

Manual

Cloud Computing Assignmnet-1

- This manual contains the steps to execute the experiments.

CPU Experiment

Steps to execute CPU Experimnet

- The script **CPUExperiments.sh** is placed in sourcecode folder
- Grant permission to bash file using command **chmod 777 CPUExperiments.sh**
- Run the script using command **./CPUExperiments.sh** to execute CPU Experiments.
- This script will execute all the 8 Experiments which are needed to be performed.
- First the experiments with varying threads will be executed followed by 600 samples for both FLOPS and IOPS operations using 4 threads.

```
ec2-54-86-46-69.compute-1.amazonaws.com - PuTTY
-bash-4.2$ ./CPUExperiments.sh
The Following Program Will compute CPU FLOPS and IOPS
Time Taken for running with 1 Thread is 5.833 Seconds
Total No Of Giga Flops is 7.5432879511028155

Time Taken for running with 2 Thread is 5.694 Seconds
Total No Of Giga Flops is 7.7274327007634005

Time Taken for running with 4 Thread is 5.568 Seconds
Total No Of Giga Flops is 7.9022990779606515

Time Taken for running with 1 Thread is 5.294 Seconds
Total No Of Giga Iops is 8.311295573007559

Time Taken for running with 2 Thread is 4.114 Seconds
Total No Of Giga Iops is 10.695187572376037

Time Taken for running with 4 Thread is 4.12 Seconds
Total No Of Giga Iops is 10.679611947132184

=====
The Following Program Will Obtain 600 Samples for FLOPS and IOPS

Following are 600 Samples for Floating Point Operations per Second With 4 Threads
7.689618850284136
7.883891985829024
8.018953907740048
7.305329856847511
7.969570862096723
8.016032292535652
8.0704326597411
8.021877653411812
8.035062335401486
8.073394777959239
8.0704326597411
7.95516161770472
8.097165969165816
```

Linpack Benchmark

Steps to execute linpack Benchmark

- Download Linpack from
http://registrationcenter.intel.com/irc_nas/8305/l_mklb_p_11.3.1.002.tgz
- Unzip using command
tar -zxvf l_mklb_p_11.3.1.002.tgz
- Navigate to directory
l_mklb_p_11.3.1.002/benchmarks_11.3.1/linux/mkl/benchmarks/linpack
- Execute command
./runme_xeon64

```
CPU frequency: 2.811 GHz
Number of CPUs: 1
Number of cores: 1
Number of threads: 1

Parameters are set to:

Number of tests: 15
Number of equations to solve (problem size) : 1000 2000 5000 10000 15000 18000 20000 22000 25000 26000 27000 30000 35000 40000 45000
Leading dimension of array : 1000 2000 5008 10000 15000 18008 20016 22008 25000 26000 27000 30000 35000 40000 45000
Number of trials to run : 4 2 2 2 2 2 2 2 2 2 1 1 1 1 1
Data alignment value (in Kbytes) : 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1

Maximum memory requested that can be used=800204096, at the size=10000


----- Timing linear equation system solver -----

Size LDA Align. Time(s) GFlops Residual Residual(norm) Check
1000 1000 4 0.041 16.2250 9.900691e-13 3.376390e-02 pass
1000 1000 4 0.040 16.7715 9.900691e-13 3.376390e-02 pass
1000 1000 4 0.039 17.1183 9.900691e-13 3.376390e-02 pass
1000 1000 4 0.039 17.0314 9.900691e-13 3.376390e-02 pass
2000 2000 4 0.296 18.0411 4.053480e-12 3.526031e-02 pass
2000 2000 4 0.294 18.1604 4.053480e-12 3.526031e-02 pass
5000 5008 4 4.309 19.3503 2.336047e-11 3.257429e-02 pass
5000 5008 4 4.297 19.4045 2.336047e-11 3.257429e-02 pass
```

