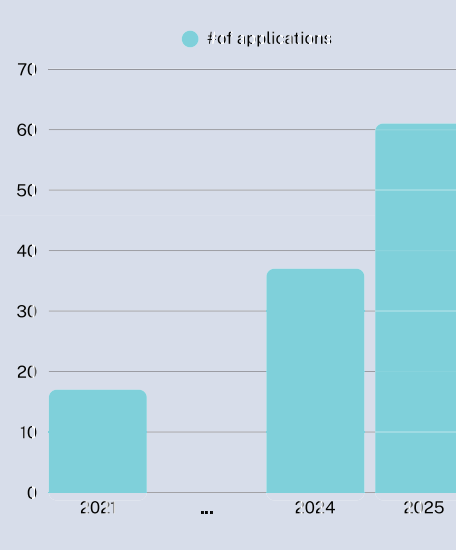




## 1 About UCSD Sea++

**Team Sea++** brings students across multiple disciplines and years together, cultivating an interest in high performance computing at the undergraduate level. The team is under UCSD's Supercomputing Club, which hosts various projects and workshops for the undergraduate engineering community at UCSD as well as the annual Single Board Cluster Competition.



### Diverse team and applicant pool:

- **Exponential rise in applications**
  - ~2x last year and ~4x 2021
- Team is an **interdisciplinary mix** of students studying computer science, computer engineering, electrical engineering, bioinformatics, and mathematics.
- **3/6** students identify as female and are supported by SDSC's Women in HPC chapter

**Gauri Renjith**  
 Captain / MLPerf

**Gauri Renjith**  
 3rd year Computer Science/Bioinformatics  
**Role: Captain, MLPerf Lead**  
 Gauri is passionate about the intersection of machine learning, HPC and biological data. She optimized molecular dynamics software at SCC24 and is excited to return this year!

**Ferrari Guan**  
 Sysadmin / HPL-MxP

**Ferrari Guan**  
 3rd Year Computer Engineering  
**Role: Sysadmin, HPL-MxP Lead**  
 Ferrari's interests lie in microcontrollers/embedded systems, Mechanical Computer-Aided Design, Printed Circuit Board (PCB) design, computer security, computer infrastructure, and operating systems. He was an alternate for the UCSD SCC24 team, where he worked on systems and networking.

**Luiz Gurrola**  
 Sysadmin / HPL

**Luiz Gurrola**  
 4th Year Math-Computer Science  
**Role: Sysadmin, HPL Lead**  
 Luiz is interested in all aspects of Computer Science with special interests in Digital Design, HPC, and Operating Systems. Luiz is constantly looking to improve his homelab and system administrator skills.

**Cecilia Li**  
 Reproducibility

**Jinru (Cecilia) Li**  
 2nd Year Electrical Engineering  
**Role: Reproducibility Lead**  
 Cecilia is excited to explore both the hardware and software components of HPC. She has hands on experience with hardware architecture research, sysadmin, and networking.

