

**VIDHI KAMDAR**

**B00659440**

**PROJECT-1**

## **LIGHTWEIGHT VS. HEAVYWEIGHT VIRTUALIZATION TECHNIQUES**

**September 21, 2017**

### **REPORT**

#### **Detailed configurations of Google Cloud instance setup**

Referred to: [http://www.cs.binghamton.edu/~huilu/slides/How\\_to\\_use\\_google\\_cloud.pdf](http://www.cs.binghamton.edu/~huilu/slides/How_to_use_google_cloud.pdf)

##### **Apply coupon code**

- Click the **Student Coupon Retrieval link** and retrieve it using your name and BU email
- Sign in into your Gmail account which you want to use for creating cloud instance
- Once you receive the coupon code, go to <https://console.cloud.google.com/education?code=>
- Make sure you are signed in to your Gmail account
- Apply the coupon code and follow the instructions
- You'll see **\$50 credit** in the Billing panel

##### **Create Google Instance**

- Go to VM instances page <https://console.cloud.google.com/compute/instances>
- Select existing project (**\*My First Project Project**)
- Click on Create instance(**\*neutron**)
- Under Machine type, select **2 vCPU** and increase the memory size from 3.75 GB to **4 GB**
- In Boot disk, change to begin **configuring boot disk**
- Under OS images, select **UBUNTU 16.04 LTS** image
- 20 GB memory
- Create

##### **Remotely Connect to your Instance**

- Using **\*Browser**, open VM instances page and click on **SSH**

##### **Stop your instance**

- Type **exit** to exit from your SSH
- Select your instance, click **STOP**

## Steps to enable Docker container and commands

Referred to: <https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/>  
<https://github.com/docker/labs/blob/master/beginner/chapters/alpine.md>

### OS Configuration

- Xenial 16.04
- X86\_64

### Install Docker CE using Repository

- **Update apt** package: `sudo apt-get update`
- Install packages:  
`sudo apt-get install apt-transport-https ca-certificates curl software-properties-common`
- Add **Docker's GPG key**:  
`curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`
- **Verify** the fingerprint: 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88:  
`sudo apt-key fingerprint 0EBFCD88`
- Setup **stable repository**:  
`sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"`
- To install Docker CE, first **update apt package**: `sudo apt-get update`
- Install the **latest version** of Docker: `sudo apt-get install docker-ce`
- **Check if it installed** correctly: `sudo docker run hello-world`

### Docker commands:

- **List all Docker images**: `sudo docker images`
- **Pull** will fetch the image from Docker registry and save to your system:  
`sudo docker pull name_of_image`
- **Run Docker container** based on the image: `docker run name_of_image command`
- **Run shell commands**: `docker run -it name_of_image command`

## Steps to enable QEMU VM, commands and configuration

Referred to: <https://qemu.weilnetz.de/doc/qemu-doc.html>

<https://medium.com/google-cloud/graphical-user-interface-gui-for-google-compute-engine-instance-78fccda09e5c>

<https://cloud.google.com/compute/docs/vpc/firewalls>

### Steps to install QEMU VM

- From your SSH cloud instance, **install QEMU**: `sudo apt-get install qemu`
- Download **ubuntu image**: `wget http://mirror.pnl.gov/releases/16.04/ubuntu-16.04.3-server-amd64.iso`
- Create image for the VM: `sudo qemu-img create ubuntu.img 10G`
- For the installation, we need graphical user interface.
  - **Update your source list**: `$ sudo apt-get update`
  - **Install Gnome** componenets: `sudo apt-get install gnome-core`
  - For virtual desktop installation, we need VNC server. To **install VNC server**: `sudo apt-get install vnc4server`
  - Start vnc4server and setup password: **vncserver**
  - **Open xstartup** file: `vim .vnc/xstartup`
  - Press I and **Replace** entire content with the following:

```
#!/bin/sh
def
export XKL_XMODMAP_DISABLE=1
unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
metacity &
gnome-settings-daemon &
gnome-panel &
nautilus &
gnome-terminal &
```
  - Create **new Fire-Wall rule** from Google VM instances page.
    - Create new rule
    - Name: **vnc-server**
    - Priority: **1000**
    - Direction: **Ingress**
    - Action on match: **Allow**
    - Source filters: **IP Ranges 0.0.0.0/0**
    - Protocol and ports: **tcp:5000-10000**
  - Download VNC Viewer: <https://www.realvnc.com/en/connect/download/viewer/>
  - Start the server from ssh cloud instance: `vncserver`
  - **Connect to the instance** using external IP: `nc 35.190.166.181 5901`
  - **Start VNCViewer**. Make new connection using external IP and VNC password.  
**Connect**
  - This shows virtual cloud desktop.

