Parallel Performance Measurements of Jacobi Solver (POSIX Threads)

Measurement 1: Parallel Performance Results (Jacobi Solver, POSIX Threads)

System Setup: *Node: ant13, AMD EPYC 7443, 48 cores, 125 GiB RAM, 4096 interlines, 75 iterations, SLURM, Hyperfine timing tool.*

1. Raw Benchmark Data

Table 1: Measured Runtimes, Speedup, and Efficiency for Jacobi Solver (4096 interlines, 75 iterations, Node: ant13)

Threads	Mean Time (s)	Speedup	Efficiency (%)
1	694.13	1.00	100
2	350.34	1.98	99
3	234.18	2.97	99
6	118.90	5.84	97
12	63.09	11.00	92
18	44.35	15.66	87
24	37.01	18.76	78

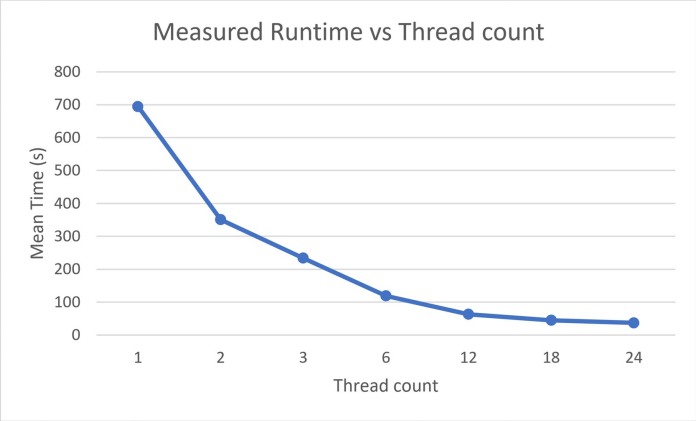

Figure 1: Mean runtime of Jacobi Solver as a function of thread count (1 thread represents the serial baseline).

2. Interpretation of Parallel Performance Results

From Figure 1, we observe that runtime decreases sharply with increasing thread count, especially between 1 and 6 threads. The first data point (1 thread) represents the original serial program, serving as the baseline for speedup calculations in Table 1.

Experimental Setup:

- **Compute Node:** ant13
- **Processor:** AMD EPYC 7443, 48 cores
- **RAM:** 125 GiB
- **Problem Size:** 4096 interlines, 75 iterations
- **Measurement Tool:** `hyperfine` (3 runs per configuration)

Key Observations:

- **Strong Scaling:** As shown in Figure 1, runtime drops significantly as the number of threads increases, confirming the effectiveness of parallelization for this workload.
- **Near-Linear Speedup at Low Thread Counts:** Doubling threads from 1 to 2 nearly halves runtime, and near-ideal speedup is observed up to 6 threads.
- **Diminishing Returns at Higher Thread Counts:** Beyond 12 threads, speedup becomes sublinear, reflecting increased parallel overhead and memory contention.
- **Efficiency Trend:** As shown in the table, parallel efficiency remains above 90% up to 12 threads, but decreases to 78% at 24 threads due to synchronization and memory bandwidth limitations.
- **Consistency:** All measurements were performed on the same node to ensure fair comparison.

Conclusion:

- Parallelization using POSIX threads yields substantial reductions in runtime for the Jacobi solver.
- The observed efficiency drop at higher thread counts is typical of shared-memory parallel computing, highlighting the impact of hardware resource limits and parallel overheads.