HPC21 Assignment 3

Zheng Liu

March 28 2021

1 Problem 1

1.1 a

Two threads work on the loop, one chunk for each. In loop 1, thread 0 spends $t_0 = (1+2+...+\frac{n-1}{2})$ms working on its chunk. Thread 1 spends $t_1 = (\frac{n}{2} + (\frac{n}{2}+1) + ... + n-1)$ms working on its part. Thread 1's job obviously takes longer so thread 0 has to wait for thread 1 to finish. Then two threads enter loop 2. In loop2, due to the $f(n-1)$, two threads take the chunk that are identical to each other's chunk in loop 1. And the theory about waiting due to imbalanced load is the same as for loop 1. Therefore, the total time spent is $2 \times t_1 = 2(\frac{n}{2} + (\frac{n}{2}+1) + ... + n-1)$ms. And each thread waits for $(t_1 - t_0)$ms in the corresponding loop.

1.2 b

The program will run faster because, for example in loop 1, thread 0 will get iteration $\#1, 3, 5, 7...$ and thread 1 will get $\#2, 4, 6, 8....$ Therefore, the loads of two threads are basically balanced in each loop. Although the threads still have to both finish loop 1 before stating loop 2, their finishing time will be close and the wait time will be much reduced.

1.3 c

I ran a few experiments and I think improvement is possible but will be subtle. For example, thread 0 can have iteration $1, 4$ while thread 1 has $2, 3$. So in some cases, the load can be even more balanced than (static,1) and benefit from even less wait time. But in dynamic mode, there is also cost of maintaining the task queue.

1.4 d

One can add nowait. Then thread 0 can enter loop 2 without waiting for the slow thread 1 in loop 1. Also, if we look across two loops, the load for two threads are balanced (thread 0: head chunk in loop 1 and tail chunk in loop 2, thread 1: tail in loop 1 and head in loop 2). it will cost each thread $(1 + 2 + ...n)$ms.

2 Problem 2

Bugs are fixed. Please see the comments added in the source code, which describe the bugs and explain the solutions.

3 Problem 3

My CPU is Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz. It has 6 cores and 2 threads per core. The following are the timings with different OMP_NUM_THREADS settings:

```
Thread Num:2
sequential-scan = 0.257015s
parallel-scan = 0.195628s
error = 0

Thread Num:3
sequential-scan = 0.252310s
parallel-scan = 0.156917s
error = 0

Thread Num:4
sequential-scan = 0.254057s
parallel-scan = 0.136905s
error = 0

Thread Num:5
sequential-scan = 0.253566s
parallel-scan = 0.131481s
error = 0

Thread Num:6
sequential-scan = 0.259150s
parallel-scan = 0.128253s
error = 0

Thread Num:7
sequential-scan = 0.246337s
parallel-scan = 0.137727s
error = 0

Thread Num:8
sequential-scan = 0.248492s
parallel-scan = 0.133477s
error = 0

Thread Num:9
sequential-scan = 0.249489s
parallel-scan = 0.131544s
error = 0

Thread Num:10
sequential-scan = 0.270189s
parallel-scan = 0.138243s
error = 0

Thread Num:11
sequential-scan = 0.256809s
parallel-scan = 0.135535s
error = 0
```

```
Thread Num:12
sequential-scan = 0.254986s
parallel-scan = 0.140050s
error = 0
```

Peak performance is reached with 6 threads. My guess is one of the reasons: data alignment, too little demand, and maybe communication cost.

4 Problem 4

The code include both the serial approach and the omp approach. The line of executing the serial function is commented out. In the timings below, G-S with 1 thread still runs the red-black variant.

