

SYSTEM VS OS VIRTUALIZATION

Homework 1

Table of Contents

Introduction:	2
Installations steps:	2
Qemu:	2
Docker:	5
CPU tests:	6
Qemu:	6
Commands.....	6
Bash scripts:	6
Output table:	7
Docker:	8
Commands.....	8
Bash scripts:	9
Output table:	9
File IO tests:	10
Qemu:	10
Commands.....	10
Bash scripts:	12
Docker:	13
Commands.....	13
.....	13
Bash scripts:	14
Output table:	15
Conclusion:	15
Github Details:	15

Introduction:

This report compares two virtualization approaches and covers their setup, configuration, operations, benchmarking, and detailed findings.

System Virtualization (Qemu)

OS Virtualization (Docker)

Installations steps:

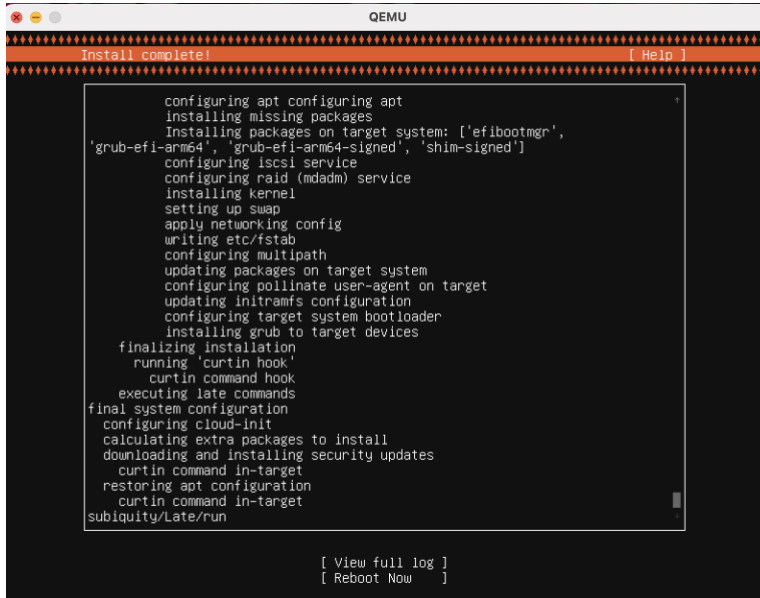
The procedures for installing QEMU and Docker in this portion of the report. The installation will vary based on each person's system setup. I'll describe how did I install MacBook air with M1 processors in this section.

Qemu:

- 1) Downloaded ARM (Apple Silicon): **Ubuntu 22.04 Server for ARM**
- 2) Installed qemu with **brew install qemu**
- 3) Run qemu command with options including iso image, raw image etc.

```
qemu-system-aarch64 \  
-monitor stdio \  
-M virt,highmem=off \  
-accel hvf \  
-cpu host \  
-smp 4 \  
-m 3000 \  
-bios QEMU_EFI.fd \  
-device virtio-gpu-pci \  
-display default,show-cursor=on \  
-device qemu-xhci \  
-device usb-kbd \  
-device usb-tablet \  
-device intel-hda \  
-device hda-duplex \  
-drive file=ubuntu-latest.raw,format=raw,if=virtio,cache=writethrough \  
-cdrom ubuntu-22.04.1-live-server-arm64.iso
```

1. -m: Denotes memory. Since we specified 2G as the input value in the command above, we are allocating 2GB of RAM capacity for our VM.
2. Hardware acceleration is denoted by the suffix "-accel."
3. -smp: The number of cores is indicated. We have specified the value of the parameter as 2, which indicates that we have provided our VM with 2 cores.
- 4) It launched the Ubuntu installation process in qemu terminal.
- 5) Configured Ubuntu with proxy, username, password etc.



```
QEMU
Install complete! [ Help ]

configuring apt configuring apt
installing missing packages
Installing packages on target system: ['efibootmgr',
'grub-efi-arm64', 'grub-efi-arm64-signed', 'shim-signed']
configuring iscsi service
configuring raid (mdadm) service
installing kernel
setting up swap
apply networking config
writing etc/fstab
configuring multipath
updating packages on target system
configuring pollinate user-agent on target
updating initramfs configuration
configuring target system bootloader
installing grub to target devices
finalizing installation
running 'curtin hook'
curtin command hook
executing late commands
final system configuration
configuring cloud-init
calculating extra packages to install
downloading and installing security updates
curtin command in-target
restoring apt configuration
curtin command in-target
subiquity/Late/run

[ View full log ]
[ Reboot Now ]
```

