

**Chameleon Instance:** compute\_haswell\_ib at CHI@TACC, Instance Name : sg\_instance

**CPU:** 2x Intel® Xeon® E5-2670 v3 @2.30GHz

**Memory:** 8x 16GB (128GB) of DDR4-2,133 ECC Registered RAM

**Disk:** 1x Seagate ST9250610NS SATA 7,200 RPM HDD

**Network:** Broadcom NetXtreme II BCM57800 1/10 Gigabit Ethernet

Launching chameleon bare metal instance after the setup.

```
cc@ubuntu:~$ exit
logout
Connection to 129.114.109.140 closed.

C:\Users\sonas>ssh cc@129.114.109.140
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-60-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed Feb  7 05:56:06 UTC 2024

System load:  0.2568359375      Temperature:   48.0 C
Usage of /:   2.2% of 217.70GB  Processes:    535
Memory usage: 1%               Users logged in: 1
Swap usage:   0%               IPv4 address for eno1: 10.52.2.96

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

*** System restart required ***
Last login: Wed Feb  7 05:39:05 2024 from 208.59.155.102
cc@ubuntu:~$
```

Initial Configuration before virtual machine and container setup.

```
cc@ubuntu:~$ sudo lxd init
Would you like to use LXD clustering? (yes/no) [default=no]: no
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (cephobject, dir, lvm, zfs, btrfs, ceph) [default=zfs]:
Create a new ZFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]:
Size in GiB of the new loop device (1GiB minimum) [default=30GiB]: 150GiB
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=lxdbr0]:
What IPv4 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]:
What IPv6 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]:
Would you like the LXD server to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:
cc@ubuntu:~$ sudo ufw allow in on lxdbr0
Rule added
Rule added (v6)
cc@ubuntu:~$ sudo ufw route allow in on lxdbr0
Rule added
Rule added (v6)
cc@ubuntu:~$ sudo ufw route allow out on lxdbr0
Rule added
Rule added (v6)
cc@ubuntu:~$
```

Creating Virtual Machine inside chameleon instance:

```
cc@sg-instance:~$ sudo lxc launch images:ubuntu/22.04 vm1 --vm -c limits.cpu=4 -c limits.memory=4GiB --device root,size=150GiB
Creating vm1
Starting vm1
cc@sg-instance:~$
```

Accessing the VM :

```
cc@sg-instance:~$ sudo lxc shell vm1
root@vm1:~#
```

Creation container inside chameleon instance:

```
cc@sg-instance:~$ sudo lxc launch images:ubuntu/22.04 container1 -c limits.cpu=4 -c limits.memory=4GiB --device root,size=150GiB
Creating container1
Starting container1
cc@sg-instance:~$
```

Accessing container :

```
cc@sg-instance:~$ sudo lxc shell container1
root@container1:~#
```

List of VM and container running on the machine

```
cc@sg-instance:~$ sudo lxc list
+-----+-----+-----+-----+-----+-----+
| NAME   | STATE | IPV4   | IPV6   | TYPE   | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| container1 | RUNNING | 10.57.62.110 (eth0) | fd42:9272:ecb6:a656:216:3eff:fe39:be9b (eth0) | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+
| vm1     | RUNNING | 10.57.62.21 (enp5s0) | fd42:9272:ecb6:a656:216:3eff:fe9e:c3aa (enp5s0) | VIRTUAL-MACHINE | 0 |
+-----+-----+-----+-----+-----+-----+
```



CPU Benchmark Table:

