Parallel Solution to the N-Queens Problem (HW1)

Marios Stavrou, Panayiotis Charalambous

January 2025

Problem Description

The N-Queens problem consists of placing N queens on an $N \times N$ chessboard such that no two queens attack each other. This includes conflicts along rows, columns, and diagonals. For $N = 8$, the problem has exactly 92 valid solutions.

Proposed Solution

The provided solution implements a parallel backtracking algorithm using POSIX threads (`pthreads`). Each thread is responsible for exploring the search space starting from a unique queen placement in the first column. The remaining columns are solved recursively using backtracking search.

Thread safety during output is ensured using a mutex, and execution time is measured using a monotonic clock. Each thread maintains its own solution count, which is aggregated at the end of execution.

Parallelization Strategy

- One thread is created per initial queen position in column zero.
- Each thread explores its own independent search tree.
- Shared output is protected using a mutex.

Results

The program correctly computes all possible solutions for the 8-Queens problem.

- Total solutions found: 92
- Wall-clock execution time: 9392 microseconds

Advantages

- Efficient use of parallelism by dividing the search space across threads.
- Simple and intuitive backtracking approach.
- Correct synchronization of shared output using mutex locks.

Disadvantages

- Threads are statically assigned, limiting scalability for larger values of N.
- No dynamic load balancing between threads.
- Printing intermediate boards significantly increases synchronization overhead.

Conclusion

This solution demonstrates an effective parallel approach to solving the N-Queens problem using pthreads. While it achieves correct results and good performance for $N = 8$, further optimizations would be required to scale efficiently for larger board sizes.