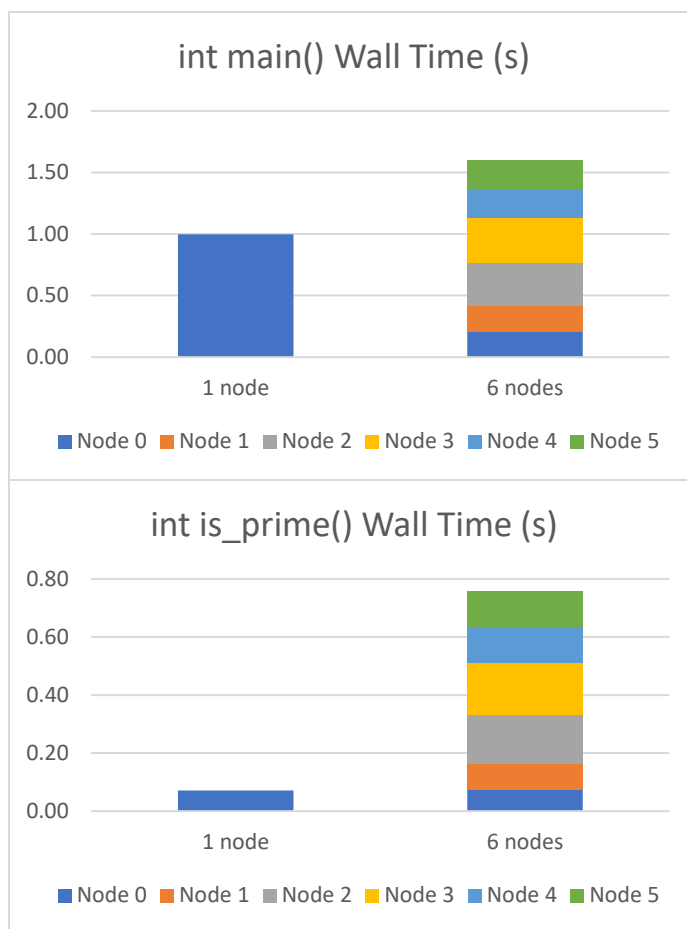Michael Rivnak

COMP3450

Leon Deligiannidis

25 June 2020

LA3 Analysis

First off, I was incredibly surprised with how quickly all of these calculations were done, even in the single threaded program. As far as results go, the main function in the single and multithreaded tests took similar amounts of time, with the multithreaded program total being slightly more due to the additional overhead even though the individual times were lower. What was unexpected is each of the individual is_prime() wall times for the multithreaded program were as long or longer than the singlethreaded program. I am not sure of the explanation for this as all of my other testing showed that the program is distributing the data as intended. One point about this is that paraprof shows the number of calls to is_prime() for each node as the same as the total for the singlethreaded program. This is different from what my testing of the data distribution showed me, and the program output also shows that there is no extra data being distributed.

For L1 Cache misses there was a single outlier in my data for the multithreaded program. Node 4 had considerably more misses than the other 5 nodes. Further, these misses were primarily on the main() and is_prime() functions. Perhaps this could be due to extra load on that node? Every other value was approximately the same, including the values from the singlethreaded program

TAU: ParaProf: C:\Users\Michael\Documents\Classwork\Parallel Computing\LA3\app_L1_parallel.ppk

File   Options   Windows   Help

Metric: PAPI_L1_DCM
Value: Exclusive

Std. Dev.
Mean
Max
Min
node 0
node 1
node 2
node 3
node 4
node 5

TAU: ParaProf: C:\Users\Michael\Documents\Classwork\Parallel Computing\LA3\app_L1_serial.ppk

File   Options   Windows   Help

Metric: PAPI_L1_DCM
Value: Exclusive

Std. Dev.
Mean
Max
Min
node 0