Pthread and OpenMP Implementation of a 9-point Stencil Operation

By: Michael Dandrea

Project Background

- Implement Pthread and OpenMP Version of a 9-point stencil operation
- Gather and plot information about speedup, timing, efficiency and serial fraction.

Pthread Implementation

```c
void *stencil_thread(void *arg)
{
    int thread_id = *((int *)arg);

    // Calculate start and end rows for this thread
    int start_row = BLOCK_LOW(thread_id, global_num_threads, global_rows);
    int end_row = BLOCK_HIGH(thread_id, global_num_threads, global_rows);

    // Only process interior points
    if (start_row == 0)
        start_row = 1;
    if (end_row == global_rows - 1)
        end_row = global_rows - 2;

    // Apply stencil to assigned rows
    for (int i = start_row; i <= end_row; i++)
    {
        for (int j = 1; j < global_cols - 1; j++)
        {
            global_temp[i][j] = (global_data[i - 1][j - 1] + global_data[i - 1][j] + global_data[i - 1][j + 1] +
                                 global_data[i][j + 1] + global_data[i + 1][j + 1] + global_data[i + 1][j] +
                                 global_data[i + 1][j - 1] + global_data[i][j - 1] + global_data[i][j]) /
                                9.0;
        }
    }

    // Wait for all threads to complete their computation
    my_pthread_barrier_wait(&global_barrier);

    return NULL;
```

OpenMP Implementation

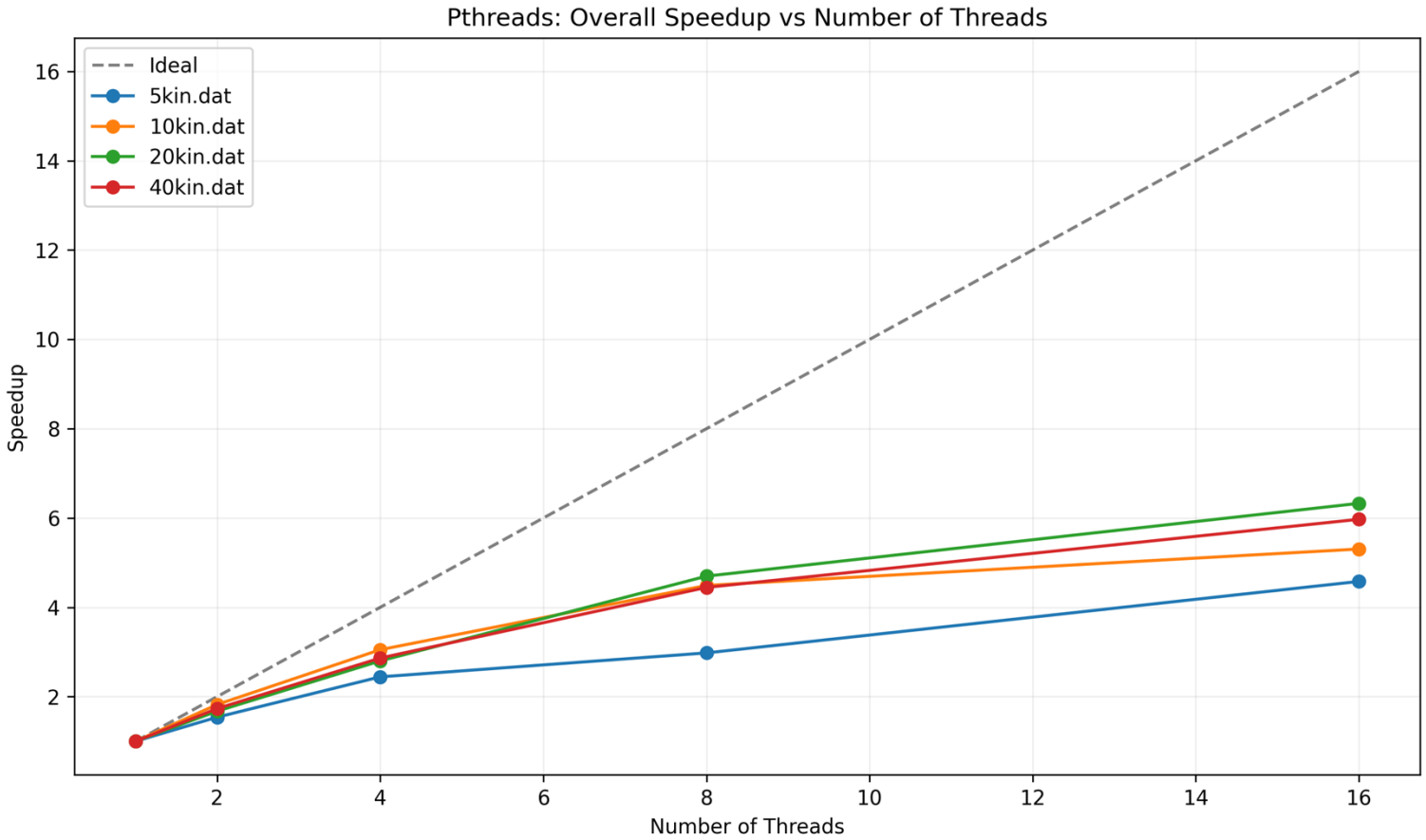
```
// Parallel stencil computation
#pragma omp parallel for
for (int i = 1; i < rows - 1; i++) {
    for (int j = 1; j < cols - 1; j++) {
        temp[i][j] = (data[i-1][j-1] + data[i-1][j] + data[i-1][j+1] +
                      data[i][j+1]   + data[i+1][j+1]   + data[i+1][j] +
                      data[i+1][j-1] + data[i][j-1] + data[i][j]) / 9.0;
    }
}
```

