CS 5220 – 2015-09-24 Preclass Questions

Michael Whittaker (mjw297)

September 23, 2015

0. I spent 2 hours on September 23.

1. I think the semantics of the various OpenMP constructs could have been explained a little bit better. When I read the paper of common OpenMP mistakes, I realized that I didn't understand a lot of the things being mentioned.

2. a) The model for the OpenMP Monte Carlo simulation is the same as the one we derived for the pthread simulation.

$$T = \frac{Nt_t}{p} + \frac{Nt_u}{b}$$

   b) I ran `omp_mc.x` twice: once with 32 threads and a batch size of 1 and another with 1 thread and a batch size of 32. This produced the following output:

```
--- Run input parameters:
rtol:      1.000000e-04
maxtrials: 1000000
nbatch:    32
0.49983 (0.000288676) from 1000032 trials
1 threads (OpenMP): 9.624004e-03 s

--- Run input parameters:
rtol:      1.000000e-04
maxtrials: 1000000
nbatch:    1
0.499869 (0.000288558) from 1000031 trials
32 threads (OpenMP): 1.224949e+00 s
```

   Plugging in these times to our model, we get the following equations:

$$0.009624004 = \frac{1000032t_t}{1} + \frac{1000032t_u}{32}$$
$$1.224949 = \frac{1000031t_t}{32} + \frac{1000031t_u}{1}$$

   Solving these equations, we get $t_t = -2.86828 \times 10^{-8}$ and $t_u = 1.22581 \times 10^{-6}$.

   c) As with the pthread simulation, our model is too naive to consider the negative impact of infinitely large batch sizes. Our model suggests that we can increase the batch size without any negative consequences. In reality this is not true. In practice, I'd choose a batch size by sweeping the space of batch sizes to see which batch size produced the fasted code.

3. `omp_mc.c` computes unnecessary computation in a critical section, but otherwise doesn't have any of the common performance mistakes listed in the paper.