Part B Report

Name: Neha Gupta
B-Number: B00818087
B-ID: ngupta4

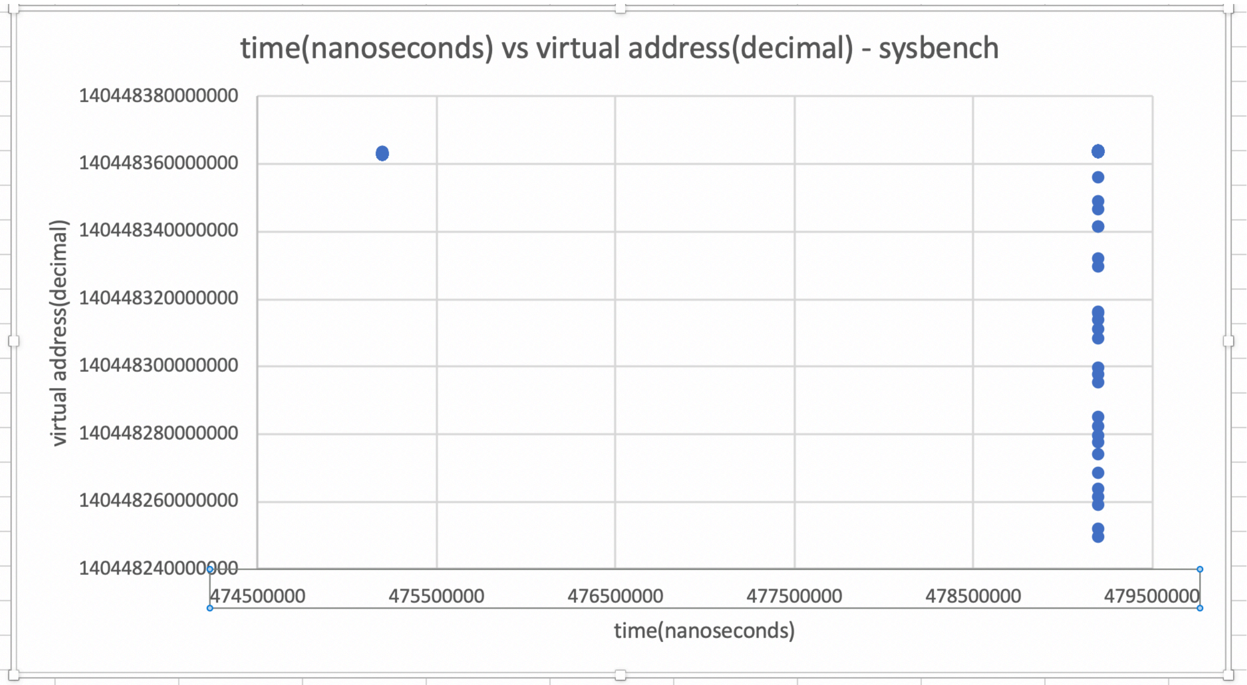
When a running program accesses a memory page that is not presently mapped into the virtual space of the process my the MMU then an exception occurs which is known as Page Fault.

In this assignment we have tracked the virtual address and the time at which the process page faults occur. For this,
I have tracked the page faults in the pre-handler function of the kprobe, accessing the time via current_kernel_time() function and accessing the virtual address via regs->si.

Virtual memory is an illusion created by the memory management unit to make the process believe that they have more space than what is actually allotted to them.

I have tested my code against 3 applications:

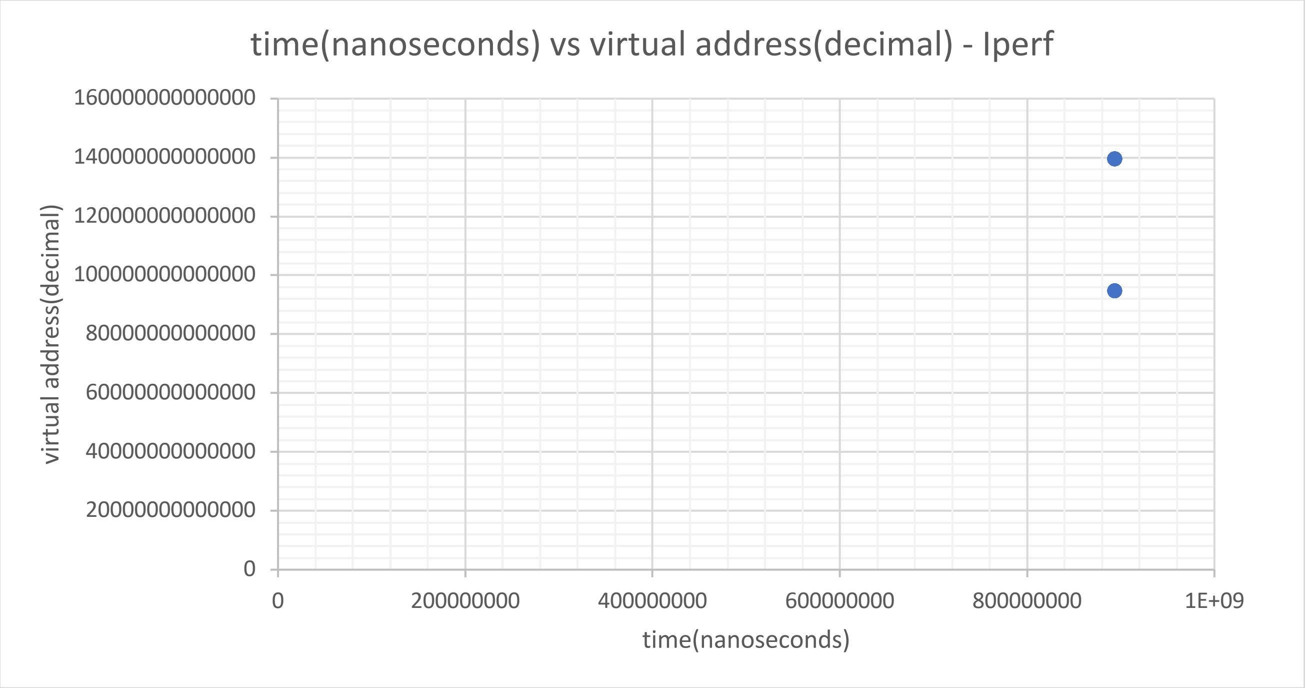
1. Sysbench (Compute Intensive)
   Sysbench fetches the system performance. Below is the graph I have obtained.



