Parallel Computing Homework 1

Tyler Scotti

January 23rd, 2026

Question 1: Parallely Calculating PI with Shared-Memory via Threads

Number of Trials: 10^8

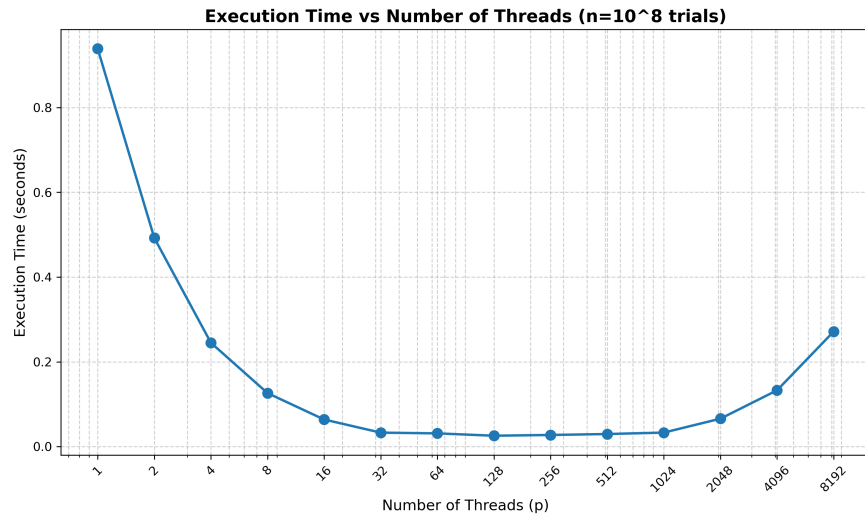### Question 1.1: Execution Time vs Number of Threads

Figure 1: Execution Time vs Threads

The graph demonstrates how execution time varies with the number of threads for 10^8 Monte Carlo trials. Execution time decreases from 0.94 seconds with a single thread to a minimum of 0.026 seconds at 128 threads, achieving substantial performance gains. Beyond 128 threads, execution time increases due to thread management overhead and synchronization costs exceeding the benefits of parallelization.

Question 1.2: Speedup vs Number of Threads

The speedup curve shows near-linear scaling from 1 to 32 threads, reaching a maximum speedup of 36.52x at 128 threads. After this peak, speedup decreases as the overhead from thread creation, scheduling, and
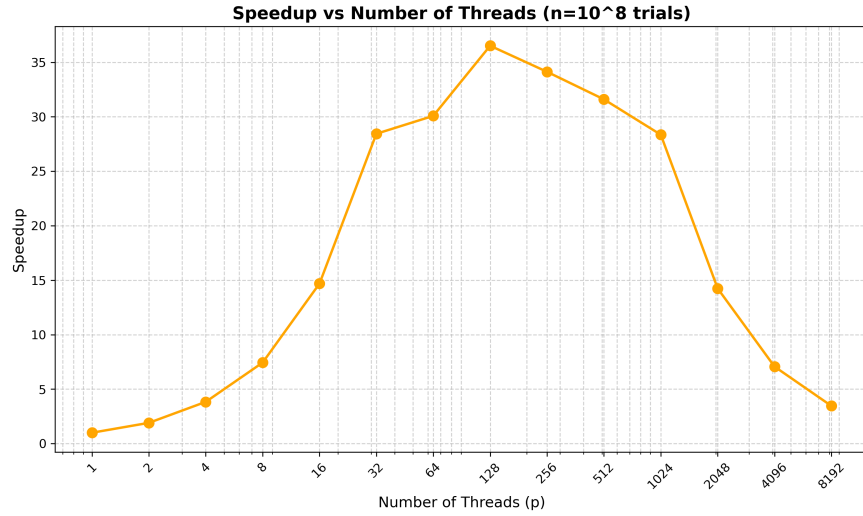
Figure 2: Speedup vs Threads

synchronization dominates. This behavior is consistent with Amdahl's Law, where the sequential portion and parallel overhead limit achievable speedup.

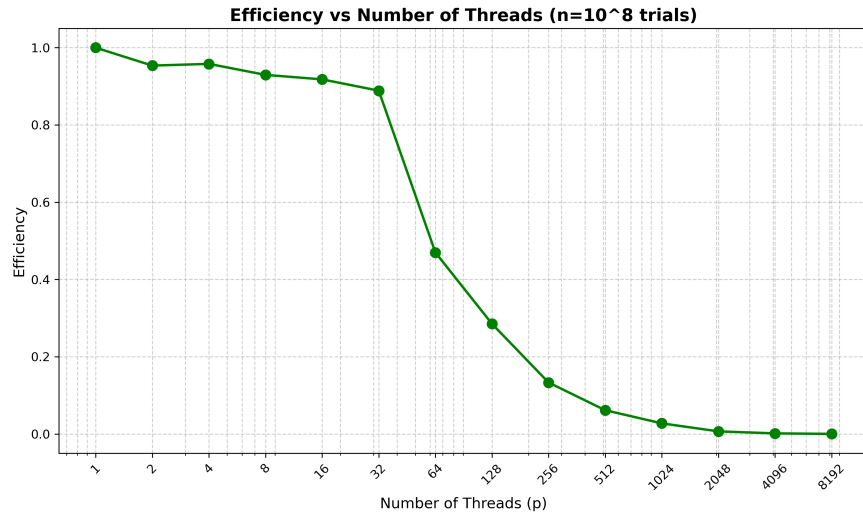## Question 1.3: Efficiency vs Number of Threads

Figure 3: Efficiency vs Threads

Parallel efficiency (defined as $\text{Efficiency} = \frac{\text{Speedup}}{p}$) remains high (>90%) for up to 32 threads, indicating effective utilization of parallel resources. Beyond this point, efficiency drops sharply, reaching approximately 28.5% at 128 threads and continuing to decline. This trend reflects diminishing returns as thread overhead becomes more significant relative to computational work per thread.

Question 1.4

Q: In your experiments, what value of p minimizes the parallel runtime?

A: 128 threads minimizes the parallel runtime at 0.0257 seconds, representing a 36.52x speedup over sequential execution. While higher thread counts are available, they introduce more overhead than performance benefit.

Question 2

Question 2.1: