

# HW 1 Report

## Cloud Computing, Winter 2024

Submitted by: Eshaan Rathi

GitHub Link: <https://github.com/eshaanrathi2/csen241/tree/main/hw1>

Important Note:

- All bash scripts (with automation) and terminal log / output is pushed to the [GitHub repo](#).
  - Proof of experiments section include 2 VMs with all (6) test cases ( $2 \times 6 = 12$  proofs). For all VMs and full terminal logs, visit above GitHub link.
  - Native system: Arm 64, Mac OS
  - Ubuntu VM version: ubuntu-20.04.5-live-server-arm64.iso
- 

### Creation of disk images:

#### 1. raw image:

Size = 30 G

Reserve space for VM's disk:

```
$ qemu-img create -f raw ubuntu_2.raw 30G
```

Copy "edk2-aarch64-code.fd" file associated with  
the "qemu-img" binary to the current directory:

```
$ cp /opt/homebrew/Cellar/qemu/8.2.0/share/qemu/edk2-aarch64-code.fd .
```

Create a file "ovmf\_vars.fd" with a size of 64 megabytes,  
filled with zero bytes:

```
$ dd if=/dev/zero conv=sync bs=1m count=64 of=ovmf_vars.fd
```

Install Ubuntu on QEMU as a cdrom:

```
$ qemu-system-aarch64 \  
-accel hvf -cpu host -M virt,highmem=off -m 3G -smp 4 \  
-drive file=edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \  
-drive file=ovmf_vars.fd,if=pflash,format=raw \  
-drive if=none,file=ubuntu_2.raw,format=raw,id=hd0 \  
-device virtio-blk-device,drive=hd0,serial="dummyserial" \  
-append "root=/dev/vda1 ro" \  
-nographic
```

```

-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-cdrom ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb \
-display default,show-cursor=on \
-chardev qemu-vdagent,id=ch1,name=vdagent,clipboard=on \
-device virtio-serial-pci \
-device virtserialport,chardev=ch1,id=ch1,name=com.redhat.spice.0
-monitor stdio

```

The created images (highlighted in terminal) are `ubuntu_2.raw` and `ubuntu_1.qcow2`

```

scripts > qemu > vm_setup > create_raw_image.txt
You, 3 hours ago | 1 author (You)
1 QEMU VM's image file format: raw
2 Size = 30 G
3
4 Reserve space for VM's disk:
5 $ qemu-img create -f raw ubuntu_2.raw 30G You, 3 hours ago * add vm setup
6
7
8 Copy "edk2-aarch64-code.fd" file associated with
9 the "qemu-img" binary to the current directory:
10 $ cp /opt/homebrew/Cellar/qemu/8.2.0/share/qemu/edk2-aarch64-code.fd .
11
12
13 Create a file "ovmf_vars.fd" with a size of 64 megabytes,
14 filled with zero bytes:
15 $ dd if=/dev/zero conv=sync bs=1m count=64 of=ovmf_vars.fd
16

eshaan@eshaans-mbp-2:~/hw1% ls
scripts
QEMU_EFI.fd
edk2-aarch64-code.fd
ovmf_vars.fd
ubuntu_1.qcow2
ubuntu_2.raw
ubuntu-20.04.5-live-server-arm64.iso
eshaan@eshaans-mbp-2:~/hw1%

```

## 2. qcow2:

QEMU VM's image file format: qcow2

Size = 30 G

Reserve space for VM's disk:

```
$ qemu-img create -f qcow2 ubuntu_1.qcow2 30G
```

Copy "edk2-aarch64-code.fd" file associated with

the "qemu-img" binary to the current directory:

```
$ cp /opt/homebrew/Cellar/qemu/8.2.0/share/qemu/edk2-aarch64-code.fd .
```

Create a file "ovmf\_vars.fd" with a size of 64 megabytes,

filled with zero bytes:

```
$ dd if=/dev/zero conv=sync bs=1m count=64 of=ovmf_vars.fd
```

Install Ubuntu on QEMU as a cdrom:

```
$ qemu-system-aarch64 \
-accel hvf -cpu host -M virt,highmem=off -m 3G -smp 6 \
-drive file=edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \
-drive file=ovmf_vars.fd,if=pflash,format=raw \
-drive if=none,file=ubuntu_1.qcow2,format=qcow2,id=hd0 \
-device virtio-blk-device,drive=hd0,serial="dummyserial" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-cdrom ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse -usb \
-display default,show-cursor=on \
-chardev qemu-vdagent,id=ch1,name=vdagent,clipboard=on \
-device virtio-serial-pci \
-device virtserialport,chardev=ch1,id=ch1,name=com.redhat.spice.0
-monitor stdio
```

### 3. Docker Image

Create a dockerfile:

```
# Use the base Ubuntu image
FROM ubuntu:20.04

# Install sysbench
RUN apt-get update
RUN apt-get install -y sysbench
RUN apt-get install -y bc

# Set the entry point to sysbench
CMD ["sysbench"]
```