- 6) Using the same command with same set of options except -cdrom, qemu terminal is launched after installation



```
QEMU
Ubuntu 22.04.1 LTS pyeole tty1
pyeole login: pyeole
Password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-58-generic aarch64)

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

System information as of Wed Feb  1 09:17:30 PM UTC 2023

System load:        0.0732421875
Usage of /:         30.4% of 18.01GB
Memory usage:       7%
Swap usage:         0%
Processes:          121
Users logged in:    0
IPv4 address for enp0s1: 10.0.2.15
IPv6 address for enp0s1: fec0::5054:ff:fe12:3456

62 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

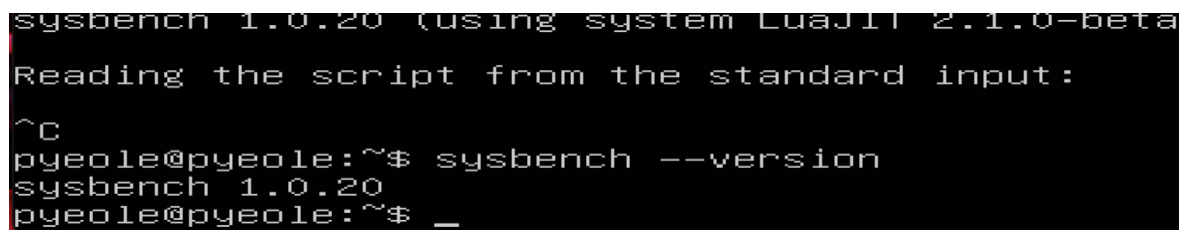
Last login: Wed Feb  1 21:15:40 UTC 2023 on tty1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

pyeole@pyeole:~$ ls
testfile
pyeole@pyeole:~$ _
```

- 7) Then First sysbench is installed qemu ubuntu with following commands:

apt update

apt install sysbench



```
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta)
Reading the script from the standard input:
^C
pyeole@pyeole:~$ sysbench --version
sysbench 1.0.20
pyeole@pyeole:~$ _
```

Docker:

- 1) Installed docker with <https://docs.docker.com/desktop/mac/install/>
- 2) Pulled the ubuntu image with the same version like qemu: i.e Ubuntu:22.04
- 3) Run the docker image with the following command which will start the container

Docker run -it --entrypoint "/bin/bash" arm64v8/ubuntu:22.04

```

-- -zsh
Last login: Wed Feb 1 13:14:34 on ttys003
pradnyaeyeole@Pradnyas-MacBook-Air ~ % docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
alpine/git           latest       9793ee61fc75   2 months ago  43.4MB
hello-world          latest       46331d942d63   10 months ago 9.14kB
zyclonite/sysbench   latest       8731aa4184ff   13 months ago 9.19MB
csmnpp/ubuntu-sysbench latest       2787c5e16909   7 years ago   336MB
pradnyaeyeole@Pradnyas-MacBook-Air ~ % docker pull arm64v8/ubuntu
Using default tag: latest
latest: Pulling from arm64v8/ubuntu
8b150fd943bc: Pull complete
Digest: sha256:61bd0b97000996232eb07b8d0e9375d14197f78aa850c2506417ef995a7199a7
Status: Downloaded newer image for arm64v8/ubuntu:latest
docker.io/arm64v8/ubuntu:latest
pradnyaeyeole@Pradnyas-MacBook-Air ~ % docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
arm64v8/ubuntu       latest       a6be1f66f70f   6 days ago     69.2MB
alpine/git           latest       9793ee61fc75   2 months ago  43.4MB
hello-world          latest       46331d942d63   10 months ago 9.14kB
zyclonite/sysbench   latest       8731aa4184ff   13 months ago 9.19MB
csmnpp/ubuntu-sysbench latest       2787c5e16909   7 years ago   336MB
pradnyaeyeole@Pradnyas-MacBook-Air ~ % docker run arm64v8/ubuntu
pradnyaeyeole@Pradnyas-MacBook-Air ~ % docker rmi a6be1f66f70f
Error response from daemon: conflict: unable to delete a6be1f66f70f (must be forced) - image is being used by stopped container 78b5e05285a5
pradnyaeyeole@Pradnyas-MacBook-Air ~ % docker rmi -f a6be1f66f70f
Untagged: arm64v8/ubuntu:latest
Untagged: arm64v8/ubuntu@sha256:61bd0b97000996232eb07b8d0e9375d14197f78aa850c2506417ef995a7199a7
Deleted: sha256:a6be1f66f70f66ef43503292e38ccbf14f2d5464e7736344783a8fc7bb339a8
pradnyaeyeole@Pradnyas-MacBook-Air ~ % docker pull arm64v8/ubuntu:22.04
22.04: Pulling from arm64v8/ubuntu
8b150fd943bc: Already exists
Digest: sha256:61bd0b97000996232eb07b8d0e9375d14197f78aa850c2506417ef995a7199a7
Status: Downloaded newer image for arm64v8/ubuntu:22.04
docker.io/arm64v8/ubuntu:22.04
pradnyaeyeole@Pradnyas-MacBook-Air ~ % docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
arm64v8/ubuntu       22.04        a6be1f66f70f   6 days ago     69.2MB
alpine/git           latest       9793ee61fc75   2 months ago  43.4MB
hello-world          latest       46331d942d63   10 months ago 9.14kB
zyclonite/sysbench   latest       8731aa4184ff   13 months ago 9.19MB
csmnpp/ubuntu-sysbench latest       2787c5e16909   7 years ago   336MB
pradnyaeyeole@Pradnyas-MacBook-Air ~ % docker run -it --entrypoint "/bin/bash" arm64v8/ubuntu:22.04
root@a2411568e373:/#

