Optimizing Azul Scores Using Hybrid Parallelism

Problem Description: Azul is a tile-laying board game in which players score points by placing colored tiles onto their personal boards to complete rows, columns, and color sets. On each turn, a player chooses tiles from either one of several factory displays or the center of the table, and must place all selected tiles of the same color onto a row of their board's pattern lines. Players are limited by which color tiles and what number of tiles are available, so for this study it will be assumed that **Players may take at most four tiles of the same color in a single round**, which constrains the number of tiles they can collect and affects subsequent scoring opportunities.

Points are scored when tiles are placed onto the wall: completing a row, completing a column, or completing all tiles of a particular color yields bonus points. Additionally, adjacency of tiles provides incremental scoring, and tiles that cannot be placed immediately incur penalties. The goal is to determine the **maximum achievable score** from a given starting configuration, accounting for both placement rules and the four-tile-per-color restriction, and within a given time limit so that I can compare performance even when the calculation doesn't finish within several hours.

This problem is combinatorial and highly sequential at the tile-placement level: each choice affects the board state and future options. The branching factor of possible moves grows rapidly with each round, making exhaustive search computationally expensive and ideal for parallel computation.

Computing Platform: I will use `slurm` on the ODU HPC platform and will probably use python to do the calculations (TBD).

Planned Hybrid Parallelism:

1. **Task-level parallelism With MPI** to handle the multiple decision trees.
2. **Data-level parallelism with CUDA** within each round, calculate all scores at once with GPU.

Anticipated Challenges:

- **Memory management:** Representing multiple board states in GPU memory efficiently without excessive copying.
- **Branching factor:** Azul's game tree grows quickly; the first round alone with the current contraints have 71 different distributions of tiles *without factoring in color*. Finding a way to prune the tree will be the difference between an unfathomable number of outcomes and finding a result.