Build Docker image:

```
$ docker build -t sysbench-image .
```

## Docker Image history:

The screenshot shows a terminal window with the following content:

```
eshaan@eshaans-mbp-2 container_setup % docker image history sysbench-image
IMAGE      CREATED     CREATED BY
2b37c3f6713a  57 seconds ago   CMD ["sysbench"]
dockerfile.v0
<missing>  57 seconds ago   RUN /bin/sh -c apt-get install -y bc # build...
<missing>  59 seconds ago   RUN /bin/sh -c apt-get install -y sysbench #...
<missing>  About a minute ago  RUN /bin/sh -c apt-get update # buildkit
dockerfile.v0
<missing>  10 days ago    /bin/sh -c #(nop)  CMD ["/bin/bash"]
<missing>  10 days ago    /bin/sh -c #(nop) ADD file:9497e9dbcd9a04e7...
<missing>  10 days ago    /bin/sh -c #(nop) LABEL org.opencontainers...
<missing>  10 days ago    /bin/sh -c #(nop) LABEL org.opencontainers...
<missing>  10 days ago    /bin/sh -c #(nop) ARG LAUNCHPAD_BUILD_ARCH
<missing>  10 days ago    /bin/sh -c #(nop) ARG RELEASE
eshaan@eshaans-mbp-2 container_setup %
```

The terminal status bar at the bottom indicates: Ln 10, Col 17 Spaces: 4 UTF-8 LF Dockerfile

# Configurations of VM and Containers

## QEMU VM:

- VM 1: image format = raw, disk size = 30G, memory = 3G, #CPU cores = 4
- VM 2: image format = raw, disk size = 30G, memory = 3G, #CPU cores = 6
- VM 3: image format = raw, disk size = 30G, memory = 2G, #CPU cores = 4
- VM 4: image format = raw, disk size = 30G, memory = 2G, #CPU cores = 6
- VM 5: image format = qcow2, disk size = 30G, memory = 3G, #CPU cores = 4
- VM 6: image format = qcow2, disk size = 30G, memory = 3G, #CPU cores = 6
- VM 7: image format = qcow2, disk size = 30G, memory = 2G, #CPU cores = 4
- VM 8: image format = qcow2, disk size = 30G, memory = 2G, #CPU cores = 6

## Docker Containers:

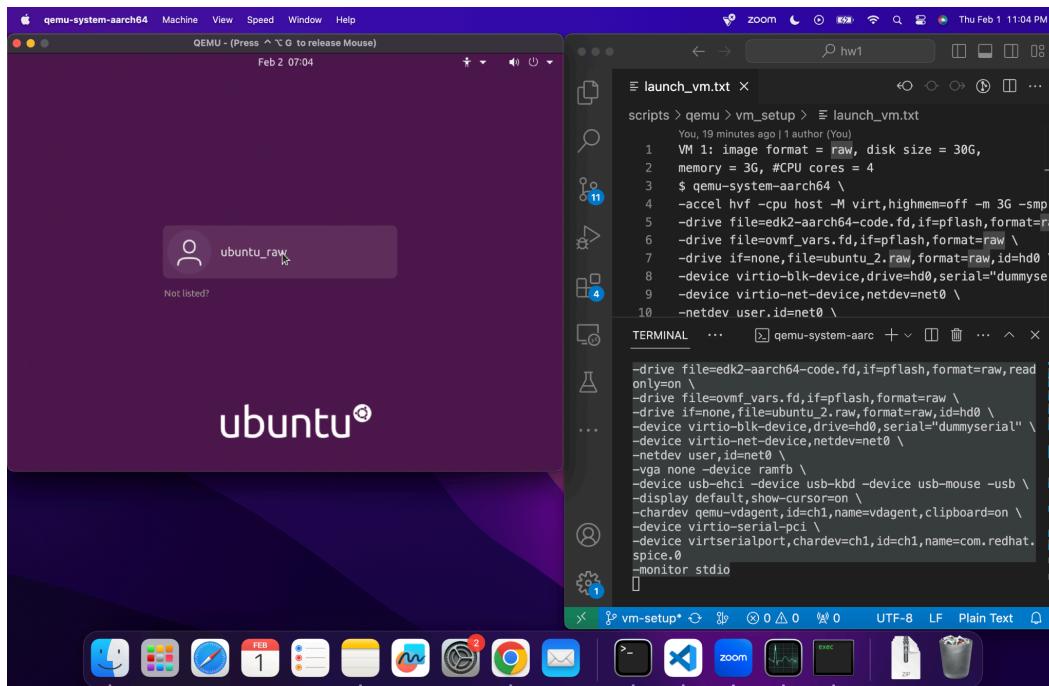
- Container 1: memory = 3G, #CPU cores = 4
  - Container 2: memory = 3G, #CPU cores = 2
  - Container 3: memory = 2G, #CPU cores = 4
  - Container 4: memory = 2G, #CPU cores = 2
-

# Launching QEMU VMs

Command for starting a VM is similar to the last command above, just that we don't need a cdrom to boot the ubuntu image.