```

- 4) Installed Sysbench version in docker container with same commands.

```

Setting up libsasl2-2:arm64 (2.1.27+dfsg2-3ubuntu1.1) .
Setting up libaio1:arm64 (0.3.112-13build1) ...
Setting up libluajit-5.1-2:arm64 (2.1.0~beta3+dfsg-6) .
Setting up libldap-2.5-0:arm64 (2.5.13+dfsg-0ubuntu0.22
Setting up libpq5:arm64 (14.6-0ubuntu0.22.04.1) ...
Setting up sysbench (1.0.20+ds-2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
root@a2411568e373:/# sysbench --version
sysbench 1.0.20
root@a2411568e373:/#

```

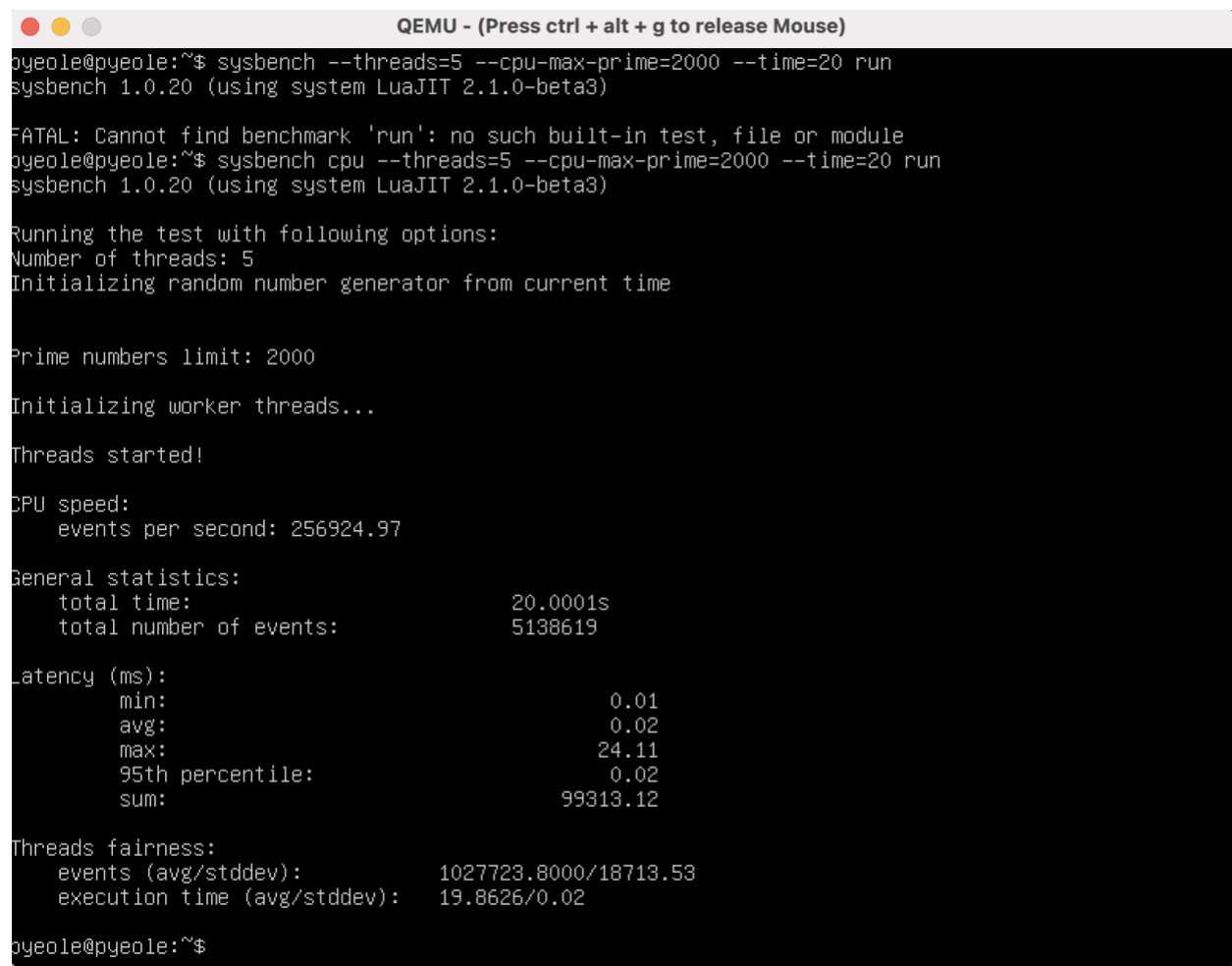
Here, the Installation process completes and now let's see how CPU and file IO tests are performed.

CPU tests:

Qemu:

Commands: `sysbench --threads=5 --cpu-max-prime=2000 --time=20 run`

With the same commands, different tests are performed with different sets of parameters. Example is shown in screenshot.



```
QEMU - (Press ctrl + alt + g to release Mouse)
pyeole@pyeole:~$ sysbench --threads=5 --cpu-max-prime=2000 --time=20 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

FATAL: Cannot find benchmark 'run': no such built-in test, file or module
pyeole@pyeole:~$ sysbench cpu --threads=5 --cpu-max-prime=2000 --time=20 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 5
Initializing random number generator from current time

