COEN Homework 1
System Vs OS Virtualization

Name: Satya Akshara Nittala
ID: 1630636

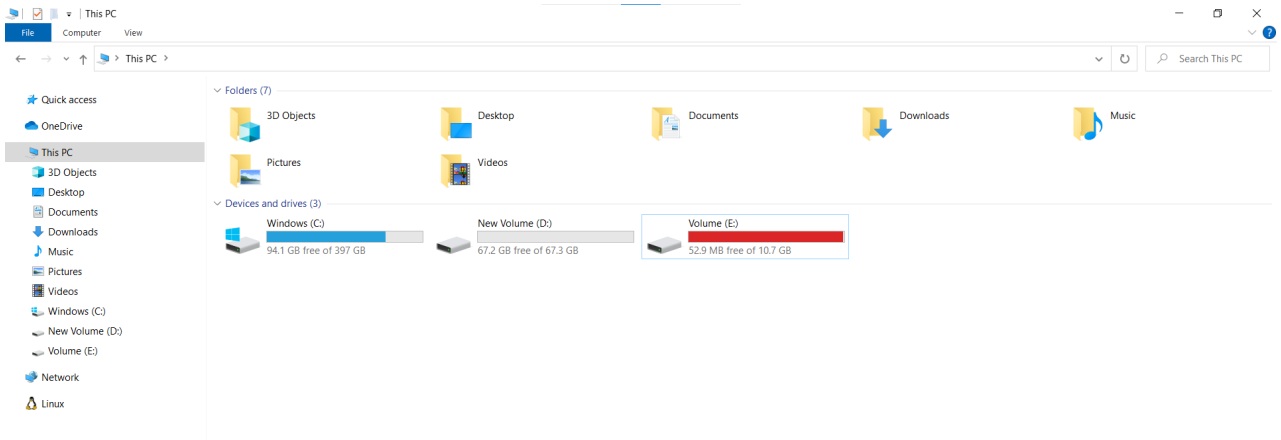
1. Environment Setup:

a. Installation of Ubuntu on Windows:

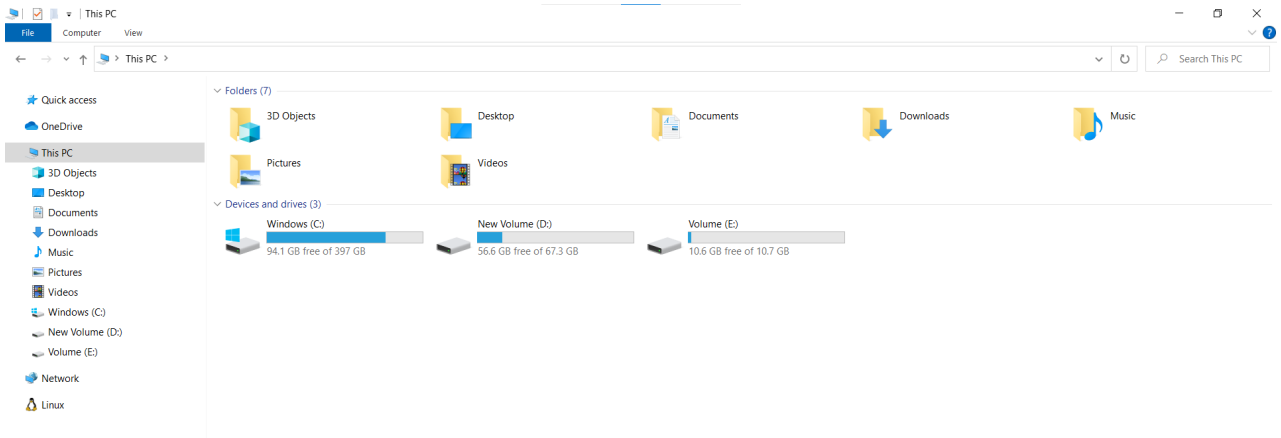
The Ubuntu image is downloaded from the following link:
https://releases.ubuntu.com/20.04.5/

b. Installation of QEMU on Windows:

1. Downloaded QEMU from https://qemu.weilnetz.de/w64/.
2. Created a partition on my computer called drive D:.

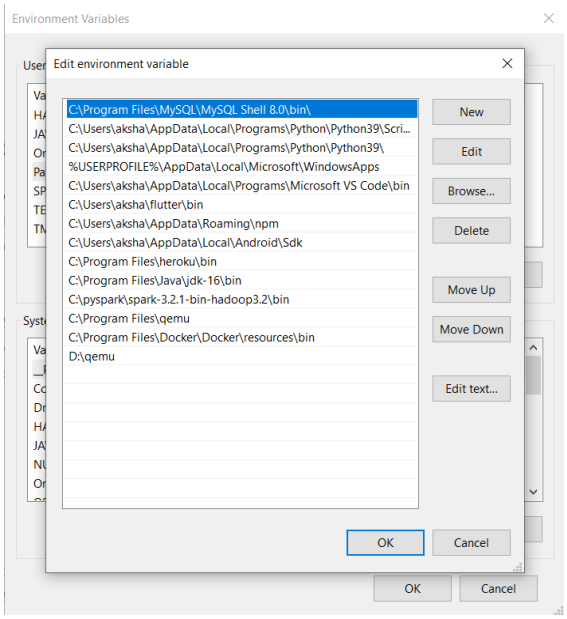


3. Installed QEMU into drive D:.



4. Added QEMU path into the environment variables.

5. Opened command prompt as administrator.



6. Ran the following commands:

```
D:
cd \qemu
qemu-img create -f qcow2 ubuntu20.img 30G
qemu-system-x86_64.exe -m 1G -smp 2 -boot order=dc -hda
ubuntu20.img -cdrom "e:\ubuntu-20.04.3-desktop-amd64.iso"
```

```
C:\Windows\System32>D:

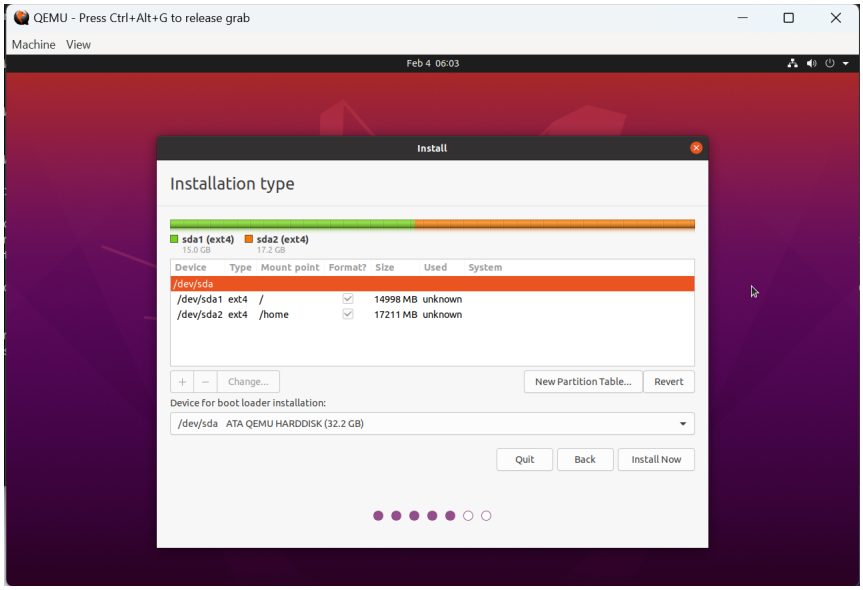
D:\>cd \qemu

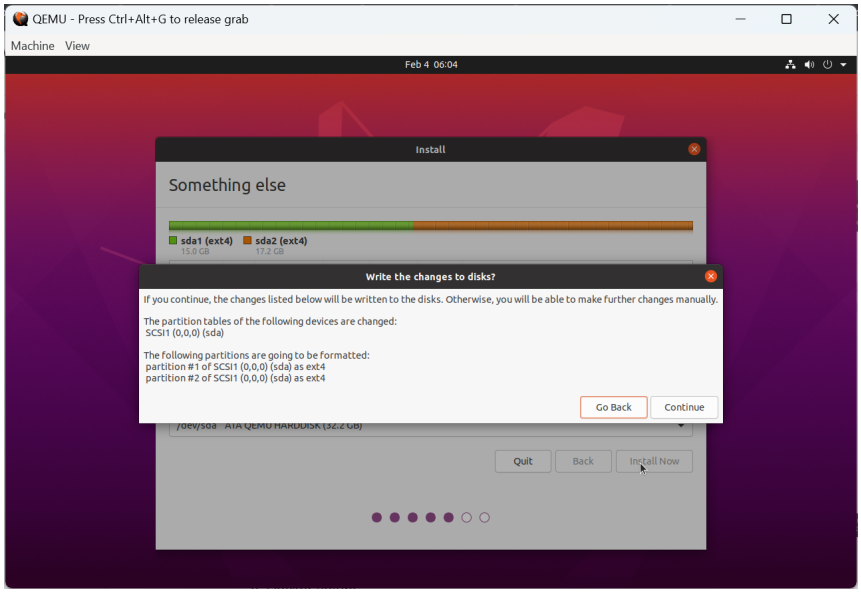
D:\qemu>qemu-img create -f qcow2 ubuntu20.img 30G
Formatting 'ubuntu20.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=32212254720 lazy_refc
ounts=off refcount_bits=16

D:\qemu>qemu-system-x86_64.exe -m 4G -smp 2 -boot order=dc -hda ubuntu20.img -cdrom "D:\ubuntu-20.04.5-desktop-amd64.iso
"

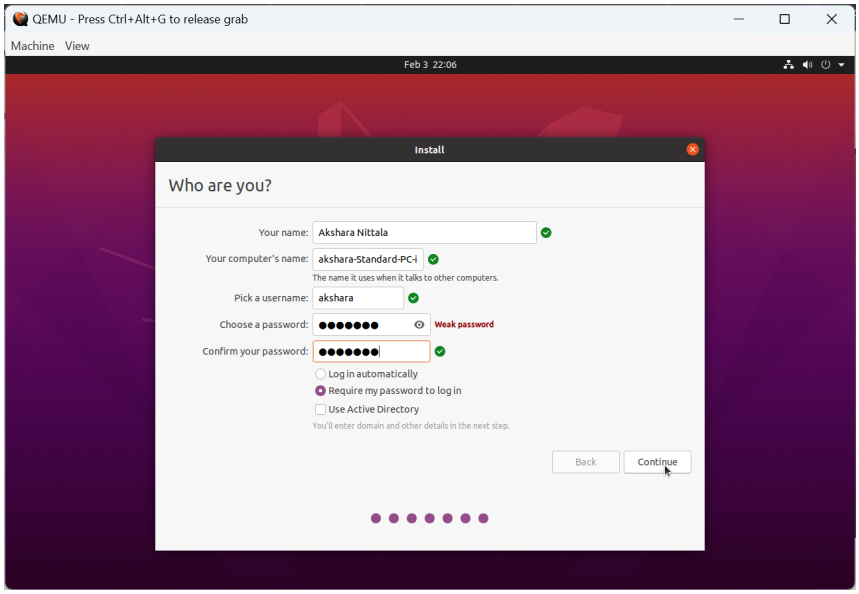
(qemu:12116): Gtk-WARNING **: 21:51:06.795: Could not load a pixbuf from icon theme.
This may indicate that pixbuf loaders or the mime database could not be found.
```

7. After this the Ubuntu VM opens in QEMU for installation.
   Below, 2 partitions were created for the installation of Ubuntu.

