

Sourabh Deshmukh  
W1648445

# **System and OS Virtualization**

**Santa Clara University**

Department of Computer Science and Engineering  
Cloud Computing CSEN - 241

Homework - 1

# Table of Contents

Environment Setup	
Host System Configuration	4
Virtualization Configuration	
QEMU Installation	6
QEMU Configuration Details	7
QEMU VM and Sysbench Setup	
Qcow2	8
Raw	12
Luks Encryption	16
Docker Installation	20
Sysbench Setup in Docker Container	22
Sysbench Experiment	
CPU Test in QEMU qcow2 VM	26
CPU Test in QEMU raw VM	31
CPU Test in Docker Container	35
Memory Test in QEMU qcow2 VM	39
Memory Test in QEMU VM	44
Memory Test in Docker Container	48
FILE-IO Test in QEMU qcow2 VM	53
FILE-IO Test in QEMU VM	58
FILE-IO Test in Docker Container	64
Experiment Observations & Analysis	
CPU Test - QEMU qcow2 VM vs Qemu raw VM vs. Docker Container	70

FileIO Test - QEMU qcow2 VM vs Qemu raw VM vs. Docker Container	71
Memory Test - QEMU qcow2 VM vs Qemu raw VM vs. Docker Container	73
Sysbench Findings and Conclusion Based on the Tests	76
Shell Script Used for Experimentation	
Shell Scripts for CPU, Memory, and FILE-IO Test	77
Automation Scripts	
Vagrantfile	80
Dockerfile	81
Resources	82

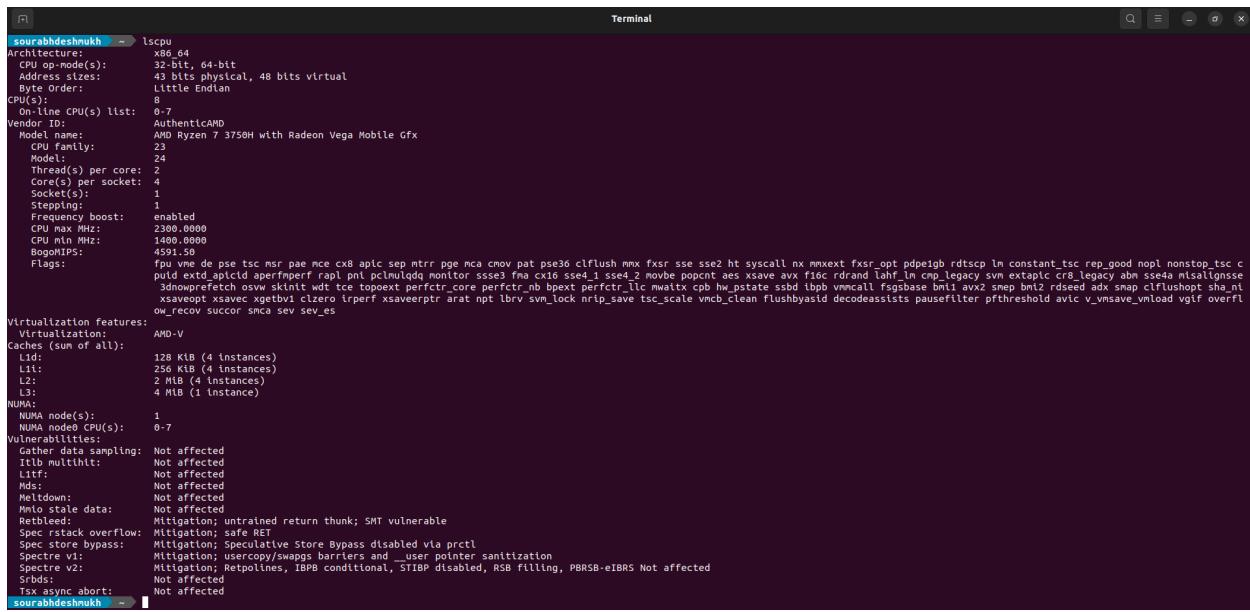
# Environment Setup

## Host system specification.

The configuration of the host system utilized for executing the experiment is as follows:

- CPU - x86\_64 AMD Ryzen 7 3750H 4-core/8-thread, 6MB cache, 4.0GHz max boost
- Memory - 16 GB RAM
- Virtualization - AMD-V
- Storage - 512GB PCIe® 3.0 NVMe™ M.2 SSD
- OS - Ubuntu 22.04.3 LTS
- OS Type - 64 Bit

```
sourabhdeshmukh$ lscpu
```

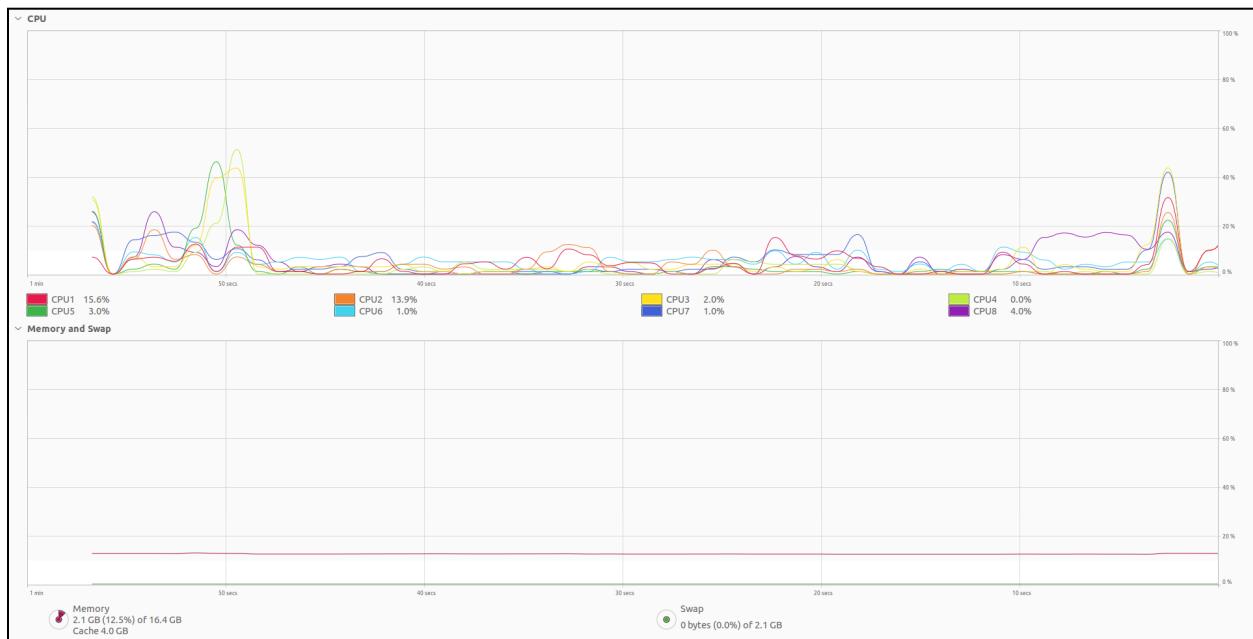


```
Architecture:          x86_64
CPU op-code set(s):   3d bits physical, 48 bits virtual
Byte Order:           Little Endian
CPU(s):               8
On-line CPU(s) list: 0-7
Vendor ID:            AuthenticAMD
Model name:           AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx
CPU Family:          23
Model:                24
Thread(s) per core:  2
Core(s) per socket:  4
Socket(s):           1
SMT threads:         1
Frequency boost:    enabled
CPU max MHz:        2300.0000
CPU min MHz:        1400.0000
BogomIPS:            4591.50
Flags:                fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sece2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good nopl nospec_longsnd pni pdi x86_ht sse41 sse42 sse43 monito sse3 fma cr4_ssd1 ssse3 3nomie popcnt aes msa avx file_adrna lsqr lsqrn legacy_smm extable cr4_legacy abm ssse3 misalignse3 dnowprefetch oswi skinit wdt tce topexec perfctr_nb hpxt perfctr_llc mwaltx cpb hw_pstate ssbd ibpb vmmcall fsqbase bml1 avx2 smp bml2 rdseed adx snap clflushopt sha_ni xsaveopt xsavec xgetbv1 clzero ioperf xsaveerptr arat npt lbrv svm_lock nrip_save tsc_scale vmcb_clean flushbyasid decodeassists pausefilter pfthreshold avic v_vmsave_vload vgif overfl ow_recov succor smca sev sev_es
Virtualization features:
Virtualization:       AMD-V
Caches (sum of all):
L1d:                 128 KIB (4 instances)
L1i:                 256 KIB (4 instances)
L2:                  2 MB (4 instances)
L3:                  4 MB (1 instance)
NUMA:
NUMA node(s):        1
NUMA node0 CPU(s):   0-7
Vulnerabilities:
Gather data sampling: Not affected
Itlb multithit:      Not affected
L1tf:                 Not affected
Mds:                  Not affected
Meltdown:             Not affected
Mmio stale data:     Not affected
Retbleed:              Mitigation; untrained return thunk; SMT vulnerable
Spec rstack overflow: Mitigation; safe RE
Spec store bypass:   Mitigation; Speculative Store Bypass disabled via prctl
Spectre v1:            Mitigation; __copy_to_user_swaps barriers and __user pointer sanitization
Spectre v2:            Mitigation; Retpolines, IBPB conditional, STIBP disabled, RSB filling, PBRSB-eIBRS Not affected
Srbds:                 Not affected
Tx6 async abort:      Not affected
sourabhdeshmukh$
```

## Disk

Device	Directory	Type	Total	Available	Used	
/dev/nvme0n1p5	/	ext4	157.4 GB	126.8 GB	22.5 GB	15%
/dev/nvme0n1p1	/boot/efi	vfat	100.7 MB	66.8 MB	33.8 MB	33%
/dev/nvme0n1p5	/var/snap/firefox/common/host-chromiumspell	ext4	157.4 GB	126.8 GB	22.5 GB	15%

## CPU



## Virtualization -

### Qemu VM Configurations -

- The Ubuntu Server 20.04.6 Virtual Machine is created with QEMU by allocating 10 GB of disk space with qcow2 file format, 2GB of RAM, and 2 CPU cores.
- The Ubuntu Server 20.04.6 Virtual Machine is created with QEMU by allocating 10 GB of disk space raw file format, 2 GB of RAM, and 2 CPU cores.
- The Ubuntu Server 20.04.6 Virtual Machine is created with QEMU by allocating 10 GB of disk space in the qcow2 file format, 2 GB of RAM, and 2 CPU cores. It is also encrypted using the LUKS encryption format. (extra credit)

### Docker Configurations -

- The Ubuntu container is created using the official Ubuntu image docker with configuration as 2 GB of RAM and 2 CPU cores.

# Virtualization Configuration

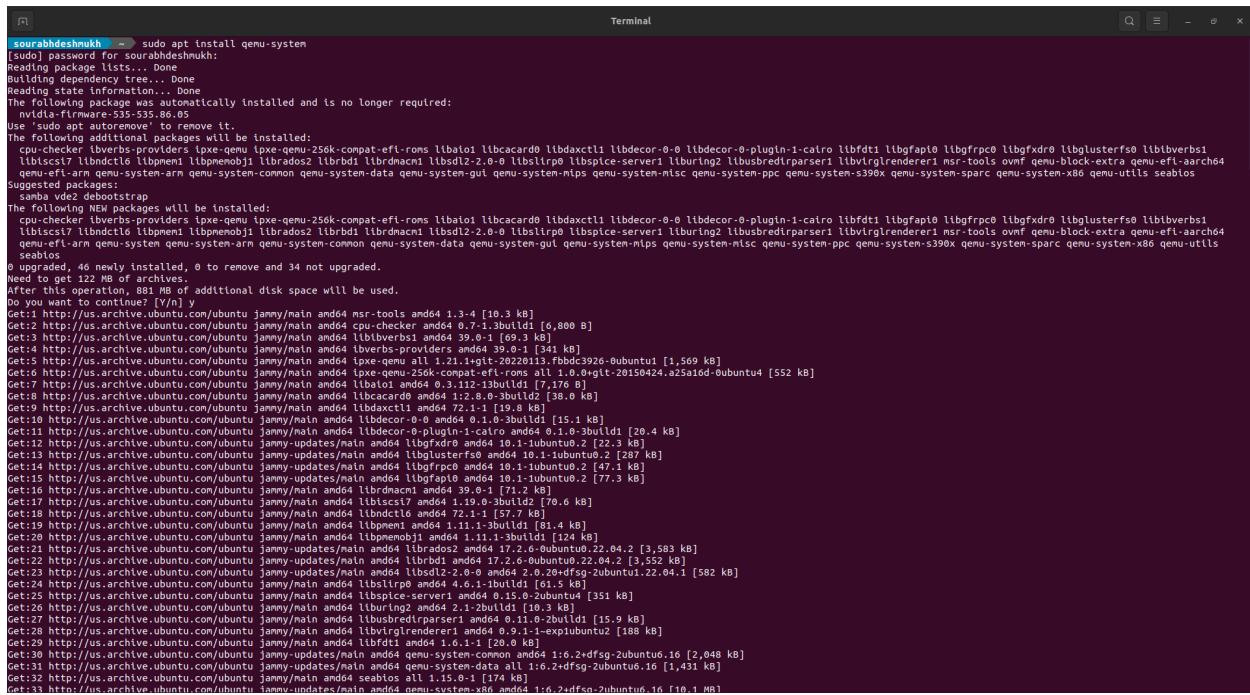
## Enable Virtualization in BIOS

- Enable virtualization in AMD Platform by setting SVM MODE to ENABLED.

## Qemu Installation -

1. Install the QEMU on the host system (here in this case Ubuntu) using the below command.

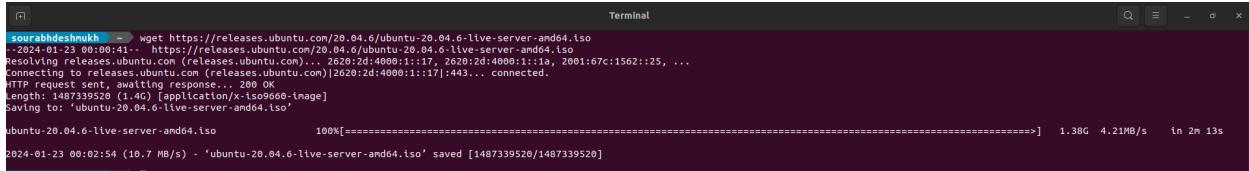
```
sourabhdeshmukh$ sudo apt install qemu-system -y
```



```
sourabhdeshmukh$ sudo apt install qemu-system
[sudo] password for sourabhdeshmukh:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  nvidia-firmware-535-535.86.05
Use 'apt remove' to remove it.
0 upgraded, 46 newly installed, 0 to remove and 34 not upgraded.
Need to get 122 MB of archives.
After this operation, 881 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 msr-tools amd64 1.3.4 [10.3 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 cpu-checker amd64 0.7-1.3build1 [6,800 B]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirt0 amd64 39.7+dfsg-0.1 [40 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libverbs-providers amd64 39.0.1-39.0.1 [341 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 ipxe-qemu amd64 1.21.1+git20220113.fbbdc3926~Ubuntu4 [1,569 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirt-daemon-all 1.0.0+git20150424.a25a1dd~Ubuntu4 [552 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirt1 amd64 0.3.12-13build1 [7,176 B]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirtx1 amd64 1:2.8.0-2build2 [38.0 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirtx2 amd64 1:2.8.0-2build2 [38.0 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirtx3 amd64 1:2.8.0-2build2 [38.0 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libde265 amd64 0.1.0-3build1 [15.1 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgfpr0 amd64 10.1~ubuntu0.2 [22.3 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libglusterfs amd64 10.1~ubuntu0.2 [287 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgfpr0c0 amd64 10.1~ubuntu0.2 [47.1 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgfpr0c1 amd64 10.1~ubuntu0.2 [77.3 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liblircclient1 amd64 1.19.0-3build1 [78.6 kB]
Get:17 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libiscsi17 amd64 1.19.0-3build12 [57.7 kB]
Get:18 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liblpmemobj1 amd64 1.11.1-3build1 [124 kB]
Get:19 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liblpmemobj1 amd64 1.11.1-3build12 [124 kB]
Get:20 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 liblpmemobj1 amd64 1.11.1-3build12 [124 kB]
Get:21 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 liblpmemobj1 amd64 1.11.1-3build12 [124 kB]
Get:22 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 liblpmemobj1 amd64 1.11.1-3build12 [124 kB]
Get:23 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 liblpmemobj1 amd64 1.11.1-3build12 [124 kB]
Get:24 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liblpmemobj1 amd64 1.11.1-3build12 [124 kB]
Get:25 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libspice-server1 amd64 0.15.0~ubuntu4 [351 kB]
Get:26 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirt2 amd64 2.1-2build1 [10.3 kB]
Get:27 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirt2parser1 amd64 2.1-2build1 [15.9 kB]
Get:28 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libvirtglrenderer1 amd64 0.3.1-1-experimental [188 kB]
Get:29 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libfdt1 amd64 1.6.1-1 [20.0 kB]
Get:30 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 qemu-system-common amd64 1:6.2+dfsg~ubuntu6.16 [2,048 kB]
Get:31 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 qemu-system-data all 1:6.2+dfsg~ubuntu6.16 [1,431 kB]
Get:32 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 seabios all 1.15.0-1 [174 kB]
Get:33 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 qemu-system-x86 amd64 1:6.2+dfsg~ubuntu6.16 [10.1 MB]
```

2. Download the Ubuntu Server 20.04.6 Server ISO file using the below command.

```
sourabhdeshmukh$ wget
https://releases.ubuntu.com/20.04.6/ubuntu-20.04.6-live-server-amd64.iso
```



```
sourabhdeshmukh$ wget https://releases.ubuntu.com/20.04.6/ubuntu-20.04.6-live-server-amd64.iso
--2024-01-23 00:00:41-- https://releases.ubuntu.com/20.04.6/ubuntu-20.04.6-live-server-amd64.iso
Resolving releases.ubuntu.com (releases.ubuntu.com)... 2620:2d:4000:1::17, 2620:2d:4000:1::1a, 2001:67c:1562::25, ...
Connecting to releases.ubuntu.com (releases.ubuntu.com)|2620:2d:4000:1::17|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1487339520 (1.4G) [application/x-isos9660-image]
Saving to: 'ubuntu-20.04.6-live-server-amd64.iso'

ubuntu-20.04.6-live-server-amd64.iso 100%[=====] 1.38G 4.21MB/s in 2m 13s

2024-01-23 00:02:54 (10.7 MB/s) - "ubuntu-20.04.6-live-server-amd64.iso" saved [1487339520/1487339520]
```

## QEMU Configuration Details

When exploring the complexities of QEMU, launching an Ubuntu OS from a previously generated `ubuntu.img` file becomes a versatile task with a number of customizable options. These options extend to defining memory, CPU characteristics, networking parameters, block device configurations, and the utilization of accelerators. Below are specific examples demonstrating the launch of an Ubuntu VM with various extra configurations

1. Accelerating with `-accel` Option: Utilizing the `-accel` option allows for performance enhancements through paravirtualized hypervisors. In the given command, the KVM accelerator is employed:

```
sourabhdeshmukh$ sudo qemu-system-x86_64 -hda ubuntu1.qcow2 -accel kvm -boot c -m 2048
```

2. Configuring Cores and CPU Type with `-smp` and `-cpu` Options: The `-smp` option is employed to specify the number of cores the Guest OS can utilize, while the `-cpu` option enables the selection of supported CPUs by QEMU. This example assigns 2 cores and emulates the 'host' CPU type:

```
sourabhdeshmukh$ sudo qemu-system-x86_64 -hda ubuntu1.qcow2 -accel kvm -boot c -m 2048 -smp 2 -cpu host
```

3. Drive Configuration with `-drive` Option: The `-drive` option is used to set the drive for the guest OS. The example below designates the `ubuntu.img` drive with the `virtio` interface:

```
sourabhdeshmukh$ sudo qemu-system-x86_64 -drive file=ubuntu1.img,if=virtio -accel kvm -boot c -m 2048 -smp 2 -cpu host
```

4. User Mode Host Network Backend with `-netdev user` Option: Employing the `-netdev user` option configures the user mode host network backend, requiring no administrator privileges to run. The command below utilizes the `-netdev user` configuration:

```
$ sudo qemu-system-x86_64 -drive file=ubuntu1.img,if=virtio -accel kvm -boot c -m 2048 -smp 2 -cpu host -netdev user,id=net0
```

Sourabh Deshmukh

W1648445

5. Advanced Command with Combined Options: The following advanced command integrates various options to launch the guest operating system with specific configurations:

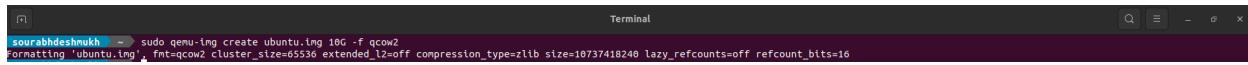
```
$ sudo qemu-system-x86_64 -accel kvm -cpu host -m 2048 -smp 2 -hda ubuntu.img -boot c -device virtio-net,netdev=vm -netdev user,id=vm,hostfwd=tcp:127.0.0.1:9001-:22
```

## QEMU VM SETUP with qcow2, raw and using luks encryption format.

### Qcow2 -

- Generate a QEMU image of Ubuntu in the qcow2 file format using the following command.

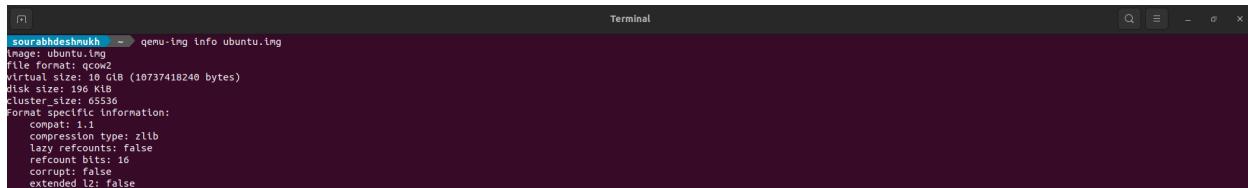
```
sourabhdeshmukh$ sudo qemu-img create ubuntu.img 10G -f qcow2
```



```
sourabhdeshmukh$ sudo qemu-img create ubuntu.img 10G -f qcow2
Formatting '/ubuntu.img' +, fnt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=10737418240 lazy_refcounts=off refcount_bits=16
```

- Information of the generated ubuntu.img qcow2 format file.

```
sourabhdeshmukh$ sudo qemu-img info ubuntu.img
```



```
sourabhdeshmukh$ qemu-img info ubuntu.img
Image: ubuntu.img
file format: qcow2
virtual size: 10 GiB (10737418240 bytes)
disk size: 196 KiB
cluster_size: 65536
Format specific information:
  compat: 1
  compression type: zlib
  lazy_refcounts: false
  refcount_bits: 16
  corrupt: false
  extended_l2: false
```

- Creating a Virtual Machine using the ubuntu.img file.
  - Since we have Downloaded the iso file for the Ubuntu server. We will install the Ubuntu VM using the iso file in the ubuntu.img file. We will use the below command to install the VM.

```
sourabhdeshmukh$ sudo qemu-system-x86_64 -hda ubuntu.img -boot d -cdrom ./ubuntu-20.04.6-live-server-amd64.iso -m 2046 -boot strict=on
```

Description of the flags used in the above command.

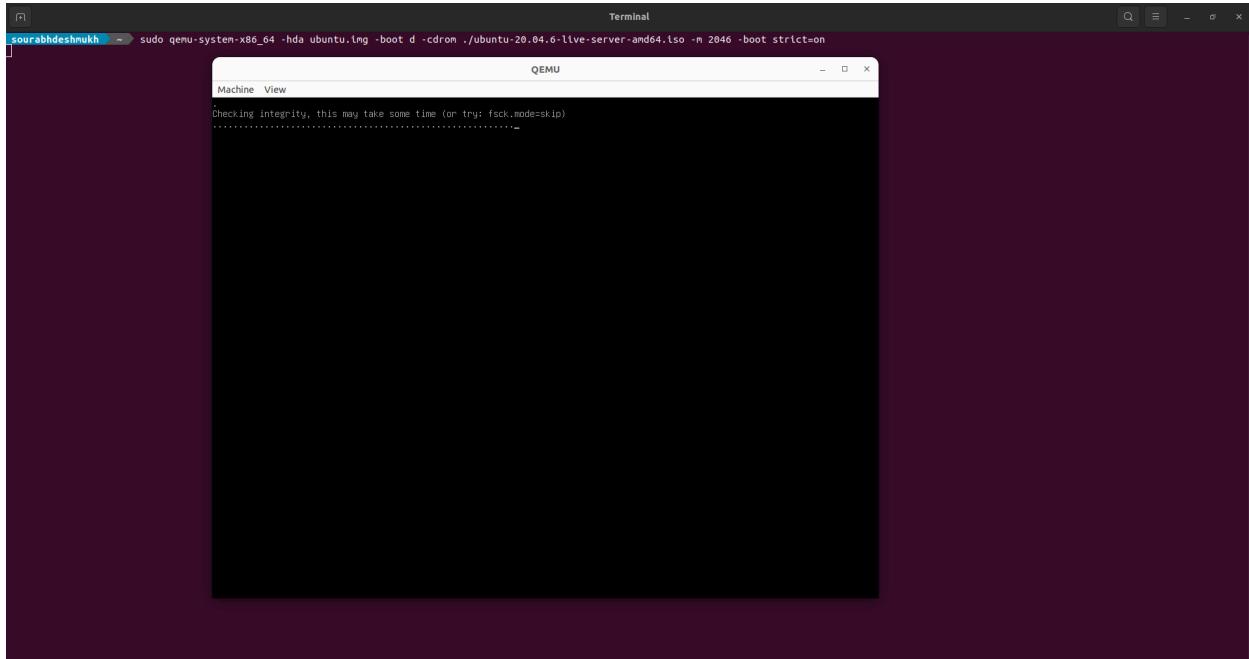
- **-hda** Specifies the file to be used as a hard disk.
- **-boot** Defines the boot order for x86 architecture, where drive letters represent different boot options. For example, a, b (for floppy drives), c (first hard disk), d (first CD-ROM),

## Sourabh Deshmukh

W1648445

n-p (Etherboot from network adapter 1-4). The default option is the hard disk boot.

