Bothello: A Multi-Strategy, CUDA-Accelerated Othello Engine

Comparative Analysis of Parallel MCTS Architectures on GPU

N. Antonelli-Dziri, A. Morille, F. Papini

Problem Statement

Othello requires efficient search to avoid tactical losses. While Monte-Carlo Tree Search (MCTS) is effective, sequential execution creates bottlenecks. This project compares three GPU parallelization strategies to determine which best overcomes GPU limitations (synchronization, thread divergence) for maximum playing strength.

Implementation Strategy

Division: Each member implements one GPU architecture against a common CPU baseline.

- **CPU Baseline (Shared):** Single-threaded UCT-MCTS reference engine
- **Student 1 — Leaf Parallelization:** GPU-offloaded playouts; optimize throughput and minimize divergence
- **Student 2 — Tree Parallelization:** Shared MCTS tree with atomic operations; manage contention
- **Student 3 — Block Parallelization:** Independent trees per block using shared memory; optimize memory access

Evaluation Framework

Performance KPIs

- **Playouts per Second:** Simulation throughput
- **Time per Move:** Search time per decision
- **Speedup:** GPU/CPU PPS ratio
- **Scaling:** PPS vs. threads/blocks

GPU Profiling KPIs

- **Memory Bandwidth:** Utilization (Nsight Compute)
- **Occupancy:** Active vs. max warps
- **Divergence:** Branch cost (Nsight Systems)

Game Strength (Tournament)

- **Win Rate:** GPU variants vs. CPU baseline (1000 games)
- **Strategy Comparison:** Cross-GPU performance at fixed time budgets
- **Elo Rating:** Absolute strength vs. external engines

Expected Outcomes

- Quantitative comparison of speedup and playing strength across parallelization strategies
- Architectural insights into GPU bottlenecks (contention, divergence, occupancy) for each approach
- Practical guidelines for high-performance parallel game tree search on GPUs

References: Parallel Monte-Carlo Tree Search (Chaslot et al.); Parallel UCT Search on GPUs (Barriga et al.); Large-Scale Parallel Monte Carlo Tree Search on GPU (Rocki et al.)