Result Gathering and Plotting

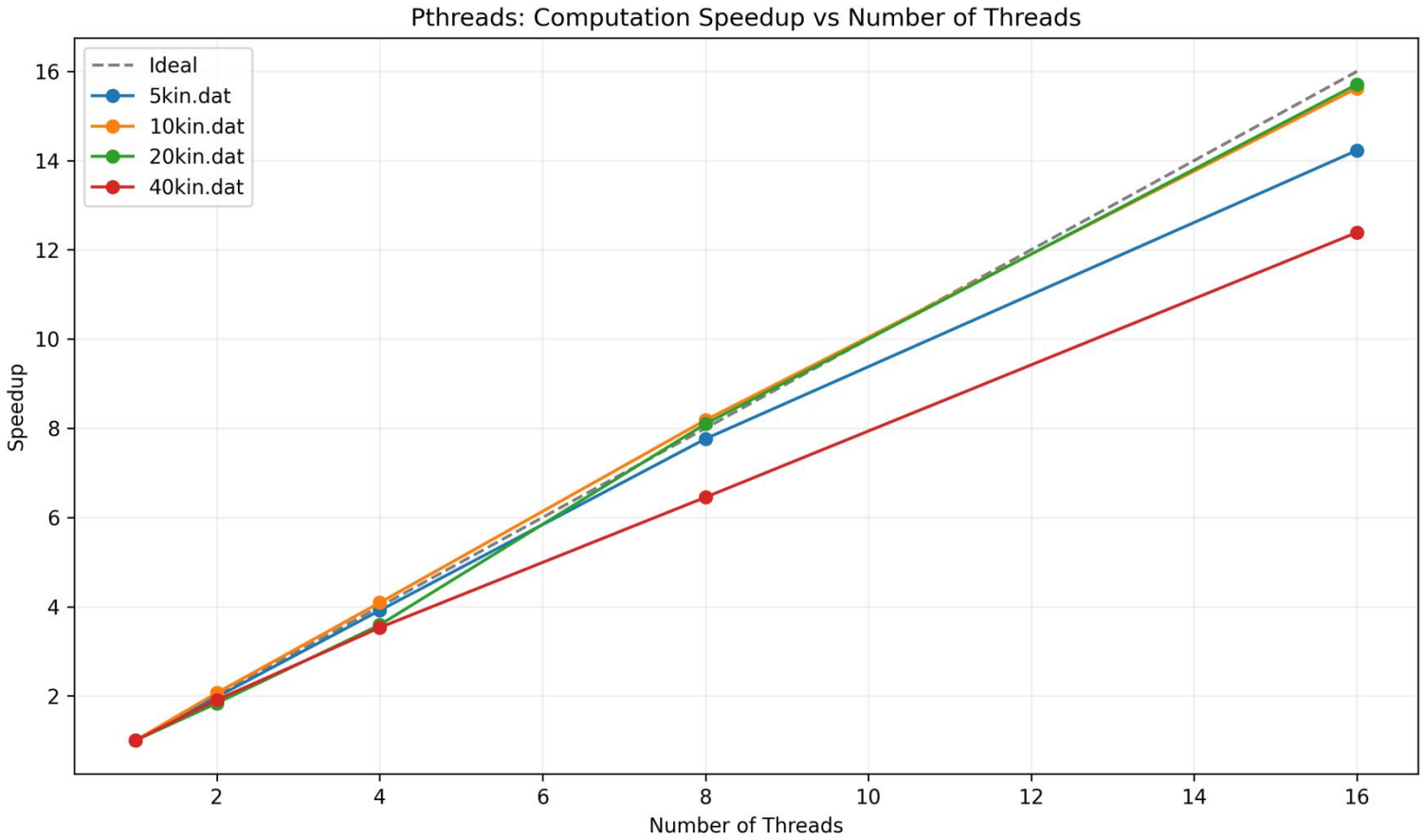
- Ran on Expanse
- SLURM script
- Four total python scripts
- run_all.sh

