Cloud Computing

Programming Assignment – 1

- Create a t2.micro instance on Amazon EC2.
- Connect to an instance and send PROG#1Aditya_Kale.tar on amazon instance using scp command.

  chmod 400 aditya.pem

  scp –i "user.pem" –r PROG1#Kale_Aditya.tar.gz ubuntu@52.36.15.23:./

- Connect to an instance and extract above tar file on amazon instance.
  ssh -i aditya.pem ubuntu@"public_ip_of_an_instance"

  tar -xvzf PROG#1Aditya_Kale.tar.gz

- For installing Java, Apache Ant and GCC on amazon instance run installation.sh file.

  ./Installation.sh

- Execute following steps to perform Benchmarking Experiment.

Memory Benchmarking:

- Compile Memory.c using gcc command.

  gcc Memory.c –lpthread.

- Execute a.out for running above complied C program.

  ./a.out
- Following results are captured after Executing Memory Benchmarking.
- Output for Memory Benchmarking

```
ubuntu@ip-172-31-45-63:~$ gcc Memory.c -lpthread
ubuntu@ip-172-31-45-63:~$ ./a.out


Executing Memory Read Write Operation for 1 Thread and of size 1 byte.
Sequential Read Write
Latency : 0.000005 ms
Throughtput:199.835366 MB/s
RANDOM Write
Latency : 0.000000 ms
Throughtput:2735.726668 MB/s


Executing Memory Read Write Operation for 2 Thread and of size 1 byte.
Sequential Read Write
Latency : 0.000005 ms
Throughtput:204.377030 MB/s
RANDOM Write
Latency : 0.000000 ms
Throughtput:3091.827902 MB/s


Executing Memory Read Write Operation for 1 Thread and of size 1024 byte.
Sequential Read Write
Latency : 0.000039 ms
Throughtput:25040.064103 MB/s
RANDOM Write
Latency : 0.000394 ms
Throughtput:2477.956103 MB/s


Executing Memory Read Write Operation for 2 Thread and of size 1024 byte.
Sequential Read Write
Latency : 0.000036 ms
Throughtput:26976.864641 MB/s
RANDOM Write
Latency : 0.000273 ms
Throughtput:3579.118563 MB/s


Executing Memory Read Write Operation for 1 Thread and of size 1048576 byte.
Sequential Read Write
Latency : 0.135100 ms
Throughtput:7401.924500 MB/s
RANDOM Write
```

Disk Benchmarking:

- Compile Disk.c using gcc command.

  gcc Disk.c –lpthread.

- Execute a.out for running above complied C program.

  ./a.out

- Following results are captured after Executing Memory Benchmarking.

