CMPE283 Assignment 2: KVM Exit Statistics

Team Members

- Sheetal Yadav : https://github.com/SheetalYadav01/linux
- Amrutha

Question 1

Sheetal Yadav took the lead in implementing the project setup. She ensured the Google Cloud Platform (GCP) project was created with nested virtualization enabled, and the outer VM was successfully set up on Google Cloud Platform. She also installed KVM and the necessary development tools, and configured the Linux server.
Amrutha Junnuri focused on researching and supporting the project steps. This included providing guidance and documentation support throughout the process. She also installed and configured the inner VM, ensuring it was set up correctly and ready for testing.
We both worked together in test the process and troubleshooting any configuration or environment-related issues.

Question 2

Development Steps

Step 1: Performed the setup of Google Cloud Platform (GCP) account

1. Created a new project named "nested-virt"
2. Install Google Cloud SDK on local machine
3. Enable Compute Engine API , after installing logged into gcloud

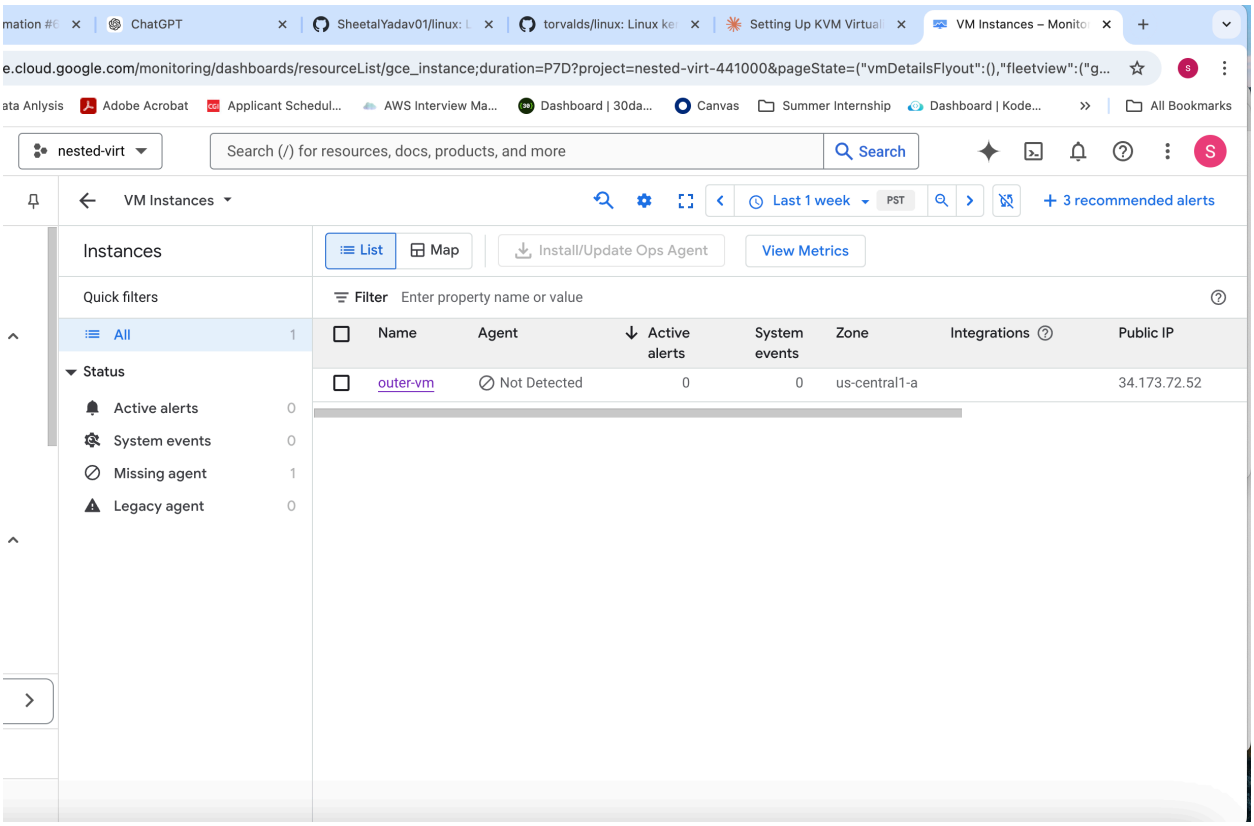
   gcloud config set project nested-virt