Pthread Results

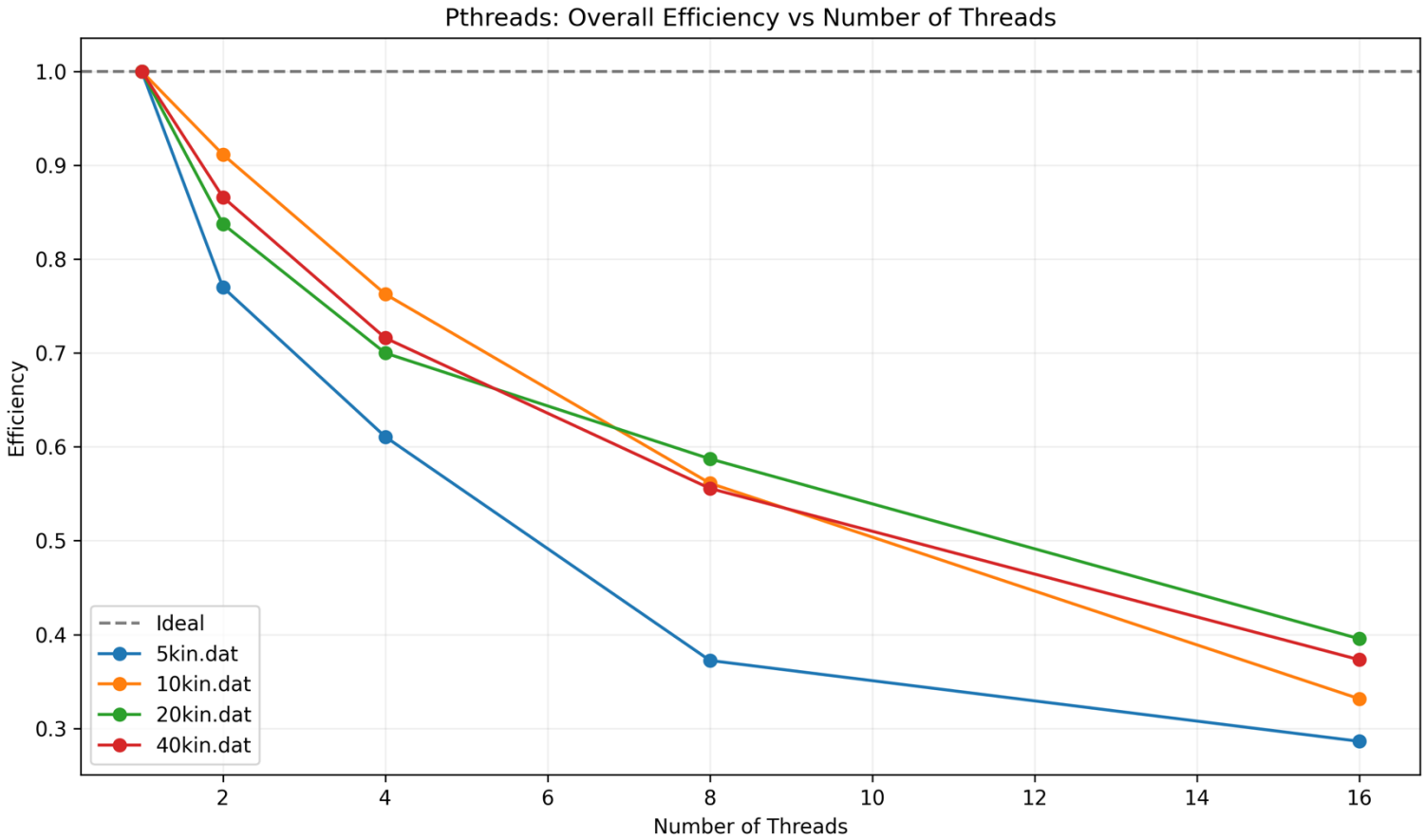
Overall Speedup



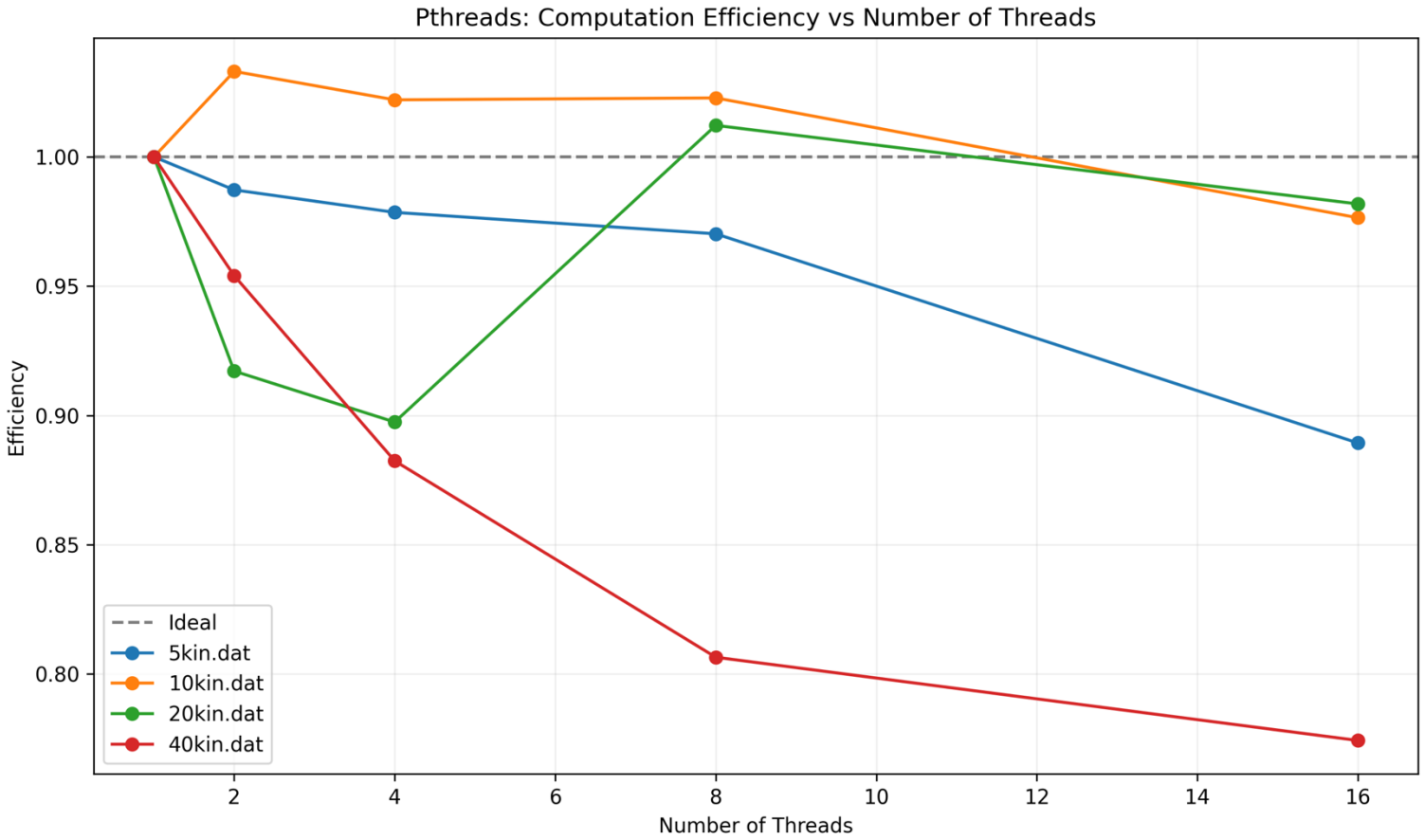
Computation Speedup



