September 15th, 2015 Pre-Class Questions

Elliot Cartee

Question 0

I started these pre-class exercises on September 14th at about 8PM and finished around 9PM

Question 1

For the slides on the Game of Life, I was confused by how exactly the coarse blocking scheme would work for the high performance implementation.

Question 2

Because of the periodic boundary conditions, this implementation stores the board state in $(n+2)^2$ bytes. Given that the size of the L3 cache on one of the totient nodes is 15MB, we see that this happens when:

$$(n+2)^2 < 15 \cdot 2^{20}$$

which becomes

$$n < \sqrt{15 \cdot 2^{20}} - 2 \approx 3963$$

Running the glider example on the totient node with $n = 2000$ for 100 generations takes 2.006 seconds. Meanwhile, running the same example with $n = 8000$ for 100 generations takes 26.666 seconds. I expected it to take more than 16 times longer since it no longer fits in the cache, but maybe I should be doing this for more generations?

Question 3

A blocking strategy that could improve operational intensity would be to dynamically divide the game board into regions that are separated from other non-empty regions by multiple rows/columns of blank spaces, so that they cannot interact in the next few time steps, making it possible to compute multiple time steps without needing to wait for the other blocks.

Question 4

In order to parallelize this code, we would need some method of communicating what cells are occupied on the borders of each domain. I would expect to see decent speedups relative to the number of processors since the small number of processors would keep communication costs low.

Question 5

I think this code could be improved by switching the loop ordering so that each processor only has to communicate the properties of particle j once and then calculate the forces with each particle i, rather than gathering the properties of particle j every time for each i. This should reduce communication costs significantly.