Prime numbers limit: 2000
Initializing worker threads...
Threads started!

CPU speed:
  events per second: 256924.97

General statistics:
  total time:          20.0001s
  total number of events: 5138619

Latency (ms):
  min:                 0.01
  avg:                 0.02
  max:                 24.11
  95th percentile:    0.02
  sum:                 99313.12

Threads fairness:
  events (avg/stddev): 1027723.8000/18713.53
  execution time (avg/stddev): 19.8626/0.02

pyeole@pyeole:~$
```

Bash scripts:

In the Qemu folder,

cpuTest.sh is a bash script which ran 3 test cases 5 times. And stored output in cpuTest_QEMU_op.txt output file.

Output table:

Test1:

QEMU	20000					
	Events/Sec	Min ms	Max ms	Avg ms	std	95 percentile
1	140	6.63	16.79	7.1	0	8.13
2	141	6.61	13.53	7.03	0	7.56
3	134	6.62	6.11	7.43	0	9.22
4	138	6.62	28.81	71.9	0	8.13
5	138	6.67	23.88	7.19	0	7.98

Test 2:

QEMU	50000					
	Events/Sec	min	max	avg	std	95 percentile
1	41.34	23.13	43.24	24.11	0	25.28
2	41.32	22.9	30.65	24.1	0	25.28
3	41.24	23.04	33.52	24.16	0	25.28
4	41.2	23.15	43.63	24.17	0	25.28
5	41.34	23.3	28.61	24.1	0	24.83

Test 3:

	Events/Sec	Min ms	Max ms	avgms	std	95 percentile
1	18.59	52.15	62.32	53.63	0	55.82
2	18.75	51.61	58.11	53.2	0	54.83
3	18.24	51.99	167.36	54.67	0	55.82
4	18.63	52.17	61.77	53.5	0	54.83

