Exploring Solutions to the N-Queens Problem

●●●

Aaron Paul & Chem Chikweze

Introduction

- The N-Queens problem involves placing N queens on an N×N chessboard so that no two queens threaten each other.

- Objective: Explore various computational methods to solve the problem efficiently.

Computational Approaches Used

- Sequential Approach: Single-threaded traditional method.

- Parallel with pthreads: Parallel computing on a single machine using multiple threads.

- Distributed with MPI: Utilizing multiple computers to distribute the computation.

- Work-Stealing Queue Model: Advanced thread management for optimized computation.

Overview of Solutions

- Key components include safe placement checks and recursive solution finding using C++, Bash, pthreads, and MPI. Challenges addressed include synchronization and data distribution.
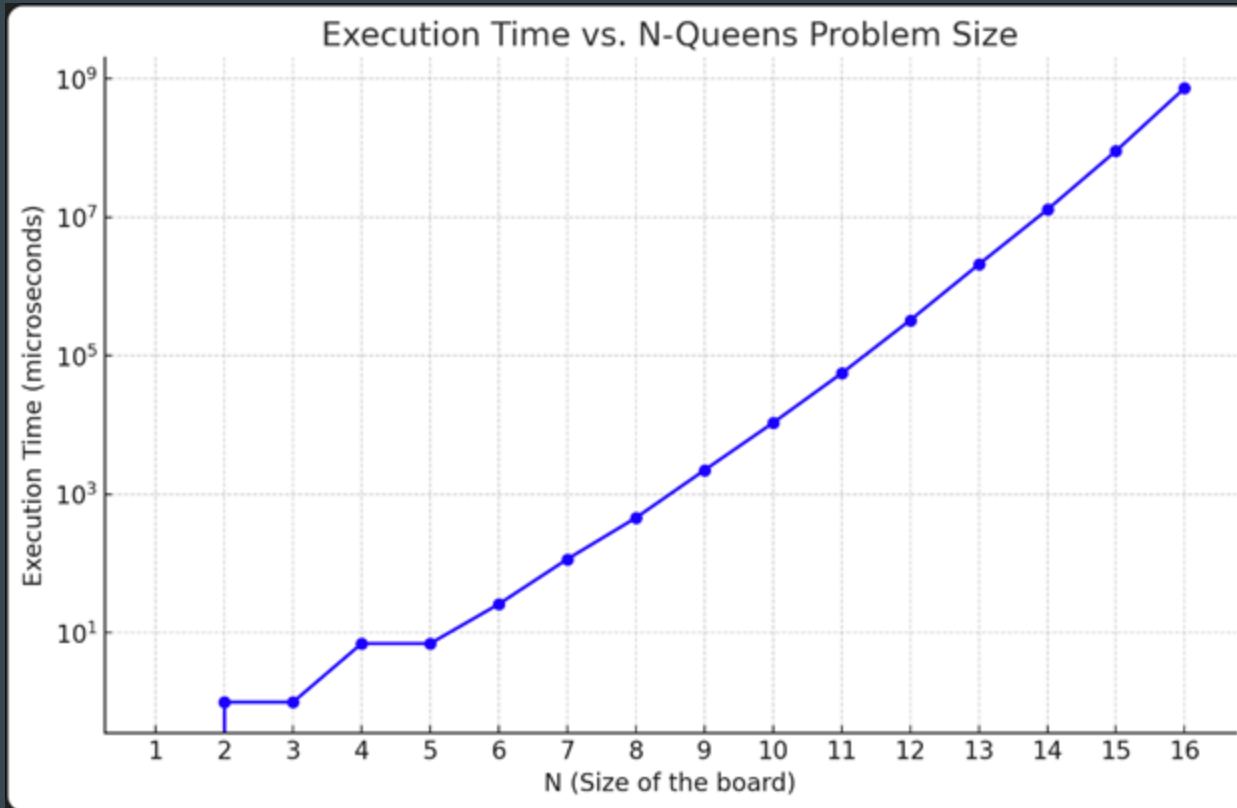
Performance and Outcomes

