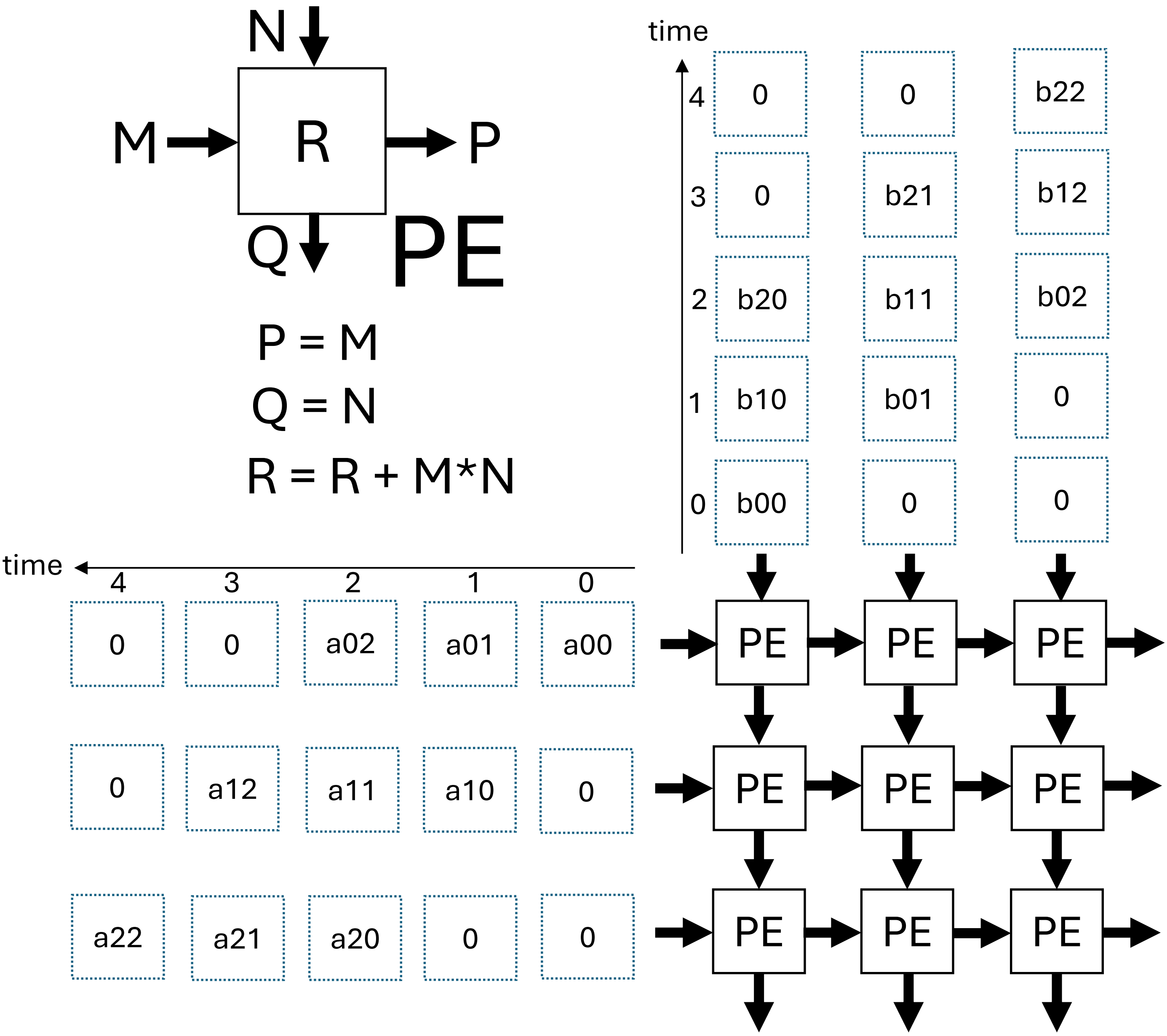
Parallelizing and distributing Systolic array operations

Modern applications are based on specialized and efficient operations. In some cases, the structure of a distributed application is organized to speed up its execution with homogeneous blocks. In computers, the organization and connection of these homogeneous blocks can be known as a systolic array.

