Assignment 2: MPI

Elijah Kin

October 10, 2024

Data Distribution

We first have process 0 read in the entire input file to a `global_life` 1D array. We then scatter this array across all processes, each receiving a chunk of the board into their own `life` array (also now 1D). Each process also has its own local `previous_life` array.

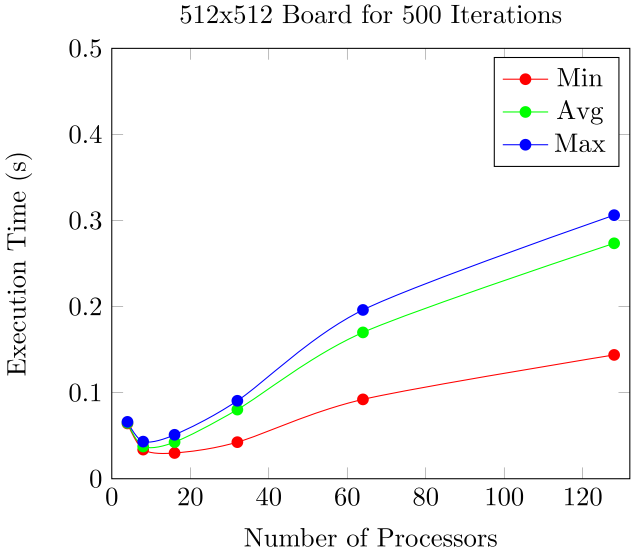
Then in each generation, each process sends the top and bottom rows of its `life` array to the ghost rows in the `previous_life` arrays of its neighboring processes. Finally after performing all iterations, we gather from all processes back into the `global_life` array on process 0, before writing to an output file.

Performance Results

Running the program on the 512x512 board for 500 iterations as a benchmark, we measure the following execution times on Zaratan.

Processes	4	8	16	32	64	128
Min (s)	0.064236	0.03386	0.029902	0.042488	0.092184	0.14385
Avg (s)	0.0649503	0.0374385	0.0425478	0.0803294	0.169988	0.273515
Max (s)	0.066074	0.043119	0.051057	0.090568	0.196102	0.306212

We also display the same data in a line plot below.



Observe that the plots for Avg and Max are quite close to each other, suggesting an approximately equal work distribution. Further, we expect that in the limit, Min is approximately half of Avg, since the first and last processes only communicate half as much as the others.