shipout/backgroundshipout/foreground

AY 2020/2021

Politecnico di Milano

Middleware Technologies
Model for Virus Spreading

Federico Armellini   Luca Pirovano   Nicolò Sonnino

Professor
Alessandro Margara

Version 1.1
June 21, 2021

Contents

1 Introduction and assignment

1.1 Description of the project

Scientists increasingly use computer simulations to study complex phenomena. In this project, you have to implement a program that simulates how a virus spreads over time in a population of individuals. The program considers N individuals that move in a rectangular area with linear motion and velocity v (each individual following a different direction). Some individuals are initially infected. If an individual remains close to (at least one) infected individual for more than 10 minutes, it becomes infected. After 10 days, an infected individual recovers and becomes immune. Immune individuals do not become infected and do not infect others. An immune individual becomes susceptible again (i.e., it can be infected) after 3 months.

The overall area is split into smaller rectangular sub-areas representing countries. The program outputs, at the end of each simulated day, the overall number of susceptible, infected, and immune individuals in each country. An individual belongs to a country if it is in that country at the end of the day.

A performance analysis of the proposed solution is appreciated (but not mandatory). In particular, we are interested in studies that evaluate (1) how the execution time changes when increasing the number of individuals and/or the number of countries in the simulation; (2) how the execution time decreases when adding more processing cores/hosts.

1.2 Assumptions and guidelines

1. The program takes in input the following parameters

    - N = number of individuals
    - I = number of individuals that are initially infected
    - W, L = width and length of the rectangular area where individuals move (in meters)
    - w, l = width and length of each country (in meters)
    - v = moving speed for an individual
    - d = maximum spreading distance (in meters): a susceptible individual that remains closer than d to at least one infected individual becomes infected
    - t = time step (in seconds): the simulation recomputes the position and status (susceptible, infected, immune) of each individual with a temporal granularity of t (simulated) seconds

2. You can make any assumptions on the behavior of individuals when they reach the boundaries of the area (for instance, they can change direction to guarantee that they remain in the area)

2 Solution Overview

2.1 Architecture chosen

MPI

2.2 Assumptions and Definitions

1. When we refer to the "main mpi process" we mean the one with rank==0
2. We assumed that the "3 month" period in order to become susceptible again is equal to 90 days
3. Velocity has as a unit of measure "how many blocks a person travels in a period of time equal to the timestep" [blocks/timestep]
4. In the source code you may find those 3 words: world, nation, subnation. The world is composed of nations, every nation is assigned to a MPI process and have subnations inside them. To make the analogy, subnations are the ones named "sub-areas representing countries" in the assignment. That has been done in order to equally distribute sub-areas to each MPI process.

2.3 General solution

2.3.1 Main

- Checking that the input is valid, if not returns and prints the error.
- The main mpi process tries to divide the "world" into countries and to decide how many countries every mpi process will receive.
- In every country, based on how many sane and infected people are given at input, the number of sane and infected people is set by the main mpi process.
- The main mpi process sends (with "mpi scatter" method) the information summariezed (number of infected and sane people and country id) of countries to every mpi process.

- All mpi process enter a for loop (one iteration for each day) where they calculate the map of infection for each day (see "infectionDayStep"), they send their result to the main mpi process with "mpi gather" method and the main mpi process prints them on console output.
- "Free" method is called to clean memory allocated all over the software

2.3.2 infectionDayStep

For every country a mpi process have to manage, for every step of the timestep, for every people inside a country:

- Check, if already infected, enough time has passed in order to heal from the virus
- Check people aroud him/her in order to understand, if they are infected, if the person we are analysing becomes infected as well (keeping in mind that there is 3 months to wait if he/her had recovered recently in order to get infected again).

2.4 Data structures

We store the people (and their information) in a list. We store the history of contacts a person had with other people in an hash table, indexed based on the id of the individual.

2.4.1 Individual

We store those information about an individual:

- Rank of the mpi process he/her belongs to
- Which sub-area (subnation) he/her belongs to
- Id
- Position (x and y) related to the subnation he/her belongs to
- If he/her is infected
- The timestamp of the last time he/her was infected
- The timestamp of the last time he/her recovered from the virus

2.4.2 Contact history

We store those information about the contact history of an individual with a specific person

- Person 1
- Person 2
- Timestamp of the first time they met
- Timestamp of the last time they met

2.5 Performance evaluation

2.5.1 Variables and parameters

- "numSubNationsPerMpiProcess": Based on how big is area of the world and the area of each country, the software assigns to each MPI process an equal amount of subnation. It is equal to circa "$(W * L/(w * l))/worldSize$"
- "NumPeopleNearIndividual": People near a certain individual, it's based on the distance needed to become infected when you cross path with an infected individual. Assuming people are equally distributed, this number is circa "$d^4/(peopleInSubnation * w * l)$"
- "HowManyTimeStepsInADay" : "$60 * 60 * 24/t$" where t is the timestamp (see assignment) expressed in seconds.
- "peopleInSubnation": we can say it's circa "$N/(numSubNationsPerMpiProcess * worldSize)$"

2.5.2 Equations

- $OverallComplexity = SetupPhaseComplexity + MainAlgorithmComplexity$
- $SetupPhaseComplexity = CalculateDistributionSubnationsToProcessComplexity + SplitPeopleComplexity + GenerateMapComplexity$
- $CalculateDistributionSubnationsToProcessComplexity = O(worldSize) + O(numSubnations)$
- $SplitPeopleComplexity = O(worldSize) + O(N)$
- $GenerateMapComplexity = numSubNationsPerMpiProcess * worldSize * ListInsertionComplexity$
- $ListInsertionComplexity = O(1)$

- $MainAlgorithmComplexity = worldSize * days * HowManyTimeStepsInADay * CalculateDayProgressComplexity$
- $CalculateDayProgressComplexity = numSubNationsPerMpiProcess * peopleInSubnation * CalculateInvidualVirusStateComplexity$
- $CalculateInvidualVirusStateComplexity = FindRectanguralsNearIndividualComplexity + FindInvidualsInNearRectangles + NumPeopleNearIndividual * FindHistotyBetweetIndividualsComplexity$
- $FindInvidualsInNearRectangles = O(peopleInSubnation)$
- $FindRectanguralsNearIndividualComplexity = distanceToBeInfected^2 * ListInsertionComplexity$
- $FindHistotyBetweetIndividualsComplexity = O(1)$

2.5.3 Conclusion

So the overall complexity is, summarized with every step:

- $O(worldSize) + O(numSubnations) + O(worldSize) + O(N) + numSubNationsPerMpiProcess * worldSize * O(1) + worldSize * days * HowManyTimeStepsInADay * numSubNationsPerMpiProcess * peopleInSubnation * (d^2 * O(1) + O(peopleInSubnation) + NumPeopleNearIndividual * O(1))$
- $worldSize * days * 60 * 60 * 24/t * (W * L/(w * l))/worldSize * N/((W * L/(w * l))/worldSize * worldSize) * (d^2 * O(1) + O(TotalPeople/((W * L/(w * l))/worldSize * worldSize)) + d^4/(N/((W * L/(w * l))/worldSize * worldSize) * w * l) * O(1))$
- $days * 60 * 60 * 24/t * (W * L/(w * l)) * N/((W * L/(w * l))) * (d^2 * O(1) + O(TotalPeople/(W * L/(w * l))) + d^4/(TotalPeople/((W * L/(w * l))) * w * l) * O(1))$
- $O(days * N^2/(W * L) * d^4 * (w * l)^2)$