Disk Experiments

Steps to execute disk experiments

- The script **DISKExperiments.sh** is placed in sourcecode folder
- Grant permission to bash file using command **chmod 777 DISKExperiments.sh**
- Run the script using command **./DISKExperiments.sh** to execute disk experiments
- This script will execute all the 24 Experiments which are needed to be performed

 ec2-54-86-46-69.compute-1.amazonaws.com - PuTTY

```
-bash-4.2$ ./DISKExperiments.sh
The following program will compute Sequential Disk Write Operations
Time 64.493 Seconds
No of Threads 1
Buffer Size 1
Total Data Write is 50.0 MB
Throughput = 0.775278 MB/S
Latency = 1.2301062E-6 Seconds

Time 84.985 Seconds
No of Threads 1
Buffer Size 1024
Total Data Write is 5120.0 MB
Throughput = 60.245926 MB/S
Latency = 1.6209602E-5 Seconds

Time 84.856 Seconds
No of Threads 1
Buffer Size 1048576
Total Data Write is 5120.0 MB
Throughput = 60.33751 MB/S
Latency = 0.016573438 Seconds

Time 48.808 Seconds
No of Threads 2
Buffer Size 1
Total Data Write is 50.0 MB
Throughput = 1.0244223 MB/S
Latency = 9.309387E-7 Seconds
```

1) Disk – Sequential Write Snapshot

ec2-54-86-46-69.compute-1.amazonaws.com - PuTTY

```
No of Threads 1
Buffer Size 1
Total Data Write is 50.0 MB
Throughput = 0.76655376 MB/S
Latency = 1.2441062E-6 Seconds
```

```
Time 85.053 Seconds
No of Threads 1
Buffer Size 1024
Total Data Write is 5120.0 MB
Throughput = 60.197758 MB/S
Latency = 1.6222573E-5 Seconds
```

```
Time 84.84 Seconds
No of Threads 1
Buffer Size 1048576
Total Data Write is 5120.0 MB
Throughput = 60.348896 MB/S
Latency = 0.016570311 Seconds
```

```
Time 49.591 Seconds
No of Threads 2
Buffer Size 1
Total Data Write is 50.0 MB
Throughput = 1.0082475 MB/S
Latency = 9.4587324E-7 Seconds
```

```
Time 42.761 Seconds
No of Threads 2
Buffer Size 1024
Total Data Write is 5120.0 MB
Throughput = 119.73527 MB/S
Latency = 8.156014E-6 Seconds
```

```
Time 42.553 Seconds
No of Threads 2
Buffer Size 1048576
Total Data Write is 5120.0 MB
Throughput = 120.32053 MB/S
Latency = 0.008311133 Seconds
```

2) Disk – Sequential Read Snapshot

```
Time 329.035
No of Threads 1
Buffer Size 1
Total Data Read is 1024.0 MB
Throughput = 3.1121309 MB/S
Latency = 3.0643773E-7 seconds
```

```
Time 16.252
No of Threads 1
Buffer Size 1024
Total Data Read is 1024.0 MB
Throughput = 63.007626 MB/S
Latency = 1.5499116E-5 seconds
```

```
Time 17.027
No of Threads 1
Buffer Size 1048576
Total Data Read is 1024.0 MB
Throughput = 60.13978 MB/S
Latency = 0.01662793 seconds
```

```
Time 664.57
No of Threads 2
Buffer Size 1
Total Data Read is 2048.0 MB
Throughput = 3.081692 MB/S
Latency = 3.0946453E-7 seconds
```

```
Time 16.119
No of Threads 2
Buffer Size 1024
Total Data Read is 2048.0 MB
Throughput = 127.05503 MB/S
Latency = 7.686138E-6 seconds
```

```
Time 17.259
No of Threads 2
Buffer Size 1048576
Total Data Read is 2048.0 MB
Throughput = 118.66272 MB/S
Latency = 0.0084272465 seconds
```