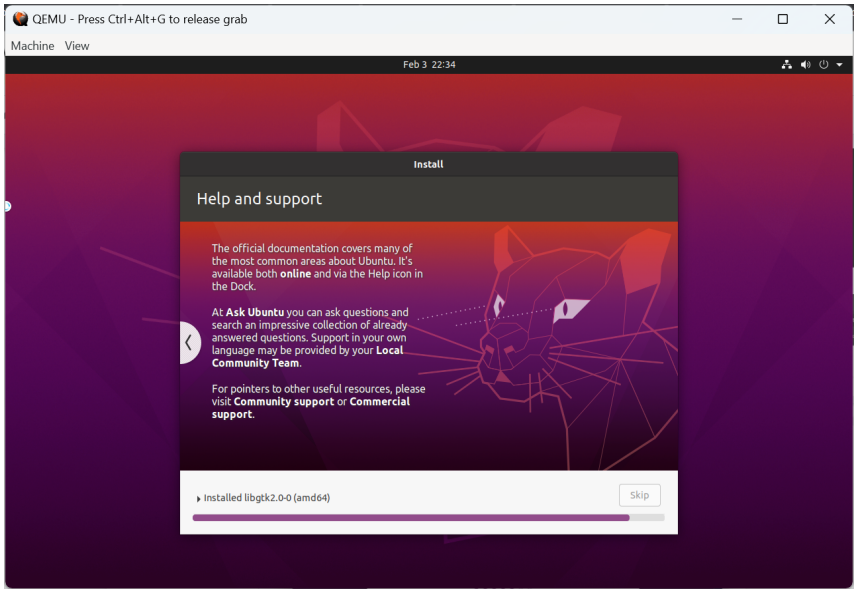




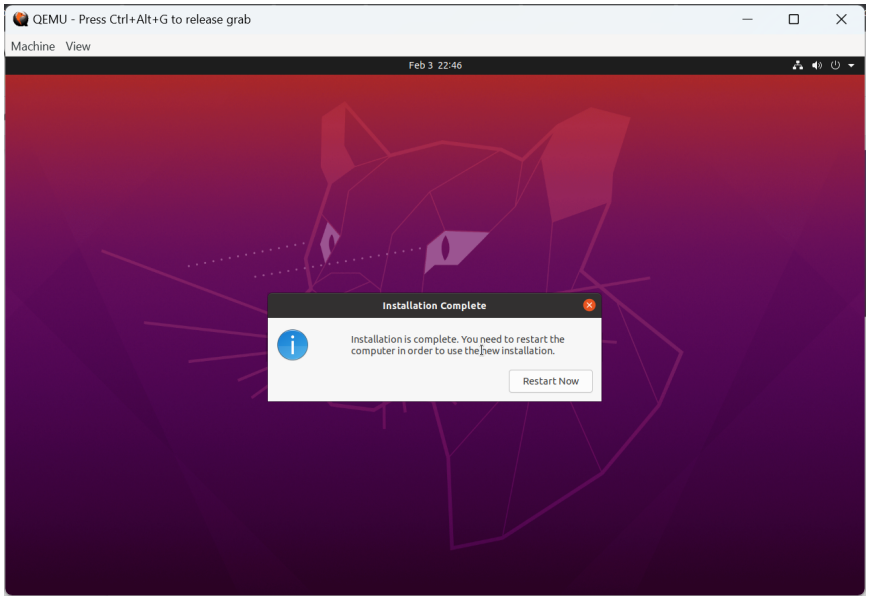
Below, the dialog for creation of a default user is opened.



After clicking continue, the installation begins.



The installation is complete. We can close the dialog and go back to the command prompt.



8. We store the downloaded Ubuntu image in the D: drive.
9. To run the Ubuntu terminal we run the following command:
   `qemu-system-x86_64.exe -m 4G -smp 2 -boot order=dc -hda ubuntu20.img -cdrom "D:\ubuntu-20.04.5-desktop-amd64.iso"`

```
C:\Windows\System32>D:

D:\>cd \qemu

D:\qemu>qemu-img create -f qcow2 ubuntu20.img 30G
Formatting 'ubuntu20.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=32212254720 lazy_refc
ounts=off refcount_bits=16

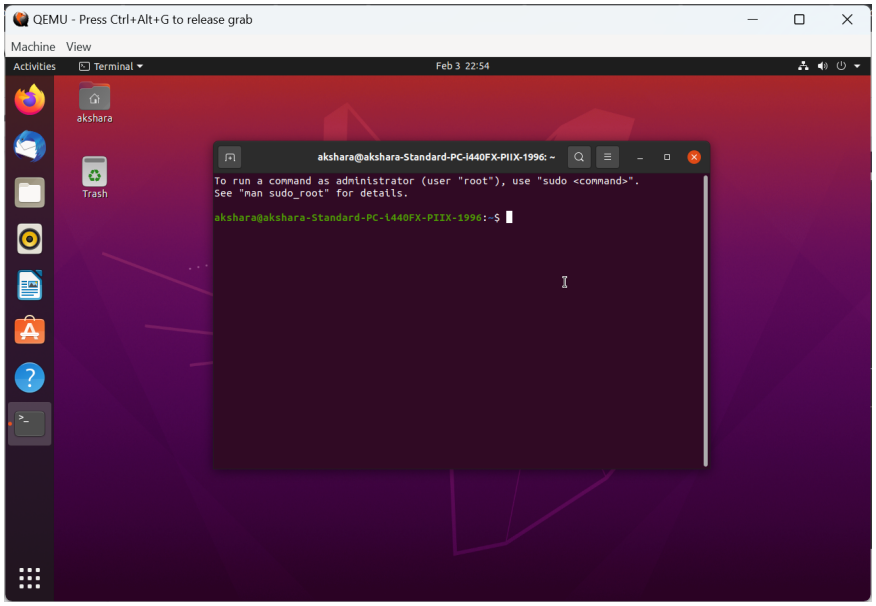
D:\qemu>qemu-system-x86_64.exe -m 4G -smp 2 -boot order=dc -hda ubuntu20.img -cdrom "D:\ubuntu-20.04.5-desktop-amd64.iso
"

(qemu:12116): Gtk-WARNING **: 21:51:06.795: Could not load a pixbuf from icon theme.
This may indicate that pixbuf loaders or the mime database could not be found.

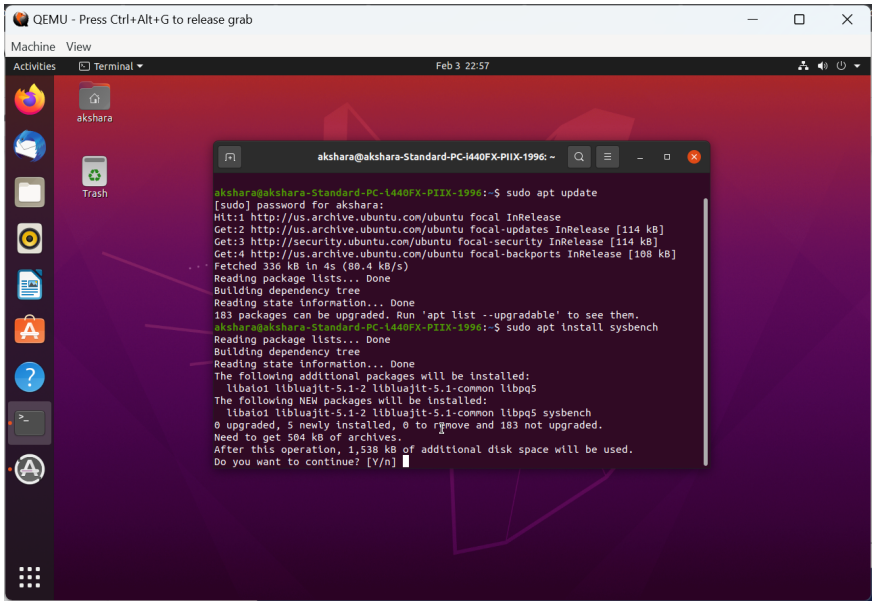
D:\qemu>qemu-system-x86_64.exe -m 4G -smp 2 -boot order=dc -hda ubuntu20.img

(qemu:10864): Gtk-WARNING **: 22:48:45.074: Could not load a pixbuf from icon theme.
This may indicate that pixbuf loaders or the mime database could not be found.
```

10. The Ubuntu VM starts. After logging in, I opened a terminal.



11. The following commands were run to install sysbench:
    ```
    sudo apt update
    sudo apt install sysbench
    ```

