Team NTU

Tso-Fei Yen
ECE / System Setup

Jui-Chien Tsou
MLPerf / Reproducibility Challenge

Wei-Chin Wang
HPL & HPL-MxP / SST

Kuan-Hsun Tu
HPL & HPL-MxP / System Setup

Hsuan-Chi Liu
MLPerf / ECE

Chia-Yi Chin
HPL & HPL-MxP/ SST

Prof. Chun-Yi Lee
Advisor

About Us!

We are the National Taiwan University (NTU) team! Our six undergraduate members unite diverse academic and personal backgrounds, strengthening our ability to tackle the challenges of the SC25 Student Cluster Competition.

Our team composition reflects both depth and breadth. Tso-Fei, Jui-Chien and Hsuan-Chi major in Computer Science and Information Engineering, while Chia-Yi and Wei-Chin are Electrical Engineering majors who transferred from Mechanical Engineering, bringing expertise in thermal dynamics and physical systems. Wei-Chin also pursues a double major, bridging hardware and software. Kuan-Hsun transferred from Ocean Engineering, adding interdisciplinary insight. This mix allows us to combine strong algorithmic and system administration skills with perspectives from physics, bioinformatics, and computational engineering.

As a team, we already demonstrated our capabilities at ASC25, where we earned both the Group Competition Award and First Prize. That experience validated our collaborative problem-solving approach and technical strength on the international stage.

Our diversity also extends to leadership. In a field traditionally male-dominated, having a female student leader reflects NTU’ s commitment to inclusion. This balance of perspectives, combined with each member’ s active contributions in hardware tuning, application optimization, and system integration, makes us a cohesive and resilient unit ready for SC25.

Fig. 1: Team NTU @ ASC25

Fig. 2: The SC25 Team members!

Hardware

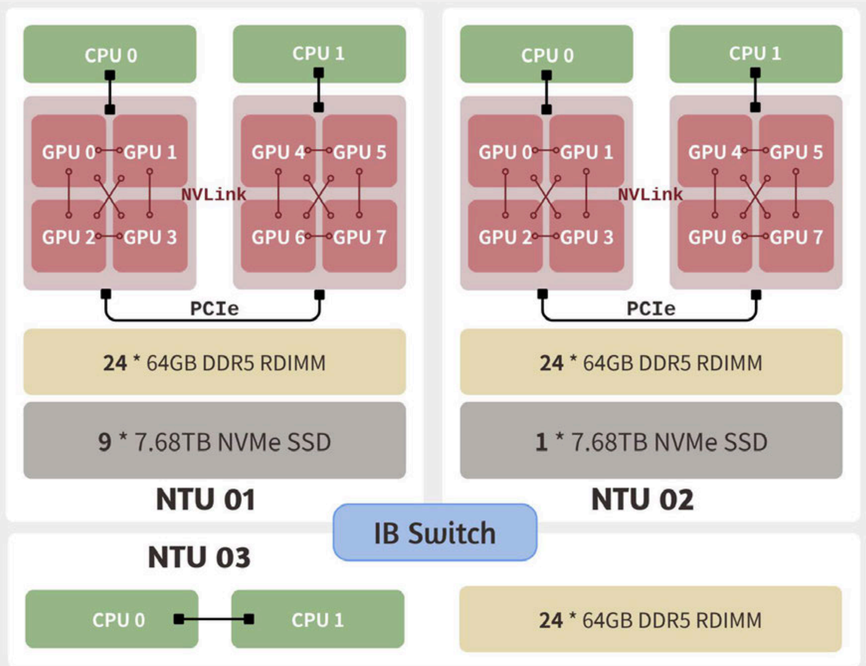

Fig.3: Cluster topology

Our cluster consists of **3 server nodes**. The topology is illustrated above.

- GPU: **Nvidia H200**
  - Compared to H100, H200 has larger memory and consumes less power with the same performance, while maintaining similar speed.
- CPU: **AMD EPYC 9655 96-Core Processor**
- Memory: **64GB DDR5 RDIMM**
  - To maximize CPU performance, each memory channel (mc) is populated with a DDR5 module → 2 CPUs/node * 12 mc/ CPU* 1 DDR5/ mc = 24 DDR5/ node.
- Hard disk: **NVMe SSD**
  - All SSDs are centralized on the head node (NTU01) as NFS storage, allowing compute nodes to share data without redundant copies while maintaining high-speed access.
- Network adapter: **NVIDIA MCX755106AS-HEAT**
  - 200Gb/s InfiniBand ensures excellent cross-node performance for distributed computing tasks, minimizing communication overhead in MPI applications.

