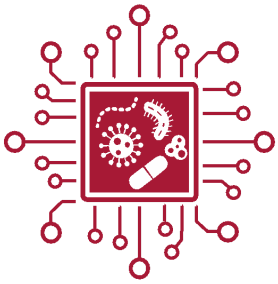
Simulation Library Optimization and Parallelization of Stochastic Simulation Code for Computational Epidemiology

Parallelizing the StochPy Library

Resistance Epidemiology Modeling Initiative
Dr. Eric Lofgren



WASHINGTON STATE UNIVERSITY
Resistance Epidemiology Modeling Initiative

REMI-HPCStochPy

Michael J McNaughton
Ryan Charit
Gerald Hoff

September 20, 2022

I. Introduction

The Resistance Epidemiology Modeling Initiative (REMI) conducts simulations modeling epidemics, particularly in smaller populations and focused on the beginning and end of epidemics. Such models can help with preventing and containing present and future epidemics and as such are of great utility. Most of the major tools for disease modeling in the field are deterministic, which function well for large population sizes. However, since the REMI lab focuses most of their research around smaller populations, a stochastic modeling tool is preferred, as stochastic models are superior for use in small populations. Unfortunately, the tools for such modeling methods are not as highly developed as would be ideal for a research field as important as epidemiology.

As such, our project problem is to create an optimized and parallelized fork of the StochPy library. This library is used by REMI to conduct stochastic simulations in computational chemistry, physics, epidemiology, and ecology. The shortcoming of StochPy is that the stochastic simulation algorithm is slow, especially when trying to perform large computations with upwards of 10 million groups of simulations. However, this algorithm is also embarrassingly parallel, which means that if the codebase were refactored to run the simulations on many cores at once, the amount of time for a job to complete would be drastically reduced.

II. Background and Related Work

Stochastic simulations are generally performed using a process called the Gilespie direct method, making use of an algorithm popularized by Daniel Gilespie. The algorithm converts the rates at which certain events take place to probabilities that that event will take place. The simulations use the probabilities generated by the algorithm to determine which of a set of potential events take place by drawing from a bag of weighted probabilities for each timestep. In the case of epidemiological simulations, events can include a new individual being infected or an infected individual recovering. An increase in the population size in such simulations also increases the frequency of events occurring and thus increases the computational intensity of the simulations which drastically reduces the speed at which simulations for higher population sizes can be completed.

The current version of StochPy is state-of-the-art at the time of the creation of this report. That is, there currently are no available tools or libraries known to the REMI lab or its associates that outperform the StochPy library for their purposes. If our fork of the library were to demonstrate significant speedup, it would become an incredibly useful asset to the lab and any other labs that work with large-scale stochastic simulation for epidemiological research.

Knowledge and skills we will need to learn in order to complete this project include developing a basic understanding of the functionality of the StochPy library in its current state, as well as Python tools and libraries for increasing the speed of programs through the use of just-in-time (JIT) compilation and parallelization, such as by using the Numba and Dask libraries.

There are libraries that can do stochastic simulations similar to StochPy in parallel, Cayenne being one of them. Professor Lofgren hasn't yet explained what differentiates these libraries from the StochPy library.

III. Project Overview

The Resistance Epidemiology Modeling Initiative is a product of collaboration between the WSU Paul G. Allen School for Global Health and the WSU School of Electrical Engineering and Computer Science. REMI aims to apply advanced computational methods to problems in public health, such as antimicrobial resistance, healthcare-associated infections, and COVID-19. In order to pursue its goals, one of REMI's most frequent activities is running simulations of epidemics centered around small, localized communities of people, such as schools and hospitals. This contrasts with other, more traditional epidemic simulations that focus on large, dense population centers like New York City.

This focus on smaller population centers necessitates the use of stochastic simulation models, as opposed to deterministic models for larger population centers. Deterministic models make use of fractional measurements, handle variance poorly, and are incapable of allowing a simulated disease to go extinct. This is problematic, as nonsensical measurements like one-quarters of an infected nurse in a hospital ward, inability to properly simulate the chances of a disease with a relatively low R_0 value spreading into a small scale outbreak, and inability for a simulated disease to die out are all detrimental when determining if the simulated disease grows into an epidemic or perishes before becoming an issue. However, as population sizes increase, stochastic simulations become more computationally intensive as the number of timesteps between events gets smaller, so running the many simulations necessary for the research with populations of moderate size sequentially even on HPC clusters can take several days.

Indeed, one of REMI's main limitations in conducting its stochastic epidemic simulations is the slow speed at which they are executed, which becomes a glaring issue when considering that upwards of tens of millions of simulations may need to be run for a single research experiment. This is a direct consequence of StochPy, the stochastic simulation Python library in use by REMI, being designed for single-threaded execution. Moreover, StochPy's authors and users have made no public attempt to optimize the library, and other stochastic simulation libraries may not meet REMI's needs or smoothly interface with existing lab codebases and digital infrastructure.

Slow simulation speeds pose a significant issue for REMI. Given that REMI is involved with research surrounding antimicrobial resistance, COVID-19, and other important public health subfields, improving the speed at which simulations are conducted is critical for accelerating potentially lifesaving research, aiding in rapid response to potential outbreaks, and assisting with formulating public policy decision recommendations. Furthermore, sufficiently improving StochPy performance would enable REMI researchers to feasibly run simulations on local devices like laptops and desktop computers, reducing dependency on the Kamiak cluster and allowing for research to be conducted more efficiently and conveniently, which would further increase research output.

