Project Milestone Report

Home

Title

Parallel 2048 Solver (Wei-Ting Tang)

URL

https://white123.github.io

Summary

So far, I have built the sequential implementation and parallelized the most outer loop using OpenMP. This enables the program to simulate four directions independently. Using the sequential implementation as the benchmark, it takes 26.4 second on average to complete the first 300 steps, running on GHC machines.

Our naive implementation using 4 threads has the average time of 11.3 second (2.34 speedup). This is an acceptable result since there are still some other works should be done sequentially. However, it is hard to apply the same strategy on a higher-count cores machine. Although the parallelism works well on the first round (depth = 0), the program doesn't improve or even slower when I try to parallelize the second round using higher thread count. In the following week, I will focus on how to evenly assign the tasks to the threads while avoid creating too many tasks.

Deliverables

The deliverables still remain unchanged, but I need to figure out a way to distribute tasks on more cores to reach the "nice to have" goal.

In the poster session, I will show the graphs comparing the solving speed using different thread counts and live demo the program solving the 2048 game at an extremely high speed.

Schedule

I have implemented the naive parallel solver and done simple benchmarks that align with the proposal's schedule. However, parallelizing the recursive code using OpenMP is harder than I thought, so I might need to spend more time on making it works.

Week	Plan
Bi-week 1 (11/30 - 12/3)	Parallelize the second round and run on high-count core machines
Bi-week 2 (12/4 - 12/6)	Analyze the bottleneck and improve workload balance
Bi-week 3 (12/7 - 12/10)	Fine-tune the program to reach the deliverable
Bi-week 4 (12/11 - 12/13)	Finalize the program and apply different experiments
Bi-week 5 (12/14 - 12/17)	Finish and submit the final report