I picked to do N=50,100,200,500 with 100000 iterations. My CPU is Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz. It has 6 cores and 2 threads per core. As seen in the outputs, number of threads has no effect on the residual. 6 threads also is a good choice here.

```
N=50
Jacobi with 1 omp threads (N=50):          G-S with 1 omp threads (N=50):
Time: 0.4908s, Residual: 0.000000          Time: 0.5761s, Residual: 0.000000

Jacobi with 2 omp threads (N=50):          G-S with 2 omp threads (N=50):
Time: 0.3059s, Residual: 0.000000          Time: 0.4401s, Residual: 0.000000

Jacobi with 3 omp threads (N=50):          G-S with 3 omp threads (N=50):
Time: 0.2388s, Residual: 0.000000          Time: 0.3712s, Residual: 0.000000

Jacobi with 4 omp threads (N=50):          G-S with 4 omp threads (N=50):
Time: 0.2248s, Residual: 0.000000          Time: 0.3754s, Residual: 0.000000

Jacobi with 5 omp threads (N=50):          G-S with 5 omp threads (N=50):
Time: 0.2295s, Residual: 0.000000          Time: 0.3671s, Residual: 0.000000

Jacobi with 6 omp threads (N=50):          G-S with 6 omp threads (N=50):
Time: 0.2064s, Residual: 0.000000          Time: 0.3805s, Residual: 0.000000

Jacobi with 7 omp threads (N=50):          G-S with 7 omp threads (N=50):
Time: 0.2551s, Residual: 0.000000          Time: 0.4076s, Residual: 0.000000

Jacobi with 8 omp threads (N=50):          G-S with 8 omp threads (N=50):
Time: 0.2451s, Residual: 0.000000          Time: 0.4248s, Residual: 0.000000

Jacobi with 9 omp threads (N=50):          G-S with 9 omp threads (N=50):
Time: 0.2411s, Residual: 0.000000          Time: 0.4095s, Residual: 0.000000

Jacobi with 10 omp threads (N=50):         G-S with 10 omp threads (N=50):
Time: 0.2485s, Residual: 0.000000          Time: 0.4197s, Residual: 0.000000

Jacobi with 11 omp threads (N=50):         G-S with 11 omp threads (N=50):
Time: 0.2443s, Residual: 0.000000          Time: 0.4604s, Residual: 0.000000

Jacobi with 12 omp threads (N=50):         G-S with 12 omp threads (N=50):
Time: 0.2454s, Residual: 0.000000          Time: 0.6911s, Residual: 0.000000
```

```
N=100
Jacobi with 1 omp threads (N=100):         G-S with 1 omp threads (N=100):
Time: 1.7145s, Residual: 0.000000          Time: 2.1102s, Residual: 0.000000

Jacobi with 2 omp threads (N=100):         G-S with 2 omp threads (N=100):
Time: 0.9339s, Residual: 0.000000          Time: 1.1711s, Residual: 0.000000

Jacobi with 3 omp threads (N=100):         G-S with 3 omp threads (N=100):
Time: 0.6692s, Residual: 0.000000          Time: 0.9007s, Residual: 0.000000

Jacobi with 4 omp threads (N=100):         G-S with 4 omp threads (N=100):
Time: 0.5561s, Residual: 0.000000          Time: 0.7779s, Residual: 0.000000

Jacobi with 5 omp threads (N=100):         G-S with 5 omp threads (N=100):
Time: 0.4920s, Residual: 0.000000          Time: 0.6895s, Residual: 0.000000

Jacobi with 6 omp threads (N=100):         G-S with 6 omp threads (N=100):
Time: 0.4252s, Residual: 0.000000          Time: 0.6397s, Residual: 0.000000

Jacobi with 7 omp threads (N=100):         G-S with 7 omp threads (N=100):
Time: 0.6153s, Residual: 0.000000          Time: 0.8383s, Residual: 0.000000

Jacobi with 8 omp threads (N=100):         G-S with 8 omp threads (N=100):
Time: 0.5873s, Residual: 0.000000          Time: 0.7714s, Residual: 0.000000

Jacobi with 9 omp threads (N=100):         G-S with 9 omp threads (N=100):
Time: 0.5305s, Residual: 0.000000          Time: 0.7768s, Residual: 0.000000

Jacobi with 10 omp threads (N=100):        G-S with 10 omp threads (N=100):
Time: 0.4947s, Residual: 0.000000          Time: 0.7458s, Residual: 0.000000

Jacobi with 11 omp threads (N=100):        G-S with 11 omp threads (N=100):
Time: 0.4697s, Residual: 0.000000          Time: 0.6813s, Residual: 0.000000

