Spaceship Glider:

This is a game pattern we need to complete. Since the shape of pattern is repeating, we can implement it with any kind of shape in its series of patterns. In addition, the pattern will translate itself across the board, and finally it will disappear or degrade to the patterns of Still Lifes.

OpenMP:

There are two versions in the folder, one is implemented with OpenMP-Sections, and another one is implemented with OpenMP-for and single, which are commonly used.

Sections Version:

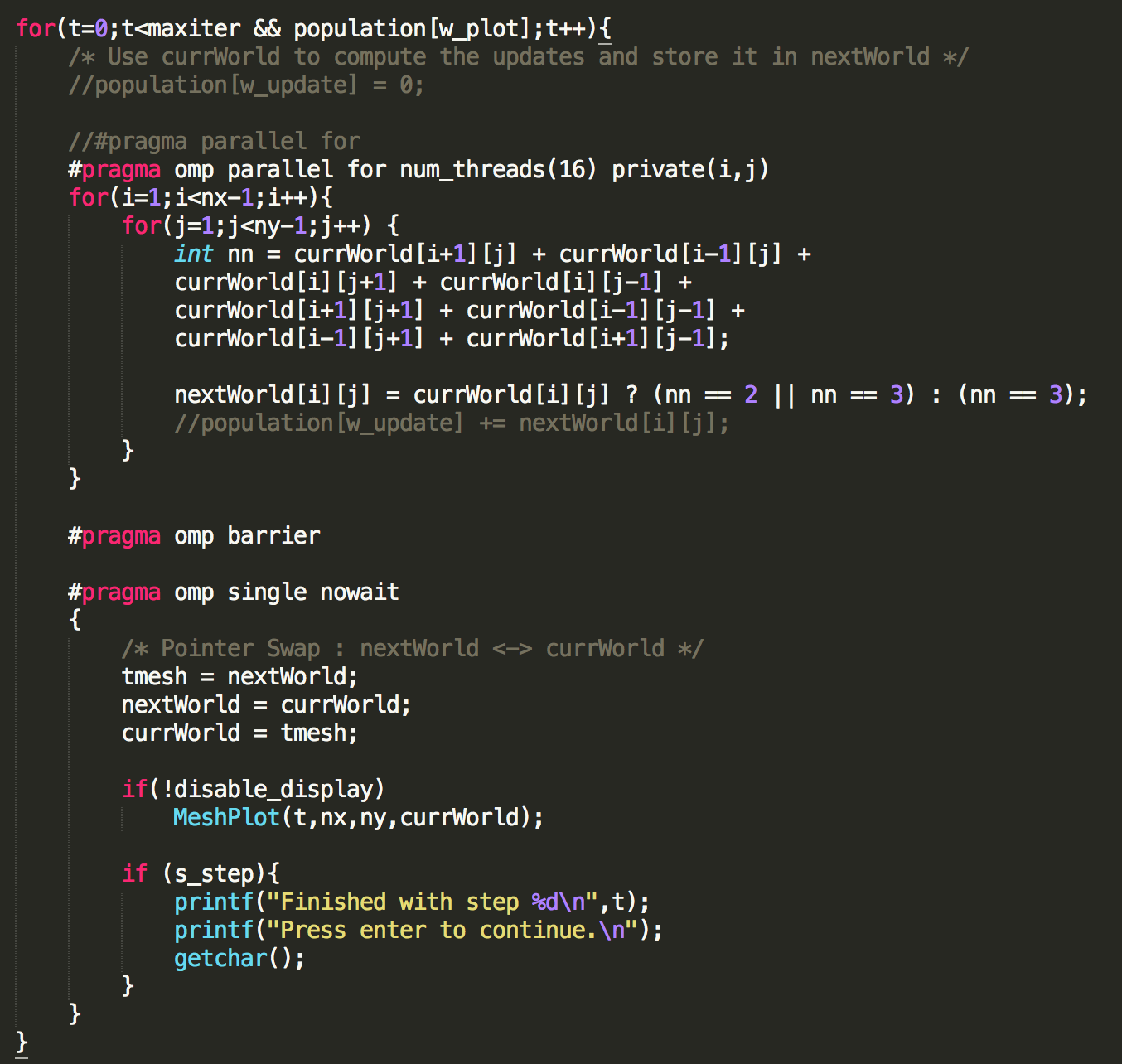
This version will get stuck since I couldn’t solve the problem of the synchronization between each section. There are two sections, one is responsible for updating the board and another is responsible for calling the plotting extension. Since we need to update the plot after each iteration, the synchronization between the sections become important. I tried to solve the problem with implementing a handcrafted mutex, but it didn’t work. Therefore, I have no choice but to give up the way to do it with omp section and tried to do it without omp section.

Common Version:

In this version of code, the program works fine, and also gets plenty of speedups. There is an outside for-loop to control number of iterations we want, and inside the for-loop there is an omp parallel for region, which is mainly responsible for updating each mesh point within the iteration.

Following the parallel for region, I placed a barrier over there. Although I think there is an implicit barrier after the parallel for region, I still added the barrier that will help us to understand the code. In addition, the barrier placed there plays an important role in task parallelism. Because we added the clause nowait at the following omp single region, the barrier will synchronize the plotting thread and the calculating threads that avoid the data-racing problem.

At the end is an omp single region, which is used to update the pointer and the graph, and assigning these kinds of work to only one thread is enough and necessary since we don’t want every thread to update the pointer which will be a mess.



* Condition n=250, iterations=100, p=0.2, seed=1234, test on cinekop

The time with display includes the work of x-window.

OpenMP with display 17.45s

OpenMP without display 0.23s

Ptherad with display 18.71s

Pthread without display 0.23s

Serial with display 17.72s

Serial without display 0.45s

* Condition n=500, iterations=300, p=0.2, seed=1234, test on cinekop

The time with display includes the work of x-window.

OpenMP without display 2.42s

Pthread without display 2.97s

Serial without display 5.11s

* Condition n=250, iterations=100, p=0.2, seed=1234, test on lufer

The time with display includes the works of x-window and network.

OpenMP with display 17.29s

OpenMP without display 0.08s

Ptherad with display 15.58s

Pthread without display 0.07s

Serial with display 18.13s

Serial without display 0.17s

* Test on Lufer, n=100, s=500, p=0.5

|  |  |  |  |
| --- | --- | --- | --- |
| Iters | 40000 | 400000 | 4000000 |
| Serial | 13.89s | 119.71s | 1033.4s |
| OpenMP 16t | 2.07s | 18.20s | 256.3s |
| Speedups | 6.71 | 6.57 | 4.03 |