CS484 MP1 report

PartA.

Summited to the gitlab.

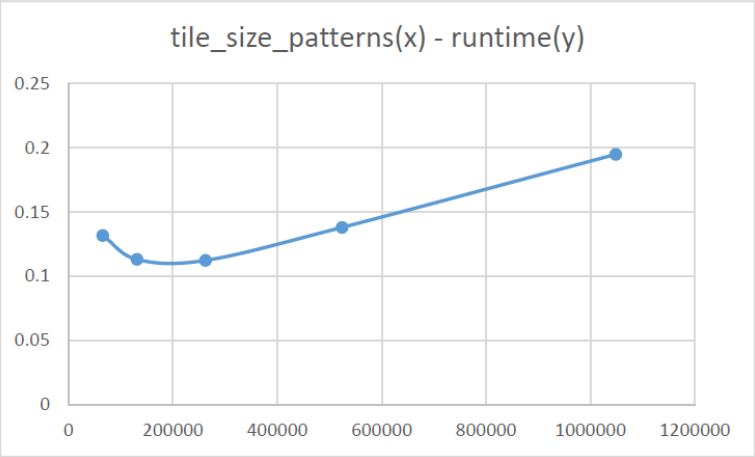
Briefly explain about the local synchronization:

First open a 2D array with p_asize * p_asize size, the most outer circle is all set to the MAXIT and the remain assize * assize set to zero. Before the cell being calculated, first check their four neighbor's iteration count. For red, if the cell's iteration is small or equal to their four neighbor, it can start calculated. For black, it should be smaller than its neighbor that it can start calculated.

PartB.

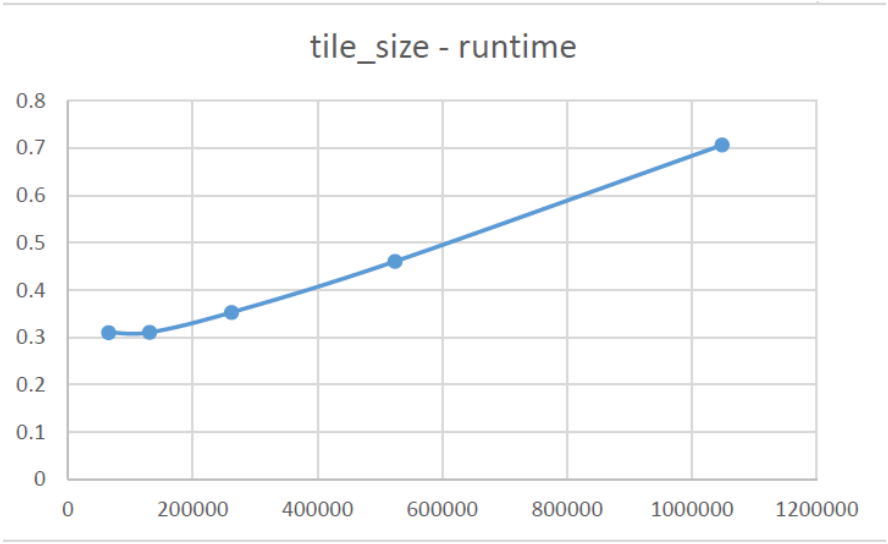
Analysis the result.

For a matrix size of 2048 and tile sizes [256 x 256, 512 x 256, 512 x 512, 1024 x 512, 1024 x1024] we plot at below:



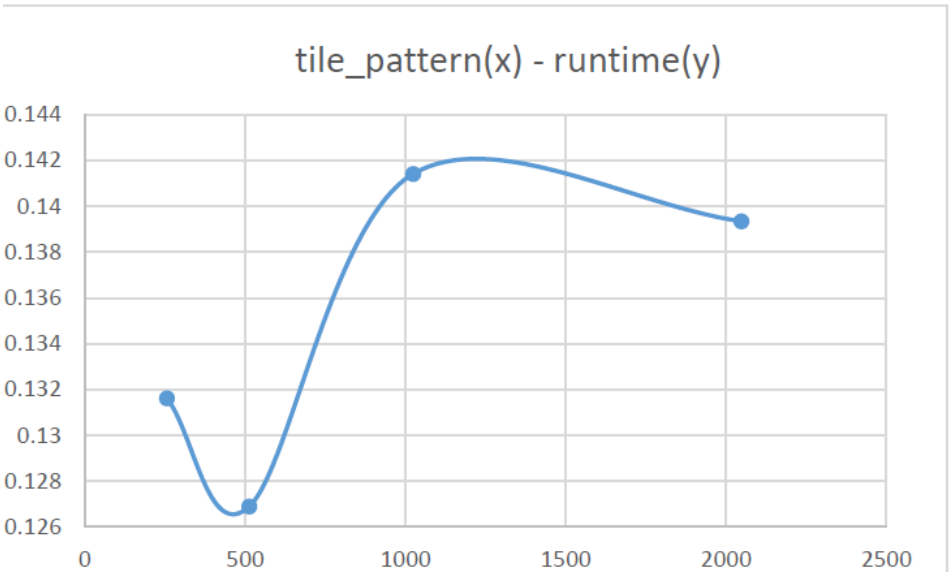
In this result, we can find out that when the thread begin to increase, the execution time begin to decrease but when the thread getting too many the execution time increase slightly.

