Parallel Sudoku Generator and Solver with OpenMP and MPI

Zhongyi Cao, Liyao Fu

3	7	9	2	4	5	8	6	1
2	8	5	3	6	1	9	7	4
1	6	4	9	7	8	2	3	5
9	5	8	6	3	4	1	2	7
4	3	7	1	9	2	6	5	8
6	2	1	8	5	7	4	9	3
7	9	2	4	1	3	5	8	6
8	4	3	5	2	6	7	1	9
5	1	6	7	8	9	3	4	2

BACKGROUND

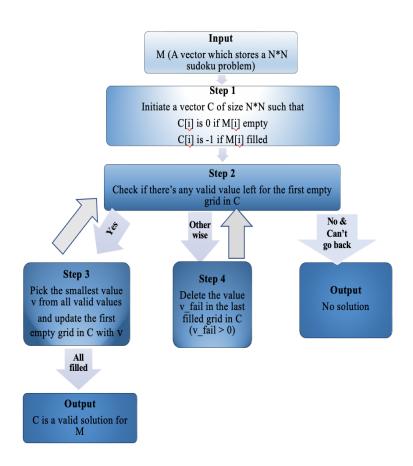
Sequential Sudoku Generator Design

- 1) Random
- 2) Valid
- 3) Solvable

Input M (number of sudokus we need) N (number of empty grids in sudoku) S (size of sudoku) Step 1 For each of the sudoku, initiate an SxS empty matrix and take step 2 – step 4 until all grids in the matrix is filled Step 2 Check if there's any valid value for the first empty grid Step 3 Step 4 Randomly pick one value v from all valid Delete the value in the last filled grid values and fill the first empty grid to v All filled Step 5 For each of the sudoku, remove N grids generated sudoku randomly **Output** M generated sudokus

Sequential Sudoku Solver Design

- 1) Row
- 2) Column
- 3) Subgrid



APPROACH 1 — OpenMP Generator & Solver Design

OpenMP Generator Design

Input

M (number of sudokus we need)
N (number of empty grids in sudoku)
S (size of sudoku)

Step 1

[OMP parallel] each of the sudoku, initiate an SxS empty matrix and take step 2 – step 4 until all grids in the matrix is filled

Step 2 [OMP parallel]check if there's any valid value for the first empty grid Step 3 Randomly pick one value v from all valid values and fill the first empty grid to v Step 4 Delete the value in the last filled grid

Step 5

[OMP parallel] each of the sudoku, remove N grids generated sudoku randomly

Output

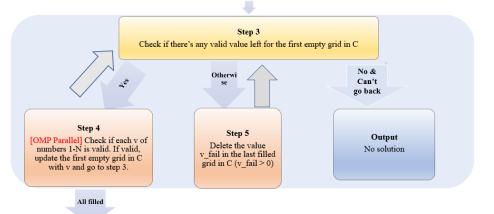
M generated sudokus

OpenMP Solver Design

Input M_1, M_2, ...M_n (n vectors each of which stores a s*s sudoku problem)

value j appears in row/column/sub-sudoku i

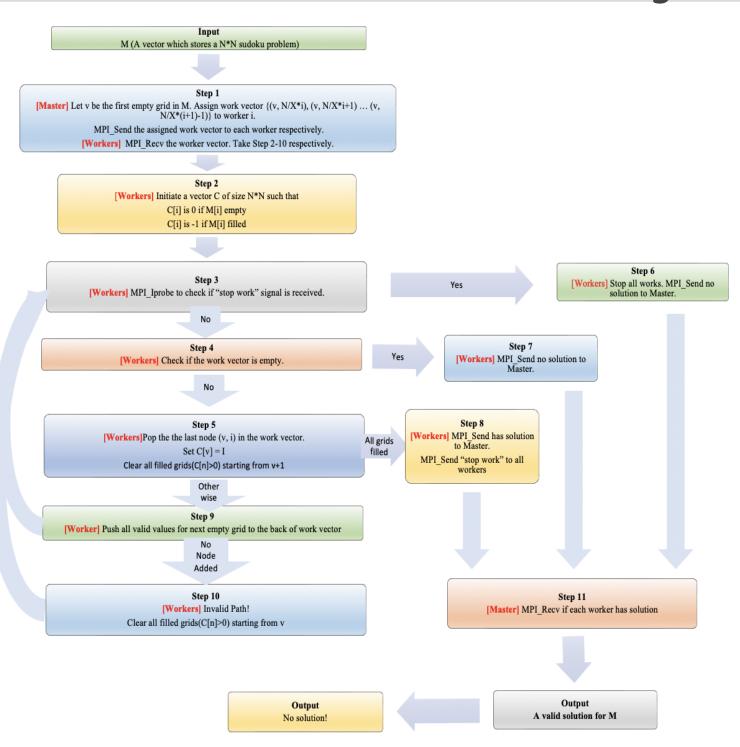
 $\label{eq:Step 1} Step \ 1$ Initiate 3 vectors of size n*s*s to such that a[k][i][j] represent in matrix k, if each



