COL732: Virtualization and Cloud Computing
Semester I, 2022-2023

Lab-3: Benchmarking Address Translation
5 October 2022
Deadline: 15 October 2022

Submission Instructions

1. You will submit the report on Gradescope (Lab 3). The benchmarks are available here.
2. You should write the report **without** taking help from your peers or referring to online resources except for documentation. Not doing any of these will be considered a breach of the honor code, and the consequences would range from zero marks in the Lab to a disciplinary committee action.
3. You can use **Piazza** for any queries related to the Lab and **avoid** asking queries on the last day. Do not make any **assumptions** about the assignment based on similar assignments available on the internet in case.

Problem Statement

Divya is a network virtualization engineer at random telecommunication. She has asked you to profile a memory subsystem of the VMM-rust framework that uses either of the following address translation schemes: extended page table and shadow page table. You are provided with the following benchmark suites:

1. Pagetable Benchmark [1]:
   This benchmark characterizes the cost of fork calls in a system. We repeatedly call fork syscall which in return creates a separate page table for each of the child processes. We benchmark the cost of the creation of a page table by measuring the end-to-end latency of executing the benchmark.

2. TLB Benchmark [2]:
   This benchmark characterizes the load on the translation lookaside buffer (TLB). We create a map data structure of variable size in memory, where the size of each value is 2048 bytes. Subsequently, we traverse the map 100 times in a random order and record the time taken to traverse the map.

Setup

1. Start a VM using the instructions shared in Lab-2
2. Create a tap network and connect the VM with the host using the following commands:
   In the host:

```
sudo ip tuntap add dev vmtap100 mode tap
sudo ip addr add 192.168.241.1/24 dev vmtap100
sudo ip link set vmtap100 up
```

   In the VM:

```
ip a add 192.168.241.2/24 dev eth0
ip link set eth0 up
```

3. Compile the benchmarks

```
gcc -static <filename.c>
```

4. Copy the benchmarks from the host to the VM using the following commands:
   In the host:

```
nc -w 3 <VM IP> -p 1234 < <filename.out>
```

   In the VM:

```
nc -l -p 1234 > <filename.out>
```

5. In the case of the Pagetable benchmark, use the time utility of Linux to get the end-to-end latency. In the case of the TLB benchmark, use the time provided by the benchmark as output.

6. To enable/disable EPT on the host machine, use the following command on the host:

```
sudo modprobe -r kvm_intel
sudo modprobe kvm_intel ept=0  # Set ept=1 to enable EPT
cat /sys/module/kvm_intel/parameters/ept # Y represents that EPT is on
and N represents that EPT is off.
```

Evaluation

1. Page table benchmark (10 marks):
    a. Experimental setup. Exact commands/programs that were run for the measurement. (2 marks)
    b. Relationship of runtime (with EPT=1) with the size of the resident memory (2 marks)
    c. Relationship of runtime (with EPT=0) with the size of the resident memory (2 marks)
    d. Justification of observations (4 marks)
2. TLB benchmark (10 marks)
    a. Experimental setup. Exact commands/programs that were run for the measurement. (2 marks)
    b. Relationship of runtime (with EPT=0) with the map size (4K – 256M) (2 marks)
    c. Relationship of runtime (with EPT=1) with the map size (4K – 256M) (2 marks)
    d. Justification of observations (4 marks)

References

1. Adams, K., & Agesen, O. (2006). A comparison of software and hardware techniques for x86 virtualization. *ACM Sigplan Notices*, *41*(11), 2-13.
2. https://github.com/torvalds/test-tlb