The above graph shows that page faults occur more in the start as the of the process then stabilizes in the middle with again a cluster at the end.

2. Iperf (Network I/O intensive)

   Iperf is used to test the network performance on a system.

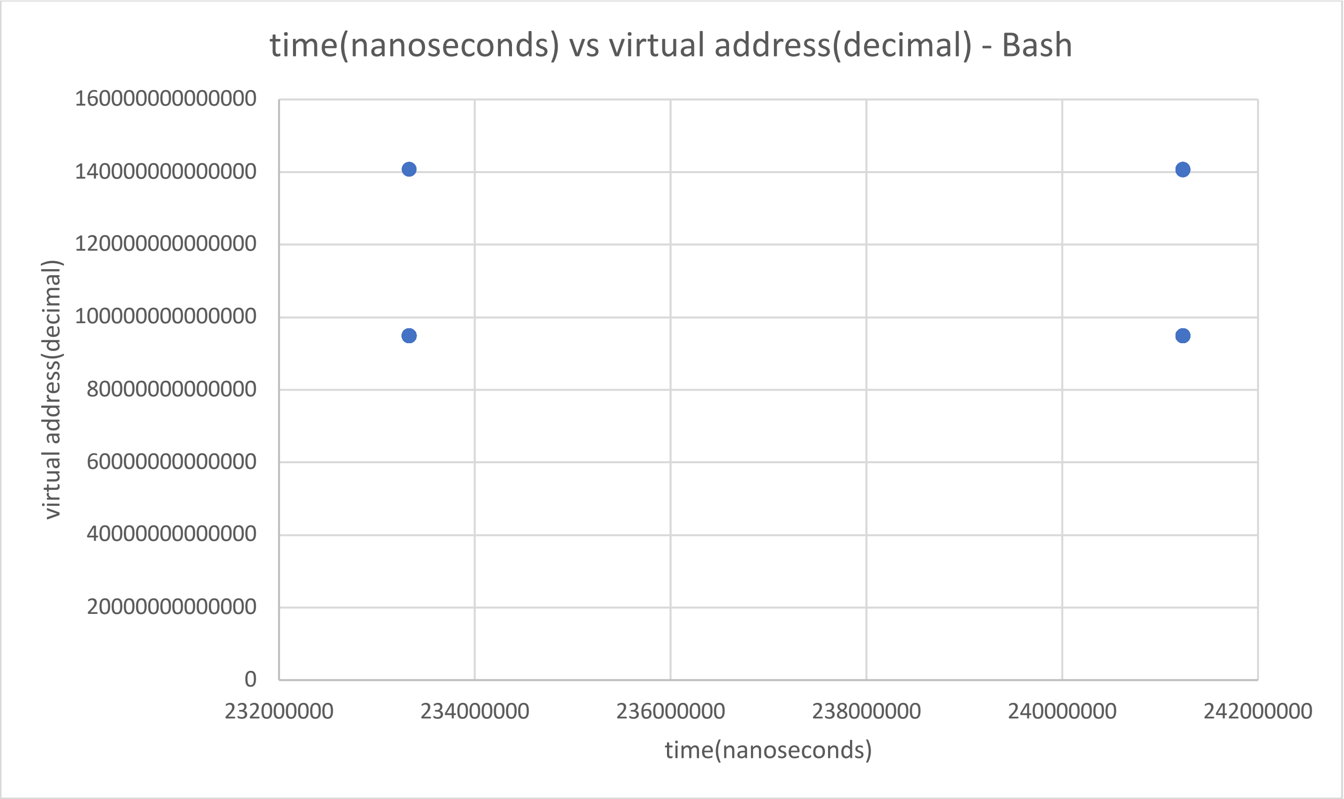


For the above process:

a. We set up the the server in one terminal
b. We test the network from the client and while is running we run the script to get the page faults.

In the above graph we see that the page faults are in a vertical line.

3. General process in the system.



In the above graph we selected a random process from the system and checked the occurance of page fault.

We see that the MMU constantly keeps updating the virtual address space with current process requesting memory access.
Due to the limitation of virtual memory there are recurring page faults.