- Output for Disk Benchmarking

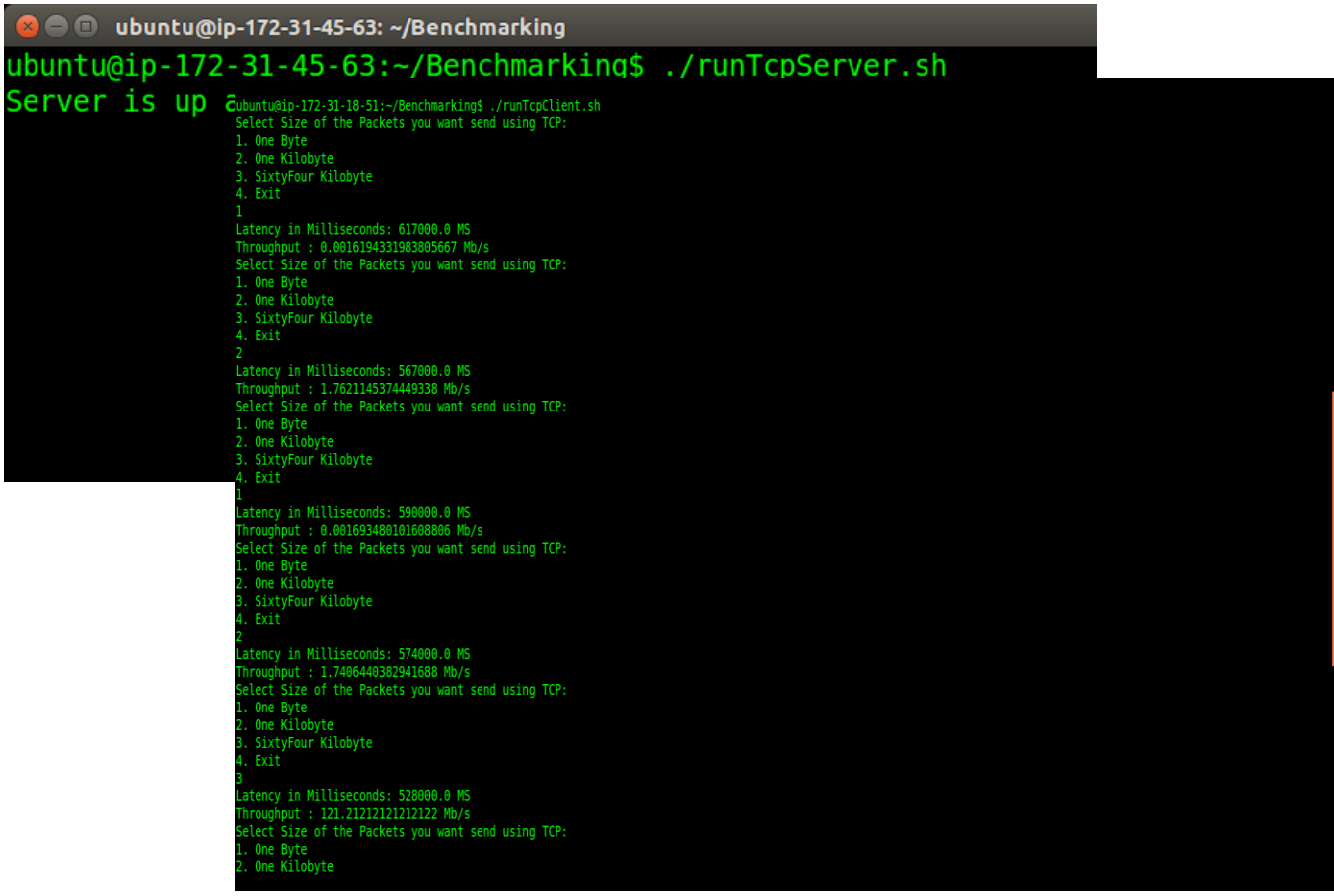
```
ubuntu@ip-172-31-45-63:~$ vi Disk.c
ubuntu@ip-172-31-45-63:~$ gcc Disk.c -lpthread
ubuntu@ip-172-31-45-63:~$ ./a.out


Executing Read Write Operation for 1 Threads and for size 1 byte.
Sequential Write
Latency : 0.000029 ms
Throughtput:32.956940 MB/s
SEQUENTIAL Read
Latency : 0.000031 ms
Throughtput:30.777836 MB/s
RANDOM Write
Latency : 0.001169 ms
Throughtput:0.815745 MB/s
RANDOM Read
Latency : 0.000154 ms
Throughtput:6.211966 MB/s

Executing Read Write Operation for 2 Threads and for size 1 byte.
Sequential Write
Latency : 0.000029 ms
Throughtput:33.281939 MB/s
SEQUENTIAL Read
Latency : 0.000030 ms
Throughtput:31.283113 MB/s
RANDOM Write
Latency : 0.001173 ms
Throughtput:0.813263 MB/s
RANDOM Read
Latency : 0.000154 ms
Throughtput:6.181102 MB/s

Executing Read Write Operation for 1 Threads and for size 1024 byte.
Sequential Write
Latency : 0.000890 ms
Throughtput:1097.347544 MB/s
SEQUENTIAL Read
Latency : 0.000250 ms
Throughtput:3899.854239 MB/s
RANDOM Write
Latency : 0.001833 ms
Throughtput:532.845809 MB/s
RANDOM Read
```

Network Benchmarking:

