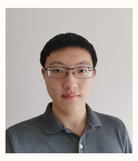
Team 2MuchCache

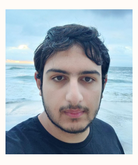
Longtian Bao, Stefanie Dao, Michael Granado, Yuchen Jing, Davit Margarian, Matthew Mikhailov (Students)
Mary Thomas[1], Bryan Chin[2] (Mentors)
Derek Bouius[3], Lewis Carroll[3], Andy Goetz[1], Robert Sinkovits[1], Mahidhar Tatineni[1] , Christopher Irving[1], Darshan Sarojini[2], Christine Fronczak[3], Matei-Alexandru Gardus[2], Arunav Gupta[2], John Li[2], Kaiwen He[2], Paul Yu[4] (Advisors)
[1] San Diego Supercomputer Center, [2] UCSD, [3] AMD, [4] Microsoft Azure

Team Members

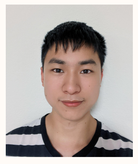
Stefanie Dao
Major: Cognitive Science
Year: Senior
Skills: Computer Vision, Computer Graphic, Distributed System, Cloud Computing
Role: Lead for PHASTA, Co-lead for HPL



Yuchen Jing
Major: Computer Science
Year: Senior
Skills: Web Backend, Systems Programming, Security, Rust, Open Source, Linux
Role: Lead for IO500, Co-lead for Data Centric Python



Davit Margarian
Major: Computer Engineering
Year: Sophomore
Skills: Computer Architecture, Embedded Systems, VLSI Design
Role: Lead for HPL, Co-lead for LAMMPS



Longtian Bao
Major: Computer Engineering
Year: Junior
Skills: Statistics, Android Development, React Native
Role: Lead for Data Centric Python, Co-lead for HPCG

Matthew Mikhailov
Major: Computer Engineering
Year: Senior
Skills: VLSI Physical Design, Computational Chemistry, Computer Architecture
Role: Team Lead. Lead for LAMMPS, Co-lead for IO500

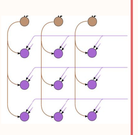
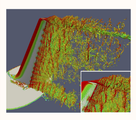
Michael Granado
Major: Computer Science
Year: Senior
Skills: System Administration, Software Engineering, Web Security
Role: Lead for HPCG, Co-lead for PHASTA

About Us

- **UC San Diego (UCSD)** is a large and diverse campus located in the border city of San Diego. The school is known for its research focus and great weather. The nearby beach is a popular attraction and so is the architecture of the campus.
- **The San Diego Supercomputing Center (SDSC)** is a leader in advanced computation (hosting HPC systems such as Expanse and Voyager) and "Big Data", which includes data integration and storage, performance modeling, data mining and predictive analytics, software development, and more.

- **Diversity Outreach**
  We held information sessions to promote SCC22 with the campus chapter of Women in Computing. We gave introductory presentations on supercomputing and opportunities at the San Diego Supercomputer Center and SC22, including the Student Cluster Competition. The sessions were well received, and contributed to the greater diversity on our team as compared to previous years. Our team also engaged with several student computing organizations to promote SCC22 on UCSD campus and open HPC training to anyone interested. We contacted student organizations for underrepresented groups in computing (female, ethnic minorities) to solicit applications for the SCC team.
- **Team Formation**
  Our proposed team with alternates has a diverse composition:
  - ✓ 22% identifying as female and 11% identifying with historically underrepresented groups
  - ✓ 1st/2nd generation immigrants and international students from 6 different countries: Vietnam, China, Ukraine, Russia, Zambia, Armenia
  - ✓ Speak 7 different languages: Chinese, English, Russian, French, Japanese, Vietnamese, Armenian
  - ✓ Major and minor in diverse disciplines: Computer Science, Computer Engineering, Cognitive Science, Economics, Mathematics, Civil Engineering
  - ✓ Have a broad range of specializations: Networked Services, VLSI Physical Design, System Administration, Security, Operating System, Computational Chemistry, Web Development, Statistics, Computer Vision, Neuroscience
  - ✓ Pursue different interests, both technical-related (research, contributing to Open Source, organizing Machine Learning workshops, ...) and non-technical related (hiking, drawing, badminton, swimming, Minecraft...) activities

The diversity of our team promotes creativity and wider skill sets, providing us with more perspectives and ideas for problem solving. Each person leads one benchmark and one application, but every team member practices every application and shares knowledge with the whole team.

