ACSE Assessment 3 Report

Description & Usage

The GitHub respiratory contains the main (Game_of_Life.cpp) and Visual Studio project file. The Game_of_Life.cpp contains initialization parameters in the very beginning (e.g. a Boolean variable for periodicity, an integer variable for the grid size (**note**: this is the grid size for each process and not the total lattice size – this can be adjusted to roughly get the desired overall grid size), and an integer variable for the total number of iterations. When executed, this will create files with iteration numbers and ids for each process in the debug folder. Merge matrices.py is the main post-processing script which merges files for each iteration into a new file called “iteration_%(iteration_number).txt” as well as outputting the image files “iteration_%(iteration_number).jpg” (**note**: the user should input the number of iterations, number of rows and columns of the processes – an example is also provided in the GitHub respiratory. Also, all the iteration files must be in the same folder). An animation script animation.py is also provided which uses the jpg image outputs from Mergematrices.py to create a Gif video (example Gif outputs for periodic and non-periodic boundary conditions are provided at GitHub).

Performance

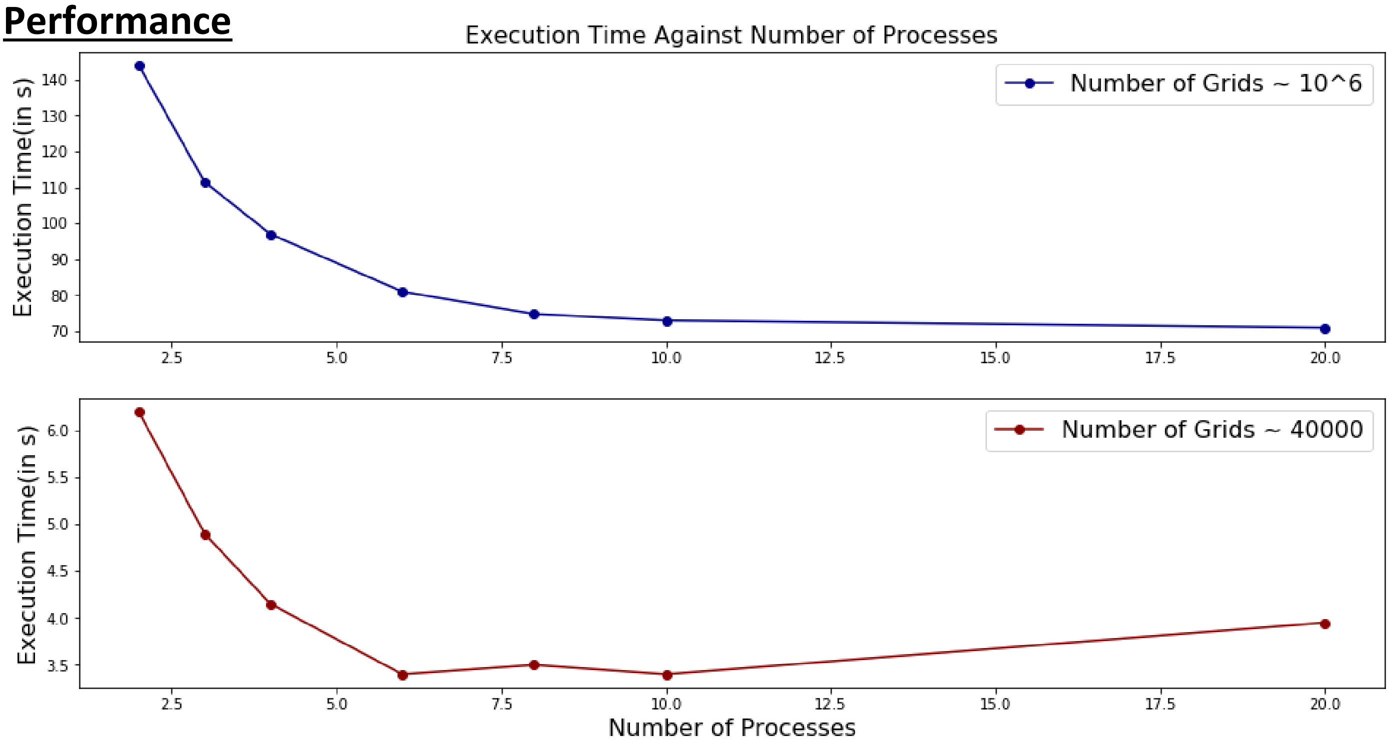
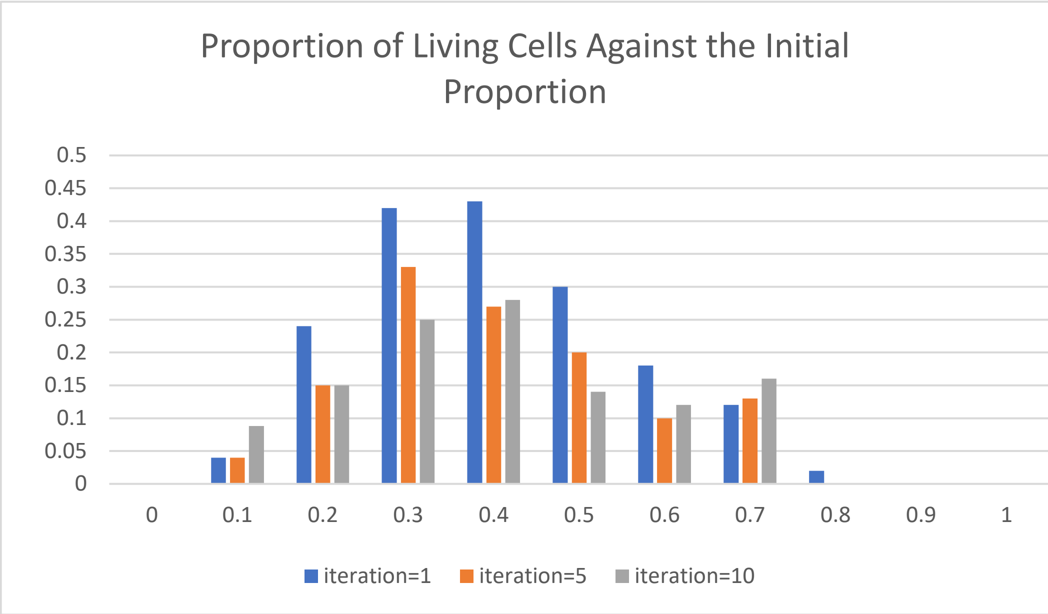