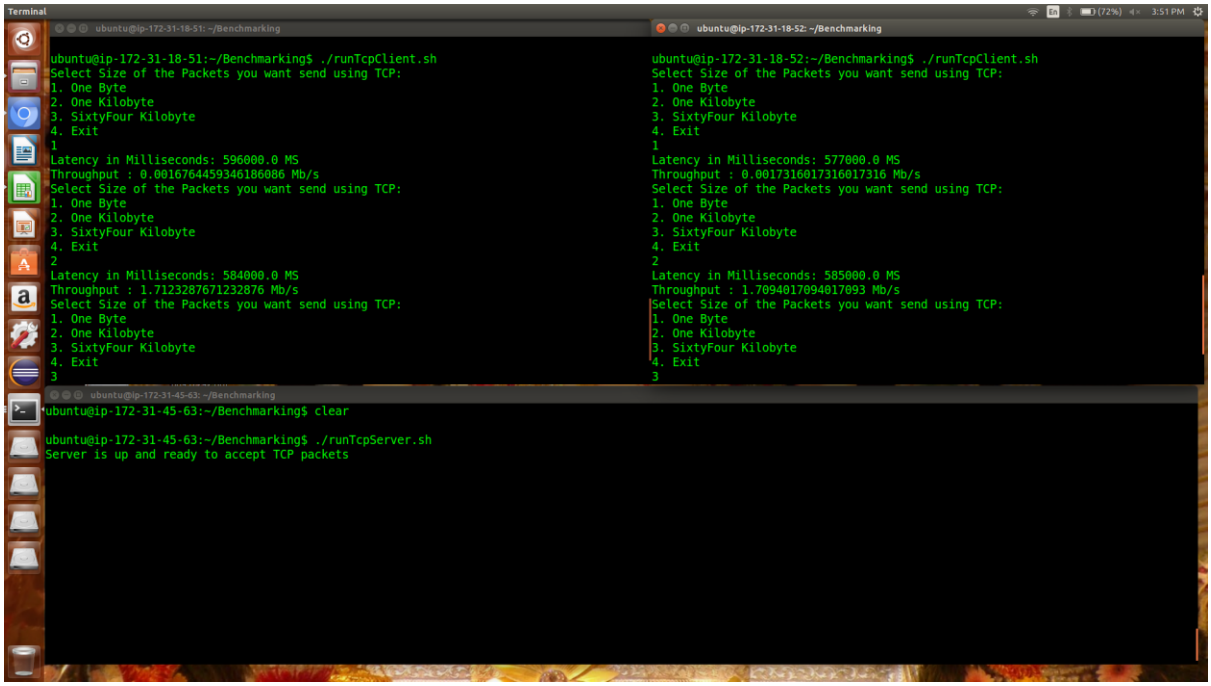
For TCP Packets.

- Run 3 amazon instances.
- Go to java project folder location.
  cd Benchmarking
- Complie code using apacahe ant
  ant
- Change serverIp in configure.properties. Set serverIp value to a private Ip of a server instance.
- Run Server first and then Client using runTcpServer.sh and runTcpClient.sh shell scripts
  ./runTcpServer.sh
  ./runTcpClient.sh
- Give proper inputs on terminal to execute the program.

- Following output is captured after running above program.



For Multiple Clients:



Repeat same procedure for UDP Packets.

- Run 3 amazon instances.
- Used Cluster ssh to execute parallel commands.
- Go to java project folder location.

