

# High Performance Computing Proseminar 2024

## Assignment 3

Stefan Wagner & Sebastian Bergner  
Team: Planning Disaster

October 20, 2024

The goal of this assignment is to extend the heat stencil application and measure its performance.

### Exercise 1

This exercise consists in extending the heat stencil application of Assignment 2 to two dimensions.

- Extend the heat stencil application to the two-dimensional case and name it `heat_stencil_2D`. Provide a sequential and an MPI implementation. Try to make your implementation as efficient as possible, also with regard to code readability.

```
1 01\_heat\_stencil\_2D/src/heat\_stencil\_2D\_seq.c
2 01\_heat\_stencil\_2D/src/heat\_stencil\_2D\_par\_blocking.c
```

- Run your programs with multiple problem and machine sizes and measure speedup and efficiency. Consider using strong scalability, weak scalability, or both. Justify your choice.

#### – Strong vs Weak Scaling:

- \* In **strong scaling**, the problem size remains constant while the number of processors is increased. This allows us to measure how much faster the computation becomes with additional resources.
- \* In **weak scaling**, the number of processors and the problem size are expanded proportionally, ensuring that each processor continues to handle the same amount of work.
- \* We use **strong scaling** because our focus is on improving the performance of a fixed problem size. In practical applications like the 2D heat stencil, the problem size often remains constant, and our goal is to minimize the computation time by utilizing more processors. Strong scaling allows us to evaluate how efficiently the application distributes a fixed workload across multiple processors, which is the key to enhancing performance in real-world scenarios.

#### – Strong scaling speedup & efficiency:

$$S(p) = \frac{T_s}{T_p} \quad E(p) = \frac{S(p)}{p}$$

where  $T_s$  is the sequential execution time and  $T_p$  is the parallel execution time with  $p$  processors.

- Illustrate the data in appropriate figures and discuss them. What can you observe?

As demonstrated in Table 1 and Figure 1, the parallel blocking version consistently outperforms the sequential version across all configurations and problem sizes. However, as problem sizes increase, while the speedup improves significantly, the efficiency declines, as illustrated in Figures 2 and 3.

Results of 2D Heat Stencil Execution												
Impl/Ranks	Problem Size											
	192				384				768			
	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)
seq/1	2.81	0.04	1.00	1.00	22.14	0.17	1.00	1.00	176.03	1.06	1.00	1.00
par_block/2	2.54	0.04	1.11	0.55	19.95	0.14	1.11	0.55	158.43	0.86	1.11	0.56
par_block/6	0.91	0.03	3.08	0.51	6.91	0.03	3.20	0.53	54.62	0.31	3.22	0.54
par_block/12	0.55	0.01	5.14	0.43	3.61	0.09	6.13	0.51	28.05	0.44	6.28	0.52
par_block/24	0.45	0.01	6.31	0.26	2.01	0.02	11.02	0.46	14.48	0.17	12.16	0.51
par_block/48	0.40	0.01	7.11	0.15	1.18	0.02	18.72	0.39	7.63	0.27	23.07	0.48
par_block/96	0.39	0.01	7.24	0.08	1.33	0.01	16.63	0.17	5.02	0.45	35.08	0.37