- Execution times and number of solutions for various board sizes were compared. Parallel and distributed methods showed improved performance and scalability.
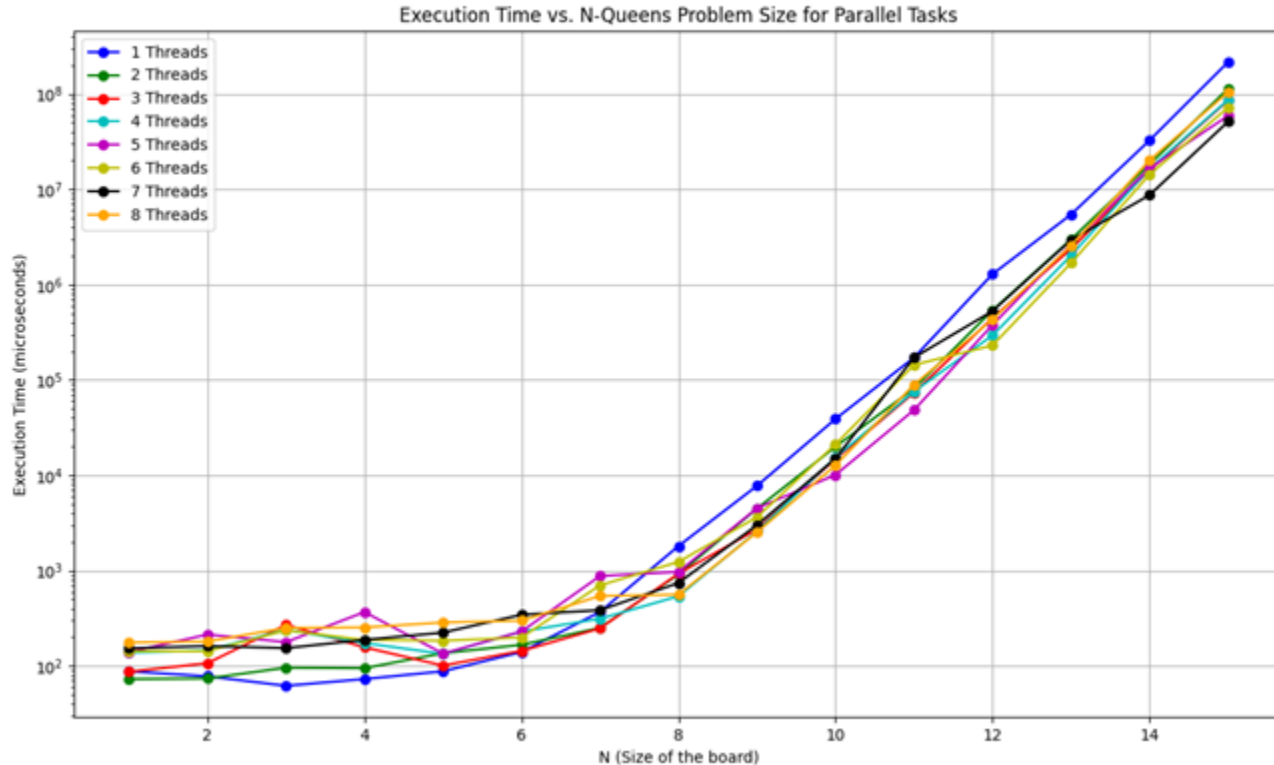
Placeholderfor sequential run image

