Mini-Project 1 – Lightweight vs. Heavyweight Virtualization Techniques

(Due by Midnight, Thursday, Feb. 27, 15% Grade)

---

1 Summary

In this assignment, you will get familiar with two virtualization techniques: one is Docker (a light-weight, container-based virtualization technique), and the other is QEMU (a user-level full virtualization technique). You will need to correctly deploy them, be familiar with the operations, measure the performance using benchmarks, and finally compare the results in a report and demo it to TA.

2 Environment Setup

Please follow this document [1] to start a Google cloud instance. We refer to this instance as *the native system*, but don't use the vCPU and memory configurations in the document. We will configure this instance using **2 vCPU, 4 GB memory, and 20 GB** disk size for this assignment. Please follow the above document [1] to install the corresponding Linux system.

3 Docker

3.1 Installation & Preparation

Please refer to this instruction [2] for Docker installation. Notice that, please install Docker CE (choose using the repository). In addition, you may want to get familiar with Docker basic operations by referring to [3].

3.2 Docker Measurement

Please use the following container image, with Sysbench pre-installed: csminpp/ubuntu-sysbench. You will measure the performance of Docker using Sysbench.

Sysbench is a benchmark tool [4]. We can have some basic ideas about the system performance by running such benchmark tools against the system under test. In this project, we will focus on the cpu and fileio test modes (section 4.1 and 4.5 in [4]) of Sysbench.

Notice that the following instructions present some ideas about how to conduct a meaningful measurement. Please read them carefully, and then produce measurement results (you will lose points, if the results are incorrectly generated and/or can not be well justified).

- You need to figure out all the test cases (under the cpu and fileio test modes).
- For each test case, you need to figure out the"right" parameters. For example, you need to select the parameter for "–cpu-max-prime" to ensure your test case won't end in a short time period (too small) or will never end (too large). For example, you need to decide the file size (e.g., –file-total-size) for Sysbench fileio test mode. Usually, it's reasonable to make a test case lasting for at least 30 seconds. Please justify your configurations.
- You may want to repeat/re-run at least 3 times for each test case, and report the average value of your results (the Sysbench will report some user-level performance data, e.g., total time).

- To reduce test variation, you need to test each case under similar test environment. For example, for the fileio test, after one test, the operating system will cache the accessed files. If we don't manually drop such cache, the following tests will finish much faster, as most I/O data will be served directly form the memory cache instead of disks. You have to manually drop such cache for example using the command: $echo\ 3 > /proc/sys/vm/drop\_caches$ in the native (not the container). (Notice that, you can only run this command under the root user. Please figure out how to enable root user under Ubuntu and switch to the root user to drop the cache before each fileio test).
- In addition to the user-level performance data generated by the benchmark, you need to collect system performance data (from the native system). For example, you can choose to use iostat (a Linux tool for collecting cpu, I/O, network usage [5]). Please do this on your native system (outside containers). You may need to collect performance data covering the whole time period while your benchmark is running. You must provide the cpu and disk I/O performance data (think about how to show this data — suppose you will have a series of performance data points). Please report the commands your use and the interpretation of your performacne related data.
- Please well organize your experimental setup, configurations, and data (using figures and/or tables); and report them in your final report.

4 QEMU

4.1 Installation & Preparation

Under the native system, you can install qemu (under Ubuntu) simply by:
$sudo apt-get install qemu

You will download the Ubuntu iso image to install your QEMU VM:
$wget http://mirror.pnl.gov/releases/16.04/ubuntu-16.04.6-server-amd64.iso

You need to create an image before installing your QEMU VM. For your convenience, we provide the commands below. Please refer to the QEMU manual for more details [6].
$sudo qemu-img create ubuntu.img 10G

4.2 Install a QEMU VM

Because installing a VM require a graphical user interface (GUI). Please refer to Section 7 of this document to enable the GUI of your Google instance (i.e., the native). You will do the following operations within GUI (you should use a VNC client). Please install the QEMU VM using the command below (which takes the iso file as a "cdrom" and the qemu image as a "hard disk"):

$sudo qemu-system-x86_64 -hda ubuntu.img -boot d -cdrom ./ubuntu-16.04.6-server-amd64.iso -m 1536

After you install the OS to the image, next time when you want to start the QEMU VM, you should use the following command (omitting the "cdrom"):

$sudo qemu-system-x86_64 -hda ubuntu.img -m 1536

Then, install the Ubuntu Linux system step by step (it may take some time to complete the whole installation).

4.3 QEMU VM Measurement

Restart your QEMU VM after installation. Install Sysbench workload in your QEMU VM, and re-do all measurements (those in Section 3.2) under the QEMU VM by following all the guidelines.

5 The Native

We refer to your Google cloud instance as the native. You will need to re-do all the measurements (those in Section 3.2 and 4.3) under the native by following all the above guidelines.

6 Submission & Grading

Submit your assignment report on the blackboard as one pdf file. You will also schedule an appointment with TA to have a demo before the deadline. **Notice that it is your responsibility to schedule the demo time. You will lose points if you fail to do so**. We will grade your assignment based this report and the demo. Basically, you should not only follow the instructions to do your assignment, but also fully understand what you are doing.