For example, to start a VM with 3G of memory and 4 CPU cores (on a 30G raw img):

```
$ qemu-system-aarch64 \
-accel hvf -cpu host -M virt,highmem=off -m 3G -smp 4 \
-drive file=edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \
-drive file=ovmf_vars.fd,if=pflash,format=raw \
-drive if=none,file=ubuntu_2.raw,format=raw,id=hd0 \
-device virtio-blk-device,drive=hd0,serial="dummyserial" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-device usb-ehci -device usb-kbd -device usb-mouse -usb \
-display default,show-cursor=on \
-chardev qemu-vdagent,id=ch1,name=vdagent,clipboard=on \
-device virtio-serial-pci \
-device virtserialport,chardev=ch1,id=ch1,name=com.redhat.spice.0
-monitor stdio
```



The figure shows how an Ubuntu VM is being launched using the above script.

# Launching Docker Containers

The following steps are to start a container1 and perform sysbench tests. Similar can be done for other containers.

Mount host directory (where testing scripts are) and start docker in interactive mode:

```
$ docker run -it --rm --cpus 4 --memory 3G -v  
/Users/eshaan/Desktop/academic/SCU/quarter6-winter24/cloud/csen241_git/hw1/scripts/docker/testing:/mnt sysbench-image /bin/bash
```

Create separate directories for each containers:

```
$ mkdir container1  
$ cd container1
```

Run bash scripts for Sysbench testing:

```
$ ./cpu_test.sh  
$ ./memory_test.sh  
$ ./fileio_test.sh
```

Sysbench results on Docker can now be seen in host filesystem /testing directory:  
scripts/docker/testing.

---

# Shell scripts for Sysbench Testing (with 5x Automation)

Shell scripts are same for all experiments of QEMU and Docker.

1. cpu\_test.sh:

[https://github.com/eshaanrathi2/csen241/blob/main/hw1/scripts/docker/testing/cpu\\_test.sh](https://github.com/eshaanrathi2/csen241/blob/main/hw1/scripts/docker/testing/cpu_test.sh)

Measures CPU workload as events/second for 5 Runs of 10000 and 15000 events.

2. memory\_test.sh:

[https://github.com/eshaanrathi2/csen241/blob/main/hw1/scripts/docker/testing/memory\\_test.sh](https://github.com/eshaanrathi2/csen241/blob/main/hw1/scripts/docker/testing/memory_test.sh)

Measures memory workload as events/second for 5 Runs of block size 2K and 4K with total memory size 15G.

3. fileio\_test.sh:

[https://github.com/eshaanrathi2/csen241/blob/main/hw1/scripts/docker/testing/fileio\\_test.sh](https://github.com/eshaanrathi2/csen241/blob/main/hw1/scripts/docker/testing/fileio_test.sh)

Measures Disk workload as Throughput (reads/sec and writes/sec) for 5 Runs of block size 2K and 4K with total memory size 15G.

Create them into executables by:

```
$ chmod +x ./cpu_test.sh
```

To run a script:

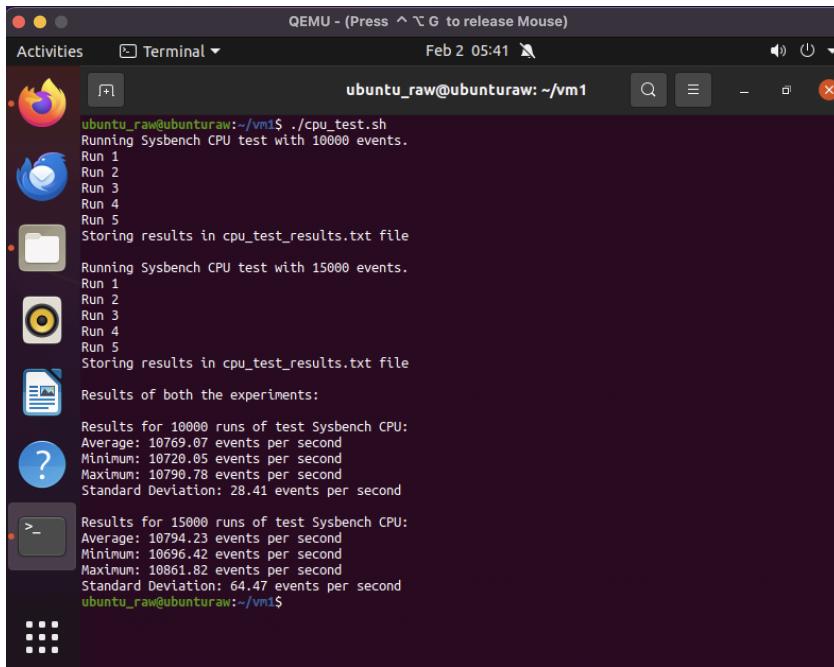
```
$ ./cpu_test.sh
```

---

# Proof of experiment (QEMU + Docker)

## QEMU VM 1:

CPU tests:



QEMU - (Press ⌘ G to release Mouse)

Activities Terminal Feb 2 05:41

```
ubuntu_raw@ubunturaw:~/vm1$ ./cpu_test.sh
Running Sysbench CPU test with 10000 events.
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in cpu_test_results.txt file

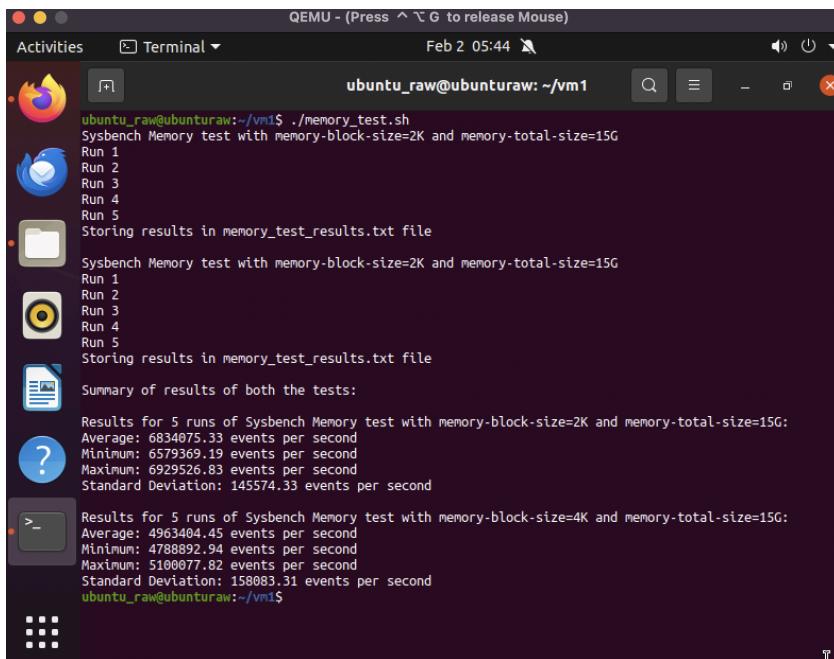
Running Sysbench CPU test with 15000 events.
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in cpu_test_results.txt file

Results of both the experiments:

Results for 10000 runs of test Sysbench CPU:
Average: 10769.07 events per second
Minimum: 10720.05 events per second
Maximum: 10790.78 events per second
Standard Deviation: 28.41 events per second

Results for 15000 runs of test Sysbench CPU:
Average: 10794.23 events per second
Minimum: 10696.42 events per second
Maximum: 10861.82 events per second
Standard Deviation: 64.47 events per second
ubuntu_raw@ubunturaw:~/vm1$
```

Memory tests:



QEMU - (Press ⌘ G to release Mouse)

Activities Terminal Feb 2 05:44

```
ubuntu_raw@ubunturaw:~/vm1$ ./memory_test.sh
Sysbench Memory test with memory-block-size=2K and memory-total-size=15G
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in memory_test_results.txt file

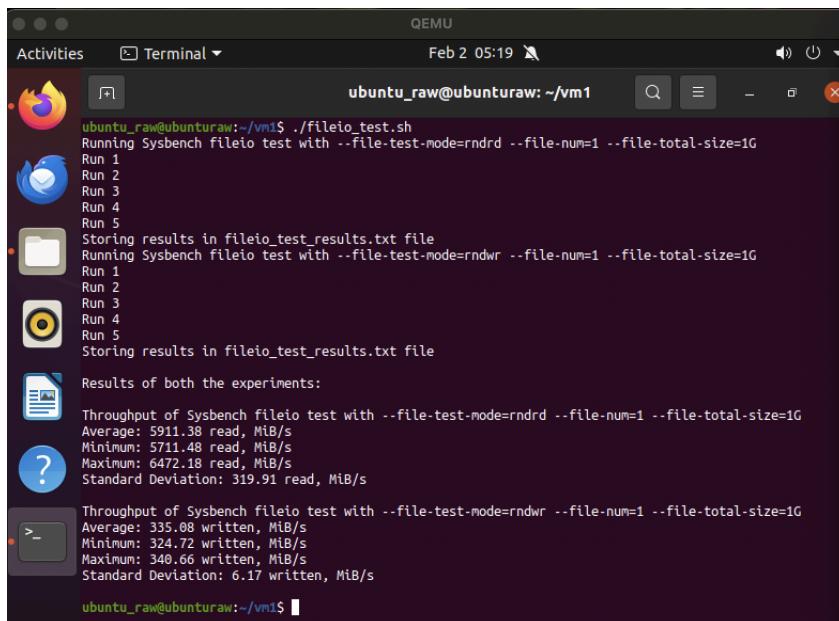
Sysbench Memory test with memory-block-size=2K and memory-total-size=15G
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in memory_test_results.txt file

Summary of results of both the tests:

Results for 5 runs of Sysbench Memory test with memory-block-size=2K and memory-total-size=15G:
Average: 6834075.33 events per second
Minimum: 6579369.19 events per second
Maximum: 6929526.83 events per second
Standard Deviation: 145574.33 events per second

Results for 5 runs of Sysbench Memory test with memory-block-size=4K and memory-total-size=15G:
Average: 4963404.45 events per second
Minimum: 4788892.94 events per second
Maximum: 5100077.82 events per second
Standard Deviation: 158083.31 events per second
ubuntu_raw@ubunturaw:~/vm1$
```

## Fileio tests:



The screenshot shows a terminal window titled "QEMU" running on an Ubuntu system. The command ./fileio\_test.sh is run, performing five runs of a Sysbench fileio test with 10 total size. It then stores results in fileio\_test\_results.txt. The throughput for the first test (rndrd mode) is summarized as follows:

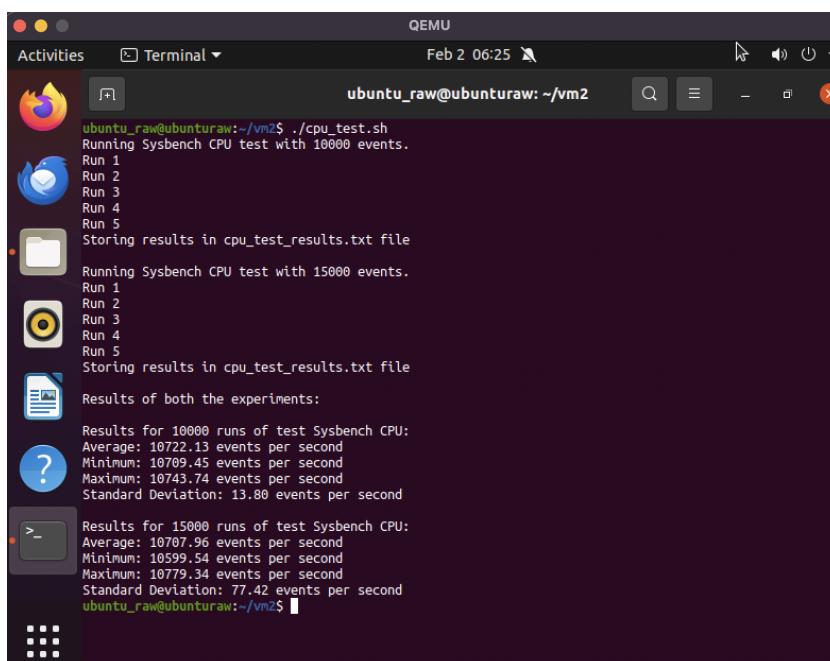
Average	Minimum	Maximum	Standard Deviation
5911.38 read, MiB/s	5711.48 read, MiB/s	6472.18 read, MiB/s	319.91 read, MiB/s

For the second test (rndwr mode), the throughput is summarized as follows:

Average	Minimum	Maximum	Standard Deviation
335.08 written, MiB/s	324.72 written, MiB/s	340.66 written, MiB/s	6.17 written, MiB/s

## QEMU VM2:

### CPU tests:



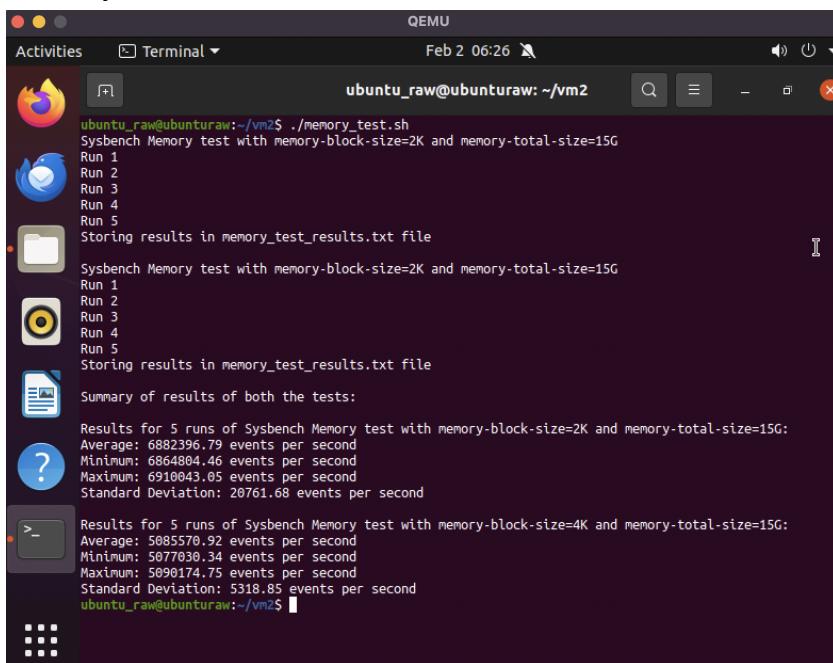
The screenshot shows a terminal window titled "QEMU" running on an Ubuntu system. The command ./cpu\_test.sh is run, performing five runs of a Sysbench CPU test with 10000 events. It then stores results in cpu\_test\_results.txt. The results for the first test (10000 events) are summarized as follows:

Average	Minimum	Maximum	Standard Deviation
10722.13 events per second	10709.45 events per second	10743.74 events per second	13.80 events per second

The results for the second test (15000 events) are summarized as follows:

Average	Minimum	Maximum	Standard Deviation
10707.96 events per second	10599.54 events per second	10779.34 events per second	77.42 events per second

## Memory tests:

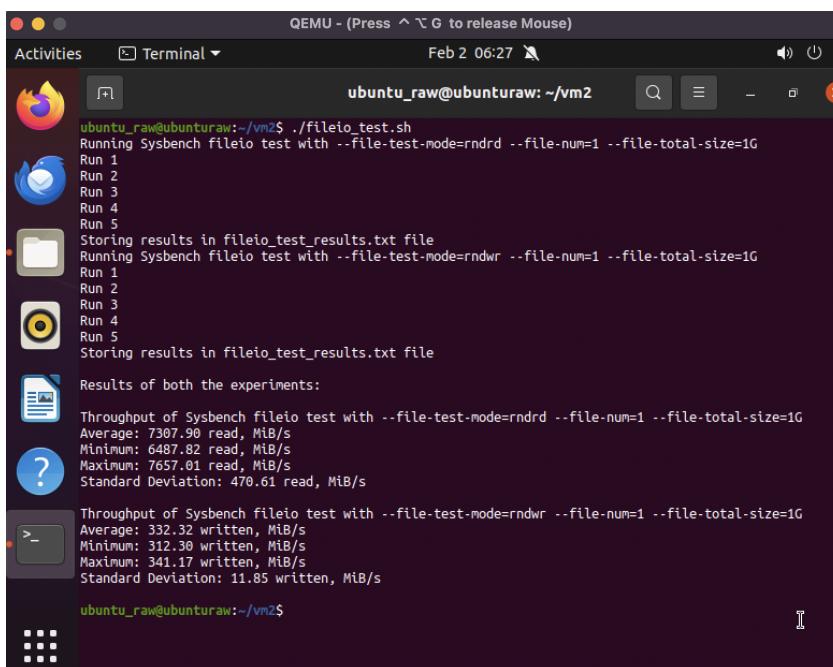


QEMU

Activities Terminal Feb 2 06:26

```
ubuntu_raw@ubunturaw:~/vm2$ ./memory_test.sh
Sysbench Memory test with memory-block-size=2K and memory-total-size=1G
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in memory_test_results.txt file
Sysbench Memory test with memory-block-size=2K and memory-total-size=1G
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in memory_test_results.txt file
Summary of results of both the tests:
Results for 5 runs of Sysbench Memory test with memory-block-size=2K and memory-total-size=1G:
Average: 6882396.79 events per second
Minimum: 6864804.46 events per second
Maximum: 6910043.05 events per second
Standard Deviation: 20761.68 events per second
Results for 5 runs of Sysbench Memory test with memory-block-size=4K and memory-total-size=1G:
Average: 5885570.92 events per second
Minimum: 5877030.34 events per second
Maximum: 5890174.75 events per second
Standard Deviation: 5318.85 events per second
ubuntu_raw@ubunturaw:~/vm2$
```

## Fileio tests:



QEMU - (Press ^ C to release Mouse)

Activities Terminal Feb 2 06:27

```
ubuntu_raw@ubunturaw:~/vm2$ ./fileio_test.sh
Running Sysbench fileio test with --file-test-mode=rndrd --file-num=1 --file-total-size=1G
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in fileio_test_results.txt file
Running Sysbench fileio test with --file-test-mode=rndwr --file-num=1 --file-total-size=1G
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in fileio_test_results.txt file
Results of both the experiments:
Throughput of Sysbench fileio test with --file-test-mode=rndrd --file-num=1 --file-total-size=1G
Average: 7307.90 read, MiB/s
Minimum: 6487.82 read, MiB/s
Maximum: 7657.01 read, MiB/s
Standard Deviation: 470.61 read, MiB/s
Throughput of Sysbench fileio test with --file-test-mode=rndwr --file-num=1 --file-total-size=1G
Average: 332.32 written, MiB/s
Minimum: 312.30 written, MiB/s
Maximum: 341.17 written, MiB/s
Standard Deviation: 11.85 written, MiB/s
ubuntu_raw@ubunturaw:~/vm2$
```