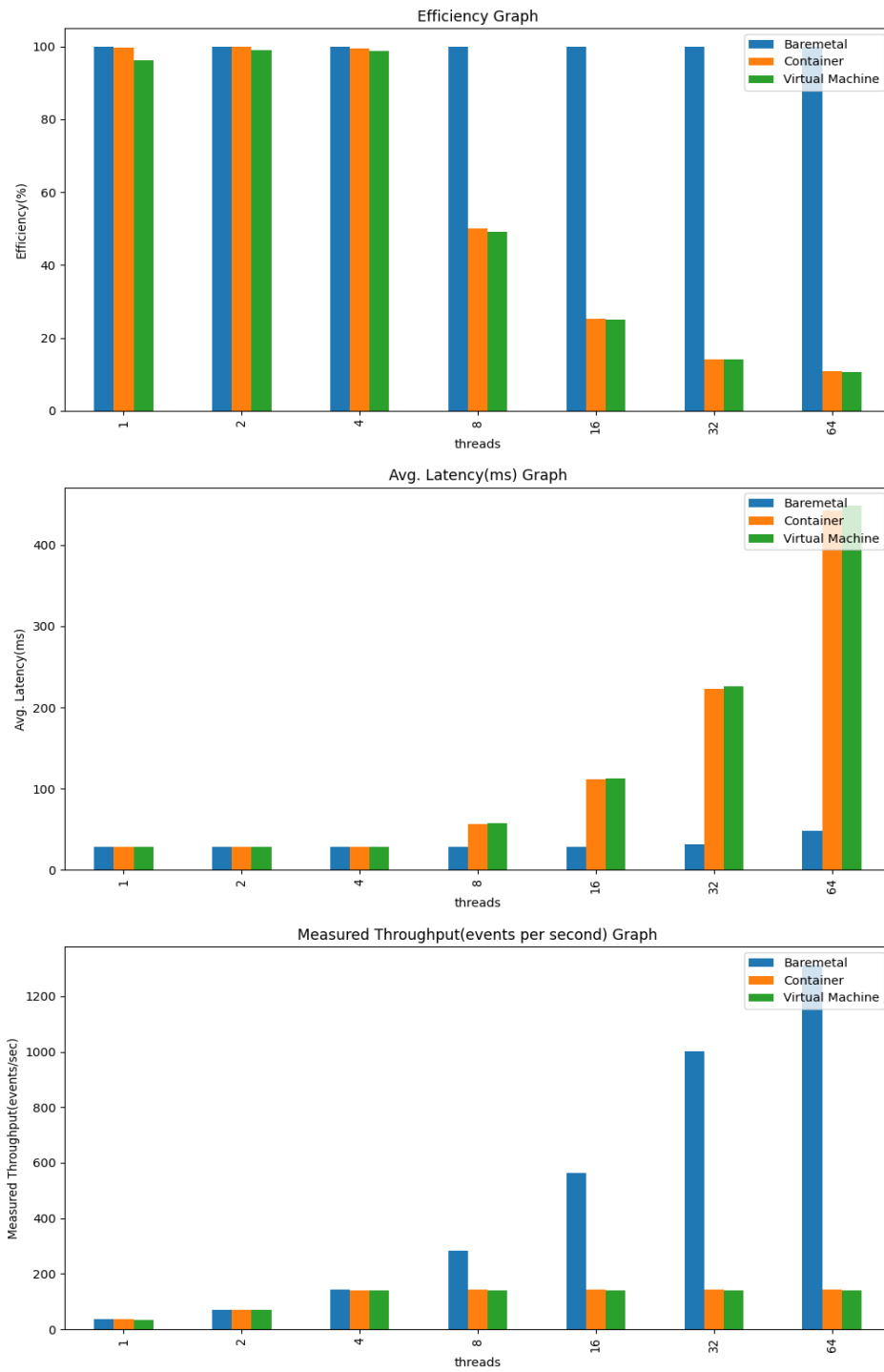
Virtualization Type	Threads	Avg. Latency (ms)	Measured Throughput(Events per Second)	Efficiency
Baremetal	1	28.10	35.57	100%
Container	1	28.22	35.42	99.58%
Virtual Machine	1	28.38	34.22	96.21%
Baremetal	2	28.11	71.13	100%
Container	2	28.14	71.03	99.86%
Virtual Machine	2	28.41	70.31	98.87
Baremetal	4	28.10	142.20	100%
Container	4	28.30	141.13	99.39%
Virtual Machine	4	28.52	140.15	98.64%
Baremetal	8	28.12	284.10	100%
Container	8	56.18	142.07	49.95%
Virtual Machine	8	57.15	139.57	49.07%
Baremetal	16	28.40	562.49	100%
Container	16	111.99	142.21	25.29%
Virtual Machine	16	113.09	140.72	25.02%
Baremetal	32	31.92	1000.66	100%
Container	32	222.95	142.05	14.20%
Virtual Machine	32	225.97	140.50	14.05%
Baremetal	64	48.57	1313.65	100%
Container	64	442.19	142.16	10.82%
Virtual Machine	64	448.25	140.41	10.69%