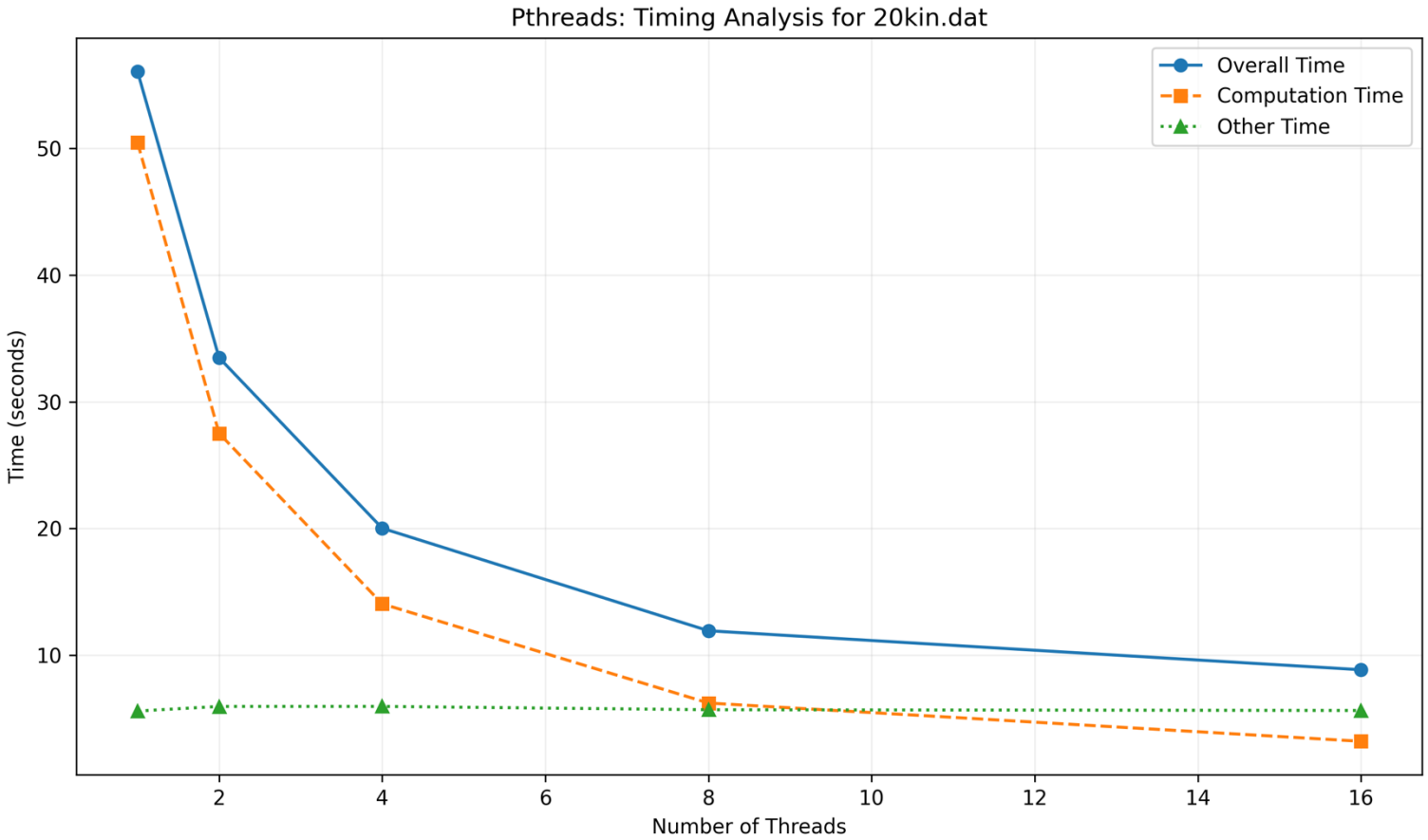
Overall Efficiency



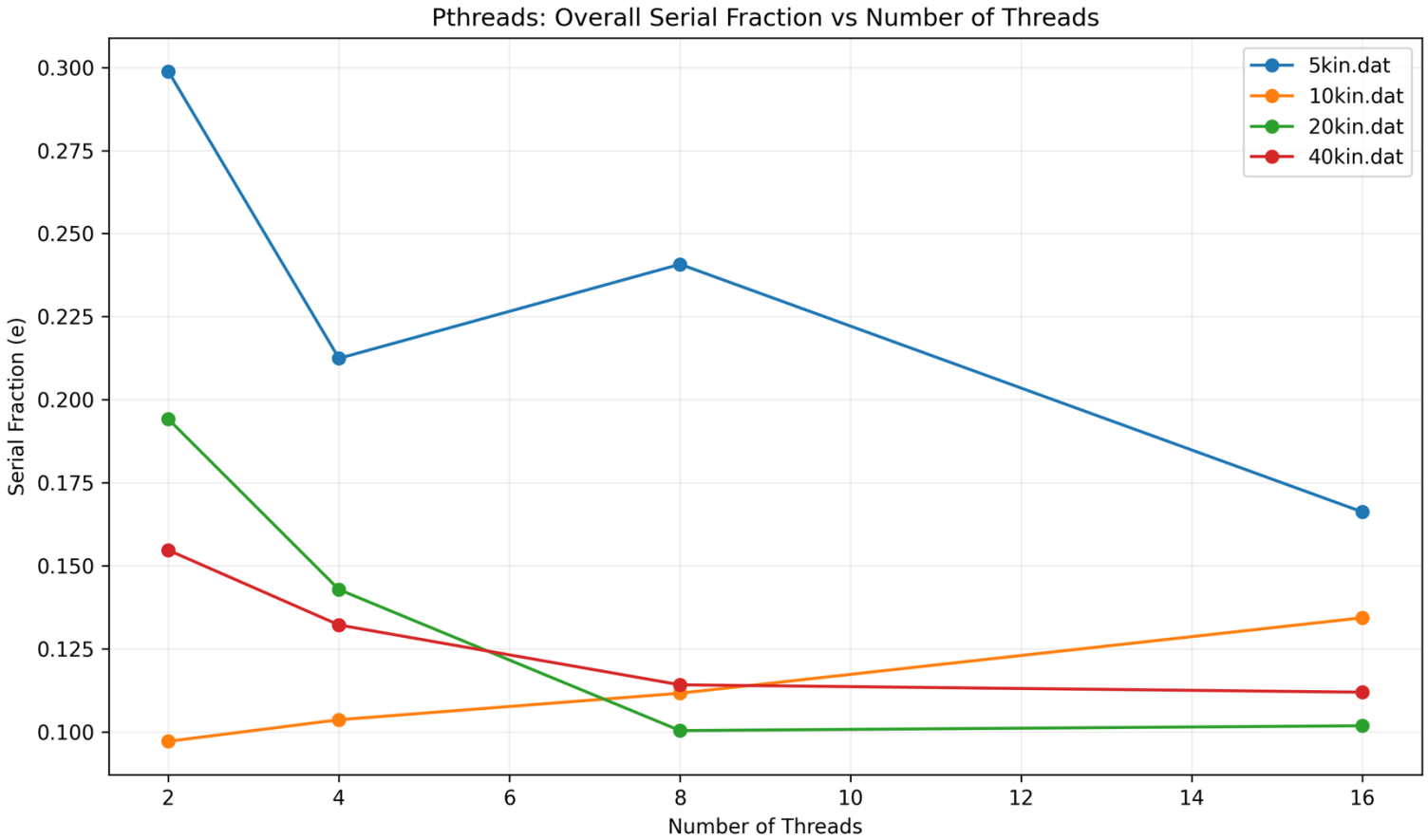
Computation Efficiency