The bare minimum intended outcome for the project is to have StochPy demonstrate a fair amount of speedup. This can be accomplished by optimizing the codebase. We can also create a large amount of speedup comparatively easily by running the disparate simulations on individual processors. StochPy, being a stochastic simulation model, calculates the interactions of individual elements in the simulation. This makes it slower than large scale simulation algorithms, but more accurate at smaller scales. These interactions between elements of the simulation are oftentimes isolated from other interactions between other elements, meaning that these computations are done separately from other groups of computations. These groups are essentially sub-simulations of the larger simulation. These sub-simulations can be done on different cores concurrently, allowing for significant speedup. This is the embarrassingly parallel

nature of the algorithm. This could be completed using threads, and it likely would not be that hard to implement. This is the minimum work for us to do.

The parallelization of sub-simulations onto standard non-GPU cores would be the easiest and likely the most effective way to create speedup. However, doing the sub-simulations on GPUs themselves could be promising for creating further speedup. Whether or not having the sub-simulations running on GPUs would be feasible or not is dependent on the architecture of the GPUs, the nature of the algorithm, and the limitations of Python.

Another area where we could see some amount of speedup is with JIT compilation. Using CPython could be beneficial. Likely JIT compilation or another form of compilation would be the only available way to work within the limitations of Python's speed. As rewriting the entire codebase in a language such as Julia or Chapel is unfeasible.

IV. Client and Stakeholder Identification and Preferences

The primary stakeholder of this project is REMI, which is represented by Dr. Eric Lofgren. At a minimum, REMI's primary need is a speed-up in the stochastic simulation functionality of the StochPy library. This can be achieved through the optimization of the codebase to make use of JIT compilation as well as parallelization of the StochPy library, specifically in regards to its stochastic simulation feature. This parallelized fork should still be capable of operating on existing lab machines and the Kamiak cluster. Stretch goals include adding the ability to employ GPUs to improve simulation speeds, such as through NVIDIA's CUDA framework, as well as the removal of unwanted popup messages accompanying the startup of StochPy.

Student members of the Lofgren Lab, along with other students working under the auspices of REMI, serve as a secondary stakeholder. For them, improving the speed of stochastic simulations is a critical need. An additional preference is improving the speed of StochPy to the point that stochastic simulations run in reasonable timeframes on portable devices like laptops, as opposed to just the Kamiak cluster.

Additionally, government institutions funding REMI's research, such as the National Institutes of Health (NIH), also act as secondary stakeholders for this project. Given the significance of REMI's research, speeding up StochPy would help fulfill NIH mission objectives. Also, it is probable that improving REMI's research capabilities would make it more competitive for future government grant funding.

Other labs and researchers using StochPy serve as indirect stakeholders for our project. Though our team is not in contact with other researchers using StochPy, producing a parallelized fork of the library stands to benefit their research activities as well. Communicating with other research teams may give input as to possible functionalities that a parallel version of StochPy could have. Of course, REMI's goals and requirements are paramount, considering that they are the sponsors of this project.

Finally, those populations which benefit from the research conducted by REMI and other labs using StochPy are tertiary stakeholders, as improvements in the speed of research at such labs could result in lifesaving policy changes and developments. Such populations could include the global community at large in the case of containing and preventing epidemics in their early stages. More specifically, the staff and patients of hospital wards, staff and students of schools, as well as staff and detainees of jails stand to benefit from REMI conducting faster simulations.

V. Glossary

CUDA - Proprietary parallel programming platform created by NVIDIA.

Graphics processing unit (GPU) - Specialized processing units that can do many small numerical operations at once. They are primarily useful for graphics processing, but can be used in some parallel programming applications.

High-performance computing (HPC) - Encompasses the subfield of computing concerned with performing advanced calculations at extremely fast speeds, especially on computer clusters.

Kamiak cluster - A high-performance computing cluster owned and operated by WSU. Used by REMI to conduct stochastic simulations and other resource-intensive computational activities.

Lofgren Lab - A WSU lab under the leadership of Dr. Eric Lofgren focused on epidemiology and public health.

Parallelization - A programming paradigm where a task is subdivided across processing units so that its subtasks can be run concurrently.

R_0 - For a communicable disease, represents how many people an infected person is expected to infect.

Resistance Epidemiology Modeling Initiative (REMI) - A research initiative established at WSU. Seeks to apply advanced computational methods to problems in public health, such as antimicrobial resistance, healthcare-associated infections, and COVID-19.

Stochastic simulation - A simulation of a system where individual variables change randomly with individual properties.

Gillespie Algorithm - A stochastic simulation algorithm. Used in the StochPy library.

Just-in-time compilation (JIT) - A compilation method that compiles the code exactly at runtime. A combination of ahead-of-time compilation and interpretation.

VI. References

Dileep Kishore, “cayenne : Python package for stochastic simulations.” Stochastic Simulation Algorithms in Python. https://cayenne.readthedocs.io/en/latest/#cayenne-python-package-for-stochastic-simulations (9/21/2022)