Step 2: Created the nested Outer VM in us-central1-a

gcloud compute instances create outer-vm \ --machine-type=e2-standard-4 \ --image-family=ubuntu-2204-lts \ --image-project=ubuntu-os-cloud \ --zone=us-central1-a

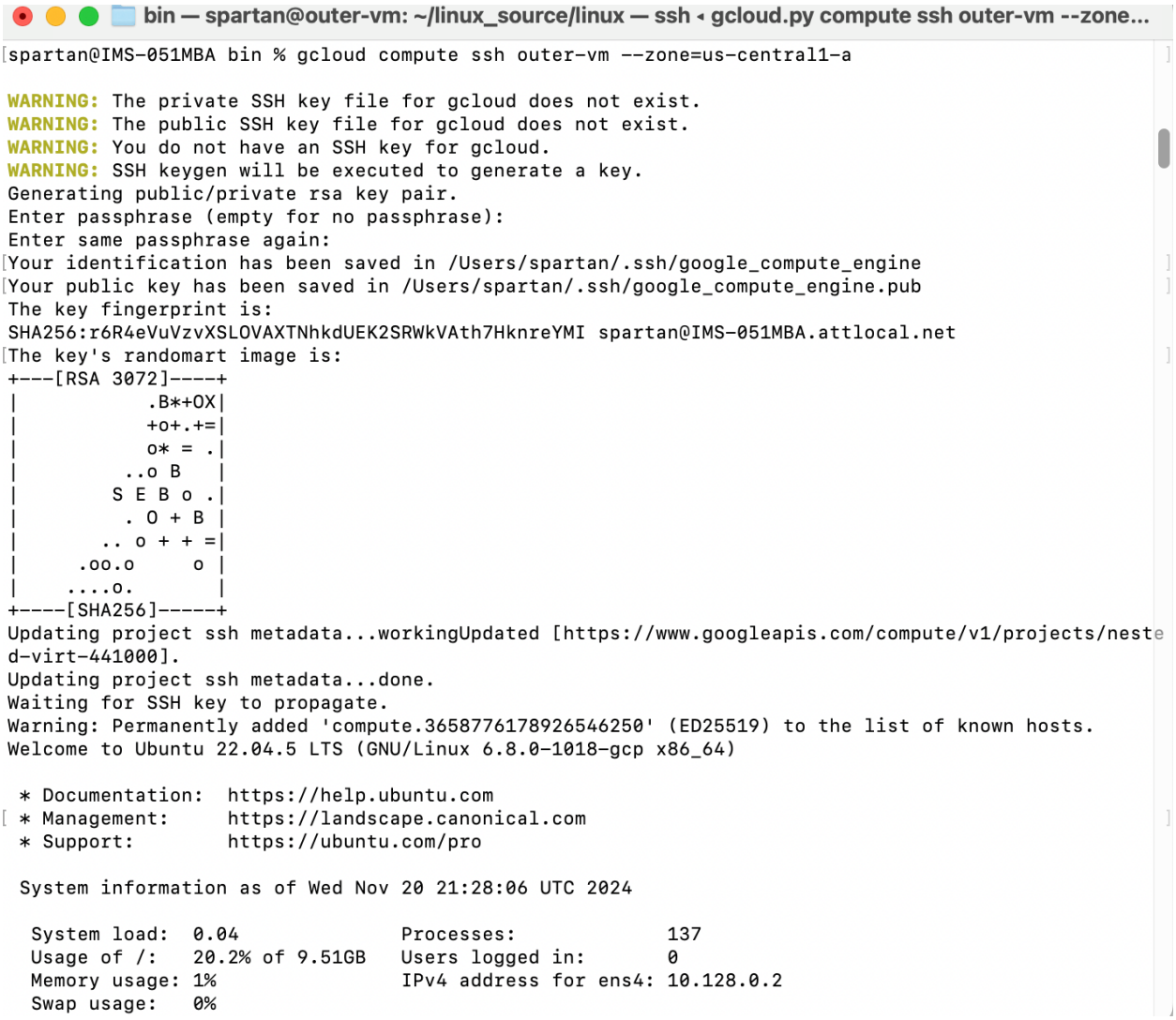
Running outer VM

```
bin — spartan@outer-vm: ~/linux_source/linux — ssh ◂ gcloud.py compute ssh outer-vm --zone...
    --image-project=ubuntu-os-cloud \
    --zone=us-central1-a

API [compute.googleapis.com] not enabled on project [nested-virt-441000]. Would
you like to enable and retry (this will take a few minutes)? (y/N)?  y

Enabling service [compute.googleapis.com] on project [nested-virt-441000]...
Operation "operations/acf.p2-132716371559-36aca101-8b52-4101-bbbf-6e57f630a402" finished successfull
y.
Created [https://www.googleapis.com/compute/v1/projects/nested-virt-441000/zones/us-central1-a/insta
nces/outer-vm].
NAME: outer-vm
ZONE: us-central1-a
MACHINE_TYPE: e2-standard-4
PREEMPTIBLE:
INTERNAL_IP: 10.128.0.2
EXTERNAL_IP: 34.173.72.52
STATUS: RUNNING
spartan@IMS-051MBA bin % gcloud compute instances list

NAME: outer-vm
ZONE: us-central1-a
MACHINE_TYPE: e2-standard-4
PREEMPTIBLE:
INTERNAL_IP: 10.128.0.2
EXTERNAL_IP: 34.173.72.52
STATUS: RUNNING
```



