CPE434-2021

Homework Chapter 10

Due March 18, 2021

1. Your virtual environment has two separate page tables, one for your host and one for your guest. In Intel machines with virtualization extensions, these tables are walked by hardware but they still need to do memory accesses which must be walked to fetch a data value. Theoretically, this should make programs with large memory footprints run much slower on a VM then on a native machine due to the extra tlb misses.
2. Using Valgrind with the cachegrind option, write a program that causes lots of cache misses and takes at least 10 seconds to run .In order to have lots of TLB misses it should access a large data structure- perhaps an array on the order of size 10 million. As in the case of the labs, this should be in the data or heap area, not the stack. Run and time this program running on your VM. Record your results. And include your source code.
3. Download a C compiler for your native machine (most likely windows10). You can get a free copy of visual studio community edition or you can get a gnu c compiler via mingw or other compiler tool. Recompile the program written for part a and run and record the timing and your source code (it should be the same as part a).
4. Try and find an array size and stride (the step size between array accesses) that demonstrates that programs in VMs with large TLB footprints run slower on a VM than on a native operating system. Report back all your experiments. Conduct at least 10 experiments.

2- In the previous homework you had a virtual address space with a page size of 64KB. For small programs this large page size results in lots of wasted space. Assume you were going to redesign this architecture for a page of 1k bytes (1024) how would you allocate the extra 6 bits to the index fields. For example you could do (0,0,+6),( 0,+6,0), (+6,0,0 ).(+2,+2,+2), For small programs consisting of one page of text and heap, and one page of stack which allocation of bits would be more efficient. Explain your answer.

3. (GRADUATE STUDENTS ONLY). The Dell NUMA document describes the disadvantages of having user memory located on a NUMA node other than the node executing the user program. It does not describe the disadvantages of having the page tables for large programs located on an inappropriate processor’s memory. This subject has been studied recently and a method called the mitosis method is described in the following two documents. Starting with these documents (any any others you find appropriate) prepare a short 3-5 slide presentation on the costs of having the page tables inappropriately located and how the mitosis system addresses this. Please be sure to include information on the performance improvement, if any, of the mitosis approach.

http://www.cs.yale.edu/homes/abhishek/reto-osdi18.pdf

https://arxiv.org/pdf/1910.05398.pdf