Epidemic simulator for the Dengue disease

Konstantinos Theodorakos

University of Antwerp

Konstantinos.Theodorakos@student.uantwerpen.be

Abstract

Dengue is a mosquito-borne tropical disease. In some cases, this disease may develop into the life-threatening hemorrhagic fever which results in low levels of blood platelets, blood plasma leakage, or even into dengue shock syndrome [3]. The main goal of this project is to construct an epidemic model & simulator that can produce realistic & accurate outbreak predictions for the Dengue disease. The input parameters will be combined with environmental and meteorological modelling of aspects relevant to disease spreading. The model will be validated with real-life data of observed disease outbreaks and it will be provided as a proof-of-concept tool implementation.

Keywords: Dengue, Epidemic Modelling, Stochastic Cellular Automata, Spatial Model, Parallelization, GPGPU

1. Problem

Dengue disease infects around 50 to 528 million people worldwide per year and it is currently present in more than 100 countries. This leads into half a million hospitalizations and around 20,000 deaths. For the 2000s, 12 countries in Southeast Asia have about 6,000 deaths and 3 million infections annually. Dengue is one of the most common vector-borne disease in a world-wide scale.

2. Proposed Solution

2.1. Phases

The main goal of this project is to construct an epidemic model & simulator that can produce realistic & accurate outbreak predictions for the

Dengue disease. The input parameters will be combined with environmental and meteorological modelling of aspects relevant to disease spreading. The model will be validated with real-life data of observed disease outbreaks and it will be provided as a proof-of-concept tool implementation.

The first phase of this project has already been completed (Figure 2). The initial version of the mathematical model has already been constructed, as part of my studies (MSc of Computer Science: Software Engineering at the University of Antwerp, Belgium). It was the main topic of my thesis, "Modelling of Stochastic Compartmental Spatio - Temporal Epidemic Simulations with Cellular Automata and acceleration with CPU and GPGPU parallelism" (see `http://epidemicsimulator.com/`), which has been successfully defended on June 2016.

The second phase is to validate the model with real-life data of observed disease outbreaks. More precisely, we will iteratively improve/extend the model and check against real-life control data until acceptable accuracy is achieved.

2.2. Approach

Population health modelling via computer simulations has been used for decades by epidemiologists and health researchers. It assists in studying population-level health phenomena. However, due to the inherent complexity, applied simulations models are not usually readily available. In addition, existing epidemic models do not adequately account for rainfall, climate variables or relevant socioeconomic factors [4].

How could this project could prevent and combat a potential outbreak?

The project will focus on:

- Computer implementation of proven epidemiological models, combined with environmental and meteorological modelling of aspects relevant to disease spreading.
- Validation of the models/simulator using real life disease outbreak and environmental/meteorological data.

- Provision of “what-if” simulation scenarios to assist in developing intervention strategies and assessing the impact of the disease.

- Investigation on which factors impact most on the speed and extension of the spread of the disease.

2.3. Tasks

1. Gathering real epidemic case data.
2. Interpretation & transformation of real data into a “comparable” format (geo-referenced environmental, climate and epidemic case time series).
3. Determination of the model calibration & validation methodology.
4. Parameter sweeps using the epidemic simulator & the real data.
5. Comparison of the predicted results versus the expected results.
6. Model calibration with the “best fit” parameters.
7. Benchmarking and optimisation of the resulting model.
8. Finalisation and publication of all the deliverables.

2.4. Timeline

Timeline:

- Model costruction phase (September 2016 - February 2017): Tasks 1-5.

- Model calibration phase (February 2017 - June 2017): Tasks 4-8.

2.4.1. Sample results

Figure 1 shows a sample graph on model execution times. There is an execution time comparison between serial, CPU parallel and GPU parallel versions of the same source code.The algorithm is *stochastic*, so it depends heavily on the random number generation performance.

Speedups are reported from 2x-4x with CPU parallelism and 13x-94x with GPU Parallelism. We started from execution times around 2-13 minutes and ended up with 2-9 seconds. Experiments were run on CPU: 3.0 GHz AMD Phenom II X4 940, 4 cores, L2: 4x512KB, L3: 6MB, 4 physical cores, Memory: 8 GB DDR2 800MHz 64bit, GPU: GTX 970 4GB, Storage: Corsair SSD F60 GB2, OS: Windows 10 x64 Version 1511 OS Build 10586.318.

3. Acknowledgments

I would like to personally thank the following people, who assisted with advice, discussions and suggestions: Prof. Dr. Jan Broeckhove and Dr. Lander Willem.

The results and the information provided on this document were part of the thesis that was submitted for the fulfillment of the Master of Computer Science degree in Software Engineering. Supervisors: Prof.Dr. Jan Broeckhove, Dr. Lander Willem. M.Sc. committee: Prof.Dr. Hans Vangheluwe. Faculty of Science, Department of Mathematics and Computer Science University of Antwerp, June 2016.

