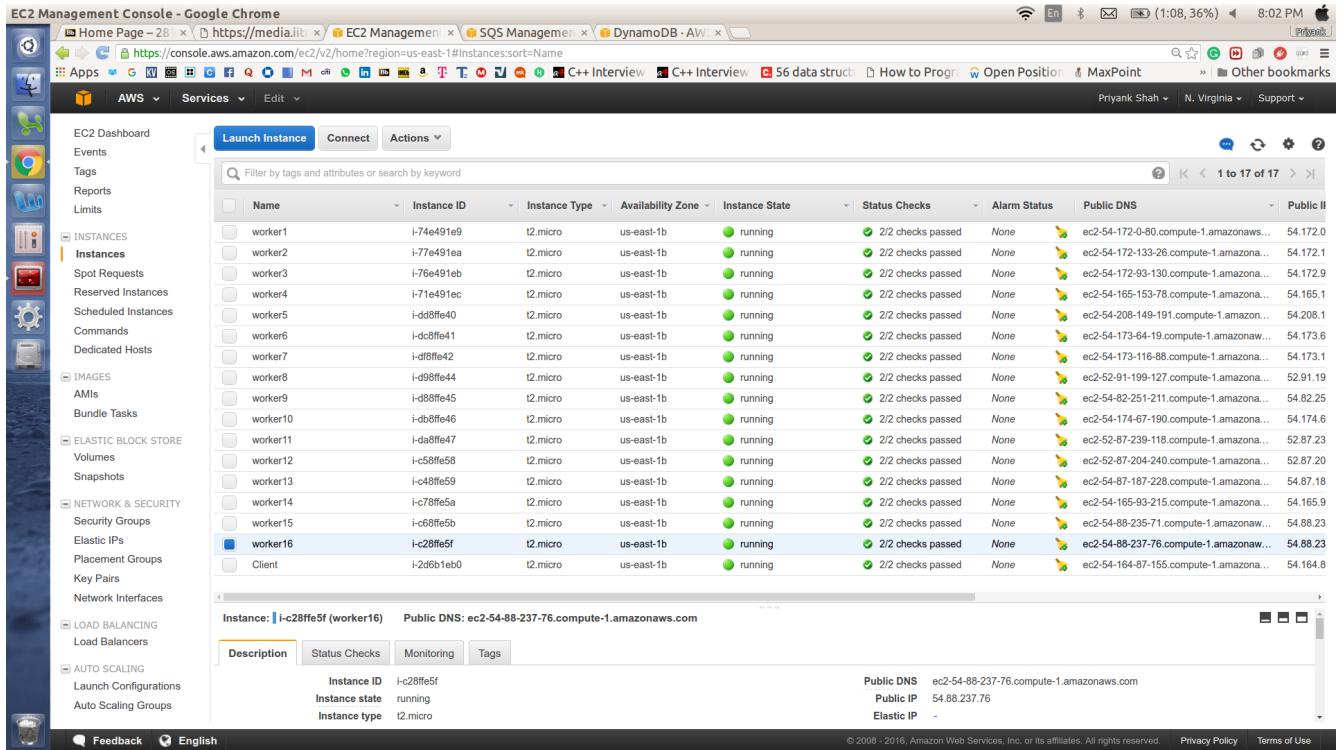
2. **DynamoDB Screen Shot:**

The screenshot shows the AWS DynamoDB Management Console. On the left, a sidebar has 'DynamoDB' selected, with 'Tables' and 'Reserved capacity' options. The main area shows a table named 'task_table'. The 'Items' tab is selected, showing 100 items. The table has two columns: 'task_id' and 'task'. The data is as follows:

task_id	task
6239	sleep 100
5219	sleep 100
4697	sleep 100
1915	sleep 100
9927	sleep 100
924	sleep 100
1171	sleep 100
972	sleep 100
5097	sleep 100
8567	sleep 100
9176	sleep 100