- To **stop the connection, disconnect from VNC Viewer** and **kill the vncserver** instance from SSH: `vncserver -kill :1`
- From your GUI, right click and **open terminal**.  
To **install QEMU VM** type: `sudo qemu-system-x86_64 -hda ubuntu.img -boot d -cdrom ./ubuntu-16.04.3-server-amd64.iso -m 1536`
  - **Ubuntu steps and configuration**
  - Ubuntu System Name: vidhi
  - User name: vidhi kamdar
  - Language settings: English(US)
  - Keyboard settings: English(US)
  - Disk partition: Entire Disk
  - Setting http proxy: None
  - Install system: Normal

## Conduct measurements and tests in native (Google Cloud), Docker and QEMU

### Native (Google Cloud):

For the native sysbench, below are the benchmarks for CPU and I/O utilization.

```
vdkamdar006@neutron: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/alien-proton-178902/zones/us-east1-d/instances/neutron?authuser=0&hl=en_US&projectNumber=822755468742
vdkamdar006@neutron:~$ sudo sysbench --test=cpu --cpu-max-prime=20000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 20000

Test execution summary:
total time:                27.9743s
total number of events:    10000
total time taken by event execution: 27.9727
per-request statistics:
  min:                     2.76ms
  avg:                     2.80ms
  max:                     5.80ms
  approx. 95 percentile:   2.86ms

Threads fairness:
  events (avg/stddev):    10000.0000/0.00
  execution time (avg/stddev): 27.9727/0.00

vdkamdar006@neutron:~$
```

```
vdkamdar006@neutron: ~ - Google Chrome
Secure | https://ssh.cloud.google.com/projects/alien-proton-178902/zones/us-east1-d/instances/neutron?authuser=0&hl=en_US&projectNumber=822755468742
Creating files for the test...
vdkamdar006@neutron:~$ sudo sysbench --num-threads=16 --test=fileio --file-total-size=10G --file-test-mode=rndrw run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 80Mb each
10Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
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 r/w test
Threads started!
Done.

Operations performed: 6014 Read, 3999 Write, 12800 Other = 22813 Total
Read 93.969Mb Written 62.484Mb Total transferred 156.45Mb (17.573Mb/sec)
1124.65 Requests/sec executed

Test execution summary:
total time:                8.9032s
total number of events:    10013
total time taken by event execution: 101.7313
per-request statistics:
  min:                     0.01ms
  avg:                     10.16ms
  max:                     98.29ms
  approx. 95 percentile:   33.98ms

Threads fairness:
  events (avg/stddev):    625.8125/32.88
  execution time (avg/stddev): 6.3582/0.10

vdkamdar006@neutron:~$ sudo sysbench --num-threads=16 --test=fileio --file-total-size=10G --file-test-mode=rndrw cleanup
sysbench 0.4.12: multi-threaded system evaluation benchmark

Removing test files...
vdkamdar006@neutron:~$ echo 3 | sudo tee /proc/sys/vm/drop_caches
3
vdkamdar006@neutron:~$
```