5	18.54	52.16	86.25	53.79	0	54.83
---	-------	-------	-------	-------	---	-------

Docker:

In docker too, like qemu,

Commands: sysbench --threads=5 --cpu-max-prime=2000 --time=20 run

With the same commands, different tests are performed with different sets of parameters.
An example is shown in the screenshot.

```
root@a2411568e373:/HW1# cd ..
root@a2411568e373:/# sysbench cpu --threads=1 --cpu-max-prime=35000 --time=10 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)
```

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time
```

```
Prime numbers limit: 35000
```

```
Initializing worker threads...
```

```
Threads started!
```

```
CPU speed:
events per second: 1701.69
```

```
General statistics:
total time: 10.0004s
total number of events: 17020
```

```
Latency (ms):
min: 0.56
avg: 0.59
max: 10.97
95th percentile: 0.61
sum: 9989.72
```

```
Threads fairness:
events (avg/stddev): 17020.0000/0.00
execution time (avg/stddev): 9.9897/0.00
```

```
root@a2411568e373:/# █
```



```
[root@a2411568e373:/# sysbench cpu --threads=15 --cpu-max-prime=15000 --time=50 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)
```

```
Running the test with following options:
Number of threads: 15
Initializing random number generator from current time
```

```
Prime numbers limit: 15000
```

```
Initializing worker threads...
```

```
Threads started!
```

```
CPU speed:
events per second: 19166.15
```

```
General statistics:
total time:          50.0008s
total number of events: 958329
```

```
Latency (ms):
min:                0.20
avg:                0.78
max:                36.38
95th percentile:   0.23
sum:                749382.65
```

```
Threads fairness:
events (avg/stddev): 63888.6000/1337.34
execution time (avg/stddev): 49.9588/0.02
```