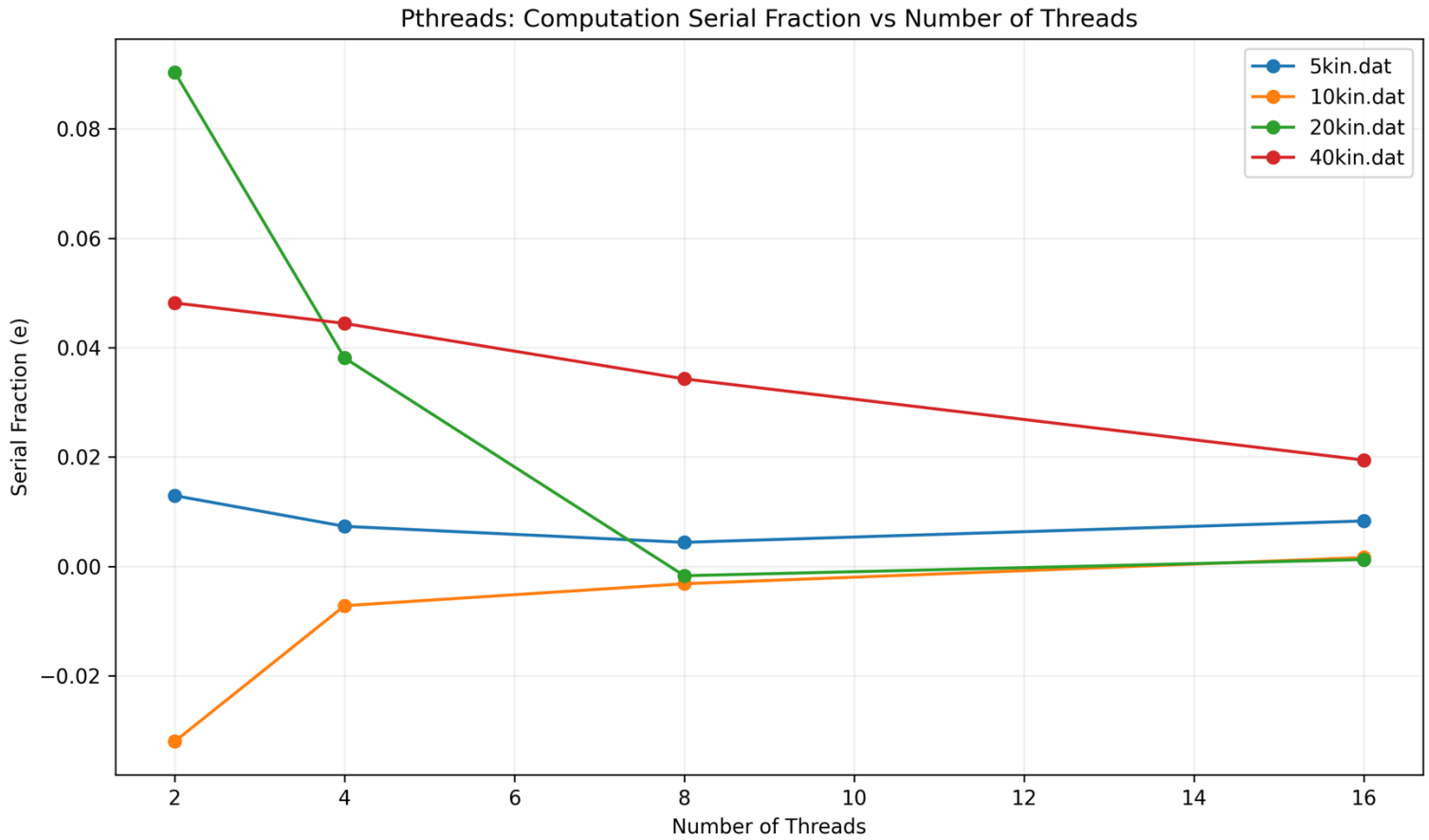
Timing Example



Overall Serial Fraction

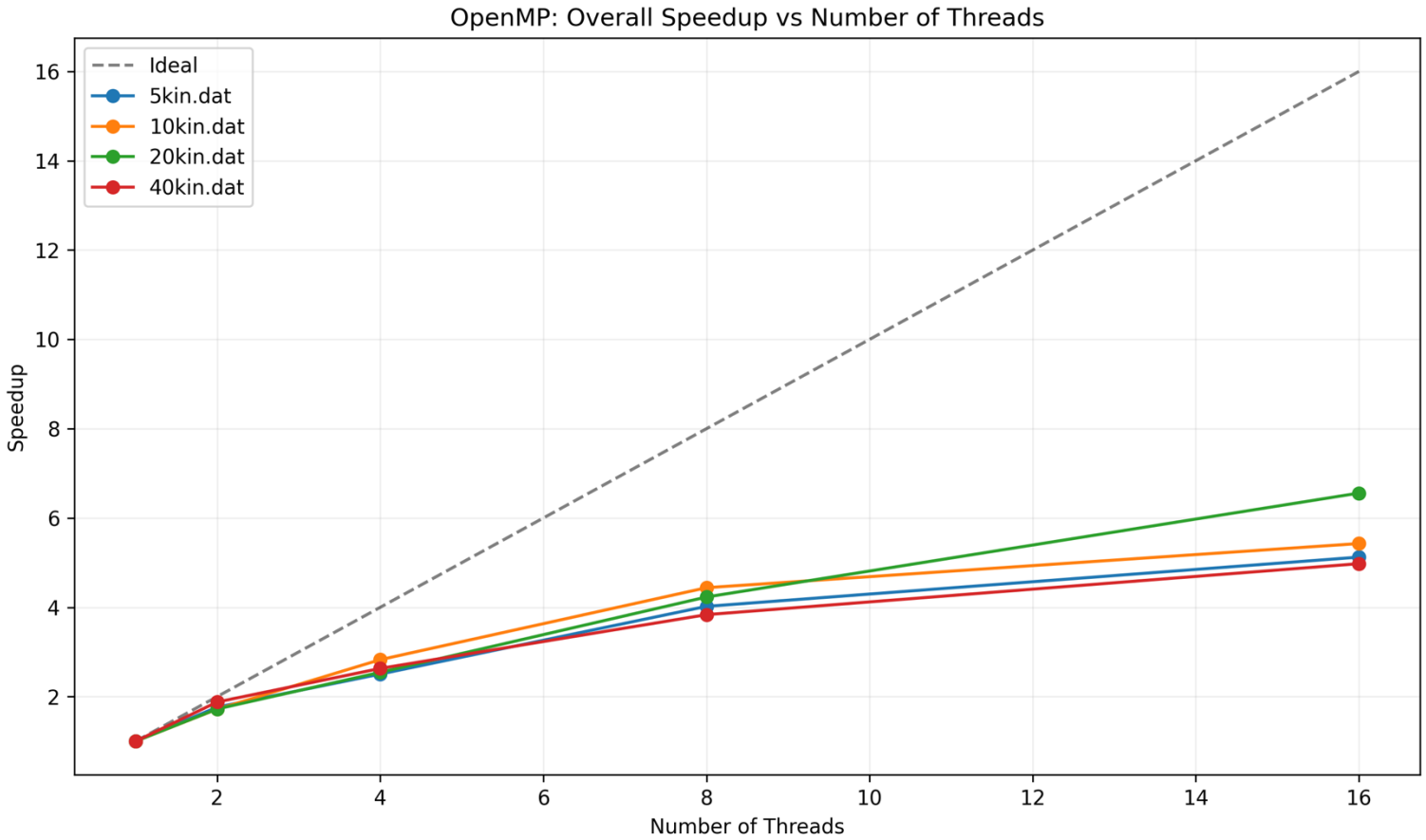