3) Disk – Random Write Snapshot

```
Time 0.037Seconds
No of Threads 1
Buffer Size 1
Total Data Write is 9.536743E-4 MB
Throughput = 0.025774982 MB/S
Latency = 3.7E-5 seconds

Time 0.026Seconds
No of Threads 1
Buffer Size 1024
Total Data Write is 0.9765625 MB
Throughput = 37.560097 MB/S
Latency = 2.6E-5 seconds


Time 14.004Seconds
No of Threads 1
Buffer Size 1048576
Total Data Write is 1000.0 MB
Throughput = 71.40817 MB/S
Latency = 0.0140039995 seconds

Time 0.026Seconds
No of Threads 2
Buffer Size 1
Total Data Write is 9.536743E-4 MB
Throughput = 0.036679782 MB/S
Latency = 2.6E-5 seconds

Time 0.046Seconds
No of Threads 2
Buffer Size 1024
Total Data Write is 0.9765625 MB
Throughput = 21.22962 MB/S
Latency = 4.6E-5 seconds

Time 16.084Seconds
No of Threads 2
Buffer Size 1048576
Total Data Write is 1000.0 MB
Throughput = 62.17359 MB/S
Latency = 0.016084 seconds
```

4) Disk – Random Read Snapshot

 ec2-54-86-46-69.compute-1.amazonaws.com - PuTTY

```
The following program will compute Random read operations
Time 30.977
No of Threads 1
Buffer Size 1
Total Data Read is 50.0 MB
Throughput = 1.6141008 MB/S
Latency = 5.9083936E-7 Seconds

Time 143.064
No of Threads 1
Buffer Size 1024
Total Data Read is 1024.0 MB
Throughput = 7.157636 MB/S
Latency = 1.3643646E-4 Seconds

Time 8.784
No of Threads 1
Buffer Size 1048576
Total Data Read is 1024.0 MB
Throughput = 116.575584 MB/S
Latency = 0.008578125 Seconds

Time 60.648
No of Threads 2
Buffer Size 1
Total Data Read is 50.0 MB
Throughput = 1.648859 MB/S
Latency = 5.783844E-7 Seconds

Time 220.15
No of Threads 2
Buffer Size 1024
Total Data Read is 1024.0 MB
Throughput = 9.302749 MB/S
Latency = 1.049757E-4 Seconds

Time 15.305
No of Threads 2
Buffer Size 1048576
Total Data Read is 1024.0 MB
Throughput = 133.81247 MB/S
Latency = 0.0074731447 Seconds
```

IOZONE Benchmark

Steps to execute IOZONE Benchmark

- Download IOZONE from
<http://www.iozone.org/src/current/iozone-3-434.i386.rpm>
- Unzip using command
tar -xvf iozone3_434.tar
- Navigate to directory
/iozone3_434/src/current
- Execute command to perform all the experiments
./iozone -a

```
et2-54-86-46-68.compute-1.amazonaws.com - PuTTY
-bash-4.2$ ./iozone
Usage: For usage information type iozone -h

-bash-4.2$ ./iozone -a
Iozone: Performance Test of File I/O
Version & Revision: 3.434 F
Compiled for 64 bit mode.
Build: linux

Contributors: William Norcott, Don Capps, Isam Crawford, Kirby Collins
Al Slater, Scott Rhine, Mike Wisner, Ken Goss
Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain Cyr,
Randy Dunlap, Mark Montague, Dan Millson, Gavin Brahmner,
Jean-Marc Luccioni, Jeff Blumberg, Benny Salevy, Dave Boontz,
Erik Hobbings, Kris Strecker, Walter Wong, Joshua Root,
Fabrice Bacchella, Shanghua Xue, Qin Li, Darsen Sawyer,
Vangel Bojakhii, Ben England, Vikentii Lapa,
Alexey Skidanov.

Run began: Thu Feb 11 23:48:07 2016

Auto Mode
Command line used: ./iozone -a
Output is in kbytes/sec
Time Resolution = 0.000001 seconds.
Processor cache size set to 1024 kbytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.

      kb  random  random  block  record  stride
      kb  read    write    read    write    read    read    write    read    write
64      4  1105556  2067979  3735102  15972885  10402176  2361267  4564786  4018152  9006179  2298136  2278628  48
64      8  1138375  1599680  7100397  20962191  15972885  2662899  5389653  4897940  10402170  2561267  2467100  64
64     16  1385075  2379626  3363612  15972885  12902017  2379626  3022727  4018152  1363612  3057153  2892445  64
64     32  1392258  1484662  3751156  20962191  15972885  1933893  6421025  3791156  9318832  1967927  3738358  32
64     64  1599680  2561267  15972885  20962191  15972885  1828508  6421025  1679761  2923952  2467100  2561267  79
128     4  1391557  1732927  6406138  14208794  10779367  1905110  3199360  4596273  8036304  1997245  1705404  41
```

