0. How much time did you spend on this pre-class exercise, and when?

4 hours. 9-21 afternoon and night

1. What are one or two points that you found least clear in the

9/22 slide decks (including the narration)?

Don’t quite understand the “Some loose ends” part of the slides

2. The pthread_mc.c file in the demo subdirectory runs a Monte Carlo

simulation to estimate the expected value of a uniform random

variable. The "-p" option sets the number of processors used,

while "-b" sets the number of trials between synchronizations.

a) Write a model for the run time for this simulation code in

terms of the number of trials (N), number of processors (p),

time per trial (t_trial), and time to update the global

counters in the critical section (t_update).

If we have N trial and p processors, each of the processor will do N/p trails and this takes N* t_trial /p. Assume each batch we do b trails(I think we need b to set up the model, or for some case, can we assume b is set in the code as b = nbatch = 50 or should we assume b = N/p or some other value? I’ll just use b instead of 50), each batch will take 1 update so it takes N*t_update /(b*p) for each processor to update. And together we need N*t_update /b time for updates. So the running time is estimated as N* t_trial /p+ N*t_update /b

b) Run the code with a few different parameter values in order

to estimate N, t_trial, and t_update for this code on

a totient compute node.

(I think there is a bug in the code that the return of is_converged should be (varX/(EX*EX*all_ntrials) < rtol*rtol || all_ntrials > maxtrials) instead of (varX/(EX*EX) < rtol*rtol || all_ntrials > maxtrials))

I find

$N \approx 3240$

and

t_update $\approx$ 2.78E-08

t_ trial $\approx$1.08E-08

c) Based on your model, suggest a strategy for choosing the batch size. How might you generalize this strategy to automatically choose batch sizes for different types of computational experiments?

(N$_{real}$-N$_{need}$)* t_ trial/p is the cost for increasing b since increasing b may let you do more trails in the last batch for each core. N*t_update /b^2 is the time you can save for increasing b(using differential) when this two are equal, we can find the best b. For example, we can find the best result of p=1 is about b = 91. In the experiment, I find it should be somewhere between 90 and 120

3. In the workq subdirectory of this directory, there is a basic work queue implementation. Following the strategy outlined in the slides, add synchronization calls in the locations marked TODO. You should run the code to make sure it behaves as expected!

Done~ successfully.

(you can see the code in /workq)