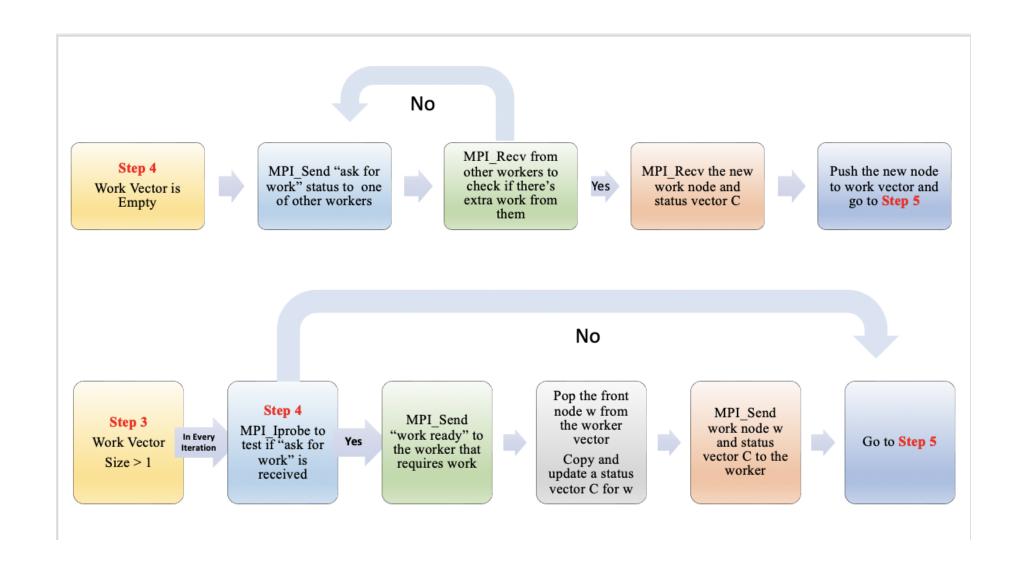
Output

C is a valid solution for M i

APPROACH 2.1—MPI Solver Design

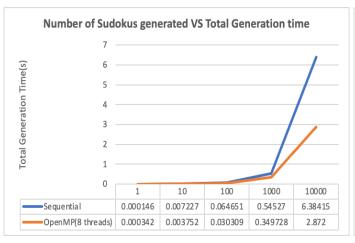


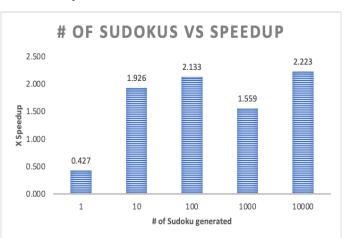
APPROACH 2.2—Work Stealing in MPI Solver



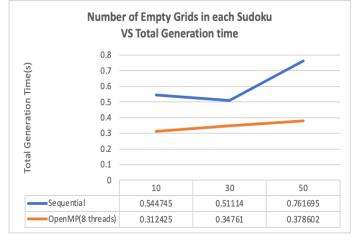
Result 1.1 — OpenMP Generator

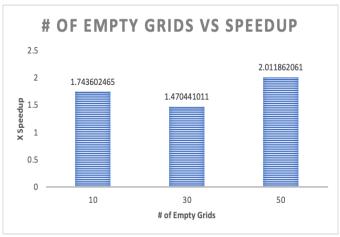
(Fix sudoku with 30 empty grids, 8 cores)





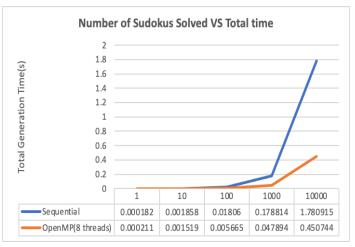
(Fix number of sudokus = 1000, 8 cores)

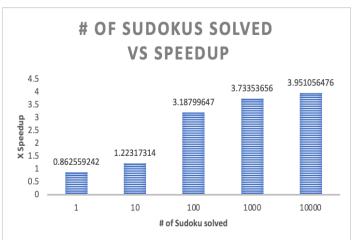




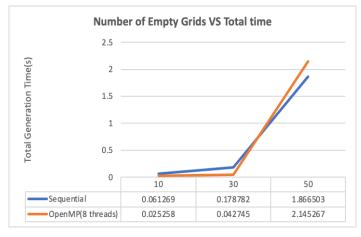
Result 2.1 — OpenMP Solver

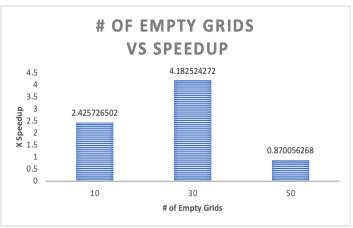
(Fix sudoku with 30 empty grids, 8 cores)





(Fix number of sudokus = 1000, 8 cores)





Result 2.2 — MPI Solver

Version 1 (No work stealing)

Version 2 (With work stealing)

