

Students: Bryan Guanrong SHAN, Boyuan GU, Ruey Yee HONG, Yoong Ken TAN, Yichi ZHANG, Zhanyan ZHU

Advisors: Bu Sung LEE, Yuan Ren LOKE, Chen WANG, James CHEN, Terry YIN, Danxu ZHANG

Introduction



Nanyang Technological University

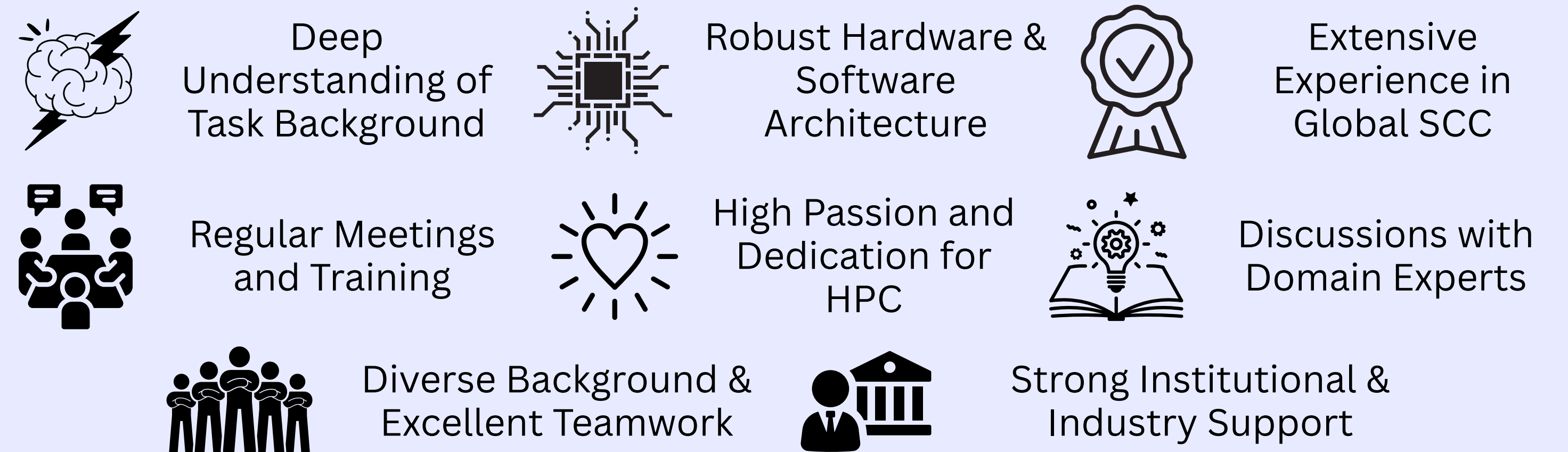
Established in 1991 in Singapore, is a globally renowned university focused on research and innovation. It excels in areas like artificial intelligence, sustainability, and biomedical sciences, with worldclass research centers and an HPC-supported Parallel and Distributed Computing Lab.

Our Team

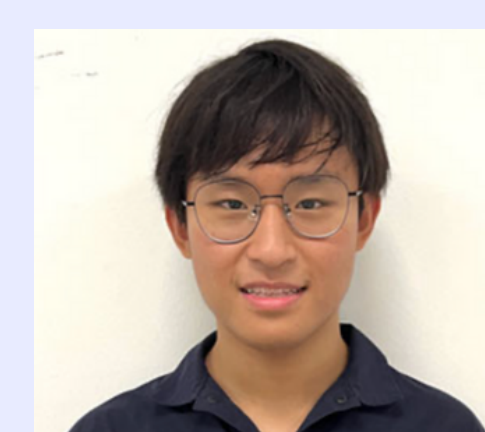
- Multinations: Singapore, Malaysia, China
- Diverse courses and field of interests: Algorithms and Modeling, Hardware and Systems Knowledge, Software and Optimization, Strategic Planning and Coordination, Benchmarking, Parallel Programming, Cyber Security
- Intern/Research experiences:
 - National Supercomputing Centre (NSCC) Singapore
 - A*STAR - Agency for Science, Technology Research
 - ASE, IMDA, IRAS and etc
 - OpenFOAM Research



Our Strengths



Team Members



Bryan Guanrong SHAN (Captain)

- Business and Computer Science, Y3
- ISC24, SC24, ISC25
- Exascale Climate Emulator, HPL, Hardware Management



Yichi ZHANG

- Data Science and Artificial Intelligence, Y3
- New Member
- Reproducibility Challenge, Mystery Application



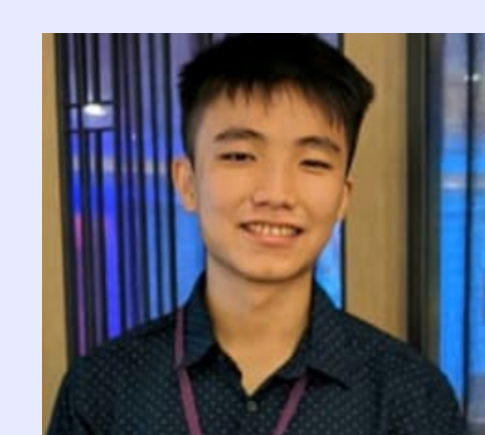
Yoong Ken TAN

- Computer Science, Y4
- ISC25
- MLPerf, SST, HPL-MxP



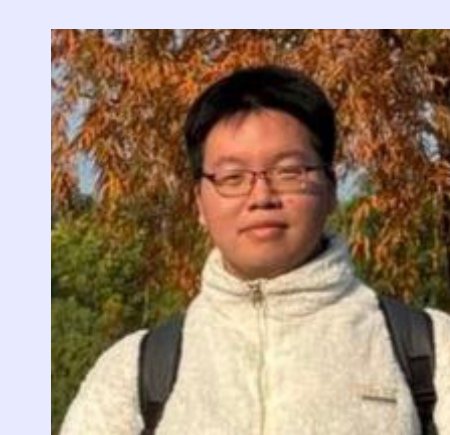
Ruey Yee HONG

- Mathematical Science, Y3
- New Member
- HPL, HPL-MxP



Zhanyan ZHU

- Computer Science, Y3
- ISC25
- MLPerf, SysAdmin, Hardware Management



Boyuan GU

- Computer Science, Y2
- New Member
- SST, Reproducibility Challenge

Our Cluster: Coffeepot



Hardware Choice:

- 3x Supermicro GPU A+ Server AS-4125GS-TNRT2 servers
- Dual sockets AMD EPYC 9654 96-Core Processors
- 6x Samsung NVMe SSD
- 20x NVIDIA H100
- NVIDIA ConnectX-7 400Gbps InfiniBand Network
- Raspberry Pi for system monitoring

System Security:

- Role-based access control (RBAC)
- Firewall with port restrictions
- Strong password policy
- Centralized logging and monitoring (Prometheus, Grafana, SIEM)



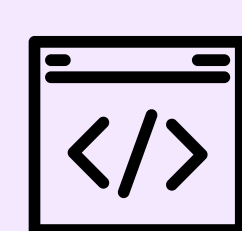
ubuntu

Software Choice:



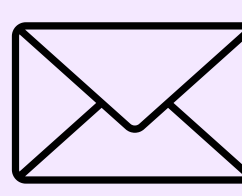
OS: Ubuntu 22.04 with Linux Kernel 6.5.0-27

Module Management: Lmod



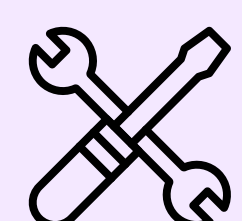
Compilers:

AMD AOCC, Intel OneAPI, GNU gcc, CUDA, nvhpc, clang-LLVM



MPIs:

MVAPICH, Intel MPI, OpenMPI



Tools:

BeeGFS, EasyBuild, Docker, Singularity, SLURM, Ansible, Prometheus, Grafana



Profiler:

IPM, ScoreP, NVIDIA's Nsights, Intel Vtune, Darshan, TAU, HPCToolkit

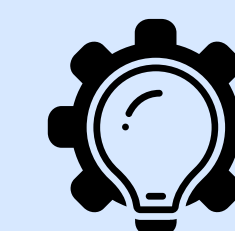


Libraries:

MKL, OpenBLAS, HDF5, NetCDF, ScaLAPACK, fftw

Application Optimisation

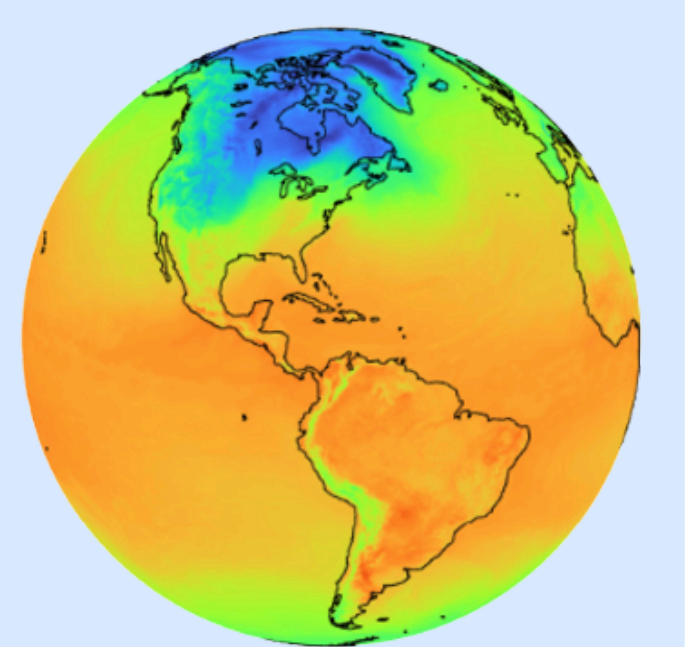
General Methodology



1. Assess different compilers and optimization flags, while benchmarking library versions to ensure compatibility and stability.
2. Profile system performance using TAU, IPM, mpiP, Nsight, and Intel VTune to identify bottlenecks and inform targeted optimizations.
3. Review relevant research literature and seek guidance from professors and industry experts to refine strategies and approaches.
4. Evaluate GPU performance by testing various GPU topologies across nodes to identify the most efficient configurations.
5. Leverage OpenACC and device offloading techniques to effectively transfer computational kernels to accelerator cards for improved performance.

Exascale Climate Emulator

- A high-performance climate simulation tool designed for detailed modeling of Earth's climate systems, leveraging exascale computing to handle large-scale atmospheric, oceanic, and terrestrial interactions.
- We optimize the application by using profiling tools to identify performance bottlenecks in the Parsec DAG, and then refining task scheduling and data dependencies to improve simulation efficiency and reduce computation time.

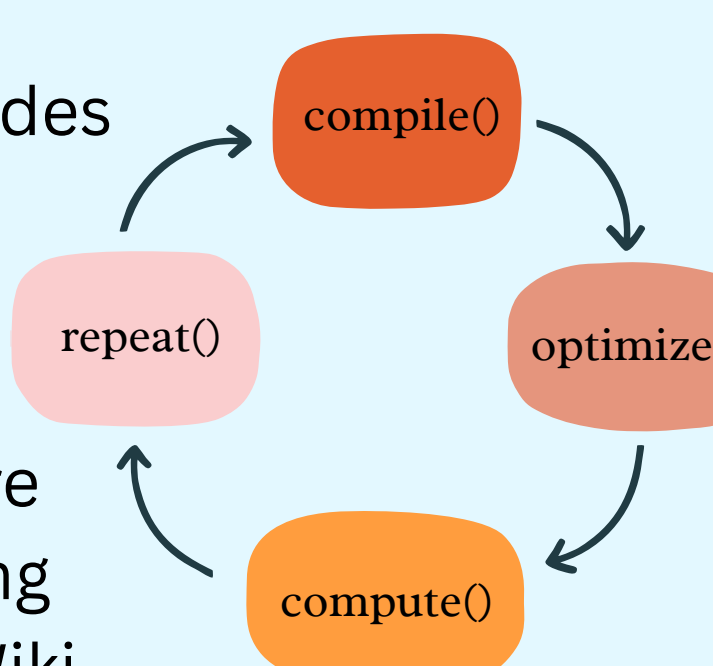


Running Strategies



Power Consumption, Monitoring, Noise Level Control, Automation on Deployment

- Tracked memory, CPU, GPU, and power usage with Prometheus and Grafana, and managed fan speeds via IPMI tools.
- Streamlined system management across three nodes using Ansible Playbooks and Terraform for consistent deployment.
- Applied fine-grained hardware control through custom tuning scripts, guided by NTUHPC Wiki references.
- Reduced noise levels using cushions and foam for effective sound insulation.



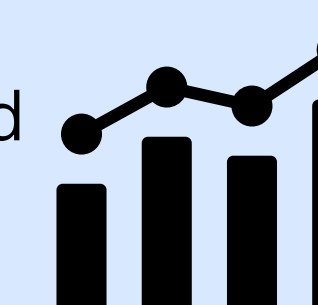
Runtime Schedule and Scripts Writing

- Evaluated HPC application scalability across nodes and cores to detect bottlenecks in load balancing, communication, and memory usage.
- Tuned MPI configurations, compiler options, and related parameters for optimized and stable performance.
- Executed benchmarks with prepared scripts to validate system setup during competition runs.
- Used SLURM to schedule multiple tasks and manage cool-down intervals between jobs.

Benchmarks

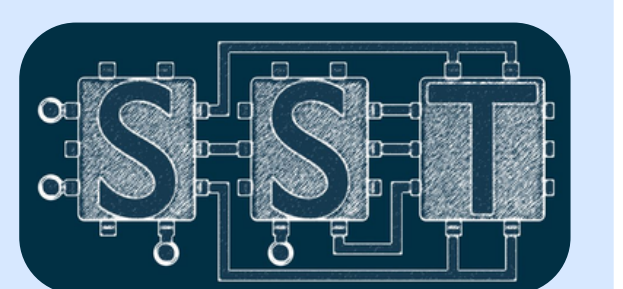
- **HPL** → Focus on maximizing floating-point throughput by optimizing memory bandwidth, enhancing parallelism, and efficiently solving dense linear systems via LU factorization across multiple nodes.
- **HPL-MxP** → Tune matrix and block sizes to fully exploit tensor cores, overlap communication with computation for optimal multi-GPU scaling, and apply NVIDIA's optimized math libraries with fine-tuning under power constraints.
- **MLPerf** → Improve GPU utilization and scaling across multiple GPUs, optimize data loading and memory management, and ensure efficient distributed training and inference workflows.

For all benchmarks, the latest NVIDIA binaries will be deployed with careful adjustments to meet power budgets.



SST

- A modular simulation framework designed for high-performance modeling of computer systems, enabling exploration of processor, memory, and interconnect architectures at scale.
- We will be trying different partitioning methods and applying optimization techniques to further improve the performance of SST.



Reproducibility Challenge

- Reproduce IMM parallel optimizations with data distribution and asynchronous parallelism.
- Evaluate performance and scalability on large real-world graphs.