Computation Serial Fraction

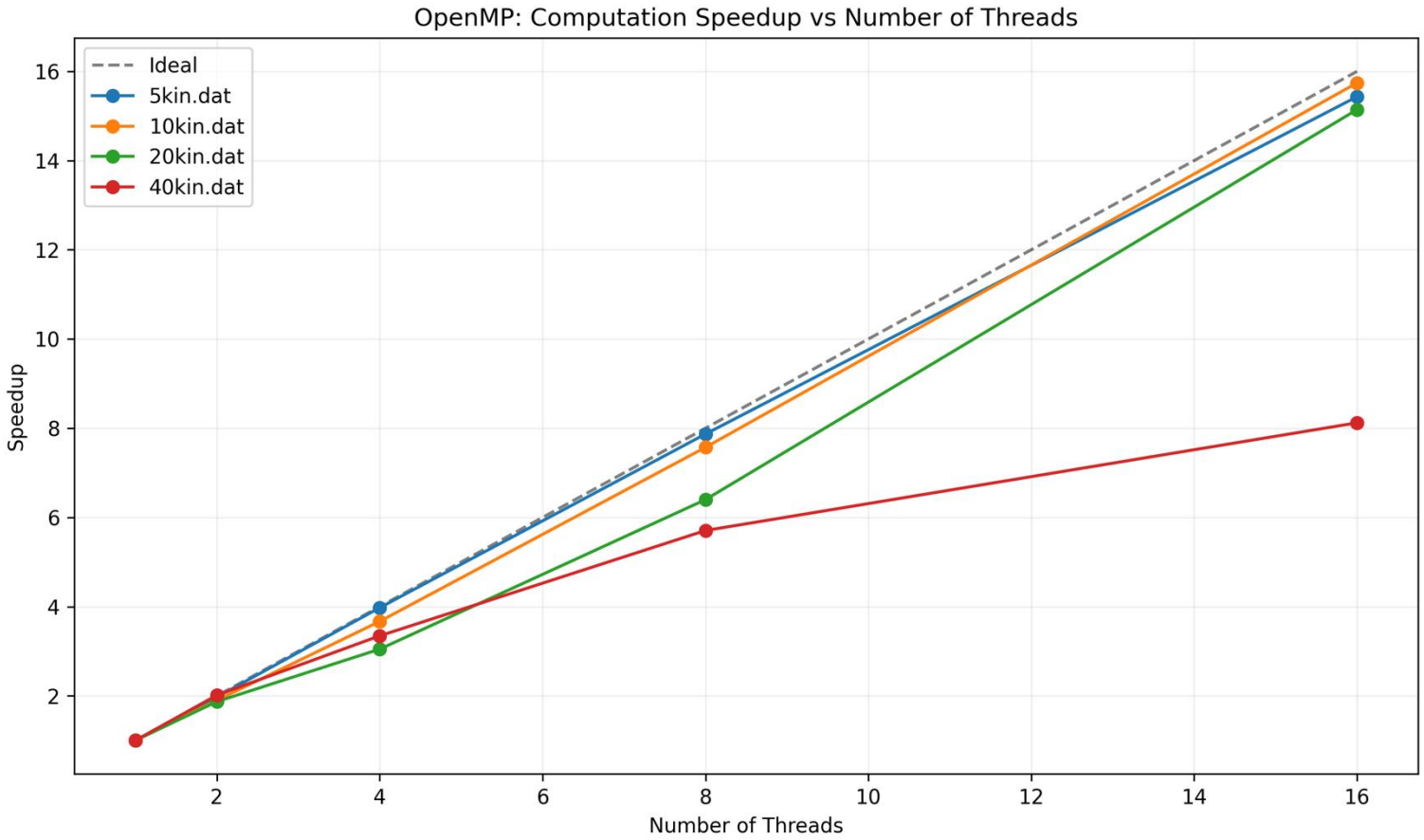


OpenMP Results

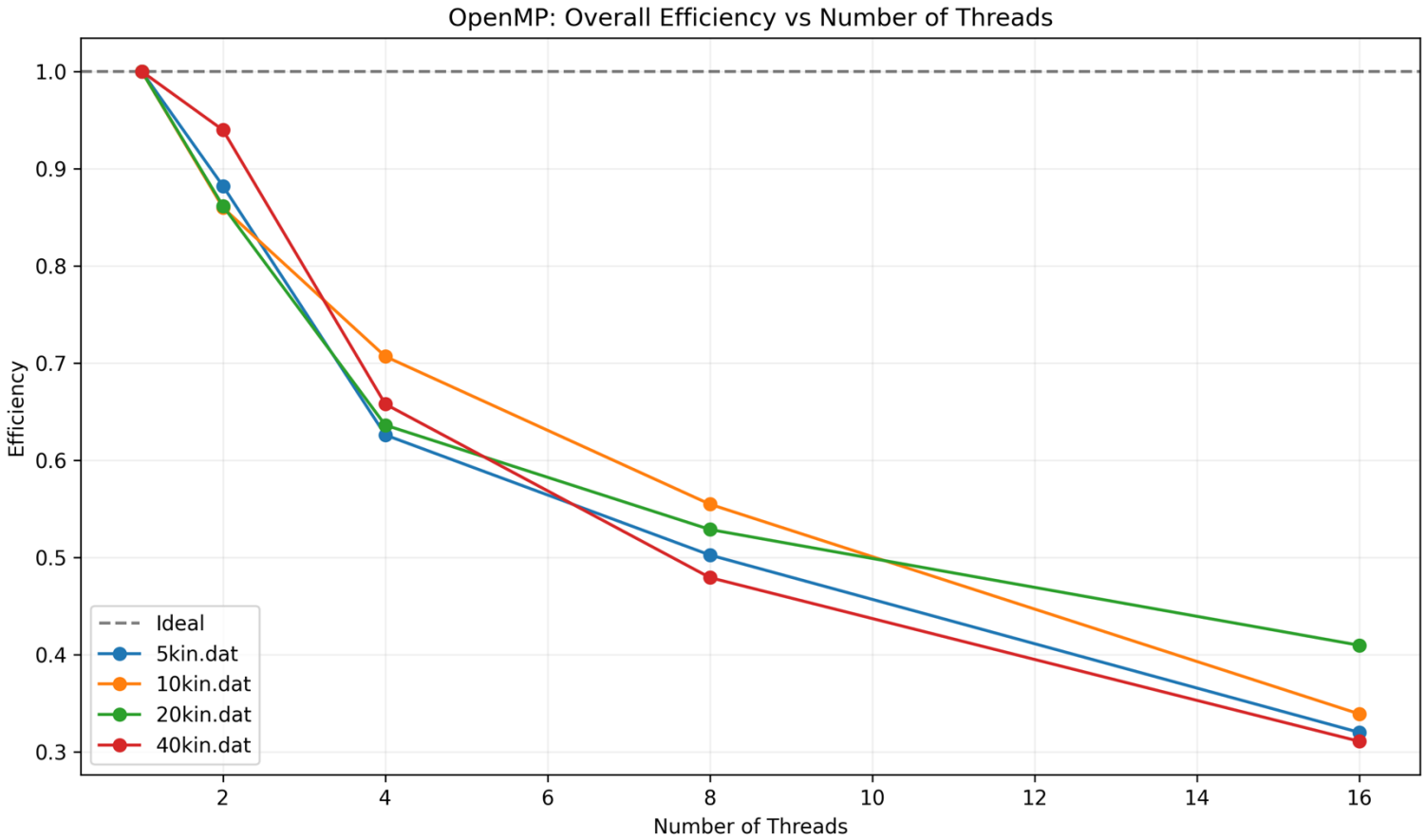
Overall Speedup