TEST: MAXPRIME

MAX PRIME NUMBER=20000

MODE: NATIVE	TOTAL TIME (seconds)	THREADS
TEST-1	27.9529	1
TEST-2	27.9315	1
TEST-3	27.9743	1

TEST: FILE I/O

TOTAL FILE SIZE = 10G

MODE: NATIVE	TOTAL TIME (seconds)	THREADS
TEST-1	8.9032	16
TEST-2	8.9127	16
TEST-3	8.8992	16

TEST: FILE I/O

TOTAL FILE SIZE = 10G

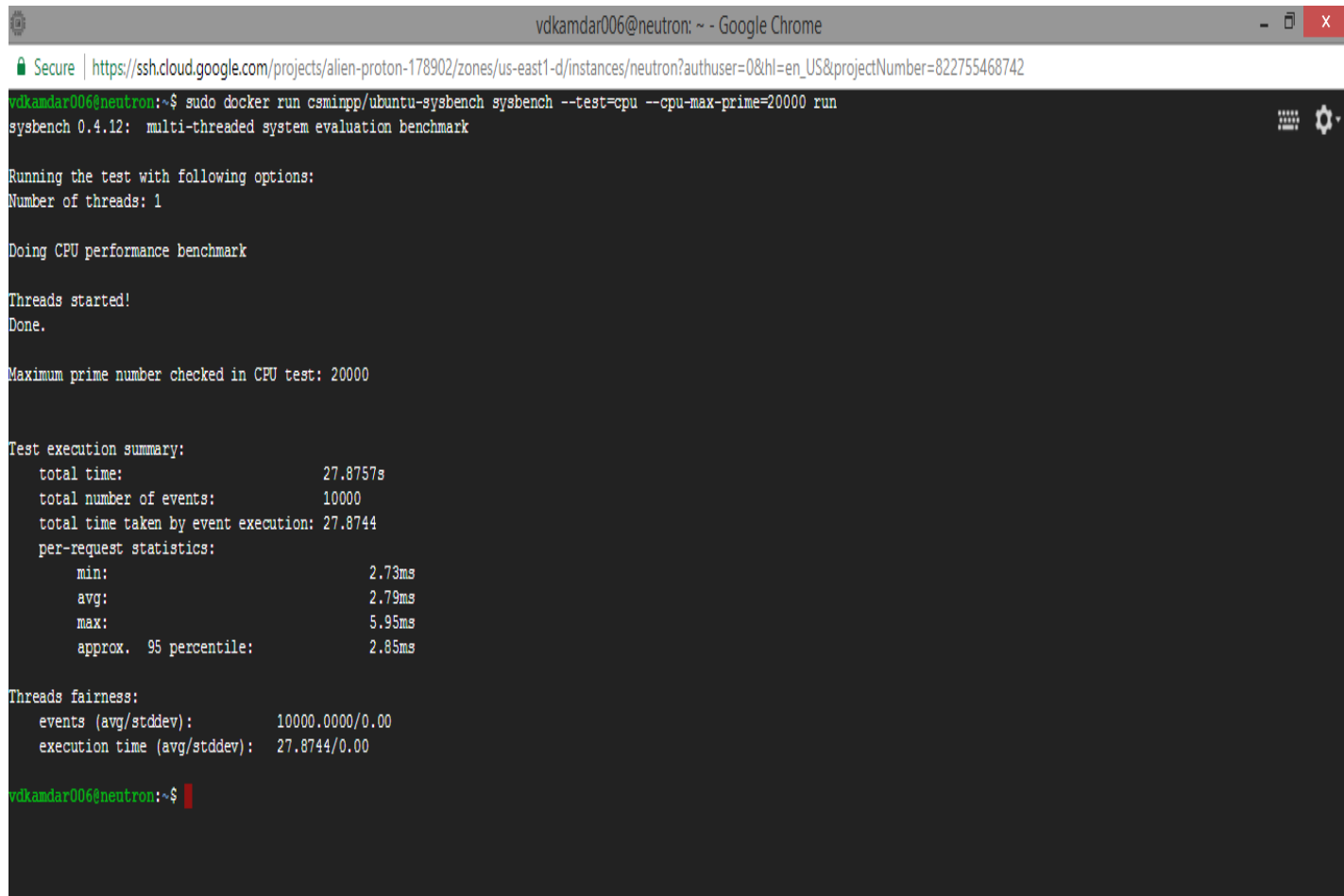
MODE: NATIVE	TOTAL TIME (seconds)	THREADS
TEST-1	41.9257	2
TEST-2	41.9182	2
TEST-3	41.9179	2

