Kelly Shiptoski

Email: kship at seas upenn edu
Github: https://github.com/krs85
Website: https://krs85.github.io/

Research Interests

- Operating systems
- Parallelism and concurrency
- Determinism and reproducibility
- Distributed systems
- System design and development

Skills

- **Languages:** Rust, C++, C
- **Linux Systems Programming**

Education

2017 – Present	Ph.D., Computer Science, **University of Pennsylvania**
	Advised by Dr. Joseph Devietti
2017 – 2019	M.S.E., Computer Science, **University of Pennsylvania**
	Graduated cum laude
2012 – 2017	B.S., Computer Science, B.A., Mathematics
	Drexel University
	Graduated cum laude

Publications

Reproducible Containers, Omar S. Navarro Leija, **Kelly Shiptoski**, Ryan Scott, Baojun Wang, Nicholas Renner, Ryan Newton, and Joseph Devietti. International Conference on Architectural Support for Programming Languages and Operating Systems *(ASPLOS '20)*, March 2020.

Industry Research Experience

Research Intern, VMWare Research Group, Summer 2020.

Research Projects

Process Cache:

- A system for providing automatic caching of computation at the process level *(WIP)*.
- Facilitates elimination of redundant computation from arbitrary Linux process trees.
- Written in Rust, utilizing asynchronous futures and ptrace.
- Leading design and implementation of the project.

Distributed Differential Datalog (D3log) - VMWare Research Group:

- An extension of the Differential Datalog language which provides automatic distribution of Differential Datalog computations across compute nodes.
- Datalog is specifically designed for incremental computation, but only operates upon one node, while D3log is designed for distributed systems.
- Contributed to the distributed runtime (written in Rust) by adapting the distributed API to allow for incremental on-the-fly reconfiguration of the nodes within the network, expanding the fault tolerance guarantees of the runtime.

Reproducible Containers (DetTrace):

- A container abstraction for Linux which guarantees both determinism and reproducibility for any unmodified Linux program run through it. Written in C++ and utilizes ptrace.
- Extended the scheduler from serialized execution to parallelization of system-call-free regions of execution, reducing the overhead of compute-bound workflows to under 2%.

Teaching

- **Graduate Teaching Assistant** for Computer Architecture (CIS 501), University of Pennsylvania, Spring 2019.
- **Teaching Assistant and Recitation Leader** for Intro to Computer Science (CIS 110), University of Pennsylvania, Spring 2018.

Other Experience

- **Computer Science Instructor** for Penn GEMS (Girls in Engineering, Math, and Science) Camp, University of Pennsylvania, June 2018.
- **Software Engineering Intern** for Thomson Reuters, Summer 2016.
- **Software Engineering Intern** for Bentley Systems, Inc., Summer 2015.
- **Software Engineering Intern** for Independence Blue Cross, Summer 2014.