EECS 358 HW1

Chenhui Zhou
Wenjie Zhang

May 2016

1 Problem1

1. The running time of the serial algorithm:
   Assuming there are two operations (plus and multiply) in the loop the serial time is: $2 * n * m$

2. The running time of the parallel algorithm:
   With the same assumption, the parallel execution time is $[2*n*m/p+m*(p+1)]$, because the initialization cost m, then for the main loop, the first loop will be executed $n * m/p$ times (hence $2 * n * m/p$), and the second loop will be executed $m * p$ times (because of the lock).

3. Since $T_s = 2*n*m, T_p = [2*n*m/p+m*(p+1)]$, $S_p = T_s/T_p = \frac{2*n*p}{2*n+p*(p+1)}$

4. The Amdahl's fraction for this parallel algorithm can be determined by rewriting the previous equation as:
   $S_p = \frac{T_s}{T_p} = \frac{2*n*p}{2*n+p*(p+1)} \Rightarrow S_p = \frac{p}{1+(p-1)*\alpha}$
   Hence, the Amdahl's fraction α is:
   $\alpha(n,p) = \frac{(p+1)*p}{2*n*(p-1)}$
   $\alpha(n,p) \to 0$ as $n \to \infty$ for fixed p
   Hence, the parallel algorithm is effective.

2 Problem2

- Run time for *pi.c*:
  12.965u 0.000s 0:12.96 100.0% 0+0k 16+0io 0pf+0w

- Run time for *pi2.c*:

  1. One Processor:
     13.630u 0.000s 0:13.63 100.0% 0+0k 0+0io 0pf+0w
  2. Four Processors:
     15.838u 0.089s 0:03.99 398.7% 0+0k 0+0io 0pf+0w

3. Eight Processors:
   20.512u 0.585s 0:02.70 781.1% 0+0k 0+0io 0pf+0w

- Run time for *pi*1.*c*:

1. One Processor for static scheduling techniques:
   13.598u 0.001s 0:13.60 99.9% 0+0k 0+0io 0pf+0w
2. Four Processors for static scheduling techniques::
   13.407u 0.000s 0:03.35 400.0% 0+0k 0+0io 0pf+0w
3. Eight Processors for static scheduling techniques:
   13.408u 0.001s 0:01.68 797.6% 0+0k 0+0io 0pf+0w
4. One Processor for dynamic scheduling techniques:
   45.370u 0.000s 0:45.37 100.0% 0+0k 0+0io 0pf+0w
5. Four Processors for dynamic scheduling techniques::
   551.648u 0.005s 2:17.94 399.9% 0+0k 0+0io 0pf+0w
6. Eight Processors for dynamic scheduling techniques:
   1114.478u 0.066s 2:19.51 798.8% 0+0k 0+0io 0pf+0w

- Run time for *multdot.c*:

1. One Processor:
   2.390u 0.995s 0:03.38 100.0% 0+0k 0+8io 0pf+0w
2. Four Processors:
   2.495u 1.782s 0:01.22 350.0% 0+0k 0+0io 0pf+0w
3. Eight Processors:
   2.634u 2.603s 0:00.68 769.1% 0+0k 0+0io 0pf+0w

3 Probelm3

- Performance for pthread on 5000 x 5000:
  - One Processor: Elapsed time = 408495 ms.
  - Four Processors: Elapsed time = 108037 ms
  - Eight Processors: Elapsed time = 89144.1 ms
- Performance for openmp on 5000 x 5000:
  - One Processor: Elapsed time = 455996 ms
  - Four Processors: Elapsed time = 116749 ms
  - Eight Processors: Elapsed time = 77443.5 ms