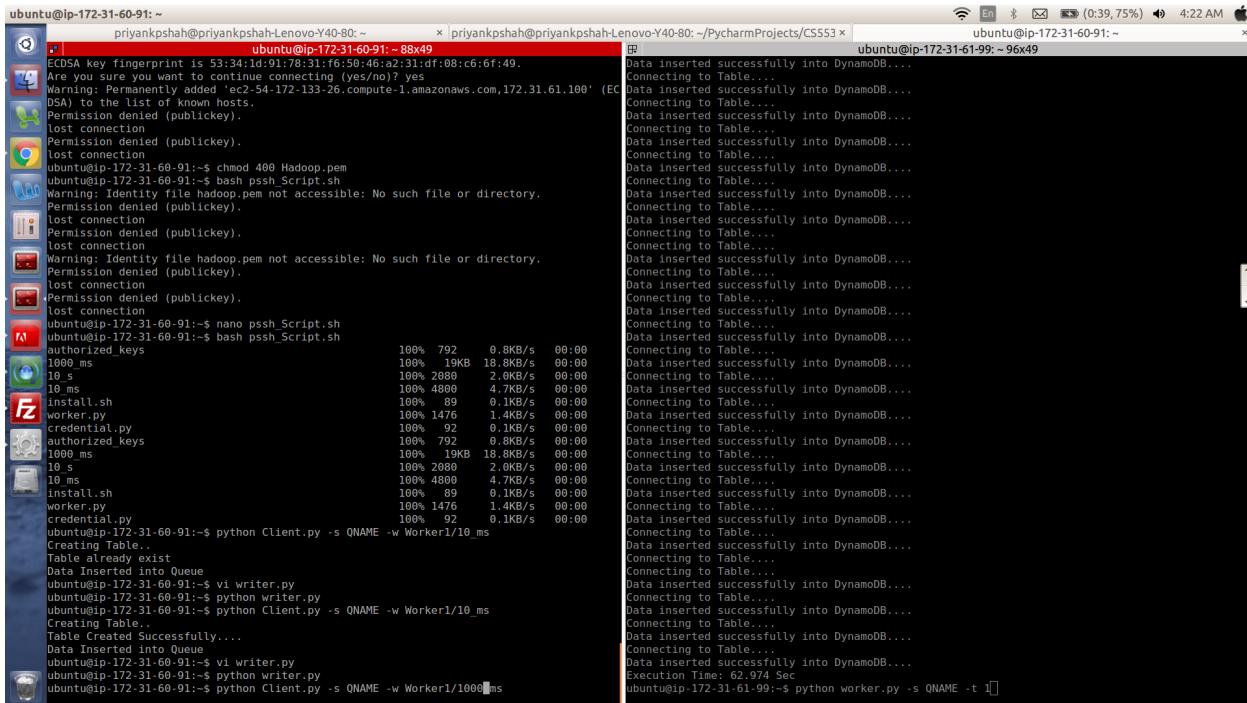
The bottom of the screen shows a toolbar with icons for 'Feedback', 'English', and download links for 'install.sh', 'pssh_Script.sh', and 'WhatsApp...jpg'.

3. EC2- Instance Screen Shot:



The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with various navigation links like EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area displays a table of instances. A specific instance, 'worker16', is selected and shown in a detailed view below the table. The detailed view includes tabs for Description, Status Checks, Monitoring, and Tags. It shows the instance ID (i-2c8ffe5f), state (running), and type (t2.micro). It also lists Public DNS (ec2-54-88-237-76.compute-1.amazonaws.com), Public IP (54.88.237.76), and Elastic IP (-). At the bottom of the page, there are links for Feedback, English, and a copyright notice from 2008.