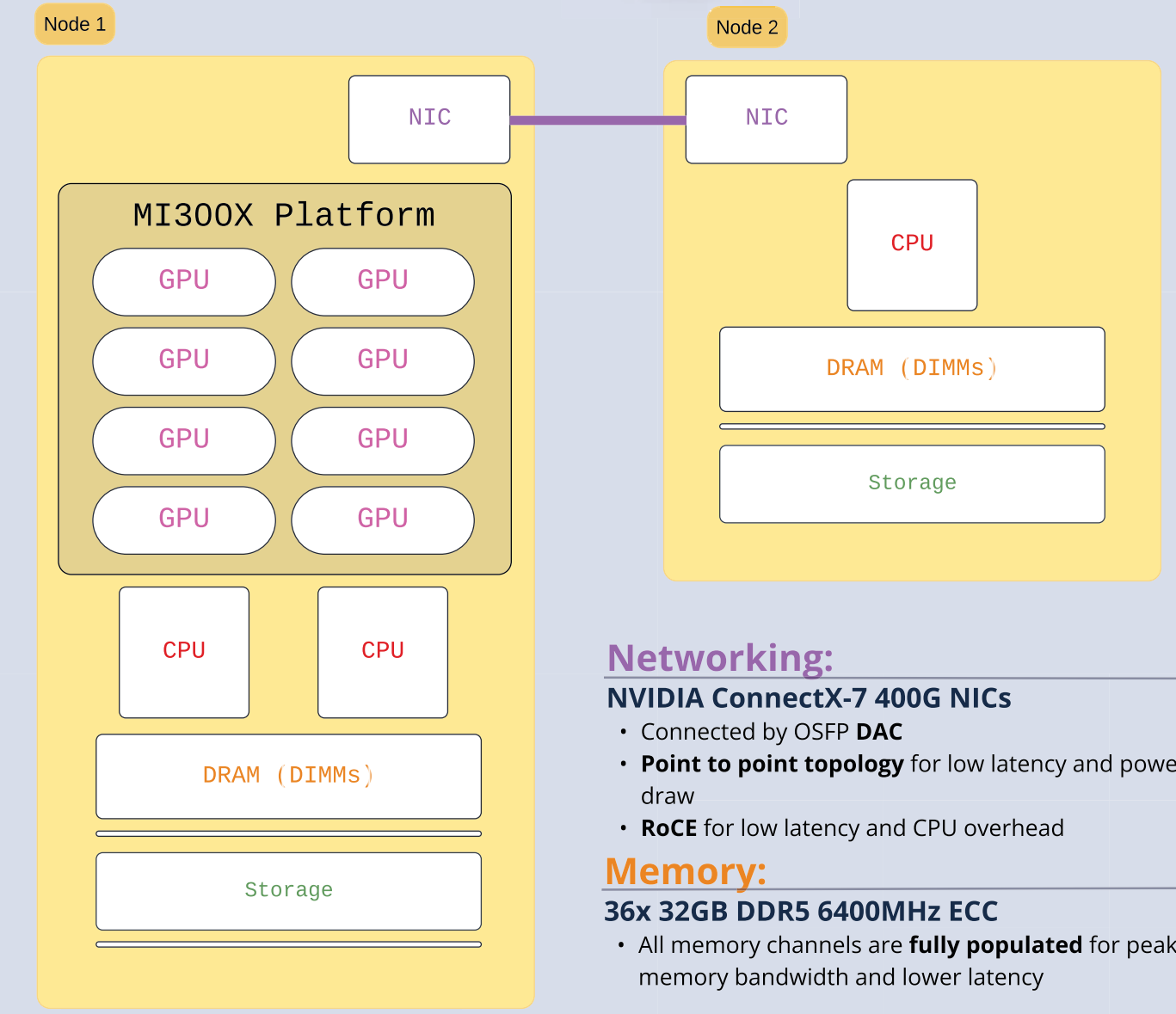
**Shing Hung**  
 SST

**Shing Hung**  
 4th Year Computer Engineering  
**Role: SST Lead**  
 Shing has diverse experience across embedded systems, hardware design, and computer science, specifically in PCB designs, microcontroller integration, and full stack website development.

**Ryan Estanislao**  
 ECE

**Ryan Estanislao**  
 2nd Year Computer Science  
**Role: ECE Lead**  
 Ryan has experience in machine learning and cybersecurity, and is further interested in the combination of computer science, data analytics, and econometrics.

## 2 Hardware Setup



**CPU:**  
 AMD EPYC Generation 5 Turin (9755 & 9745, 128C)  
 • 12 memory channels with 6400 MT/s, 614 GB/s Bandwidth  
 • Zen 5 architecture = higher frequencies and faster DRAM  
 • Node 1 has 2 AMD EPYC 9755s, Node 2 has 1 AMD EPYC 9745

**GPU:**  
 8x AMD MI300X OAM GPUs  
 • Quick memory transfer - 5.3 TB/s  
 • Easily handles large amounts of data - 192 GB HBM RAM per GPU  
 • Strong performance - 2614.9 FP8 TFLOPS, 1307.4 FP16 TFLOPS, 304 GPU compute units

**AMD MI300X Platform**  
 • Baseboard provides 128 GB/s links between any 2 GPUs

**Storage:**  
 ScaleFlux CSD 3310 7.68TB  
 • Hardware data  
 • 300x higher compression efficiency and 2x better storage IO

**Samsung PM9A3 1.9TBM.2 NVMe**

## 3 Software Stack

- OS: Rocky Linux**
  - Free and open source
  - Familiarity from mentors and team members via projects
- Filesystem: NFS**
  - Minimal setup, reasonable for small scale cluster
  - Access to files from either node
- Deployment & Management: Ansible Scripts**
  - For infrastructure setup and dependency installation
  - Ensures system and deployment reproducibility
  - Simplifies system setup and resets
- Monitoring: Grafana**
  - Prometheus backend for metric storage
- Tools: GNU Tools, HIPCC, ROCm, Docker, Anaconda, OpenMPI**



## 4 Application/Benchmark Optimization

### HPL and HPL MxP

**Building HPL/MxP:**

- Adapt the build script from **AMD Infinity Hub** for compatibility with our system's architecture and configuration

**Running HPL:**

- To achieve optimal performance, we follow the **recommended parameter configurations** from AMD Infinity Hub.
- For multi-node running, we need to test all the settings and find the **best asymmetric parameters** for our hardware setup.

**Running HPL-MxP:**

- To achieve the optimal performance, we refer to the parameters and sizes provided in the AMD Infinity Hub.
- To get the best multi-GPU run, we will rigorously test all the parameters options for our specific hardware architecture.