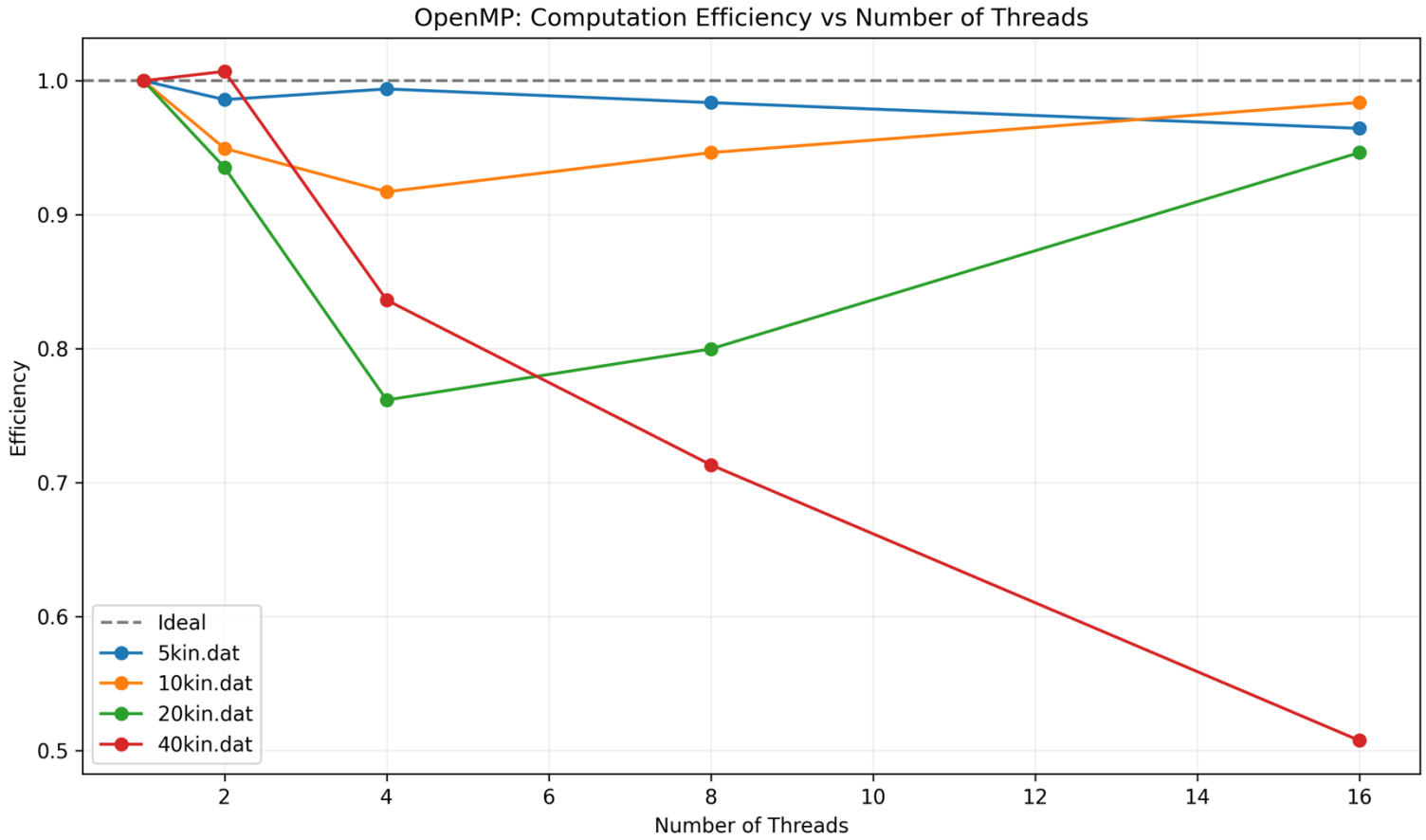
Computation Speedup



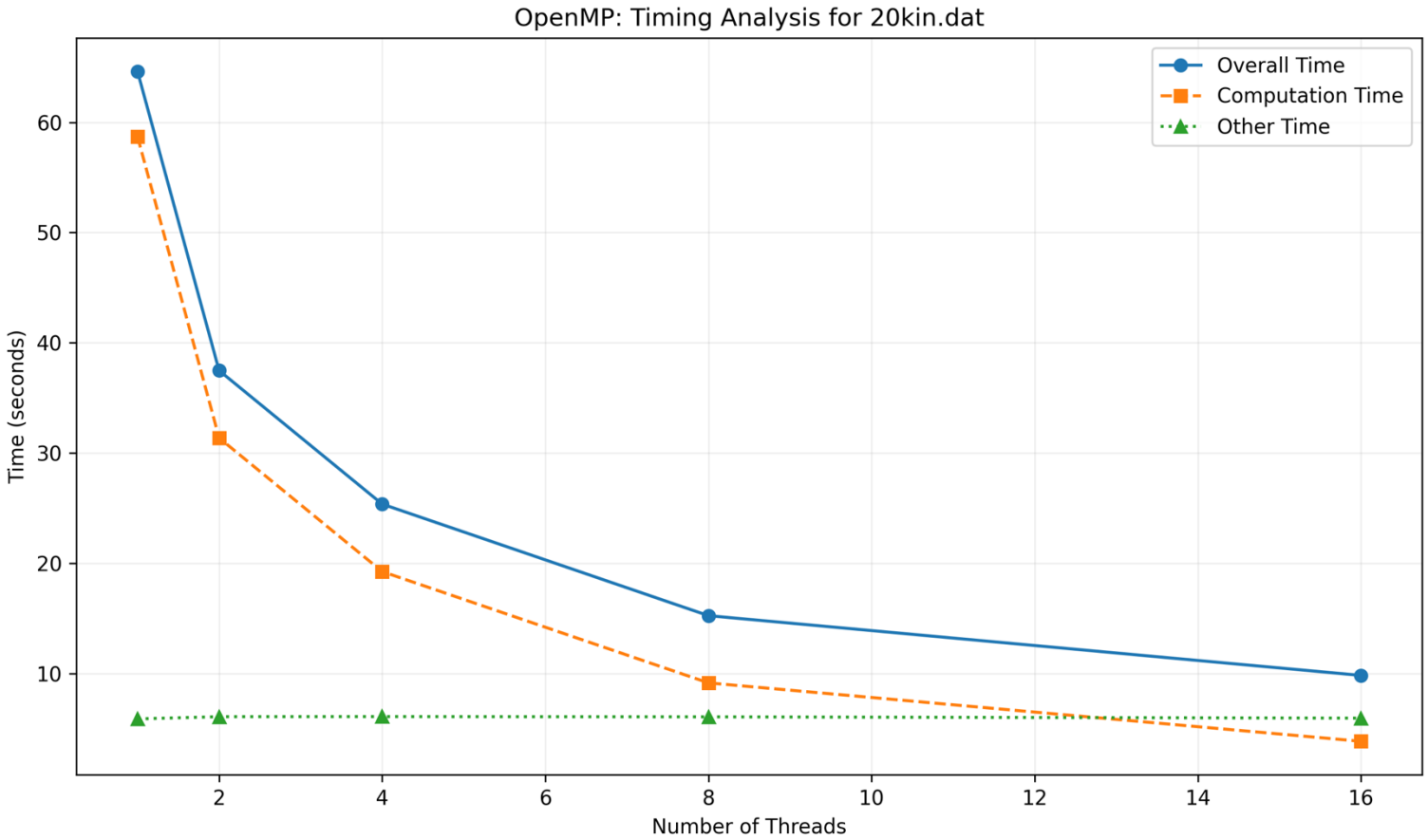
Overall Efficiency