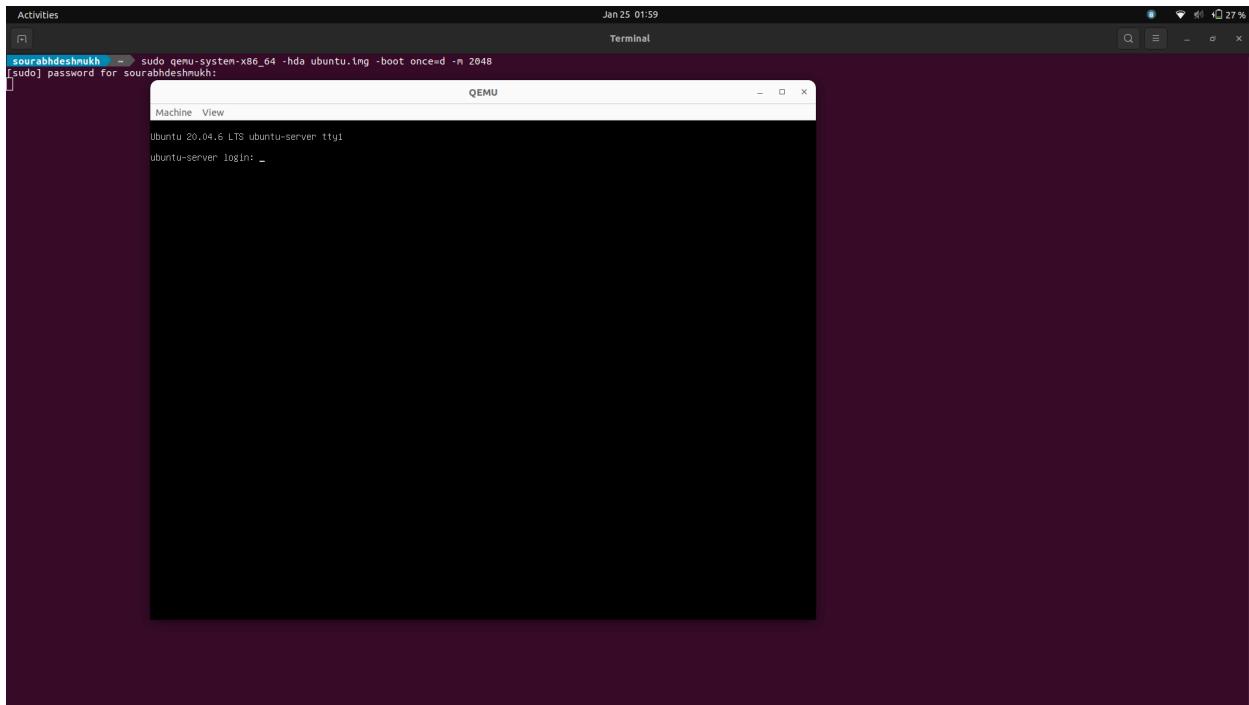
- **-cdrom** Indicates the .iso file to be utilized as the base for the image being created.
- **-m** Sets the startup RAM for the guest OS to the specified value, in this case, 2048 MB.



- Starting the Virtual machine using the ubuntu.img file.

```
sourabhdeshmukh$ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048
```

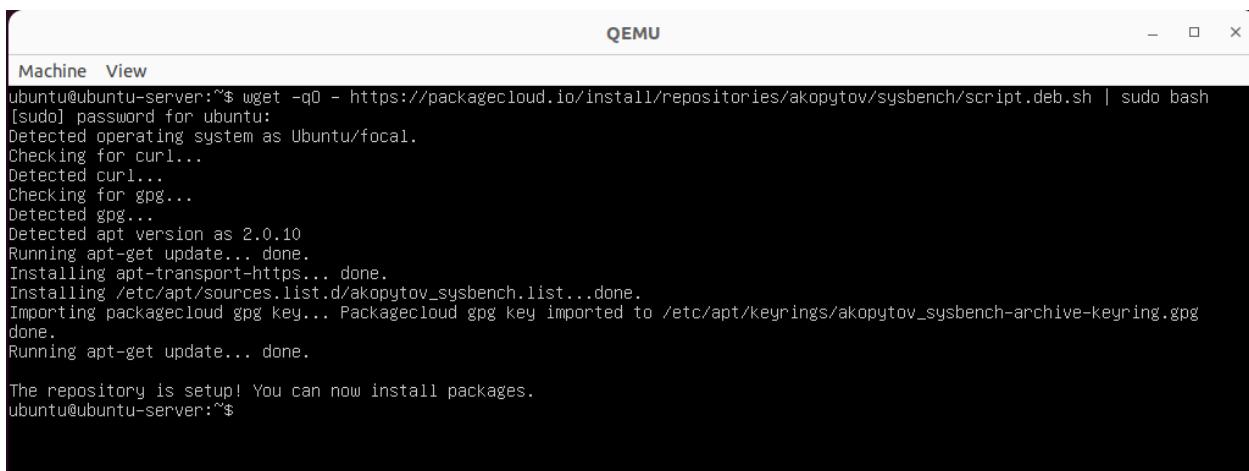
**Sourabh Deshmukh**  
**W1648445**



## Sysbench Setup on Qcow2 VM

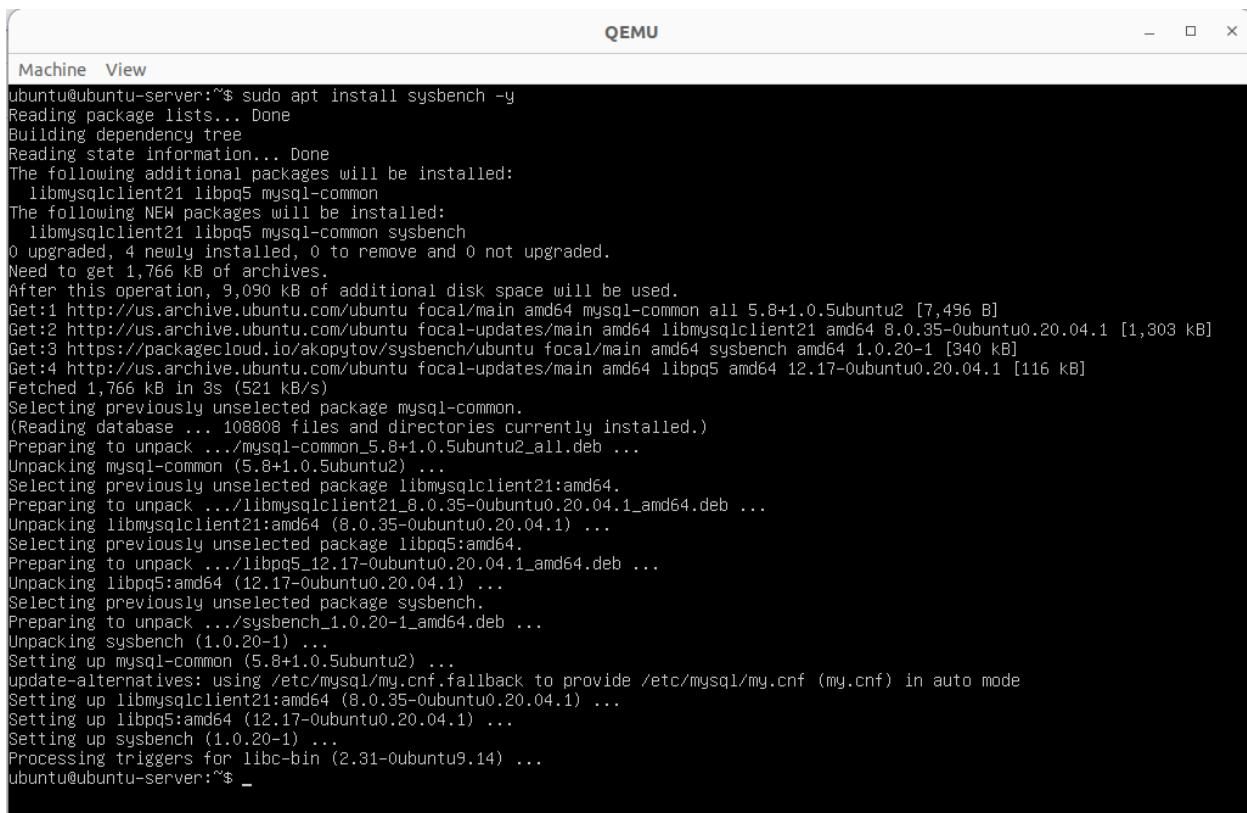
- We will download the repository to install the sysbench using the below command.

```
qcow2$ wget -qO -  
https://packagecloud.io/install/repositories/akopytov/sysbench/script.deb.s  
h | sudo bash
```



- Install the sysbench using below command.

```
qcow2$ sudo apt install sysbench -y
```

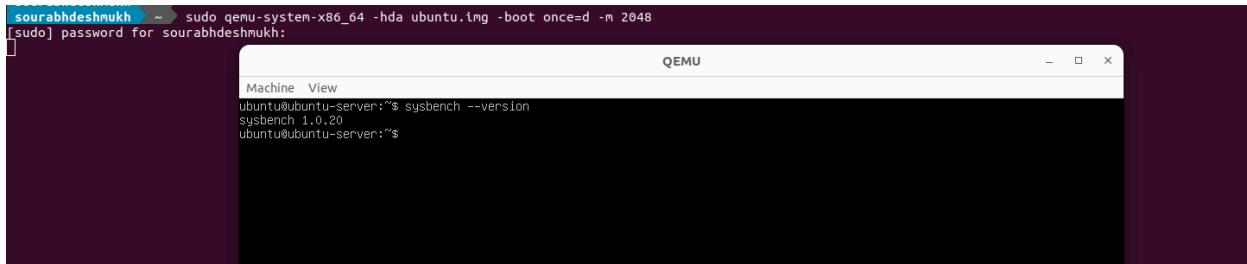


The screenshot shows a terminal window titled "QEMU" with the title bar "Machine View". The terminal output is as follows:

```
ubuntu@ubuntu-server:~$ sudo apt install sysbench -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libmysqlclient21 libpq5 mysql-common
The following NEW packages will be installed:
  libmysqlclient21 libpq5 mysql-common sysbench
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,766 kB of archives.
After this operation, 9,090 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 mysql-common all 5.8+1.0.5ubuntu2 [7,496 B]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libmysqlclient21 amd64 8.0.35-0ubuntu0.20.04.1 [1,303 kB]
Get:3 https://packagecloud.io/akopytov/sysbench/ubuntu focal/main amd64 sysbench amd64 1.0.20-1 [340 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpq5 amd64 12.17-0ubuntu0.20.04.1 [116 kB]
Fetched 1,766 kB in 3s (521 kB/s)
Selecting previously unselected package mysql-common.
(Reading database ... 108808 files and directories currently installed.)
Preparing to unpack .../mysql-common_5.8+1.0.5ubuntu2_all.deb ...
Unpacking mysql-common (5.8+1.0.5ubuntu2) ...
Selecting previously unselected package libmysqlclient21:amd64.
Preparing to unpack .../libmysqlclient21_8.0.35-0ubuntu0.20.04.1_amd64.deb ...
Unpacking libmysqlclient21:amd64 (8.0.35-0ubuntu0.20.04.1) ...
Selecting previously unselected package libpq5:amd64.
Preparing to unpack .../libpq5_12.17-0ubuntu0.20.04.1_amd64.deb ...
Unpacking libpq5:amd64 (12.17-0ubuntu0.20.04.1) ...
Selecting previously unselected package sysbench.
Preparing to unpack .../sysbench_1.0.20-1_amd64.deb ...
Unpacking sysbench (1.0.20-1) ...
Setting up mysql-common (5.8+1.0.5ubuntu2) ...
update-alternatives: using /etc/mysql/my.cnf.fallback to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Setting up libmysqlclient21:amd64 (8.0.35-0ubuntu0.20.04.1) ...
Setting up libpq5:amd64 (12.17-0ubuntu0.20.04.1) ...
Setting up sysbench (1.0.20-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.14) ...
ubuntu@ubuntu-server:~$ _
```

- Checking the installation of sysbench on the Virtual Machine.

```
qcow2$ sysbench --version
```



The screenshot shows a terminal window titled "QEMU" with the title bar "Machine View". The terminal output is as follows:

```
sourabhdeshmukh ~ $ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048
[sudo] password for sourabhdeshmukh:
ubuntu@ubuntu-server:~$ sysbench --version
sysbench 1.0.20
ubuntu@ubuntu-server:~$
```

## RAW -

- Generate a QEMU image of Ubuntu in the raw file format using the following command.

```
sourabhdeshmukh$ sudo qemu-img create Ubuntu.img 10G -f raw
```

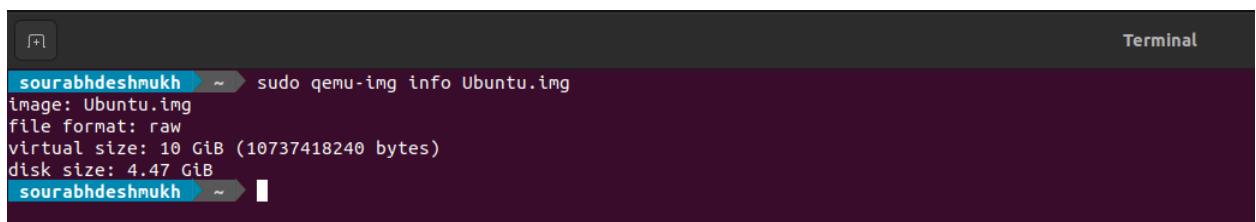


A terminal window titled "Terminal" showing the command "sudo qemu-img create -f raw Ubuntu.img 10G". The output indicates that the image is being formatted and its size is 10737418240 bytes.

```
sourabhdeshmukh ~ sudo qemu-img create -f raw Ubuntu.img 10G
Formatting 'Ubuntu.img', fmt=raw size=10737418240
sourabhdeshmukh ~
```

- Information of the generated ubuntu.img qcow2 format file.

```
sourabhdeshmukh$ sudo qemu-img info Ubuntu.img
```



A terminal window titled "Terminal" showing the command "sudo qemu-img info Ubuntu.img". The output provides details about the image, including its name, file format, virtual size, and disk size.

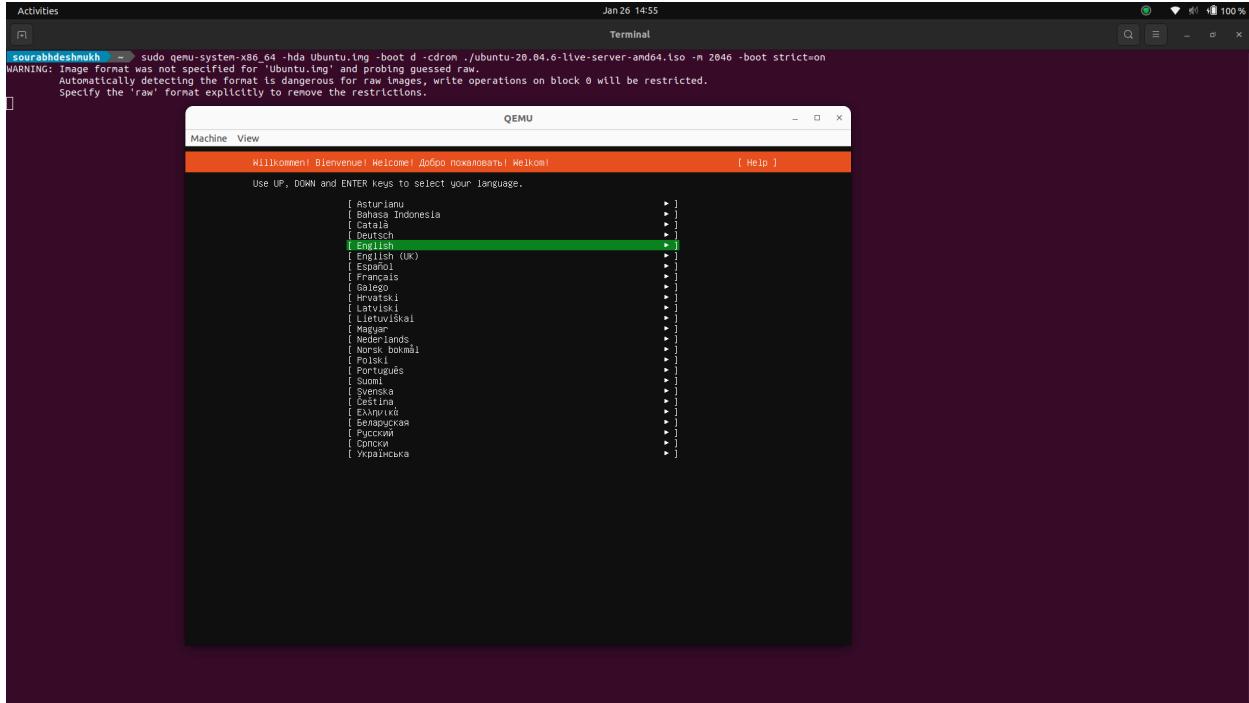
```
sourabhdeshmukh ~ sudo qemu-img info Ubuntu.img
image: Ubuntu.img
file format: raw
virtual size: 10 GiB (10737418240 bytes)
disk size: 4.47 GiB
sourabhdeshmukh ~
```

- Creating Virtual Machine using the ubuntu.img file.
  - Since we have Downloaded the iso file for Ubuntu server. We will install the Ubuntu VM using the iso file in the ubuntu.img file. We will use the below command to install the VM.

```
sourabhdeshmukh$ sudo qemu-system-x86_64 -hda Ubuntu.img -boot d -cdrom
./ubuntu-20.04.6-live-server-amd64.iso -m 2046 -boot strict=on
```

# Sourabh Deshmukh

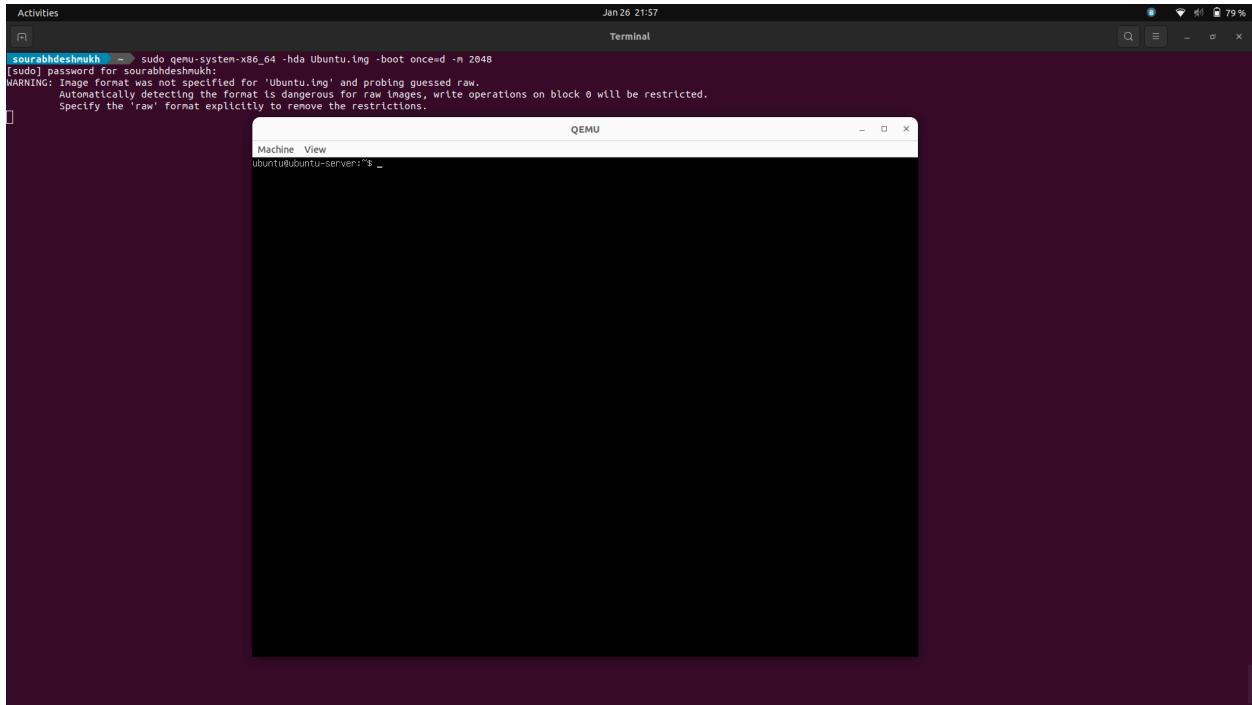
## W1648445



Description of the flags used in the above command.

- **-hda** Specifies the file to be used as a hard disk.
  - **-boot** Defines the boot order for x86 architecture, where drive letters represent different boot options. For example, a, b (for floppy drives), c (first hard disk), d (first CD-ROM), n-p (Etherboot from network adapter 1-4). The default option is the hard disk boot.
  - **-cdrom** Indicates the .iso file to be utilized as the base for the image being created.
  - **-m** Sets the startup RAM for the guest OS to the specified value, in this case, 2048 MB.
- 
- Start the virtual machine using the Ubuntu.img file.

```
$ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048
```



## Sysbench Setup on Raw VM

- We will download the repository to install the sysbench using the below command.

```
raw$ wget -qO -
https://packagecloud.io/install/repositories/akopytov/sysbench/script.deb.s
h | sudo bash
```

- Install the sysbench using below command.

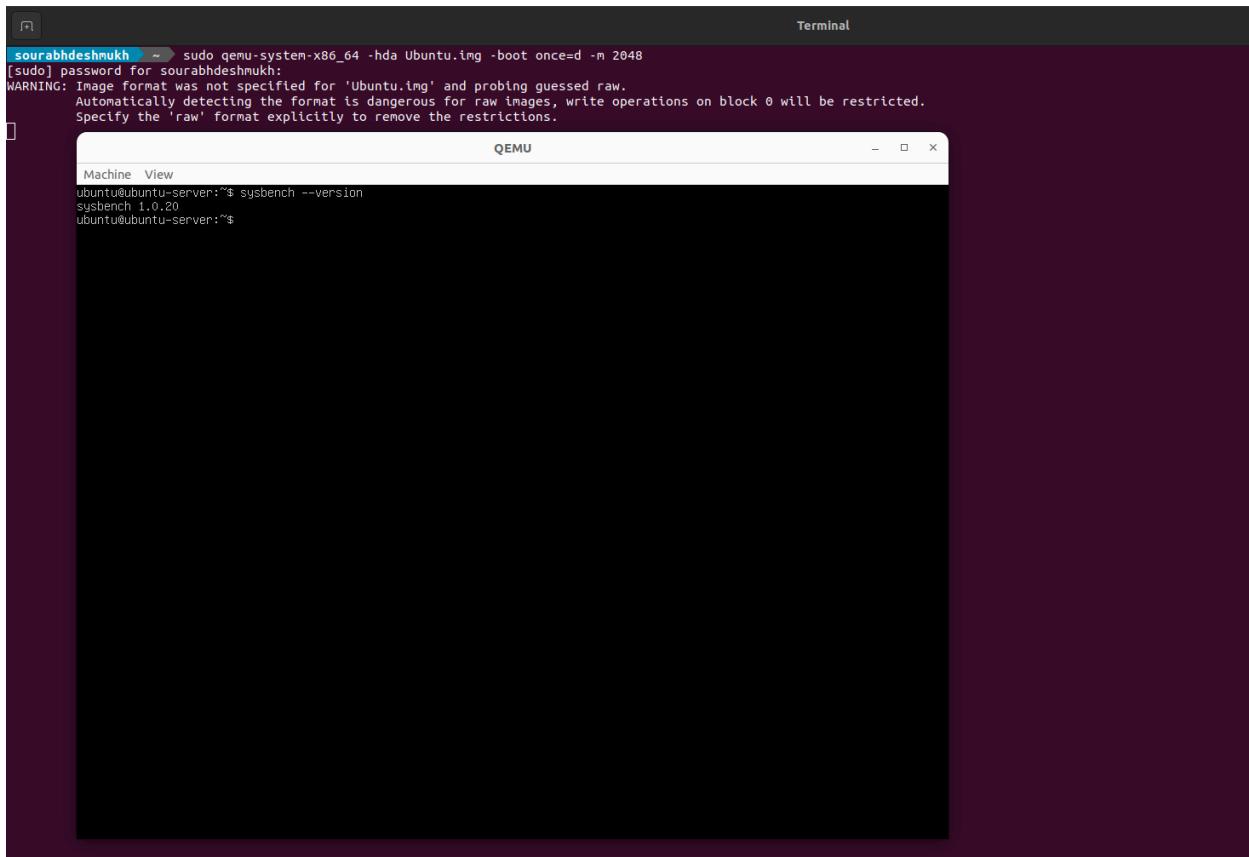
```
raw$ sudo apt install sysbench -y
```

- Checking the installation of sysbench on the Virtual Machine.

```
raw$ sysbench --version
```

## Sourabh Deshmukh

W1648445



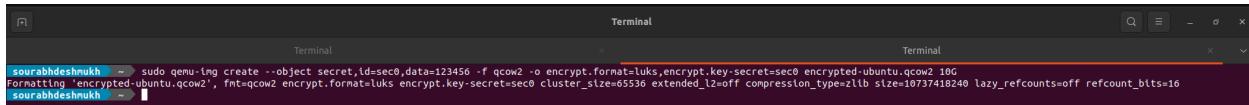
Sourabh Deshmukh

W1648445

## LUKS Encryption -

- Generate an encrypted QEMU image of Ubuntu in the qcow2 file format using the following command with luks encryption format.

