

Meet the Team

Ke-Ying Chen

- Contributions: ECE, HPL
- Skill: Parallel Computing, Competition Programming
- Trait: Team Leader



Bing-Hong Lin

- Contributions: ECE, HPL
- Skill: Parallel Computing, Distributed Systems
- Trait: Persistent, Hakka



Jo-Ching Ma

- Contributions: ECE
- Skill: Database
- Trait: Articulate



Jonathan Hsu

- Contributions: SST
- Skill: System, Robot, LLM Applications
- Trait: Comfy



Yi-Jia Chen

- Contributions: SST, cluster setup
- Skill: Profile
- Trait: Extroverted



Min-Shin Tsai

- Contributions: SST
- Skill: Cyber Security
- Trait: Detail-oriented



Team introduction:

Team NTHU, led by Prof. Jerry Chou, is a diverse group of talents from various disciplines, demonstrating strong interdisciplinarity and innovation. Although this is our first participation in the SCC, we have already developed excellent teamwork through the HPCAI and ASC contests. Our team stands out for its passion, rigorous training, and solid technical foundation, which together drive our continued excellence in high-performance computing.

Preparation

Our training

- First-Time Participants** — Every member joins SCC for the first time, we received systematic training to help us prepare for the competition.
- Mock Competitions** — Team-wide rehearsals to simulate real pressure and refine coordination for the competition and mystery task.
- Optimized Library Setup** — Pre-built common libraries with tuned compiler flags for maximum performance.
- Weekly Coaching Sessions** — Regular meetings with senior coaches and Professor Chou for technical guidance.

What We Gained

- Technical Foundations** — Strengthened understanding of simulation workflows and system-level operations in the office hours.
- Reusable Automation** — Developed clean, modular scripts for repeatable HPC workflows.
- System Management Skills** — Practiced software installation and dependency handling on real clusters.

Innovation

Automation Scripts

- Build & Deployment Tools** — Automated scripts for compilation, parameter validation, and log analysis.

Educational Impact

- HPC Outreach Program** — Organized biannual HPC camps and shared hands-on practices to inspire new learners.
- HPC Course TA** — Provided technical assistance and guided students through practical assignments on HPC cluster management and usage.

Resource Management

Compute Resource Management

- Automated Job Scheduling** — Maximizes cluster utilization through adaptive automation.
- Dynamic Time Estimation** — Predicts and refines runtime before and during execution.

Time & Team Management

- Team Collaboration** — One representative from each application team tackles the mystery challenge.
- Automated Node Monitoring** — Discord bot report system and node status in real time.

Application Exercises and Benchmarks

General Methodology

- Collaboration** — Tasks are assigned based on individual skills and availability, which accelerates optimization work and maximizes the utilization of computing resources.
- Profiling** — Using Intel VTune Profiler and NVIDIA profiling tools to identify application hotspots and analyze detailed time usage.
- Code Improving** — We reduced communication overhead and computation cost through cache optimization and branch prediction to improve performance.
- Compile** — We enable modern compiler features.

HPL

- Profiling** — We use NVIDIA Nsight Systems to identify bottlenecks and analyze communication patterns.
- Parameter Tuning** — We adjust PxQ, broadcast algorithms, and HPL look-ahead depth to reduce network/memory overhead and improve panel factorization.
- Computation Optimization** — We leverage external BLAS, SIMD vectorization, and compiler optimizations flags to improve performance.

SST

- Testing and Optimization** — We use scripts to test different parameters on sample cases, analyze influential factors, and develop a custom partitioner to achieve better load balancing.

ECE

- Profiling** — We use NVIDIA Nsight Systems to identify scheduling issues and detect idle time.
- Parameter Tuning** — We optimize block & tile sizes, scheduling, and enable adaptive memory to reduce overhead.
- Tensor Core Optimization** — We leverage CUTLASS and cuBLAS libraries to fully utilize tensor cores for high-performance matrix operations.

Software Stack

Scheduler

HyperQueue

Profile Tools

VTune, iPerf, Nsight Compute, Nsight System

MPI

IntelMPI, OpenMPI, MVAPICH4, UCX, XPMEM

LIBs

CUDA, GNU RISC-V toolchain, CUTCLASS, Open BLAS, AOCL-BLAS

System & Compilers

OS - Ubuntu 24.04 LTS
FS - CephFS
Intel OneAPI 24, NVHPC 24
AOCC 5, GNU13, LLVM