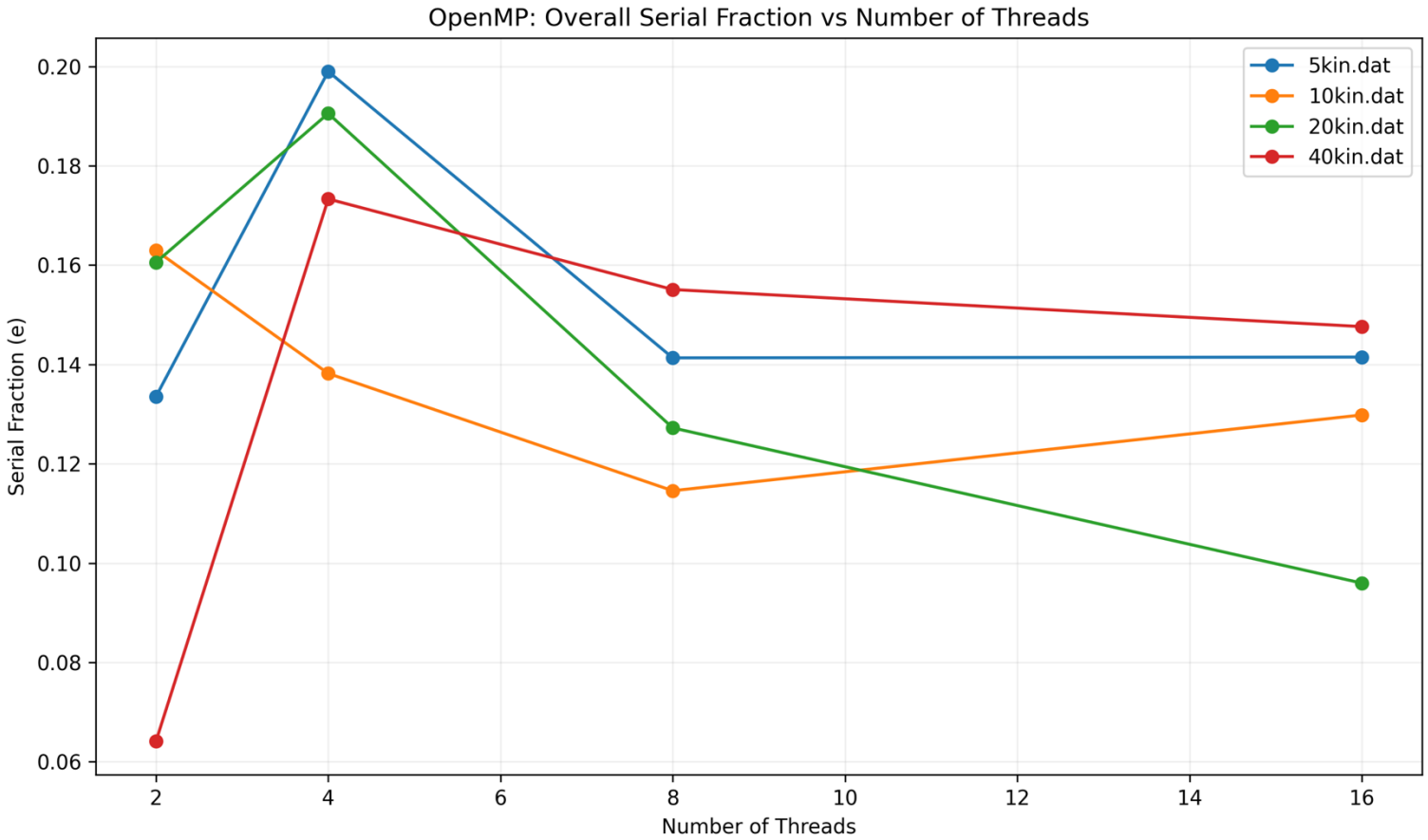
Computation Efficiency



Timing Example



Overall Serial Fraction



Computation Serial Fraction