Software

- OS: Ubuntu 24.04.3
  - Kernel: Linux node1 6.8.0-71-generic
- Compiler: GNU compiler and Intel oneAPI
- MPI: OpenMPI and Intel MPI
- Profiling tools: Intel VTune

Strategies

Time and Team Managment

- Weekly meeting: syncing weekly progress and gathering insights from every member.
- At least 2 people per task: provide high availability and diverse ideas for each task.

Optimization Methods

HPL & HPL-MxP

- Run HPL and HPL-MxP with different configurations to identify performance trends.
- Utilize NVIDIA Enroot to ensure a stable environment with minimal overhead.
- Leverage InfiniBand, NVSHMEM, and GDRCopy to minimize communication overhead.

MLPerf

- We enabled FP8 for compute + FP8 KV-cache to reduce memory traffic and achieve higher throughput.
- We implemented Triton Inference Server to enable data parallelism and allow harness dispatch requests over two nodes via gRPC.
- We adjust per GPU power caps and CPU frequency to balance system power and throughput.

Structure Simulation Toolkit (SST)

- We simulate and optimize under various complex settings, such as 32 Vene cores with a complete NoC and all thread pipelines, because different cores may have different traffic patterns and workloads.

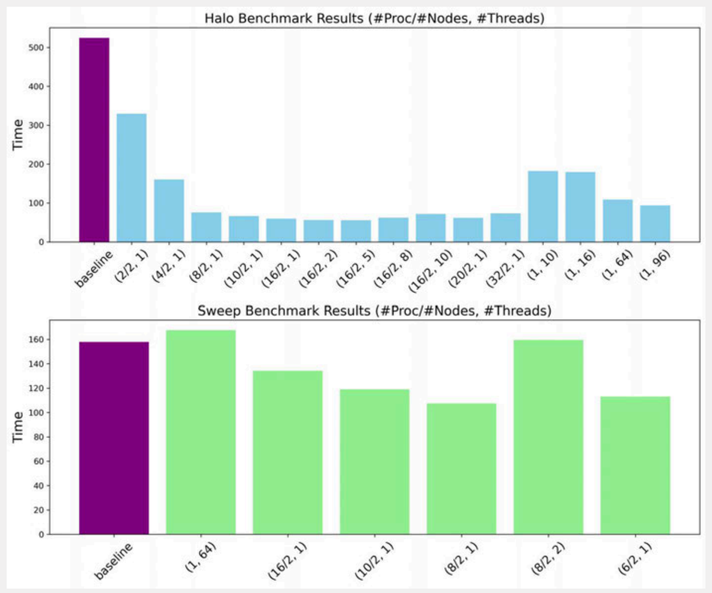

Fig.6: SST Network Simulation for Different Configurations

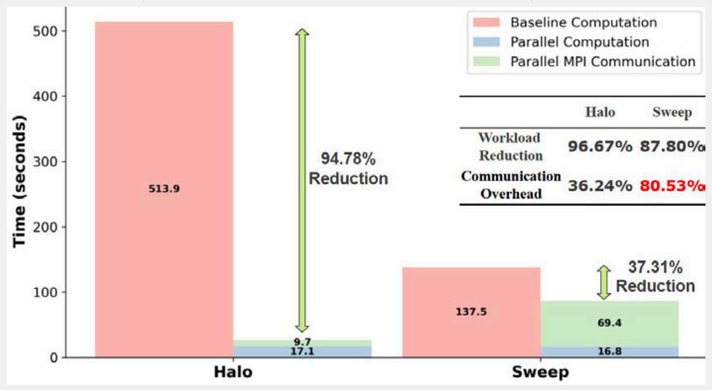

Fig.8: Runtime Savings and Communication Overhead in Parallel Execution

Exascale Climate Emulator (ECE)

- We use CPU-only MPI to retrieve mean trend parameters for the mean trend removal task.
- Following the paper: Boosting Earth System Model Outputs And Saving PetaBytes in their Storage Using Exascale Climate Emulators, we implement the SHT-based covariance + PaRSEC on Cholesky decomposition pipeline on larger or more complex climate dataset.
- We configured PaRSEC DTD execution with mixed precision on our two-node H200 cluster. We further tuned our tiles/ranks/GPUs configuration to maximize throughput while maintaining accuracy.