Table 1: 2D Heat Stencil Measurement Results

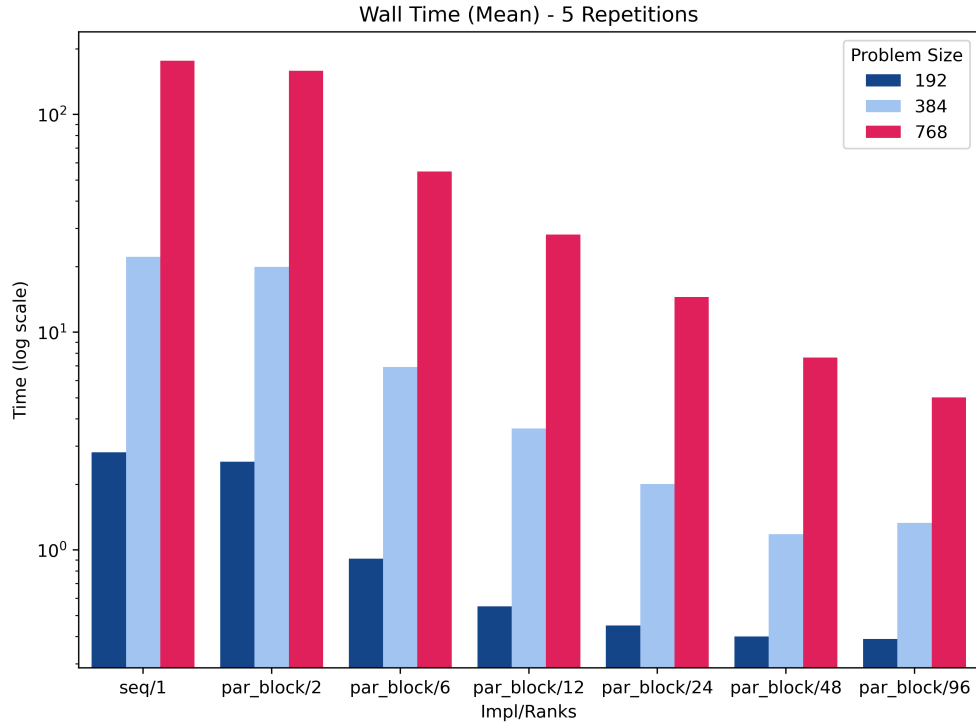


Figure 1: 2D Heat Stencil: Wall Time

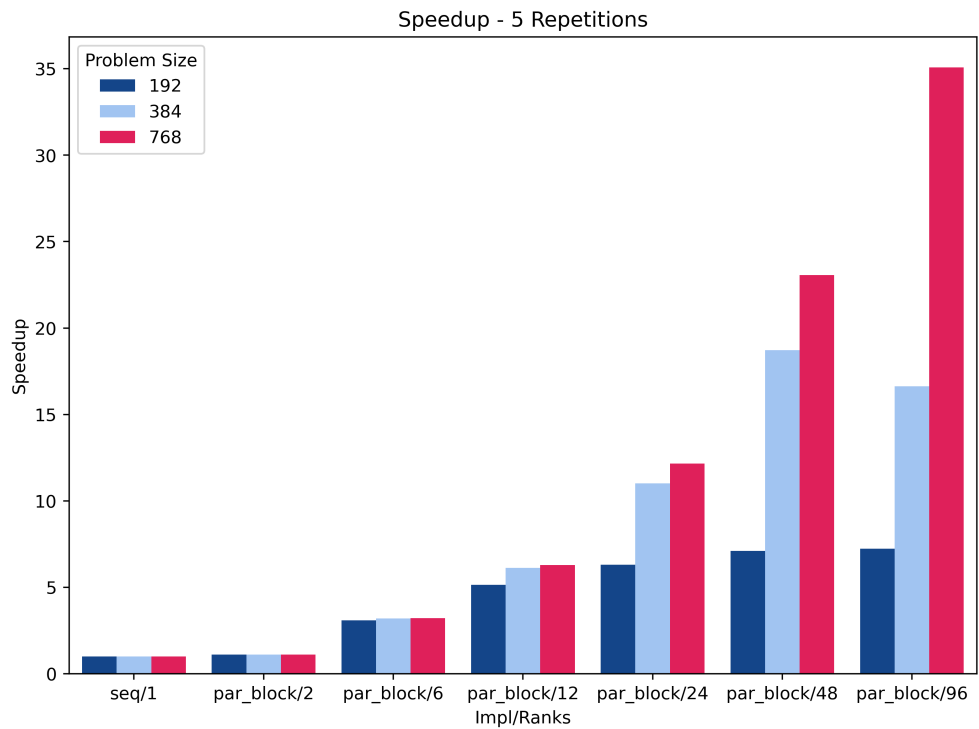


Figure 2: 2D Heat Stencil: Speedup

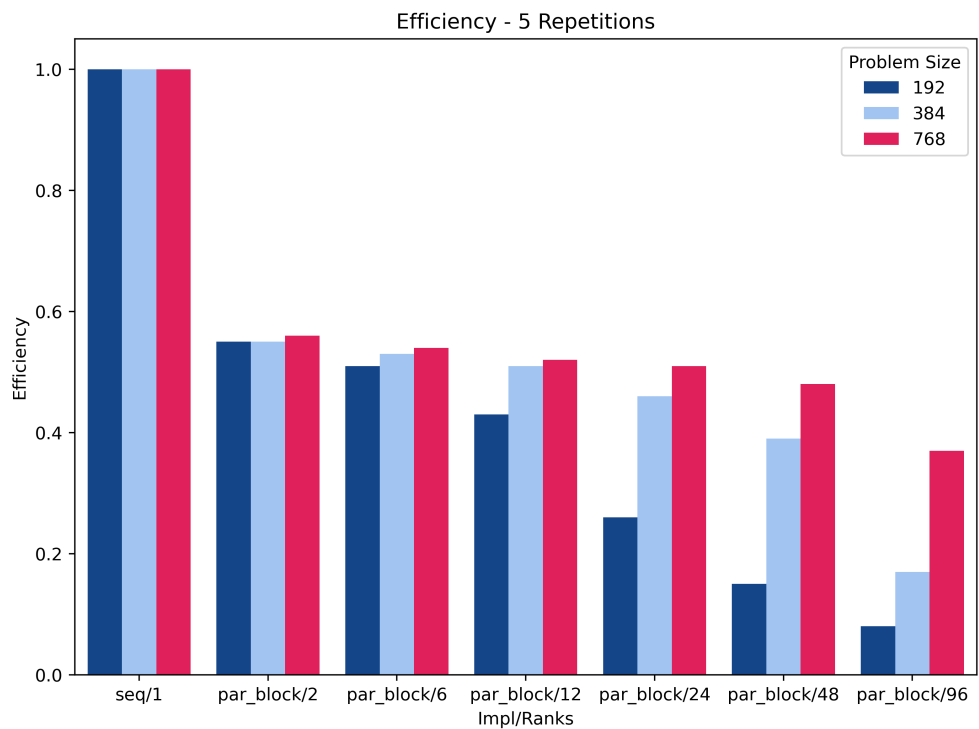


Figure 3: 2D Heat Stencil: Efficiency

- Measure and illustrate one domain-specific and one domain-inspecific performance metric. What can you observe?

- **Heat Propagation Updates per Second**

Heat propagation updates per second serve as the domain-specific performance metric. This metric tracks how many updates to the heat values are performed per second during the simulation, providing a direct measure of the algorithm’s efficiency in propagating heat across the grid. The number of updates is inherently linked to both the grid size and the number of iterations, making it a key indicator of computational performance.

As illustrated in Figure 4, the parallel blocking version consistently outperforms the sequential version across all configurations and problem sizes, as expected.

