Universitatea Politehnica Timișoara
Facultatea de Automatică și Calculatoare
Departamentul Calculatoare și
Tehnologia Informației

Parallel Algorithms Project: Game of life with MPI and OpenMPI

Student:
Vladimir NIȚU-ANTONIE

Game of life without parallelization

The Game of Life, also known simply as Life, is a cellular automaton devised by the British mathematician John Horton Conway in 1970.

The cells from *Game of Life* have a set of rules in order to survive:

- if a cell is alive(value is 1) but has less than 2 or more than 3 neighbors that are alive (cells next to him have the value of 1) the cell will die in the next generation
- if a cell is dead(value is 0) but has exactly 3 neighbors that are alive, then the cell will be alive in the next generation

The algorithm for the simple Game of Life was inspired from [1] and has the following steps:

- the rows, columns, episodes and the file from where the initial episode for the cells are read from the keyboard;
- the initial matrix is created;
- for each episode, the matrix is verified with the conditions from the game of life rules in order to find if the the cells are dead or alive;
- after the final episode is finished, the matrix together with its final state is written and the time of the algorithm is computed.

Game of Life with parallelization: MPI

For the installation of the MPI, I used the following installation guide [2], it will only work for Visual Studio.

For the parallelization I rewrite the initial algorithm as it follows:

- the rows, columns, episodes and the file from where the initial episode for the cells are read from the keyboard;
- the struct MatrixDetails is used to announce where the change will happen;
- the MPI is initialized, we will use 2 arrays in order to compute the final matrix;

- the initial matrix is initialized;
- the algorithm will calculate how much to process in each task and which is the starting point for each task;
- it is needed to verify if the task needs data from another task;
- after we make sure that each task does not interfere on the same row or column, the next step is to calculate how many changes happened in the matrix;
- Since MPI-recv and MPI-send could have problems with synchronization if the the MPI-Wait is not done correctly, the algorithm will use another MPI provided function MPI-Allgather [4] that is thread-safe and does not need the actual lock;
- the next step will be to calculate what each task will send or display, for this the MPI-Allgatherv is used [5], which is also thread-safe and allows to the received messages to have different lengths and be stored at arbitrary locations in the receive buffer;
- the matrix with its final state is obtained and after the time is ellapsed, all the data is written in a file.

Game of Life with parallelization: OPENMPI

For the installation of the MPI, I used the following installation guide [3], it will only work for Visual Studio.

For this algorithm I used the initial that I have used for the following steps:

- the rows, columns,chunks,processes, episodes and the file from where the initial episode, for the cells are read from the keyboard;
- the initial matrix is created;
- for each episode, the matrix is verified with the conditions from the game of life rules in order to find if the the cells are dead or alive;
- this time the omp is used, it will use paralellization for each cell and the neighbours' calculation;

- the dynamic schedule will be used with the chunks that were received before;
- after the final episode is finished, the matrix together with its final state is written and the time of the algorithm is computed.

How to use

Command for normal Game of life: gameOfLife.exe nrRow nrCol episoded filepath.
Command for Game of life with MPI: mpiexec -N nrThreads gameOfLife-mpi.exe nrRow nrCol episoded filepath.
Command for Game of life with OpenMPI: gameOfLife.exe nrRow nrCol episoded filepath chunckSize nrThreads.

Statistics

Rows	Collumns	Episodes	Algorithm	ThreadNr	Chunks	Time
50	50	10	GOL	1	0	0.001000
50	50	10	MPI	5	0	0.002000
50	50	10	OpenMPI	5	10	0.002000
50	50	100	GOL	1	0	0.150000
50	50	100	MPI	5	0	0.014000
50	50	100	OpenMPI	5	10	0.013000
50	50	1000	GOL	1	0	0.150000
50	50	1000	MPI	5	0	0.071000
50	50	1000	OpenMPI	5	10	0.100000
100	100	1000	GOL	1	0	0.678000
100	100	1000	MPI	5	0	0.2670000
100	100	1000	OpenMPI	5	10	0.321000
150	150	1000	GOL	1	0	1.476000
150	150	1000	MPI	5	0	0.616000
150	150	1000	OpenMPI	5	10	0.759000
500	500	1000	GOL	1	0	11.492000
500	500	1000	MPI	5	0	6.333000
500	500	1000	OpenMPI	5	10	7.4980000

In conclusion, for the computed time, the Open MPI seems to be from the time perspective with 45 percent faster than the normal Game of Life, where MPI seems to be at almost 55 percent faster.

Bibliography

[1] gameoflife. tutorial: https://www.youtube.com/watch?v=FWSR$_{7}kZuYgab_{c}hannel = TheCodingTrain.$

[2] Installmpi. tutorial:https://www.youtube.com/watch?v=BA$_{M}qi3a9HIt = 349sab_{c}hannel = WilliamWilliam.$

[3] Installopenmpi. tutorial: https://www.youtube.com/watch?v=hQOkxFspN2Aab$_{c}hannel = NurulHanisahNajwa.$

[4] Mpi-allgather. tutorial:https://www.mpich.org/static/docs/latest/www3/MPI$_{A}llgather.htm$

[5] Mpi-allgatherv. https://www.rookiehpc.com/mpi/docs/mpi$_{a}llgatherv.php.$