

MPI Assignment 2 – The Game of Life

The aim of this assignment is to write a parallel version of Conway's Game of Life (details are in the lecture notes associated with this assignment and can also be found online).

You must use domain decomposition in which each processor only allocates enough memory to store its portion of the domain (together with any padding required to receive the data from neighbouring processes). The easiest way to divide the domain is into vertical or horizontal strips, but this can become inefficient quite rapidly as the number of processors grows as it will result in very long thin domains with a large number of boundary elements relative to their area. Far more efficient is to divide the simulation into rectangular domains that are as close to square as possible. The communication pattern involved is virtually identical to that implemented in Exercise 3 of worksheet 2 and will involve transferring a layer of grid cells just inside the edge of the domain from the neighbouring processors and receiving a layer of grid cells situated just outside the domain back from the neighbours. This will take the form of peer-to-peer communications with no master process.

I want you to be able to run the simulation with either fixed or periodic boundaries (have a constant which decides what type of boundary is to be used). With periodic boundaries you need to maintain communications across the periodic boundary. Watch out for the situation where the matching periodic boundary is on the same processes (this can easily occur when using only a few processes).

You should make the code able to use an overall domain of arbitrary size so that you can test your simulator's efficiency as the domain size is changed. You can decide on the initial configuration of the cells, which can be random if you wish (how is the pattern evolution influenced by the proportion of live cells?). You can generate these initial distributions on the various processes rather than having them be generated on a single process and then transferred.

You should carry out the simulation for a specified number of steps. At each step each process should generate their own output file. You should therefore write a separate program or script which collates these files. This can be done in Python in order to be able to easily generate graphical outputs.

In addition to documented source code you should also hand in a short analysis of the performance of the code.

The due date for the assignment is midnight on Sunday 17 March.