Placeholder for distributed run
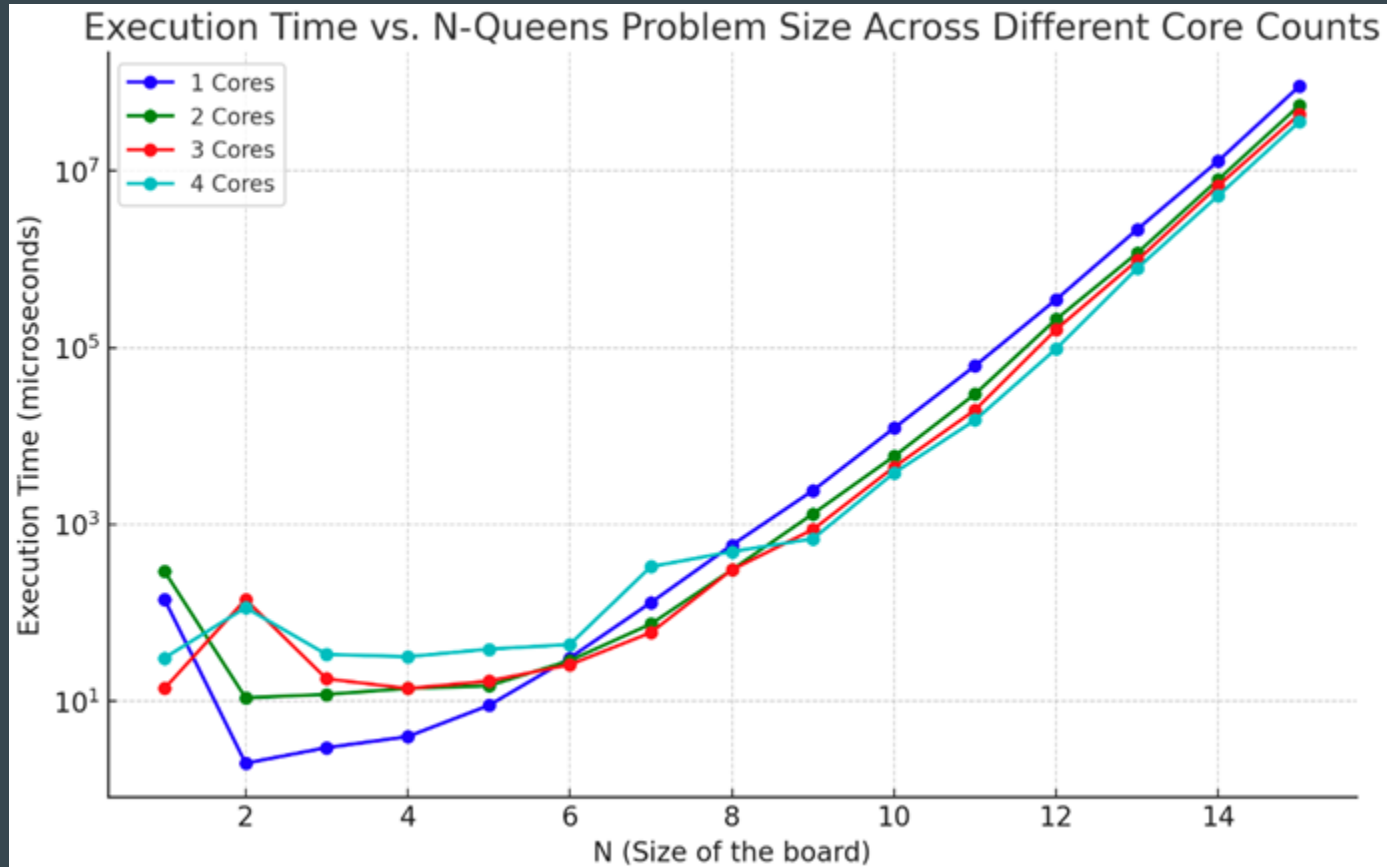
Sequential Solution

Execution Time vs. N-Queens Problem Size

Parallel Tasks

Execution Time vs. N-Queens Problem Size for Parallel Tasks

Multi Core Solution

Execution Time vs. N-Queens Problem Size Across Different Core Counts

Conclusion and Future Work

- Each approach has its strengths and can be further optimized. Future work could explore hybrid approaches and further optimize thread management.

Thank You and Q&A

- Thank you for your attention!