MPI Programming Assignment– The Game of Life

He Zhu

Feb, 23, 2020

1. Introduction

In a brief description, the MPI version of Conway's game of life is realized by two different approaches. Specifically, the first one is to divide the domain into various horizontal strips. However, it is not efficient for very long thin domains. The second approach is to divide the domain into rectangular domains. Meanwhile, it will ensure that domains are divided as close to square.

This report will introduce the communication design between each domain and the performance of Efficiency and Speedup as changing with the size of the problem and the number of cores used.

2. Approach Analysis

(a) The approach of horizontal strips. Here, we use four processors as an example. As figure 1 shown, the domain is divided into four different horizontal strips(blue area). The number represents the processor id number. The two more rows need to be added to receive the data from process above or below. Gray area is empty area and green area represents received data.

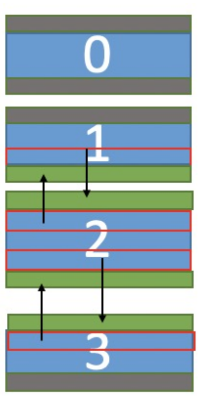

Figura 1: The diagram of strip approach

Each processor needs to communicate with other processors just like id '2'.

(b) The approach of rectangular domains. In this approach, the processor id number needs to be arranged in 2D coordinates. Here, we use nine processors as an example, the specific diagram is shown in figure 2. Therefore, the exact location of each process can be easily represented using corresponding x- and y- coordinate.

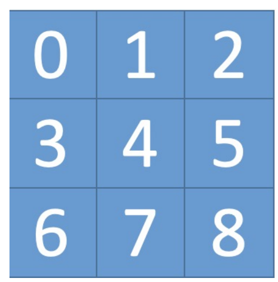

Figura 2: Coordinate arrangements

The following figure 3 illustrates the specific communication method between each processor. Each processor first communicates with the left and right neighbours. Then it communicates with above and below processors. It is worth noting that the number of transmitted data between horizontal processors is different from between vertical processors. It ensures that the data in corner will be correctly transmitted.

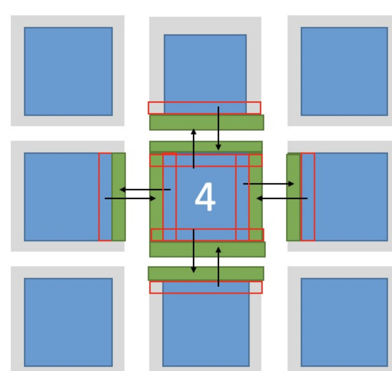

Figura 3: The diagram of rectangular domain model

3. Performance Analysis

This section explores how the Efficiency and Speedup ratio changes with the number of processors used and the size of the problem. The following simulation results are obtained in the periodic condition and the number of iteration is set to be 200. As the figure 4 and 5 displayed, three matrices with different sizes are simulated by two approaches to demonstrate the corresponding parallel efficiency.

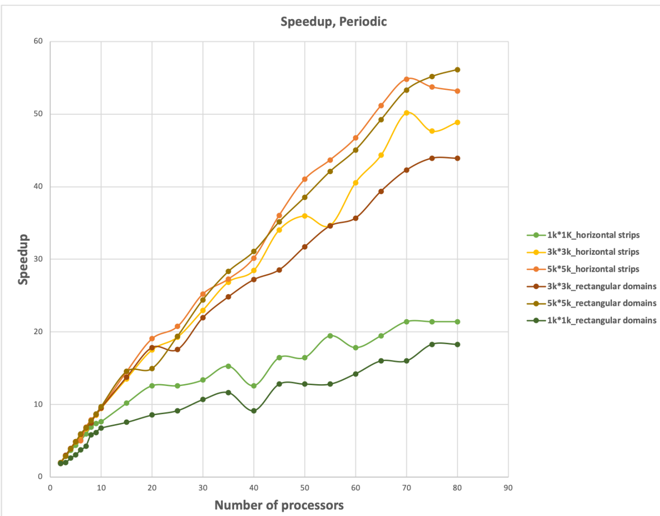

Figura 4: Speedup, periodic

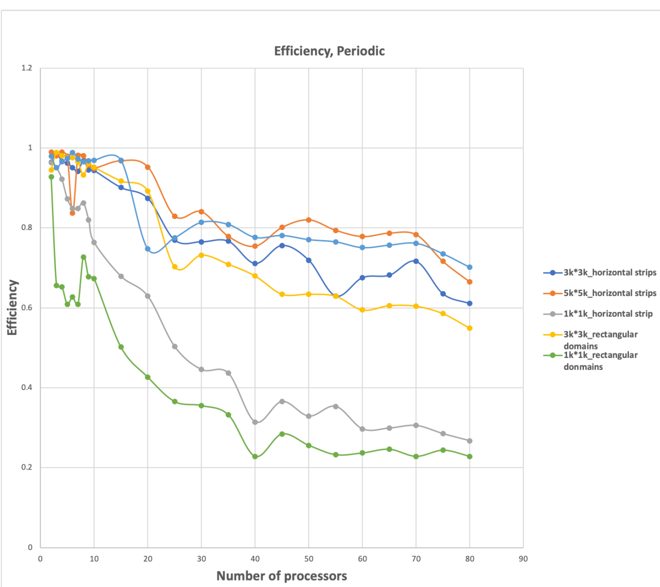

Figura 5: Efficiency, periodic

We already know the speedup ratio increases as increasing the number of processors and it will eventually reach speedup bottleneck. Meanwhile, the speedup simulation result indicates the effect of speedup is more significant on large size matrix. The main reason is that the communication time between each domain of small size matrix takes an significant proportion of the simulation time and it eventually affects the speedup performance.

In the efficiency simulation of different problem size, the efficiency decreases as the number of processes increases. When the calculation time becomes smaller and smaller, until it is smaller than the communication time, the communication delay cannot be masked and the communication and synchronization cost will increase. Therefore, the higher the degree of parallelism, the worse the speedup ratio will be, which also leads to a decrease in efficiency. The speedup and efficiency simulation results indicate a suitable of degree of parallelism cannot only increase the speed of execution but also not waste too much resources.

In addition, a notable result is that the horizontal strips approach has better speedup and efficiency performance than using rectangular domains approach, which seems to be in consistent with the initial expectations. This is mainly because that we only choose square matrix to perform simulation and the size of matrix is not quite large. In very long thin domains with a great amount of boundary elements, the horizontal strips approach will become inefficient quite rapidly as the number of processors. Moreover, rectangular domains approach has more sub-matrix, which generates the greater number of transmission and synchronizations.

In the end, due to the unstable performance of HPC, the results plotted in figure 4 and 5 fluctuate with increasing the number of processors. This requires data to be simulated multiple times and then averaged.