```
root@a2411568e373:/# █
```

Bash scripts:

In Docker folder,

cpuTest.sh is a bash script which ran 3 test cases 5 times. And stored output in cpuTest_Docker_op.txt output file.

Output table:

Test1:

Docker	20000					
	Events/Sec	min	max	avg	std	95 percentile
1	468	2.01ms	18446744073678.15ms	2.21ms	0	2.54ms
2	469	1.96ms	18446744073677.96ms	2.11ms	0	2.47ms

3	466	2.00ms	18446744073678.05ms	2.12ms	0	2.41ms
4	462	1.99ms	18446744073677.48ms	2.16ms	0	2.33ms
5	467	2.01ms	4.82ms	2.14ms	0	2.26ms

Test2:

Docker	50000					
	Events/Sec	Min ms	max	Avg ms	std	95 percentile ms
1	132	7.08	18446744073683.76ms	7.55	0	8.29
2	135	7.04	18446744073682.76ms	7.42	0	7.79
3	134	7.08	18446744073687.00ms	7.45	0	7.89
4	128	7.05	18446744073686.71ms	7.82	0	8.62
5	14	7.07	18446744073685.14ms	69.31	0	10.54

Test 3:

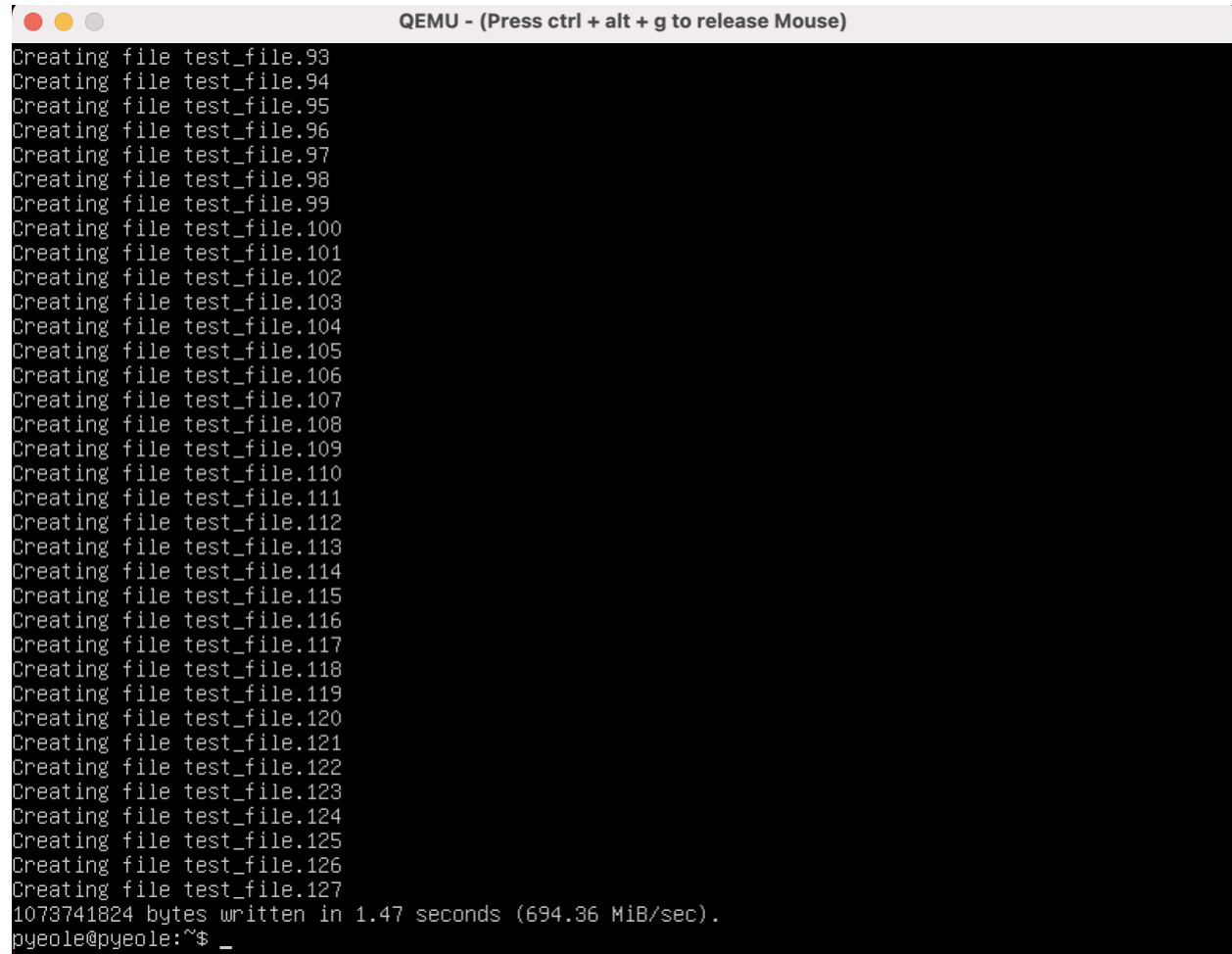
Dcoker	90000					
	Events/Sec	Min ms	Max ms	Avg ms	std	95 percentile ms
1	59	16.27	18446744073699.46ms	17.03	0	18.05
2	59	16.26	18446744073694.41ms	16.96	0	17.9
3	58	16.28	18446744073697.96ms	17.16	0	18.74
4	59	16.24	18446744073693.59ms	17.02	0	18.08
5	58	16.31	18446744073696.18ms	17.11	0	18.5

File IO tests:

Qemu:

Commands:

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



```
QEMU - (Press ctrl + alt + g to release Mouse)
Creating file test_file.93
Creating file test_file.94
Creating file test_file.95
Creating file test_file.96
Creating file test_file.97
Creating file test_file.98
Creating file test_file.99
Creating file test_file.100
Creating file test_file.101
Creating file test_file.102
Creating file test_file.103
Creating file test_file.104
Creating file test_file.105
Creating file test_file.106
Creating file test_file.107
Creating file test_file.108
Creating file test_file.109
Creating file test_file.110
Creating file test_file.111
Creating file test_file.112
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
1073741824 bytes written in 1.47 seconds (694.36 MiB/sec).
pyeole@pyeole:~$ _
```

```
QEMU - (Press ctrl + alt + g to release Mouse)
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         33343.55
  fsyncs/s:         42879.55

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   520.99

General statistics:
  total time:              10.0044s
  total number of events:  760559

Latency (ms):
  min:                    0.00
  avg:                    0.21
  max:                    9.22
  95th percentile:       0.92
  sum:                    159605.04