Efficiency is calculated as below:

Efficiency: (Throughputs from virtualization types (VM/Container)/ Throughput (Bare Metal)) \*100

## CPU Benchmark Graph:

### CPU Benchmark Results





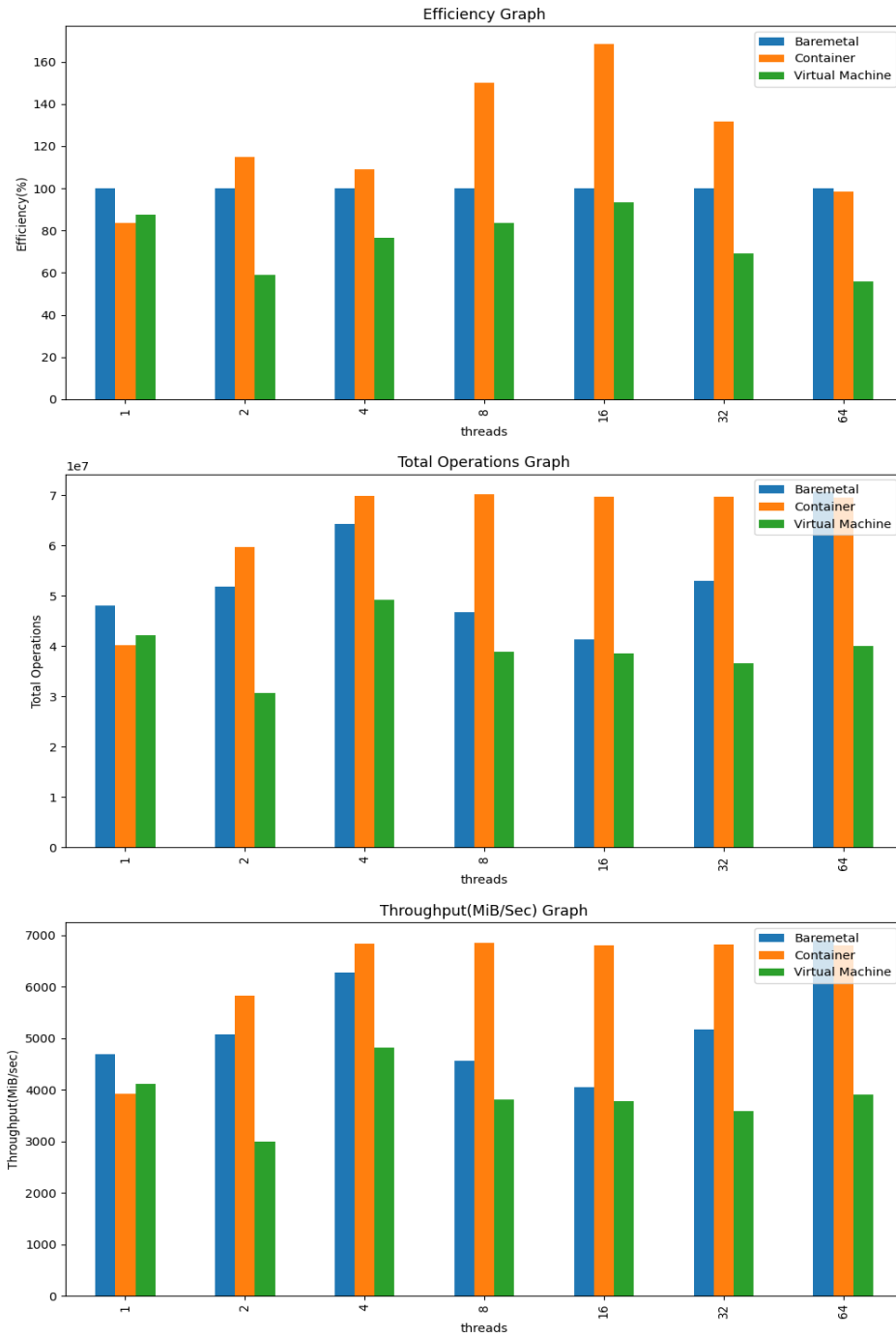
Virtualization Type	Threads	Block Size (KB)	Operation	Access Pattern	Total Operations	Throughput (MiB/Sec)	Efficiency
Baremetal	1	1	Read	Random	48071336	4693.11	100%
Container	1	1	Read	Random	40171325	3921.80	83.50%
Virtual Machine	1	1	Read	Random	42186319	4117.78	87.69%
Baremetal	2	1	Read	Random	51907593	5067.62	100%
Container	2	1	Read	Random	59707204	5829.06	114.98%
Virtual Machine	2	1	Read	Random	30689825	2995.61	59.18%
Baremetal	4	1	Read	Random	64279635	6275.59	100%
Container	4	1	Read	Random	69969765	6831.42	108.88%
Virtual Machine	4	1	Read	Random	49282398	4811.58	76.69%
Baremetal	8	1	Read	Random	46750516	4564.14	100%
Container	8	1	Read	Random	70164429	6850.56	149.89%
Virtual Machine	8	1	Read	Random	39002606	3808.08	83.45%
Baremetal	16	1	Read	Random	41420717	4043.82	100%
Container	16	1	Read	Random	69730501	6808.14	168.61%
Virtual Machine	16	1	Read	Random	38636366	3772.24	93.29%
Baremetal	32	1	Read	Random	52968302	5171.43	100%
Container	32	1	Read	Random	69812426	6816.10	131.73%
Virtual Machine	32	1	Read	Random	36686249	3581.86	69.26%
Baremetal	64	1	Read	Random	70704561	6903.14	100%
Container	64	1	Read	Random	69604425	6795.70	98.44%
Virtual Machine	64	1	Read	Random	40042492	3909.46	56%