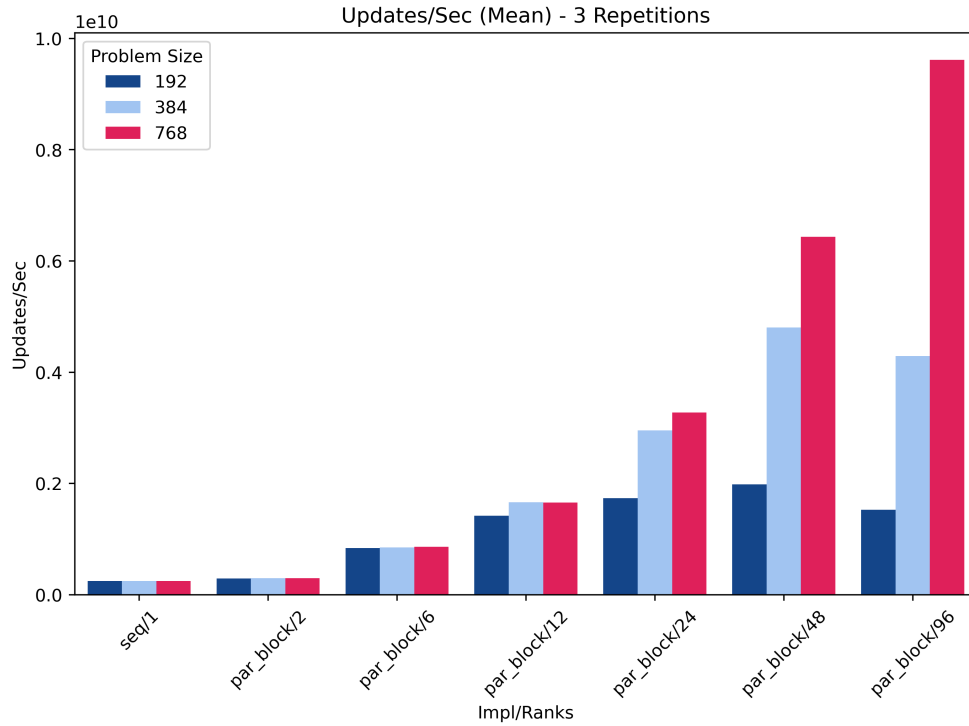


Figure 4: 2D Heat Stencil: Updates/Sec

- **Wall Time**

Wall time is used as the non-domain-specific performance metric. It captures the total computational time required by the algorithm, offering insight into the overall efficiency of the simulation. This metric is particularly valuable for evaluating how long the algorithm takes to complete for different problem sizes or configurations. Refer to the data in Table 1 and the corresponding visualization in Figure 1.

- How can you verify the correctness of your applications?

- **Sequential Version:**

To verify the correctness of the sequential 2D version, we applied the same verification checks used for the 1D version, ensuring that the output remained consistent across multiple executions and different problem sizes. We confirmed that the 2D output was correct based on our understanding of how it should appear in 2D, informed by the results from the 1D output.

– **Parallel Version:**

To verify the correctness of the parallel applications, we compared the output of the parallel implementation with that of the sequential version. We ensured that both produced identical results across multiple executions. By running tests with different problem sizes and configurations, we confirmed that the parallel application consistently matched the sequential output, demonstrating the correctness of the parallelization.

- Insert the wall times for the sequential version and for 96 cores for  $N=768 \times 768$  and  $T=N*100$  into the comparison spreadsheet: [Docs](#)

## Exercise 2

This exercise consists in comparing blocking and non-blocking communication for the heat stencil applications.

- Provide an MPI implementation for the 1D and 2D heat stencil that uses non-blocking communication. If you already implemented a non-blocking version, provide a blocking version, but ensure the non-blocking version works as described below.

```
1 02_heat_stencil_blocking_vs_non_blocking/1D/src/heat_stencil_1D_seq.c
2 02_heat_stencil_blocking_vs_non_blocking/1D/src/heat_stencil_1D_par_blocking.c
3 02_heat_stencil_blocking_vs_non_blocking/1D/src/heat_stencil_1D_par_non_blocking.c
4 02_heat_stencil_blocking_vs_non_blocking/2D/src/heat_stencil_2D_seq.c
5 02_heat_stencil_blocking_vs_non_blocking/2D/src/heat_stencil_2D_par_blocking.c
6 02_heat_stencil_blocking_vs_non_blocking/2D/src/heat_stencil_2D_par_non_blocking.c
```

- Structure your program such that you 1) start a non-blocking ghost cell exchange, 2) compute the inner cells which do not require the result of the ghost cell exchange, 3) block until the ghost cell exchange has finished, and 4) compute the remaining cells.
- Run your programs with multiple problem and machine sizes and compare both versions.

Results of 1D Heat Stencil Execution																
Impl/Ranks	Problem Size															
	768				1536				3072				6144			
	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)
seq/1	0.56	0.00	1.00	1.00	2.37	0.01	1.00	1.00	9.66	0.03	1.00	1.00	38.49	0.09	1.00	1.00
par_b/2	0.79	0.00	0.71	0.35	2.80	0.01	0.85	0.42	10.72	0.05	0.90	0.45	41.80	0.07	0.92	0.46
par_nb/2	0.66	0.00	0.84	0.42	2.36	0.01	1.00	0.50	9.01	0.04	1.07	0.54	35.40	0.18	1.09	0.54
par_b/6	0.67	0.00	0.84	0.14	1.73	0.00	1.37	0.23	5.17	0.02	1.87	0.31	17.28	0.02	2.23	0.37
par_nb/6	0.46	0.00	1.21	0.20	1.27	0.01	1.87	0.31	3.96	0.01	2.44	0.41	13.80	0.03	2.79	0.46
par_b/12	0.57	0.01	0.98	0.08	1.32	0.01	1.79	0.15	3.57	0.17	2.71	0.23	10.51	0.22	3.66	0.31
par_nb/12	0.39	0.01	1.44	0.12	0.91	0.01	2.61	0.22	2.62	0.17	3.69	0.31	8.03	0.11	4.80	0.40
par_b/24	1.48	0.00	0.38	0.02	2.92	0.01	0.81	0.03	6.03	0.01	1.60	0.07	12.86	0.07	2.99	0.12
par_nb/24	0.80	0.00	0.70	0.03	1.50	0.01	1.58	0.07	2.93	0.01	3.29	0.14	5.76	0.07	6.68	0.28
par_b/48	1.50	0.01	0.37	0.01	2.94	0.03	0.81	0.02	5.85	0.03	1.65	0.03	12.17	0.06	3.16	0.07
par_nb/48	0.84	0.01	0.67	0.01	1.56	0.01	1.51	0.03	3.02	0.03	3.20	0.07	5.94	0.07	6.49	0.14
par_b/96	1.54	0.02	0.36	0.00	2.99	0.02	0.79	0.01	5.87	0.02	1.65	0.02	11.94	0.04	3.22	0.03
par_nb/96	0.83	0.01	0.68	0.01	1.54	0.01	1.54	0.02	3.02	0.03	3.19	0.03	5.95	0.06	6.47	0.07