App	Description	Cluster Strategy	Cloud Strategy
HPL 	• Solves a uniformly random dense linear system of equations using LU decomposition • Performance scales polynomially due to low I/O overhead and approximates the peak FP performance of a system • Dense matrix arithmetic is extremely efficient on GPUs	• Optimize HPL for maximum hardware utilization if possible • Use a public version of HPL optimized for the MI250 Accelerator • MI250 Accelerator shows very high peak FP64 performance that will translate into a high HPL score	• Use A100 GPUs for their peak FP performance: AMD Instinct not available in the cloud • Use containerization for portability and ease of configuration • Tune containers for peak performance
HPCG	• Tests a suite of operations such as sparse matrix-vector multiplication • Runtime dominated by Symmetric Gauss-Siedel Smoother: an iterative method of preconditioning linear equations so that conjugate gradient can solve them faster • Heavily memory bound: influenced by memory bandwidth and size	• Use an existing public version of HPCG optimized for the MI250 Accelerator • MI250 has very large HBM capacity and high memory bandwidth that will benefit HPCG performance	• Use A100 GPUs for great memory bandwidth and high floating point performance • Use containerization for portability and ease of configuration • Tune containers for peak performance
IO500	• Components: bandwidth & metadata • Bandwidth is primarily dependent on interconnect and storage speed • Metadata is primarily dependent on filesystem speed and parallelism of metadata in filesystem	• Fast but small SSD for metadata, bulky and slower SSD for content • Cache metadata aggressively • Use SSD optimized parallel file systems	• Use instances with best SSDs and optimized for storage I/O • Optimize instances for efficiency • Use SSD optimized parallel file systems
PHASTA 	• Parallel Hierarchic Adaptive Stabilized Transient Analysis of compressible and incompressible Navier Stokes equations • Smulations defined using CAD modeling systems, mesh generators, and attribute definition interfaces • Fluid dynamics typically demand high memory capacity and bandwidth	• Run the application on the CPU • Carefully tune OpenMPI parameters • Monitor CPU usages to reach higher efficiency • Fluid dynamics benefits from our large cache and per-core performance	• Run the application on the best CPUs as PHASTA benefits from high memory bandwidth and high core performance • Tune OpenMPI parameters
LAMMPS	• LAMMPS is a classical molecular dynamics code with supports a large variety of potentials and ensembles • Long range coulombic interaction calculations are FFT heavy	• Run the application on our cluster's GPUs, using the HIP version of LAMMPS • Carefully tune OpenMPI parameters	• Run the application on NVIDIA GPUs using the CUDA version if optimizing for total performance • Run the application on CPUs if optimizing for performance per dollar
Data Centric Python 	• Uses 2x Intel Xeon Gold 6130 CPU (32 cores) and NVIDIA V100 GPU to run DaCe (Data Centric Parallel Programming) programs • Paper also utilizes Intel and Xilinx to reproduce the results on FPGA • Converts Python code with NumPy to Stateful Dataflow multiGraphs IR	• Compare DaCe to high performing Python math libraries on CPU & GPU • Find differences between our cluster and their hardware	• Compare DaCe to best available libraries on low-performance and exotic hardware • Look into using vendor-specific libraries to take advantage of hardware
Mystery 	• In preparation for the Mystery Application, we will practice compiling with Make and CMake to link the highest performing libraries with ease.	• Benchmark and practice simultaneously with other applications to save resources • Identify performance bottlenecks and use optimization solutions practiced during competition training	• Identify the highest performing existing port of the mystery application and run on corresponding hardware • Set cloud settings based on the performance bottleneck: compute, memory, storage, or network

Sponsors

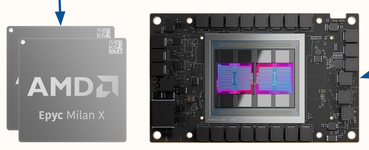
UCSD, SDSC San Diego Supercomputer Center, AMD, Azure, Hewlett Packard Enterprise, Dell, NVIDIA, Supermicro, NSF

Hardware Configuration

Over 4 Power Cables !

2 TB RAM
DDR4 at 3200 MT/s
32x 64GB in 8 channels
204.8 GB/s bandwidth

2x **AMD Epyc 7773X** CPUs
64 cores | 128 threads
768 MB stacked-die cache



- 3D V Cache enormously benefits single-threaded CPU performance
- HBM useful for benchmarks
- Running on one node minimizes parallelization and I/O overhead
- Die-to-die interconnect and Infinity Fabric link between GPUs minimizes I/O overhead
- Small, fast SSD can be used for metadata caching while large SSD is used for data
- AMD EPYC CPUs provide best in class performance per Watt

3.84 TB SSD
6.9 GB/s Read
1 GB/s Write
Up to 1M IOPS

4x **AMD Instinct MI250** Accelerators
832x CDNA2 CUs (53k simultaneous threads)
4x 128 GB HBM2e at 3.3 TB/s
800 GB/s Dual-die Interconnect per GPU
90.5 TFLOP/s FP64 Matrix per GPU

HPC applications and benchmarks are frequently both memory and I/O limited. 3D V Cache drastically increases our CPU cache size, helping with memory bottlenecks on CPUs. Additionally, the HBM on our accelerators helps with remaining memory bottlenecks in our system. Our D2D interface means that we have fewer MPI tasks for the same performance, so we save on the I/O bottleneck. Finally, our Infinity Fabric, HBM, and large storage also provides us with additional I/O benefits.

