Parallel Sudoku Solver

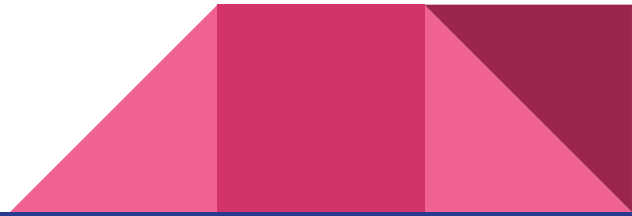
BY:
170050020 Kiran Dapkar
170050046 Nirmal Rajput
170050054 Chaw Akametta Enling
170050061 Sameer Prajapati

About Sudoku

- Standard Sudoku is a classic puzzle where the user must fill in a 9x9 grid with numbers such that each row, column and 3x3 grid will have the numbers 1 through 9 occur only once.
- Some Sudoku puzzles can go beyond 9x9 grids, but our implementation will focus specifically on the 9x9 size.
- All Sudoku puzzles will have some given numbers on them, and the difficulty of the puzzle varies depending on how many numbers are given, as well as the location of the numbers that are given.

Challenge

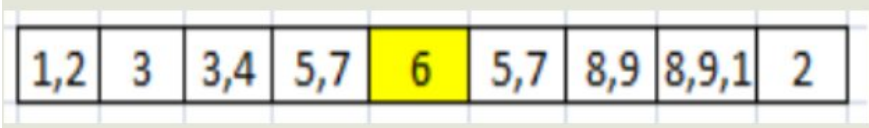
- This problem is challenging because for a standard 9x9 grid Sudoku puzzle, there are 6.671×10^{21} different possible solutions.
- Even if the player is given around 20 numbers, that still leaves a lot of solutions that need to be permuted in order to find the one correct solution.
- It is proven that Sudoku solving is NP-complete problem



Humanistic Approach: Strategies

1. **Elimination:**

   This occurs when there is only one valid value for a cell. Therefore, that value must belong in the cell.

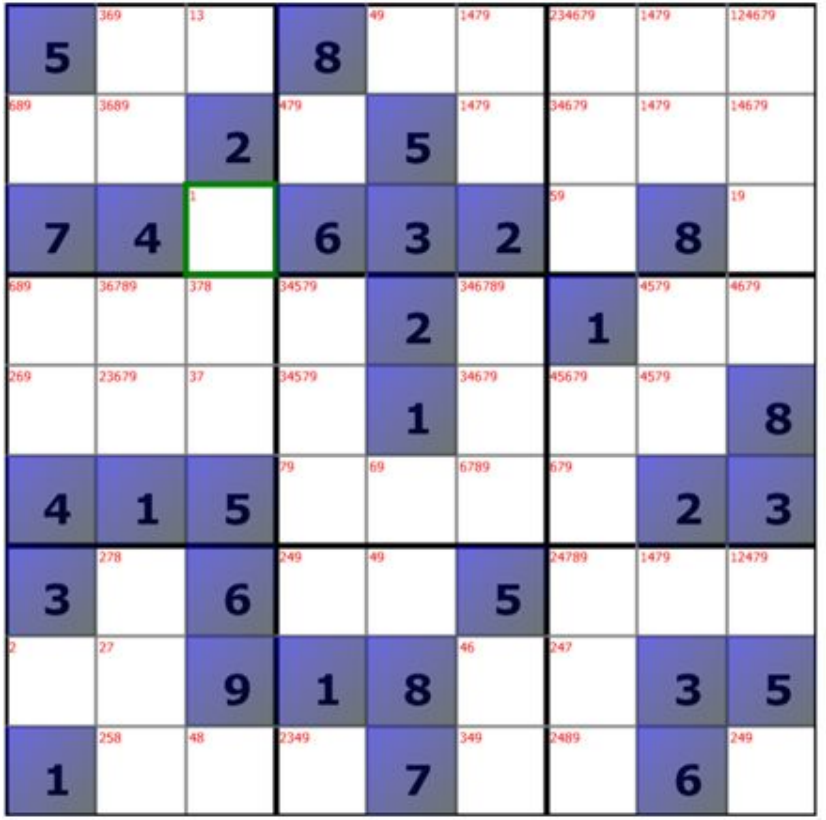


1,2	3	3,4	5,7	6	5,7	8,9	8,9,1	2

Humanistic Approach: Strategies

1. **Elimination:**

The figure here shows the highlighted cell in green which could be assigned one possible value that happens to be 1 too since values 2,4,5,7 are in the same minigrid.
And also 3, 6, 8 are on the same row and 9 is on the same row.