Table 2: 1D Heat Stencil Measurement Results

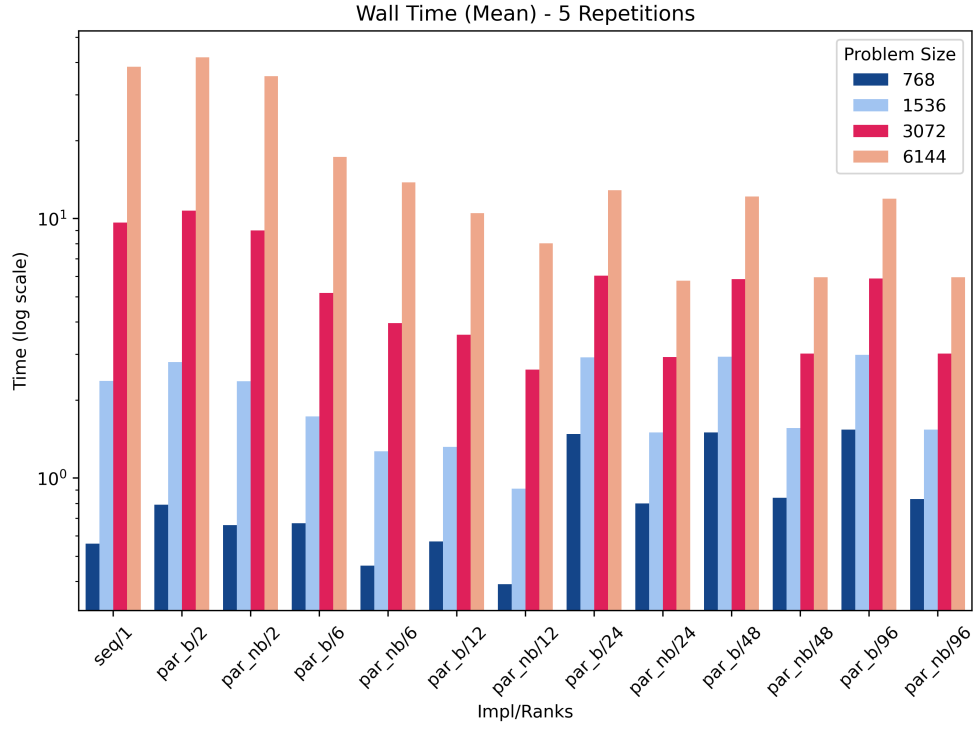


Figure 5: 1D Heat Stencil: Wall Time

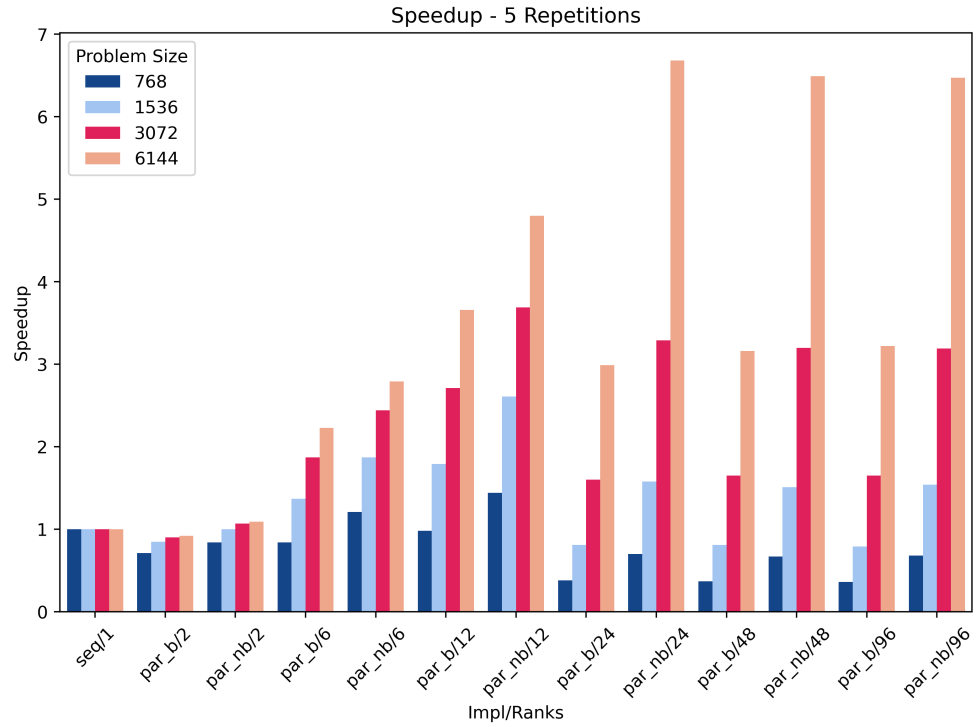


Figure 6: 1D Heat Stencil: Speedup

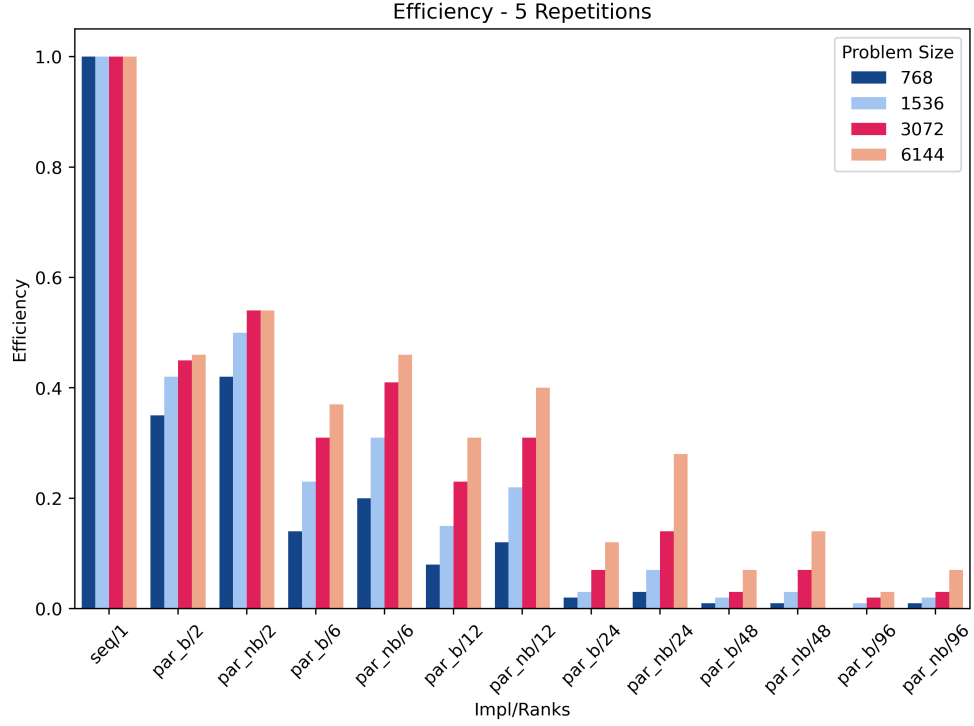


Figure 7: 1D Heat Stencil: Efficiency

Results of 2D Heat Stencil Execution												
Impl/Ranks	Problem Size											
	192				384				768			
	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)	$\mu$ [s]	$\sigma$ [s]	S(p)	E(p)
seq/1	2.77	0.04	1.00	1.00	22.06	0.14	1.00	1.00	175.29	0.56	1.00	1.00
par_b/2	2.50	0.01	1.11	0.55	19.78	0.18	1.12	0.56	158.23	0.49	1.11	0.55
par_nb/2	2.24	0.01	1.24	0.62	17.69	0.07	1.25	0.62	140.20	0.61	1.25	0.63