4. 1 Worker Experiment:



The screenshot shows three terminal windows on an Ubuntu EC2 instance (ip-172-31-60-91). The first window shows the user navigating through files and executing bash scripts to set up Hadoop and PSSH. The second window shows the user running a Python script named Client.py with arguments -s QNAME -w Worker1/10_ms, which creates a table in DynamoDB and inserts data. The third window shows the user running another Python script named worker.py with the same arguments, which also interacts with the DynamoDB table.

```
ubuntu@ip-172-31-60-91:~$ chmod 400 hadoop.pem
ubuntu@ip-172-31-60-91:~$ bash pssh.Script.sh
Warning: Identity file hadoop.pem not accessible: No such file or directory.
Permission denied (publickey).
lost connection
lost connection
Warning: Identity file hadoop.pem not accessible: No such file or directory.
Permission denied (publickey).
lost connection
authorized_keys
1000 ms
10 s
10 ms
install.sh
worker.py
authorized_keys
1000 ms
10 s
10 ms
install.sh
worker.py
credential.py
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_ms
Creating Table...
Table Created Successfully...
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ vi writer.py
ubuntu@ip-172-31-60-91:~$ python writer.py
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_ms
Creating Table...
Data Inserted into Queue
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ python worker.py -s QNAME -t 1
```

5. Efficiency Experiment 2 Workers:

The terminal window shows the execution of a Hadoop job with two workers. The command run is:

```
ubuntu@ip-172-31-60-91:~$ hadoop job -jar CS553_AS3.jar Client.py credential.py dynamodb.py host_list pssh_script.sh worker1 workload.txt writer.py
```

The output shows the process of connecting to tables and inserting data into DynamoDB:

```
Last login: Tue May 3 00:42:10 2016 from 104.194.114.187
ubuntu@ip-172-31-60-91:~$ ls
Client.py credential.py CS553_AS3 Hadoop.pem install.sh Worker1 WORKLOAD.txt writer.py
ubuntu@ip-172-31-60-91:~$ vi Client.py
ubuntu@ip-172-31-60-91:~$ vi writer.py
ubuntu@ip-172-31-60-91:~$ python writer.py
ubuntu@ip-172-31-60-91:~$ vi writer.py
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Creating Table...
Table Created Successfully....
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Creating Table...
Table Created Successfully....
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Creating Table...
Table Created Successfully....
```

Two separate windows show the progress of the workers:

- Upsilon**: ubuntu@ip-172-31-61-99:~ 88x24
- Upsilon**: ubuntu@ip-172-31-61-100:~ 96x24

Both workers show a continuous loop of connecting to tables and inserting data into DynamoDB.

6. Efficiency Experiment 4 Workers:

The terminal window shows the execution of a Hadoop job with four workers. The command run is:

```
ubuntu@ip-172-31-61-102:~$ hadoop job -jar CS553_AS3.jar Client.py credential.py dynamodb.py host_list pssh_script.sh worker1 worker2 worker3 worker4 workload.txt writer.py
```

The output shows the process of connecting to tables and inserting data into DynamoDB:

```
Last login: Tue May 3 00:42:10 2016 from 104.194.114.187
ubuntu@ip-172-31-60-91:~$ ls
Client.py credential.py CS553_AS3 Hadoop.pem install.sh Worker1 WORKLOAD.txt writer.py
ubuntu@ip-172-31-60-91:~$ vi Client.py
ubuntu@ip-172-31-60-91:~$ vi writer.py
ubuntu@ip-172-31-60-91:~$ python writer.py
ubuntu@ip-172-31-60-91:~$ vi writer.py
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Creating Table...
Table Created Successfully....
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Creating Table...
Table Created Successfully....
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Creating Table...
Table Created Successfully....
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Creating Table...
Table Created Successfully....
```

Four separate windows show the progress of the workers:

- Upsilon**: ubuntu@ip-172-31-61-99:~ 44x24
- Upsilon**: ubuntu@ip-172-31-61-101:~ 43x24
- Upsilon**: ubuntu@ip-172-31-61-100:~ 48x24
- Upsilon**: ubuntu@ip-172-31-61-102:~ 47x24

Each worker shows a continuous loop of connecting to tables and inserting data into DynamoDB.