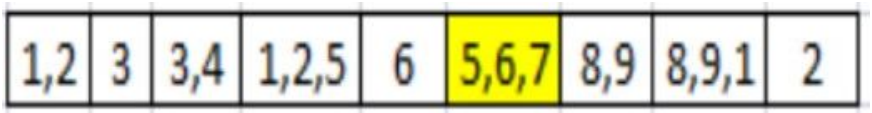
Humanistic Approach: Strategies

2. LONE RANGER:

This is a number that is valid for only one cell in a row, column, or box.

There may be other numbers that are valid for that particular cell,

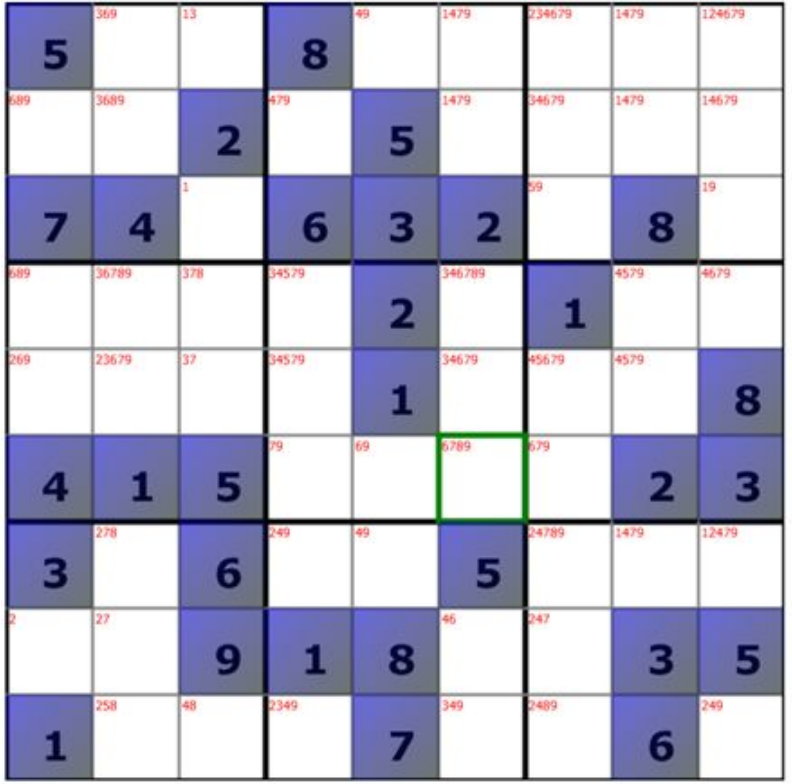
But since that number cannot go anywhere else in the row, column, or box, it must be the value for that cell.



Humanistic Approach: Strategies

2. LONE RANGER:

- The green cell shown in the figure here has 4 possible values.
- Intuitively, a human solving this grid will most likely to skip this cell because it seems “too early to predict its value” because of the several values possible.
- But if we take a closer look at the cell’s row, note that the number 8 appears only once in the row and it appears as a possible value for that cell.
- Hence, 8 is indeed the correct value for that cell.



Strategies Contd.

3. TWINS:

These are a pair of numbers that appear together twice in the same two cells and only in those two cells. When twins are found, all other numbers can be eliminated as possible values for that cell.

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

The top row depicts a row where the highlighted cells contain twins. The bottom row shows the possible values left for those cells after the twins rule has been applied. In the top row, 5 and 7 show up together twice and only twice in the row. Because they are twins, the 3 and 6 that were valid values for those highlighted cells can now be eliminated as possible values.