- Execute the command `./iozone -a -i 0 -i 1 -i 2 -s 2097152 -r 1024` to test iozone using 2 GB file and 1 MB buffer.

```
ec2-54-86-46-69.compute-1.amazonaws.com - PyTTY
-bash-4.2$ ./iozone -a -i 0 -i 1 -i 2 -s 2097152 -r 1024
iozone: Performance Test of File I/O
Version $Revision: 3.434 $
Compiled for 64 bit mode.
Build: linux

Contributors: William Morcott, Don Capps, Isom Crawford, Kirby Collins
Al Slater, Scott Rhine, Mike Wisner, Ken Goss
Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
Bandy Dunlap, Mark Montague, Dan Million, Gavin Brehner,
Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
Erik Hobbings, Kris Strecker, Walter Wong, Joshua Root,
Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
Vangel Bojaxhi, Ben England, Vikentsi Lapa,
Alexey Skidanov.

Run began: Fri Feb 12 00:30:08 2016

Auto Mode
File size set to 2097152 kB
Record Size 1024 kB
Command line used: ./iozone -a -i 0 -i 1 -i 2 -s 2097152 -r 1024
Output is in Kbytes/sec
Time Resolution = 0.000001 seconds.
Processor cache size set to 1024 kbytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.

      kB  reclen  write  rewrite  read  reread  read  random  random  blkwd  record  stride
      2097152  1024  62618  63367  43194  57759  55930  66243  read  rewrite  read  fwrite  frewrite

iozone test complete.
-bash-4.2$
-bash-4.2$
-bash-4.2$
-bash-4.2$
```


Network Experiments

Steps to execute Network experiments

TCP

For Running Server

- First Start the server.
- Type Command **java Network_Tcp_Server** to start the server.
- After starting the server console will ask whether to test on 1 Thread or 2 Threads.
- Once the number of threads is provided console will wait for client to connect.
- The server needs to be started again for testing with different thread counts due to port confliction.

```
ec2-user@ip-172-31-51-156:~/Cloud
[ec2-user@ip-172-31-51-156 Cloud]$ java Network_Tcp_Server
Please Number of threads 1 or 2
2
Socket has been Created
Waiting for Connection
```

For Running Client

- Before running client please start the server.
- Type command **java Network_Tcp_Client** to start the client.
- After starting the client the console will ask the IP address of the server to which connection needs to be made.
- After providing IP address console will ask whether to test using 1 Thread or 2 Threads , enter the same number of threads provided while starting the server
- After providing thread count client will perform experiments for all 3 Block sizes.
- The client needs to be started again for testing with different thread counts due to port confliction.

```
ec2-54-86-46-69.compute-1.amazonaws.com - PuTTY
-bash-4.2$ java Network_Tcp_Client
Please Enter IP address of Server you want to Communicate with
172.31.51.156
Please Enter no of Threads 1 or 2
2
Communicating With Thread 1 Thread 0
```