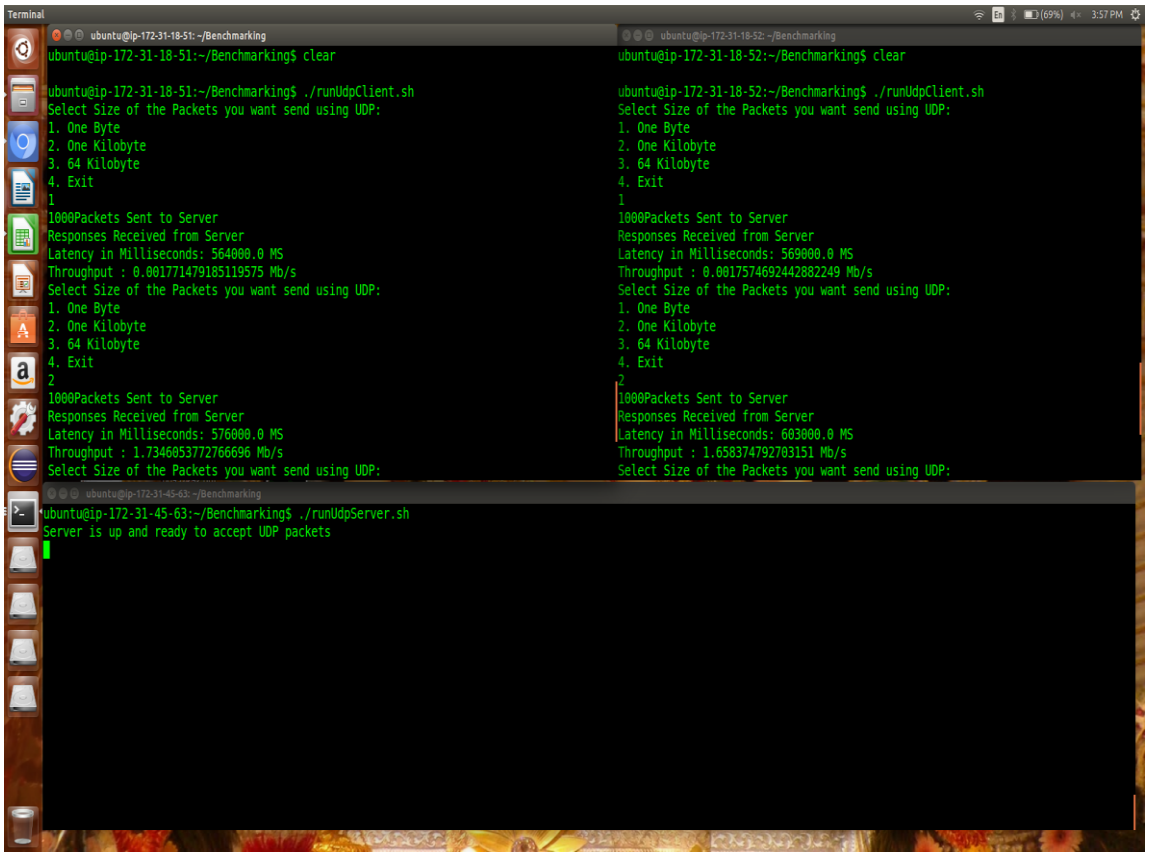
  cd Benchmarking
- Complie code using apacahe ant

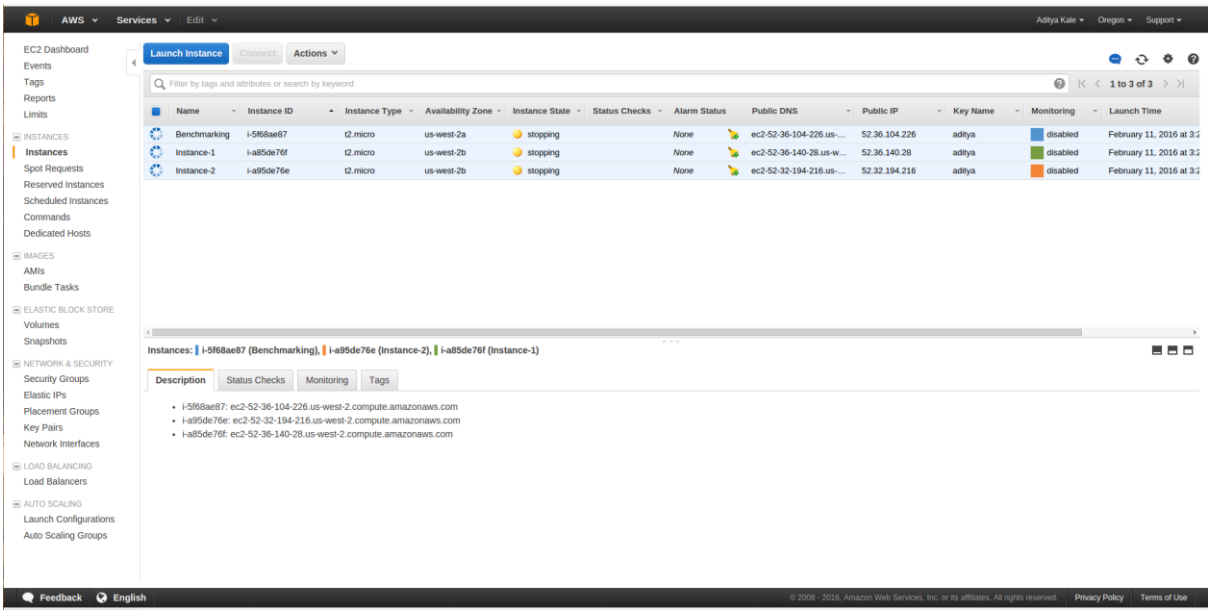
  ant
- Change serverIp in configure.properties. Set serverIp value to a private Ip of a server instance.
- Run Server first and then Client using runUdpServer.sh and runUdpClient.sh shell scripts on three different instances.

