

# Reusable software for high performance computing

October 15 2018, by Julie Stewart

---



Sunita Chandrasekaran, an assistant professor of computer and information sciences, is designing frameworks to adapt code to increasingly powerful computer systems. She is working with complex patterns known as wavefronts, which are pictured in the background of this image. Credit: Joy Smoker

The world's fastest supercomputer can now perform 200,000 trillion calculations per second, and several companies and government agencies around the world are competing to build a machine that will have the computer power to simulate networks on the scale of the human brain.

This extremely powerful hardware requires extremely powerful software, so existing software code must be continually updated to keep up.

Sunita Chandrasekaran, an assistant professor of computer and information sciences at the University of Delaware, is perfectly suited for this challenge. Under a new grant from the National Science Foundation, she is designing frameworks to adapt [code](#) to increasingly powerful systems. She is working with complex patterns known as wavefronts, which are commonly found in scientific codes used in analyzing the flow of neutrons in a [nuclear reactor](#), extracting patterns from biomedical data or predicting atmospheric patterns.

Chandrasekaran is an expert on parallel programming—writing [software code](#) that can run simultaneously on many multi-core processors. Parallel programming is an increasingly important discipline within computer science as more and more universities and companies use powerful supercomputers to analyze wide swaths of data, from scientific results to consumer behavior insights and more.

Chandrasekaran is looking at scientific applications to see how they were written, how they have been performing on outdated architectures, what kind of programming models have been used, and what challenges have arisen.

"Most of the time the programming models are created in a broad stroke," she said. "Because they are generalized to address a large pool of commonly found parallel patterns, often the models miss creating features for some complex parallel patterns, such as wavefronts, that are hidden in some scientific applications."

A wavefront allows for the analysis of patterns in fewer steps. The question is: How do you get the [programming model](#) to do that?

One such example is Minisweep, a miniapp that models scenarios within a nuclear reactor by "sweeping" across a grid with squares that represent points in space and are used to calculate the positions, energies, and flows of neutrons. This parent application to Minisweep is used to reduce the odds of a meltdown and to safeguard engineers who work around the nuclear reactor from radiation exposure. Earlier this year, Chandrasekaran and doctoral student Robert Searles demonstrated how they modified the miniapp to perform 85.06 times faster than code that was not parallelized. This work was recently presented in the premier Platform for Advanced Scientific Computing (PASC) 2018 conference and published by the Association for Computing Machinery (ACM).

"We wondered: Is this [pattern](#) specific to Minisweep?," she said. "Or is it going to exist in other codes? Are there other codes that could benefit if I were to put this type of pattern in a programming model and create an implementation and evaluate it?"

For example, Chandrasekaran discovered that some algorithms in bioinformatics, the study of large sets of biological data, contained similar patterns. She suspects that by adapting the software written for Minisweep, she can make great strides toward improving the code. She will try this with data from Erez Lieberman Aiden, assistant professor of molecular and human genetics at Baylor College of Medicine and assistant professor of computer science at Rice University.

Chandrasekaran met Aiden when he visited UD to give a talk titled "Parallel Processing of the Genomes, by the Genomes and for the Genomes."

Chandrasekaran was inspired by Aiden's work with DNA sequences. He uses a computing tool to find long-range interactions between any two elements on the same chromosome, in turn showing the genetic basis of diseases. Chandrasekaran suspected that she could utilize existing patterns and update the code, allowing for faster analysis of this

important biological data.

"The goal is not to simply create a software tool," she said. "The goal is to build real-life case studies where what I create will matter in terms of making science easy."

Directive-based parallel programming models such as OpenACC and OpenMP will be explored to do this.

Chandrasekaran aims to maintain performance and portability as she redesigns algorithms. She will also keep the scientists who use the algorithms in mind.

"You can't create a programming [model](#) by only looking at the application or only looking at the architecture," she said. "There has to be some balance."

This project will benefit scientific application developers who are not necessarily computer scientists. "They can concentrate more on the science and less on the software," said Chandrasekaran. Scientists come to her with data sets and problems that take hours, days, sometimes months to compute, and she figures out how to make them run faster, thus enabling newer science.

Chandrasekaran will analyze data supplied by Aiden at Baylor and physicists at Oak Ridge National Lab. Searles will also work on the project, and Chandrasekaran is looking for an additional graduate student with an aptitude for [parallel programming](#) to help with this project.

**More information:** Robert Searles et al. Abstractions and Directives for Adapting Wavefront Algorithms to Future Architectures, *Proceedings of the Platform for Advanced Scientific Computing*

*Conference on - PASC '18* (2018). [DOI: 10.1145/3218176.3218228](https://doi.org/10.1145/3218176.3218228)

Provided by University of Delaware

Citation: Reusable software for high performance computing (2018, October 15) retrieved 19 April 2024 from <https://phys.org/news/2018-10-reusable-software-high.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.