Efficiency is calculated as below:

Efficiency: (Throughputs from virtualization types (VM/Container)/ Throughput (Bare Metal)) \*100

# Memory Benchmark Graph

Memory Benchmark Results











```
root@vml1:~# echo sysbench fileio --file-num=128 --file-block-size=4096 --file-total-size=120G --file-test-mode=rndrd --file-io-mode=sync --file-extra-flags=direct --threads=64 run >> chameleonVMLogs_disk.log
root@vml1:~# sysbench fileio --file-num=128 --file-block-size=4096 --file-total-size=120G --file-test-mode=rndrd --file-io-mode=sync --file-extra-flags=direct --threads=64 run >> chameleonVMLogs_disk.log
root@vml1:~# echo sysbench fileio --file-num=128 --file-block-size=4096 --file-total-size=120G --file-test-mode=rndrd --file-io-mode=sync --file-extra-flags=direct --threads=64 cleanup >> chameleonVMLogs_disk.log
root@vml1:~# sysbench fileio --file-num=128 --file-block-size=4096 --file-total-size=120G --file-test-mode=rndrd --file-io-mode=sync --file-extra-flags=direct --threads=64 cleanup >> chameleonVMLogs_disk.log
```

### Disk Benchmark Table:

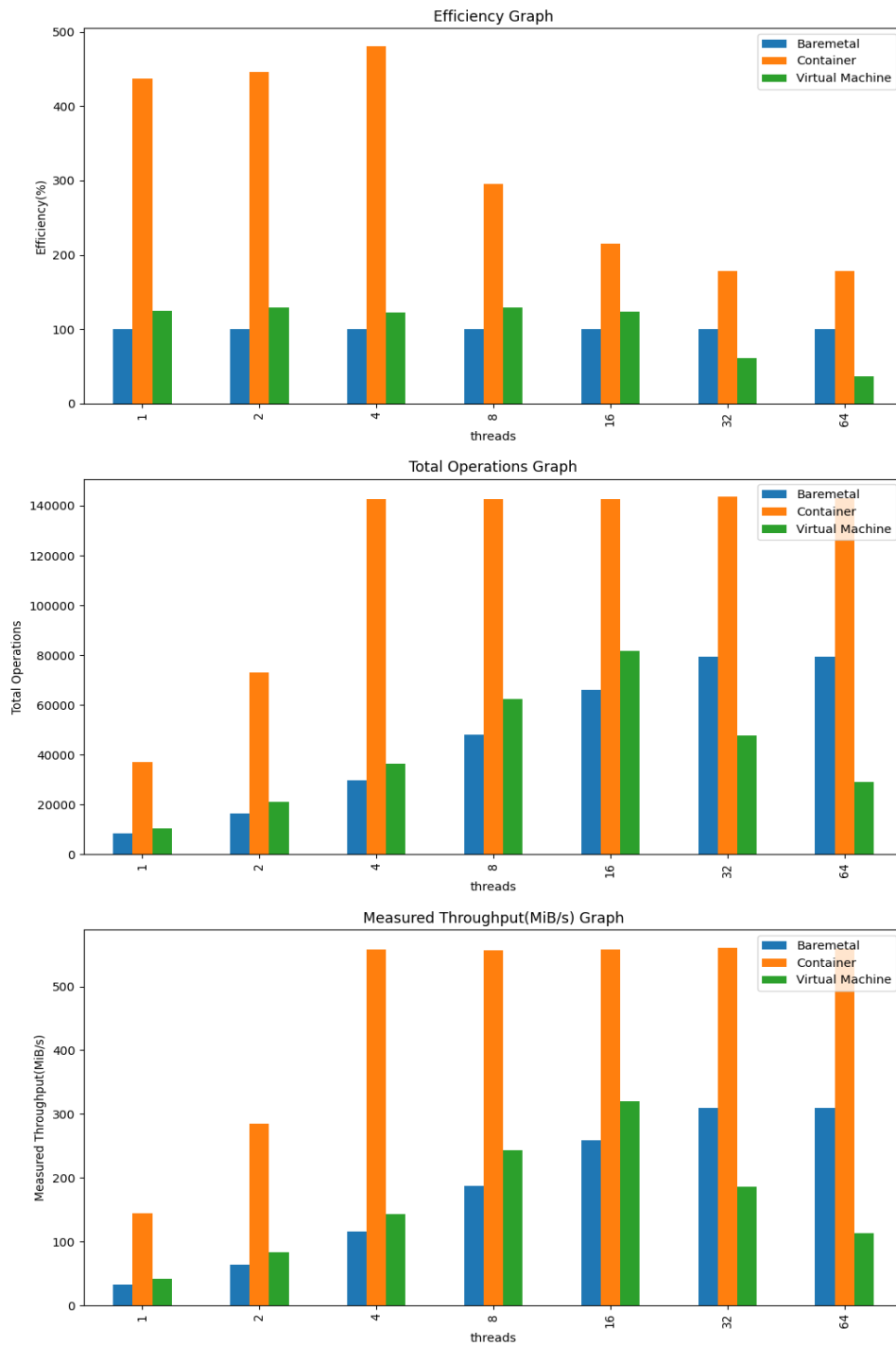
Virtualization Type	Threads	Block Size (KB)	Operation	Access Pattern	I/O mode	I/O flag	Total Operations	Measured Throughput (MiB/s)	Efficiency
Baremetal	1	4	Read	Random	SYNC	DirectIO	8472.14	33.09	100%