For instance, the execution of a matrix multiplication can be organized for the execution of a set of fixed operations of a routine in independent processing elements (PE).

$$\begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{bmatrix}$$



P = M

Q = N

R = R + M*N

time			
4	0	0	b22
3	0	b21	b12
2	b20	b11	b02
1	b10	b01	0
0	b00	0	0

4	3	2	1	0
0	0	a02	a01	a00
0	a12	a11	a10	0
a22	a21	a20	0	0

In the example in the Figure, 9 PEs are described and organized to execute the same operations with different data (e.g., a00, a01, etc.) that propagates among them for the evaluation times (from 0 to 4). At the end, each PE stores one element on the result matrix C.

a) Describe in MPI the PE structures and the systolic array to operate the matrix multiplication of two inputs A and B with dimensions:

- (500x500)

- (1,000x1,000)

- (2,000x2,000)

Use input .csv files to provide the input matrices and one .csv file to store the output C matrix.

b) Compare the execution of the MPI description with a CUDA kernel used to execute the matrix multiplication of two inputs with dimensions (2,000x2,000).

- identify the main performance bottlenecks in both cases.

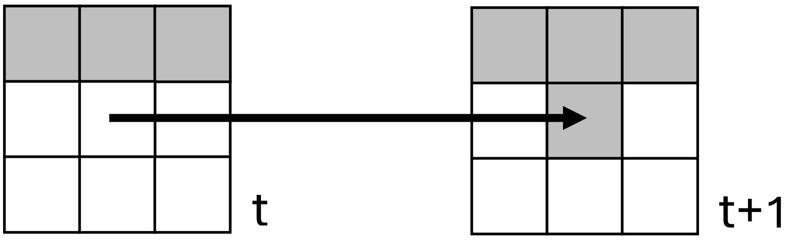
- identify the advantages and restrictions of each strategy.

Parallelizing Conway's game of life

Conway's Game of Life is a cellular-automaton, zero player game, developed by mathematician John Conway in 1970. The game is played on an infinite grid of 2-D square cells, and its evolution is only determined by its initial state.

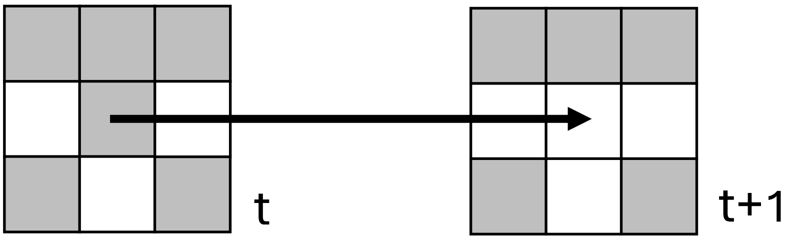
The rules of the game are simple, and describe the evolution of the grid:

Birth: a cell that is dead at time **t** will be alive at time **t + 1** if exactly 3 of its eight neighbors were alive at time **t**.

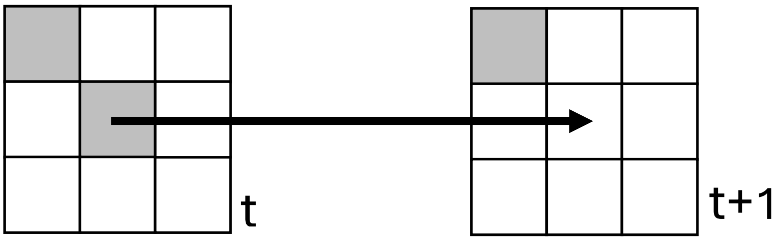


Death: a cell can die by:

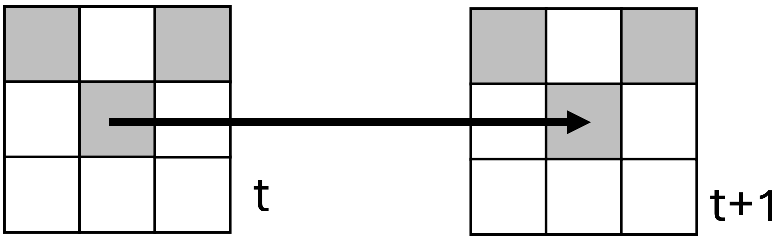
- Overcrowding: if a cell is alive at time **t** and 4 or more of its neighbors are also alive at time **t**, the cell will be dead at time **t + 1**.



- Exposure: If a live cell at time **t** has only 1 live neighbor or no live neighbors, it will be dead at time **t + 1**.



Survival: a cell survives from time **t** to time **t + 1** if and only if 2 or 3 of its neighbors are alive at time **t**.



Starting from the initial configuration, these rules are applied, and the game board evolves across generations (**t**), playing the game by itself!