Links & References

[1] Project official web page - *“Modelling of Stochastic Compartmental Spatio - Temporal Epidemic Simulations with Cellular Automata and acceleration with CPU and GPGPU parallelism”* - `http://epidemicsimulator.com/`

[2] Project Presentation - *“Modelling of Stochastic Compartmental Spatio - Temporal Epidemic Simulations with Cellular Automata and acceleration with CPU and GPGPU parallelism”* `https://1drv.ms/b/s!AgM7aH_rFcPzgaIIW0fCFLk8QRFkOA`

[3] Wikipedia: *Dengue Fever* - `https://en.wikipedia.org/wiki/Dengue_fever`

[4] *World Health Organization: Comprehensive Guidelines for Prevention and Control of Dengue and Dengue Haemorrhagic Fever.* Page 15 - `http://www.searo.who.int/entity/vector_borne_tropical_diseases/documents/SEAROTPS60/en/`

[5] Sitmo. *High Quality C++ Parallel Random Number Generator* - `http://www.sitmo.com/article/parallel-random-number-generator-in-c/`

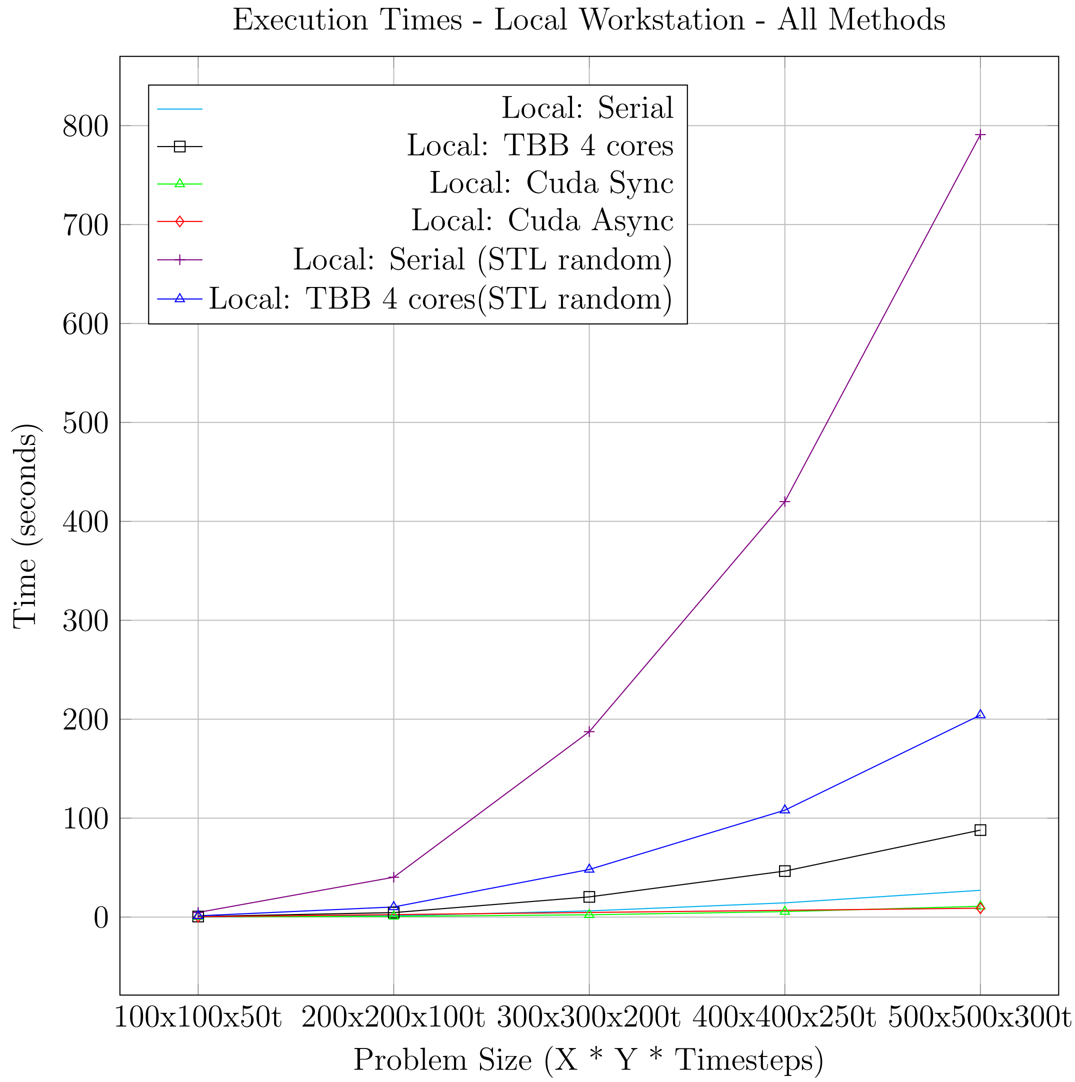

Figure 1: Execution time comparison of all the available methods.The algorithm is *stochastic*, so it depends heavily on the random number generation performance. Using the Sitmo PRNG[5] library in multiple platforms, speedups are reported from 2x-4x with CPU parallelism and 13x-94x with GPU Parallelism. We started from execution times around 2-13 minutes and ended up with 2-9 seconds. Experiments were run on CPU: 3.0 GHz AMD Phenom II X4 940, 4 cores, L2: 4x512KB, L3: 6MB, 4 physical cores, Memory: 8 GB DDR2 800MHz 64bit, GPU: GTX 970 4GB, Storage: Corsair SSD F60 GB2, OS: Windows 10 x64 Version 1511 OS Build 10586.318.