Container	1	4	Read	Random	SYNC	DirectIO	36995.53	144.51	436.64%
Virtual Machine	1	4	Read	Random	SYNC	DirectIO	10533.74	41.15	124.36%
Baremetal	2	4	Read	Random	SYNC	DirectIO	16385.59	64.01	100%
Container	2	4	Read	Random	SYNC	DirectIO	73021.00	285.24	445.99%
Virtual Machine	2	4	Read	Random	SYNC	DirectIO	2119:7.93	82.80	129.66%
Baremetal	4	4	Read	Random	SYNC	DirectIO	29730.09	116.13	100%
Container	4	4	Read	Random	SYNC	DirectIO	142774.52	557.71	481.03%
Virtual Machine	4	4	Read	Random	SYNC	DirectIO	36511.00	142.62	122.86%
Baremetal	8	4	Read	Random	SYNC	DirectIO	47989.17	187.46	100%
Container	8	4	Read	Random	SYNC	DirectIO	142558.80	556.87	295.98%
Virtual Machine	8	4	Read	Random	SYNC	DirectIO	62340.65	243.52	129.57%
Baremetal	16	4	Read	Random	SYNC	DirectIO	66110.73	258.25	100%
Container	16	4	Read	Random	SYNC	DirectIO	142803.90	557.83	215.54%
Virtual Machine	16	4	Read	Random	SYNC	DirectIO	81807.83	319.56	123.39%
Baremetal	32	4	Read	Random	SYNC	DirectIO	79348.79	309.96	100%
Container	32	4	Read	Random	SYNC	DirectIO	143622.23	561.02	178.18%
Virtual Machine	32	4	Read	Random	SYNC	DirectIO	47705.30	186.35	60.88%
Baremetal	64	4	Read	Random	SYNC	DirectIO	79291.55	309.73	100%
Container	64	4	Read	Random	SYNC	DirectIO	143077.77	558.90	178.53%
Virtual Machine	64	4	Read	Random	SYNC	DirectIO	28982.09	113.21	36.96%