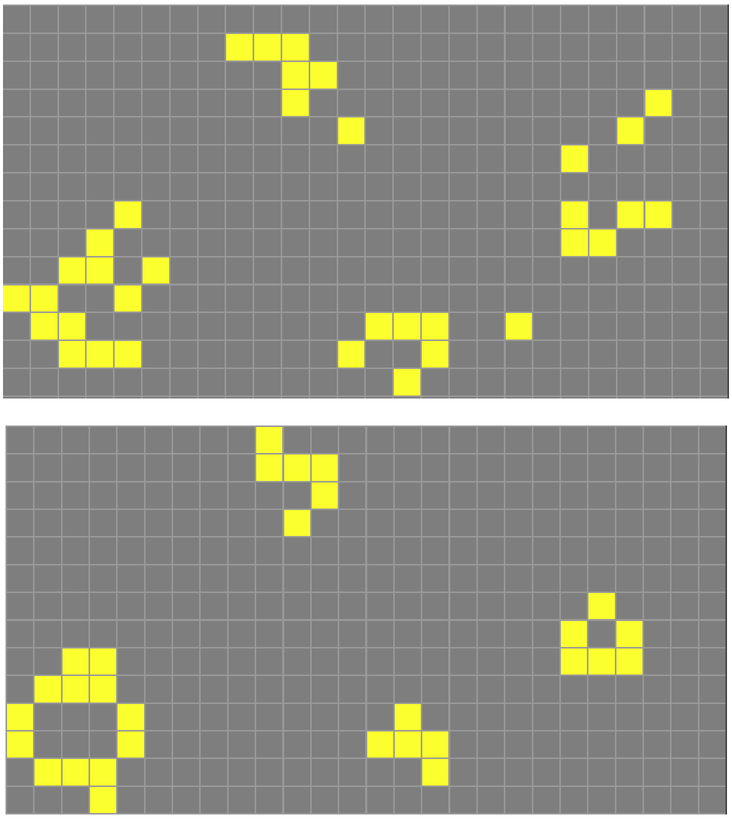


Figure 1. Example of the evolution of the game after one generation

Employ any of the parallelism and distributed computing techniques studied in the course to implement the Conway's Game of Life for a universe with 1,000 x 1,000 boxes. Use an input text file to provide the initialization state for the game.

For each cell, store the number of generations that a cell was alive and the number of generations that each cell remains consecutive alive.

The solution must execute and store the results of the game after 500 generations.

Fundamentals of multi-dimensional data processing

In several applications, including modern neural networks, computer vision, and astrophysics, data pre- and post-processing are used to extract several characteristics from an input to a system. These techniques resort to filters (stencil elements) to collect information from the input and analyze information (e.g., borders, or main components).

In 2-dimensional applications, such as multimedia, the stencils/filters are applied to all input elements of an image to calculate, or process features from the image.

The following equation describes an example of an stencil/filter w used to calculate the weighted average g(x,y) from an input f(x,y), which is used to remove noise from images.

In this case, the filter w is applied to each input pixel of the image as described by the equation. The borders of the resulting image g can be replaced by zero values.

$$g_{(x,y)} = \frac{1}{16} \sum_{j=-1}^{1} \sum_{i=-1}^{1} w_{(i,j)} * f_{(x+i,y+j)}$$

$$W = \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 3 & 4 & 3 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

The common steps to use a filter are:

1 - Decomposition of the image in their main components: e.g., the RGB image is composed of three components (R, G, and B).

2- Execution of the equation on each input pixel of the three components (e.g., R, G and B) of the image to calculate g(x,y).

3- Assembly of the three components (e.g., R, G, B) into an image.

A similar procedure can be applied on videos, by processing each frame.

a) Use MPI or OpenMP to provide a pre-processing solutions and apply the weighted average filter on 4K input images (3840 x 2160 pixels). Evaluate the effectiveness of the filter by using noise in the images (salt-and-pepper noise).

Hints: - Employ a C library, e.g., OpenCV, to perform the decomposition in components and assembly the image.