Results obtained using 1 Thread and varying block sizes – Tcp

```
Communication With Thread : Thread-0
Recieved Packet Size: 1
Time 0.011
No of Threads 1
Buffer Size 1
Throughput = 0.010181818 Mb/S

Communication With Thread : Thread-1
Recieved Packet Size: 1024
Time 0.004
No of Threads 1
Buffer Size 1024
Throughput = 28.671999 Mb/S

Communication With Thread : Thread-2
Recieved Packet Size: 65536
Time 0.018
No of Threads 1
Buffer Size 65536
Throughput = 407.77954 Mb/S
```

Results obtained using 2 Thread and varying block sizes – Tcp

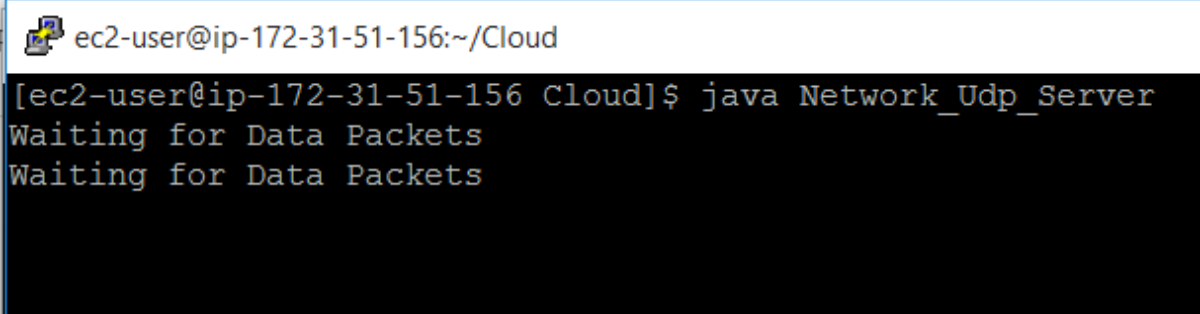
```
ec2-54-86-46-69.compute-1.amazonaws.com - PuTTY

Communicating With Thread : Thread-0
Recieved Packet Size: 1
Communicating With Thread : Thread-1
Recieved Packet Size: 1
Time 0.016
No of Threads 2
Buffer Size 1
Throughput = 0.01 Mb/S
Communicating With Thread : Thread-2
Recieved Packet Size: 1024
Communicating With Thread : Thread-3
Recieved Packet Size: 1024
Time 0.005
No of Threads 2
Buffer Size 1024
Throughput = 32.768 Mb/S
Communicating With Thread : Thread-4
Recieved Packet Size: 65536
Communicating With Thread : Thread-5
Recieved Packet Size: 65536
Time 0.018
No of Threads 2
Buffer Size 65536
Throughput = 582.54224 Mb/S
```

UDP

For Running Server

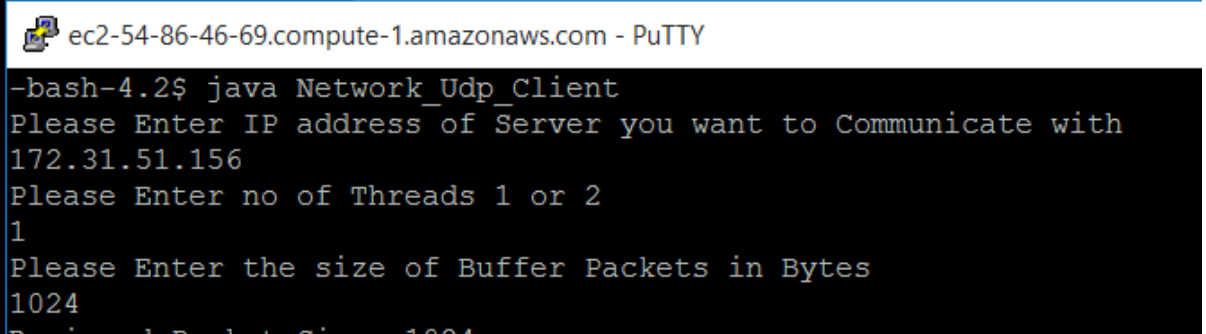
- First Start the server.
- Type Command **java Network_Udp_Server** to start the server.
- Once the server is started it will wait for client to connect.
- Once one experiment is completed the server needs to be restarted due to port confliction.



```
ec2-user@ip-172-31-51-156:~/Cloud
[ec2-user@ip-172-31-51-156 Cloud]$ java Network_Udp_Server
Waiting for Data Packets
Waiting for Data Packets
```