The reason to this may be that the loading of the work is not that much so the overhead of synchronize may beat the multithread. So if we increase the work to each thread and make the thread number still the same the result may explain it. For a matrix size of 4096 and tile sizes [512 x 512, 1024 x 512, 1024 x 1024, 2048 x 1024, 2048 x2048]

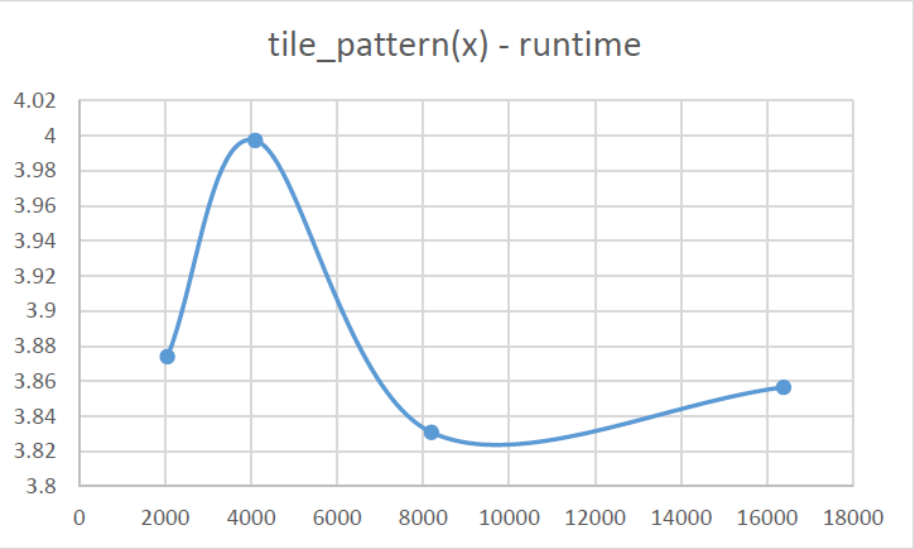


We found it that when the thread increase the runtime decrease and when the thread approach 64, the execution time is not increasing.

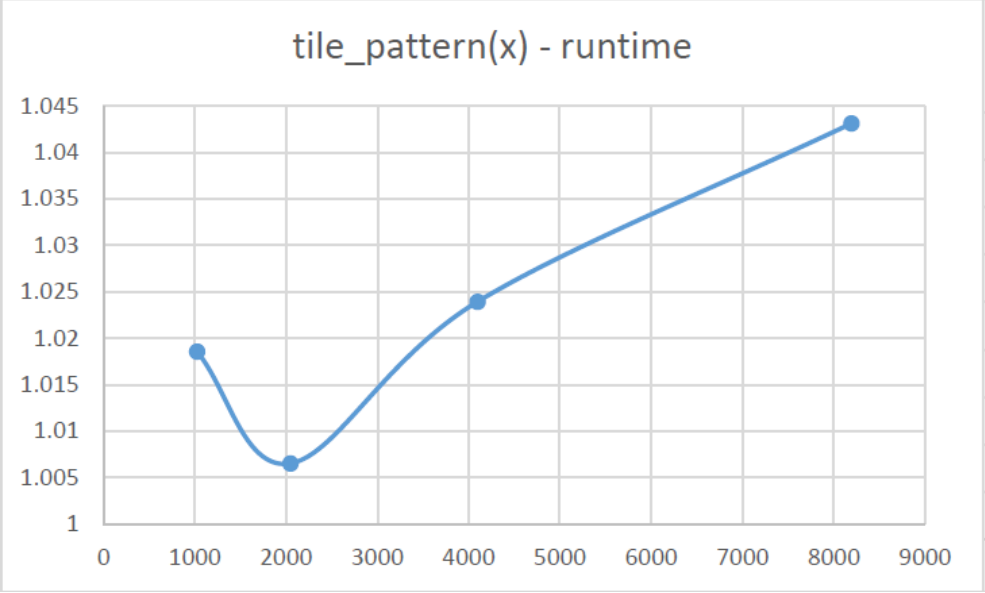
Another analysis is when we fix the thread number but change the tile_pattern, For a matrix size of 2048 and tile sizes [256 x 256, 512 x 128, 1024 x 64, 2048 x 32]



In the case the thread number is fixed to 64. But we can see that when the tile_x is 512 and the tile_y is 256 the performance is the best. And when the tile_x equal 1024 and tile_y equal 128 the performance is the worst.
Well I have done another size of test. For a matrix size of 16384 and tile sizes [2048 x 2048, 4096 x 1024, 8192 x 512, 16384 x 256] which is also have 64 thread to all the case but the result is completely different.



Which the 4096 x 1024 has the worst performance and 8192 x 512 has the best performance, which make me really confuse. So I decide to do another test For a matrix size of 8192 and tile sizes [1024 x 1024, 2048 x 512, 4096 x 1024, 8192 x 512]



It looks much similar to the first one.