HY-342 Assignment 2
Game of Life

Nikoletta Arvaniti csd4844

1. Average Speedup for 1, 2 and 4

glid1.txt

(100 generations)

Average time for 1 thread: 2.327228 seconds

Average time for 2 threads: 1.247953 seconds

Average time for 4 threads: 0.669406 seconds

Speedup for 2: AVG1Thread / AVG2Threads = 1.87

Speedup for 4: AVG1Thread1 / AVG2Threads = 3.48

(1000 generations)

Average time for 1 thread: 48.748606 seconds

Average time for 2 threads: 24.79116 seconds

Average time for 4 threads: 13.12771 seconds

Speedup for 2: AVG1Thread / AVG2Threads = 1.97

Speedup for 4: AVG1Thread / AVG4Threads = 3.71

glid2.txt

(100 generations)

Average time for 1 thread: 237.950964seconds

Average time for 2 threads: 121.777917 seconds

Average time for 4 threads: 69.162317 seconds

Speedup for 2: AVG1Thread / AVG2Threads = 1.95

Speedup for 4: AVG1Thread1 / AVG2Threads = 3.44

(1000 generations)

Average time for 1 thread: 2395.194044 seconds

Average time for 2 threads: 1227.118204 seconds

Average time for 4 threads: 713.629278 seconds

Speedup for 2: AVG1Thread / AVG2Threads = 1.95

Speedup for 4: AVG1Thread / AVG4Threads = 3.36

2. Variance of execution time for each configuration

glid1.txt

(100 generations)

- 1 thread -> variance: 0.00005

- 2 threads -> variance: 0.00025

- 4 threads -> variance: 0.00019

(1000 generations)

- 1 thread -> variance: 7.83×10^{-6}

- 2 threads -> variance: 2.56×10^{-5}

- 4 threads -> variance: 1.21×10^{-5}

glid2.txt

(100 generations)

- 1 thread -> variance: 0.0633

- 2 threads -> variance: 1.14

- 4 threads -> variance: 0.0611

(1000 generations)

- 1 thread -> variance: 116.10

- 2 threads -> variance: 5.54

- 4 threads -> variance: 655.83

3. Overhead with parallelism 1 vs. without -fopenmp

Even when using a single thread, OpenMP introduces overhead due to the initialization of data structures required for thread management, which adds extra cost compared to a serial program. The compiler also inhibits certain optimizations when used with OpenMP. Source: [Stack Overflow](#)

4. Difference in speedup based on generations and the size of input

The speedup slowly increases as the number of generations grows because computational work starts overwhelming the overhead. When, however, the size of input (glid2.txt) is greater, the speedup does not improve as much due to memory constraints, additional cache misses, and potential synchronization overhead.

PS: I run the .c in karydi.csd.uoc.gr, it was okay for glid1.txt but I did not have enough space to run the glid2.txt so I did it locally. (Mac OS M2)