

Sudoku Solver Milestone Report

Team Members: Madeleine Hagar and Jennifer Chou

COMPLETED WORK: We have researched and tested different sequential algorithms. The main algorithms we considered were the naive algorithm, backtracking, and cross-hatching with backtracking. We decided to parallelize the regular backtracking algorithm. The naive algorithm seemed very straightforward, but we felt that since backtracking was a more optimized version of the sudoku solver, we would want to parallelize that instead for better performance. We didn't think the cross-hatching with backtracking would be a good choice because this algorithm was an optimization of backtracking that identified elements that are most filled for better pruning, but if we were using parallelization, we would need to keep track of with elements are already being checked to do it in order, so it might be simpler to do parallelization of regular backtracking. We modified our starter sequential algorithm so that it would apply to different sized sudoku puzzles and came up with different tests with timing calculations for 9x9 and 16x16 sized boards. We did a lot of research on different parallel algorithms for a sudoku solver and started on an OpenMP implementation as well as a Cuda implementation. We also looked into the possibilities of using MPI for a solver. We have uploaded our work so far to our github repository: <https://github.com/jcchou12/Sudoku-Solver>

PROGRESS: We are a little bit behind where we would have liked to be due to a set back in the recursive calls in the OpenMP implementation. We needed to change our algorithm drastically to use different data structures because the performance of the OpenMP implementation we had didn't improve at all. We still think we would be able to produce all of our deliverables, but we are not sure about the "nice to haves" anymore. Other than that, we have completed all other goals from our proposal.

SCHEDULE AND GOALS:

½ Weeks	Goals
April 16 - April 19	<ul style="list-style-type: none">• Complete Project Milestone Report - Jennifer and Maddie
April 20 - April 22	<ul style="list-style-type: none">• Complete first parallel algorithm - Jennifer• Complete analysis on first parallel algorithm - Maddie• Continue with the second parallel algorithm - Maddie
April 23 - April 26	<ul style="list-style-type: none">• Complete second parallel algorithm - Maddie• Do analysis on second parallel algorithm - Jennifer
April 28 - April 30	<ul style="list-style-type: none">• Do a comparison analysis on the two parallel algorithms - Jennifer and Maddie
May 1 - May 5	<ul style="list-style-type: none">• Complete Final Report (due Thursday, May 4) - Jennifer and Maddie• Prepare for Poster Session (Friday, May 5) - Jennifer and Maddie• Create the poster - Maddie• Create the demo/graph - Jennifer

POSTER SESSION: We would like to show a demo of the sudoku solver as it solves the board, but we are not completely sure if this is feasible in the time we have since it seems difficult to do and may slow down the performance by a lot which could slow down testing. If we are low on time, we will show a graph of the different speedups of different boards for our 3 algorithms.

PRELIMINARY RESULTS:

	Solvable - Random	Solvable - Sparse	Unsolvable - Random	Unsolvable - Sparse
Sequential 9x9	0.000778499	0.00307145	0.000207506	0.000259454
Sequential 16x16	0.00029953	0.00491142	0.000178818	75.194

ISSUES: We ran into some issues with parallelizing the recursive section of the sequential backtracking algorithm. We had initially thought we could parallelize that in OpenMP by just sharing a variable to check when the recursion should stop for all threads. However, with this method, we found that the performance did not improve at all, possibly due to the increase in shared data. This set us back quite a bit because we weren't expecting no performance improvement at all and needed to change our algorithm. Other than that, we feel like it is just a matter of coding and continuing doing the work and research.