### MLPerf

- AMD provides **containerized setup** for MI300X nodes
- utilizes **FP8 quantization**
- explore **supported backends** (Ray, SGLang) and attention mechanisms *if time permits*
- We will:
  - test adjustable **performance environment variables**
  - test increased % of **GPU memory utilization**
  - **profile GPU usage** and efficiency with **ROCm Profile Data** tool

### Time and Power Strategies

**Time**

- **Coordinate applications and benchmarks runs** between teammates
- Schedule **mystery app runs** for the end of the 48 hour period

**Power**

- **Raspberry Pi** to monitor and log power and data
- **Grafana dashboard** to monitor power draw of each node, **webhooks** to send alerts
- **Manually lower the max TDP** of the GPUs if going over power

### SST

- Build and containerize SST with **Docker** for portability

**AMD CPU-specific strategies**

- **AOCL library** and architecture-specific flags to leverage hardware
- **Memory/NUMA optimizations**

**OpenMPI runtime process tuning:**

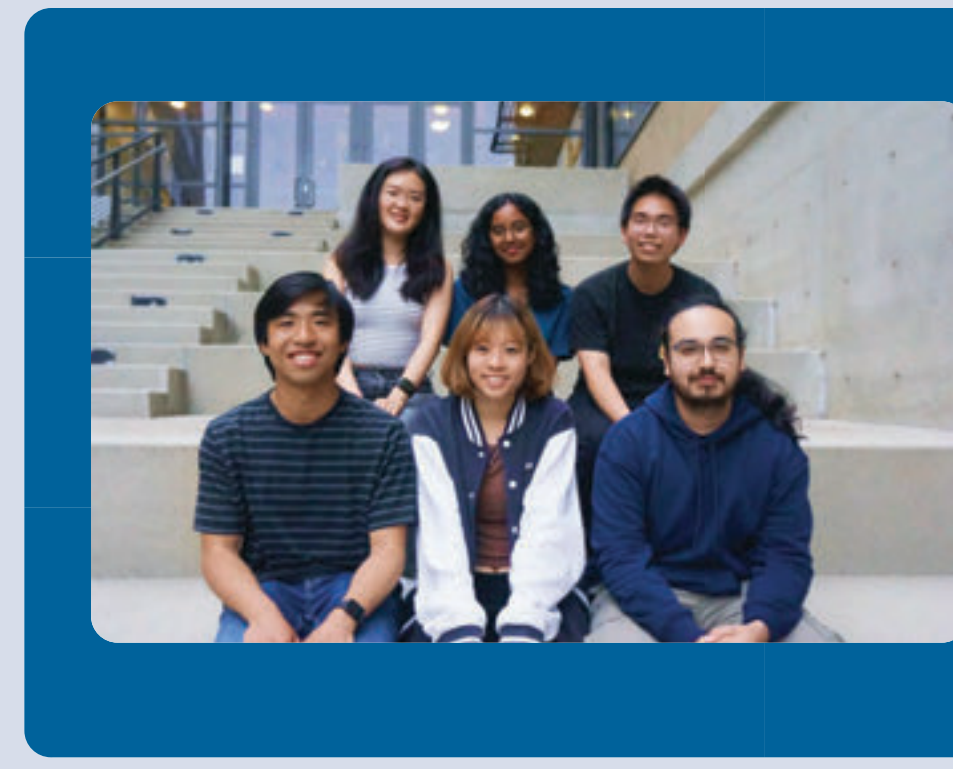
- **Binding ranks** to specific cores and sockets
- Testing OpenMPI flags to improve communication
- Improve **cache locality**

### Climate Emulation

- Install **HIP-enabled dependencies** (ex. StarPU, Chameleon) and ROCm math libraries to accelerate the linear algebra that underlies the emulator
- Profile with **ROCm tools** and adjust StarPU and Chameleon parameters on AMD's HPC Fund
- Explore and understand covariance approximations and how to take advantage to lower compute costs

### Reproducibility Challenge

- Aim to understand the **two kernels**
- Compare **FA-BSP** implementation vs **BSP (Ripples)** implementation
- Build **IMM HCLib** package and reproduce **figures** prior to competition for familiarity
- Run the provided graphs to collect:
  - **Runtime** metrics
  - **Scalability** metrics
  - **Memory** metrics
  - **Quality** metrics
- **Document hardware and software stack** extensively to streamline paper writing



### Our Competitive Advantage

Our team is extensively supported by the University of California, San Diego (the Department of Computer Science and Engineering in particular) and the San Diego Supercomputing Center. With access to SDSC's Expanse system and vendor research clusters to prepare and supportive faculty mentors and competition alumni, **we are confident that we can bring out the best of our team and our cluster.**