For Running Client

- Before running client please start the server.
- Type Command **java Network_Udp_Client** to start the client.
- After starting the client the console will ask the IP address of the server to which connection needs to be made.
- After providing IP address console will ask whether to test using 1 Thread or 2 Threads , provide the number of threads.
- The console will ask for the buffer size on which testing needs to be done, please provide the buffer packet size.



```
ec2-54-86-46-69.compute-1.amazonaws.com - PuTTY
-bash-4.2$ java Network_Udp_Client
Please Enter IP address of Server you want to Communicate with
172.31.51.156
Please Enter no of Threads 1 or 2
1
Please Enter the size of Buffer Packets in Bytes
1024
Received Packet Size: 1024
```

Results obtained using 1 Thread and varying block sizes – UDP

```
Recieved Packet Size: 1
Time 0.006 ms
No of Threads 1
Buffer Size 1
Throughput = 0.02 Mb/s

Recieved Packet Size: 1024
Time 0.03 ms
No of Threads 1
Buffer Size 1024
Throughput = 30.67 Mb/s

Recieved Packet Size: 63488
Time 2.1 ms
No of Threads 1
Buffer Size 63488
Throughput = 410.37181 Mb/s
```

Results obtained using 2 Thread and varying block sizes – UDP

```
Recieved Packet Size: 1
Recieved Packet Size: 1
Time 0.046 ms
No of Threads 2
Buffer Size 1
Throughput = 0.03 Mb/s

Recieved Packet Size: 1024
Recieved Packet Size: 1024
Time 0.025 ms
No of Threads 2
Buffer Size 1024
Throughput = 63.67 Mb/s

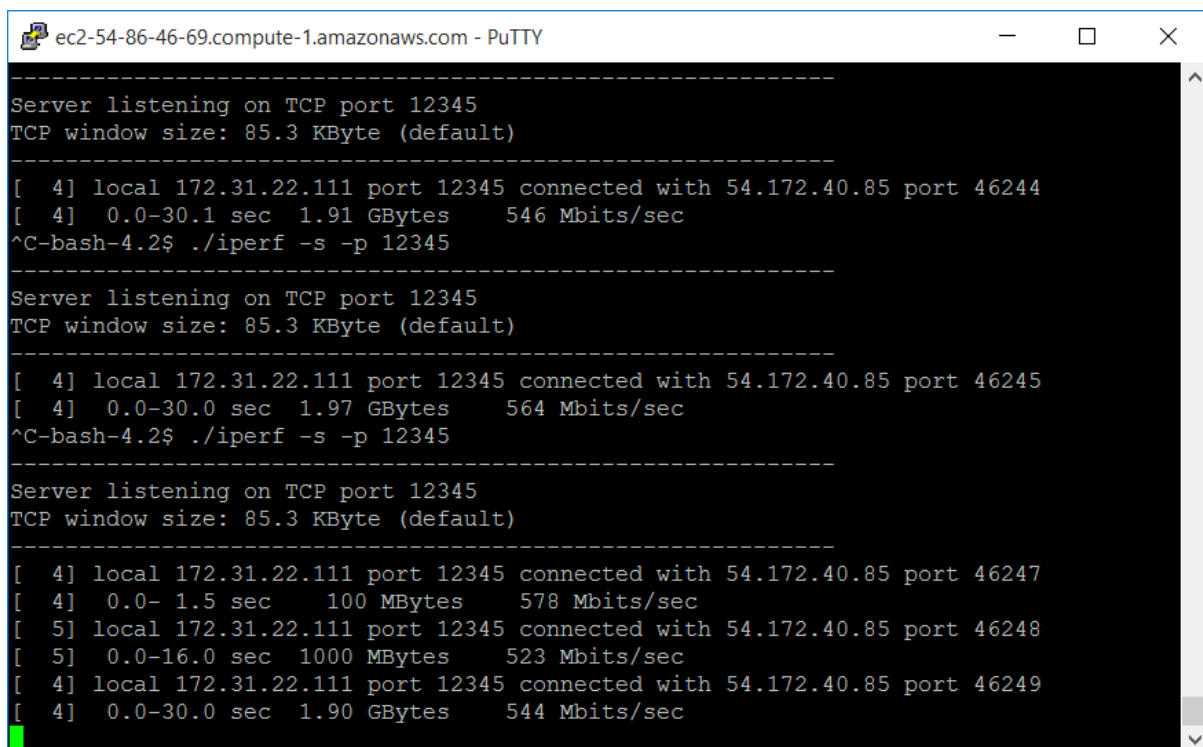
Recieved Packet Size: 63488
Recieved Packet Size: 63488
Time 1.8 ms
No of Threads 2
Buffer Size 63488
Throughput = 645.37181 Mb/s
```