Step 3: Perfromed the SSH into Outer VM

gcloud compute ssh --zone "us-central1-a" "outer-vm" --project "nested-virt-441000"



Step 4: Performed the necessary update and installation of toold on VM Environment to to build the linux kernel

sudo apt update && sudo apt upgrade -y
sudo apt install -y build-essential git fakeroot libncurses5-dev libssl-dev bc bison flex libelf-dev

Step 5: Fork the code into the github repo for Linux kernel build from below mentioned link

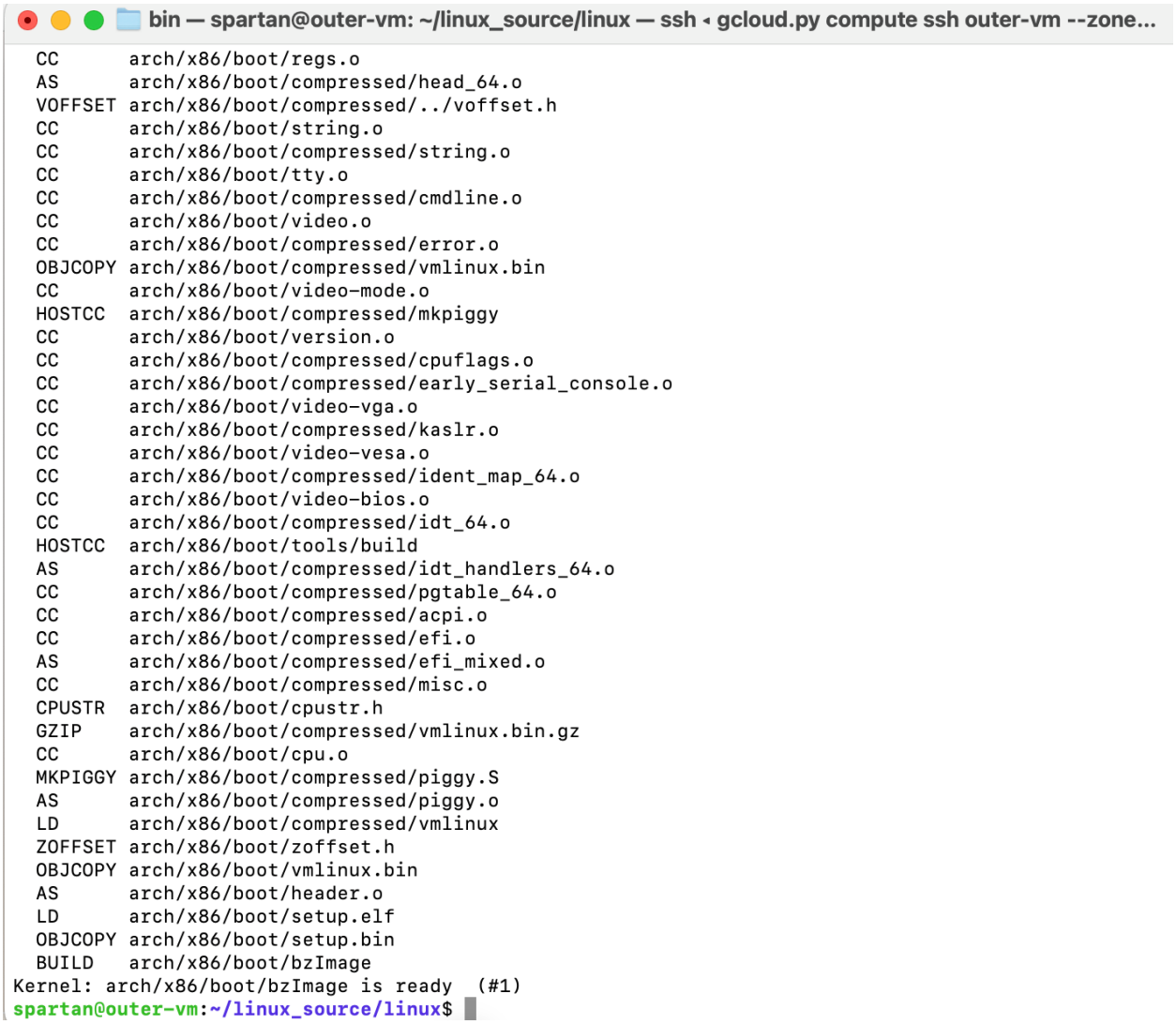
https://github.com/torvalds/linux

https://github.com/SheetalYadav01/linux

Step 6: Performed the new Kernal installation

- Configured the kernel for the VM: make defconfig
- Build the kernel : make -j$(nproc)
-
- Installed the new kernel: sudo make modules_install
  sudo make install
- Update the bootloader and reboot:
  sudo update-grub
  sudo reboot

- Reconnect the VM after reboot: gcloud compute ssh outer-vm --zone=us-central1-a



```
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/40-force-partuuid.cfg'
Sourcing file `/etc/default/grub.d/50-cloudimg-settings.cfg'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
GRUB_FORCE_PARTUUID is set, will attempt initrdless boot
Found linux image: /boot/vmlinuz-6.12.0-g80db457e8d28
Found initrd image: /boot/initrd.img-6.12.0-g80db457e8d28
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
spartan@outer-vm:~/linux_source/linux$
```

```
bin — spartan@outer-vm: ~/linux_source/linux — ssh ◂ gcloud.py compute ssh outer-vm --zone=us-central1-a — 125×39
spartan@IMS-051MBA bin % gcloud compute ssh outer-vm --zone=us-central1-a

Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.12.0-g80db457e8d28 x86_64)

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

 System information as of Thu Nov 21 00:26:54 UTC 2024

  System load:  0.9               Processes:             104
  Usage of /:   1.0% of 968.99GB  Users logged in:       0
  Memory usage: 1%                IPv4 address for ens4: 10.128.0.2
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Wed Nov 20 21:28:08 2024 from 162.195.240.180
spartan@outer-vm:~$ cd linux
grep -r "vmexit" arch/ | grep handler
-bash: cd: linux: No such file or directory
grep: arch/: No such file or directory
spartan@outer-vm:~$ ls
linux_source
spartan@outer-vm:~$ cd linux_source
```