- Use online tools to add noise on 4K definition images at 25%, 50%, 75% and 90% amount of noise.

b) Adapt your program solution to use a local bias averaging filter:

$$w = \frac{1}{30} \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 2 & 1 & 2 & 1 \\ \hline 1 & 1 & 2 & 1 & 1 \\ \hline 1 & 2 & 1 & 2 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

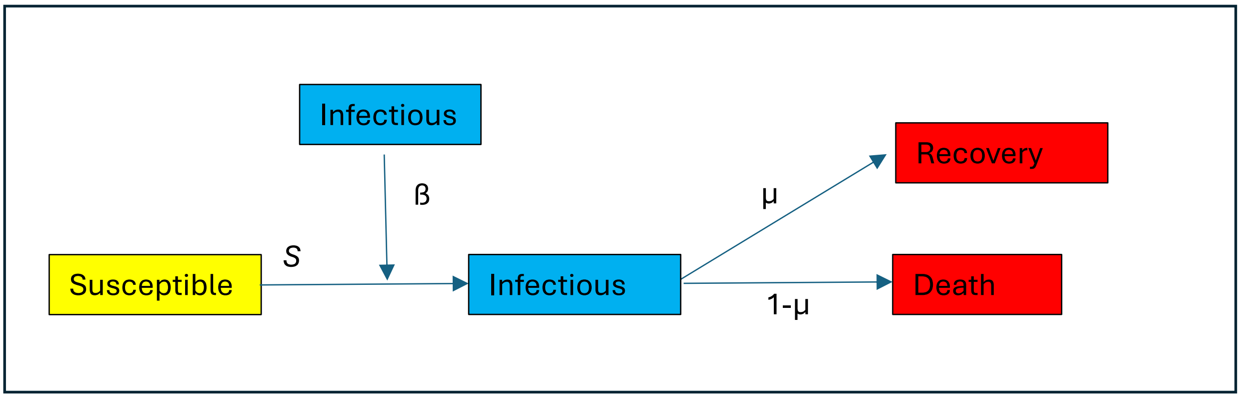
$$g_{(x,y)} = \sum_{j=-2}^{2} \sum_{i=-2}^{2} w_{(i,j)} * f_{(x+i,y+j)}$$

c) describe in CUDA a solution to apply the weighted average filter on 4K input images (3840 x 2160 pixels).

Modeling the propagation of infections in a population

Epidemiologists usually model the propagation of an infection disease resorting to parallel algorithms based on single-agent models devised to work in parallel[1]. Epidemic models describe the disease progression in different states or compartments, that every individual in the population may transit depending on their health status.

Assuming to adopt a susceptible-infected-recovered (SIR) model, where a person changes its compartment according to the interaction of this person with other contagious persons. The next figure represents the model.



The model assumes that the disease can be only spread from a person who is infected (infectious) and propagate the disease to a person who is susceptible for being infected. The disease is transmittable only if the persons are within a distance defined as infection radius (Ird). In the case, a person who is infected meets a Susceptible person, the probability of infection is given by the multiplication of the Contagious factor of the disease (ß) and the susceptibility of the person (S), in the case, the resulting value is higher than the Infection threshold (*Ith*), the person becomes infected. It is assumed also, that the disease has an incubation period *Inc*. *Inc* is the time during which the disease remains in a person, and it is assumed that on this period of time, the person becomes infectious. At the end of the infection duration, the person is recovered with a probability μ, or death with a probability 1-μ. A person who is recovered can become immune or not with a probability of 50%.

The model requires to define the following parameters:

NP: Number of persons in the model

INFP: Initial percentage of infected persons

Imm: initial percentage of immune persons in the population

S_avg: infection susceptibility of the population, it is a real number from 0-1. It is the susceptibility average of the full population.

[1] Perra N, Gonçalves B. Modeling and Predicting Human Infectious Diseases. Social Phenomena. 2015 Apr 23:59–83. doi: 10.1007/978-3-319-14011-7_4. PMCID: PMC7123706.

WxH Area of the region of study [in square meters], assume that a person normally occupies 1m X 1m area, more than one person can occupy one 1x1 square contemporary.

ND: Number of simulation days

Inc: incubation period in days

ß: Contagiousness factor of the disease

Ith: Infection threshold

Ird: Infection radius of the disease. Radius is defined as an integer value in meters.

μ: Probability to recover after being infected

It is required to model the environment area (WxH), by distributing homogeneously the persons in the model. Assumes that every person moves every day randomly in a unique direction by one meter.

At the end of the simulation, a report showing the infection facts on the population is required to be produced.

For example, perform a simulation using the following values:

NP: 1.000.000

INFP: 5%

Imm: 1%

S_avg: 0.5

WxH 10.000m X 4.500m

ND: 200

Inc: 4

ß: 0.8

Ith: 0.5

Ird: 1m

μ: 0.6

keeping similar values, perform simulations modifying the Ith value, μ, ß, and S. Which value is more critical for the population?