par_b/6	0.91	0.01	3.06	0.51	6.95	0.10	3.17	0.53	54.16	0.05	3.24	0.54
par_nb/6	0.87	0.02	3.19	0.53	6.21	0.10	3.55	0.59	48.19	0.25	3.64	0.61
par_b/12	0.58	0.03	4.82	0.40	3.63	0.16	6.07	0.51	27.95	0.33	6.27	0.52
par_nb/12	0.51	0.01	5.45	0.45	3.33	0.06	6.62	0.55	24.74	0.21	7.09	0.59
par_b/24	0.45	0.01	6.17	0.26	2.03	0.03	10.88	0.45	14.65	0.41	11.97	0.50
par_nb/24	0.38	0.01	7.34	0.31	1.81	0.03	12.16	0.51	13.15	0.82	13.33	0.56
par_b/48	0.39	0.01	7.04	0.15	1.39	0.18	15.85	0.33	7.67	0.52	22.84	0.48
par_nb/48	0.29	0.01	9.43	0.20	1.10	0.05	20.08	0.42	7.43	0.48	23.59	0.49
par_b/96	0.48	0.20	5.77	0.06	1.38	0.04	16.04	0.17	5.59	0.74	31.34	0.33
par_nb/96	0.63	0.01	4.43	0.05	1.14	0.18	19.36	0.20	4.40	0.25	39.86	0.42