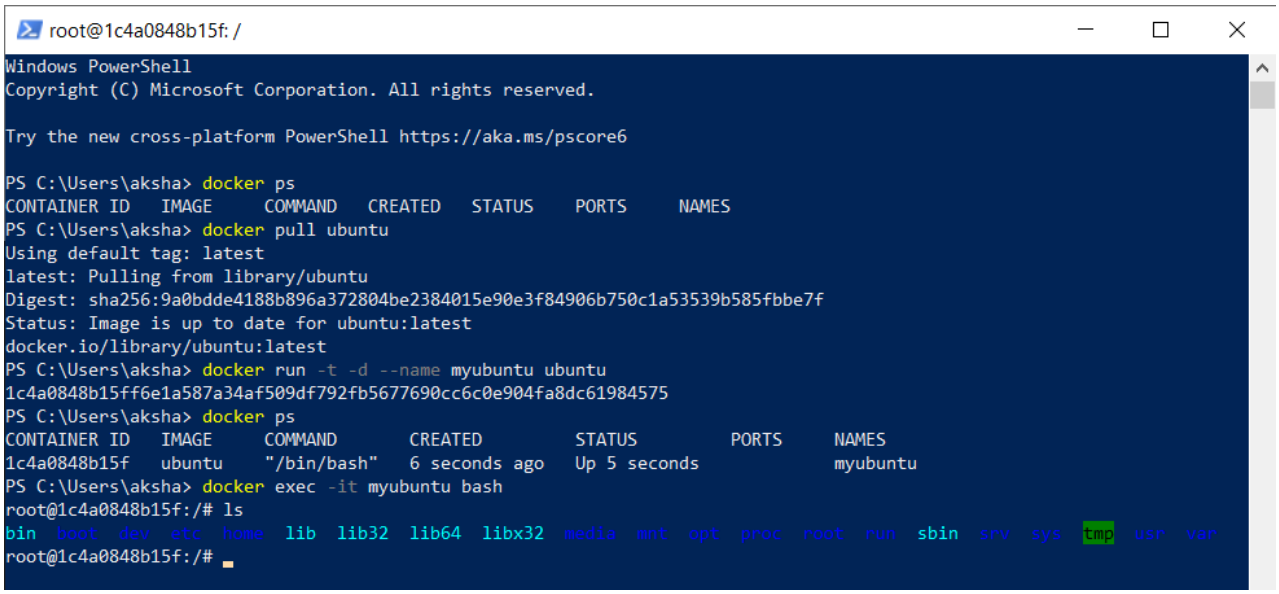


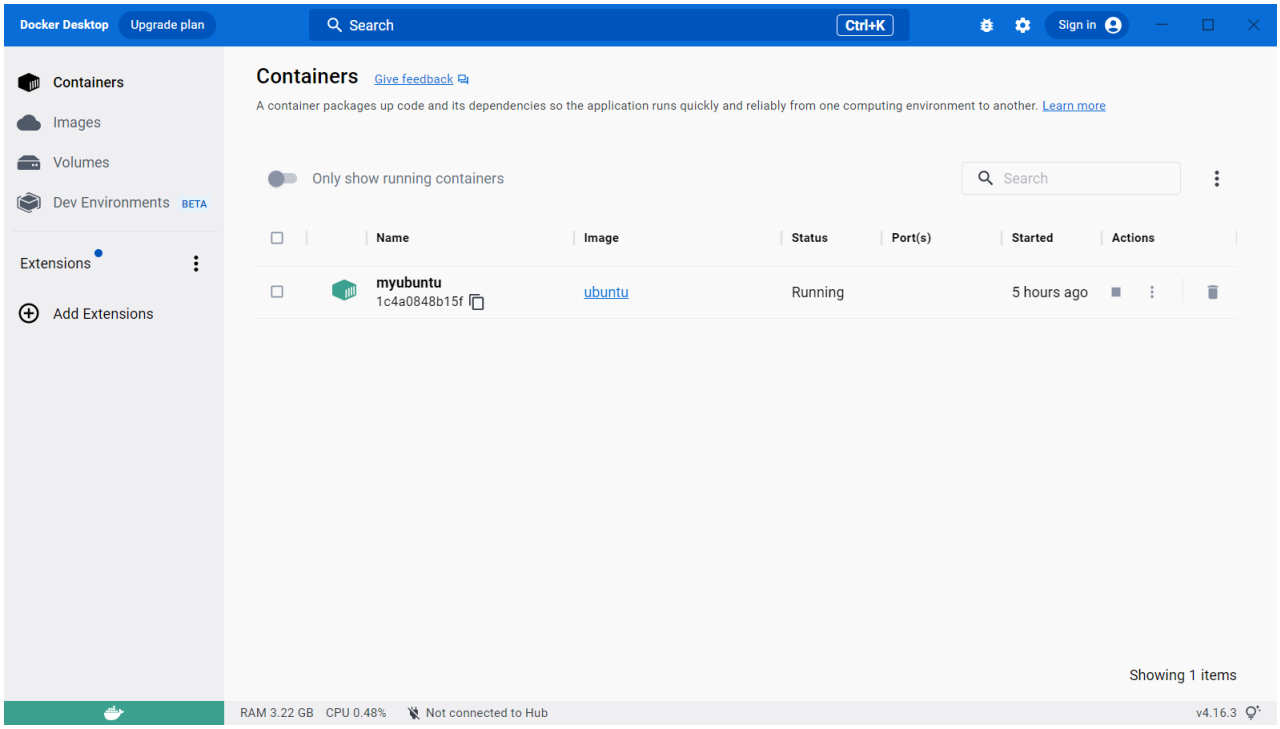
12. With this, the installation of Ubuntu in QEMU has been completed.

c. <u>Installation of Docker on Windows:</u>

1. Docker desktop is installed from the following link: https://docs.docker.com/desktop/install/windows-install/
2. Then we go to Windows Powershell type the following commands:
    ```
    docker ps
    docker pull ubuntu
    ```

`docker run -t -d –name ubuntu ubuntu` (creates a container)
`docker exec -it ubuntu bash` (starts ubuntu bash on that container)





3. The following commands are run to install sysbench:
   `apt update`
   `apt install sysbench`
4. With this, the installation of Ubuntu in Docker has been completed.

1) CPU Testing

For CPU testing, I will be running 3 test cases to evaluate CPU performance between Docker and QEMU.

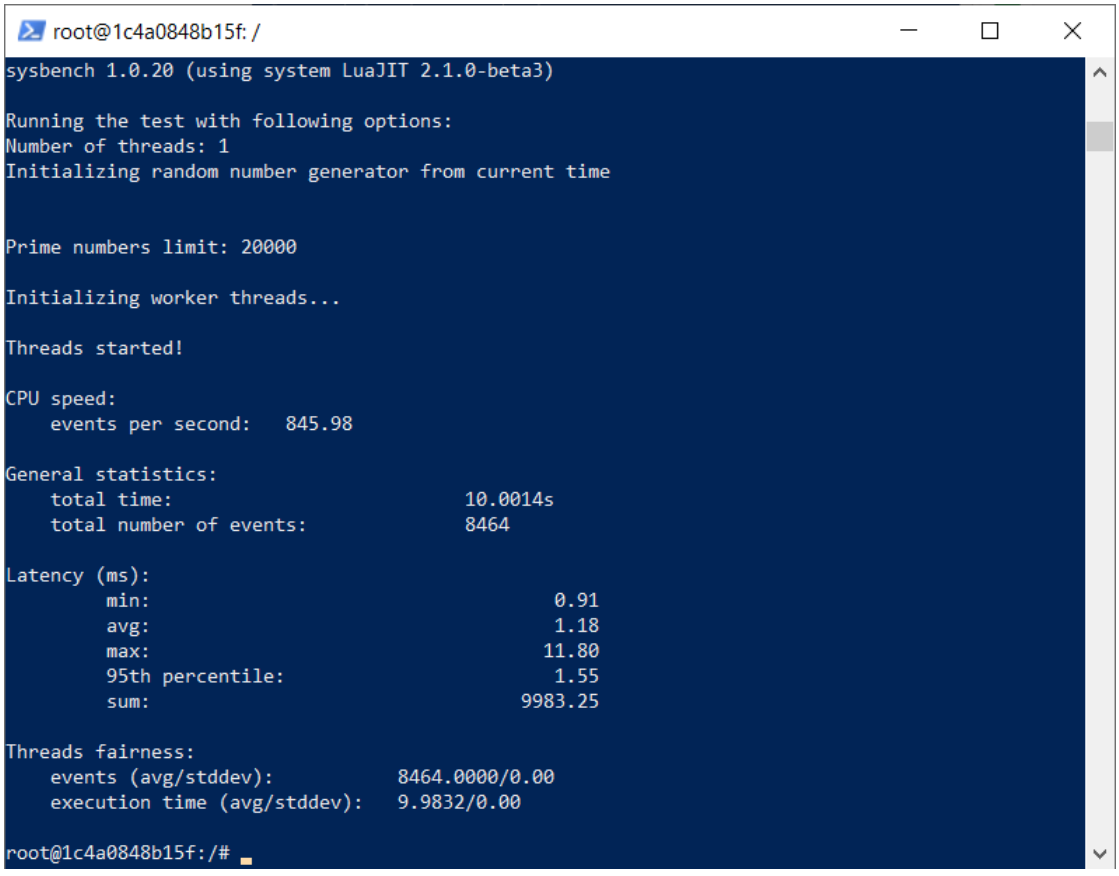
The sysbench command used for the related test cases is:

```
sysbench cpu --cpu-max-prime={some_value}
--num-threads={some_value} --time= {some_value} run
```

a) Docker Ubuntu VM:

1. `sysbench cpu -threads=1 -cpu-max-prime=20000 -time=10 run`

Output:

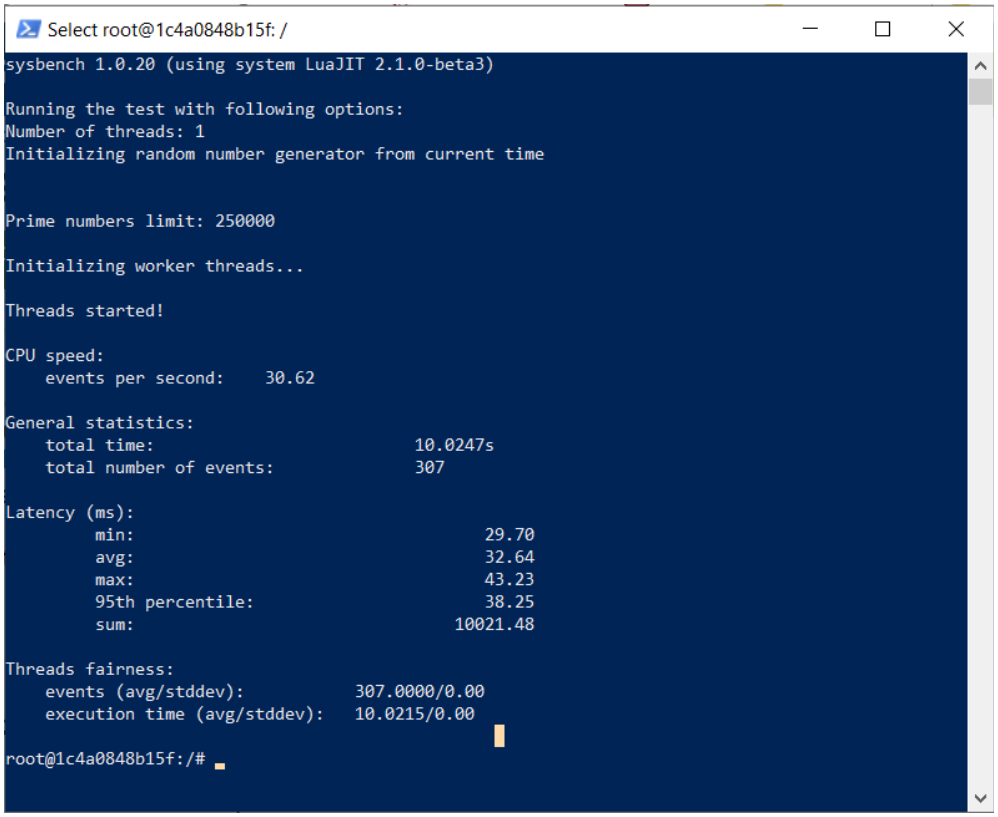


After 5 iterations:

Iteration Number	Total Time	CPU Speed	Average Latency
1	10.0008	816.10	1.22
2	10.0006	841.06	1.19
3	10.0058	848.65	1.18
4	10.0003	803.62	1.24
5	10.0014	845.98	1.18
Minimum	10.0003	803.62	1.18
Maximum	10.0058	848.65	1.24
Average	10.00178	831.082	1.202
Standard Deviation	0.002042	17.9375	0.024

2. `sysbench cpu --threads=1 --cpu-max-prime=250000 --time=10 run`

Output:

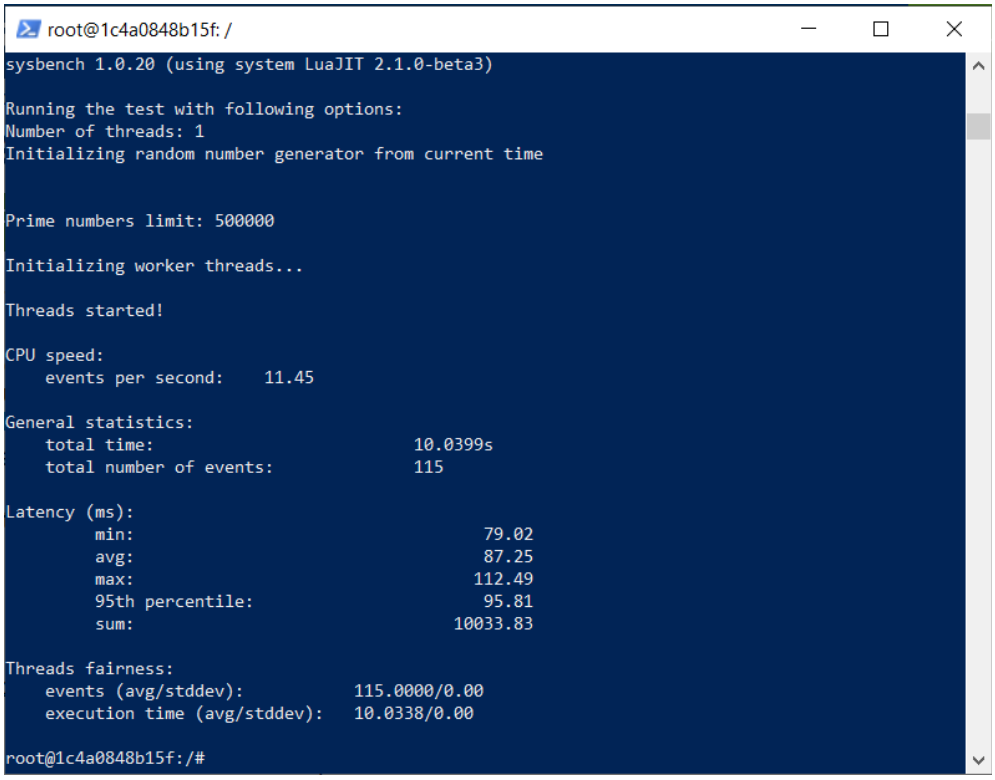


After 5 iterations:

Iteration Number	Total Time	CPU Speed	Average Latency
1	10.0247	30.62	32.64
2	10.0317	30.39	32.88
3	10.0053	30.33	32.90
4	10.0251	30.39	32.86
5	10.0193	30.44	32.84
Minimum	10.0053	30.33	32.64
Maximum	10.0251	30.62	32.90
Average	10.02122	30.434	32.824
Standard Deviation	0.008878	0.099318	0.094149

3. `sysbench cpu –threads=1 –cpu-max-prime=500000 –time=10 run`

Output:



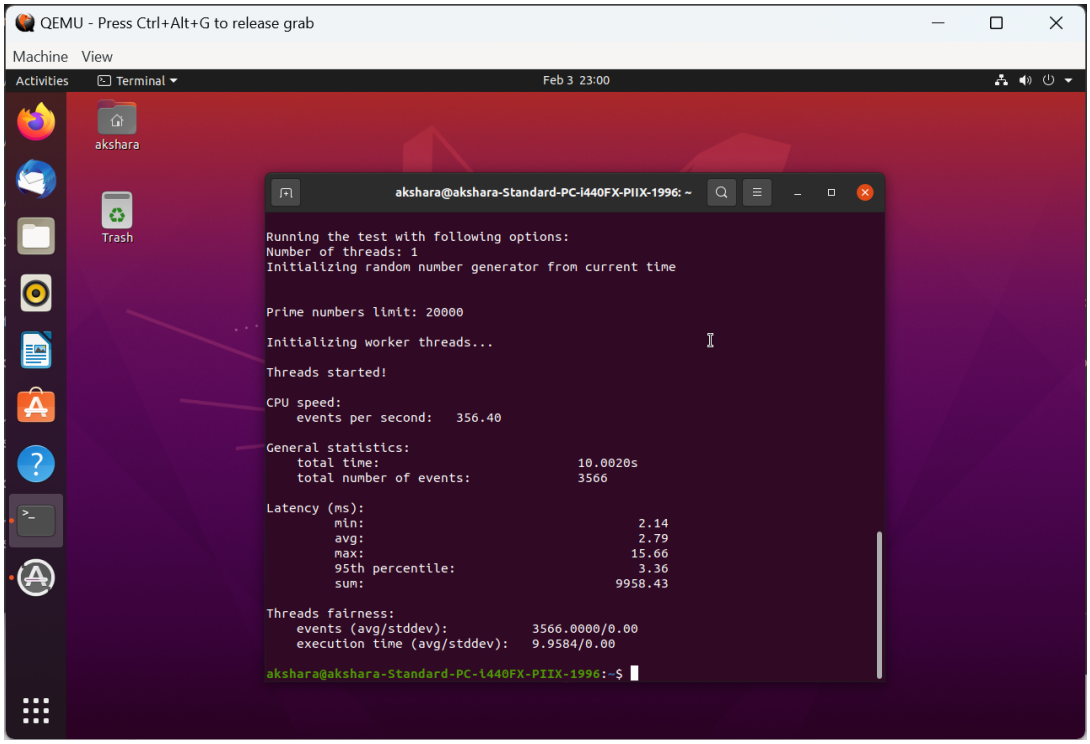
After 5 iterations:

Iteration Number	Total Time	CPU Speed	Average Latency
1	10.0399	11.45	87.25
2	10.0139	11.38	87.82
3	10.0788	11.50	86.87
4	10.0773	11.61	86.10
5	10.0423	11.45	87.23
Minimum	10.0139	11.38	86.10
Maximum	10.0788	11.61	87.82
Average	10.05044	11.478	87.054
Standard Deviation	0.02465	0.076263	0.565706

b) QEMU Ubuntu VM:

1. `sysbench cpu –threads=1 –cpu-max-prime=20000 –time=10 run`

Output:

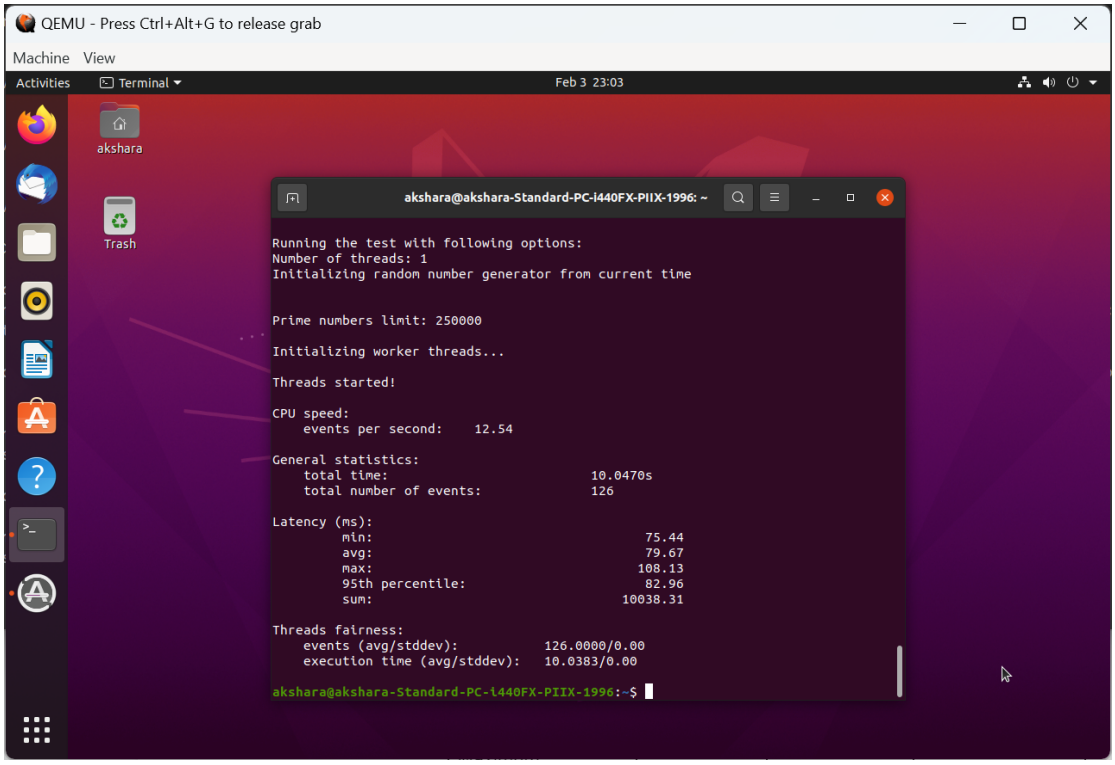


After 5 iterations:

Test Number	Total Time	CPU Speed	Average Latency
1	10.0020	350.40	2.79
2	10.0029	358.11	2.78
3	10.0019	364.04	2.74
4	10.0016	365.54	2.73
5	10.0025	364.23	2.75
Minimum	10.0016	350.40	2.73
Maximum	10.0029	365.54	2.79
Average	10.00218	360.464	2.758
Standard Deviation	0.0004621	5.649101	0.023152

2. `sysbench cpu -threads=1 -cpu-max-prime=250000 -time=10 run`

Output:

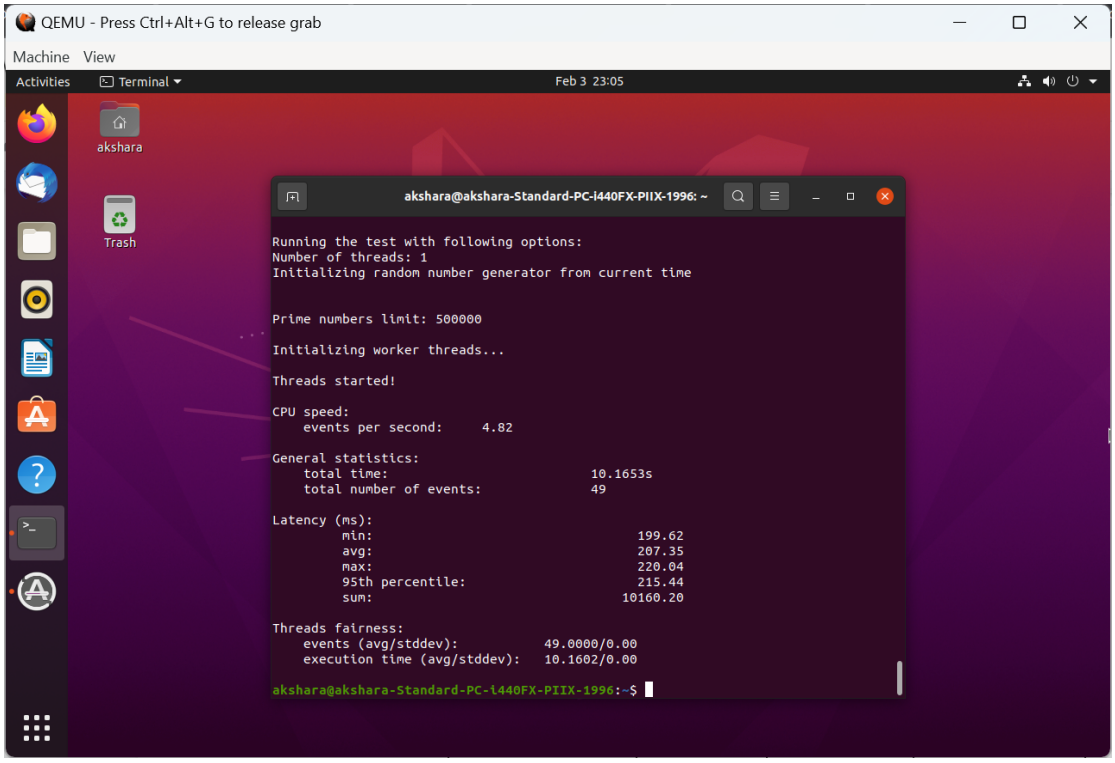


After 5 iterations:

Test Number	Total Time	CPU Speed	Average Latency
1	10.0470	12.34	79.07
2	10.0365	12.55	79.58
3	10.0495	12.04	82.98
4	10.0390	12.46	81.62
5	10.0452	12.16	80.96
Minimum	10.0365	12.04	79.07
Maximum	10.0495	12.55	82.98
Average	10.04344	12.31	80.842
Standard Deviation	0.0049066	0.1878297	1.4087498

3. `sysbench cpu –threads=1 –cpu-max-prime=500000 –time=10 run`

Output:



After 5 iterations:

Test Number	Total Time	CPU Speed	Average Latency
1	10.1653	4.82	267.35
2	10.0600	4.77	209.49
3	10.0950	4.69	219.65
4	10.1250	4.32	225.36
5	10.0873	4.85	250.49
Minimum	10.0600	4.32	209.49
Maximum	10.1653	4.85	467.35
Average	10.10652	4.69	234.468
Standard Deviation	0.0359522	0.192769	21.280787

Conclusion:

After comparing the experiment performed on Docker Ubuntu VM and QEMU Ubuntu VM, we initially see that in both the VMs the cpu speed decreases as the cpu max prime increases. The average latency increases as the cpu max prime increases.