Efficiency is calculated as below:

Efficiency: (Throughputs from virtualization types (VM/Container)/ Throughput (Bare Metal)) \*100

## Disk Benchmark Graph:

Disk Benchmark Results





Network Benchmark Table:

Virtualization Type	Server	Client Threads	Latency (ms)	Measured Throughput (Gbits/s)	Efficiency
Baremetal	1	1	2.91	29.8	100%
Container	1	1	2.56	33.9	113.42%
Virtual Machine	1	1	5.2	16.7	56.04%
Baremetal	1	2	2.96	58.7	100%
Container	1	2	2.6	65.3	111.23%
Virtual Machine	1	2	3.87	47.0	80.15%
Baremetal	1	4	3.34	104	100%
Container	1	4	4.5	58.5	56.25%
Virtual Machine	1	4	3.31	94.0	90.38%
Baremetal	1	8	3.71	188	100%
Container	1	8	9.33	57.0	30.32%
Virtual Machine	1	8	7.6	82.7	43.99%
Baremetal	1	16	9.23	236	100%
Container	1	16	23.27	70.2	29.75%
Virtual Machine	1	16	12.97	79.3	33.64%
Baremetal	1	32	10.5	261	100%
Container	1	32	32.68	67.3	25.81%
Virtual Machine	1	32	29.7	74.5	28.58%
Baremetal	1	64	14.9	275	100%
Container	1	64	75.03	70.5	25.64%
Virtual Machine	1	64	57.1	74.6	27.09%

Efficiency is calculated as below:

Efficiency: (Throughputs from virtualization types (VM/Container)/ Throughput (Bare Metal)) \*100



## Network Benchmark Graph:

Network Benchmark Results