Threads fairness:
  events (avg/stddev):    47534.9375/821.73
  execution time (avg/stddev):  9.9753/0.00

pyeole@pyeole:~$ _

sum:                    159605.04

Threads fairness:
  events (avg/stddev):    47534.9375/821.73
  execution time (avg/stddev):  9.9753/0.00

pyeole@pyeole:~$ sysbench --num-threads=16 --test=fileio --file-total-size=1G --file-test-mode=rndwr
Cleanup
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Removing test files...
pyeole@pyeole:~$ _
```

Bash scripts:

In the Qemu folder,

fileioTest.sh is a bash script that ran 3 test cases 5 times. And stored output in fileioTest_QEMU_op.txt output file.

Docker:

Commands:

```
sysbench --num-threads=16 --test=fileio --file-total-size=1G --file-test-mode=rndwr prepare
```

```
sysbench --num-threads=16 --test=fileio --file-total-size=1G --file-test-mode=rndwr run
```

```
sysbench --num-threads=16 --test=fileio --file-total-size=1G --file-test-mode=rndwr cleanup
```

```
root@2411568e373:~# sysbench --num-threads=16 --test=fileio --file-total-size=1G --file-test-mode=rndwr prepare
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

128 files, 8192Kb each, 1024Mb total
Creating files for the test...
Extra file open flags: (none)
Creating file test_file.0
Creating file test_file.1
Creating file test_file.2
Creating file test_file.3
Creating file test_file.4
Creating file test_file.5
Creating file test_file.6
Creating file test_file.7
Creating file test_file.8
Creating file test_file.9
Creating file test_file.10
Creating file test_file.11
Creating file test_file.12
Creating file test_file.13
Creating file test_file.14
Creating file test_file.15
Creating file test_file.16
Creating file test_file.17
Creating file test_file.18
Creating file test_file.19
Creating file test_file.20
Creating file test_file.21
Creating file test_file.22
Creating file test_file.23
Creating file test_file.24
Creating file test_file.25
Creating file test_file.26
Creating file test_file.27
Creating file test_file.28
Creating file test_file.29
Creating file test_file.30
Creating file test_file.31
Creating file test_file.32
Creating file test_file.33
Creating file test_file.34
Creating file test_file.35
Creating file test_file.36
Creating file test_file.37
Creating file test_file.38
Creating file test_file.39
```

```
-- zsh      ...mebrew/share/qemu -- pyeole@linux10616:~ -- zsh      /opt/homebrew/share/qemu -- zsh      ..n -it --entrypoint /bin/bash arm64v8/ubuntu:22.04 +
root@a2411568e373:/# sysbench --num-threads=16 --test=fileio --file-total-size=1G --file-test-mode=rndwr run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
16iB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         30991.33
  fsyncs/s:        39865.00

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   484.24

General statistics:
  total time:       10.0149s
  total number of events: 707628

Latency (ms):
  min:              0.00
  avg:              0.23
  max:              5.13
  95th percentile: 0.70
  sum:              159706.28

Threads fairness:
  events (avg/stddev): 44226.7500/480.55
  execution time (avg/stddev): 9.9816/0.00

root@a2411568e373:/# sysbench --num-threads=16 --test=fileio --file-total-size=1G --file-test-mode=rndwr cleanup
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Removing test files...
root@a2411568e373:/#
```

Bash scripts:

In Docker folder,

fileioTest.sh is a bash script which ran 3 test cases 5 times. And stored output in fileioTest_Docker_op.txt output file.

Output table:

Docker						
	Transfer Rate Mb/sec	Requests/sec	min ms	avg ms	max ms	std
1	72.75	4656	0.02	0.15	0.62	91.93
2	57.67	3691	0.02	0.16	22.69	72.73
3	44.683	2859.74	0.02	0.13	7.29	79.52
4	42.87	2745	0.02	0.18	26.02	104.92
5	67.12	4296.22	0.02	0.18	8.94	91.78

Conclusion:

- 1) The CPU events/second is higher in OS virtualization than in System Virtualization.
- 2) When both System and OS virtualization's maxprime limit is increased, the number of CPU events per second drops.
- 3) When we increase the number of threads for fileIO testing, the throughput rises.
- 4) According to the CPU tests and fileio tests, OS virtualization performs better than System virtualization.

Github Details:

Repository link https://github.com/pradnyayeole/COEN241_CC_Winter23_pyeole

Folder: HW1

Commit hash: b7d53e8e06d799ba5b6ccc6302e3e7b93ce0efa2