The report should include (but not limited to):

- Detailed configurations of your experimental setup (your Google Cloud instance). – 5 points
- Present main steps to enable a Docker container. In addition, please describe the operations you use to manage Docker containers (and some other operations which you think are also important). – 10 points
- Present main steps to enable a QEMU VM. In addition, please present the detailed QEMU commands, and VM configurations. – 10 points
- Present how you conduct your performance measurements in three different scenarios (the native, Docker, and QEMU. – 15 points
- Present how you use performance tools to collect performance data. For CPU utilization, you should at least divide them into two parts including user-level and kernel-level. For I/O, you should present I/O throughput, latency, and disk utilization – 15 points
- Please understand the performance data, analyze the data, and then present them in an understandable way in your report. – 15 points
- Writing – 10 points

During the demo, you will be asked to:

- Run your docker container. – 5 points
- Run your QEMU VM instance – 5 points
- Run sysbench inside your docker container and QEMU VM instance – 5 points
- Run your performance measurement tools – 5 points

7 Appendix

Please follow the instructions in this article [7] to enable GUI for the native (your Google instance). Notice that, however, there are two places you need to pay attention, and following the instructions below may solve possible problems.

7.1 .vnc/xstartup

Replace the whole content of .vnc/xstartup with the following content (DO NOT follow the article above).

```
#!/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 &
```

7.2 Open the firewall

Navigate to the "Firewall rules" item (in Google Cloud Console) shown as in Figure 1. Create a new firewall rule as in Figure 2. Notice that though this is an easy way to open ports for vncserver, it is not a safe solution (as we open too many ports). You can choose to use a safer way to do so.

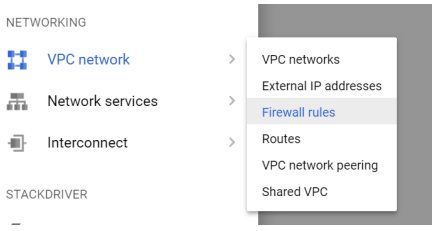

Figure 1: Open firewall rules.

References

[1] Please refer to the Steps-to-use-GCP.pdf which we have uploaded to myCourse.

[2] Docker Installation. `https://docs.docker.com/install/linux/docker-ce/ubuntu/`.

[3] Running your first container. `https://github.com/docker/labs/blob/master/beginner/chapters/alpine.md`.

[4] Sysbench. `http://imysql.com/wp-content/uploads/2014/10/sysbench-manual.pdf`.

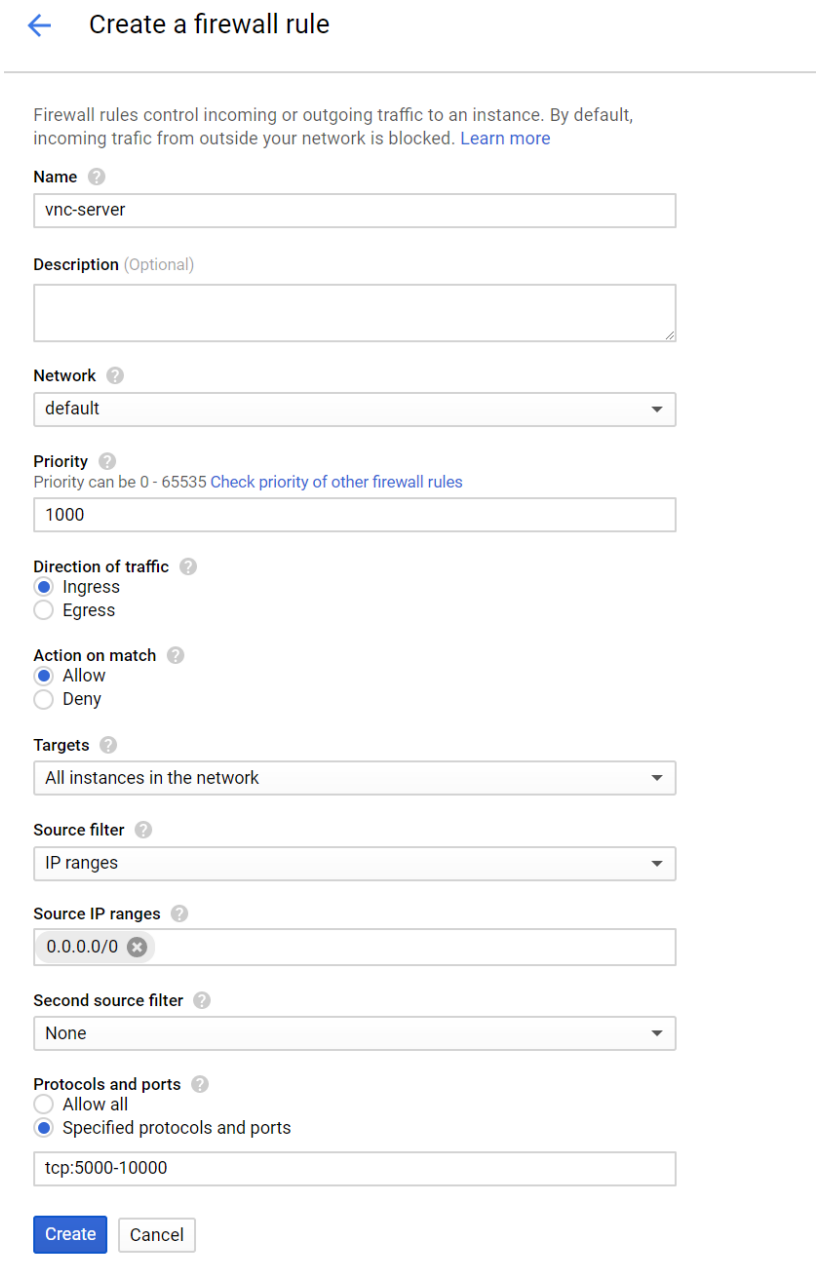

Figure 2: Firewall rules.

[5] IOSTAT. https://linux.die.net/man/1/iostat.

[6] QEMU https://qemu.weilnetz.de/doc/qemu-doc.html.

[7] Graphical user interface (GUI) for Google Compute Engine instance. https://medium.com/google-cloud/graphical-user-interface-gui-for-google-compute-engine-instance-78fccda09e5c.