7.4 Worker Throughput Experiment:

```
ubuntu@ip-172-31-60-91:~
```

```
queue = connection.create_queue(queueName)
File "/usr/local/lib/python2.7/dist-packages/boto/sqs/connection.py", line 96, in create_queue
    return self.get_object('CreateQueue', params, Queue)
File "/usr/local/lib/python2.7/dist-packages/boto/connection.py", line 1208, in get_object
    raise self.ResponseError(response.status, response.reason, body)
boto.exception.SQSError: SQSError: 400 Bad Request
<?xml version='1.0'?><ErrorResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/"><Error><Type>Sender</Type><Code>AWS.SimpleQueueService.QueueDeletedRecently</Code><Message>You must wait 60 seconds after deleting a queue before you can create another with the same name.</Message><Detail/></Error><RequestId>65f114b7-f41b-5ac6-b7b3-5b160f810a38</RequestId></ErrorResponse>
```

```
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Traceback (most recent call last):
  File "Client.py", line 91, in <module>
    remoteWorker(queue_name,worker_no,filename)
  File "Client.py", line 34, in remoteWorker
    queue = connection.create_queue(queueName)
  File "/usr/local/lib/python2.7/dist-packages/boto/sqs/connection.py", line 96, in create_queue
    return self.get_object('CreateQueue', params, Queue)
  File "/usr/local/lib/python2.7/dist-packages/boto/connection.py", line 1208, in get_object
    raise self.ResponseError(response.status, response.reason, body)
boto.exception.SQSError: SQSError: 400 Bad Request
<?xml version='1.0'?><ErrorResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/"><Error><Type>Sender</Type><Code>AWS.SimpleQueueService.QueueDeletedRecently</Code><Message>You must wait 60 seconds after deleting a queue before you can create another with the same name.</Message><Detail/></Error><RequestId>28e62a0-f1fc-54dc-88f0-2d010a5d5db</RequestId></ErrorResponse>
```

```
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Traceback (most recent call last):
  File "Client.py", line 91, in <module>
    remoteWorker(queue_name,worker_no,filename)
  File "Client.py", line 34, in remoteWorker
    queue = connection.create_queue(queueName)
  File "/usr/local/lib/python2.7/dist-packages/boto/sqs/connection.py", line 96, in create_queue
    return self.get_object('CreateQueue', params, Queue)
  File "/usr/local/lib/python2.7/dist-packages/boto/connection.py", line 1208, in get_object
    raise self.ResponseError(response.status, response.reason, body)
boto.exception.SQSError: SQSError: 400 Bad Request
<?xml version='1.0'?><ErrorResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/"><Error><Type>Sender</Type><Code>AWS.SimpleQueueService.QueueDeletedRecently</Code><Message>You must wait 60 seconds after deleting a queue before you can create another with the same name.</Message><Detail/></Error><RequestId>feet73t14-1747-57b5-924d-78b5ece0c022</RequestId></ErrorResponse>
```

```
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10_s
Creating Table...
Table Created Successfully....
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~
```

```
ubuntu@ip-172-31-61-99:~
```

```
Data inserted successfully into DynamoDB...
Connecting to Table...
Data inserted successfully into DynamoDB...
Execution Time: 101.131 Sec
ubuntu@ip-172-31-61-99:~
```

```
ubuntu@ip-172-31-61-100:~
```

```
Data inserted successfully into DynamoDB...
Connecting to Table...
Data inserted successfully into DynamoDB...
Execution Time: 100.992 Sec
ubuntu@ip-172-31-61-100:~
```

```
ubuntu@ip-172-31-61-101:~
```

```
Data inserted successfully into DynamoDB...
Connecting to Table...
Data inserted successfully into DynamoDB...
Execution Time: 101.071 Sec
ubuntu@ip-172-31-61-101:~
```

```
ubuntu@ip-172-31-61-102:~
```

```
Data inserted successfully into DynamoDB...
Connecting to Table...
Data inserted successfully into DynamoDB...
Execution Time: 101.179 Sec
ubuntu@ip-172-31-61-102:~
```