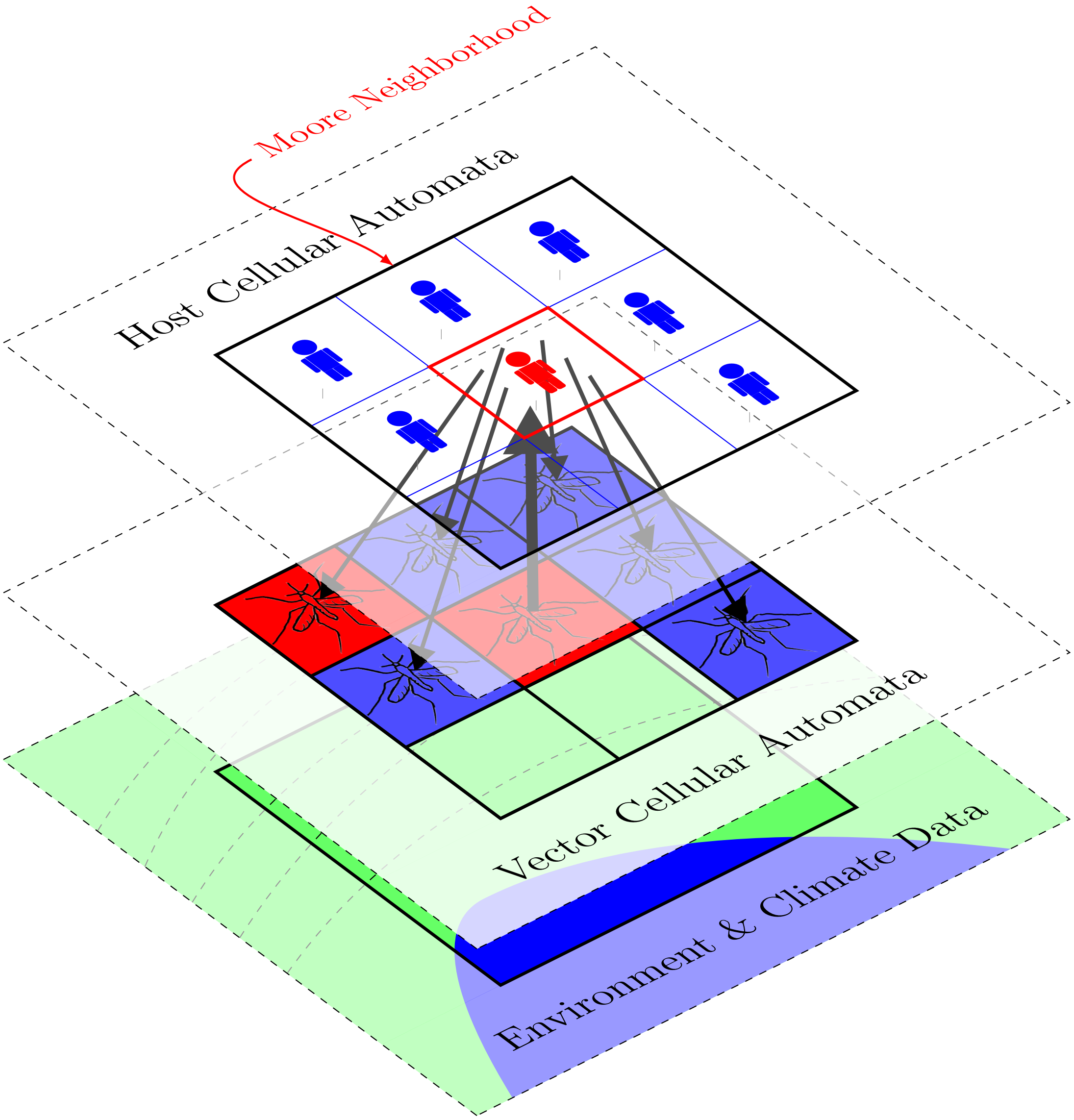

Figure 2: This figure shows how the CA layers for the Dengue Simulator interact. First, all the Infected Vectors(Mosquitoes) in Cellular Automata CA_V, within the current cell i_V, interact with all the Hosts(Humans) in the cell i_H of the same geographical region. Next, a fraction ϕ_h of Infected humans within the cell's i_H *Moore neighborhood* of the CA_H, interact with each cell of the CA_V contained within a $MooreNeighborhood$ of radius $r = 1$. Climate and Environmental spatial layers of data affect the rates of the Dengue Disease transmission through a normalized fraction.