Homework 1, Part 2

Cooper Faber

April 23, 2021

1 Write-Up

For the middle part of this assignment, we investigated performance improvements regarding Loop Unrolling and ILP with regards to dependent computations. There were 2 timed loops - a reference, which was provided, and a reduction loop, which partitions the given array into N partitions, where N is given at runtime. In the below figure, the reference loop is the green line, and the reduction loop is red.

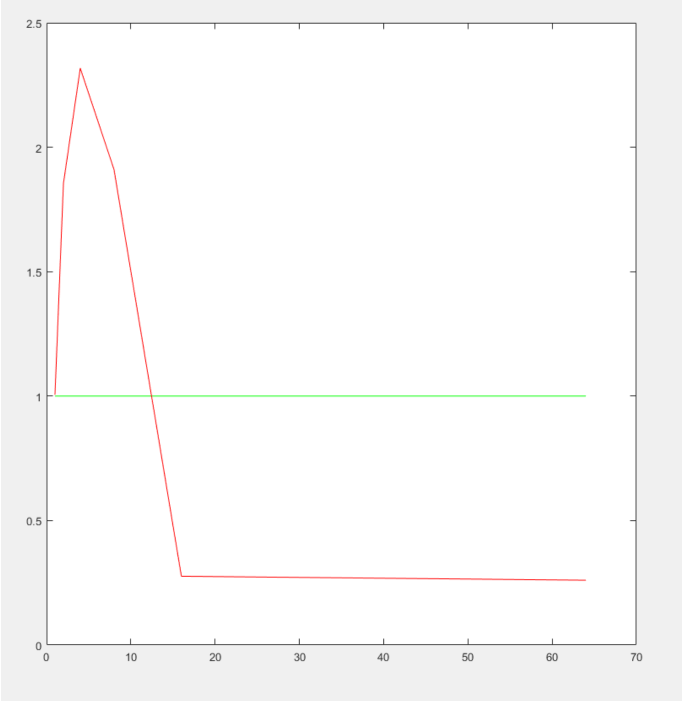

Figure 1: Results

2 Analysis

Surprisingly, the speedup sharply drops between 4 partitions and 16, becoming significantly slower as N increases (and hits a probable maximum at 4 partitions). A likely explanation for this is that the overhead for partitioning the array becomes significant, causing performance issues. Additionally, the reduction loop requires a cleanup at the end where all partitions are summed into the 0th slot, which scales based on the amount of partitions. This, too, could cause significant overhead as N is increased. Finally, as N is increased, the relative difference in simultaneous computation goes down, as overhead increases. Going from 1 to 2 partitions, for example, doubles the amount of possible simultaneous computations, but only increases the final loop by 1 iteration. Following this trend shows that further partitions increase the size of the cleanup loop linearly, but do not increase the speed of the computations linearly. This results in a negative trend as N increases, following the above graph.