Instructor Guide

Two scientific applications are demonstrated to show the problem of why parallelization is eventually needed for large problems.

The codes are included and ready to compile/run. Encourage students to run both of these and to try varying the command-line arguments.

Drug Design

To compile: g++ -o dd_serial dd_serial.cpp
To run: ./dd_serial

Optional command line arguments (in this order):

1) maximum length of the randomly generated ligand strings
2) number of ligands generated
3) protein string to which ligands will be compared

Try a combination of small and large values to show that as the problem grows, the computational needs grow very quickly.

Pandemic

To compile: make clean then make in the root directory
To run: ./pandemic_serial

Optional command line arguments:

-n the number of people in the model
-i the number of initially infected people
-w the width of the environment
-h the height of the environment
-t the number of time days in the model
-T the duration of the disease (in days)
-c the contagiousness factor of the disease
-d the infection radius of the disease
-D the deadliness factor of the disease

-m the number of microseconds in between days in the model (used to speed up or slow down the animation)

Try a combination of small and large values to show that as the problem grows, the computational needs grow very quickly.

Common Pitfalls for Students and Instructors

This may be the first time students have run scientific programs before, so they may need assistance with the code compilation (detailed instructions are in the Instructor Guide and in the slides).

X11 is required for visualizing the pandemic example, although this can be disabled in the code if visualization is not prefered.

Depending on the parameters given for either problem, students may try to run the code and think that the program is frozen. Usually, it just means that the computational needs are extreme, so it may take a long time to run.