For rest of the VMs (3-4 and 7-8) see

<https://github.com/eshaanrathi2/csen241/tree/main/hw1/results/qemu>

# Docker Container 1:

## CPU tests:

The screenshot shows a terminal window titled "hw1" with the following content:

```
EXPLORER          E docker > testing > container1 > E cpu_test_results.txt U
scripts > docker > testing > container1 > E cpu_test_results.txt
  1 Results for 10000 runs of test Sysbench CPU:
  2 Average: 8367.19 events per second
  3 Minimum: 8313.11 events per second
  4 Maximum: 8394.14 events per second
  5 Standard Deviation: 31.24 events per second

  6 Results for 15000 runs of test Sysbench CPU:
  7 Average: 8363.40 events per second
  8 Minimum: 8326.99 events per second
  9 Maximum: 8378.41 events per second
 10 Standard Deviation: 21.38 events per second

> qemu
 11 root@082f321fc17b:/mnt/container1# ./cpu_test.sh
Running Sysbench CPU test with 10000 events.
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in cpu_test_results.txt file

 12 root@082f321fc17b:/mnt/container1# ./cpu_test.sh
Running Sysbench CPU test with 15000 events.
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in cpu_test_results.txt file

Results of both the experiments:

Results for 10000 runs of test Sysbench CPU:
Average: 8367.19 events per second
Minimum: 8313.11 events per second
Maximum: 8394.14 events per second
Standard Deviation: 31.24 events per second

Results for 15000 runs of test Sysbench CPU:
Average: 8363.40 events per second
Minimum: 8326.99 events per second
Maximum: 8378.41 events per second
```

## Memory tests

The screenshot shows a terminal window titled "hw1" with the following content:

```
EXPLORER          E docker > testing > container1 > E memory_test_results.txt U
scripts > docker > testing > container1 > E memory_test_results.txt
  2 Results for 5 runs of Sysbench Memory test with memory-block-size=2K and memory-total-size=15G:
  3 Average: 5279007.34 events per second
  4 Minimum: 5260766.60 events per second
  5 Maximum: 5287293.26 events per second
  6 Standard Deviation: 11208.89 events per second

  7 Results for 5 runs of Sysbench Memory test with memory-block-size=4K and memory-total-size=15G:
  8 Average: 3902347.84 events per second
  9 Minimum: 3897472.49 events per second
 10 Maximum: 3904826.21 events per second
 11 Standard Deviation: 2942.32 events per second

> qemu
 12 root@082f321fc17b:/mnt/container1# ./memory_test.sh
Sysbench Memory test with memory-block-size=2K and memory-total-size=15G
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in memory_test_results.txt file

 13 root@082f321fc17b:/mnt/container1# ./memory_test.sh
Sysbench Memory test with memory-block-size=2K and memory-total-size=15G
Run 1
Run 2
Run 3
Run 4
Run 5
Storing results in memory_test_results.txt file

Summary of results of both the tests:

Results for 5 runs of Sysbench Memory test with memory-block-size=2K and memory-total-size=15G:
Average: 5279007.34 events per second
Minimum: 5260766.60 events per second
Maximum: 5287293.26 events per second
Standard Deviation: 11208.89 events per second

Results for 5 runs of Sysbench Memory test with memory-block-size=4K and memory-total-size=15G:
Average: 3902347.84 events per second
Minimum: 3897472.49 events per second
Maximum: 3904826.21 events per second
Standard Deviation: 2942.32 events per second
root@082f321fc17b:/mnt/container1#
```

## Fileio tests:

**For rest of the Containers (2-4) see**

<https://github.com/eshaanrathi2/csen241/tree/main/hw1/results/docker>

## Results

All testing scripts and results are under

<https://github.com/eshaanrathi2/csen241/tree/main/hw1/results>

### VM1:

Results for 10000 runs of test Sysbench CPU:

Average: 10769.07 events per second

Minimum: 10720.05 events per second

Maximum: 10790.78 events per second

Standard Deviation: 28.41 events per second

Results for 15000 runs of test Sysbench CPU:

Average: 10794.23 events per second

Minimum: 10696.42 events per second

Maximum: 10861.82 events per second

Standard Deviation: 64.47 events per second

Results for 5 runs of Sysbench Memory test with memory-block-size=2K and memory-total-size=15G:

Average: 6834075.33 events per second

Minimum: 6579369.19 events per second

Maximum: 6929526.83 events per second

Standard Deviation: 145574.33 events per second

Results for 5 runs of Sysbench Memory test with memory-block-size=4K and memory-total-size=15G:

Average: 4963404.45 events per second

Minimum: 4788892.94 events per second

Maximum: 5100077.82 events per second

Standard Deviation: 158083.31 events per second

Throughput of Sysbench fileio test with --file-test-mode=rndrd --file-num=1 --file-total-size=1G

Average: 5911.38 read, MiB/s

Minimum: 5711.48 read, MiB/s

Maximum: 6472.18 read, MiB/s

Standard Deviation: 319.91 read, MiB/s

Throughput of Sysbench fileio test with --file-test-mode=rndwr --file-num=1  
--file-total-size=1G

Average: 335.08 written, MiB/s

Minimum: 324.72 written, MiB/s

Maximum: 340.66 written, MiB/s

Standard Deviation: 6.17 written, MiB/s

## VM 2:

Results for 10000 runs of test Sysbench CPU:

Average: 10722.13 events per second

Minimum: 10709.45 events per second

Maximum: 10743.74 events per second

Standard Deviation: 13.80 events per second

Results for 15000 runs of test Sysbench CPU:

Average: 10707.96 events per second

Minimum: 10599.54 events per second

Maximum: 10779.34 events per second

Standard Deviation: 77.42 events per second

Results for 5 runs of Sysbench Memory test with memory-block-size=2K and  
memory-total-size=15G:

Average: 6882396.79 events per second

Minimum: 6864804.46 events per second

Maximum: 6910043.05 events per second

Standard Deviation: 20761.68 events per second

Results for 5 runs of Sysbench Memory test with memory-block-size=4K and  
memory-total-size=15G:

Average: 5085570.92 events per second

Minimum: 5077030.34 events per second

Maximum: 5090174.75 events per second

Standard Deviation: 5318.85 events per second

Throughput of Sysbench fileio test with --file-test-mode=rndrd --file-num=1  
--file-total-size=1G

Average: 7307.90 read, MiB/s

Minimum: 6487.82 read, MiB/s

Maximum: 7657.01 read, MiB/s

Standard Deviation: 470.61 read, MiB/s

Throughput of Sysbench fileio test with --file-test-mode=rndwr --file-num=1  
--file-total-size=1G

Average: 332.32 written, MiB/s

Minimum: 312.30 written, MiB/s

Maximum: 341.17 written, MiB/s

Standard Deviation: 11.85 written, MiB/s

## For detailed results of all VMs, see

<https://github.com/eshaanrathi2/csen241/tree/main/hw1/results/qemu>

### Docker Container 1:

Results for 5 runs of Sysbench Memory test with memory-block-size=2K and  
memory-total-size=15G:

Average: 5279007.34 events per second

Minimum: 5260766.60 events per second

Maximum: 5287293.26 events per second

Standard Deviation: 11208.89 events per second

Results for 5 runs of Sysbench Memory test with memory-block-size=4K and  
memory-total-size=15G:

Average: 3902347.84 events per second

Minimum: 3897472.49 events per second

Maximum: 3904826.21 events per second

Standard Deviation: 2942.32 events per second

Results for 5 runs of Sysbench Memory test with memory-block-size=2K and  
memory-total-size=15G:

Average: 5279007.34 events per second

Minimum: 5260766.60 events per second

Maximum: 5287293.26 events per second

Standard Deviation: 11208.89 events per second

Results for 5 runs of Sysbench Memory test with memory-block-size=4K and  
memory-total-size=15G:

Average: 3902347.84 events per second

Minimum: 3897472.49 events per second

Maximum: 3904826.21 events per second

Standard Deviation: 2942.32 events per second

Throughput of Sysbench fileio test with --file-test-mode=rndrd --file-num=1  
--file-total-size=1G

Average: 32.43 read, MiB/s

Minimum: 32.09 read, MiB/s

Maximum: 33.11 read, MiB/s

Standard Deviation: .38 read, MiB/s

Throughput of Sysbench fileio test with --file-test-mode=rndwr --file-num=1  
--file-total-size=1G

Average: 31.09 written, MiB/s

Minimum: 30.75 written, MiB/s

Maximum: 31.42 written, MiB/s

Standard Deviation: .22 written, MiB/s

**For detailed results of all Containers, see**

<https://github.com/eshaanrathi2/csen241/tree/main/hw1/results/docker>

---

# Analysis of Results

## A. CPU:

- raw image format is faster by 20% compared to both qcow2 and docker.
- qcow2 and Docker are found to have similar performance.
- Across all docker configurations, container1 configuration was found to be the best performer in terms of CPU, conforming to the fact that it has the highest number of CPUs and Memory.
- Increasing memory and cores have little or no effect in raw image format.
- Increasing memory and cores have a positive effect in qcow2 image format.
- Docker performed slightly better than QEMU with qcow2

## B. Memory:

- raw image format is faster by 20% compared to both qcow2 and docker.
- Docker performed slightly better than QEMU with qcow2.
- Increasing Memory and number of CPUs have positive effect in all 3 image formats.
- VM2 was found to be the best performer.

## C. FileIO:

- qcow2 image format outperforms raw image format by 5-10 %.
  - Docker image was found to be 200 times slower than qcow2 format.
  - Reducing memory from 3G to 2G results in reduced performance by 20 times.
  - Increasing CPU from 4 to 6 or 2 to 4, increases the performance.
- 

# Conclusion

QEMU VMs outperform Docker Containers in CPU, Memory and especially in File IO tasks. Overall, raw image format is found to be the best performer. Perhaps the reason being it doesn't store metadata, doesn't have overhead of snapshots. But especially for file IO tasks, qcow2 can be a good choice over other formats as qcow2 dynamically compresses images to save space. Docker by far shows poor performance in file IO tasks, perhaps due to the mounting of the host filesystem, layering and then running containers on top.