TEST: FILE I/O

TOTAL FILE SIZE = 10G

MODE: NATIVE	TOTAL TIME (seconds)	THREADS
TEST-1	69.5195	1
TEST-2	69.5072	1
TEST-3	69.5059	1

## Docker:



The screenshot shows a terminal window titled "vdkamdar006@neutron: ~ - Google Chrome". The address bar shows a secure SSH connection to a Google Cloud instance. The terminal content shows the execution of a Docker command to run sysbench in a container. The output indicates that sysbench 0.4.12 is being used for a multi-threaded system evaluation benchmark. The test is configured with --test=cpu and --cpu-max-prime=20000. The execution shows the test running with 1 thread, performing a CPU performance benchmark. The test execution summary shows a total time of 27.8757s, 10000 events, and a total time taken by event execution of 27.8744s. The per-request statistics show a minimum of 2.73ms, an average of 2.79ms, a maximum of 5.95ms, and an approximate 95th percentile of 2.85ms. The threads fairness section shows 10000.0000/0.00 events and 27.8744/0.00 execution time.

```
vdkamdar006@neutron:~$ sudo docker run csmirpp/ubuntu-sysbench sysbench --test=cpu --cpu-max-prime=20000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 20000

Test execution summary:
total time:                27.8757s
total number of events:    10000
total time taken by event execution: 27.8744
per-request statistics:
  min:                     2.73ms
  avg:                     2.79ms
  max:                     5.95ms
  approx. 95 percentile:   2.85ms

Threads fairness:
events (avg/stddev):       10000.0000/0.00
execution time (avg/stddev): 27.8744/0.00

vdkamdar006@neutron:~$
```

