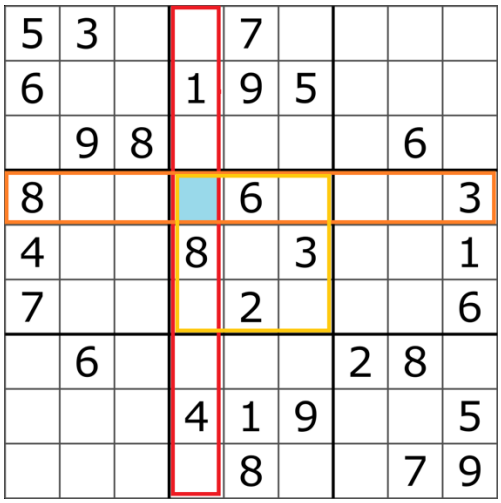
CSE531 Programming Lab 1: OpenMP

Due: Feb 23, 23:00

In this assignment, we are going to implement a parallel Sudoku solver using OpenMP.

Problem Description

Sudoku is a logic-based number placement puzzle. The goal is to fill a 9x9 grid with digits so that each column, each row, and each of the nine 3x3 sub-grids that compose the grid contain all of the digits from 1 to 9. The puzzle starts with some grid cells filled with numbers, and the rest are blank. Note that not all Sudoku puzzles have a solution or have a unique solution.



A simple Sudoku solver can be implemented by the following backtracking algorithm:

```
1. Find an empty cell. If there are none, return solved
2. Assign it a number (1-9)
3. Check Sudoku
3.1 If valid, goto 1
3.2 If invalid, goto 2 and pick another number
3.3 If invalid and all numbers have been tried, backtrack
3.4 If there is nowhere to backtrace, return no solution
```

Input / Output Format

1. The following files are provided on Canvas:

```
main.cc       # you can adjust benchmark options for scalability plots
benchmark.h   # DO NOT MODIFY
solver_seq.cc # DO NOT MODIFY: sequential version
solver_omp.cc # TODO: a template for your parallel version
Makefile      # TODO: modify the run rule
dataset_easy.txt
dataset_hard.txt
dataset_benchmark.txt
```

2. The program accepts 1 (or more) parameters: `./executable $dataset`. It will output the benchmark result of the solver:

```
$ ./solver_seq dataset_easy.txt dataset_benchmark.txt
Benchmark of dataset_easy.txt
Validation walltime (usec): 13821
Benchmark walltime (usec): 10000036
793997 board/sec
1.25945 usec/board
Benchmark of dataset_benchmark.txt
Validation walltime (usec): 12147712
Benchmark walltime (usec): 12141244
823 board/sec
1214.12 usec/board
```

3. Input: The input data preprocessing is already handled in `benchmark.h`. But for your reference, each line of the dataset is the row-major representation of one 9x9 Sudoku board except the line starting with `#`, and empty tiles are represented by `.`.
4. Output: We only need to implement the following function in the solver. If there is a solution, write to `solution` and return 1; otherwise, return 0. If there are multiple solutions, return **any valid solution**. Note that `solution` has already allocated 81 characters.

```
extern "C" size_t Solver(const char *input, char *solution)
```

Report

The report must contain the following:

1. Title, name, PSU ID
2. Explain your implementations in the following aspects:
   - How do you partition the task?
   - What technique do you use to reduce execution time or increase scalability?
3. Experiment & Analysis
   - System & compiler specs (e.g., DevCloud Xeon 6128 GCC 7.4.0)
   - Benchmark results on the **easy, hard, and benchmark dataset**
     (0) sequential version
     (1) parallel version using **8 OpenMP threads of Xeon 6128 on DevCloud**
     (2) your best result using any number of threads on any platform
   - Scalability plots
     (1) fixed the problem size and increase threads
     (2) fixed threads and increase the problem size
   - Each plot must contain at least 4 data points, and make sure your plots are properly labeled and formatted.
   - Discussion based on plots.
   - Any other discussions or analyses are encouraged. Make sure to explain how and why you do these experiments.

Rubrics

1. Correctness (50%)

   - Easy (10%) Hard (20%) Benchmark (20%) dataset
   - Your implementation should pass the validation.
   - Your implementation should be faster (or not significantly slower) than the sequential version.

2. Performance (15%): Based on the fastest version using **8 threads of Xeon 6128 on DevCloud under the benchmark dataset** among all students.
3. Report (35%)

Submission

Upload these files to Canvas:
Please do not upload any dataset!
Any corrupted files will be regarded as a failure of submission.

```
Makefile
solver_omp.cc
Lab1_Report.pdf
```

Reminders

1. Since we have limited resources, please start your work ASAP. Do not leave it until the last day!
2. Copying any codes from the Internet is not allowed, but discussions are encouraged.
3. Office hour holding by Scott: Tuesday 15:00-16:00 @ Westgate Bldg W341