

---

# NOAH GOLDSTEIN

SOFTWARE ENGINEER

[goldstein.w.n@gmail.com](mailto:goldstein.w.n@gmail.com)  
(412)-401-5362

---

## Profile

I am a computer programmer with a primary interest and competency in low level optimization using C, C++ and x86\_64 Assembly and high performance parallel algorithms and data-structures.

---

## Education

### Washington University in St. Louis

Major: Computer Science — 3.7 GPA 2016-2020  
TA: Computer Systems (Head TA), Operating Systems, Parallel Programming, Object Oriented Programming.

### Carnegie Mellon, School of Computer Science

Intro to Computer Systems 15-213 — 4.0 GPA June 2018-Aug 2018  
Summer course.

### University of Illinois at Urbana-Champaign

Advisor: Vikram Adve Aug 2020-Present  
Pursuing a PhD in Computer Science.

---

## Awards

### Deutsche Bank "Powering Progress" Case Competition Finalist

A case competition to devise the most effective plan for aiding a non-profit group in New York City which assisted LGBTQ youth. My team was selected as finalists and flown to New York to give our final pitch in person.

### Washington University: Outstanding Senior

One of 10 students selected for their contributions to the department.

---

## Research

### Author: Responsive Parallelism with Futures and State

Wrote three benchmarks; A web proxy, email server, and job scheduler in both standard cilk and a newly developed priority scheduling cilk my team was working on. The goal of this paper was to show that with equally complex language semantics, measured both in lines of code and compile time, the priority scheduler could 1) reduce the latency on high priority tasks and 2) increase the throughput of the entire application. The paper was published in PLDI '20.

### Author: Priority Scheduler for Interactive Applications

Performance engineered the three benchmarks from the previous paper for both standard cilk and priority cilk. The three applications were optimized for minimizing IO latency over a TCP connection. As hoped, the benchmarks using the new priority schedulers showed significant lower latency for high priority tasks (sometimes as great as an order of magnitude) as well as higher throughput for the entire application. The paper was published in SPAA '20.

---

## Opensource

### Glibc: x86\_64 Hand Optimized Assembly String and Memory Functions

Primarily [contributions](#) are to the avx2 and evex implementations of key functions including memcpy, memset, memcmp, strlen, memchr, strchr. In addition smaller changes and bug fixes include a wider variety of files and function. Performance improvements range from -5% to +50% (generally around 10-20%)

### Linux: Small Patch to Add New Feature to Liburing

Small [contribution](#) to add feaute allowing file descriptors to be skipped during IORING\_REGISTER\_FILES\_UPDATE (bulk update).

---

## Experience

### Amazon Inc.

Software Engineer

June 2020-July 2021

### Q-Net Software

Intern

May 2019-Aug 2019