TEST: MAXPRIME

MAX PRIME NUMBER=20000

MODE: DOCKER	TOTAL TIME (seconds)	THREADS
TEST-1	27.8744	1
TEST-2	27.8367	1
TEST-3	27.8566	1

TEST: FILE I/O

TOTAL FILE SIZE = 10G

MODE: DOCKER	TOTAL TIME (seconds)	THREADS
TEST-1	8.8821	16
TEST-2	8.8796	16
TEST-3	8.8802	16

TEST: FILE I/O

TOTAL FILE SIZE = 10G

MODE: DOCKER	TOTAL TIME (seconds)	THREADS
TEST-1	41.5238	2
TEST-2	41.5189	2
TEST-3	41.5197	2

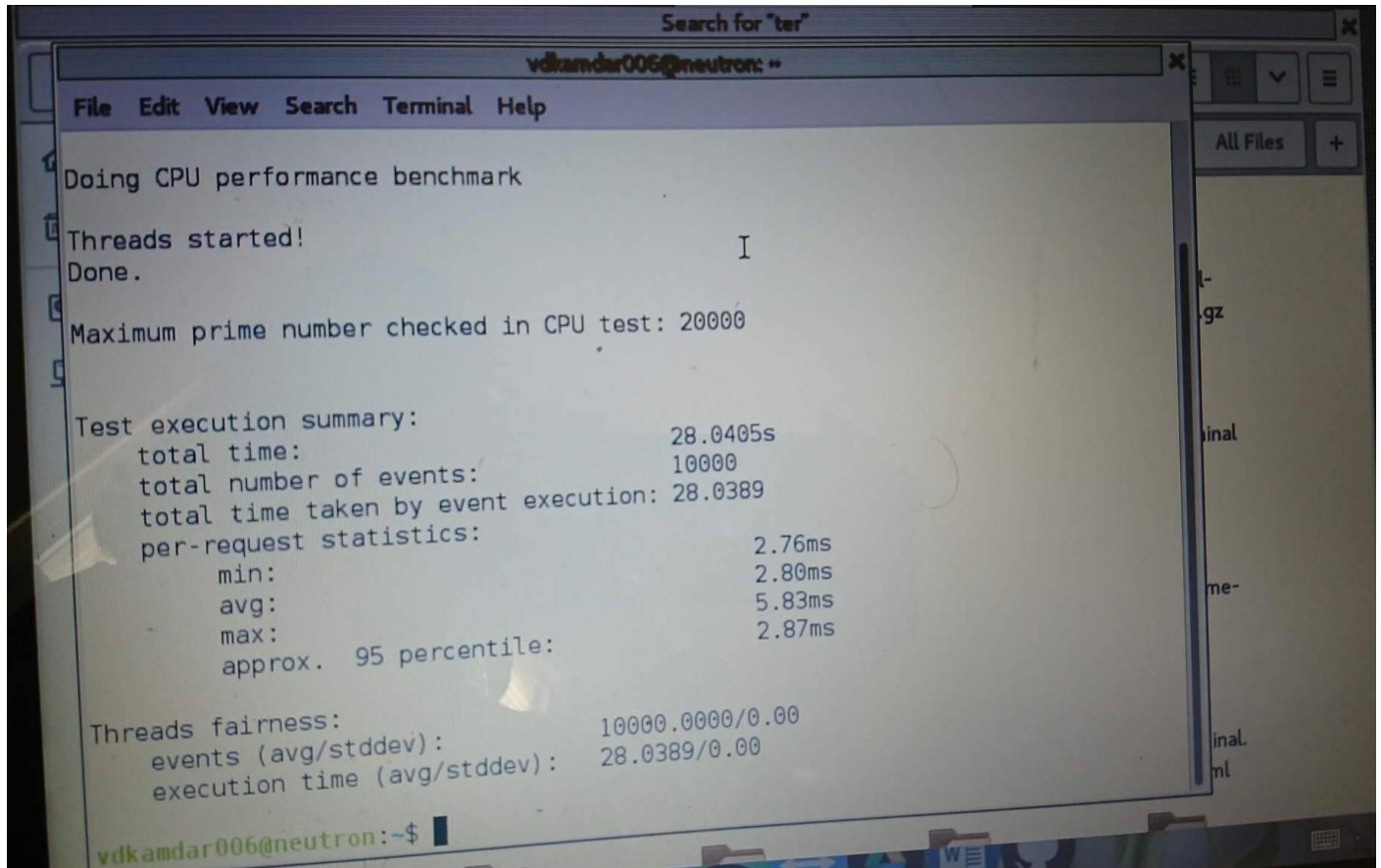
TEST: FILE I/O

TOTAL FILE SIZE = 10G

MODE: DOCKER	TOTAL TIME (seconds)	THREADS
TEST-1	69.5195	1
TEST-2	69.4572	1
TEST-3	69.4459	1



QEMU:

A screenshot of a terminal window titled "Search for 'ter'" with a menu bar (File, Edit, View, Search, Terminal, Help) and a file explorer on the right. The terminal output shows a CPU performance benchmark. It starts with "Doing CPU performance benchmark", followed by "Threads started!" and "Done.". It then reports "Maximum prime number checked in CPU test: 20000". A "Test execution summary:" follows, listing: "total time: 28.0405s", "total number of events: 10000", "total time taken by event execution: 28.0389", and "per-request statistics:" with sub-values for min (2.76ms), avg (2.80ms), max (5.83ms), and approx. 95 percentile (2.87ms). Finally, it shows "Threads fairness:" with "events (avg/stddev): 10000.0000/0.00" and "execution time (avg/stddev): 28.0389/0.00". The prompt "vdkamdar006@neutron:~\$" is visible at the bottom.

```
Search for "ter"
vdkamdar006@neutron: ~
File Edit View Search Terminal Help

Doing CPU performance benchmark
Threads started!
Done.

Maximum prime number checked in CPU test: 20000

Test execution summary:
total time: 28.0405s
total number of events: 10000
total time taken by event execution: 28.0389
per-request statistics:
  min: 2.76ms
  avg: 2.80ms
  max: 5.83ms
  approx. 95 percentile: 2.87ms

Threads fairness:
  events (avg/stddev): 10000.0000/0.00
  execution time (avg/stddev): 28.0389/0.00

vdkamdar006@neutron:~$
```

TEST: MAXPRIME

MAX PRIME NUMBER=20000

MODE: QEMU	TOTAL TIME (seconds)	THREADS
TEST-1	27.9529	1
TEST-2	27.9315	1
TEST-3	27.9743	1

TEST: FILE I/O

TOTAL FILE SIZE = 4G

MODE: QEMU	TOTAL TIME (seconds)	THREADS
TEST-1	14.1564	2
TEST-2	14.1761	2
TEST-3	14.1596	2

TEST: FILE I/O

TOTAL FILE SIZE = 4G

MODE: QEMU	TOTAL TIME (seconds)	THREADS
TEST-1	32.9195	1
TEST-2	32.8452	1
TEST-3	32.8911	1

SOME observations:

- Generally it takes around 28.97896 for CPU to compute max\_prime of 20000.
- Docker and QEMU are light weight virtualization technique and hence runs slightly faster.
- Docker and QEMU are better with limited functionalities and better approaches that we present to them.