Software Configuration

System Software
- For our OS, we plan to use Ubuntu 22.04 for recent software. We will also consult and evaluate performance of different OSes and kernel versions on our hardware.
- We will deploy AMD pre-optimized containers for our hardware whenever possible. If not, we will containerize everything with Singularity (performance permitting).
- SLURM for job scheduling and Spack & lmod for package management.
- BeeGFS or other parallel file system for IO500 based on benchmarking results.

Compilers & Libraries
- GCC, OpenMP, AOCC, OpenMPI, and AOCL (BLIS + libFLAME + ScaLAPACK + FFTW + ...).
- These libraries are AMD compiled and pre-optimized, matching our system hardware well and allowing us to use all the compiler flags recommend by AMD.

Installation Tools
- VM images to upload all software beforehand and SLURM for on-the-fly installation.
- Self-hosted install scripts and programs in Git repos allows versioning and automation.
- Write automatic deployment scripts and containerize applications if possible, for maximum reproducibility and system stability.

Performance Monitoring
- Grafana dashboard for instance and cloud with CPU and GPU usage monitoring.
- Monitor power usage with a power meter reporting data to the dashboard.
- Write custom software to make automatic estimations of performance & decision making with the data collected to the dashboard.

Cloud Software
- Infiniband on Nvidia GPUs with NVIDIA HPC SDK.

Cloud Configuration

Azure SKU	Google Cloud SKU	AWS SKU

• **Dpsv5-series**
  ○ Ampere Altra Arm processor
  ○ Good low-cost option
• **HB-series**
  ○ AMD EPYC™ (Milan)
  ○ High performance CPU
• **NC-series**
  ○ NVIDIA A100 GPUs
  ○ High performance GPU
	• **Tau T2D**
	  ○ Ampere Altra Arm processor
	  ○ Good low-cost option
	• **Tau T2A**
	  ○ AMD EPYC™ (Milan)
	  ○ High performance CPU
	• **C2D Series**
	  ○ NVIDIA A100 GPUs
	  ○ High performance GPU
		• **EC2 C7g**
		  ○ Arm-based AWS Graviton3
		  ○ Good low-cost option
		• **EC2 Compute-Optimized Series**
		  ○ AMD EPYC™ (Milan)
		  ○ High performance GPU
		• **EC2 P4**
		  ○ NVIDIA A100 GPUs
		  ○ High performance GPU

Competition Preparation

Training
- Attend SDSC HPC User Training classes (or self guided tutorials)
- Two weekly "flipped" classroom training days
  1. Students will meet together take a look at the topic of the week and work on the material together
  2. Mentor sync meeting where students ask questions and learn additional information on the weekly topic
- All team members will practice compiling and running benchmarks on the local SDSC Expanse Supercomputer
- Team will transition to running applications on physical cluster
- We will hold hackathons aimed at simulating the competition

Benchmarking
- Those who are designated as co-leads and leads will engage in further benchmarking and exploratory analysis
  - Benchmarks will be run with different tuning parameters to analyze the optimal configuration
  - Applications will be run with various inputs in order to prepare different competition scenarios
- By using this data-driven approach, we hope to be able to gain a strong understanding of application and benchmark scaling so that we can better infer performance vs cost

Managing Resources

Time
- Equal distribution of applications/benchmarks
  - Each member must pick one benchmark and one non-mystery application to focus on, structured in a lead/co-lead format
- Running benchmarks minimally
  - Due to the more predictable nature of the benchmarks, we aim to run the benchmarks only once or twice and transition our benchmark team leads to the mystery application

Power
- Geist Power meter will be interfaced with our Grafana Dashboard, allowing us to effectively monitor the power distribution in our cluster
- *cpupower* will monitor cpu power states
  - Will be used in conjunction with the Grafana Dashboard so we may track per core usage
  - By tracking CPU utilization, we hope to gain an advantage in making decisions such as power optimizations, efficiently distributing our cores, and running CPU bound and GPU bound applications simultaneously
- *rocm-smi* will manage energy between hardware components
- Perform dedicated training tasks associated with power management
  - Write out intermediate states, shut down the node, restart node, and start application again from previous state
  - Practice using different power settings on the cluster

Cloud Cost
- Write custom, automatic performance vs cost estimator
  - By interfacing our software with our Grafana Dashboard, we will be able to keep track of estimates in real-time
- Cloud usage reports
  - Cloud usage reports will be provided as inputs into an estimator to tune our estimates