Table 3: 2D Heat Stencil Measurement Results



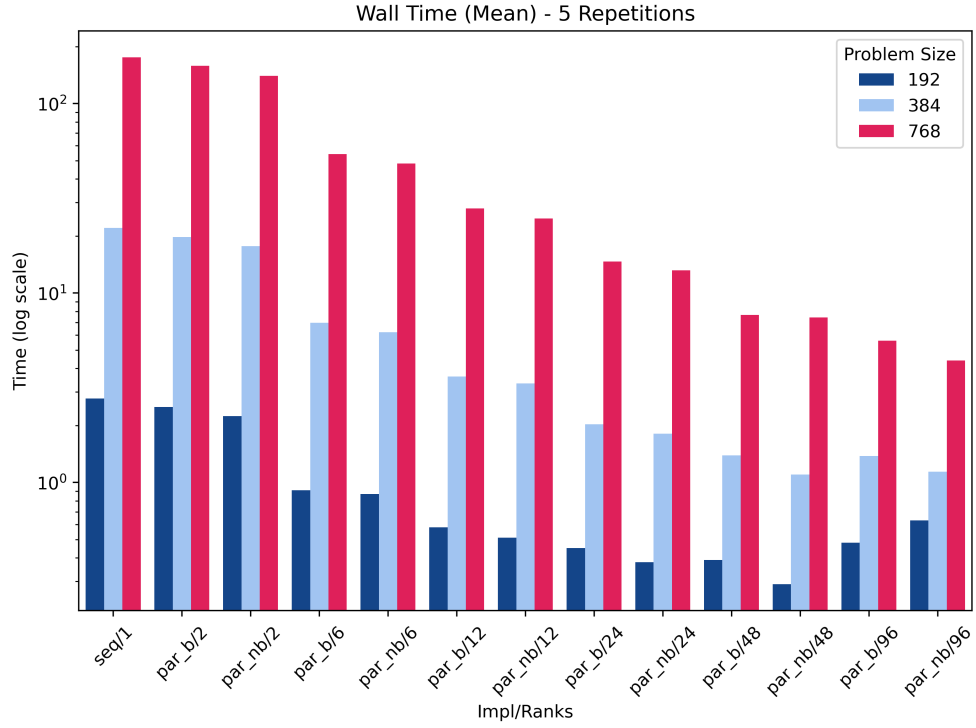


Figure 8: 2D Heat Stencil: Wall Time

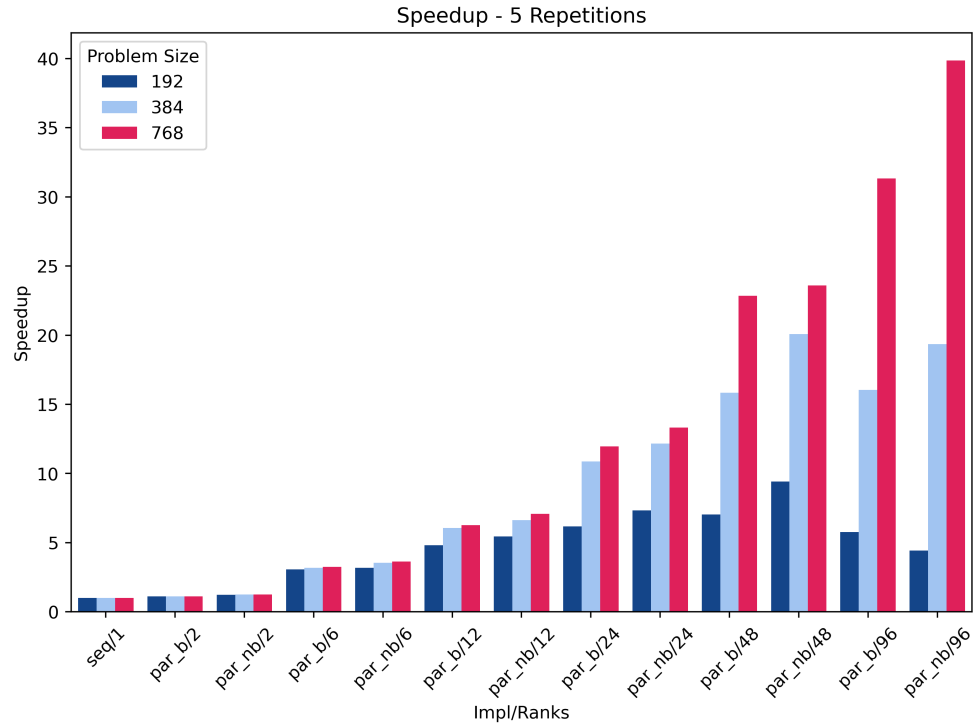


Figure 9: 2D Heat Stencil: Speedup

