CS550: Final Project Report

Project Title: Tracking Page Fault For A Process

Submitted By : Shruti Agrawal (B00815433)

I. Problem

Plot the virtual addresses tracked using kernel module that takes the process-ID of an active process as an argument, as a scatter-plot graph with X-axis representing the time and Y-axis representing the virtual address.Used three different types of target applications named sysbench (compute intensive), iperf (network I/O intensive) and sort (compute and I/O intensive) to observe interesting trends in memory access patterns of a process.

II. Software Design and Implementation

Software tools used to implement this project are :

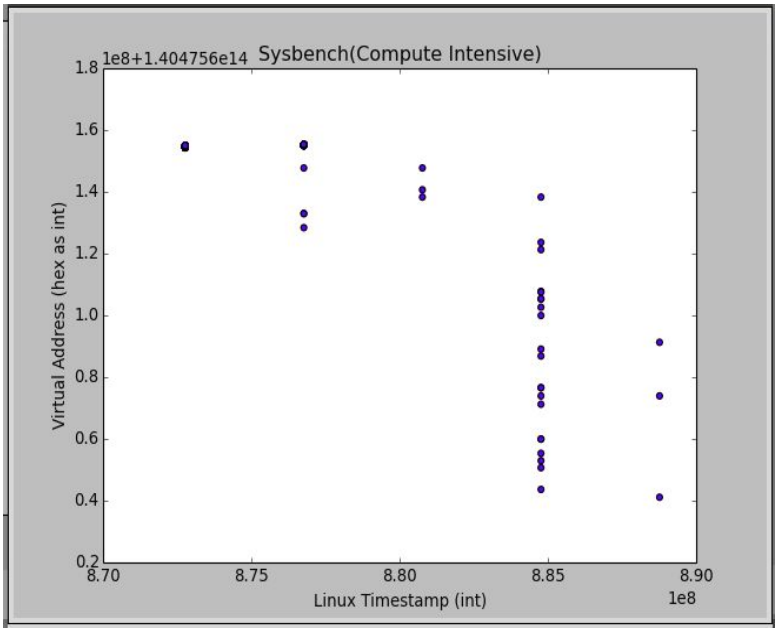
- kprobe module
- sysbench tool.
- iperf tool
- sort tool

A. Software Design and Tools Used

1. **sysbench application** : sysbench provides benchmarking capabilities for Linux. sysbench supports testing CPU, memory, file I/O, mutex performance.For the purpose of the report, CPU performance has been tested using

CMD : sysbench cpu --cpu-max-prime=50000 --time=30 run & echo $!
The maximum prime number checked in the CPU test 50,000 in a time duration of 30seconds.

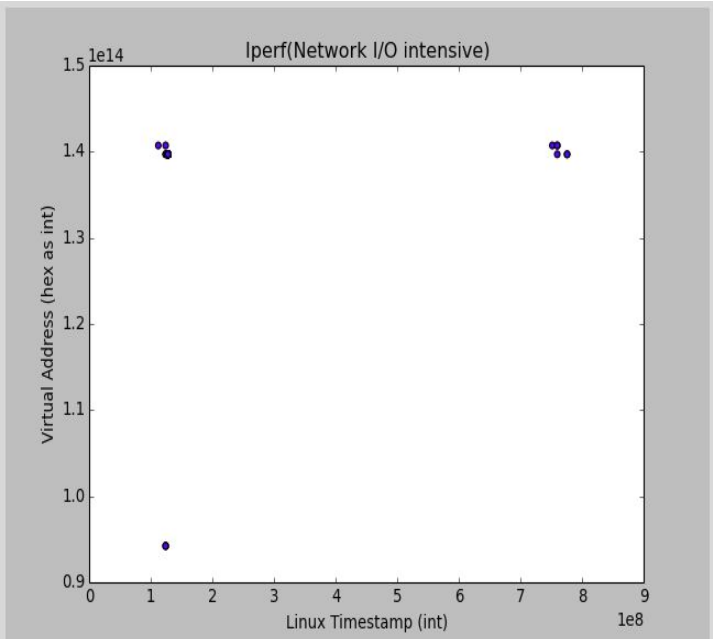
Plotted Graph :



2. iperf (network I/O intensive) : iPerf is a command-line tool used in diagnosing network speed issues by measuring the maximum network throughput a server can handle. It is particularly useful when experiencing network speed issues, as you can use iPerf to determine which server is unable to reach maximum throughput.

CMD : iperf -s & //To launch iperf on the server.

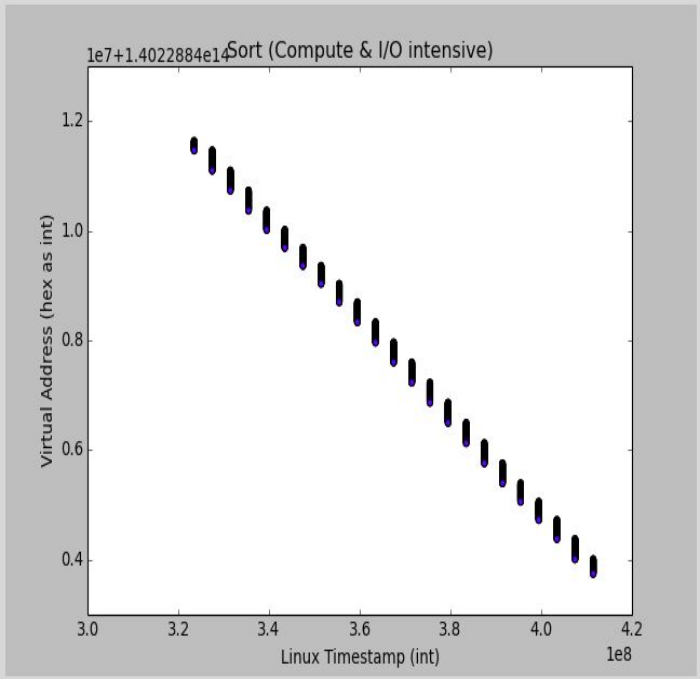
iperf -c localhost -b 10m -l 10M -n 500240M -P 70 // connect localhost to the server establishing 70 connections with some bandwidth.



3. SORT Application : Sort is a linux utility tool that allows to sort the data in a given file

CMD: shuf -i 0-36000000 -n 10000000000 > test.txt

sort test.txt > sort_test.txt &



B. Analysis of Graphs :

- For a compute intensive application, we expect page faults to occur at:
  - Beginning of the process to load the program and data needs to be bootstrap.
  - At certain intervals while the process is running to access intermediate data needed for computation.
  - Same behaviour is observed in our plot, as we see page faults occurring from beginning to end at certain intervals.

- For network intensive application, we expect page faults to occur when the network connections are created as well as destroyed.
- Similar behaviour is observed in our graph. For our scenario, we created 70 parallel connections so we see page faults at the beginning. There is another group of page faults observed at the end when connections are dropped. It seems the server has to maintain some state for each connection and as the total number of connections increases, we see page faults occurring to store or access data for these connections.

For compute & I/O intensive application : The page faults seem to occur throughout the process.

- For IO operations, it's possible the data needed for IO is not in RAM and thereby generates page faults.
- Compute operations as explained in compute intensive process, we can expect page faults to occur throughout the process randomly
- For our use case, we use sort tool in linux, which is a merge sort, we think it's because of the nature of merge sort we also see page faults sequential across the address.

References

https://www.kernel.org/doc/Documentation/kprobes.txt

https://elixir.bootlin.com/linux/latest/source/samples/kprobes

https://wiki.gentoo.org/wiki/Sysbench