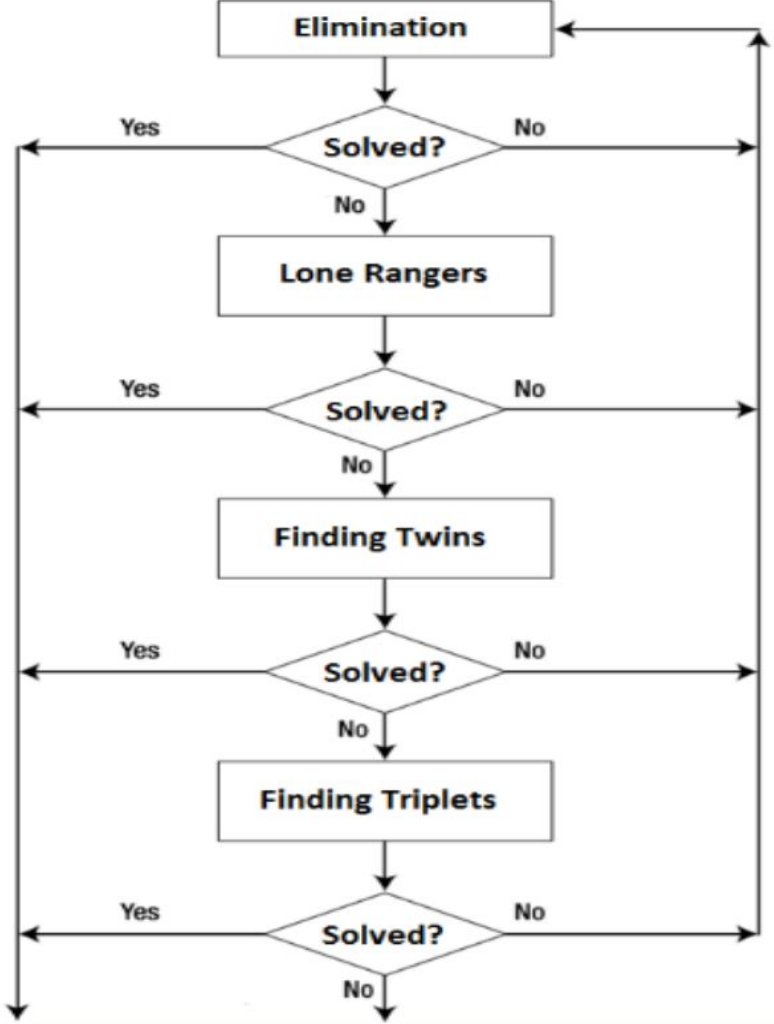
Strategies Contd.

4. TRIPLETS:

Triplets follow the same rules as twins, except with three values over three cells.

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

The top row depicts a row of a sudoku board before triplets is applied. The bottom row is the resulting valid values after triplets has been applied. In the top row, 1, 2, and 4 only appear three times in the row and they all appear in the exact same three cells, so they are triplets.


Elimination
Yes
Solved?
No
No
Lone Rangers
Yes
Solved?
No
No
Finding Twins
Yes
Solved?
No
No
Finding Triplets
Yes
Solved?
No
No

Brute Force Algorithm

- When all logical means have been used to solve a Sudoku puzzle and the puzzle remains unsolved, we performed some Brute Force algorithm.
- Brute force algorithm uses depth first search approach.
- First, we fill in the first empty cell with the smallest valid number.
- Then, we try to fill in the next empty cell in the same way.
- If we reach an empty cell that does not have any valid numbers, we backtrack to the most recently filled cell and increment its number by 1.
- If the number cannot be incremented, then we backtrack again.

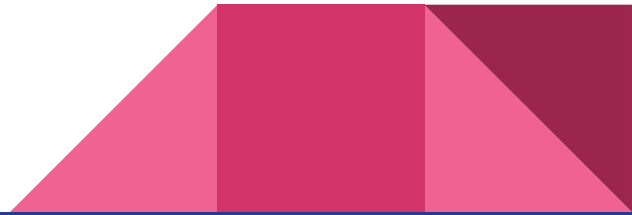
Algorithm

- Parallelizing by box for each strategy.
- First, elimination is run in parallel across all boxes. Once it’s done then lone rangers will run in parallel across all boxes, and so on.
- Finding possible values using row & column scan and minigrid scan
- These 2 functions set to run in parallel at the same time because they are not dependant on each other
- However, a lock needs to be set on the possible values of the cell.

Algorithm

Brute force algorithm uses depth first search approach.