Step 7: Modified the KVM code

- Located the KVM exit handler code in the kernel source code at path arch/x86/kvm/vmx/vmx.c
- Added counters for each exit type and printed statistics every 10,000 exits.
- Rebuilt and installed the modified kernel.

```
6466    }
6467
6468    /* Modify the vmx handle exit function */
6469    static int _vmx_handle_exit(struct kvm_vcpu *vcpu, fastpath_t exit_fastpath) {
6470        struct vcpu_vmx *vmx = to_vmx(vcpu);
6471        union vmx_exit_reason exit_reason = vmx->exit_reason;
6472        u32 vectoring_info = vmx->idt_vectoring_info;
6473        u16 exit_handler_index;
6474
6475        /* Add the counters at the start of the function */
6476        int exit_type = exit_reason.basic; // Get the exit type number
6477
6478        kvm_exit_counters[exit_type]++; // Increment the exit type counter
6479        kvm_total_exits++; // Increment the total exit counter
6480
6481        /* Print every 10,000 total exits */
6482        if (kvm_total_exits % 10000 == 0) {
6483            int i;
6484            for (i = 0; i < 256; i++) {
6485                if (kvm_exit_counters[i] > 0) {
6486                    printk(KERN_INFO "KVM Exit: %d (%s) occurred %llu times\n", i, kvm_exit_reason_to_string(i), kvm_exit_counte
6487                }
6488            }
6489        }
6490
6491            /*
```

