

# YASH LALA

[yashlala.com](http://yashlala.com) ◇ [github.com/yashlala](https://github.com/yashlala) ◇ [linkedin.com/in/yashlala](https://linkedin.com/in/yashlala)

(510)-400-5572 ◇ [yash.lala@yale.edu](mailto:yash.lala@yale.edu) ◇ Palo Alto, CA

## EDUCATION

---

Yale Ph.D. in Computer Science

2023 - Present

UCLA B.S. in Computer Science

GPA: 3.782, 2018 - 2022

## PROFESSIONAL EXPERIENCE

---

### Khandelwal Lab, Yale CS Department

*PhD Student*

Aug 2023 - Present

*Supervisor: Prof. Anurag Khandelwal*

- Currently researching performance limitations in kernel-based RDMA and CXL disaggregated memory runtimes. Publication ongoing.
- Developed extensive knowledge of the the Linux kernel's memory management, RDMA, page placement, and swap subsystems. Particular focus on optimizing high throughput, low-latency runtimes, such as memory migration systems and RDMA SmartNIC network stacks. Comfortable working at the hardware-software interface.

### SOLAR Lab, UCLA CS Department

*Student Researcher*

Sept 2021 - Sept 2022

*Supervisor: Prof. Harry Xu*

- Volunteered during the school year, employed full-time to work on kernel patches over the summer. Developed kernel mechanisms to facilitate allow for transparent memory disaggregation. Worked heavily with kernel programming and debugging tools, such as QEMU + GDB, serial port debugging, and perf.
- Independently developed patches for the Linux kernel's swap subsystems, with the goal of merging these changes upstream. Patchset extends the cpuset controller to allow per-cgroup control of active swap devices. Associated refactoring has positive implications for swap throughput, and makes it easy to manage frontswap-based remote memory systems. Patchset available at [github.com/yashlala/canvas-linux](https://github.com/yashlala/canvas-linux).
- Developed a patchset to improve the Linux kernel's physical page allocation latency. The patch reduces tail latencies by refilling the percpu low-order free page lists asynchronously using RCU.
- Profiled swapout latencies for RDMA-based remote memory systems under various workloads and prefetch strategies.

### CSSI Program, UCLA CS Department

*Tutor Undergrad (TA)*

July 2022

- Taught introductory data science to high school students for an intensive summer program. Led discussion sections, prepared discussion material and assignments, graded papers, and advised students.

### Veritas Technologies LLC

*SDE Intern*

June 2021 - Sept 2021

- Worked on large-scale data consolidation and backup devices (NetBackup Flex platform). Implemented functionality allowing Flex nodes to automatically discover new backup nodes over the datacenter network, then to securely assimilate them into a backup cluster. Primarily worked with Ansible, Docker, and various glue languages.

### Pringle Lab, Stanford Genetics Department

*Undergraduate Research Intern*

June 2017 - August 2017

- Tested algal species for selective binding to various lectin proteins in order to understand the chemical processes behind coral bleaching. Poster available at [yashlala.com/pringle-poster.pdf](http://yashlala.com/pringle-poster.pdf).

## PROJECTS

---

### NDN Multicast

May 2022 - Present

- Worked on extending routing protocols for NDN (Named Data Networks). Extended NLSR (a link-state routing algorithm for NDN) to allow for efficient multicast delivery of NDN Interest packets. Student paper available at [yashlala.com/nlsr-poster.pdf](http://yashlala.com/nlsr-poster.pdf).

### GRU4RecBE: Session Based Recommendations with Features

March 2021 - June 2021

- Developed session-based recommendation system in PyTorch which extends the GRU4REC architecture with rich item features extracted from the pre-trained BERT architecture. Non-attentive model outperforms state-of-the-art session-based models over the benchmark MovieLens 1M and MovieLens 20M datasets. [Paper accepted to AAAI Student track](#).

## MISCELLANEOUS

---

Passionate about teaching: led student [seminar series](#), volunteered with [ACM TeachLA](#) and as a docent for the [Computer History Museum](#). Certified as a operator for the punch-card based [IBM 1401](#).