Jacobi with 12 omp threads (N=100):        G-S with 12 omp threads (N=100):
Time: 0.4681s, Residual: 0.000000          Time: 0.7169s, Residual: 0.000000


N=200
Jacobi with 1 omp threads (N=200):         G-S with 1 omp threads (N=200):
Time: 7.3480s, Residual: 0.401424          Time: 8.5767s, Residual: 0.001261

Jacobi with 2 omp threads (N=200):         G-S with 2 omp threads (N=200):
Time: 3.5888s, Residual: 0.401424          Time: 4.1403s, Residual: 0.001261

Jacobi with 3 omp threads (N=200):         G-S with 3 omp threads (N=200):
Time: 2.8154s, Residual: 0.401424          Time: 3.0981s, Residual: 0.001261

Jacobi with 4 omp threads (N=200):         G-S with 4 omp threads (N=200):
Time: 1.9335s, Residual: 0.401424          Time: 2.3105s, Residual: 0.001261
```

```
Jacobi with 5 omp threads (N=200):        G-S with 5 omp threads (N=200):
Time: 1.6095s, Residual: 0.401424         Time: 1.9343s, Residual: 0.001261

Jacobi with 6 omp threads (N=200):        G-S with 6 omp threads (N=200):
Time: 2.2350s, Residual: 0.401424         Time: 1.7593s, Residual: 0.001261

Jacobi with 7 omp threads (N=200):        G-S with 7 omp threads (N=200):
Time: 2.2093s, Residual: 0.401424         Time: 2.4993s, Residual: 0.001261

Jacobi with 8 omp threads (N=200):        G-S with 8 omp threads (N=200):
Time: 2.0018s, Residual: 0.401424         Time: 2.4667s, Residual: 0.001261

Jacobi with 9 omp threads (N=200):        G-S with 9 omp threads (N=200):
Time: 1.8859s, Residual: 0.401424         Time: 2.5259s, Residual: 0.001261

Jacobi with 10 omp threads (N=200):       G-S with 10 omp threads (N=200):
Time: 1.6811s, Residual: 0.401424         Time: 2.0522s, Residual: 0.001261

Jacobi with 11 omp threads (N=200):       G-S with 11 omp threads (N=200):
Time: 1.8133s, Residual: 0.401424         Time: 1.9045s, Residual: 0.001261

Jacobi with 12 omp threads (N=200):       G-S with 12 omp threads (N=200):
Time: 1.7450s, Residual: 0.401424         Time: 1.8679s, Residual: 0.001261



N=500
Jacobi with 1 omp threads (N=500):        G-S with 1 omp threads (N=500):
Time: 44.9468s, Residual: 168.599828      Time: 49.1777s, Residual: 89.181283

Jacobi with 2 omp threads (N=500):        G-S with 2 omp threads (N=500):
Time: 22.9619s, Residual: 168.599828      Time: 25.6149s, Residual: 89.181283

Jacobi with 3 omp threads (N=500):        G-S with 3 omp threads (N=500):
Time: 16.9541s, Residual: 168.599828      Time: 19.0541s, Residual: 89.181283

Jacobi with 4 omp threads (N=500):        G-S with 4 omp threads (N=500):
Time: 13.2977s, Residual: 168.599828      Time: 15.1846s, Residual: 89.181283

Jacobi with 5 omp threads (N=500):        G-S with 5 omp threads (N=500):
Time: 10.3660s, Residual: 168.599828      Time: 12.3430s, Residual: 89.181283

Jacobi with 6 omp threads (N=500):        G-S with 6 omp threads (N=500):
Time: 9.8882s, Residual: 168.599828       Time: 10.8510s, Residual: 89.181283

Jacobi with 7 omp threads (N=500):        G-S with 7 omp threads (N=500):
Time: 14.1339s, Residual: 168.599828      Time: 15.9501s, Residual: 89.181283

Jacobi with 8 omp threads (N=500):        G-S with 8 omp threads (N=500):
Time: 12.5955s, Residual: 168.599828      Time: 14.2905s, Residual: 89.181283

Jacobi with 9 omp threads (N=500):        G-S with 9 omp threads (N=500):
Time: 11.8108s, Residual: 168.599828      Time: 13.2888s, Residual: 89.181283
```

```
Jacobi with 10 omp threads (N=500):        G-S with 10 omp threads (N=500):
Time: 11.1602s, Residual: 168.599828       Time: 12.1275s, Residual: 89.181283

Jacobi with 11 omp threads (N=500):        G-S with 11 omp threads (N=500):
Time: 10.0776s, Residual: 168.599828       Time: 12.2237s, Residual: 89.181283

Jacobi with 12 omp threads (N=500):        G-S with 12 omp threads (N=500):
Time: 9.8357s, Residual: 168.599828        Time: 11.8345s, Residual: 89.181283
```