HPC_HW_Chap4_PartA
Chen Fang

Problem 4.3

With –O0 optimization, the multi-thread calculation yields the same result as the single-threaded calculation. With –O2 optimization, however, the multi-threaded calculation will stall when thread number increases to a certain value. Due to the flag and sum variables are shared, these two variables should be declared volatile. And the result is correct with and without optimization.

Problem 4.8

a)

The program will stop at time 1 because the mutex has been acquired by the other thread at time 0.

b)

There is no problem if busy-waiting is used because the the flag will not be updated until one thread finishes its work, while the second thread keeps waiting.

c)

No problem with semaphores either. When work assigned at time 0 is completed, the semaphore will be updated to “unlock” status such that it can be accessed by the thread at time 1.

Problem 4.12

I have the feeling that it will be unsafe in some cases, but I couldn’t find a counter example.