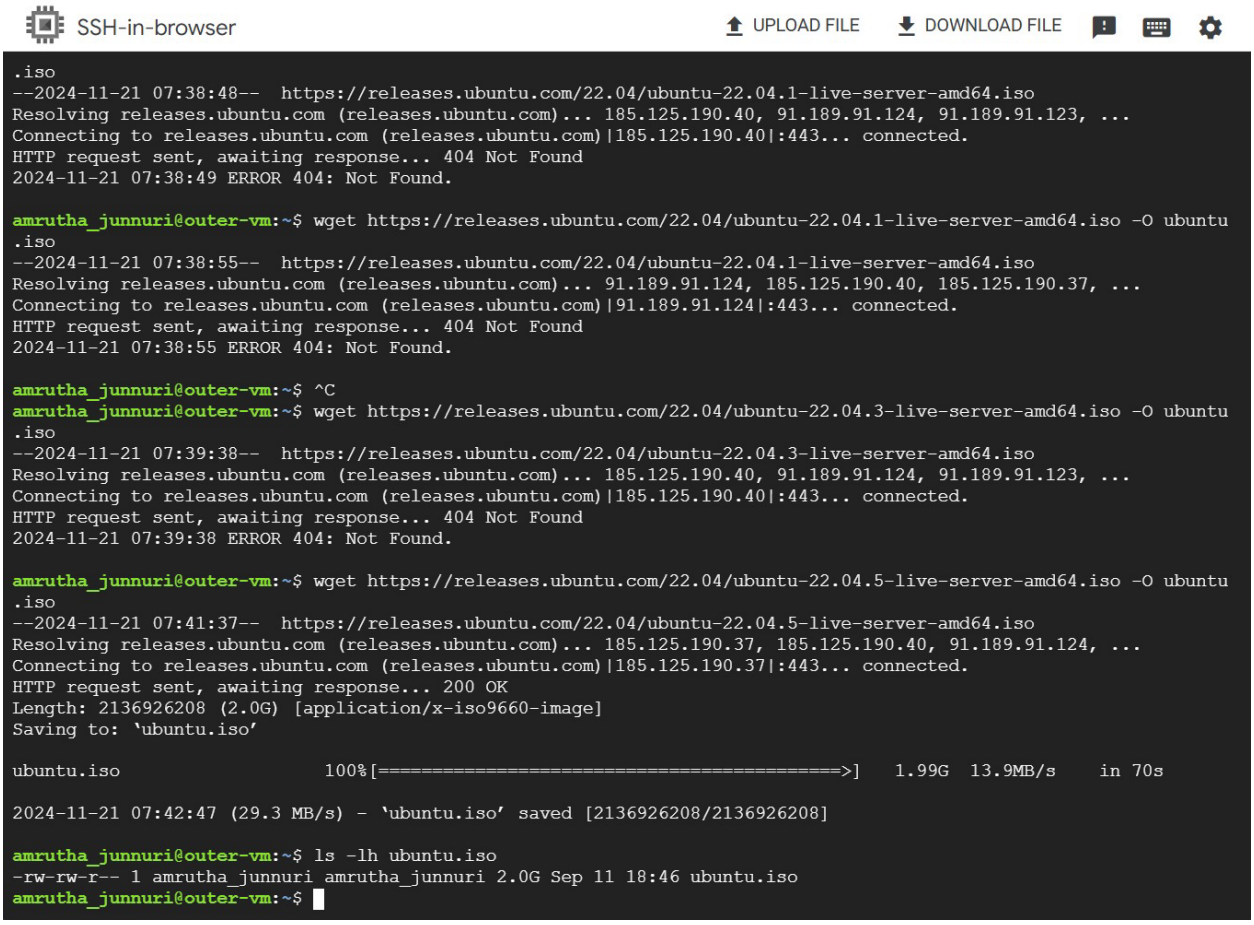
Step 8: Rebuild the Kernel with new changes

make -j$(nproc)
sudo make modules_install
sudo make install
sudo update-grub
sudo reboot

- Reconnected to the VM after reboot:

gcloud compute ssh outer-vm --zone=us-central1-a

Step 9: Tested the Modified Kernel
Created an **inner VM** on the outer VM:



Installed QEMU and KVM:
sudo apt install -y qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils

Started a virtual machine using QEMU:
qemu-system-x86_64 -enable-kvm -m 2048 -hda <inner-vm.img >

Constantly monitored the kernel logs to see the VM exit stats in interval of time
dmesg | tail -n 50

Question 3: Frequency of Exits

The number of exits generally increases at a steady rate during normal VM operations. However, specific activities like booting the VM or performing heavy I/O operations cause noticeable spikes in exit counts. These spikes are due to the intensive processes involved in these operations. For example, exit types such as 30 (Unknown) and 28 (Unknown) are among the most frequently observed, each exceeding a million instances, because they are often invoked during critical operations managed by the hypervisor. Additionally, exit types like 1 (External Interrupt) and 10 (CPUID) also significantly contribute to the overall exit count. On average, a complete VM boot involves approximately 1.5 to 2 million exits, with periods of heightened

activity reflecting the intensive tasks being performed.

Question 4: Most and Least Frequent Exit Types

Among the various exit types observed during VM runtime, Exit Reason 30 (Unknown) and Exit Reason 28 (Unknown) are the most frequent. These exit types occur often during operations such as CPU initialization and interrupt handling, making them the bulk of the exit events. Collectively, these exits surpass a million instances. On the other hand, the least frequent exit type observed is Exit Reason 0 (Exception or NMI), which rarely occurs, indicating that exception handling and non-maskable interrupts are not commonly needed during regular VM operations.