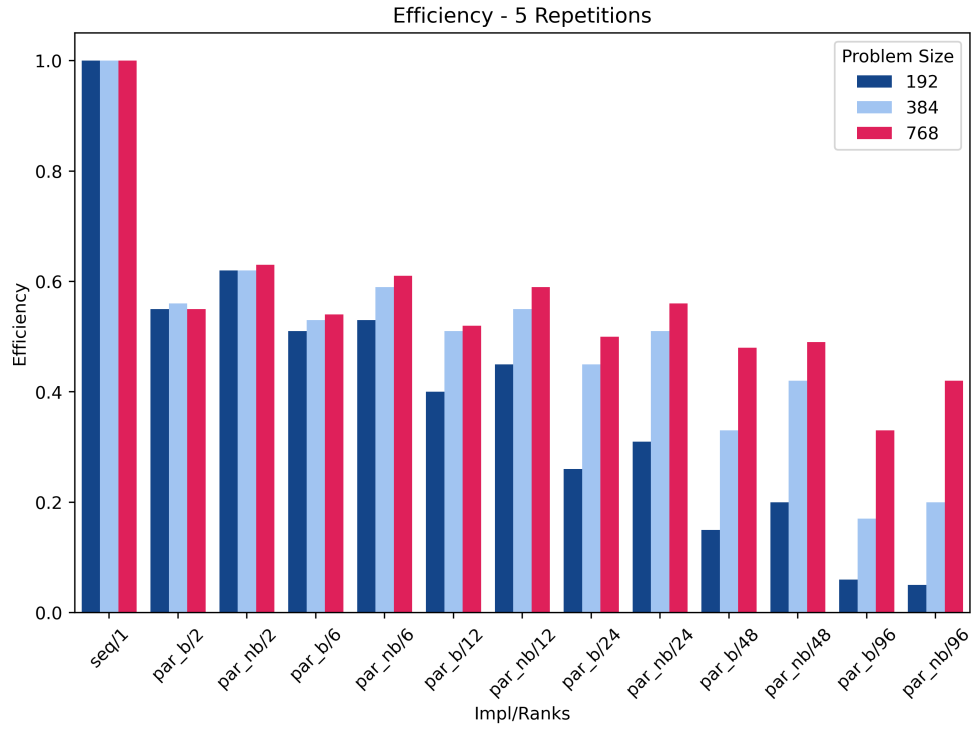


Figure 10: 2D Heat Stencil: Efficiency

- Insert wall time for 96 cores for  $N=768 \times 768$  and  $T=N \times 100$  into the comparison spreadsheet: [Docs](#)