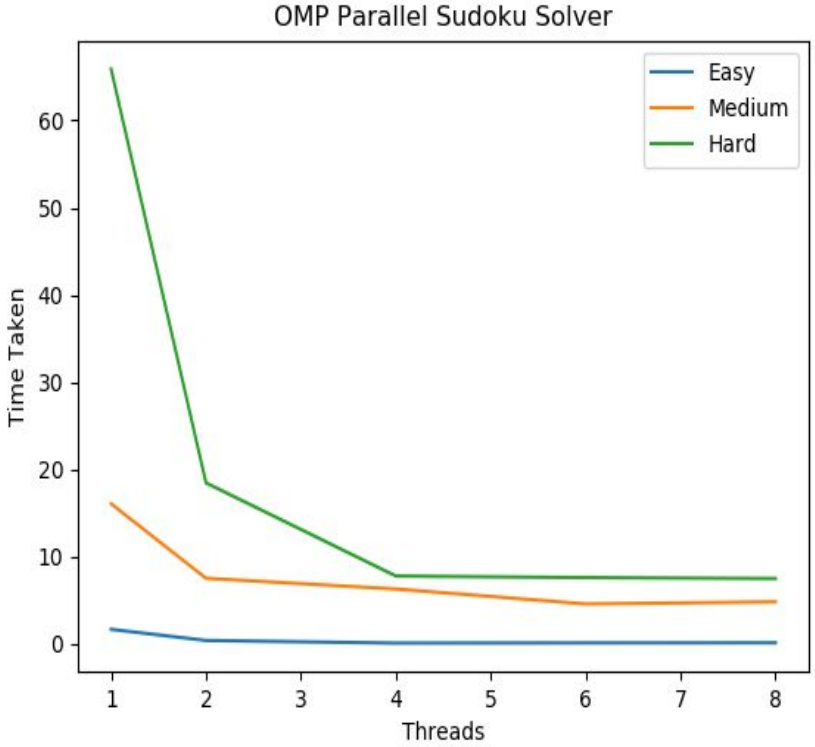
- First, we fill in the first empty cell with the smallest valid number.
- Then, we try to fill in the next empty cell in the same way.
- If we reach an empty cell that does not have any valid numbers, we backtrack to the most recently filled cell and increment its number by 1.
- If the number cannot be incremented, then we backtrack again.
- We continue doing this until there are no more empty cells on the board.
- This means we found the solution and we return it. Or we backtracked to the first unfilled cell.

