



# Nathan Sobotka

651-756-9597 | [nathan.a.sobotka@gmail.com](mailto:nathan.a.sobotka@gmail.com) | [github.com/nsobotka](https://github.com/nsobotka)

## EDUCATION

<b>Stanford</b> PhD Student   Computer Science	<b>Stanford, CA</b> 2029
<b>University of Pennsylvania</b> Master of Science in Engineering   Computer and Information Science   GPA: 4.00/4.00 Bachelor of Science in Engineering   Computer Science, Minor: Mathematics   GPA: 3.98/4.00	<b>Philadelphia, PA</b> May 2024 May 2024

## WORK

<b>Reconfigurable Dataflow Architecture Research</b> <i>PhD   Professor Kunle Olukotun</i>	<b>Stanford, CA</b> September 2024 – December 2024
<ul style="list-style-type: none"><li>Developing a hardware simulator for reconfigurable dataflow architectures, focusing on creating a dynamic memory system capable of handling dynamism commonly seen in mixture of expert models.</li><li>Implemented and tested the system in a hardware description language (BlueSpec), compiling down from STeP.</li></ul>	
<b>Taint Primitive Tracking (TPT)</b> <i>PhD Rotation   Professor Caroline Trippel</i>	<b>Stanford, CA</b> September 2024 – December 2024
<ul style="list-style-type: none"><li>Optimized hardware/software co-designed Spectre defense by developing a lightweight hardware extension to efficiently and accurately track protection of data in memory. Simulated in Gem5 using the O3 CPU</li></ul>	
<b>Robust Profile Guided Runtime Prefetch Generation (RPG2)</b> <i>Research Assistant   Professor Joe Devietti</i>	<b>Philadelphia, PA</b> May 2023 – August 2023
<ul style="list-style-type: none"><li>Achieved up to 2.15x speedup on C/C++ binaries by dynamic insertion of cache prefetch instructions followed by systematic tuning of prefetch distances, under the guidance of Professor Joe Devietti</li><li>Evaluated RPG2 on hundreds of benchmarks, scrutinizing speedup, MPKI, IPC, and comparing with APT-GET. Advanced prior work by retaining speedup and eliminating slowdown by dynamically disabling prefetching</li></ul>	
<b>NASA Langley Research Center   Safety Critical Avionics Systems Branch</b> <i>Combinatorial and Property Based Testing Intern</i>	<b>Hampton, VA</b> May 2023 – August 2023
<ul style="list-style-type: none"><li>Developed a Haskell library to enhance NASA's testing technology by combining static analysis with property based testing, identifying program inputs that are unlikely to be found with random or enumeration based testing</li><li>Implemented software to address limitations found after reviewing multiple academic papers on testing techniques in functional programming languages. This included failings of symbolic execution and randomized testing</li></ul>	
<b>DeepSpec NSF Expedition: The Vellvm Project</b>  <i>Research Assistant   Professor Zdancewic</i>	<b>Philadelphia, PA</b> May 2022 – January 2023
<ul style="list-style-type: none"><li>Developed a Coq monad library for public use by defining equivalence for the error, option, list, set, multiset, CPS, ID, and state monad. Proved fundamental theorems true, including the monad laws, for ease of future use</li><li>Tested VELLVM's memory model using unit tests written in LLVM and C and automated tests written in QuickCheck. Proved LLVM compiler optimizations correct in Coq with Professor Zdancewic (GitHub: Vellvm )</li></ul>	
<b>University of Pennsylvania Computer and Information Science Department</b> <i>TA (CIS-5710, CIS-2400) &amp; Peer Tutor (MATH-3120, CIS-1600)</i>	<b>Philadelphia, PA</b> Jan 2022 – May 2024
<ul style="list-style-type: none"><li>Taught 400 undergraduate and graduate students Introductions to Computer Systems and Computer Organization and Design, covering all facets of computer architecture.</li></ul>	

## PUBLICATIONS

<b>Robust Profile Guided Runtime Prefetch Generation</b>	April 2024
<ul style="list-style-type: none"><li>Yuxuan Zhang, Nathan Sobotka, Soyeon Park, Saba Jamilan, Tanvir Khan, Baris Kasikci, Gilles A Pokam, Heiner Litz, Joseph Devietti. Robust Profile Guided Runtime Prefetch Generation. In 2024 ASPLOS.</li></ul>	

## PROJECTS

<b>Search Engine</b>   <i>Java</i>	March 2023 – May 2023
<i>CIS-5550: Internet and Web Systems Project</i> <ul style="list-style-type: none"><li>Employed Amazon AWS to host a highly distributed search engine with over 125-thousand results, utilizing a hand-built crawler to find pages, an indexer to compute pagerank / tf / idf values, and a frontend to display findings</li></ul>	
<b>Operating System</b>   <i>C</i>	October 2022 – December 2022
<i>CIS-3800: Operating Systems Project</i> <ul style="list-style-type: none"><li>Developed a UNIX-like operating system, complete with three level priority scheduling and a FAT file system</li><li>Simulated using user threads to emulate OS context switches, along with a custom shell to interact with the OS</li><li>Tested using a logging system to verify features like background processes, blocking, waiting, and CPU utilization</li></ul>	

## TECHNICAL SKILLS & INTERESTS

<b>Computer</b>	Rust, BlueSpec, C, C++, Python, Java, Coq, Haskell, Verilog, OCaml, SQL (mySQL, Oracle)
-----------------	---