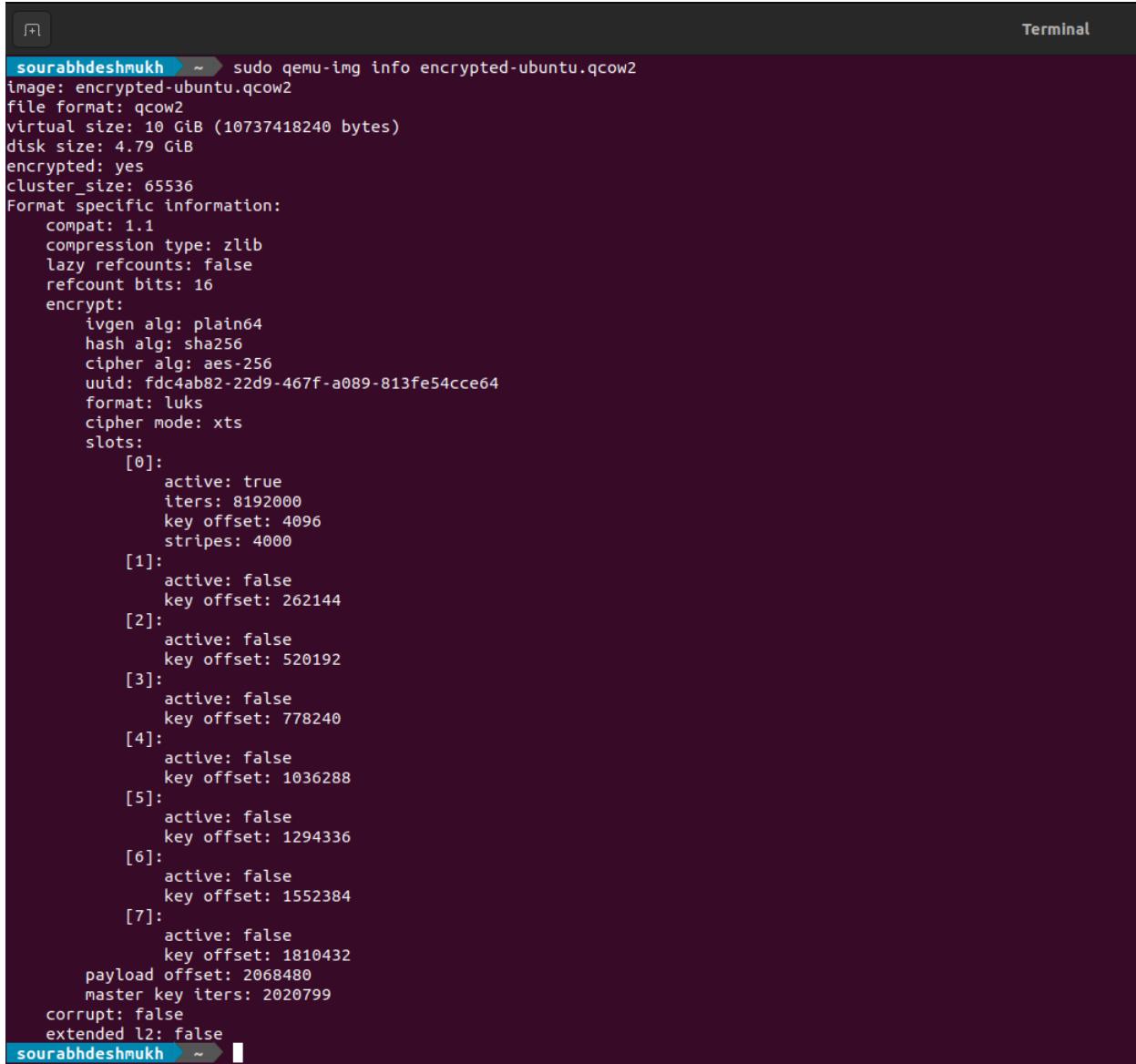
```
sourabhdeshmukh$ sudo qemu-img create --object secret,id=sec0,data=123456  
-f qcow2 -o encrypt.format=luks,encrypt.key-secret=sec0  
encrypted-ubuntu.qcow2 10G
```



Description of the new flags used in the above command.

- --object secret,id=sec0,data=123456: This part creates a QEMU secret object named sec0 with a data value of 123456. Secrets in QEMU can be used for various purposes, including encryption keys.
- -o encrypt.format=luks,encrypt.key-secret=sec0: It indicates that the encryption format is LUKS (Linux Unified Key Setup), and it associates the secret object sec0 as the key for encryption.
- Information on the generated ubuntu.img qcow2 format file.

```
sourabhdeshmukh$ sudo qemu-img info encrypted-ubuntu.qcow2
```



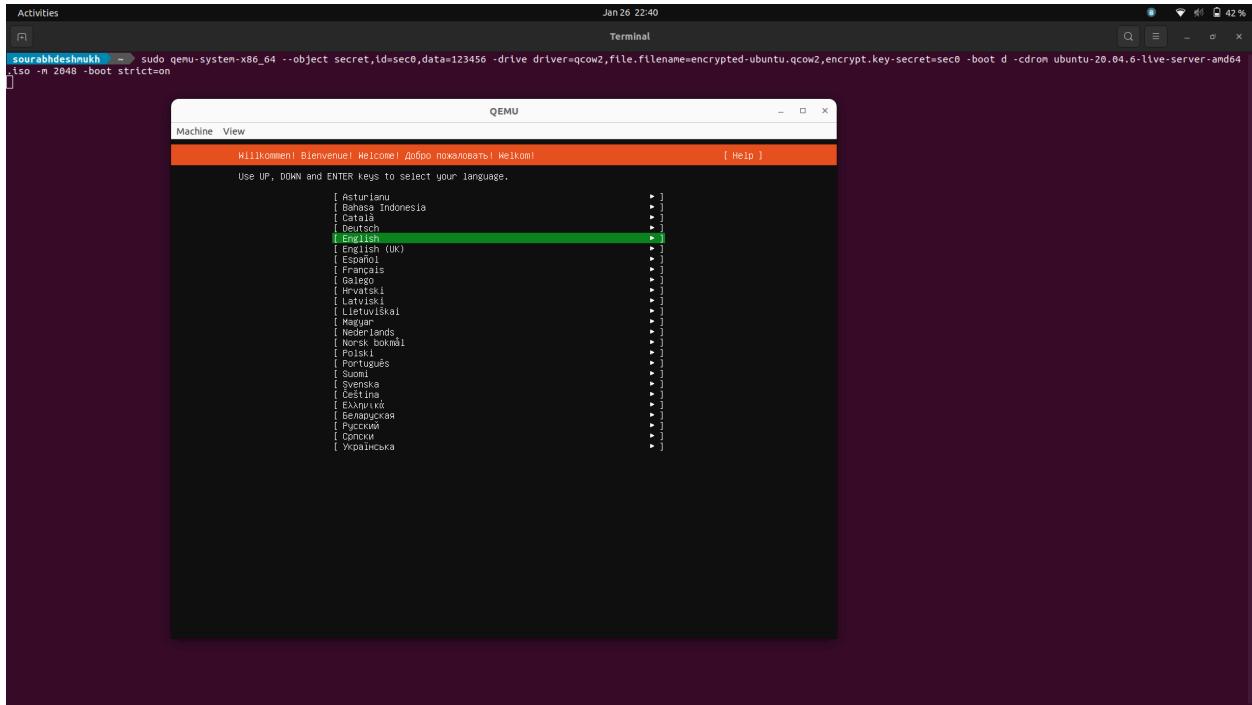
```
sourabhdeshmukh ~ ➔ sudo qemu-img info encrypted-ubuntu.qcow2
image: encrypted-ubuntu.qcow2
file format: qcow2
virtual size: 10 GiB (10737418240 bytes)
disk size: 4.79 GiB
encrypted: yes
cluster_size: 65536
Format specific information:
  compat: 1.1
  compression type: zlib
  lazy refcounts: false
  refcount bits: 16
  encrypt:
    ivgen alg: plain64
    hash alg: sha256
    cipher alg: aes-256
    uuid: fdc4ab82-22d9-467f-a089-813fe54cce64
    format: luks
    cipher mode: xts
  slots:
    [0]:
      active: true
      iters: 8192000
      key offset: 4096
      stripes: 4000
    [1]:
      active: false
      key offset: 262144
    [2]:
      active: false
      key offset: 520192
    [3]:
      active: false
      key offset: 778240
    [4]:
      active: false
      key offset: 1036288
    [5]:
      active: false
      key offset: 1294336
    [6]:
      active: false
      key offset: 1552384
    [7]:
      active: false
      key offset: 1810432
  payload offset: 2068480
  master key iters: 2020799
  corrupt: false
  extended l2: false
sourabhdeshmukh ~ ➔
```

- Creating a Virtual Machine using the ubuntu.img file.
  - Since we have Downloaded the iso file for Ubuntu server. We will install the Ubuntu VM using the iso file in the ubuntu.img file. We will use the below command to install the VM.

```
sourabhdeshmukh$ sudo qemu-system-x86_64 --object
secret,id=sec0,data=123456 -drive
driver=qcow2,file.filename=encrypted-ubuntu.qcow2,encrypt.key-secret=sec0
-boot d -cdrom ubuntu-20.04.6-live-server-amd64.iso -m 2048 -boot strict=on
```

# Sourabh Deshmukh

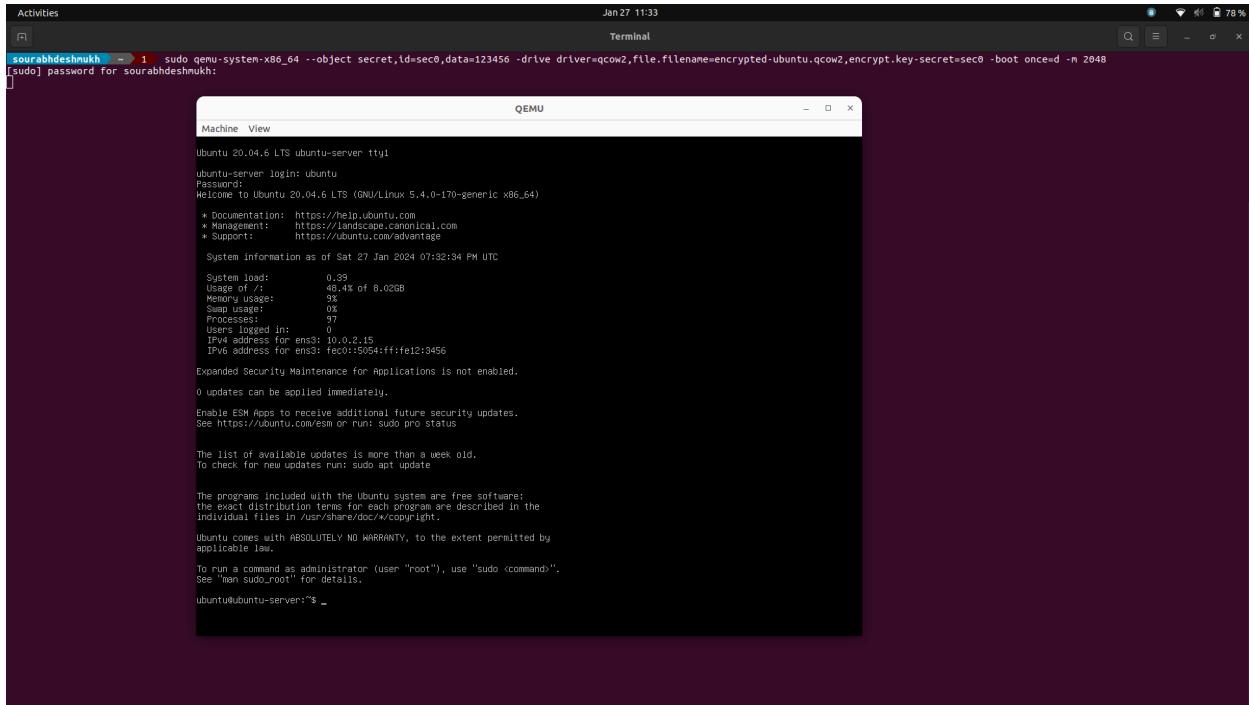
## W1648445



Description of the flags used in the above command.

- **-hda** Specifies the file to be used as a hard disk.
- **-boot** Defines the boot order for x86 architecture, where drive letters represent different boot options. For example, a, b (for floppy drives), c (first hard disk), d (first CD-ROM), n-p (Etherboot from network adapter 1-4). The default option is the hard disk boot.
- **-cdrom** Indicates the .iso file to be utilized as the base for the image being created.
- **-m** Sets the startup RAM for the guest OS to the specified value, in this case, 2048 MB.
- **--object secret,id=sec0,data=123456** This flag passes a QEMU secret object named sec0 with a data value of 123456. This secret can be used for various purposes, including encryption keys.
- **-drive**  
**driver=qcow2,file.filename=encrypted-ubuntu.qcow2,encrypt.key-secret=sec0** This flag configures a virtual drive for the virtual machine. It specifies that the driver is qcow2, sets the filename to encrypted-ubuntu.qcow2, and uses the secret object sec0 as the key for encryption. This indicates that the virtual machine will use an encrypted qcow2 image file.
- Start the virtual machine using the Ubuntu.img file.

```
sourabhdeshmukh$ sudo qemu-system-x86_64 --object secret,id=sec0,data=123456 -drive driver=qcow2,file.filename=encrypted-ubuntu.qcow2,encrypt.key-secret=sec0 -boot once=d -m 2048
```



### • Sysbench Setup on Encrypted VM

- We will download the repository to install the sysbench using the below command.

```
luke-ubuntu$ wget -qO -
https://packagecloud.io/install/repositories/akopytov/sysbench/script.deb.s
h | sudo bash
```

- Install the sysbench using below command.

```
luke-ubuntu$ sudo apt install sysbench -y
```

- Checking the installation of sysbench on the Virtual Machine.

```
luke-ubuntu$ sysbench --version
```

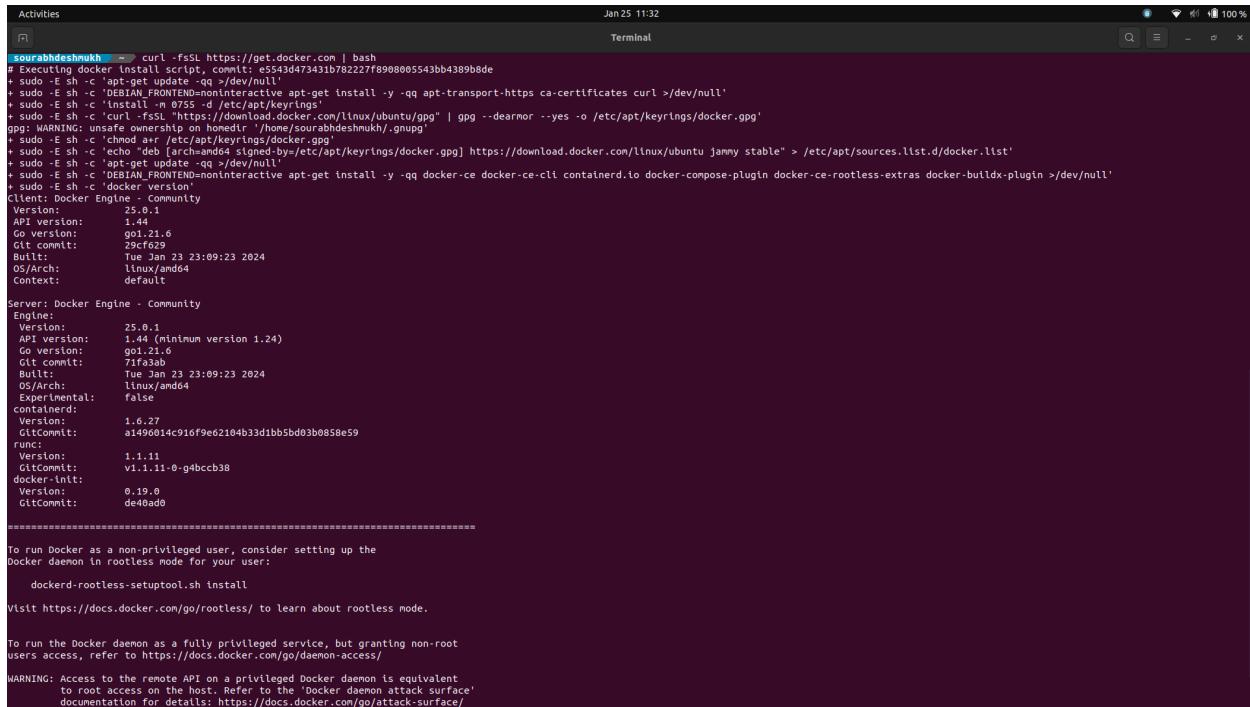
**Sourabh Deshmukh**

**W1648445**

## Docker Installation -

- Install the docker using the below command on the host machine.

```
sourabhdeshmukh$ curl -fsSL https://get.docker.com | bash
```



```
sourabhdeshmukh$ curl -fsSL https://get.docker.com | bash
# Executing docker_install script, commit: 0a94d73431b7822278908005543bb4389b8de
+ sudo -E sh -c 'apt-get update >/dev/null'
+ sudo -E sh -c 'DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null'
+ sudo -E sh -c 'install -m 0755 -d /etc/apt/keyrings'
+ sudo -E sh -c 'curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | gpg --dearmor -yes -o /etc/apt/keyrings/docker.gpg'
gpg: WARNING: unsafe ownership of '/home/sourabhdeshmukh/.gnupg'
+ sudo -E sh -c 'chmod a+r /etc/apt/keyrings/docker.gpg'
+ sudo -E sh -c 'echo "deb [signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy stable" > /etc/apt/sources.list.d/docker.list'
+ sudo -E sh -c 'DEBIAN_FRONTEND=noninteractive apt-get install -y -qq docker-ce docker-ce-cli containerd.io docker-compose-plugin docker-ce-rootless-extras docker-buildx-plugin >/dev/null'
+ sudo -E sh -c 'docker version'
Client: Docker Engine - Community
Version:           25.0.1
API version:      1.44
Go version:       go1.23.6
Git commit:       29cf629
Built:            Tue Jan 23 23:09:23 2024
OS/Arch:          linux/amd64
Context:          default

Server: Docker Engine - Community
Engine:
  Version:          25.0.1
  API version:     1.44 (minimum version 1.24)
  Go version:      go1.23.6
  Git commit:      7f1f63ab
  Built:           Tue Jan 23 23:09:23 2024
  OS/Arch:         linux/amd64
  Experimental:   false
  containerd:
    Version:        1.6.27
    GitCommit:      a1496014c910f9e62104b33d1bb5bd03b0850e59
  runc:
    Version:        1.1.11
    GitCommit:      v1.1.11-0-g4bccb38
  docker-init:
    Version:        0.19.0
    GitCommit:      de40ad0
=====
To run Docker as a non-privileged user, consider setting up the
Docker daemon in rootless mode for your user:
  dockerd-rootless-setuptool.sh install
Visit https://docs.docker.com/go/rootless/ to learn about rootless mode.

To run the Docker daemon as a fully privileged service, but granting non-root
users access, refer to https://docs.docker.com/go/daemon-access/
WARNING: Access to the remote API on a privileged Docker daemon is equivalent
to root access on the host. Refer to the 'Docker daemon attack surface'
documentation for details: https://docs.docker.com/go/attack-surface/
```

- The above command installs the components required for docker and creates a group called docker. Now start the docker service using the below command.

```
sourabhdeshmukh$ sudo systemctl start docker
```

- Check the status of Docker Service using the below Command.

```
sourabhdeshmukh$ sudo systemctl status docker
```

## Sourabh Deshmukh

W1648445

```
sourabhdeshmukh ~ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-01-25 11:32:44 PST; 5min ago
     TriggeredBy: ● docker.socket
Main PID: 7009 (dockerd)
   Tasks: 14
     Memory: 29.8M
        CPU: 2.10us
      CGroup: /system.slice/docker.service
             └─7009 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jan 25 11:32:44 dhruv99-Zephyrus-G-U502DU-G502DU systemd[1]: Starting Docker Application Container Engine...
Jan 25 11:32:44 dhruv99-Zephyrus-G-U502DU-G502DU dockerd[7009]: time="2024-01-25T11:32:44.1317880+08:00" level=info msg="Starting up"
Jan 25 11:32:44 dhruv99-Zephyrus-G-U502DU-G502DU dockerd[7009]: time="2024-01-25T11:32:44.133548660+08:00" level=info msg="Detected 127.0.0.53 nameserver, assuming systemd-resolved, so using resolv.conf: /run/systemd/resolve/resolv.conf"
Jan 25 11:32:44 dhruv99-Zephyrus-G-U502DU-G502DU dockerd[7009]: time="2024-01-25T11:32:44.133548660+08:00" level=info msg="Loading containers: start."
Jan 25 11:32:44 dhruv99-Zephyrus-G-U502DU-G502DU dockerd[7009]: time="2024-01-25T11:32:44.638364045+08:00" level=info msg="Loading containers: done."
Jan 25 11:32:44 dhruv99-Zephyrus-G-U502DU-G502DU dockerd[7009]: time="2024-01-25T11:32:44.723349795+08:00" level=info msg="Docker daemon" commit="7ifa3ab" containerd-snapshotter=false storage-driver=overlay2 v=2.1.10
Jan 25 11:32:44 dhruv99-Zephyrus-G-U502DU-G502DU dockerd[7009]: time="2024-01-25T11:32:44.723566304+08:00" level=info msg="Daemon has completed initialization"
Jan 25 11:32:44 dhruv99-Zephyrus-G-U502DU-G502DU dockerd[7009]: time="2024-01-25T11:32:44.759128108+08:00" level=info msg="API listen on /run/docker.sock"
Jan 25 11:32:44 dhruv99-Zephyrus-G-U502DU-G502DU systemd[1]: Started Docker Application Container Engine.
Lines 1-21/21 (END)
```

- Checking the Docker installation by running hello-world container.

```
sourabhdeshmukh$ docker run hello-world
```

```
sourabhdeshmukh ~ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31e05944: Pull complete
Digest: sha256:4b0d7811b06914499dbc560e6a20eb57ff6655aaea4a80c50b0c5491968cbc2e6
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

- Now pull the ubuntu docker image for creating sysbench custom image using the below command.

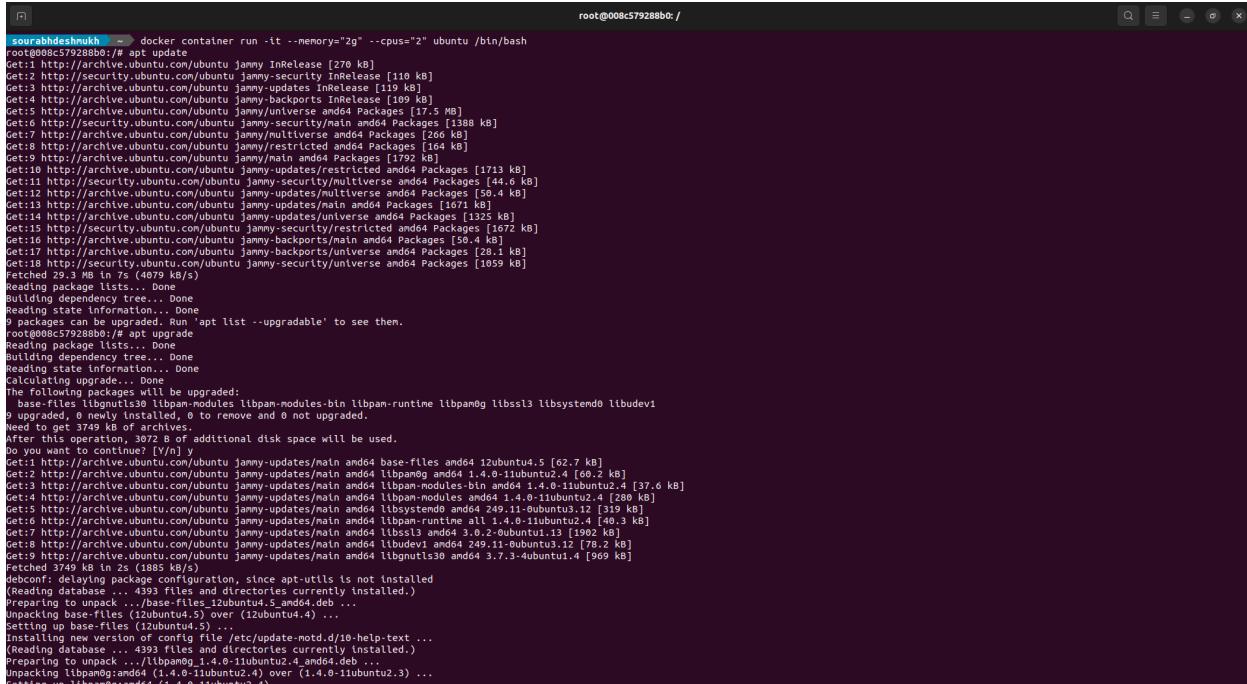
```
sourabhdeshmukh$ docker pull ubuntu:latest
```

```
sourabhdeshmukh ~ docker pull ubuntu:latest
latest: Pulling from library/ubuntu
29020e85b20: Pull complete
Digest: sha256:e01734dd55e76b87c4af8db8821b1feae4146dd47341e4d431118c7dd00a7
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

- Run the ubuntu container using the downloaded base image.

