A Simple Model for Virus Spreading

Davide Remondina, Marco Rodino, Andrea Zanin

Project overview

In this project we have built a distributed simulation of a virus spreading in a population of individuals living in different countries. The simulation is based on the SIR model and is implemented using C++ and the MPI library.

The individuals move in a straight line with constant speed changing direction with probability 0.01 at each step and bouncing when they hit the world boundaries. The individuals can be in one of three states: susceptible, infected or recovered. An individual can get infected by being close to another infected individual for long enough; an infected individual recovers after a fixed amount of time; recovered individuals are immune for a period of time and then become susceptible again.

The simulations outputs the number of infected individuals in a set of predefined countries at the end of each simulated day. It also periodically outputs the state of all the individuals in the simulation, this data is then processed to produce a gif showing the evolution of the simulation (see file `sir.gif` in the repository).

Implementation

The algorithm we implemented subdivides the world into rectangular blocks and assigns each block to a node (the blocks do not correspond to the countries); each node is responsible for simulating the individuals in the blocks it owns. Since the individuals can move from one block to another we need to exchange data between the nodes at each step of the simulation.

At every step our algorithm redistributes the individuals among the nodes so that each node has all the individuals in the blocks it owns and the ones in the borders of those blocks, the latter are the ones for which the distance from an owned block is less than the spreading distance. The individuals in the border areas are duplicated: their data is sent to both the node that owns the block they are in and to the nodes that own the blocks they are bordering; in the nodes that don't own that individual the data is used only to check contagions with individuals that are owned by that node.

After the initialization each node follows this algorithm:

```
for each step of the simulation:
    for each owned individual:
        update the position of the individual

    for each node N:
        send all previously-owned individuals located in a block
        owned by N or its border to N

    for each owned individual:
        for each individual:
            check contagions
```

```
for each owned individual:
    update the health status of the individual

if should log statistics:
    for each country:
        count the owned susceptible/infected/recovered
        individuals in that country

        send the count to the master node

        if this is the master node:
            log the aggregate statistics
```

MPI details

The individuals are represented as a custom MPI datatype to simplify adding or removing fields from the individuals' data.

At every step each node gathers the number of individuals that it will receive from every node using MPI_Gather; then it gathers the individuals from the nodes using MPI_Gatherv; this is necessary because the number of individuals sent from one node to another at any iteration is not known in advance.
This gathering proceeds sequentially one destination node at a time.

The computation of the country statistics is done locally by each node and then the local results are aggregated using MPI_Reduce, finally the result is logged by the master node.

Performance analysis

We want to determine how changing the number of individuals (M) and countries (C) affects the execution time of the simulation; in order to do this we will consider fixed all the other parameters of the simulation (world size, block sizes, nodes count, spreading distance, ...).

The execution time is determined by the time spent doing computation and the time spent exchanging data among nodes.

Data exchange

At each step an individual is sent to a node if:

- the individual has moved to that node in that iteration
- the individual is in the border area of that node

The data exchange at each step is thus $O(M \cdot (a+b))$ where a is the probability of moving to a new node and b is the expected number of blocks such that the individual is in the block's border area.

The parameters a and b depend on the spreading distance, speed of the individuals and the size of the blocks, but not on the number of individuals or countries, so we will consider them as constants in this analysis.

Computation time

The computations performed on each node at each step are:

- checking if two individuals are close enough to infect each other, including individuals in bordering nodes: $O(((1+b)\cdot M)^2)$
- updating the position and health status of each individual: $O(M)$
- computing the nodes to which each individual must me moved: $O(M)$

Furthermore at each logging step for each country we scan all the individuals to count the number of infected individuals in that country and then log the results for each country: $O(CM + cC)$ where c is the cost of logging one country which is considered constant.

Execution time

Summing the data exchange and computation times and dropping the constant we get that the execution time for each step is $O(M^2 + M + CM + C)$.

Tests

To verify the above analysis we have run the simulation with different values of M and C and measured the execution time, the raw data is available in the `simulations.tsv` file in the repository.

We run all the tests using 10.000 steps and 4 nodes (all on the same machine) and the following are the results:

Individuals	Countries	executionTime (s)
500	0	7.5790
1000	0	16.192
1500	0	28.5
2000	0	43.317
2500	0	60.78
3000	0	81.73
3500	0	105.32
4000	0	133.77
1000	10000	20.338
1000	20000	23.003
1000	30000	25.41
1000	40000	28.506
1000	50000	32.19
1000	60000	34.851

Individuals	Countries	executionTime (s)
500	10000	11.399
500	20000	13.359
500	30000	15.977
500	40000	19.311
500	50000	21.08

Using R we fitted the following model to the data:

$$\text{executionTime} = a \cdot \text{M}^2 + b \cdot \text{M} + c \cdot \text{C} + d \cdot \text{M} \cdot \text{C} + e \cdot \text{C}^2 + f$$

All the coefficients were statistically significant (p-value < 0.005) except for e (p-value = 0.54), this is consistent with the theoretical analysis since the execution time is not expected to depend on the square of the number of countries. The R^2 of the model is 0.999 indicating that there aren't other significant factors that affect the execution time.

We conclude that the experimental data matches the theoretical analysis.