Lab Course: Efficient Programming of Multicore Processors and Supercomputers

Group 1: Jonas Mayer, Paul Preißner, Konrad Pröll

July 4, 2017

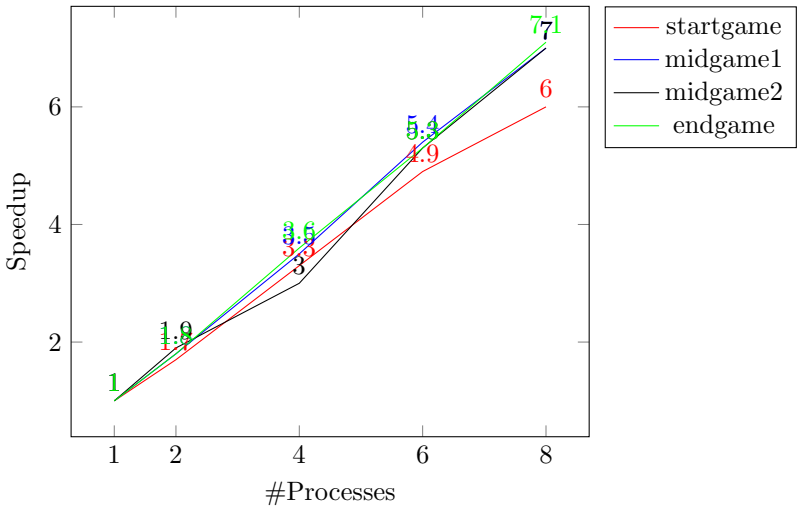
6.2 Parallelization of Minimax

What is a good decomposition strategy of the workload to get load balance?

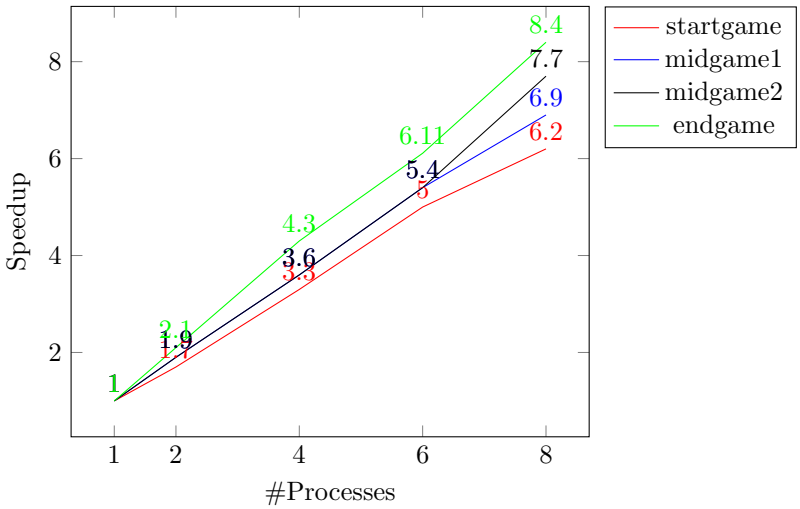
We decided to distribute the tasks of evaluating the topmost layer of nodes in a roundrobin fashion between the processes. For more sophisticated algorithms like alpha-beta-pruning, this is obviously not a very good design, but for minimax, this leads to a reasonable load balance, because everyone of those nodes is exactly the same amount of work.

What is the speedup you get for the different scenarios from subtask 1, related to the performance metric "evaluations/sec", using 2,4,6,8 processes?

Speedup for Minimax with depth 2



Speedup for Minimax with depth 3



I have absolutely no idea what happened with depth 3 in the position endgame, but those results were reproducable.

When and why does the achievable speedup depend on the current game position?

The game position has a huge effect on the shape of the game tree. Early on, nearly every leave has the same depth (i. e. _maxdepth), while in later situations, there might some leaves with shorter depths, because an earlier move would end the game. As those effects depend largely on the setting of _maxdepth, which we were not able to run to high because of the bad performance of our algorithm, above numbers do not necessarily show that. But this issue should still exist with alpha-beta-pruning, so we should come up with a more reasonable decomposition.