# Sourabh Deshmukh

## W1648445



```
sourabhdeshmukh$ docker container run -it --memory="2g" --cpus="2" ubuntu /bin/bash
root@008c579288b0:/ # apt update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [110 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/amd64 Packages [1388 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-backports/amd64 Packages [172 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1723 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1713 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [44.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [50.4 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1671 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1235 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1672 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [50.4 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [28.1 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1059 kB]
Fetched 29.3 MB in 7s (4079 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@008c579288b0:/ # apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  base-files libgnutls30 libpam-modules-bin libpam-runtime libpam0g libssl3 libsystemd libudev1
9 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 3072 B of additional disk space.
After this operation, 3072 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 base-files amd64 12ubuntu4.5 [62.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpam0g amd64 1.4.0-11ubuntu2.4 [60.2 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpam-modules-bin amd64 1.4.0-11ubuntu2.4 [37.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpam-runtime amd64 1.4.0-11ubuntu2.4 [31.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsystemd amd64 249.11-1ubuntu3.12 [319 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpam-runtime all 1.4.0-11ubuntu2.4 [40.3 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libssl3 amd64 3.0.2-0ubuntu1.13 [1902 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libudev1 amd64 249.11-1ubuntu3.12 [78.2 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgnutls30 amd64 3.7.3-4ubuntu1.4 [969 kB]
Fetched 3072 B in 1s (4079 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
(Reading database... 4393 files and directories currently installed.)
Preparing to unpack .../base-files_12ubuntu4.5_amd64.deb ...
Unpacking base-files (12ubuntu4.5) over (12ubuntu4.4) ...
Setting up base-files (12ubuntu4.5) ...
Installing new version of config file /etc/update-motd.d/10-help-text ...
(Reading database... 4393 files and directories currently installed.)
Preparing to unpack .../libpam0g_1.4.0-11ubuntu2.4_amd64.deb ...
Unpacking libpam0g:amd64 (1.4.0-11ubuntu2.4) over (1.4.0-11ubuntu2.3) ...

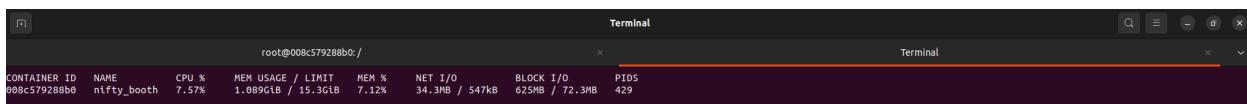
```

- Update the components inside the container using the command below. Since we are using the command inside the container. The container by default runs with the root user, so here we will not be using the sudo.

```
008c579288b0$ apt update
```

- Check the Memory and CPU stats of the Ubuntu container using the command below. This command is run on the host.

```
sourabhdeshmukh$ docker stats <ubuntu-container-id>
```



CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
008c579288b0	nifty_booth	7.57%	1.089GB / 15.3GB	7.12%	34.3MB / 547kB	625MB / 72.3MB	429

## Sysbench Setup in the Ubuntu Docker container.

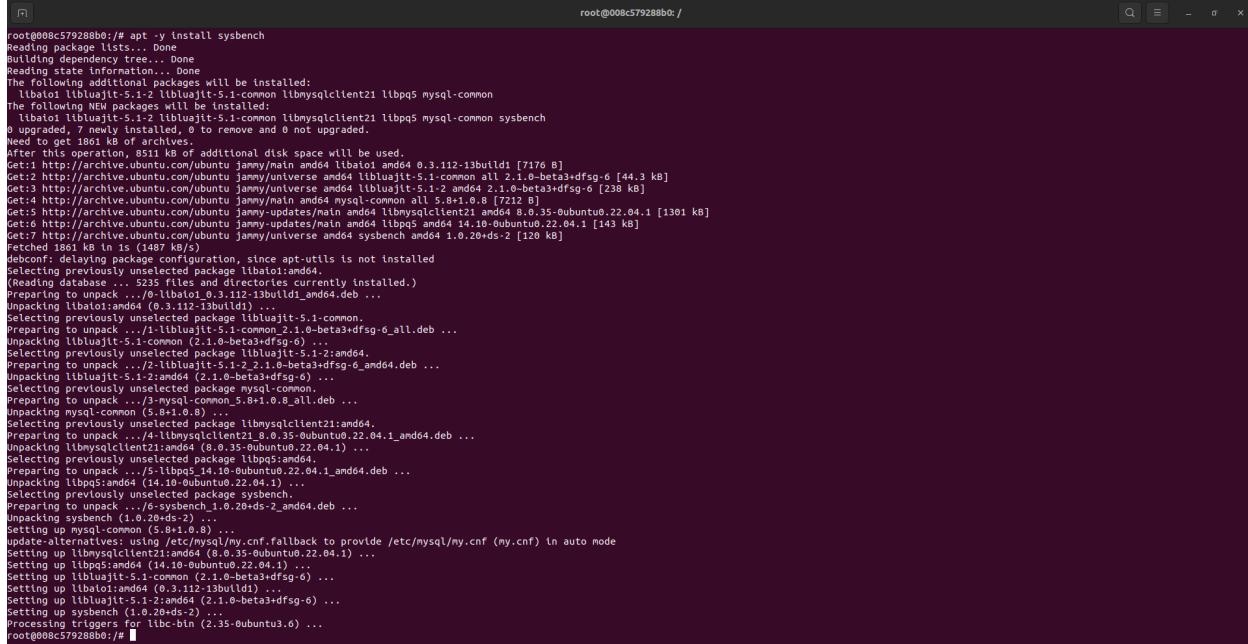
- With the Ubuntu container now running, we can proceed to install sysbench inside the container using the following commands.

```
008c579288b0$ wget -qO -
https://packagecloud.io/install/repositories/akopytov/sysbench/script.deb.s
h | sudo bash
```

## Sourabh Deshmukh

### W1648445

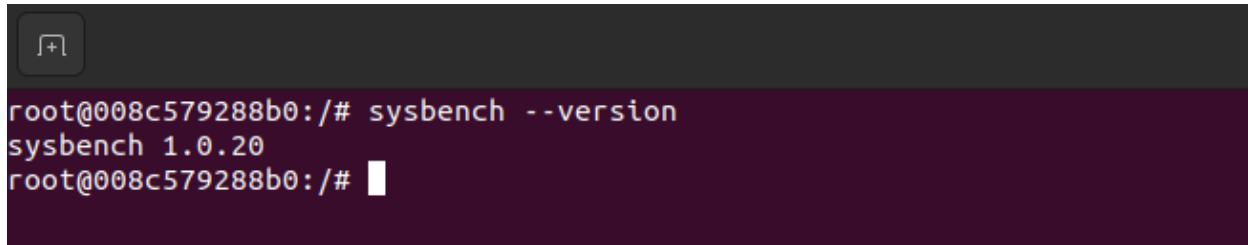
```
008c579288b0$ apt install -y sysbench
```



```
root@008c579288b0:/# apt -y install sysbench
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
libl10n libluajit-5.1-2 libluajit-5.1-common libmysqlclient21 libpq5 mysql-common
The following NEW packages will be installed:
libl10n libluajit-5.1-2 libluajit-5.1-common libmysqlclient21 libpq5 mysql-common sysbench
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 1861 kB of archives.
After this operation, 8511 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 libl10n and64 0.3.112-13build1 [7176 B]
Get:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libluajit-5.1-common all 2.1.0-beta3+dfsg-6 [44.3 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 libmysqlclient21 libpq5 mysql-common sysbench [1301 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7212 B]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libmysqlclient21 libpq5 mysql-common sysbench [1301 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpq5 and64 14.10+0ubuntu0.22.04.1 [143 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/universe amd64 sysbench and64 1.0.20+ds-2 [120 kB]
Fetched 1861 kB in 1s (1487 kB/s)
debconf: delaying configuration of sysbench, since apt-utils is not installed
Selecting previously unselected package libl10n:amd64.
(Reading database ... 5235 files and directories currently installed.)
Preparing to unpack .../0-libl10n 0.3.112-13build1_amd64.deb ...
Unpacking libl10n:amd64 (0.3.112-13build1) ...
Selecting previously unselected package libluajit-5.1-common.
Preparing to unpack .../1-libluajit-5.1-common 2.1.0-beta3+dfsg-6_all.deb ...
Unpacking libluajit-5.1-common (2.1.0-beta3+dfsg-6) ...
Selecting previously unselected package libluajit-5.1-2:amd64.
Preparing to unpack .../2-libluajit-5.1-2_2.1.0-beta3+dfsg-6_amd64.deb ...
Unpacking libluajit-5.1-2:amd64 (2.1.0-beta3+dfsg-6) ...
Selecting previously unselected package mysql-common.
Preparing to unpack .../3-mysql-common_5.8+1.0.8_all.deb ...
Unpacking mysql-common (5.8+1.0.8) ...
Selecting previously unselected package libmysqclient21:amd64.
Preparing to unpack .../4-libmysqclient21_8.0.35-0ubuntu0.22.04.1_amd64.deb ...
Unpacking libmysqclient21:amd64 (8.0.35-0ubuntu0.22.04.1) ...
Selecting previously unselected package libpq5:amd64.
Preparing to unpack .../5-libpq5_14.10+0ubuntu0.22.04.1_amd64.deb ...
Unpacking libpq5:amd64 (14.10+0ubuntu0.22.04.1) ...
Selecting previously unselected package sysbench.
Preparing to unpack .../6-sysbench_1.0.20+ds-2_amd64.deb ...
Unpacking sysbench (1.0.20+ds-2) ...
Setting up mysql-common (5.8+1.0.8)
update-alternatives: using /etc/mysql/my.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Setting up libmysqlclient21:amd64 (8.0.35-0ubuntu0.22.04.1) ...
Setting up libpq5:amd64 (14.10+0ubuntu0.22.04.1) ...
Setting up libluajit-5.1-common (2.1.0-beta3+dfsg-6) ...
Setting up libl10n:amd64 (0.3.112-13build1) ...
Setting up libluajit-5.1-2:amd64 (2.1.0-beta3+dfsg-6) ...
Setting up sysbench (1.0.20+ds-2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
root@008c579288b0:/#
```

- Check the sysbench version installed using the following command inside the Ubuntu container.

```
008c579288b0$ sysbench -version
```



```
root@008c579288b0:/# sysbench --version
sysbench 1.0.20
root@008c579288b0:/#
```

- Now we have configured the sysbench on the Ubuntu container. We will save this progress to our new custom image by using the below command.

```
sourabhdeshmukh$ docker commit 008c579288b0
sourabhdeshmukh/ubuntu-sysbench:v1
```

- We can check the generated image using the following command.

```
sourabhdeshmukh$ docker image ls
```

## Sourabh Deshmukh

W1648445

- If we want to know more about how the docker image is generated then we can use the docker image history command that will show the steps and changes in the creation of an image. It helps track modifications and understand how the image was built.

```
sourabhdeshmukh$ docker image history 77c84a06cdbe
```

```
root@008c579288b0: /  
sourabhdeshmukh ~ ➔ docker commit 008c579288b0 sourabhdeshmukh/ubuntu-sysbench:v1  
sha256:77c84a06cdbe0b2e2c1d78422276021c7b2ef461e362e7eb0e7f773c3c5d6bbd  
sourabhdeshmukh ~ ➔ docker image ls  
REPOSITORY TAG IMAGE ID CREATED SIZE  
sourabhdeshmukh/ubuntu-sysbench v1 77c84a06cdbe 17 seconds ago 158MB  
ubuntu latest e34e831650c1 2 weeks ago 77.9MB  
hello-world latest d2c94e258dcbe 8 months ago 13.3kB  
sourabhdeshmukh ~ ➔ docker image history 77c84a06cdbe  
IMAGE CREATED CREATED BY SIZE COMMENT  
77c84a06cdbe 38 seconds ago /bin/bash 80.5MB  
e34e831650c1 2 weeks ago /bin/sh -c #(nop) CMD ["/bin/bash"] 0B  
<missing> 2 weeks ago /bin/sh -c #(nop) ADD file:c646150c866c8b5ec... 77.9MB  
<missing> 2 weeks ago /bin/sh -c #(nop) LABEL org.opencontainers... 0B  
<missing> 2 weeks ago /bin/sh -c #(nop) LABEL org.opencontainers... 0B  
<missing> 2 weeks ago /bin/sh -c #(nop) ARG LAUNCHPAD_BUILD_ARCH 0B  
<missing> 2 weeks ago /bin/sh -c #(nop) ARG RELEASE 0B  
sourabhdeshmukh ~ ➔
```

- Now we will store this image in the dockerhub repository for later use. For that, we need to have a dockerhub account. To store the image to docker hub, we will perform the docker login and push the custom image using the docker push command

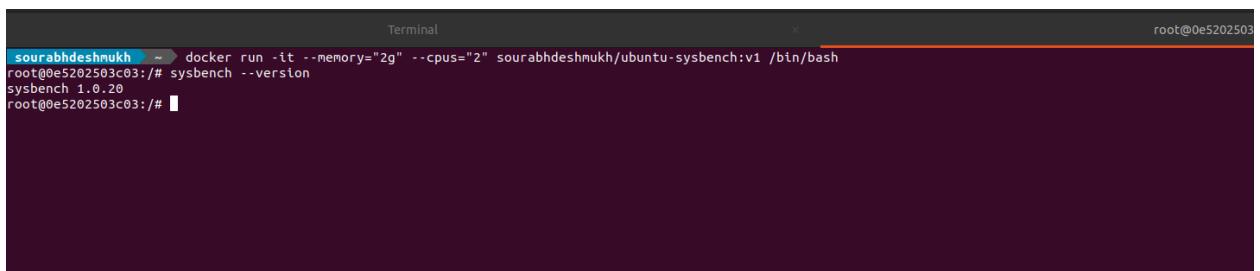
```
sourabhdeshmukh$ docker login
```

```
sourabhdeshmukh$ docker push sourabhdeshmukh/ubuntu-sysbench:v1
```

- Now we will run a container from the newly generated image sourabhdeshmukh/ubuntu-sysbench:v1 and will check whether the sysbench we installed inside the ubuntu container is also present in the newly generated image. We will run the below command to launch the container and then verify the sysbench version inside the container.

```
sourabhdeshmukh$ docker run -it -memory="2g" -cpus="2"  
sourabhdeshmukh/ubuntu-sysbench:v1
```

```
ubuntu-sysbench-container$ sysbench -version
```



A screenshot of a terminal window titled "Terminal". The window has a dark background and light-colored text. At the top right, it shows "root@0e5202503c03:/#". The text in the terminal is as follows:

```
sourabhdeshmukh ~ docker run -it --memory="2g" --cpus="2" sourabhdeshmukh/ubuntu-sysbench:v1 /bin/bash
root@0e5202503c03:/# sysbench --version
sysbench 1.0.20
root@0e5202503c03:/#
```

# SYSBENCH EXPERIMENT ON VMs (qcow2, raw) and Docker

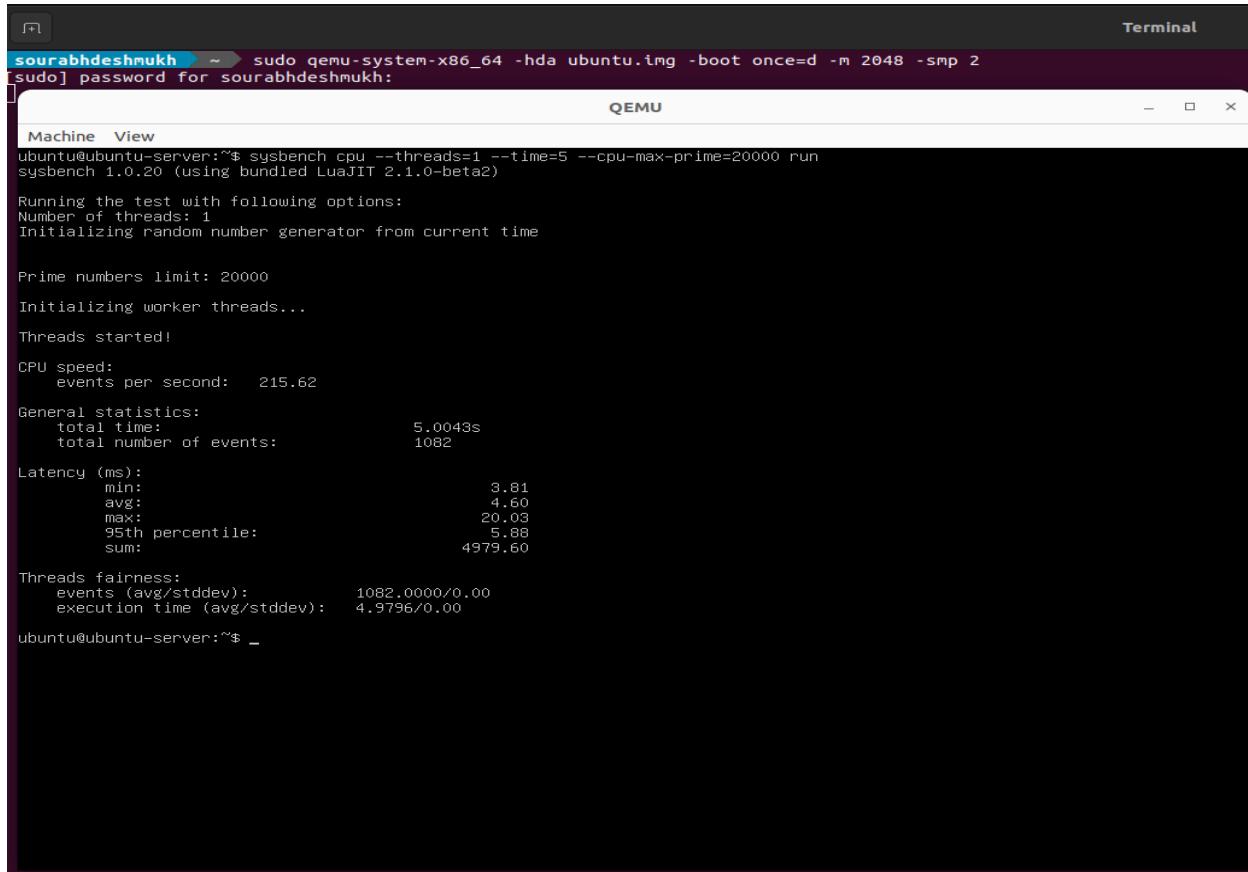
## CPU Test

### CPU Tests on QEMU QCOW2

#### 1. Test cases with 2 CPU and 2 RAM

Test Case 1:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```



The screenshot shows a terminal window titled "Terminal" with a dark background. The command `sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 2` is being entered. A password prompt "[sudo] password for sourabhdeshmukh:" is visible. Below the terminal, a QEMU window titled "QEMU" is open, showing the output of the sysbench CPU test. The test parameters are listed: threads=1, time=5, and cpu-max-prime=20000. The test starts and provides detailed statistics: Prime numbers limit: 20000, CPU speed: events per second: 215.62, General statistics: total time: 5.00043s, total number of events: 1082, Latency (ms): min: 3.81, avg: 4.60, max: 20.03, 95th percentile: 5.88, sum: 4979.60, Threads fairness: events (avg/stddev): 1082.0000/0.00, execution time (avg/stddev): 4.9796/0.00. The test concludes with "ubuntu@ubuntu-server:~\$".

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 2
[sudo] password for sourabhdeshmukh:
[QEMU]
Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 215.62
General statistics:
total time: 5.00043s
total number of events: 1082
Latency (ms):
min: 3.81
avg: 4.60
max: 20.03
95th percentile: 5.88
sum: 4979.60
Threads fairness:
events (avg/stddev): 1082.0000/0.00
execution time (avg/stddev): 4.9796/0.00
ubuntu@ubuntu-server:~$
```

Sourabh Deshmukh

**W1648445**

## Test Case 2:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
sourabhdeshmukh ~ ➜ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 2
[sudo] password for sourabhdeshmukh:
                                          QEMU
Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 200000
Initialzing worker threads...
Threads started!
CPU Speed:
events per second: 9.87

General statistics:
total time: 5.0589s
total number of events: 50

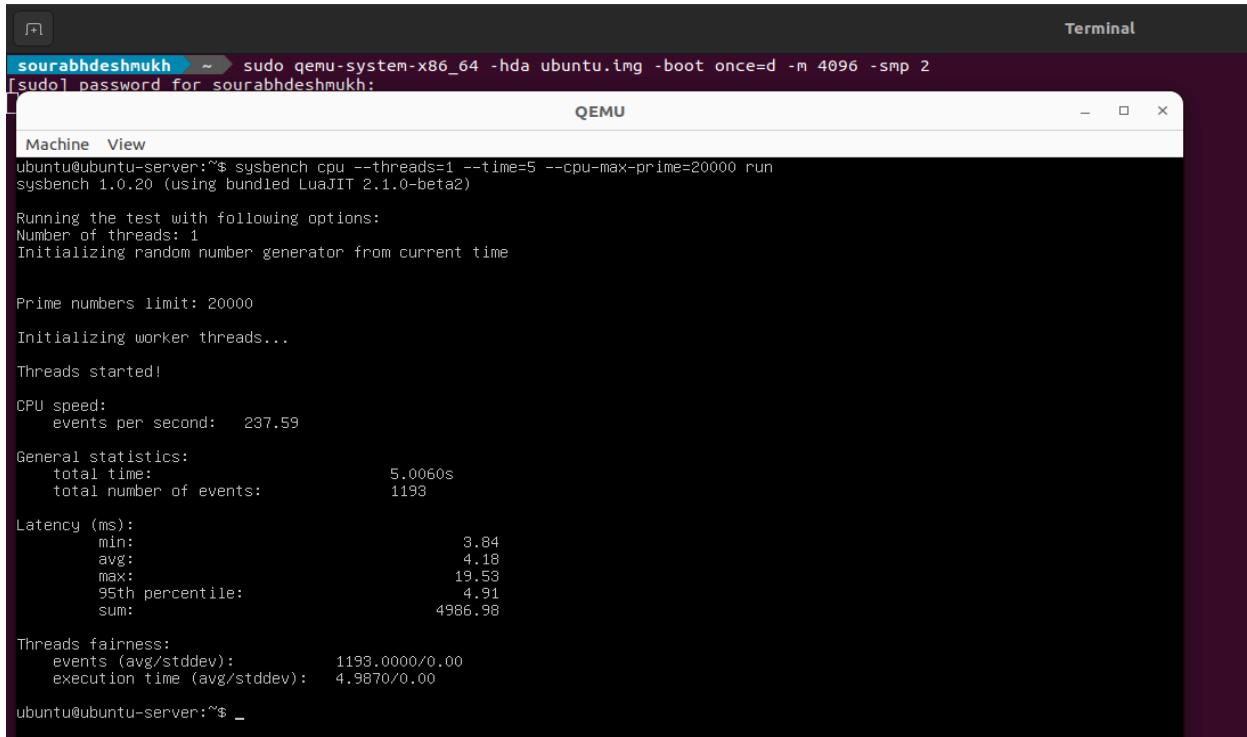
Latency (ms):
min: 88.37
ave: 100.86
max: 141.10
95th percentile: 137.35
sum: 5047.86

Threads fairness:
events (avg/stddev): 50.0000/0.00
execution time (avg/stddev): 5.0479/0.00
ubuntu@ubuntu-server:~$ _
```

## **2. Test cases with 2 CPU and 4 RAM**

## Test Case 1:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```



```
sourabhdeshmukh ~ > sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 2
[sudo] password for sourabhdeshmukh:
[QEMU]
Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000
Initializing worker threads...

Threads started!

CPU speed:
events per second: 237.59

General statistics:
total time: 5.0060s
total number of events: 1193

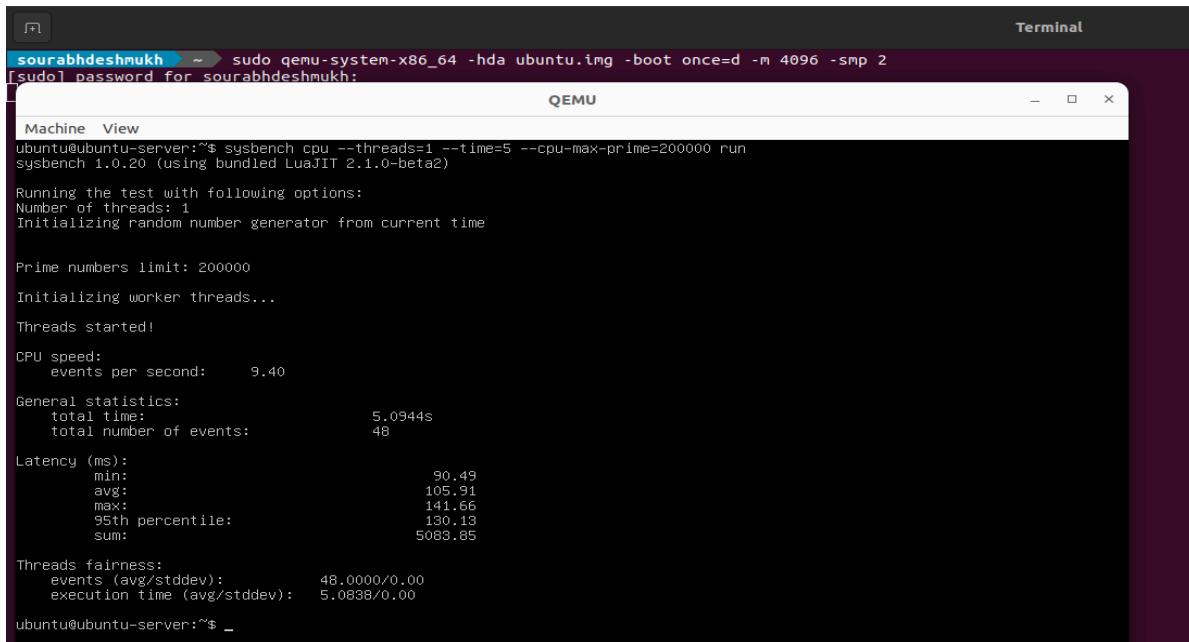
Latency (ms):
min: 3.84
avg: 4.18
max: 19.53
95th percentile: 4.91
sum: 4986.98

Threads fairness:
events (avg/stddev): 1193.0000/0.00
execution time (avg/stddev): 4.9870/0.00

ubuntu@ubuntu-server:~$ _
```

### Test Case 2:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```



```
sourabhdeshmukh ~ > sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 2
[sudo] password for sourabhdeshmukh:
[QEMU]
Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 200000
Initializing worker threads...

Threads started!

CPU speed:
events per second: 9.40

General statistics:
total time: 5.0944s
total number of events: 48

Latency (ms):
min: 90.49
avg: 105.91
max: 141.66
95th percentile: 130.13
sum: 5083.85

Threads fairness:
events (avg/stddev): 48.0000/0.00
execution time (avg/stddev): 5.0838/0.00

ubuntu@ubuntu-server:~$ _
```

## 3. Test cases with 4 CPU and 2 RAM

### Test Case 1:

## Sourabh Deshmukh

W1648445

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 4
[sudo] password for sourabhdeshmukh:

Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 229.99

General statistics:
total time: 5.0062s
total number of events: 1154

Latency (ms):
min: 3.90
avg: 4.32
max: 12.98
95th percentile: 5.09
sum: 4983.45

Threads fairness:
events (avg/stddev): 1154.0000/0.00
execution time (avg/stddev): 4.9834/0.00

ubuntu@ubuntu-server:~$
```

### Test Case 2:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 4
[sudo] password for sourabhdeshmukh:

Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 200000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 8.68

General statistics:
total time: 5.0528s
total number of events: 44

Latency (ms):
min: 90.65
avg: 114.66
max: 161.59
95th percentile: 150.29
sum: 5045.26

Threads fairness:
events (avg/stddev): 44.0000/0.00
execution time (avg/stddev): 5.0453/0.00

ubuntu@ubuntu-server:~$
```

## 4. Test cases with 4 CPU and 4 RAM

### Test Case 1:

## Sourabh Deshmukh

W1648445

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 4
[sudo] password for sourabhdeshmukh:
[QEMU]
Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 233.20

General statistics:
total time: 5.0039s
total number of events: 1168

Latency (ms):
min: 3.82
avg: 4.27
max: 9.82
95th percentile: 5.37
sum: 4984.85

Threads fairness:
events (avg/stddev): 1168.0000/0.00
execution time (avg/stddev): 4.9848/0.00
ubuntu@ubuntu-server:~$ _
```

## Test Case 2:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 4
[sudo] password for sourabhdeshmukh:
[QEMU]
Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 200000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 9.58

General statistics:
total time: 5.0039s
total number of events: 48

Latency (ms):
min: 90.78
avg: 104.07
max: 157.87
95th percentile: 132.49
sum: 4995.58

Threads fairness:
events (avg/stddev): 48.0000/0.00
execution time (avg/stddev): 4.9956/0.00
ubuntu@ubuntu-server:~$ _
```

## CPU Tests on QEMU RAW

### 1. Test case with 2 CPU and 2 RAM

#### Test Case 1:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 2
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 200.30

General statistics:
total time: 5.0081s
total number of events: 1004

Latency (ms):
min: 4.66
avg: 4.97
max: 6.88
95th percentile: 5.77
sum: 4985.62

Threads fairness:
events (avg/stddev): 1004.0000/0.00
execution time (avg/stddev): 4.9856/0.00

ubuntu@ubuntu-server:~$ _
```

#### Test Case 2:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 2
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 200000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 10.46

General statistics:
total time: 5.0583s
total number of events: 53

Latency (ms):
min: 90.02
avg: 95.29
max: 117.19
95th percentile: 112.67
sum: 5050.63

Threads fairness:
events (avg/stddev): 53.0000/0.00
execution time (avg/stddev): 5.0506/0.00

ubuntu@ubuntu-server:~$ _
```

## 2. Test case with 2 CPU and 4 RAM

### Test Case 1:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 2
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
          Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
          Specify the 'raw' format explicitly to remove the restrictions.

Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000
Initializing worker threads...
Threads started!

CPU speed:
  events per second: 235.95

General statistics:
  total time:           5.0061s
  total number of events: 1184

Latency (ms):
  min:                 3.91
  avg:                 4.21
  max:                 8.56
  95th percentile:    4.91
  sum:                4966.33

Threads fairness:
  events (avg/stddev): 1184.0000/0.00
  execution time (avg/stddev): 4.9863/0.00

ubuntu@ubuntu-server:~$ _
```

### Test Case 2:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 2
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
          Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
          Specify the 'raw' format explicitly to remove the restrictions.

Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 200000
Initializing worker threads...
Threads started!

CPU speed:
  events per second: 10.48

General statistics:
  total time:           5.0428s
  total number of events: 53

Latency (ms):
  min:                 88.05
  avg:                 94.93
  max:                 142.84
  95th percentile:    112.67
  sum:                5081.47

Threads fairness:
  events (avg/stddev): 53.0000/0.00
  execution time (avg/stddev): 5.0315/0.00

ubuntu@ubuntu-server:~$ _
```

### 3. Test case with 4 CPU and 2 RAM

#### Test Case 1:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 4
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 238.65

General statistics:
total time: 5.0051s
total number of events: 1198

Latency (ms):
min: 3.88
avg: 4.16
max: 8.70
95th percentile: 5.00
sum: 4988.36

Threads fairness:
events (avg/stddev): 1198.0000/0.00
execution time (avg/stddev): 4.9884/0.00

ubuntu@ubuntu-server:~$
```

#### Test Case 2:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 4
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 200000

Initializing worker threads...

Threads started!

CPU speed:
events per second: 9.31

General statistics:
total time: 5.0345s
total number of events: 47

Latency (ms):
min: 91.15
avg: 106.95
max: 134.23
95th percentile: 130.13
sum: 5026.61

Threads fairness:
events (avg/stddev): 47.0000/0.00
execution time (avg/stddev): 5.0266/0.00

ubuntu@ubuntu-server:~$
```

#### 4. Test case with 4 CPU and 4 RAM

##### Test Case 1:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 4
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View QEMU
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 20000
Initializing worker threads...
Threads started!
CPU speed:
  events per second: 208.94

General statistics:
  total time: 5.0051s
  total number of events: 1047

Latency (ms):
  min: 3.80
  avg: 4.76
  max: 13.20
  95th percentile: 7.56
  sum: 4983.79

Threads fairness:
  events (avg/stddev): 1047.0000/0.00
  execution time (avg/stddev): 4.9838/0.00

ubuntu@ubuntu-server:~$
```

##### Test Case 2:

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 4
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View QEMU
ubuntu@ubuntu-server:~$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

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

Prime numbers limit: 200000
Initializing worker threads...
Threads started!
CPU speed:
  events per second: 10.40

General statistics:
  total time: 5.0001s
  total number of events: 53

Latency (ms):
  min: 90.29
  avg: 95.60
  max: 188.74
  95th percentile: 97.55
  sum: 5066.78

Threads fairness:
  events (avg/stddev): 53.0000/0.00
  execution time (avg/stddev): 5.0668/0.00

ubuntu@ubuntu-server:~$
```

## CPU Tests on Docker Container

### 1. Test case with 2 CPU and 2 RAM

#### Test Case 1:

```
docker$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ docker run -it --memory="2g" --cpus="2" sourabhdeshmukh/ubuntu-sysbench:v1 /bin/bash
root@319cabda9bb0:/# sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 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: 20000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 715.39

General statistics:
total time: 5.0001s
total number of events: 3578

Latency (ms):
min: 1.31
avg: 1.40
max: 5.36
95th percentile: 1.67
sum: 4998.74

Threads fairness:
events (avg/stddev): 3578.0000/0.00
execution time (avg/stddev): 4.9987/0.00
root@319cabda9bb0:/#
```

#### Test Case 2:

```
docker$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
root@319cabda9bb0:/# sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 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: 200000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 29.47

General statistics:
total time: 5.0198s
total number of events: 148

Latency (ms):
min: 33.49
avg: 33.91
max: 68.04
95th percentile: 33.72
sum: 5019.19

Threads fairness:
events (avg/stddev): 148.0000/0.00
execution time (avg/stddev): 5.0192/0.00
root@319cabda9bb0:/#
```

## 2. Test case with 2 CPU and 4 RAM

### Test Case 1:

```
docker$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ docker run -it --memory="4g" --cpus="2" sourabhdeshmukh/ubuntu-sysbench:v1 /bin/bash
root@43649c9038ed:/# sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 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: 20000
Initializing worker threads...
Threads started!

CPU speed:
  events per second: 721.92

General statistics:
  total time:          5.0002s
  total number of events: 3612

Latency (ms):
  min:                 1.34
  avg:                 1.38
  max:                 1.88
  95th percentile:    1.52
  sum:                4998.82

Threads fairness:
  events (avg/stddev): 3612.0000/0.00
  execution time (avg/stddev): 4.9988/0.00
root@43649c9038ed:/#
```

### Test Case 2:

```
docker$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
root@43649c9038ed:/# sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 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: 200000
Initializing worker threads...
Threads started!

CPU speed:
  events per second: 29.46

General statistics:
  total time:          5.0212s
  total number of events: 148

Latency (ms):
  min:                 33.54
  avg:                 33.92
  max:                 68.17
  95th percentile:    33.72
  sum:                5020.49

Threads fairness:
  events (avg/stddev): 148.0000/0.00
  execution time (avg/stddev): 5.0205/0.00
root@43649c9038ed:/#
```

### 3. Test case with 4 CPU and 2 RAM

#### Test Case 1:

```
docker$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ docker run -it --memory="2g" --cpus="4" sourabhdeshmukh/ubuntu-sysbench:v1 /bin/bash
root@d14f158f931f:/# sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 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: 20000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 706.40

General statistics:
total time: 5.0011s
total number of events: 3535

Latency (ms):
min: 1.32
avg: 1.41
max: 4.65
95th percentile: 1.67
sum: 4999.78

Threads fairness:
events (avg/stddev): 3535.0000/0.00
execution time (avg/stddev): 4.9998/0.00
root@d14f158f931f:/#
```

#### Test Case 2:

```
docker$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
root@d14f158f931f:/# sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 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: 200000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 25.72

General statistics:
total time: 5.0134s
total number of events: 129

Latency (ms):
min: 33.62
avg: 38.86
max: 67.82
95th percentile: 41.85
sum: 5012.37

Threads fairness:
events (avg/stddev): 129.0000/0.00
execution time (avg/stddev): 5.0124/0.00
root@d14f158f931f:/#
```

#### 4. Test case with 4 CPU and 4 RAM

##### Test Case 1:

```
docker$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 run
```

```
sourabhdeshmukh ~ ➔ docker run -it --memory="4096g" --cpus="4" sourabhdeshmukh/ubuntu-sysbench:v1 /bin/bash
root@804d2a75d4a0:/# sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000 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: 20000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 740.42

General statistics:
total time: 5.0010s
total number of events: 3705

Latency (ms):
min: 1.31
avg: 1.35
max: 2.31
95th percentile: 1.64
sum: 4999.68

Threads fairness:
events (avg/stddev): 3705.0000/0.00
execution time (avg/stddev): 4.9997/0.00
root@804d2a75d4a0:/# █
```

##### Test Case 2:

```
docker$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 run
```

```
root@804d2a75d4a0:/# sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 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: 200000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 29.40

General statistics:
total time: 5.0314s
total number of events: 148

Latency (ms):
min: 33.62
avg: 33.99
max: 67.51
95th percentile: 33.72
sum: 5030.68

Threads fairness:
events (avg/stddev): 148.0000/0.00
execution time (avg/stddev): 5.0307/0.00
root@804d2a75d4a0:/# █
```

## Memory Test

### Memory Tests on QEMU QCOW2

#### 1. Test case with 2 CPU and 2 RAM

##### Test Case 1:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=1G --memory-scope=global  
--threads=2 --time=10 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 2  
[sudo] password for sourabhdeshmukh:  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 1048576KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 51 ( 4.97 per second)  
52224.00 MiB transferred (5089.19 MiB/sec)  
  
General statistics:  
total time: 10.2515s  
total number of events: 51  
  
Latency (ms):  
min: 276.90  
avg: 401.21  
max: 613.57  
95th percentile: 511.33  
sum: 20461.55  
  
Threads fairness:  
events (avg/stddev): 25.5000/1.50  
execution time (avg/stddev): 10.2308/0.01  
ubuntu@ubuntu-server:~$
```

##### Test Case 2:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=2G --memory-scope=global  
--threads=2 --time=10 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 2
[sudo] password for sourabhdeshmukh:

Machine View
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Running memory speed test with the following options:
block size: 2097152KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 2 ( 0.02 per second)
4096.00 MiB transferred (31.21 MiB/sec)

General statistics:
    total time: 131.2279s
    total number of events: 2

Latency (ms):
    min: 131199.93
    avg: 131200.22
    max: 131200.52
    95th percentile: 100000.00
    sum: 262400.45

Threads fairness:
    events (avg/stddev): 1.0000/0.00
    execution time (avg/stddev): 131.2002/0.00
ubuntu@ubuntu-server:~$ _
```

## 2. Test case with 2 CPU and 4 RAM

### Test Case 1:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=1G --memory-scope=global
--threads=2 --time=10 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 2
[sudo] password for sourabhdeshmukh:

Machine View
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Running memory speed test with the following options:
block size: 1048576KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!
Total operations: 51 ( 4.97 per second)
52224.00 MiB transferred (5087.36 MiB/sec)

General statistics:
    total time: 10.2462s
    total number of events: 51

Latency (ms):
    min: 295.88
    avg: 399.57
    max: 531.35
    95th percentile: 484.44
    sum: 20378.10

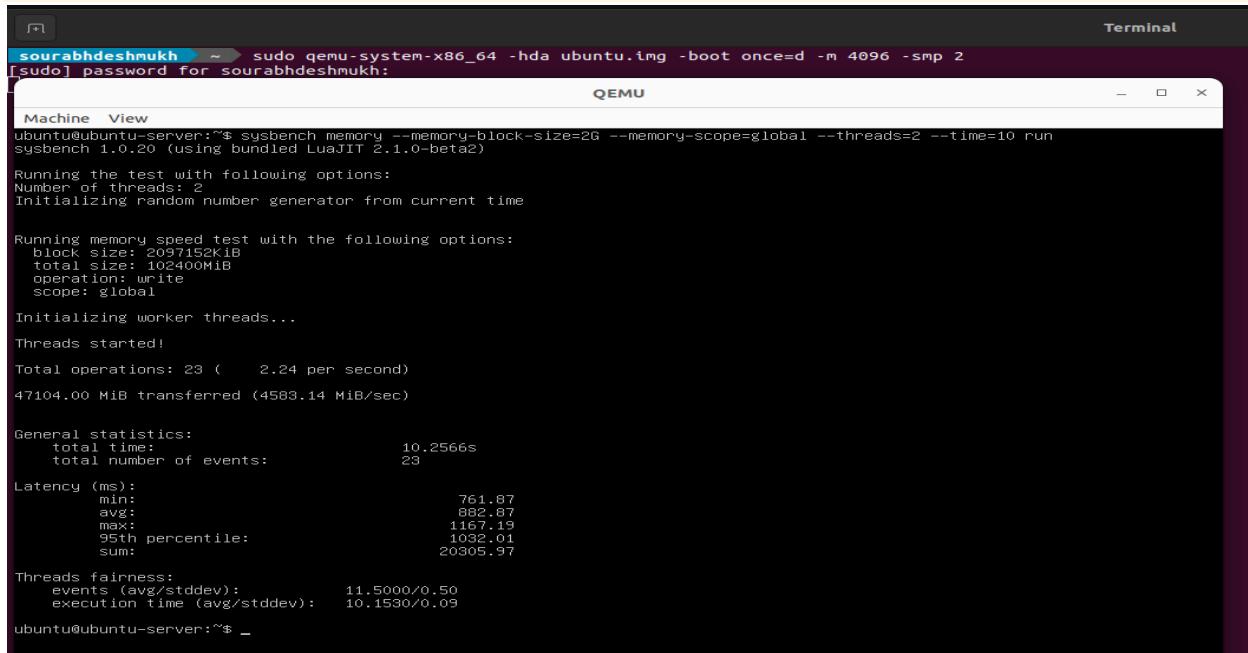
Threads fairness:
    events (avg/stddev): 25.5000/0.50
    execution time (avg/stddev): 10.1890/0.05
ubuntu@ubuntu-server:~$ _
```

Sourabh Deshmukh

W1648445

### Test Case 2:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=2G --memory-scope=global  
--threads=2 --time=10 run
```

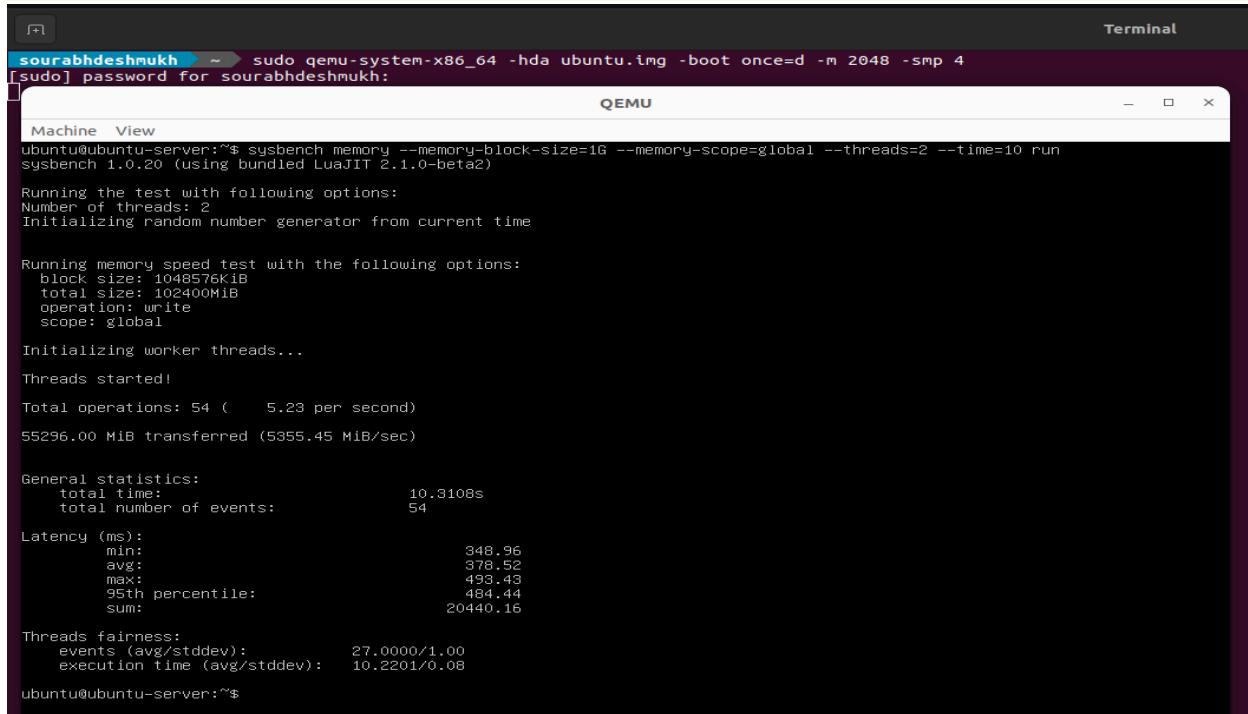


```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 2  
[sudo] password for sourabhdeshmukh:  
QEMU  
Machine View  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 2097152KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 23 ( 2.24 per second)  
47104.00 MiB transferred (4589.14 MiB/sec)  
  
General statistics:  
total time: 10.2566s  
total number of events: 23  
  
Latency (ms):  
min: 761.87  
avg: 882.87  
max: 1167.19  
95th percentile: 1032.01  
sum: 20305.97  
  
Threads fairness:  
events (avg/stddev): 11.5000/0.50  
execution time (avg/stddev): 10.1530/0.09  
ubuntu@ubuntu-server:~$ _
```

### 3. Test case with 4 CPU and 2 RAM

#### Test Case 1:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=1G --memory-scope=global  
--threads=2 --time=10 run
```



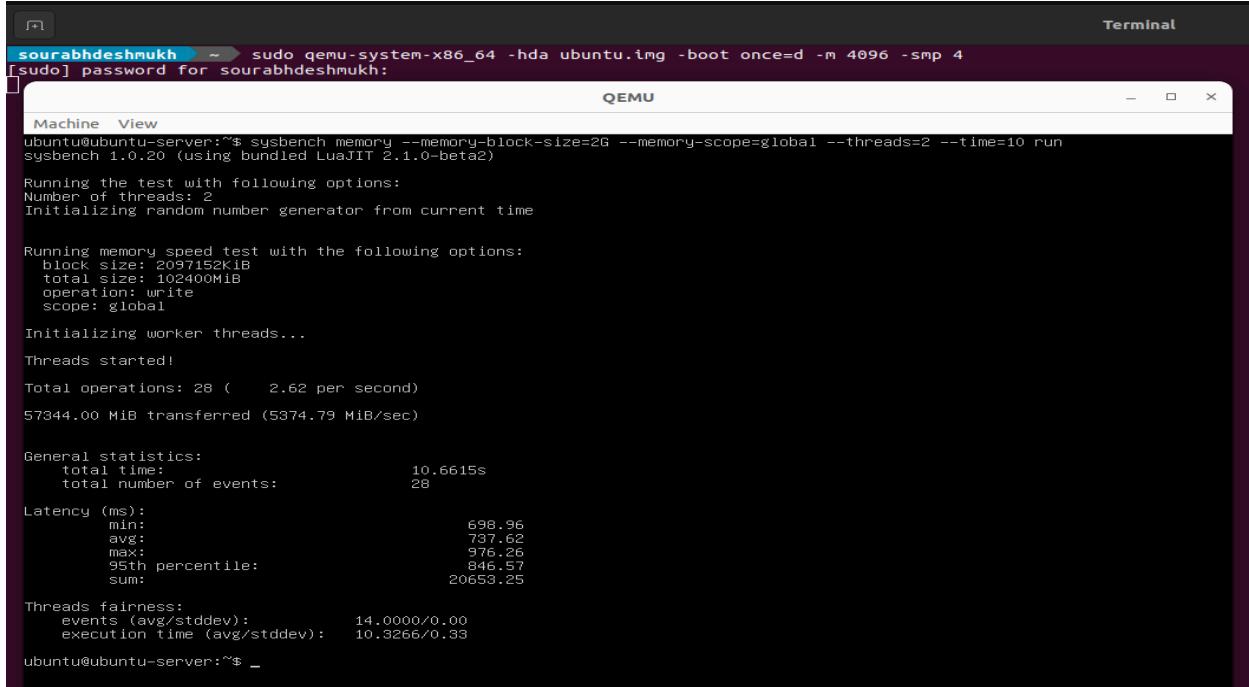
```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 4  
[sudo] password for sourabhdeshmukh:  
QEMU  
Machine View  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 1048576KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 54 ( 5.23 per second)  
55296.00 MiB transferred (5355.45 MiB/sec)  
  
General statistics:  
total time: 10.3108s  
total number of events: 54  
  
Latency (ms):  
min: 348.96  
avg: 378.52  
max: 493.43  
95th percentile: 484.44  
sum: 20440.16  
  
Threads fairness:  
events (avg/stddev): 27.0000/1.00  
execution time (avg/stddev): 10.2201/0.08  
ubuntu@ubuntu-server:~$ _
```

Sourabh Deshmukh

W1648445

### Test Case 2:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=2G --memory-scope=global  
--threads=2 --time=10 run
```

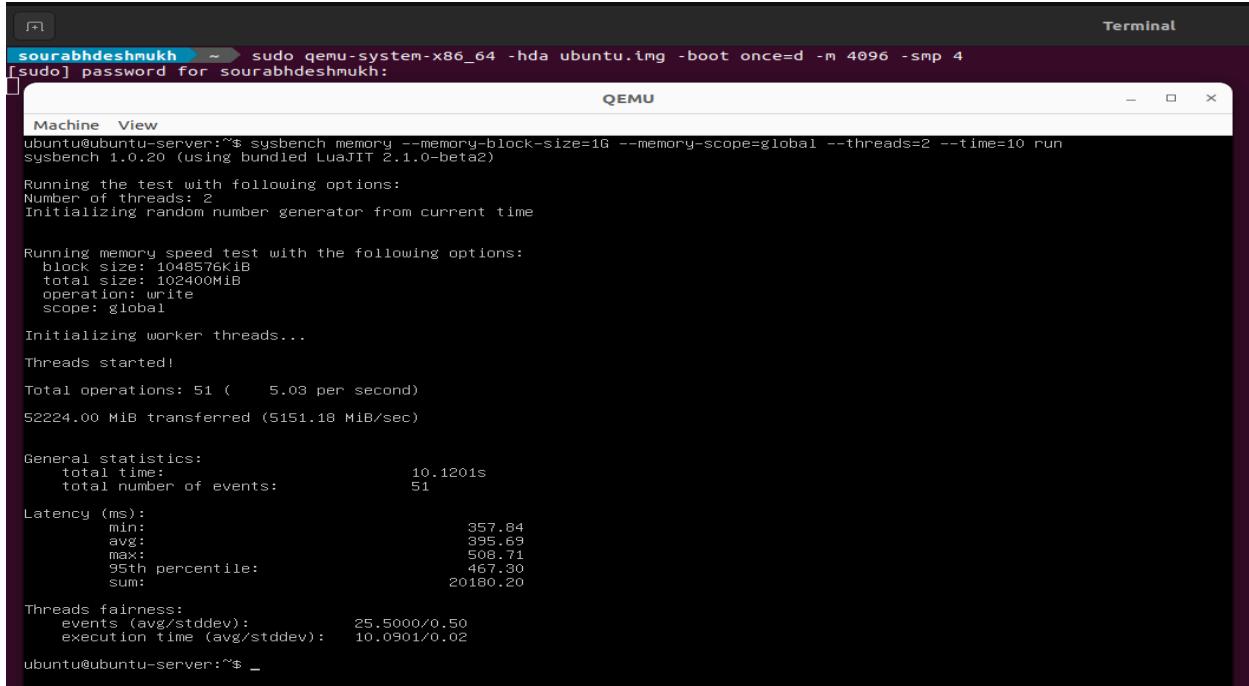


```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 4  
[sudo] password for sourabhdeshmukh:  
Machine View QEMU  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 2097152KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
Total operations: 28 ( 2.62 per second)  
57344.00 MiB transferred (5374.79 MiB/sec)  
  
General statistics:  
total time: 10.6615s  
total number of events: 28  
  
Latency (ms):  
min: 698.96  
avg: 737.62  
max: 976.26  
95th percentile: 846.57  
sum: 20653.25  
  
Threads fairness:  
events (avg/stddev): 14.0000/0.00  
execution time (avg/stddev): 10.3266/0.33  
ubuntu@ubuntu-server:~$ _
```

## 4. Test case with 4 CPU and 4 RAM

### Test Case 1:

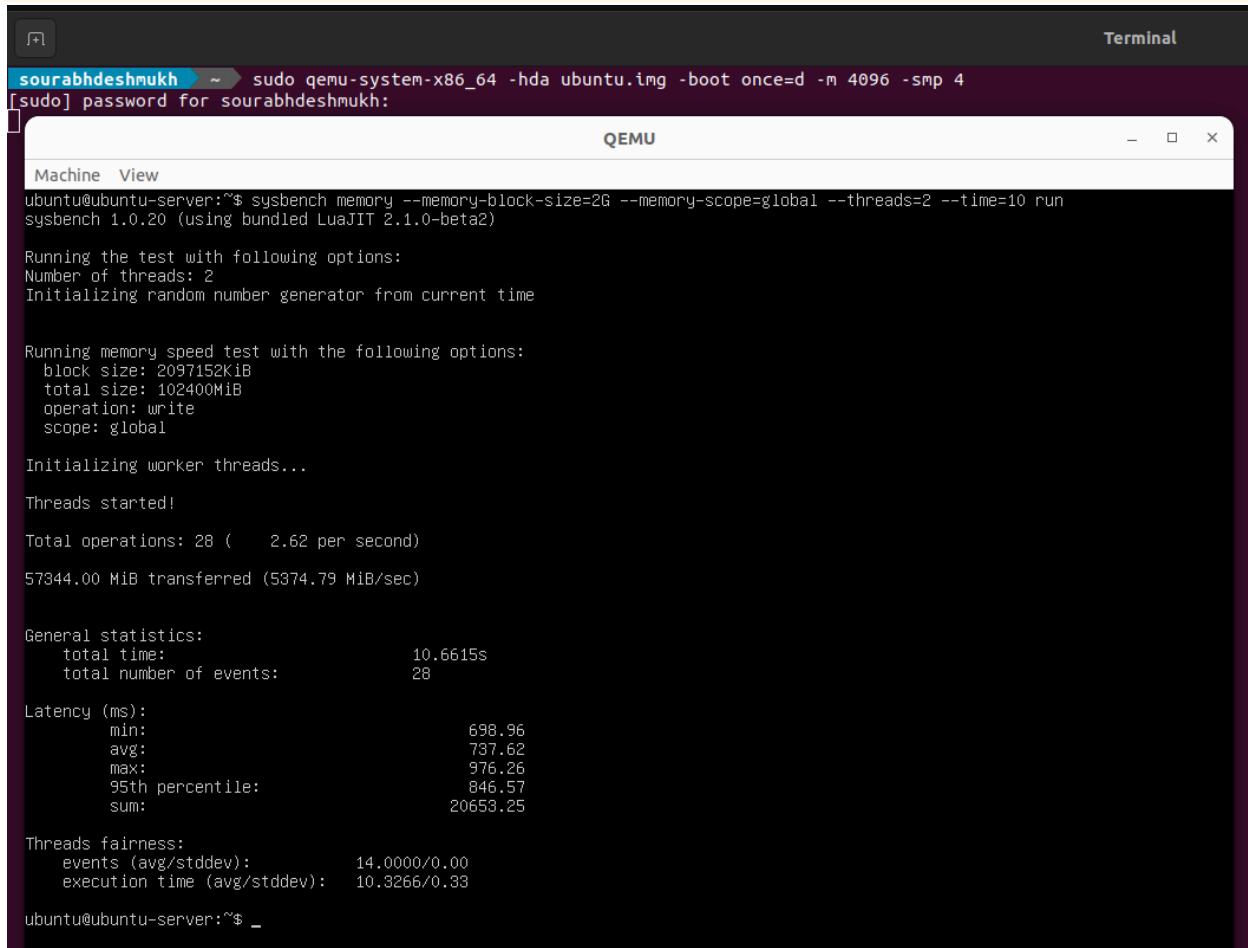
```
sourabhdeshmukh$ sysbench memory --memory-block-size=1G --memory-scope=global  
--threads=2 --time=10 run
```



```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 4  
[sudo] password for sourabhdeshmukh:  
Machine View QEMU  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 1048576KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
Total operations: 51 ( 5.03 per second)  
52224.00 MiB transferred (5151.18 MiB/sec)  
  
General statistics:  
total time: 10.1201s  
total number of events: 51  
  
Latency (ms):  
min: 357.84  
avg: 395.69  
max: 508.71  
95th percentile: 467.30  
sum: 20180.20  
  
Threads fairness:  
events (avg/stddev): 25.5000/0.50  
execution time (avg/stddev): 10.0901/0.02  
ubuntu@ubuntu-server:~$ _
```

**Test Case 2:**

```
sourabhdeshmukh$ sysbench memory --memory-block-size=2G --memory-scope=global  
--threads=2 --time=10 run
```



The screenshot shows a terminal window titled "Terminal" with a dark background. The command `sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 4` is entered, followed by a password prompt "[sudo] password for sourabhdeshmukh:". Below the terminal, a QEMU window titled "QEMU" is visible, showing the output of the sysbench memory test. The test details include: Number of threads: 2, Initializing random number generator from current time, Running memory speed test with the following options: block size: 2097152KiB, total size: 102400MiB, operation: write, scope: global. It then initializes worker threads, starts threads, and provides performance metrics: Total operations: 28 ( 2.62 per second), 57344.00 MiB transferred (5374.79 MiB/sec). It also displays general statistics and latency details. The test concludes with the command `ubuntu@ubuntu-server:~$ _`.