IPERF Benchmark

Steps to execute IPERF Benchmark

- Download IPERF from
https://iperf.fr/download/iperf_2.0.2/iperf_2.0.2-4_amd64.tar.gz
- Unzip using command
tar -zxvf iperf_2.0.2-4_amd64.tar.gz
- Navigate to directory
home/ec2-user/Cloud/iperf_2.0.2-4_amd64
- Execute command
./iperf -s -p 12345

This command starts the server instance with port 12345

A screenshot of a PuTTY terminal window titled "ec2-54-86-46-69.compute-1.amazonaws.com - PuTTY". The terminal displays the output of the IPERF server running on TCP port 12345. It shows three separate test runs, each with a local client at 172.31.22.111 and a remote client at 54.172.40.85. The first test shows a connection on port 46244 with 1.91 GBytes transferred at 546 Mbits/sec. The second test shows a connection on port 46245 with 1.97 GBytes transferred at 564 Mbits/sec. The third test shows two connections: one on port 46247 (100 MBytes at 578 Mbits/sec) and another on port 46248 (1000 MBytes at 523 Mbits/sec), followed by a connection on port 46249 (1.90 GBytes at 544 Mbits/sec).

```
-----  
Server listening on TCP port 12345  
TCP window size: 85.3 KByte (default)  
-----  
[ 4] local 172.31.22.111 port 12345 connected with 54.172.40.85 port 46244  
[ 4] 0.0-30.1 sec 1.91 GBytes 546 Mbits/sec  
^C-bash-4.2$ ./iperf -s -p 12345  
-----  
Server listening on TCP port 12345  
TCP window size: 85.3 KByte (default)  
-----  
[ 4] local 172.31.22.111 port 12345 connected with 54.172.40.85 port 46245  
[ 4] 0.0-30.0 sec 1.97 GBytes 564 Mbits/sec  
^C-bash-4.2$ ./iperf -s -p 12345  
-----  
Server listening on TCP port 12345  
TCP window size: 85.3 KByte (default)  
-----  
[ 4] local 172.31.22.111 port 12345 connected with 54.172.40.85 port 46247  
[ 4] 0.0- 1.5 sec 100 MBytes 578 Mbits/sec  
[ 5] local 172.31.22.111 port 12345 connected with 54.172.40.85 port 46248  
[ 5] 0.0-16.0 sec 1000 MBytes 523 Mbits/sec  
[ 4] local 172.31.22.111 port 12345 connected with 54.172.40.85 port 46249  
[ 4] 0.0-30.0 sec 1.90 GBytes 544 Mbits/sec
```