8.8 Worker Throughput Experiment Initiation:

```
ubuntu@ip-172-31-61-99:~
```

```
ubuntu@ip-172-31-60-91:~
```

```
python Client.py -s QNAME -w Worker1/10_ms
Creating Table...
Table Created Successfully....
```

```
ubuntu@ip-172-31-61-99:~
```

```
python worker.py -s QNAME -t 8
```

```
ubuntu@ip-172-31-61-99:~
```

```
python worker.py -s QNAME -t 8
```

```
ubuntu@ip-172-31-61-100:~
```

```
python worker.py -s QNAME -t 8
```

```
ubuntu@ip-172-31-52-90:~
```

```
python worker.py -s QNAME -t 8
```

```
ubuntu@ip-172-31-61-101:~
```

```
python worker.py -s QNAME -t 8
```

```
ubuntu@ip-172-31-52-79:~
```

```
python worker.py -s QNAME -t 8
```

```
ubuntu@ip-172-31-61-102:~
```

```
python worker.py -s QNAME -t 8
```

```
ubuntu@ip-172-31-52-85:~
```

```
python worker.py -s QNAME -t 8
```

9.8 Worker Throughput Experiment (1 Sec):

```
ubuntu@lp-172-31-61-99: ~
[ ]                                     ubuntu@ip-172-31-60-91: ~ 93x51
 Ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10.ms
Creating Table..
Data Created Successfully....
Data Inserted Into Queue
ubuntu@ip-172-31-60-91:~$ vi writer.py
ubuntu@ip-172-31-60-91:~$ python writer.py
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/1000.ms
Creating Table..
Data Created Successfully....
Data Inserted Into Queue
ubuntu@ip-172-31-60-91:~$ vi writer.py
ubuntu@ip-172-31-60-91:~$ python writer.py
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10.s
Creating Table..
Table already exist
Data Inserted Into Queue
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10.s
Creating Table..
Data Created Successfully....
Data Inserted Into Queue
ubuntu@ip-172-31-60-91:~$ 
```

10.8 Worker Throughput Experiment (10 Sec):

```
ubuntu@ip-172-31-61-99: ~                               ubuntu@ip-172-31-60-91: ~ 93x51
[?]                                         ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10 ms
Creating Table..
Table Created Successfully....
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ vi writer.py
ubuntu@ip-172-31-60-91:~$ python writer.py
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/1000 ms
Creating Table..
Table Created Successfully....
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ vi writer.py
ubuntu@ip-172-31-60-91:~$ python writer.py
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10 s
Creating Table..
Table already exist
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ python Client.py -s QNAME -w Worker1/10 s
Creating Table..
Table Created Successfully....
Data Inserted into Queue
ubuntu@ip-172-31-60-91:~$ 
```

11. 16 Worker Throughput Experiment (10 ms):