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 4  
[sudo] password for sourabhdeshmukh:  
Machine View QEMU  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 2097152KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 28 ( 2.62 per second)  
57344.00 MiB transferred (5374.79 MiB/sec)  
  
General statistics:  
total time: 10.6615s  
total number of events: 28  
  
Latency (ms):  
min: 698.96  
avg: 737.62  
max: 976.26  
95th percentile: 846.57  
sum: 20653.25  
  
Threads fairness:  
events (avg/stddev): 14.0000/0.00  
execution time (avg/stddev): 10.3266/0.33  
ubuntu@ubuntu-server:~$ _
```

## Memory Tests on QEMU RAW

### 1. Test case with 2 CPU and 2 RAM

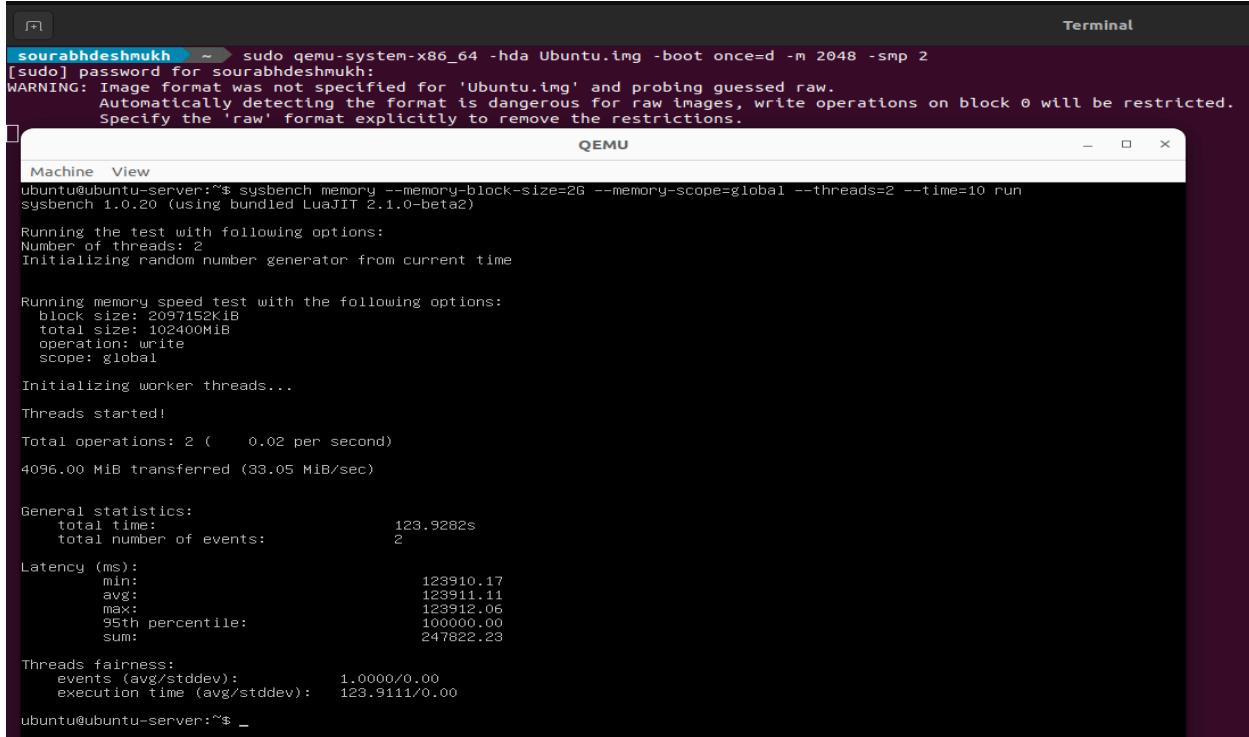
#### Test Case 1:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=1G --memory-scope=global  
--threads=2 --time=10 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 2  
[sudo] password for sourabhdeshmukh:  
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.  
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.  
Specify the 'raw' format explicitly to remove the restrictions.  
Machine View QEMU  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuajIT 2.1.0-beta2)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 1048576KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 50 ( 4.97 per second)  
51200.00 MiB transferred (5093.91 MiB/sec)  
  
General statistics:  
total time: 10.0446s  
total number of events: 50  
  
Latency (ms):  
min: 282.82  
avg: 400.94  
max: 569.29  
95th percentile: 520.62  
sum: 20047.09  
  
Threads fairness:  
events (avg/stddev): 25.0000/1.00  
execution time (avg/stddev): 10.0235/0.01  
ubuntu@ubuntu-server:~$ _
```

#### Test Case 2:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=2G --memory-scope=global  
--threads=2 --time=10 run
```



```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 2
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View QEMU
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Running memory speed test with the following options:
block size: 2097152KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 2 ( 0.02 per second)
4096.00 MiB transferred (33.05 MiB/sec)

General statistics:
total time: 123.9282s
total number of events: 2

Latency (ms):
min: 123910.17
avg: 123911.11
max: 123912.06
95th percentile: 100000.00
sum: 247822.23

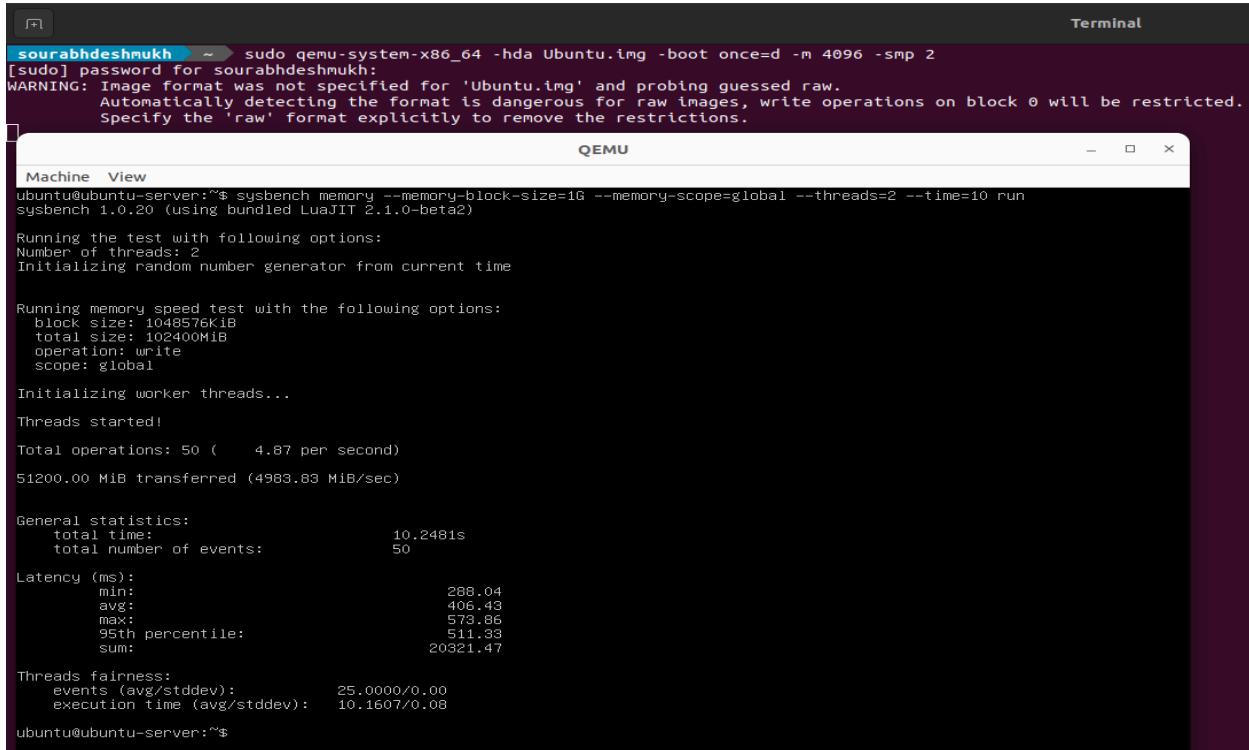
Threads fairness:
events (avg/stddev): 1.0000/0.00
execution time (avg/stddev): 123.9111/0.00

ubuntu@ubuntu-server:~$
```

## 2. Test case with 2 CPU and 4 RAM

### Test Case 1:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=1G --memory-scope=global
--threads=2 --time=10 run
```



```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 2
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View QEMU
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Running memory speed test with the following options:
block size: 1048576KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...
Threads started!

Total operations: 50 ( 4.87 per second)
51200.00 MiB transferred (4983.83 MiB/sec)

General statistics:
total time: 10.2481s
total number of events: 50

Latency (ms):
min: 288.04
avg: 406.43
max: 573.86
95th percentile: 511.33
sum: 20321.47

Threads fairness:
events (avg/stddev): 25.0000/0.00
execution time (avg/stddev): 10.1607/0.08

ubuntu@ubuntu-server:~$
```

### Test Case 2:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=2G --memory-scope=global  
--threads=2 --time=10 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 2  
[sudo] password for sourabhdeshmukh:  
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.  
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.  
Specify the 'raw' format explicitly to remove the restrictions.  
Machine View QEMU  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 2097152KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
Total operations: 21 ( 1.94 per second)  
43008.00 MiB transferred (3968.75 MiB/sec)  
  
General statistics:  
total time: 10.8223s  
total number of events: 21  
  
Latency (ms):  
min: 797.52  
avg: 993.99  
max: 1339.99  
95th percentile: 1191.92  
sum: 20861.18  
  
Threads fairness:  
events (avg/stddev): 10.5000/1.50  
execution time (avg/stddev): 10.4306/0.39  
ubuntu@ubuntu-server:~$ _
```

### 3. Test case with 4 CPU and 2 RAM

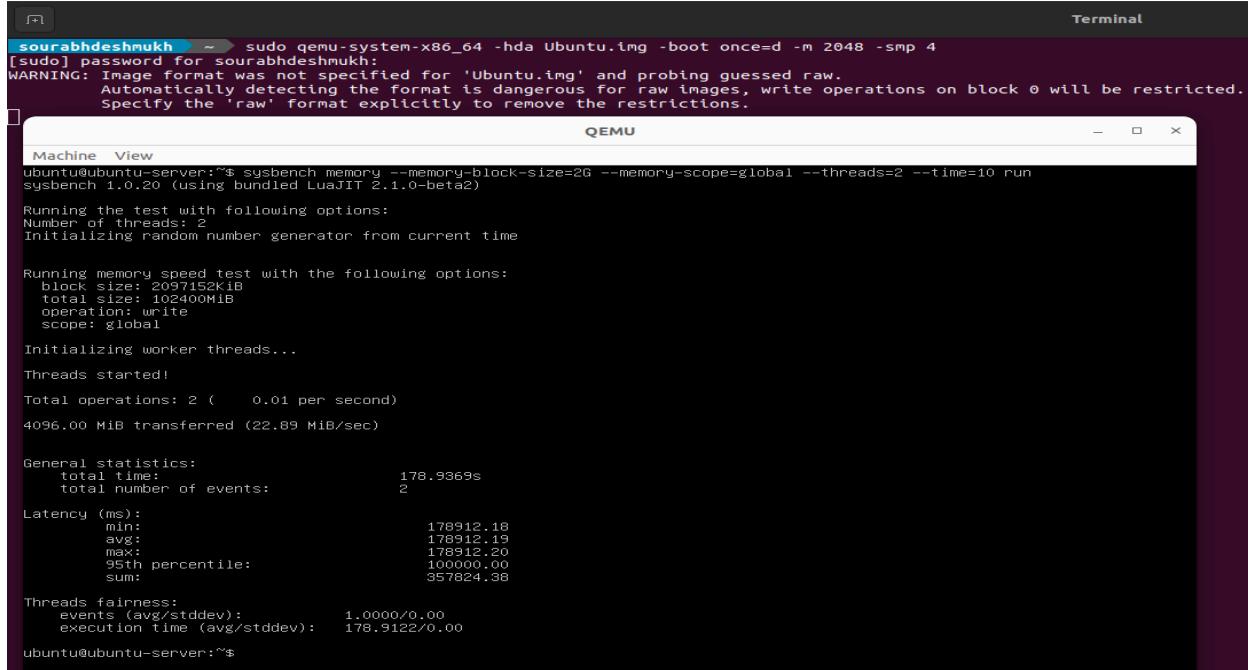
#### Test Case 1:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=1G --memory-scope=global  
--threads=2 --time=10 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 4  
[sudo] password for sourabhdeshmukh:  
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.  
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.  
Specify the 'raw' format explicitly to remove the restrictions.  
Machine View QEMU  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 1048576KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
Total operations: 54 ( 5.26 per second)  
55296.00 MiB transferred (5389.83 MiB/sec)  
  
General statistics:  
total time: 10.2426s  
total number of events: 54  
  
Latency (ms):  
min: 342.71  
avg: 374.69  
max: 457.37  
95th percentile: 434.83  
sum: 20233.02  
  
Threads fairness:  
events (avg/stddev): 27.0000/1.00  
execution time (avg/stddev): 10.1165/0.12  
ubuntu@ubuntu-server:~$ _
```

### Test Case 2:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=2G --memory-scope=global  
--threads=2 --time=10 run
```

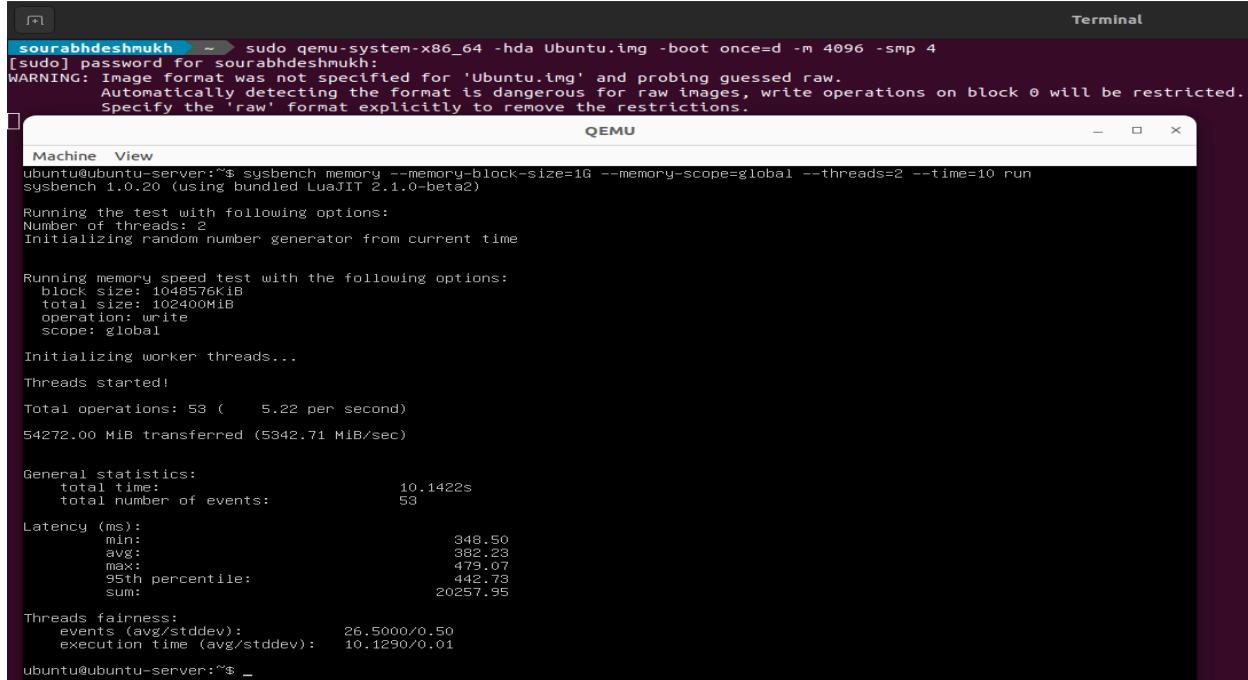


```
sourabhdeshmukh ~ > sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 4  
[sudo] password for sourabhdeshmukh:  
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.  
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.  
Specify the 'raw' format explicitly to remove the restrictions.  
Machine View  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 2097152kB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 2 (~ 0.01 per second)  
4096.00 MiB transferred (22.89 MiB/sec)  
  
General statistics:  
total time: 178.9369s  
total number of events: 2  
  
Latency (ms):  
min: 178912.18  
avg: 178912.19  
max: 178912.20  
95th percentile: 100000.00  
sum: 357824.98  
  
Threads fairness:  
events (avg/stddev): 1.0000/0.00  
execution time (avg/stddev): 178.9122/0.00  
ubuntu@ubuntu-server:~$
```

### 4. Test case with 4 CPU and 4 RAM

#### Test Case 1:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=1G --memory-scope=global  
--threads=2 --time=10 run
```



```
sourabhdeshmukh ~ > sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 4  
[sudo] password for sourabhdeshmukh:  
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.  
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.  
Specify the 'raw' format explicitly to remove the restrictions.  
Machine View  
ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 1048576kB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 53 (~ 5.22 per second)  
54272.00 MiB transferred (5342.71 MiB/sec)  
  
General statistics:  
total time: 10.1422s  
total number of events: 53  
  
Latency (ms):  
min: 348.50  
avg: 382.23  
max: 479.07  
95th percentile: 442.78  
sum: 20257.95  
  
Threads fairness:  
events (avg/stddev): 26.5000/0.50  
execution time (avg/stddev): 10.1290/0.01  
ubuntu@ubuntu-server:~$
```

Sourabh Deshmukh

W1648445

### Test Case 2:

```
sourabhdeshmukh$ sysbench memory --memory-block-size=2G --memory-scope=global  
--threads=2 --time=10 run
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 4  
[sudo] password for sourabhdeshmukh:  
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.  
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.  
Specify the 'raw' format explicitly to remove the restrictions.  
QEMU  
Machine View  
ubuntu@Ubuntu-server:~$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 2097152Kib  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 26 ( 2.53 per second)  
53248.00 MiB transferred (5174.90 MiB/sec)  
  
General statistics:  
total time: 10.2682s  
total number of events: 26  
  
Latency (ms):  
min: 718.60  
avg: 785.94  
max: 1087.45  
95th percentile: 960.30  
sum: 20434.37  
  
Threads fairness:  
events (avg/stddev): 13.0000/1.00  
execution time (avg/stddev): 10.2172/0.04  
ubuntu@Ubuntu-server:~$
```

## Memory Tests on Docker Container

### 1. Test case with 2 CPU and 2 RAM

#### Test Case 1:

```
docker$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2  
--time=10 run
```

# Sourabh Deshmukh

W1648445

```
root@319cabda9bb0:/# sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Running memory speed test with the following options:
  block size: 1048576KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 100 (    9.96 per second)
102400.00 MiB transferred (10194.57 MiB/sec)

General statistics:
  total time:          10.0425s
  total number of events: 100

Latency (ms):
  min:                169.70
  avg:                200.38
  max:                255.20
  95th percentile:   253.35
  sum:               20038.25

Threads fairness:
  events (avg/stddev): 50.0000/0.00
  execution time (avg/stddev): 10.0191/0.02

root@319cabda9bb0:/#
```

## Test Case 2:

```
docker$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2
--time=10 run
```

```
root@319cabda9bb0:/# sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Running memory speed test with the following options:
  block size: 2097152KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 50 (    5.48 per second)
102400.00 MiB transferred (11219.38 MiB/sec)

General statistics:
  total time:          9.1256s
  total number of events: 50

Latency (ms):
  min:                351.46
  avg:                364.88
  max:                502.88
  95th percentile:   419.45
  sum:               18243.95

Threads fairness:
  events (avg/stddev): 25.0000/0.00
  execution time (avg/stddev): 9.1220/0.00

root@319cabda9bb0:/#
```

## 2. Test case with 2 CPU and 4 RAM

### Test Case 1:

```
docker$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2  
--time=10 run
```

```
root@43649c9038ed:/# sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 1048576KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 98 ( 9.74 per second)  
100352.00 MiB transferred (9978.54 MiB/sec)  
  
General statistics:  
    total time: 10.0547s  
    total number of events: 98  
  
Latency (ms):  
    min: 177.91  
    avg: 205.05  
    max: 280.80  
    95th percentile: 267.41  
    sum: 20095.10  
  
Threads fairness:  
    events (avg/stddev): 49.0000/0.00  
    execution time (avg/stddev): 10.0475/0.01  
root@43649c9038ed:/#
```

### Test Case 2:

```
docker$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2  
--time=10 run
```

```
root@43649c9038ed:/# sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 2097152KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 50 ( 5.04 per second)  
102400.00 MiB transferred (10315.31 MiB/sec)  
  
General statistics:  
    total time: 9.9255s  
    total number of events: 50  
  
Latency (ms):  
    min: 353.63  
    avg: 396.39  
    max: 595.55  
    95th percentile: 580.02  
    sum: 19819.55  
  
Threads fairness:  
    events (avg/stddev): 25.0000/0.00  
    execution time (avg/stddev): 9.9098/0.02  
root@43649c9038ed:/#
```

### 3. Test case with 4 CPU and 2 RAM

#### Test Case 1:

```
docker$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2  
--time=10 run
```

```
root@d14f158f931f:/# sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 1048576KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 100 ( 9.99 per second)  
102400.00 MiB transferred (10233.84 MiB/sec)  
  
General statistics:  
total time: 10.0027s  
total number of events: 100  
  
Latency (ms):  
min: 172.20  
avg: 199.72  
max: 269.40  
95th percentile: 240.02  
sum: 19972.40  
  
Threads fairness:  
events (avg/stddev): 50.0000/0.00  
execution time (avg/stddev): 9.9862/0.02  
root@d14f158f931f:/#
```

#### Test Case 2:

```
docker$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2  
--time=10 run
```

```
root@d14f158f931f:/# sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 2097152KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 46 ( 4.44 per second)  
94208.00 MiB transferred (9085.03 MiB/sec)  
  
General statistics:  
total time: 10.3668s  
total number of events: 46  
  
Latency (ms):  
min: 351.73  
avg: 450.70  
max: 607.30  
95th percentile: 511.33  
sum: 20732.16  
  
Threads fairness:  
events (avg/stddev): 23.0000/0.00  
execution time (avg/stddev): 10.3661/0.00  
root@d14f158f931f:/#
```

#### 4. Test case with 4 CPU and 4 RAM

##### Test Case 1:

```
docker$ sysbench memory --memory-block-size=1G --memory-scope=global --threads=2  
--time=10 run
```

```
root@804d2a75d4a0:/# sysbench memory --memory-block-size=1G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 1048576KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 100 ( 10.36 per second)  
102400.00 MiB transferred (10604.59 MiB/sec)  
  
General statistics:  
    total time: 9.6539s  
    total number of events: 100  
  
Latency (ms):  
    min: 173.82  
    avg: 192.65  
    max: 236.16  
    95th percentile: 223.34  
    sum: 19264.98  
  
Threads fairness:  
    events (avg/stddev): 50.0000/0.00  
    execution time (avg/stddev): 9.6325/0.02  
root@804d2a75d4a0:/# █
```