Fig 2: Shows execution time (on a quad core CPU) as a function of number of processes for different periodic grid sizes.

The number of grids in each process was calculated as round ($\sqrt{\#of Grids/\#of Processes}$). From the above plot, it can be observed that there is a roughly linear relationship between the number of grid points and the execution time (i.e. with 2 processors increasing the grid number by 25 times increased the execution time by slightly less than 25 times). This is very indicative that in the regime where message passing time is not significant compared to the computation time (e.g. large grid sizes per processor), the performance of the parallel algorithm can improve significantly with increasing the number of processors. This is what we observe in the above plots where doubling number of processors (from 2 to 4) reduced execution time by 30%. Slightly more than 4 processors also improved the performance slightly (i.e. virtual cores also improve the performance). In the 40000-grid plot, after 10 processes, the execution time starts to rise. This is perhaps due to grid sizes per processors getting sufficiently small such that the time spent waiting for messages to arrive is getting significant compared to the time spent doing the actual computation. However, this is not the case for the 1000000-grid plot since to grid sizes per processors is still sufficiently large at 20 processors. This is indicative that at grid size of 1000000, running the same simulation on a supercomputer with many nodes could continue to yield significant performance improvements similar to the ones observed when moving from 2 to 4 processors.

Pattern of Evolution



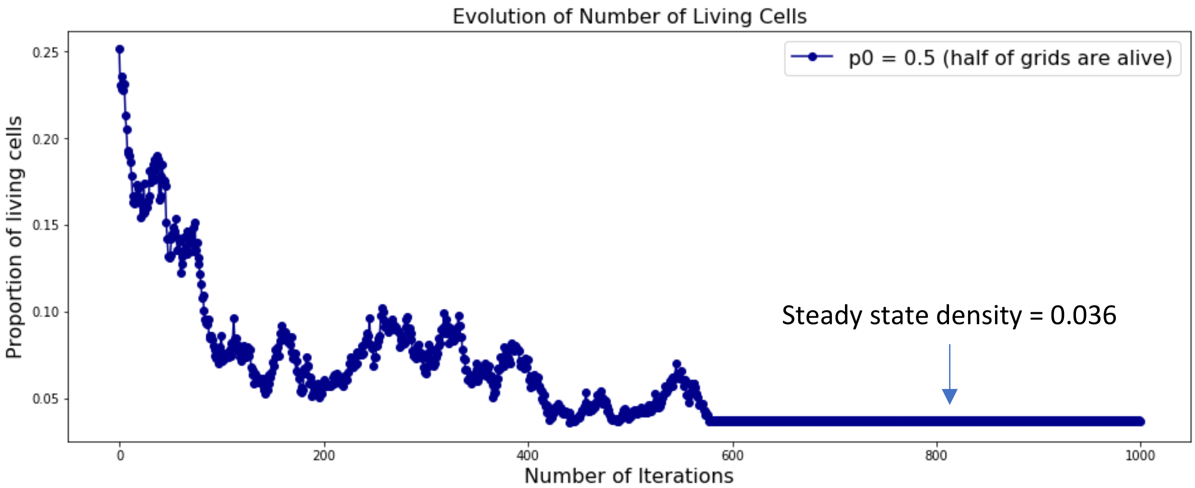


Fig 2: The first figure shows the proportion of living cells after 1, 5, and 10 iterations for various initial proportions (for a periodic grid size of 20 by 20). The second figure shows a longer evolution pattern (for 1000 iterations) for a periodic grid size of 100 by 100.

The first figure shows that in the short-term initial proportion of 0.3-0.4 gives the highest number of living cell proportions. This is intuitively understandable given that a cell needs 2 or 3 out of its 8 nearest neighbours to stay alive, 3/8 to come to life, and ≥ 4/8 or < 2/8 to die. The expectation value of a given cell's number of living neighbours is equal to the initial proportion of the living cells and, therefore, it makes sense that proportions between 2/8 and 4/8 (25% to 50%) are the most favoured ones for maximizing the number of living cells in the short term.

F Bagnoli et al. [1] were the first to investigate the game of life with random initial conditions. Their mean field analysis and experiments suggest that the system approaches a nontrivial asymptotic density for all $0<p_0<1$ (figure 2b also shows this asymptotic behaviour). For initial proportions in the range $0.15<p_0<0.75$ they generalised the behaviour of this system into 3 intervals. First, a region extending from t=0 to $t \sim L^{1/2}$ (0 to ~10 for fig 2) presenting large fluctuations in p; second, a scaling region characterized by a power-law dependence between $t \sim L^{1/2}$ to $t \sim L^{4/3}$ (10 to 464); and finally, a steady state extending from $L^{3/4}$ to infinite. Indeed, our experiment in Fig 2b also shows a similar trend as the steady state configuration is reached somewhere between 400 and 600 ($\sim L^{4/3}$=464).

Reference

[1] J. B. C. Garcia, M. A. F. Gomes, T. I. Jyh, T. I. Ren, and T. R. M. Sales. Nonlinear dynamics of the cellular-automaton ‘‘game of Life’’. Phys. Rev. E 48, 3345–3351 (1993)