CprE 381 Homework 11

[Note: This is your final homework of the semester. You are tasked with answering your curiosity regarding the real cache structures within the computers around you. While this homework will appear short on first glance, it will take some time, some effort, and certainly some thought. However, you will come away with some hard-won real-world understanding of computers. As always, enjoy.]

1. Programs and Caches
    a. Find at least one computer on which you can run the included C programs. While a lab computer or your laptop is fine, if possible, try testing on other platforms. *[In the worst case, you can complete this on the ECpE Linux lab computers.]* Compile the three test programs, run the programs for the following inputs and report their output. *[These inputs are just to help confirm that you can compile and run the examples; they are not the solution to the below problems.]*
    ```
    ./test1 16 20 8
    ./test2 16 28 8
    ./test3 16 20 8 2
    ```
    b. Open up the c code files for the tests and look at the meaning of the arguments (they are not the same throughout the tests). Play around with the arguments to get a feel for how they impact the runtime of the core loop in each test. Select a reasonably large array size (likely larger than the size used above) and iteration count that allows the execution time to be roughly 1s on your machine. For test1, run several different values and observe the differing behavior. Try to explain to yourself why this behavior is being observed. Once you can explain the behavior, move on to the following parts. *[There is nothing to turn in for this part, but it is probably the most important part of the HW.]*
    c. Make a plot of j vs CPU time from the results of test1.c and report the input arguments you used. Does the processor in your computer/platform have a cache? Use the plot to justify your answer. Specifically point out one piece of sizing information that this plot gives you and write down your reasoning.
    d. Make a plot of j vs CPU time from the results of test2.c where you use the information gained above to select the additional command-line inputs; report the input arguments you used. Specifically point out another piece of sizing information that this plot gives you and write down your reasoning.
    e. Make a plot of j vs CPU time from the results of running test3.c where you use both pieces of information from above to set your additional inputs; report the input arguments you used. Specifically identify another piece of sizing information that this plot gives you and write down your reasoning.
    f. Finally, put it all together. What is the caching structure of the processor in your computer/platform? Specifically give its total data size, block size, and associativity (i.e., # of ways per set). Then report the specific processor on which you are running (e.g., run “cat /proc/cpuinfo” on a Linux system and report the

“model name”) and attempt to look up the cache sizes and configurations (specifically look for the L1 data cache). Does this match your experimental observations?