##### Test Case 2:

```
docker$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2  
--time=10 run
```

```
root@804d2a75d4a0:/# sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run  
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Running memory speed test with the following options:  
block size: 2097152KiB  
total size: 102400MiB  
operation: write  
scope: global  
  
Initializing worker threads...  
Threads started!  
  
Total operations: 50 ( 4.89 per second)  
102400.00 MiB transferred (10008.59 MiB/sec)  
  
General statistics:  
    total time: 10.2278s  
    total number of events: 50  
  
Latency (ns):  
    min: 338.55  
    avg: 408.88  
    max: 541.58  
    95th percentile: 484.44  
    sum: 20444.02  
  
Threads fairness:  
    events (avg/stddev): 25.0000/0.00  
    execution time (avg/stddev): 10.2220/0.01  
root@804d2a75d4a0:/# █
```

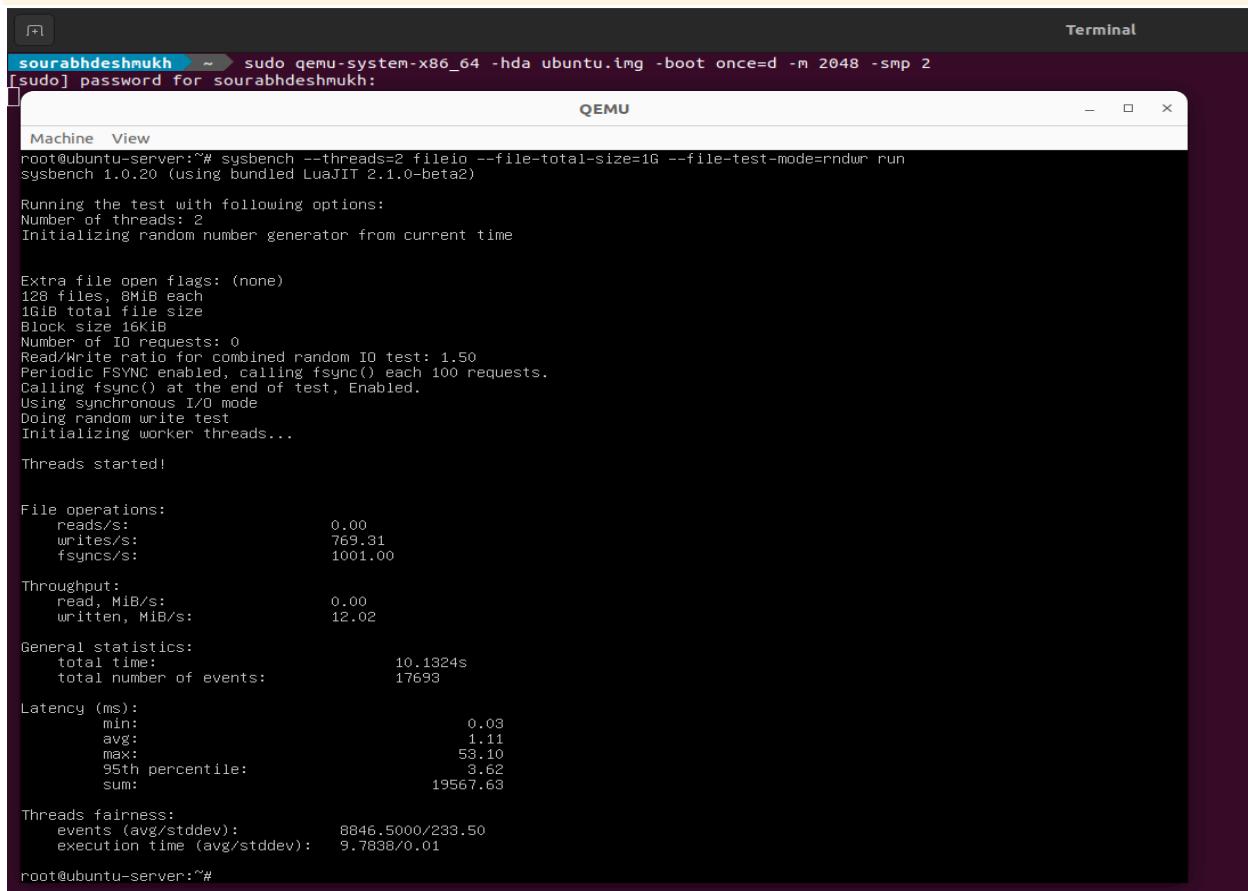
## FILEIO Test

### Fileio Tests on QEMU QCOW2

#### 1. Test case with 2 CPU and 2 RAM

##### Test Case 1:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr cleanup
```



The screenshot shows a terminal window titled "Terminal" with the title bar "QEMU". The terminal output is as follows:

```
sourabhdeshmukh ~ $ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 2  
[sudo] password for sourabhdeshmukh:  
Machine View  
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
  
Running the test with following options:  
Number of threads: 2  
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: 789.31  
fsyncs/s: 1001.00  
  
Throughput:  
read, MiB/s: 0.00  
written, MiB/s: 12.02  
  
General statistics:  
total time: 10.1324s  
total number of events: 17693  
  
Latency (ms):  
min: 0.08  
avg: 1.11  
max: 53.10  
95th percentile: 3.62  
sum: 19567.63  
  
Threads fairness:  
events (avg/stddev): 8846.5000/233.50  
execution time (avg/stddev): 9.7838/0.01  
  
root@ubuntu-server:~#
```

##### Test Case 2:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr cleanup
```

```
sourabhdeshmukh ~ ➔ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 2  
[sudo] password for sourabhdeshmukh:  
QEMU  
Machine View  
Building file test file 127  
1073741824 bytes written in 15.73 seconds (65.11 MiB/sec).  
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqwr run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
Max files: 800B each  
1GB total filesize  
Block size: 16KB  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing sequential rewrite test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
 reads/s: 0.00  
 writes/s: 2445.91  
 fsyncs/s: 3153.24  
  
Throughput:  
 read, MiB/s: 0.00  
 written, MiB/s: 38.22  
  
General statistics:  
 total time: 10.0475s  
 total number of events: 56058  
  
Latency (ms):  
 min: 0.03  
 avg: 0.34  
 max: 29.76  
 95th percentile: 0.63  
 sum: 19291.63  
  
Threads fairness:  
 events (avg/stddev): 28029.0000/246.00  
 execution time (avg/stddev): 9.6459/0.01  
root@ubuntu-server:~#
```

## 2. Test case with 2 CPU and 4 RAM

## Test Case 1:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr cleanup
```

```
[sourabhdeshmukh ~]# sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 2
[sudo] password for sourabhdeshmukh:
[sourabhdeshmukh ~]# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of Threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1024MiB total memory size
Block size: 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
 reads/s:           0.00
 writes/s:          2245.96
 fsyncs/s:          2892.02

Throughput:
 read, MiB/s:       0.00
 written, MiB/s:    35.09

General statistics:
 total time:        10.0535s
 total number of events: 51445

Latency (ms):
 min:                0.03
 avg:                0.37
 max:                39.63
 95th percentile:   0.65
 sum:               19273.15

Threads fairness:
 events (avg/stddev): 25722.5000/548.50
 execution time (avg/stddev): 9.6366/0.03

[sourabhdeshmukh ~]#
```

**Sourabh Deshmukh**

**W1648445**

**Test Case 2:**

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr cleanup
```

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 2  
[sudo] password for sourabhdeshmukh:  
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run  
sysbench 1.0.20 (using bundled LuAJIT 2.1.0-beta2)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 8MiB each  
1GiB 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:        907.95  
  fsyncs/s:       1177.87  
  
Throughput:  
  read, MiB/s:      0.00  
  written, MiB/s:   14.19  
  
General statistics:  
  total time:        10.1271s  
  total number of events: 20879  
  
Latency (ms):  
  min:                0.03  
  avg:                0.94  
  max:                18.37  
  95th percentile:    3.19  
  sum:            19585.52  
  
Threads fairness:  
  events (avg/stddev): 10439.5000/119.50  
  execution time (avg/stddev): 9.7928/0.00  
  
root@ubuntu-server:~#
```

**3. Test case with 4 CPU and 2 RAM**

**Test Case 1:**

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr cleanup
```

```

sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 4
[sudo] password for sourabhdeshmukh:
[QEMU]
Machine View
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqwr run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Doing (fsync()) at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
reads/s: 0.00
writes/s: 1565.58
fsyncs/s: 2020.29

Throughput:
read, MiB/s: 0.00
written, MiB/s: 24.46

General statistics:
total time: 10.0879s
total number of events: 35933

Latency (ms):
min: 0.03
avg: 0.54
max: 31.99
95th percentile: 0.99
sum: 19288.29

Threads fairness:
events (avg/stddev): 17966.5000/62.50
execution time (avg/stddev): 9.6441/0.00
root@ubuntu-server:~#

```

### Test Case 2:

```

sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr prepare
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr run
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr cleanup

```

```

sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 4
[sudo] password for sourabhdeshmukh:
[QEMU]
Machine View
Ubuntu@ubuntu-server:~$ sysbench memory --memory-block-size=2G --memory-scope=global --threads=2 --time=10 run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Running memory speed test with the following options:
block size: 2097152KiB
total size: 102400MiB
operation: write
scope: global

Initializing worker threads...

Threads started!

Total operations: 2 ( 0.01 per second)
4096.00 MiB transferred (21.78 MiB/sec)

General statistics:
total time: 188.0210s
total number of events: 2

Latency (ms):
min: 187985.29
avg: 187985.91
max: 187986.53
95th percentile: 100000.00
sum: 375971.82

Threads fairness:
events (avg/stddev): 1.0000/0.00
execution time (avg/stddev): 187.9859/0.00
ubuntu@ubuntu-server:~$ 

```

#### 4. Test case with 4 CPU and 4 RAM

##### Test Case 1:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr cleanup
```

The screenshot shows a terminal window titled "Terminal". The terminal content is as follows:

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 4  
[sudo] password for sourabhdeshmukh:  
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 8MiB each  
1GiB total file size  
Block size 16KiB  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing sequential rewrite test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s:          0.00  
  writes/s:        1581.56  
  fsyncs/s:        1976.41  
  
Throughput:  
  read, MiB/s:     0.00  
  written, MiB/s:  23.93  
  
General statistics:  
  total time:      10.1085s  
  total number of events: 35246  
  
Latency (ms):  
  min:              0.08  
  avg:              0.55  
  max:             37.31  
  95th percentile: 0.95  
  sum:            19300.26  
  
Threads fairness:  
  events (avg/stddev): 17623.0000/16.00  
  execution time (avg/stddev): 9.6501/0.01  
  
root@ubuntu-server:~# _
```

##### Test Case 2:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr cleanup
```

# Sourabh Deshmukh

W1648445

```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 4096 -smp 4
[sudo] password for sourabhdeshmukh:
QEMU
Machine View
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndrw run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB 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: 497.94
 fsyncs/s: 651.56

Throughput:
 read, MiB/s: 0.00
 written, MiB/s: 7.78

General statistics:
 total time: 10.8036s
 total number of events: 12210

Latency (ms):
 min: 0.03
 avg: 1.60
 max: 38.18
 95th percentile: 5.37
 sum: 19544.40

Threads fairness:
 events (avg/stddev): 6105.0000/21.00
 execution time (avg/stddev): 9.7722/0.01

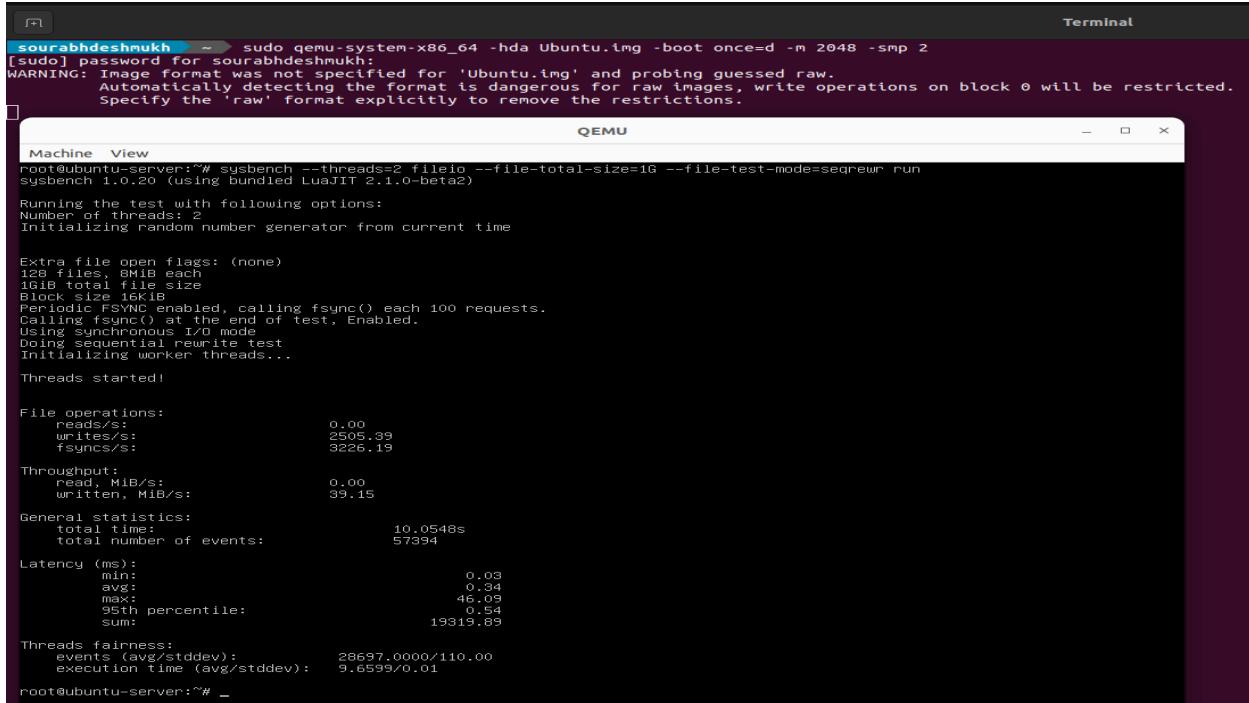
root@ubuntu-server:~# _
```

## Fileio Tests on QEMU RAW

## 1. Test case with 2 CPU and 2 RAM

## Test Case 1:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr cleanup
```



```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 2
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
10GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         2505.39
  fsyncs/s:         3226.19

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   39.15

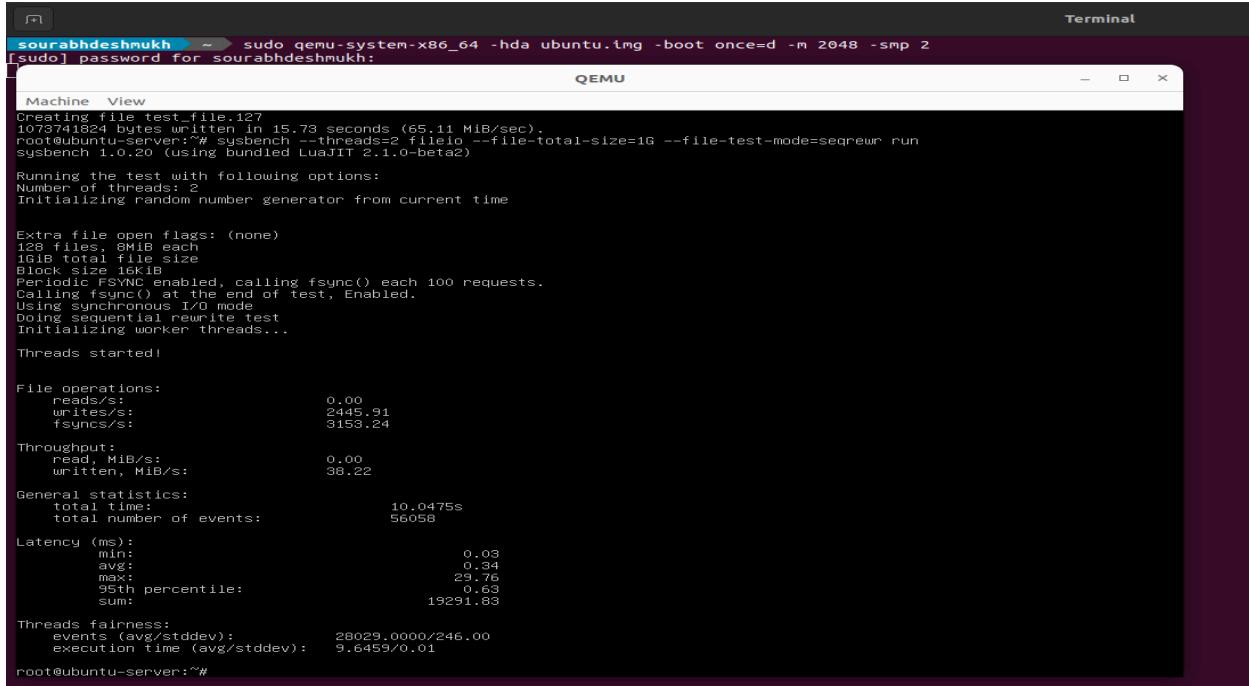
General statistics:
  total time:       10.0548s
  total number of events: 57394

Latency (ms):
  min:              0.03
  avg:              0.34
  max:              46.09
  95th percentile:  0.54
  sum:              19319.69

Threads fairness:
  events (avg/stddev): 28697.0000/110.00
  execution time (avg/stddev): 9.6599/0.01
root@ubuntu-server:~# _
```

### Test Case 2:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr prepare
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr run
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr cleanup
```



```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda ubuntu.img -boot once=d -m 2048 -smp 2
[sudo] password for sourabhdeshmukh:
Machine View
Creating file test_file.127
1073741824 bytes written in 15.73 seconds (65.11 MiB/sec).
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
10GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:         2445.91
  fsyncs/s:         3153.24

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   38.22

General statistics:
  total time:       10.0475s
  total number of events: 56058

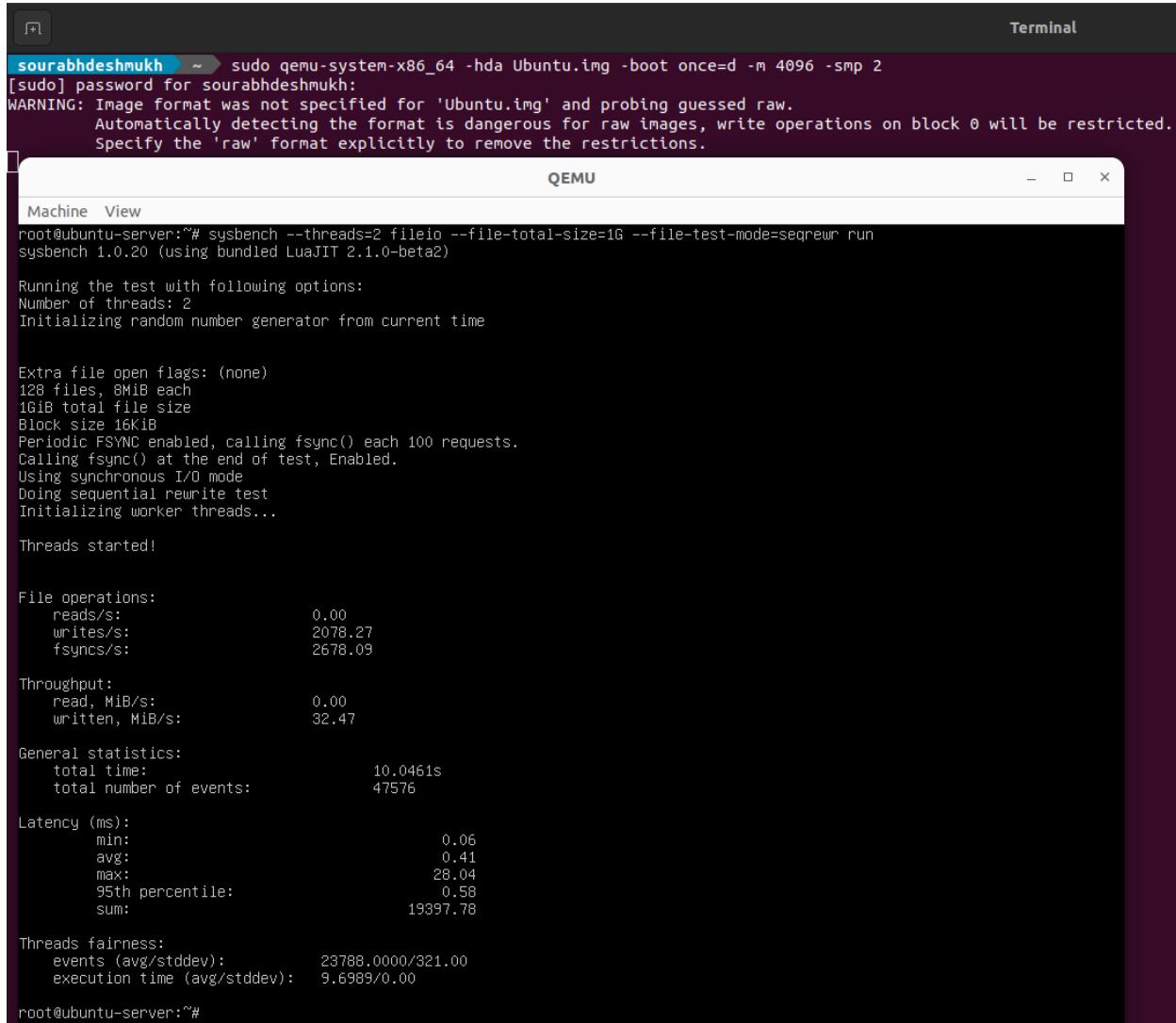
Latency (ms):
  min:              0.03
  avg:              0.34
  max:              29.76
  95th percentile:  0.63
  sum:              19291.83

Threads fairness:
  events (avg/stddev): 28029.0000/246.00
  execution time (avg/stddev): 9.6459/0.01
root@ubuntu-server:~# _
```

## 2. Test case with 2 CPU and 4 RAM

### Test Case 1:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr cleanup
```



```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 2  
[sudo] password for sourabhdeshmukh:  
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.  
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.  
Specify the 'raw' format explicitly to remove the restrictions.  
[  ] QEMU [x]  
Machine View  
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 8MiB each  
1GiB total file size  
Block size 16KiB  
Periodic FSYNC enabled, calling fsync() each 100 requests.  
Calling fsync() at the end of test, Enabled.  
Using synchronous I/O mode  
Doing sequential rewrite test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
reads/s: 0.00  
writes/s: 2078.27  
fsyncs/s: 2678.09  
  
Throughput:  
read, MiB/s: 0.00  
written, MiB/s: 32.47  
  
General statistics:  
total time: 10.0461s  
total number of events: 47576  
  
Latency (ms):  
min: 0.06  
avg: 0.41  
max: 28.04  
95th percentile: 0.58  
sum: 19397.78  
  
Threads fairness:  
events (avg/stddev): 23788.0000/321.00  
execution time (avg/stddev): 9.6989/0.00  
root@ubuntu-server:~#
```

### Test Case 2:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr cleanup
```

```

[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndrw run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Number of IO requests: 0
Request size random 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:        859.82
  fsyncs/s:       1115.72

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   13.43

General statistics:
  total time:        10.11152s
  total number of events: 19736

Latency (ms):
  min:                0.03
  avg:                0.39
  max:               23.41
  95th percentile:    3.36
  sum:            19561.21

Threads fairness:
  events (avg/stddev): 9868.0000/178.00
  execution time (avg/stddev): 9.7806/0.01
root@ubuntu-server:~#

```

### 3. Test case with 4 CPU and 2 RAM

#### Test Case 1:

```

sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=seqrewr prepare
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=seqrewr run
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=seqrewr cleanup

```

```

[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        1648.07
  fsyncs/s:       2126.51

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   25.75

General statistics:
  total time:        10.0683s
  total number of events: 37763

Latency (ms):
  min:                0.03
  avg:                0.51
  max:               29.56
  95th percentile:    0.90
  sum:            19274.80

Threads fairness:
  events (avg/stddev): 18881.5000/86.50
  execution time (avg/stddev): 9.6371/0.01
root@ubuntu-server:~#

```

Sourabh Deshmukh

W1648445

### Test Case 2:

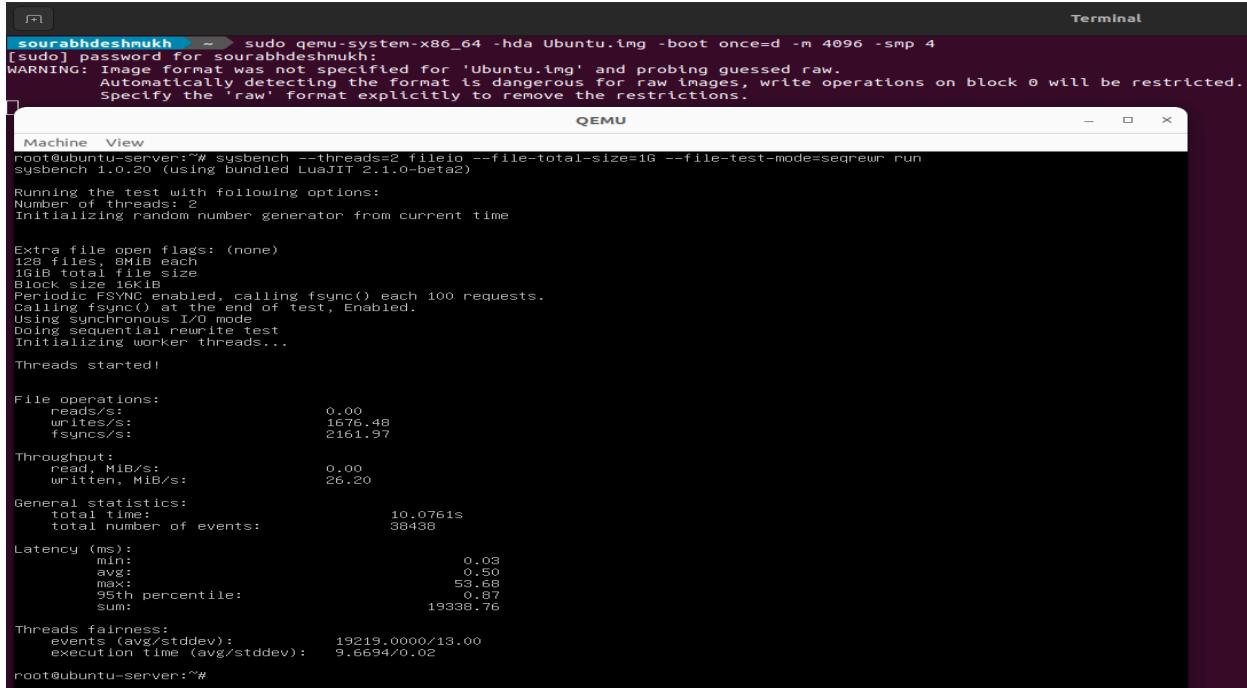
```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr cleanup
```