  ./runUdpServer.sh

  ./runUdpClient.sh
- Give proper inputs on terminal to execute the program.
- Following output is captured after running above program.

```
ubuntu@ip-172-31-18-51: ~/Benchmarking
ubuntu@ip-172-31-18-51:~/Benchmarking$ ./runUdpClient.sh
Select Size of the Packets you want send using UDP:
1. One Byte
2. One Kilobyte
3. 64 Kilobyte
4. Exit
1
1000Packets Sent to Server
Responses Received from Server
Latency in Milliseconds: 595000.0 MS
Throughput : 0.0016806722689075631 Mb/s
Select Size of the Packets you want send using UDP:
1. One Byte
2. One Kilobyte
3. 64 Kilobyte
4. Exit
2
1000Packets Sent to Server
Responses Received from Server
Latency in Milliseconds: 571000.0 MS
Throughput : 1.7513134851138354 Mb/s
Select Size of the Packets you want send using UDP:
1. One Byte
2. One Kilobyte
3. 64 Kilobyte
4. Exit
3
1000Packets Sent to Server
Responses Received from Server
Latency in Milliseconds: 837000.0 MS
Throughput : 76.41791044776119 Mb/s
Select Size of the Packets you want send using UDP:
1. One Byte
2. One Kilobyte
3. 64 Kilobyte
4. Exit
```




AWS
Services
Edit
Aditya Kale
Oregon
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Spot Requests
Reserved Instances
Scheduled Instances
Commands
Dedicated Hosts
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
LOAD BALANCING
Load Balancers
AUTO SCALING
Launch Configurations
Auto Scaling Groups
Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 3 of 3
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS
Public IP
Key Name
Monitoring
Launch Time
Benchmarking
i-5f68ae87
t2.micro
us-west-2a
stopping
None
ec2-52-36-104-226.us-...
52.36.104.226
aditya
disabled
February 11, 2016 at 3:
Instance-1
i-a85de76f
t2.micro
us-west-2b
stopping
None
ec2-52-36-140-28.us-w...
52.36.140.28
aditya
disabled
February 11, 2016 at 3:
Instance-2
i-a95de76e
t2.micro
us-west-2b
stopping
None
ec2-52-32-194-216.us-...
52.32.194.216
aditya
disabled
February 11, 2016 at 3:
Instances: i-5f68ae87 (Benchmarking), i-a95de76e (Instance-2), i-a85de76f (Instance-1)
Description
Status Checks
Monitoring
Tags
i-5f68ae87: ec2-52-36-104-226.us-west-2.compute.amazonaws.com
i-a95de76e: ec2-52-32-194-216.us-west-2.compute.amazonaws.com
i-a85de76f: ec2-52-36-140-28.us-west-2.compute.amazonaws.com
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use