	Docker Ubuntu VM (CPU Speed)	QEMU Ubuntu VM (CPU Speed)
Cpu max prime = 20000	831.082	360.464
Cpu max prime = 250000	30.434	12.31
Cpu max prime = 500000	11.478	4.69

When we compare the cpu speed for the same conditions, we see that QEMU Ubuntu Vm has performed much better than Docker Ubuntu VM.

2) File I/O Testing

For File I/O Testing, I will be testing 2 modes sysbench supports, that is:

- Sequential Rewrite (seqrewr)
- Combined Random Read/Write (rmdrw)

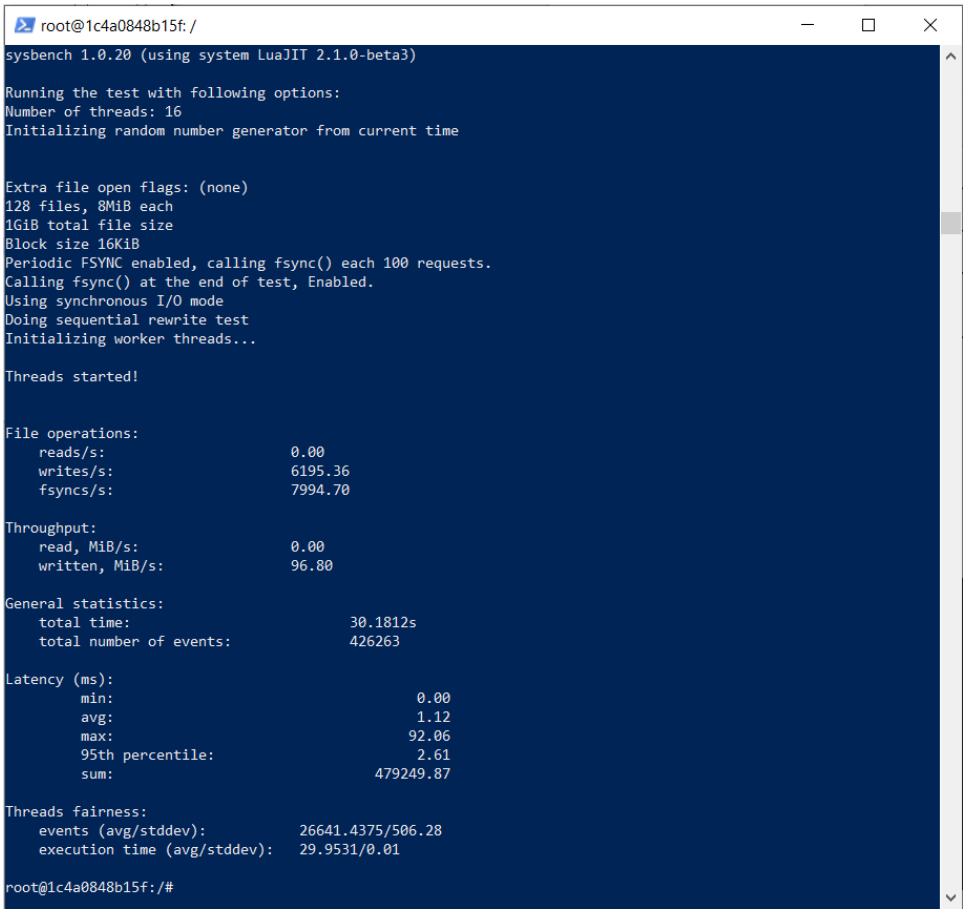
3 Test cases will be run to evaluate the I/O performance between Docker and QEMU.

a) Docker Ubuntu VM:

I. Sequential Rewrite (seqrewr):

1.
```
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=1G –file-test-mode=seqrewr prepare
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=1G –file-test-mode=seqrewr run
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=1G –file-test-mode=seqrewr cleanup
```

Output:

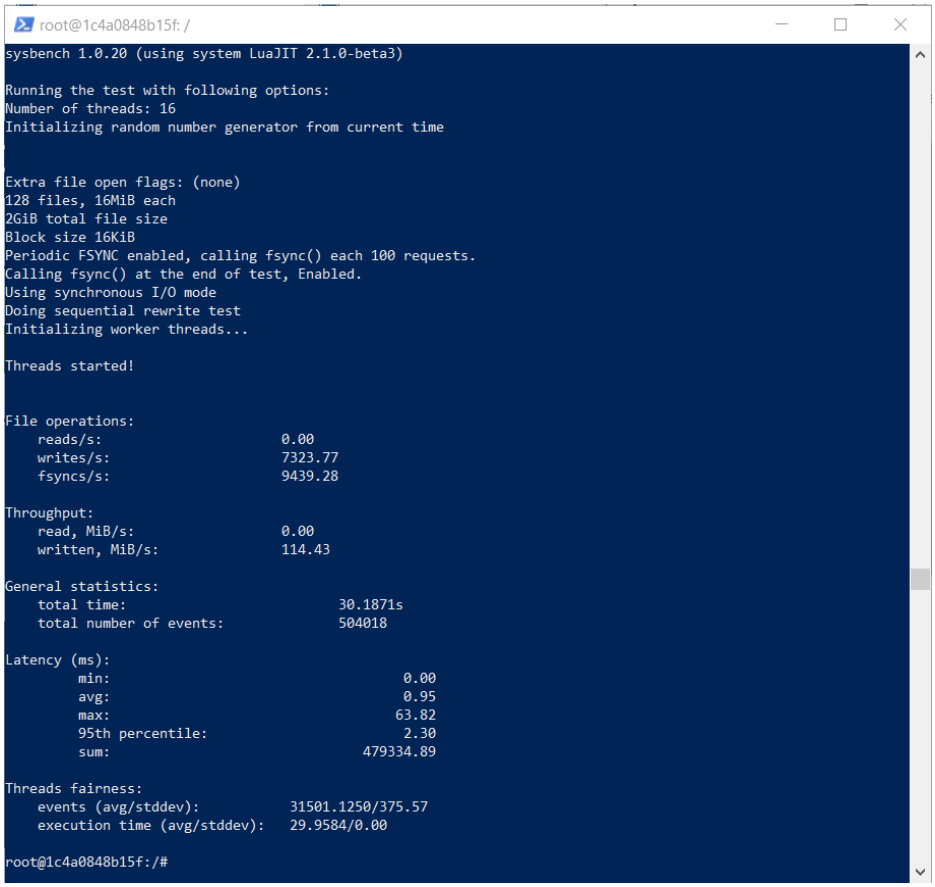


After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	96.80	30.1812
2	0	112.14	30.2342
3	0	120.42	30.1764
4	0	121.39	30.1684
5	0	117.78	30.1907
Minimum	0	96.80	30.1684
Maximum	0	121.39	30.2342
Average	0	113.65	30.19018
Standard Deviation	0	9.00319	0.023164

2. `sysbench --num-threads=16 --test=fileio --time=30 --file-total-size=2G --file-test-mode=seqrewr prepare`
   `sysbench --num-threads=16 --test=fileio --time=30 --file-total-size=2G --file-test-mode=seqrewr run`
   `sysbench --num-threads=16 --test=fileio --time=30 --file-total-size=2G --file-test-mode=seqrewr cleanup`

Output:



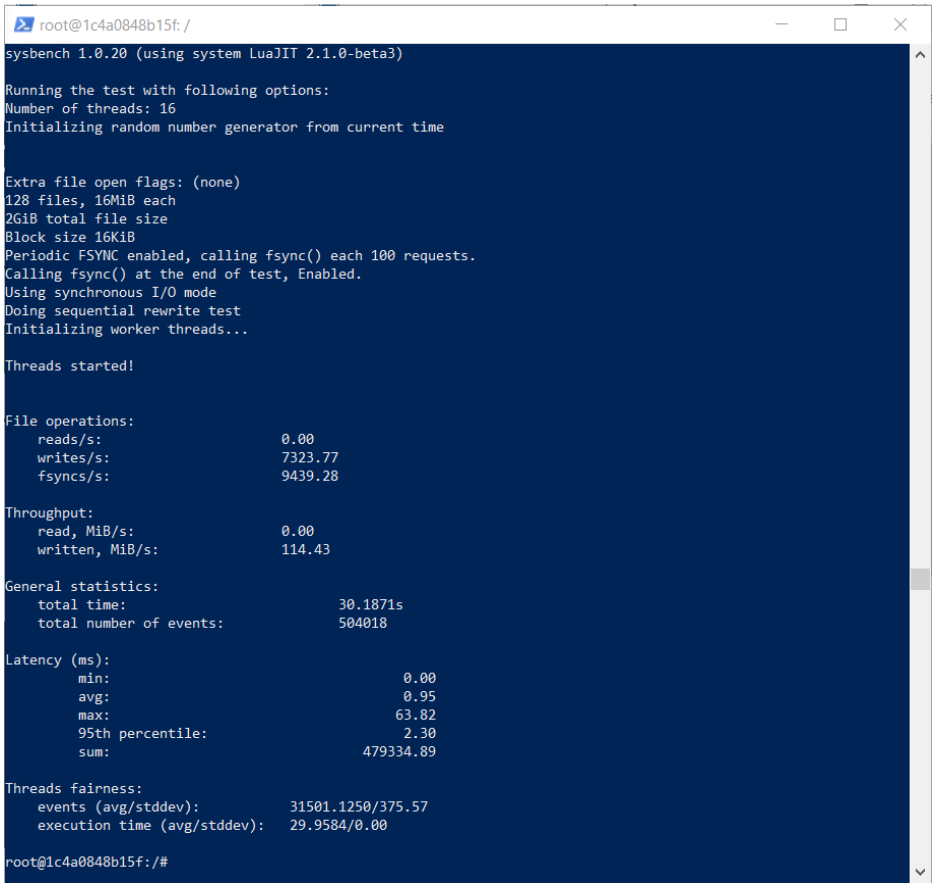
After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	113.78	30.1812
2	0	103.60	30.1902
3	0	114.43	30.1871
4	0	117.18	30.1999