```
Terminal  
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 2048 -smp 4  
[sudo] password for sourabhdeshmukh:  
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.  
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.  
Specify the 'raw' format explicitly to remove the restrictions.  
QEMU  
Machine View  
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run  
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)  
  
Running the test with following options:  
Number of threads: 2  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 8MiB each  
16GiB 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: 708.71  
 fsyncs/s: 928.29  
  
Throughput:  
 read, MiB/s: 0.00  
 written, MiB/s: 11.07  
  
General statistics:  
 total time: 10.1478s  
 total number of events: 16324  
  
Latency (ms):  
 min: 0.03  
 avg: 1.20  
 max: 12.86  
 95th percentile: 3.89  
 sum: 19612.38  
  
Threads fairness:  
 events (avg/stddev): 8162.0000/44.00  
 execution time (avg/stddev): 9.8062/0.00  
root@ubuntu-server:~#
```

## 4. Test case with 4 CPU and 4 RAM

### Test Case 1:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr prepare  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=seqrewr cleanup
```



```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 4
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqwr run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
10GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        1676.48
  fsyncs/s:        2161.97

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   26.20

General statistics:
  total time:           10.0761s
  total number of events: 38438

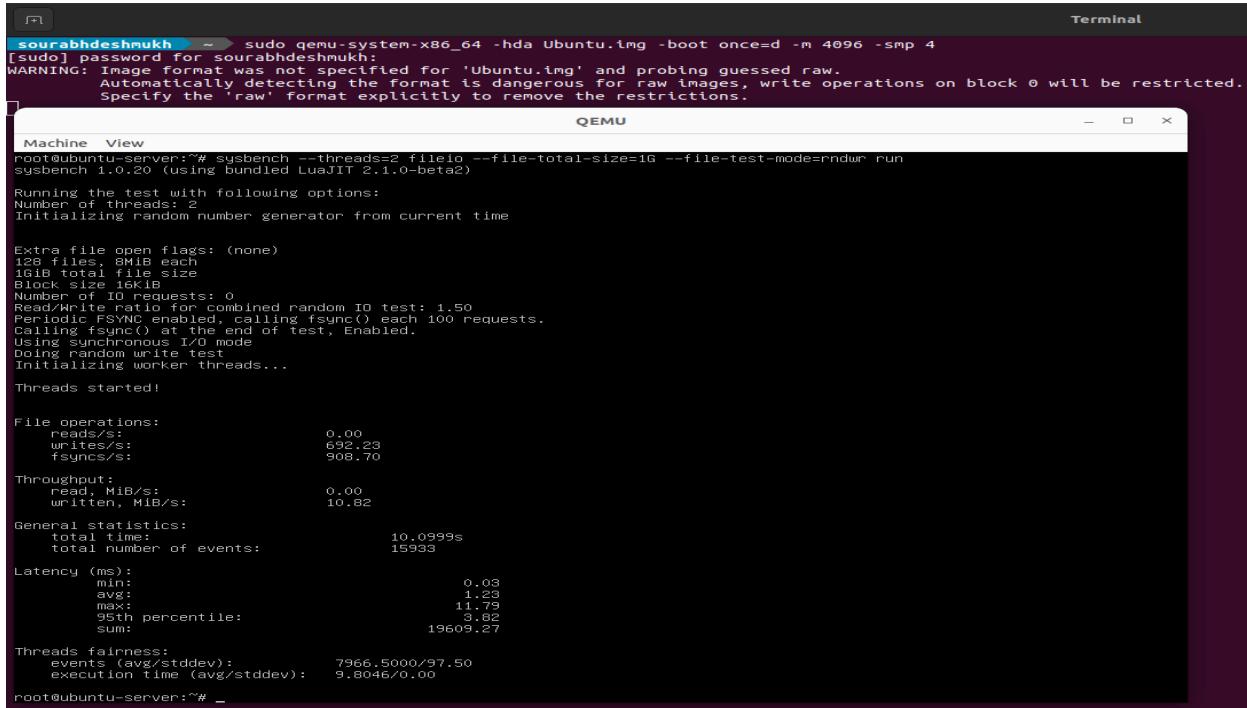
Latency (ms):
  min:                  0.03
  avg:                  0.50
  max:                 53.68
  95th percentile:     0.87
  sum:                 19338.76

Threads fairness:
  events (avg/stddev):    19219.0000/13.00
  execution time (avg/stddev): 9.6694/0.02

root@ubuntu-server:~#
```

### Test Case 2:

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr prepare
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr run
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr cleanup
```



```
sourabhdeshmukh ~ sudo qemu-system-x86_64 -hda Ubuntu.img -boot once=d -m 4096 -smp 4
[sudo] password for sourabhdeshmukh:
WARNING: Image format was not specified for 'Ubuntu.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.

Machine View
root@ubuntu-server:~# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run
sysbench 1.0.20 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
10GiB 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:        692.23
  fsyncs/s:        908.70

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   10.82

General statistics:
  total time:           10.0999s
  total number of events: 15933

Latency (ms):
  min:                  0.03
  avg:                  1.23
  max:                 11.79
  95th percentile:     3.82
  sum:                 19609.27

Threads fairness:
  events (avg/stddev):    7966.5000/97.50
  execution time (avg/stddev): 9.8046/0.00

root@ubuntu-server:~# _
```

## Fileio Tests on Docker Container

### 1. Test case with 2 CPU and 2 RAM

#### Test Case 1:

```
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
prepare
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
run
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
cleanup
```

```
root@319cabda9bb0:/# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        18207.72
  fsyncs/s:        23318.88

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   284.50

General statistics:
  total time:       10.0055s
  total number of events: 415290

Latency (ms):
  min:              0.00
  avg:              0.05
  max:             17.97
  95th percentile:  0.04
  sum:            19898.27

Threads fairness:
  events (avg/stddev):  207645.0000/842.00
  execution time (avg/stddev):  9.9491/0.00

root@319cabda9bb0:/#
```

#### Test Case 2:

```
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr
prepare
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr
cleanup
```

```
root@319cabda9bb0:/# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB 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:        4726.03
  fsyncs/s:        6073.80

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   73.84

General statistics:
  total time:           10.00069s
  total number of events: 107833

Latency (ms):
  min:                 0.00
  avg:                 0.19
  max:                11.85
  95th percentile:     0.59
  sum:                19962.13

Threads fairness:
  events (avg/stddev): 53916.5000/656.50
  execution time (avg/stddev): 9.9811/0.00
root@319cabda9bb0:/#
```

## 2. Test case with 2 CPU and 4 RAM

### Test Case 1:

```
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
prepare
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
run
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
cleanup
```

```
root@43649c9038ed:/# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        18121.57
  fsyncs/s:        23218.30

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   283.15

General statistics:
  total time:           10.0033s
  total number of events: 413335

Latency (ms):
  min:                 0.00
  avg:                 0.05
  max:                25.42
  95th percentile:     0.04
  sum:                19893.59

Threads fairness:
  events (avg/stddev): 206667.5000/91.50
  execution time (avg/stddev): 9.9468/0.00
root@43649c9038ed:/#
```

**Test Case 2:**

```
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr
prepare
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr
cleanup
```

```
root@43649c9038ed:/# sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000 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: 200000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 29.46

General statistics:
total time: 5.0212s
total number of events: 148

Latency (ms):
min: 33.54
avg: 33.92
max: 68.17
95th percentile: 33.72
sum: 5020.49

Threads fairness:
events (avg/stddev): 148.0000/0.00
execution time (avg/stddev): 5.0205/0.00

root@43649c9038ed:/#
```

**3. Test case with 4 CPU and 2 RAM**

**Test Case 1:**

```
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
prepare
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
run
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
cleanup
```

## Sourabh Deshmukh

W1648445

```
root@d14f158f931f:/# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        18006.95
  fsyncs/s:       23072.58

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:  281.36

General statistics:
  total time:           10.0061s
  total number of events: 410837

Latency (ms):
  min:                 0.01
  avg:                 0.05
  max:                15.29
  95th percentile:     0.04
  sum:            19902.33

Threads fairness:
  events (avg/stddev): 205418.5000/2005.50
  execution time (avg/stddev): 9.9512/0.00
root@d14f158f931f:/#
```

### Test Case 2:

```
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr
prepare
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr
cleanup
```

```
root@d14f158f931f:/# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB 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:        4731.51
  fsyncs/s:       6077.59

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:  73.93

General statistics:
  total time:           10.0145s
  total number of events: 108029

Latency (ms):
  min:                 0.00
  avg:                 0.18
  max:                11.86
  95th percentile:     0.59
  sum:            19961.29

Threads fairness:
  events (avg/stddev): 54014.5000/497.50
  execution time (avg/stddev): 9.9806/0.00
root@d14f158f931f:/#
```

#### 4. Test case with 4 CPU and 4 RAM

##### Test Case 1:

```
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
prepare
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
run
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr
cleanup
```

```
root@804d2a75d4a0:/# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB 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:        4732.58
  fsyncs/s:       6070.86

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   73.95

General statistics:
  total time:           10.0328s
  total number of events: 108176

Latency (ms):
  min:                 0.00
  avg:                 0.18
  max:                12.47
  95th percentile:     0.59
  sum:            19962.44

Threads fairness:
  events (avg/stddev): 54088.0000/183.00
  execution time (avg/stddev): 9.9812/0.00

root@804d2a75d4a0:/# █
```

##### Test Case 2:

```
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr
prepare
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr run
docker$ sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=rndwr
cleanup
```

```
root@164c69f49ba6:/# sysbench --threads=2 fileio --file-total-size=1G --file-test-mode=seqrewr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 8MiB each
1GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential rewrite test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        18344.69
  fsyncs/s:       23502.98

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   286.64

General statistics:
  total time:      10.0055s
  total number of events: 418570

Latency (ms):
  min:              0.00
  avg:              0.05
  max:             16.78
  95th percentile:  0.04
  sum:            19902.25

Threads fairness:
  events (avg/stddev): 209285.0000/457.00
  execution time (avg/stddev): 9.9511/0.00

root@164c69f49ba6:/# █
```

# Experiment Observations & Analysis

## CPU Test - QEMU VM qcow2 vs Qemu VM raw vs. Docker Container -

To conduct the CPU Test, I've considered the following three primary parameters:

1. **--threads** Specifies the number of threads utilized for the test.
2. **--cpu-max-prime** Determines the maximum number up to which the tested numbers are checked for primality.
3. **--time** Sets the maximum allowable time for the process to complete.

By modifying the parameters mentioned above, two different CPU Tests are executed, with the CPU and memory combinations, and the corresponding results are presented in the respective test tables, showcasing various metrics.

Test 1 -

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=20000
run
```

	cpu	memory	min	max	avg	events/sec (std)	No of events
qcow2	2	2	3,81	20.03	4.60	11.629992	1082
raw	2	2	4.66	6.88	4.97	17.465265	1004
docker	2	2	1.31	5.36	1.40	18.886975	3578
qcow2	2	4	3.84	19.53	4.18	10.23469	1193
raw	2	4	3.91	8.56	4.21	6.152221	1184
docker	2	4	1.34	1.88	1.38	5.792899	3612
qcow2	4	2	3.90	12,98	4,32	4.427414	1154
raw	4	2	3.88	8.70	4.16	14.375037	1198
docker	4	2	1.32	4.65	1.41	16.037932	3535
qcow2	4	4	3.82	9.82	4,27	12.386788	1168
raw	4	4	3.80	13.20	4.76	14.520588	1047
docker	4	4	1.31	2.31	1.35	4.194768	3705

Test 2 -

```
sourabhdeshmukh$ sysbench cpu --threads=1 --time=5 --cpu-max-prime=200000
run
```

	cpu	memory	min	max	avg	events/sec (std)	No of events
qcow2	2	2	88.37	141.10	100.96	0.738329	50
raw	2	2	90.02	117.1	95.29	0.485304	53
docker	2	2	33.49	68.04	33.91	0.715486	148
qcow2	2	4	90.49	141.66	105.91	0.850482	48
raw	2	4	88.05	142.8	94.93	0.242755	53
docker	2	4	33.54	68.17	33.92	0.855950	148
qcow2	4	2	90.65	161.59	114.66	0.788562	44
raw	4	2	91.15	134.23	106.95	0.567864	47
docker	4	2	33.62	67.82	38.86	1.310584	129
qcow2	4	4	90.78	157.87	104.07	0.431103	48
raw	4	4	90.29	188.74	95.60	0.254067	53
docker	4	4	33.62	67.51	33.99	0.130576	148

**FILEIO Test - QEMU VM qcow2 vs Qemu VM raw vs. Docker Container -**

For conducting the FILE-IO Test, I have considered the following three primary parameters:

1. **--threads** Specifies the number of threads to be utilized for the test.
2. **--file-total-size** Indicates the total size of the file or files to be generated.
3. **--file-test-mode** Dictates the mode of the file test, with five available modes of file I/O:
  - a. rndrd
  - b. rndrw
  - c. rndwr
  - d. seqrd
  - e. seqrewr
  - f. Seqwr

By modifying the parameters mentioned above, two different File-IO Tests are executed, with the CPU and memory combinations, and the corresponding results are presented in the respective test tables, showcasing various metrics.

**Test 1**

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=seqrewr prepare
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=seqrewr run
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=seqrewr cleanup
```

	cpu	memory	min	max	avg	Writes/s(std)	No of events
qcow2	2	2	0.03	53.10	1.11	60.24309	17693
raw	2	2	0.03	46.09	0.34	60.289732	57394
docker	2	2	0.00	17.97	0.05	68.491642	415290
qcow2	2	4	0.03	39.63	0.37	99.072773	51445
raw	2	4	0.06	28.04	0.41	141.42167	47576
docker	2	4	0.00	25.42	0.05	5349.8153	413335
qcow2	4	2	0.03	31.29	0.54	41.645894	35933
raw	4	2	0.03	29.56	0.51	54.521047	37763
docker	4	2	0.01	15.29	0.05	101.51108	410837
qcow2	4	4	0.03	37.31	0.55	35.029638	35246
raw	4	4	0.03	53.68	0.50	94.159135	38438
docker	4	4	0.00	12.47	0.18	5.7197447	108176

**Test 2**

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=rndwr prepare
```

Sourabh Deshmukh

W1648445

```
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr run  
sourabhdeshmukh$ sysbench --threads=2 fileio --file-total-size=1G  
--file-test-mode=rndwr cleanup
```

	cpu	memory	min	max	avg	Writes/s(std)	No of events
qcow2	2	2	0.03	29.76	0.34	196.87927	56058
raw	2	2	0.03	12.55	1.00	15.844762	19561
docker	2	2	0.00	11.85	0.19	16.028662	107833
qcow2	2	4	0.03	18.37	0.94	20.377348	20879
raw	2	4	0.03	23.41	0.99	22.642533	19736
docker	2	4	0.00	13.99	0.18	14.329736	107977
qcow2	4	2	0.03	39.12	1.22	12.421953	16140
raw	4	2	0.03	12.86	1.20	19.012075	16324
docker	4	2	0.00	11.86	0.18	28.238823	108029
qcow2	4	4	0.03	38.13	1.60	237.31051	12210
raw	4	4	0.03	11.79	1.23	17.094678	15933
docker	4	4	0.00	16.78	0.05	130.44531	418570

### MEMORY Test - QEMU VM qcow2 vs Qemu VM raw vs. Docker Container -

To conduct the Memory Test, I've considered the following three primary parameters:

1. **--threads** Specifies the number of threads utilized for the test.
2. **--memory-block-size** the size of memory blocks utilized for the benchmarking process.
3. **--time** Sets the maximum allowable time for the process to complete.
4. **--memory-scope** It defines the specific memory area or conditions under consideration during the sysbench testing.

By modifying the parameters mentioned above, two different Memory Tests are executed, with the CPU and memory combinations, and the corresponding results are presented in the respective test tables, showcasing various metrics.

Test 1

```
sourabhdeshmukh$ sysbench memory --memory-block-size=1G  
--memory-scope=global --threads=2 --time=10 run
```

	cpu	memory	min	max	avg	operation/sec (std)	No of Operations
qcow2	2	2	276.90	613.57	401.21	0.092087	51
raw	2	2	282.82	569.29	400.94	0.238013	50
docker	2	2	169.70	255.20	200.38	1.229728	100
qcow2	2	4	295.88	531.35	399.57	0.096591	51
raw	2	4	288.04	573.86	406.43	0.142021	50
docker	2	4	177.91	280.80	205.05	0.887339	98
qcow2	4	2	348.96	493.43	378.52	0.296176	54
raw	4	2	342.71	457.37	374.69	0.093968	54
docker	4	2	172.20	269.40	199.72	0.645035	100
qcow2	4	4	357.84	508.71	395.69	0.188600	51
raw	4	4	348.50	479.07	382.23	0.146867	53
docker	4	4	173.82	236.16	192.65	0.705584	100

Test 2

```
sourabhdeshmukh$ sysbench memory --memory-block-size=2G  
--memory-scope=global --threads=2 --time=10 run
```

	cpu	memory	min	max	avg	operation/sec (std)	No of Operations
qcow2	2	2	131199.93	131200.52	131200.22	0	2

**Sourabh Deshmukh**

**W1648445**

raw	2	2	123910.17	123912.06	123911.11	0	2
docker	2	2	351.46	502.88	364.88	0.334544	50
qcow2	2	4	761.87	1167.19	882.87	0.068702	23
raw	2	4	797.52	1339.99	993.39	0.153525	21
docker	2	4	353.63	595.55	396.39	0.621948	50
qcow2	4	2	187985.29	187986.53	187985.91	0	2
raw	4	2	178912.18	178912.20	178912.19	0	2
docker	4	2	351.73	450.70	607.30	0.456695	46
qcow2	4	4	698.96	976.26	737.62	0.150433	28
raw	4	4	718.60	1087.45	785.94	0.057271	26
docker	4	4	388.55	541.58	408.88	0.316496	50

# Sysbench Findings and Conclusion based on the tests.

The analysis above relies on two distinct test cases for each CPU, File-IO, and Memory test, comprised of various CPU and RAM configurations.

Furthermore, each test is conducted five times to assess the consistency of results across different cases.

Following are some of the conclusions that can be drawn from the above test results obtained from sysbench tests -

## Conclusions -

### CPU -

1. Docker consistently performs well in terms of CPU usage across various configurations.
2. Raw storage in QEMU and qcow2 have similar trends, with qcow2 showing slightly higher CPU usage.
3. The number of events per second is higher for Docker, indicating better event processing efficiency.
4. Increasing the number of CPUs generally leads to increased CPU usage, while memory configurations show varying impacts on performance.
5. Docker appears to provide better efficiency in terms of CPU usage and event processing compared to QEMU with qcow2 or raw vm, with raw vm being more stable than qcow2.

### Memory -

1. Comparing qcow2 VM and raw VM the performance findings were moderate CPU and memory usage, Stable operation times, and consistent number of operations.
2. According to observations, docker is generally superior in terms of CPU and memory efficiency. Along with that, it has lower operation times and higher throughput, showcasing efficient containerized execution.
3. The above observations depict that Docker provides a more efficient and scalable environment for the given memory test scenarios compared to QEMU VMs with qcow2 and raw disk formats.

### FileIO -

1. Raw VM tends to have higher maximum writes per second, but Docker may outperform it in terms of average and standard deviation for certain configurations.
2. qcow2 often exhibits a higher standard deviation, indicating more variability in performance.
3. Docker consistently shows lower CPU usage and tends to have lower standard deviation, making it a more stable option in terms of performance consistency.
4. Raw VM performance is competitive, with a balance between average writes per second and standard deviation.

## Shell Script Used For Experimentation.

The Following shell script performs the tests for CPU, Memory, and FileIO. The specific sysbench commands make use of shell script arrays to get the parameters required for running the sysbench test commands.

The script performs 2 test cases for CPU, Memory, and FileIO and for each test case, it performs 5 test runs also the results obtained from the test runs are stored in /tmp directory.

Shell Script for CPU, Memory and FileIO sysbench tests.

```
sourabhdeshmukh$ ./sysbench.sh
```

```
#!/bin/bash
```

```
usage=$(basename "$0") [-h] -- The following program performs the sysbench test cases for the CPU, MEMORY and FileIO. It currently supports 2 tests cases for each with 5 test runs for individual test case.
```

```
where:
```

```
-h show this help text"
```

```
while getopts ':h:' option; do
  case "$option" in
    h) echo "\n$usage"
        exit
        ;;
    :) printf "\nmissing argument for -%s !!!\n\n" "$OPTARG" >&2
       echo "$usage" >&2
       exit 1
       ;;
    \?) printf "\nillegal option: -%s !!!\n\n" "$OPTARG" >&2
       echo "$usage" >&2
       exit 1
       ;;
  esac
done
shift $((OPTIND - 1))

PRIMES_UPTO=( "20000" "200000" )
```

```
TEST_MODE=("rndwr" "seqrewr")
TEST_RUNS=5
TEST_CASES=2
LogFile="/tmp/sysbench.log"
CPULogFile="/tmp/cpu_sysbench.log"
FILEIOLogFile="/tmp/fileio_sysbench.log"
MEMORYLogFile="/tmp/memory_sysbench.log"

for ((i=0; i<$TEST_CASES;i++))
do
    echo `date` "[INFO] Starting ${i+1} Test Case." | tee -a $LogFile
    for (( j=1; j <=$TEST_RUNS; j++ ))
    do
        echo `date` "[INFO] Running ${j} run of CPU Test Case ${i+1}" |
tee -a $CPULogFile
        sysbench cpu --threads=1 --cpu-max-prime=${PRIMES_UPTO[$i]} --time=5
run | tee -a $CPULogFile

        if [ $? -ne 0 ]; then
            echo `date` "[ERROR] Error occurred while performing ${i+1} CPU test
case." | tee -a $CPULogFile
            exit 1
        else
            echo `date` "[INFO] Completed ${j} run of CPU Test Case
${i+1}" | tee -a $CPULogFile
        fi

        echo `date` "[INFO] Running ${j} run of FILEIO Test Case ${i+1}" | tee
-a $FILEIOLogFile
        sudo sync; echo 3 > /proc/sys/vm/dropcaches
        sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=${TEST_MODE[$i]} prepare | tee -a $FILEIOLogFile
        sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=${TEST_MODE[$i]} run | tee -a $FILEIOLogFile
        sysbench --threads=2 fileio --file-total-size=1G
--file-test-mode=${TEST_MODE[$i]} cleanup | tee -a $FILEIOLogFile

        if [ $? -ne 0 ]; then
            echo `date` "[ERROR] Error occurred while performing ${i+1} FILEIO
test case." | tee -a $FILEIOLogFile
            exit 1
        else
            echo `date` "[INFO] Completed ${j} run of FILEIO Test Case
```

```
${{i+1}}" | tee -a $FILEIOLogFile
fi

sysbench memory --memory-block-size=${MEMBLOCK[$i]}
--memory-scope=global --threads=2 --time=10 run | tee -a $MEMORYLogFile

if [ $? -ne 0 ]; then
    echo `date` "[ERROR] Error occurred while performing ${i+1} Memory
test case." | tee -a $MEMORYLogFile
    exit 1
else
    echo `date` "[INFO] Completed ${j} run of Memory Test Case
${{i+1}}" | tee -a $MEMORYLogFile
fi

done
echo `date` "[INFO] Completed ${i+1} Test Case." | tee -a $LogFile
echo "" | tee -a $LogFile
done
```

# Automation Scripts

## Vagrantfile used for qemu virtual machines.

### Vagrantfile -

```
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented
  # below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search
  # for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "aspyatkin/ubuntu-20.04-server"

  config.vm.provider "virtualbox" do |server|
    server.cpus = 2
    server.memory = "2048"
  end

  config.vm.provision "shell", path: "userdata.sh"
  config.vm.synced_folder "./scripts", "/sysbench"
end
```

Sourabh Deshmukh  
W1648445

## Dockerfile that is used for Docker sysbench image creation.

### Dockerfile -

```
FROM sourabhdeshmukh/ubuntu-sysbench:v1
MAINTAINER Sourabh Deshmukh <sourabh.deshmukh.988@gmail.com>
WORKDIR /app
COPY ./scripts/* /app/
RUN chmod +x /app/sysbench.sh
CMD [ "./sysbench.sh" ]
```

The Dockerfile uses sourabhdeshmukh/ubuntu-sysbench:v1 as a base image that we created earlier.

## Resources

- Github Repository for Homework Assignment 1 -  
<https://github.com/sourabhdeshmukh/Cloud-CSEN-241/tree/main/Assignments/HW1>
- Custom Sysbench Docker Image created during the experiment -  
<https://hub.docker.com/r/sourabhdeshmukh/ubuntu-sysbench>
- Dockerfile  
<https://github.com/sourabhdeshmukh/Cloud-CSEN-241/blob/main/Assignments/HW1/docker/Dockerfile>
- Vagrantfile  
<https://github.com/sourabhdeshmukh/Cloud-CSEN-241/blob/main/Assignments/HW1/vagrant/Vagrantfile>
- Sysbench Test Shell Script  
<https://github.com/sourabhdeshmukh/Cloud-CSEN-241/blob/main/Assignments/HW1/scripts/sysbench.sh>
- Sysbench Installation Reference  
<https://lindevs.com/install-sysbench-on-ubuntu>
- Dockerfile Reference  
<https://docs.docker.com/engine/reference/builder/>
- Luks Encryption Reference  
[https://bugzilla.redhat.com/show\\_bug.cgi?id=1406803](https://bugzilla.redhat.com/show_bug.cgi?id=1406803)  
<https://www.qemu.org/docs/master/system/secrets.html>
- Results and Proofs of the Experiments conducted.  
<https://github.com/sourabhdeshmukh/Cloud-CSEN-241/tree/main/Assignments/HW1/img>
  - CPU  
<https://github.com/sourabhdeshmukh/Cloud-CSEN-241/tree/main/Assignments/HW1/img/CPU>
  - Memory  
<https://github.com/sourabhdeshmukh/Cloud-CSEN-241/tree/main/Assignments/HW1/img/memory>
  - FileIO  
<https://github.com/sourabhdeshmukh/Cloud-CSEN-241/tree/main/Assignments/HW1/img/File-io>