

# Programming OpenMP

## *Tasking Introduction*

**Christian Terboven**



# *Tasking Motivation*

# Sudoku for Lazy Computer Scientists

- Lets solve Sudoku puzzles with brute multi-core force

	6					8	11			15	14			16	
15	11				16	14			12			6			
13		9	12					3	16	14		15	11	10	
2		16		11		15	10	1							
	15	11	10			16	2	13	8	9	12				
12	13			4	1	5	6	2	3				11	10	
5		6	1	12		9		15	11	10	7	16		3	
	2				10		11	6		5		13		9	
10	7	15	11	16				12	13					6	
9						1			2	16	10			11	
1		4	6	9	13			7		11		3	16		
16	14			7		10	15	4	6	1				13	8
11	10		15				16	9	12	13			1	5	4
		12		1	4	6		16				11	10		
		5		8	12	13		10			11	2			14
3	16			10			7			6				12	

- (1) Search an empty field
- (2) Try all numbers:
  - (2 a) Check Sudoku
    - If invalid: skip
    - If valid: Go to next field
- Wait for completion

# Parallel Brute-force Sudoku

- This parallel algorithm finds all valid solutions

	6					8	11			15	14			16	
15	11				16	14			12			6			
13		9	12					3	16	14		15	11	10	
2		16		11		15	10	1							
	15	11	10			16	2	13	8	9	12				
12	13			4	1	5	6	2	3				11	10	
5		6	1	12		9		15	11	10	7	16		3	
	2				10		11	6		5		13		9	
10	7	15	11	16				12	13					6	
9						1			2	16	10			11	
1		4	6	9	13			7		11		3	16		
16	14			7		10	15	4	6	1				13	8
11	10		15				16	9	12	13			1	5	4
		12		1	4	6		16				11	10		
		5		8	12	13		10			11	2			14
3	16			10			7			6				12	

- (1) Search an empty field

```
#pragma omp parallel
#pragma omp single
such that one task starts the
execution of the algorithm
```

- (2) Try all numbers:

- (2 a) Check Sudoku

- If invalid: skip

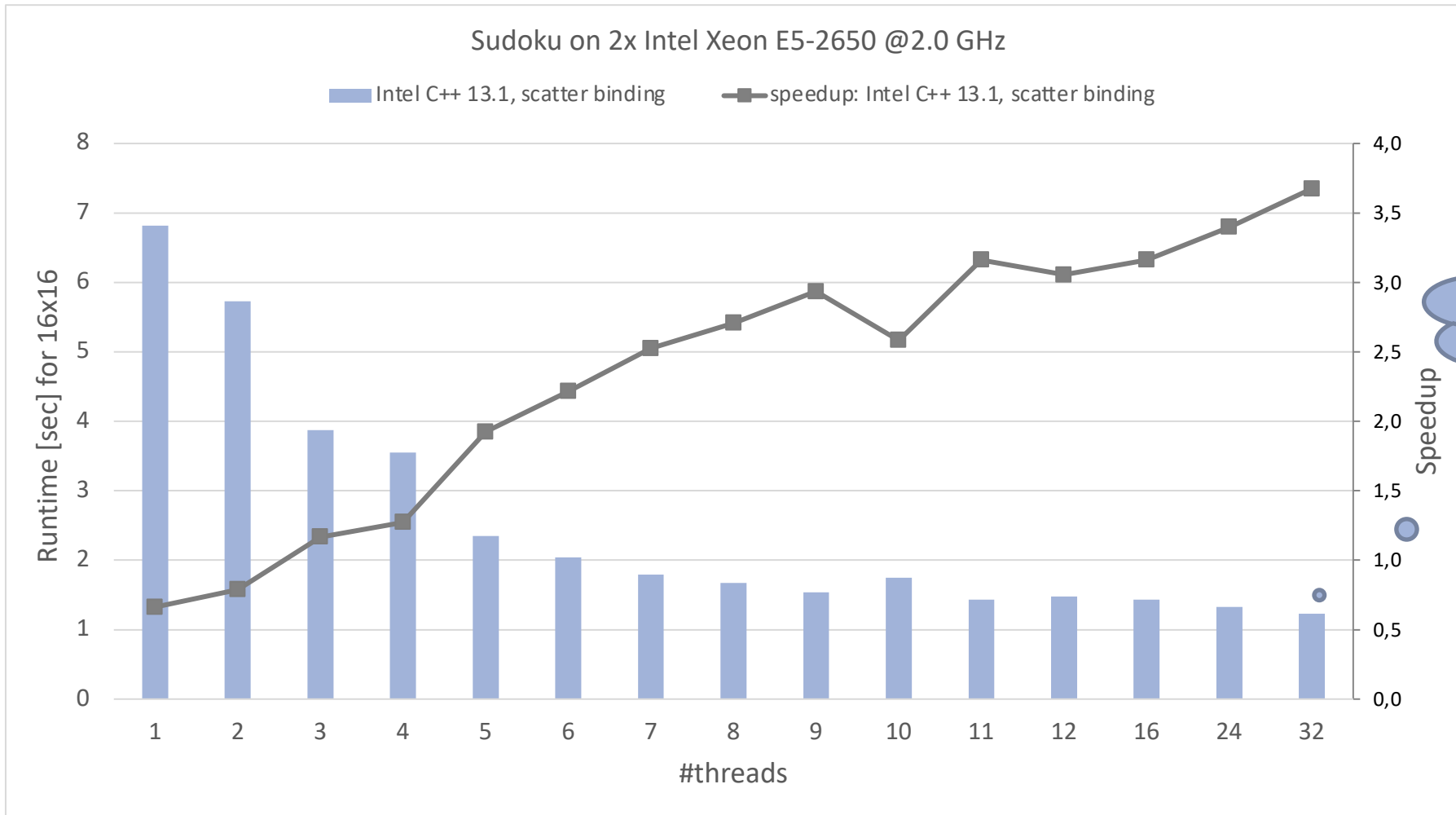
- If valid: Go to next

```
#pragma omp task
needs to work on a new copy
of the Sudoku board
```

- Wait for completion

```
#pragma omp taskwait
wait for all child tasks
```

# Performance Evaluation



Is this the best we can do?