5	0	121.07	30.1847
Minimum	0	103.60	30.1812
Maximum	0	121.07	30.1999
Average	0	114.012	30.18862
Standard Deviation	0	5.80499	0.0063634

3. `sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=4G –file-test-mode=seqrewr prepare`
   `sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=4G –file-test-mode=seqrewr run`
   `sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=4G –file-test-mode=seqrewr cleanup`

   Output:



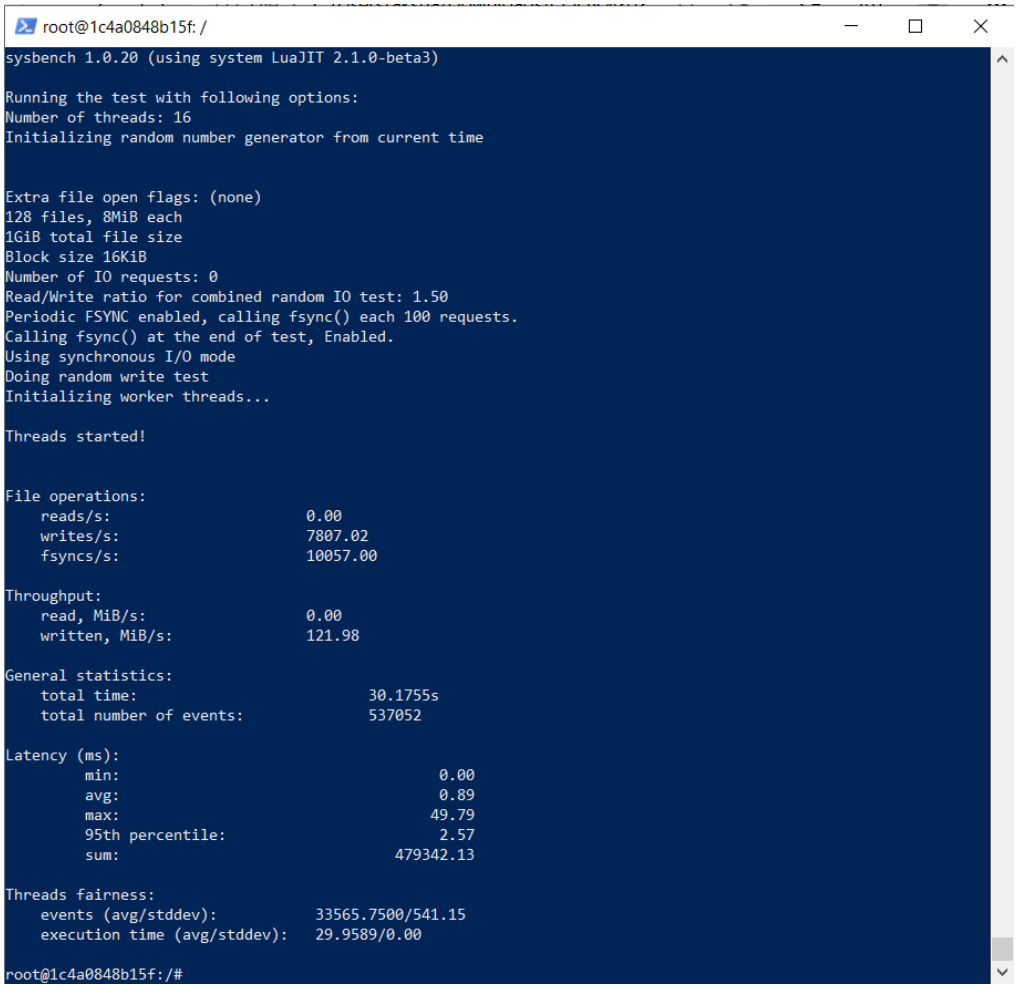
   After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	113.54	30.1643
2	0	118.71	30.1760
3	0	114.43	30.1871
4	0	121.93	30.1625
5	0	119.89	30.1949
Minimum	0	113.54	30.1625
Maximum	0	121.93	30.1949
Average	0	117.7	30.17696
Standard Deviation	0	3.215823	0.0126091

II. Combined Random Read/Write (rmdrw)

1.
```
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=1G –file-test-mode=rndwr prepare
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=1G –file-test-mode=rndwr run
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=1G –file-test-mode=rndwr cleanup
```

   Output:



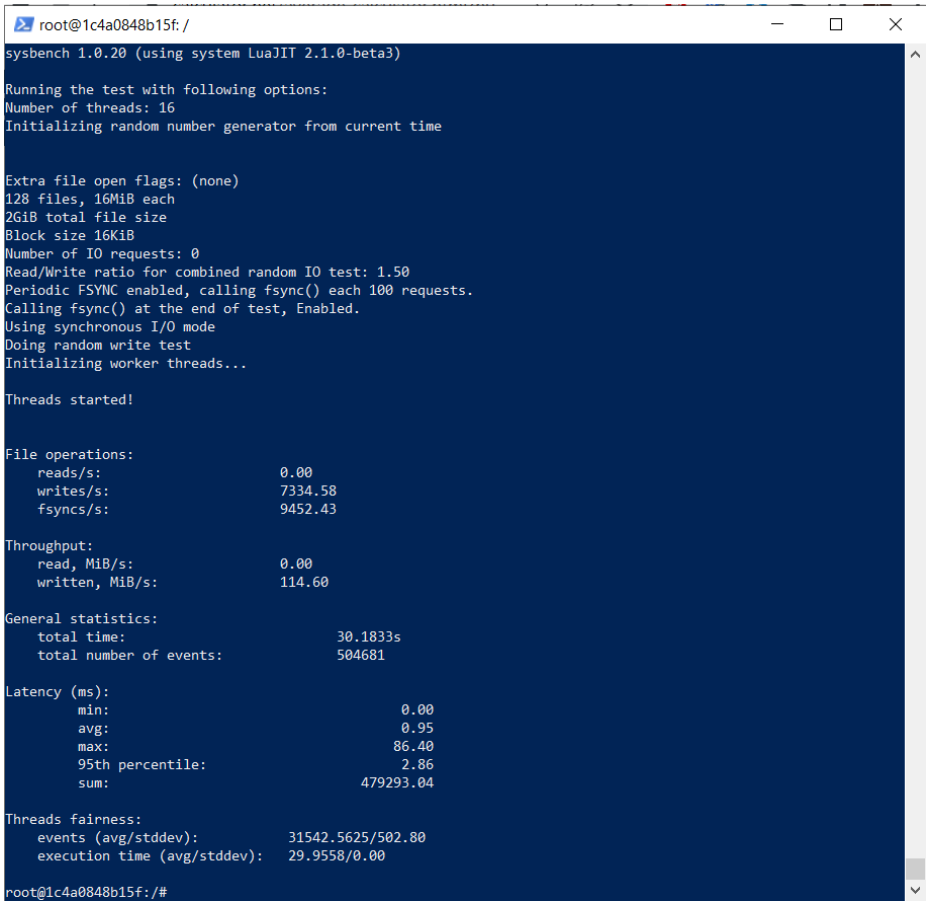
After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	121.87	30.2158
2	0	121.98	30.1755
3	0	120.43	30.1751
4	0	114.98	30.1797
5	0	119.89	30.1816
Minimum	0	114.98	30.1751
Maximum	0	121.98	30.2158
Average	0	119.83	30.18554

Standard Deviation	0	2.556177	0.0153303

2.
```
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=2G –file-test-mode=rndwr prepare
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=2G –file-test-mode=rndwr run
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=2G –file-test-mode=rndwr cleanup
```

Output:



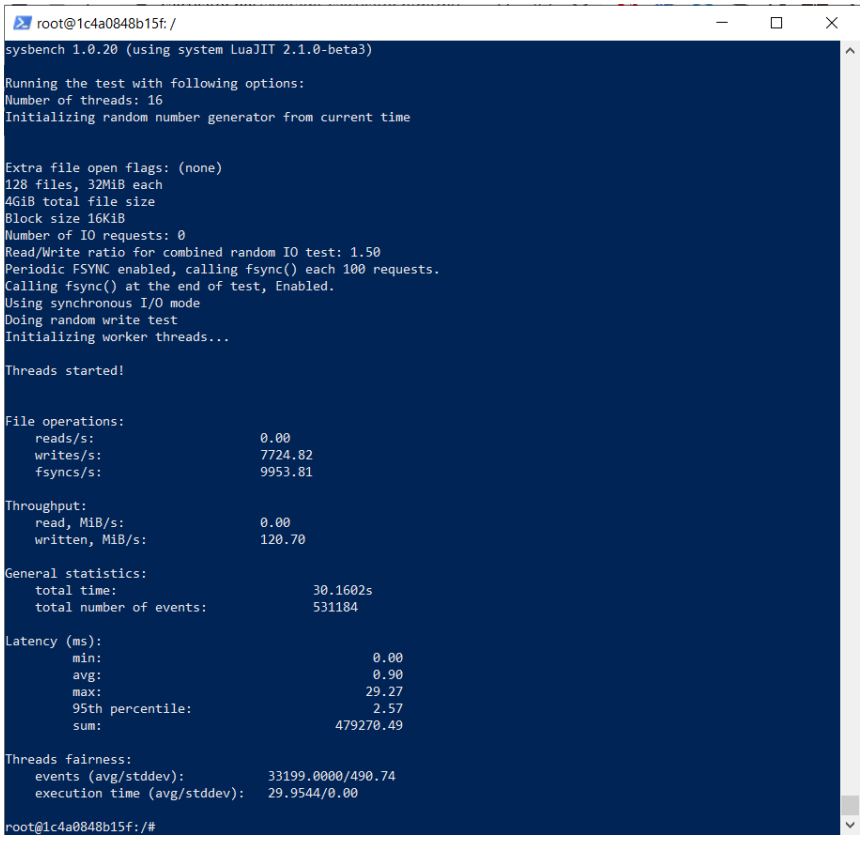
After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	114.60	30.1833
2	0	126.44	30.1738

3	0	110.87	30.1995
4	0	117.91	30.1725
5	0	112.45	30.2065
Minimum	0	110.87	30.1725
Maximum	0	126.44	30.2065
Average	0	116.454	30.18712
Standard Deviation	0	5.52251	0.013672

3. `sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=4G –file-test-mode=rndwr prepare`
   `sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=4G –file-test-mode=rndwr run`
   `sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=4G –file-test-mode=rndwr cleanup`

Output:



After 5 iterations:

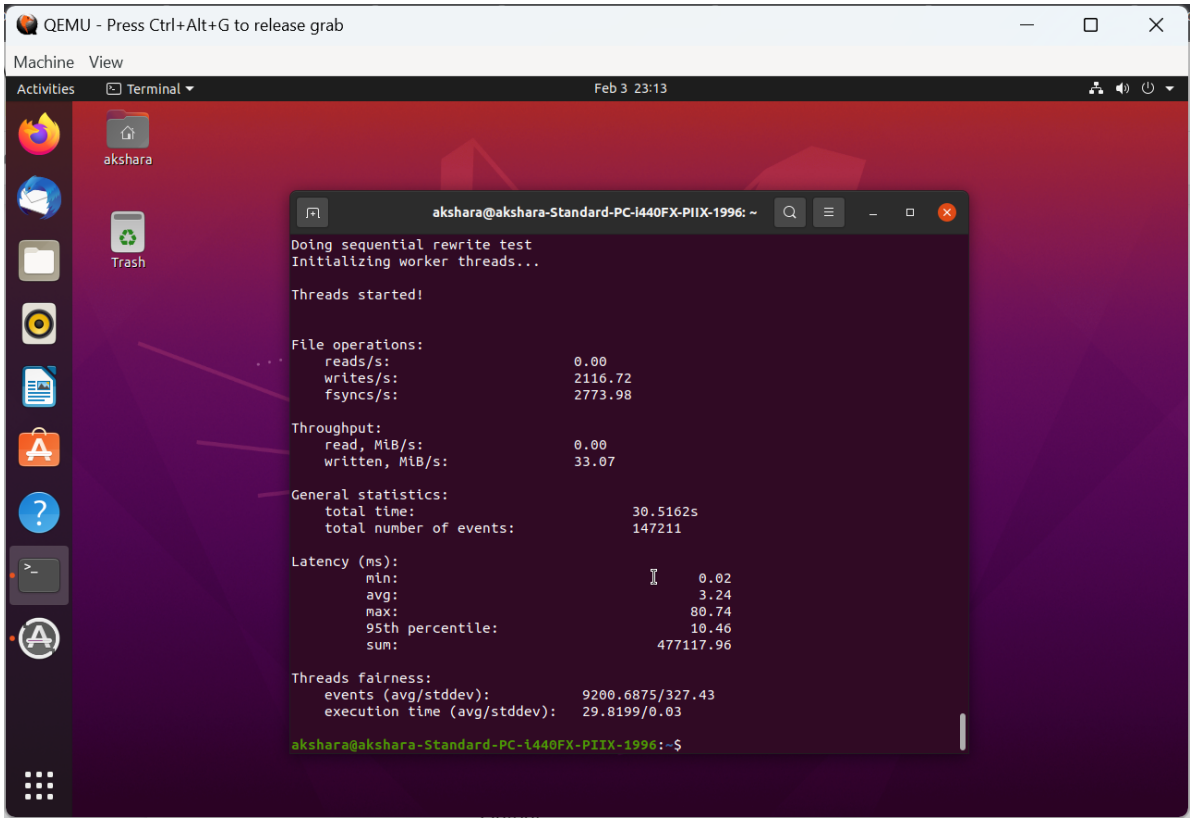
Test Number	Read Throughput	Write Throughput	Total Time
1	0	120.70	30.1602
2	0	120.96	30.1601
3	0	119.56	30.1849
4	0	116.77	30.1861
5	0	119.58	30.1816
Minimum	0	116.77	30.1601
Maximum	0	120.70	30.1861
Average	0	119.51	30.17458
Standard Deviation	0	1.485342	0.011874

b) QEMU Ubuntu VM:

I. Sequential Rewrite (seqrewr):

1.
```
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=1G –file-test-mode=seqrewr prepare
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=1G –file-test-mode=seqrewr run
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=1G –file-test-mode=seqrewr cleanup
```

Output:

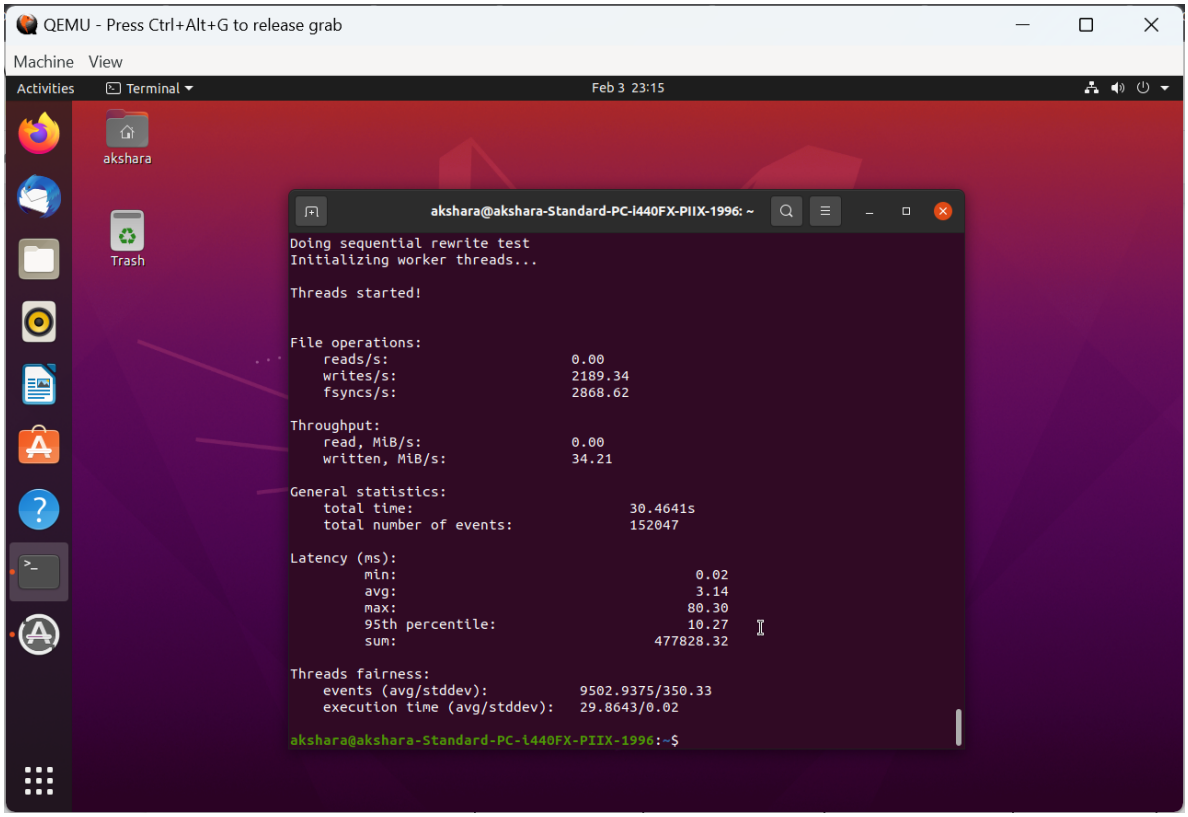


After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	33.07	30.5162
2	0	34.67	30.6179
3	0	32.54	30.4254
4	0	33.54	30.3957
5	0	32.97	30.3659
Minimum	0	32.54	30.3659
Maximum	0	34.67	30.6179
Average	0	33.358	30.46422
Standard Deviation	0	0.7289828	0.0918567

2. sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=2G –file-test-mode=seqrewr prepare
   sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=2G –file-test-mode=seqrewr run
   sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=2G –file-test-mode=seqrewr cleanup

Output:



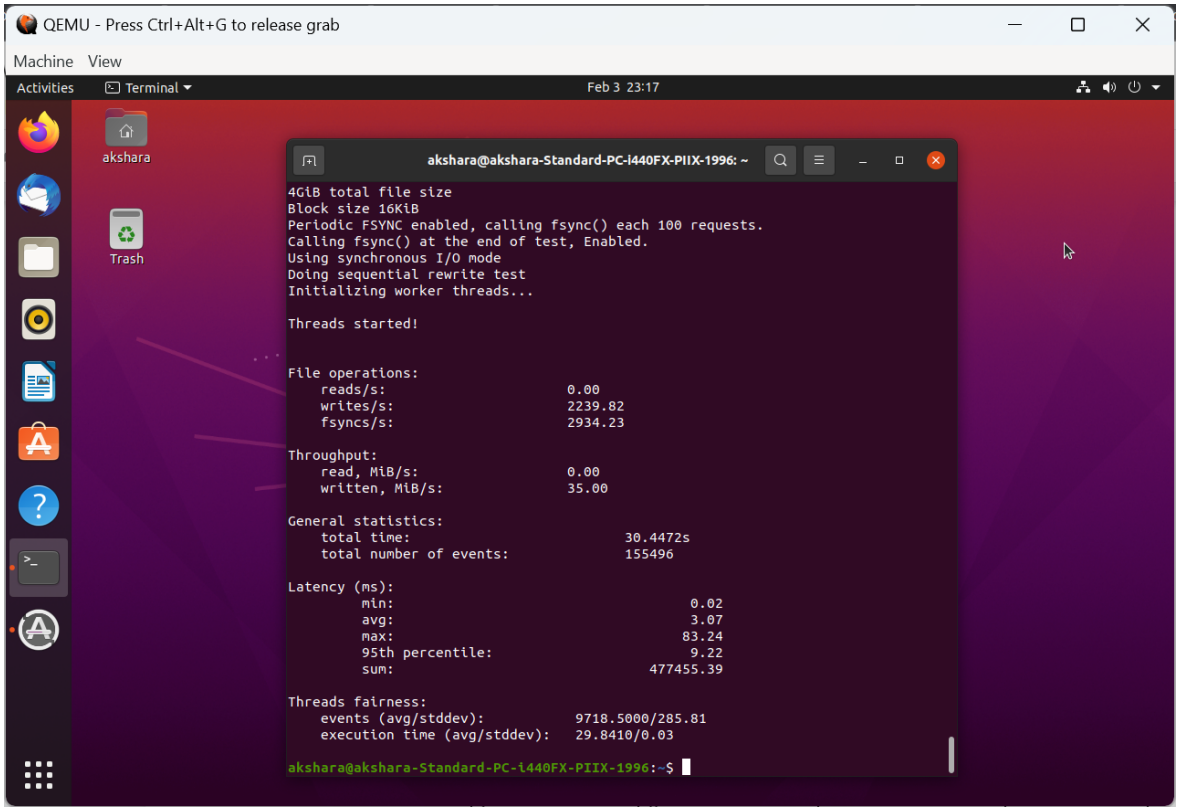
After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	34.21	30.4641
2	0	33.52	30.4562
3	0	32.94	30.2781
4	0	33.68	30.3842
5	0	34.17	30.2759
Minimum	0	32.94	30.2759

Maximum	0	34.21	30.4641
Average	0	33.704	30.3717
Standard Deviation	0	0.467187	0.0821862

3.
```
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=4G –file-test-mode=seqrewr prepare
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=4G –file-test-mode=seqrewr run
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=4G –file-test-mode=seqrewr cleanup
```

Output:



After 5 iterations:

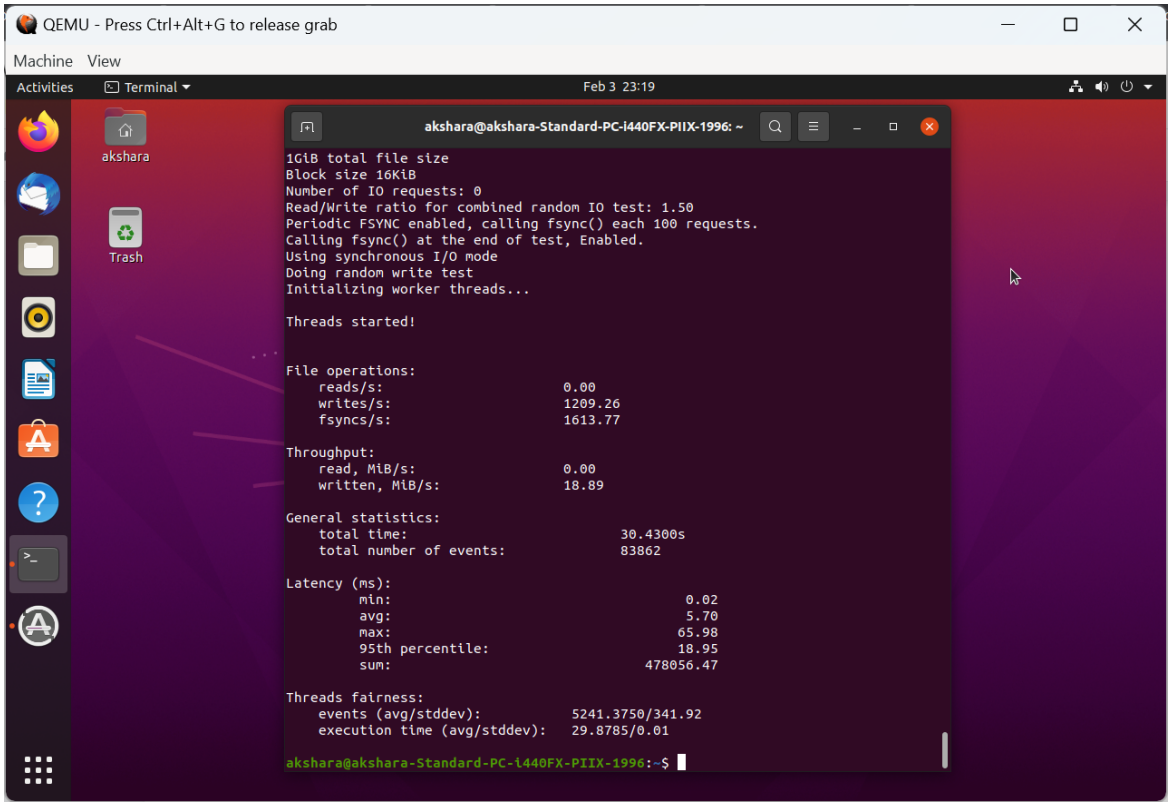
Test Number	Read Throughput	Write Throughput	Total Time
1	0	35	30.4472

2	0	34.23	30.3389
3	0	34.16	30.4125
4	0	35.12	30.4225
5	0	33.98	30.4328
Minimum	0	33.98	30.3389
Maximum	0	35.12	30.4472
Average	0	34.498	30.41078
Standard Deviation	0	0.467606	0.0377315

III. Combined Random Read/Write (rmdrw):

1. ```
   sysbench –num-threads=16 –test=fileio –time=30
   –file-total-size=1G –file-test-mode=rndwr prepare
   sysbench –num-threads=16 –test=fileio –time=30
   –file-total-size=1G –file-test-mode=rndwr run
   sysbench –num-threads=16 –test=fileio –time=30
   –file-total-size=1G –file-test-mode=rndwr cleanup
   ```

   Output:
   ```



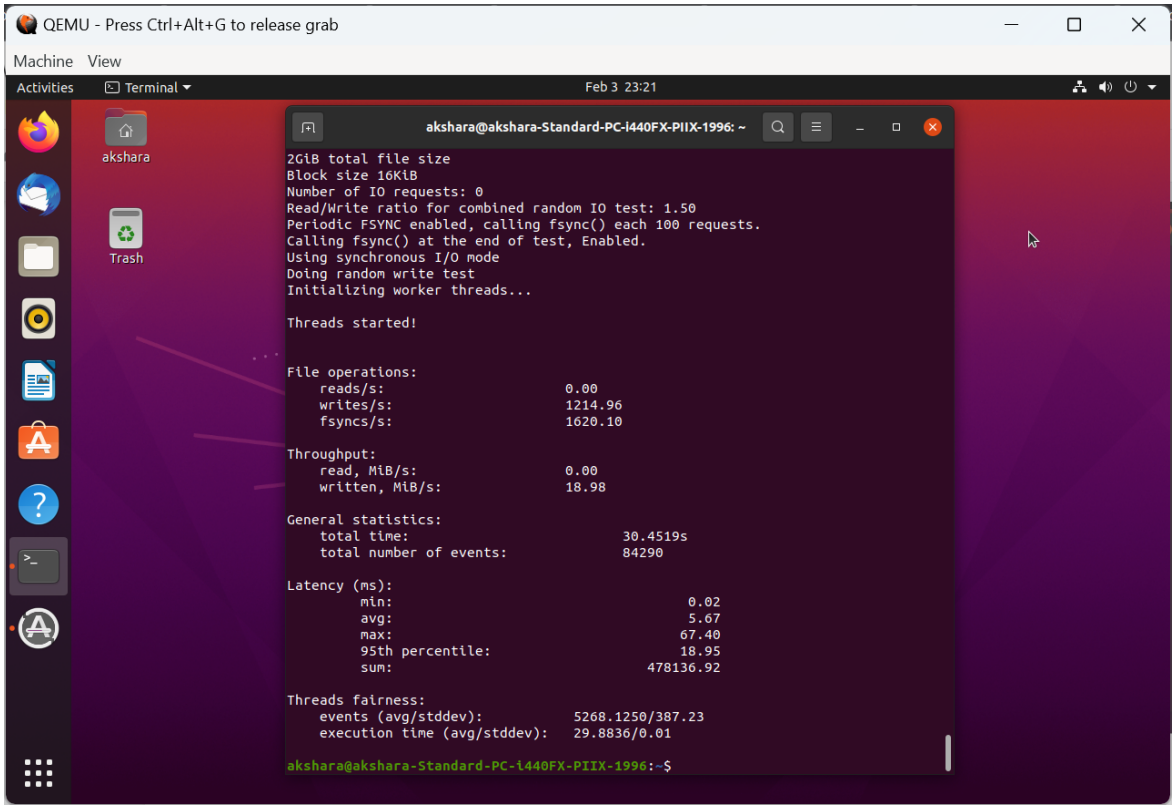
After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	18.89	30.4300
2	0	18.60	30.5234
3	0	18.25	30.4125
4	0	18.63	30.6327
5	0	18.85	30.4481
Minimum	0	18.25	30.6327
Maximum	0	18.89	30.4125
Average	0	18.644	30.48934
Standard Deviation	0	0.228175	0.081041

2. `sysbench –num-threads=16 –test=fileio –time=30 –file-total-size=2G –file-test-mode=rndwr prepare`

```
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=2G –file-test-mode=rndwr run
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=2G –file-test-mode=rndwr cleanup
```

Output:



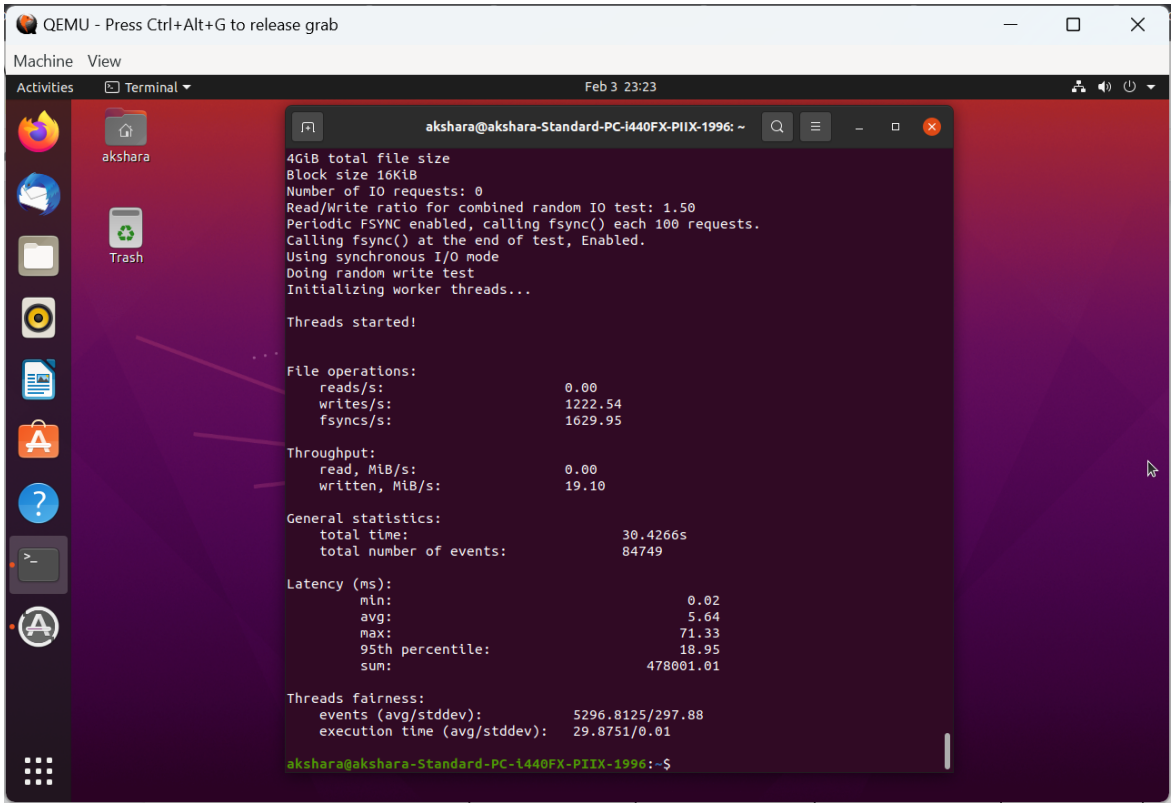
After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	18.98	30.4519
2	0	18.36	30.5684
3	0	18.24	30.4261
4	0	18.94	30.3957
5	0	18.96	30.3912
Minimum	0	18.24	30.3912
Maximum	0	18.98	30.5684

Average	0	18.696	30.44666
Standard Deviation	0	0.325797	0.0647202

3.
```
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=4G –file-test-mode=rndwr prepare
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=4G –file-test-mode=rndwr run
sysbench –num-threads=16 –test=fileio –time=30
–file-total-size=4G –file-test-mode=rndwr cleanup
```

Output:



After 5 iterations:

Test Number	Read Throughput	Write Throughput	Total Time
1	0	19.10	30.4266
2	0	19.23	30.4127
3	0	18.14	30.2257

4	0	18.68	30.3587
5	0	18.54	30.3988
Minimum	0	18.14	30.2257
Maximum	0	19.23	30.4266
Average	0	18.738	30.3645
Standard Deviation	0	0.3932632	0.0730139

Conclusion for Sequential Rewrite:

After comparing the experiment performed on Docker Ubuntu VM and QEMU Ubuntu VM, the conclusion is that the sequential rewrite for QEMU VM is much faster than the Docker VM.

Conclusion for Combined Random Read/Write:

After comparing the experiment performed on Docker Ubuntu VM and QEMU Ubuntu VM, the conclusion is the combined random read/write for QEMU VM is 3 times better than that of the Docker VM

Git Hub Repository Information:

Repository link for all projects: https://github.com/anittala/cloud-computing
Repository link for HW1: https://github.com/anittala/cloud-computing/tree/main/HW1
Username of of github account: anittala