This screenshot shows 16 terminal windows running simultaneously on an Ubuntu system. Each window displays the execution of a Python script named 'Client.py' with parameters '-s QNAME -w Worker1/10_ms'. The output shows the client connecting to a table and inserting data into a DynamoDB database. The process is repeated 38 times per window. The execution time for each iteration is consistently around 102 ms.

```

ubuntu@ip-172-31-60-91: ~
[1] ubuntu@ip-172-31-60-91: ~ 38x51
[1] ubuntu@ip-172-31-52-86: ~ 38x11
[1] ubuntu@ip-172-31-52-82: ~ 38x11
[1] ubuntu@ip-172-31-61-99: ~ 38x11
[1] ubuntu@ip-172-31-52-87: ~ 38x11
[1] ubuntu@ip-172-31-52-83: ~ 38x11
[1] ubuntu@ip-172-31-61-100: ~ 38x11
[1] ubuntu@ip-172-31-52-90: ~ 28
[1] ubuntu@ip-172-31-52-88: ~ 38x11
[1] ubuntu@ip-172-31-52-84: ~ 38x11
[1] ubuntu@ip-172-31-61-101: ~ 38x11
[1] ubuntu@ip-172-31-52-79: ~ 28
[1] ubuntu@ip-172-31-52-81: ~ 38x11
[1] ubuntu@ip-172-31-52-80: ~ 38x11
[1] ubuntu@ip-172-31-61-102: ~ 38x11
[1] ubuntu@ip-172-31-52-85: ~ 28
[1] ubuntu@ip-172-31-52-86: ~ 38x11
[1] ubuntu@ip-172-31-52-82: ~ 38x11
[1] ubuntu@ip-172-31-61-99: ~ 38x11
[1] ubuntu@ip-172-31-52-87: ~ 38x11
[1] ubuntu@ip-172-31-52-83: ~ 38x11
[1] ubuntu@ip-172-31-61-100: ~ 38x11
[1] ubuntu@ip-172-31-52-90: ~ 28
[1] ubuntu@ip-172-31-52-88: ~ 38x11
[1] ubuntu@ip-172-31-52-84: ~ 38x11
[1] ubuntu@ip-172-31-61-101: ~ 38x11
[1] ubuntu@ip-172-31-52-79: ~ 28
[1] ubuntu@ip-172-31-52-81: ~ 38x11
[1] ubuntu@ip-172-31-52-80: ~ 38x11
[1] ubuntu@ip-172-31-61-102: ~ 38x11
[1] ubuntu@ip-172-31-52-85: ~ 28

```

12. 16 Worker Throughput Experiment(10 Sec):

This screenshot shows 16 terminal windows running simultaneously on an Ubuntu system. Each window displays the execution of a Python script named 'Client.py' with parameters '-s QNAME -w Worker1/10_ms'. The output shows the client connecting to a table and inserting data into a DynamoDB database. The process is repeated 38 times per window. The execution time for each iteration is consistently around 100 ms.

```

ubuntu@ip-172-31-60-91: ~
[1] ubuntu@ip-172-31-60-91: ~ 38x51
[1] ubuntu@ip-172-31-52-86: ~ 38x11
[1] ubuntu@ip-172-31-52-82: ~ 38x11
[1] ubuntu@ip-172-31-61-99: ~ 38x11
[1] ubuntu@ip-172-31-52-87: ~ 38x11
[1] ubuntu@ip-172-31-52-83: ~ 38x11
[1] ubuntu@ip-172-31-61-100: ~ 38x11
[1] ubuntu@ip-172-31-52-89: ~ 28
[1] ubuntu@ip-172-31-52-88: ~ 38x11
[1] ubuntu@ip-172-31-52-84: ~ 38x11
[1] ubuntu@ip-172-31-61-101: ~ 38x11
[1] ubuntu@ip-172-31-52-79: ~ 28
[1] ubuntu@ip-172-31-52-81: ~ 38x11
[1] ubuntu@ip-172-31-52-80: ~ 38x11
[1] ubuntu@ip-172-31-61-102: ~ 38x11
[1] ubuntu@ip-172-31-52-85: ~ 28
[1] ubuntu@ip-172-31-52-86: ~ 38x11
[1] ubuntu@ip-172-31-52-82: ~ 38x11
[1] ubuntu@ip-172-31-61-99: ~ 38x11
[1] ubuntu@ip-172-31-52-87: ~ 38x11
[1] ubuntu@ip-172-31-52-83: ~ 38x11
[1] ubuntu@ip-172-31-61-100: ~ 38x11
[1] ubuntu@ip-172-31-52-89: ~ 28
[1] ubuntu@ip-172-31-52-88: ~ 38x11
[1] ubuntu@ip-172-31-52-84: ~ 38x11
[1] ubuntu@ip-172-31-61-101: ~ 38x11
[1] ubuntu@ip-172-31-52-79: ~ 28
[1] ubuntu@ip-172-31-52-81: ~ 38x11
[1] ubuntu@ip-172-31-52-80: ~ 38x11
[1] ubuntu@ip-172-31-61-102: ~ 38x11
[1] ubuntu@ip-172-31-52-85: ~ 28

```