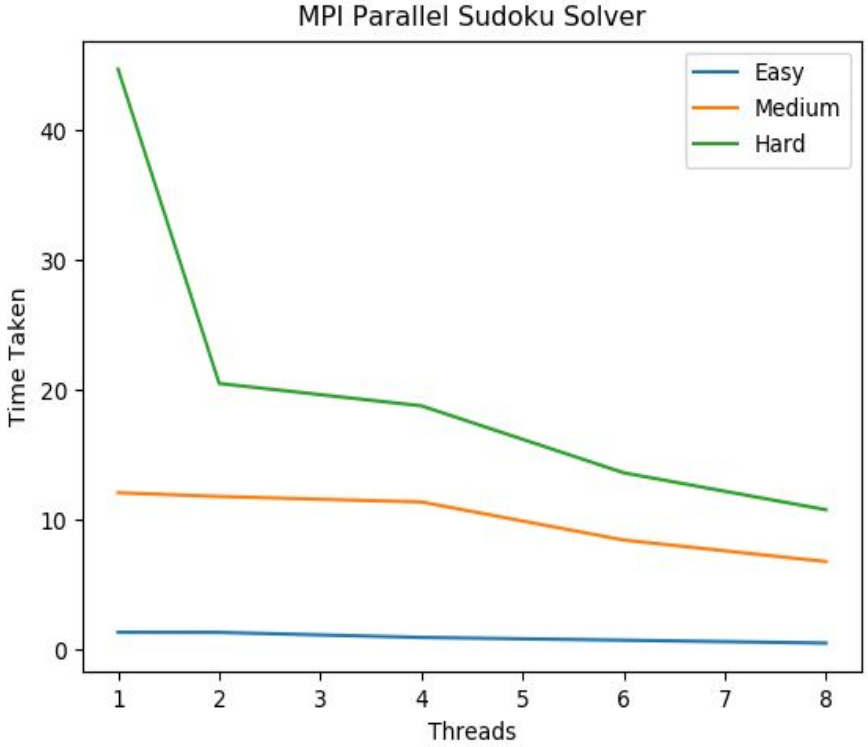


Algorithm

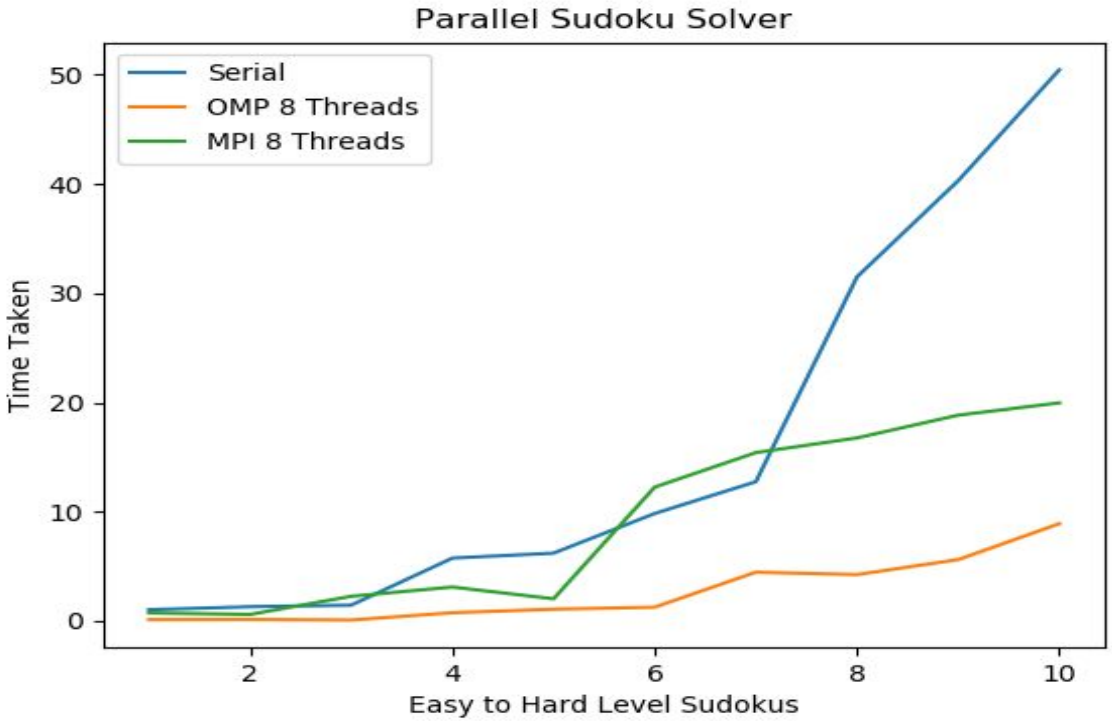
- In this case, no solution exists for the board. We then combine our serial brute force algorithm with a shared stack.
- We create a board for all permutations of valid numbers of the first 7 cells and fill in those 7 cells with the permutations. We then push all those boards onto the stack.
- Next, multiple threads pop from the stack in parallel and try to find a solution using brute force. If a partial board popped from the stack isn't the solution, the board is discarded and the thread pops another one off the stack.
- The first thread to find a solution will abort all other threads.
- Finally, the main thread will return either the solution, or no solution if the stack is empty.

Results


OMP Parallel Sudoku Solver
Easy
Medium
Hard
Time Taken
0
10
20
30
40
50
60
1
2
3
4
5
6
7
8
Threads


MPI Parallel Sudoku Solver
Easy
Medium
Hard
Time Taken
0
10
20
30
40
1
2
3
4
5
6
7
8
Threads

Results


Parallel Sudoku Solver
Serial
OMP 8 Threads
MPI 8 Threads
Time Taken
0
10
20
30
40
50
2
4
6
8
10
Easy to Hard Level Sudokus

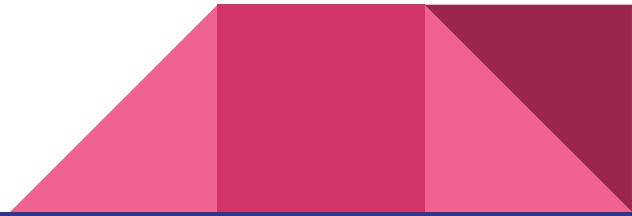
Conclusion

- We ran our serial sudoku solver, OpenMp Sudoku solver and MPI Sudoku Solver with some of the hard & medium level sudokus.
- Parallelising sudoku solver using OpenMp gives us the results 4 to 6 times much faster than serial version since data sharing is much better in OpenMp.
- With MPI the result in 3 to 4 times faster than the serial version

Contributions

1. **Sameer Prajapati and Kiran Dapkar:**
    a. OpenMp and Serial Version of sudoku solver
    b. Report writing
    c. Code formatting and improvising

2. **Akametta and Nirmal Rajput:**
    a. MPI version of Sudoku Solver
    b. Slides, references
    c. debugging



References

- https://alitarhini.wordpress.com/2011/04/06/parallel-depth-first-sudoku-solver-algorithm/
- http://en.wikipedia.org/wiki/Sudoku_solving_algorithms#Brute-force_algorithm
- https://alitarhini.wordpress.com/2012/02/27/parallel-sudoku-solver-algorithm/