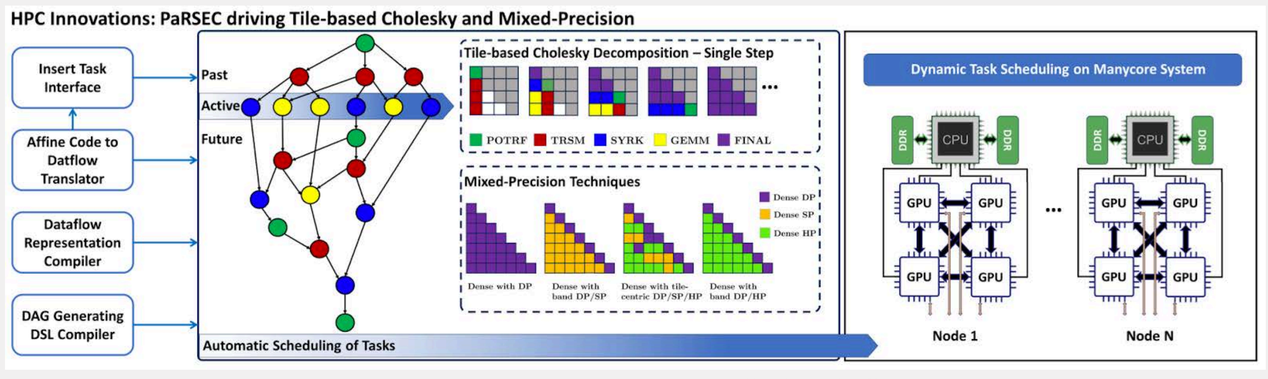

Fig. 10: PaRSEC workload distribution. Image source: **https://arxiv.org/pdf/2408.04440**

Reproducibility Challenge (IM Problem)

- Follow the paper to implement Actor IMM / Actor IMM-2D on our cluster.
- Perform strong scaling test and measure end-to-end and kernel time to test the performance and use VTune to profile the program.
- Tune the application with different MPI ranks and OpenMP thread number.
- Modify co-occurrence matrix implementation and design more efficient data structures.

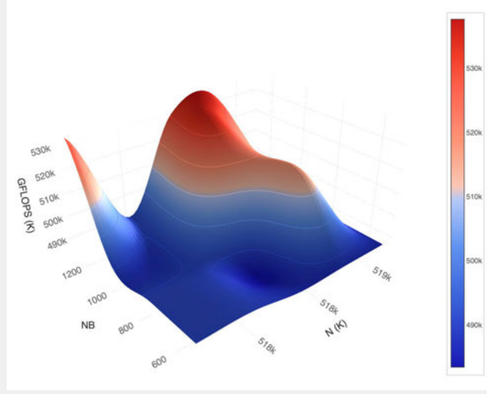

Fig.4: Performance (GFlops) of HPL under different parameters (N and NB)

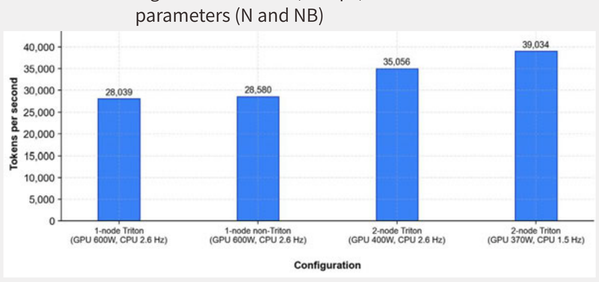

Fig.5: MLPerf tuning result: Triton server inference with 2 nodes outperforms single node implementation

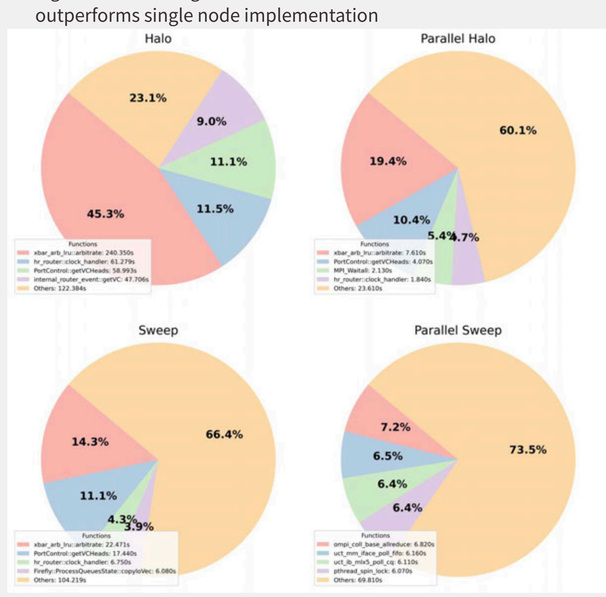

Fig.7: Time Breakdown in Serial and Parallel Execution

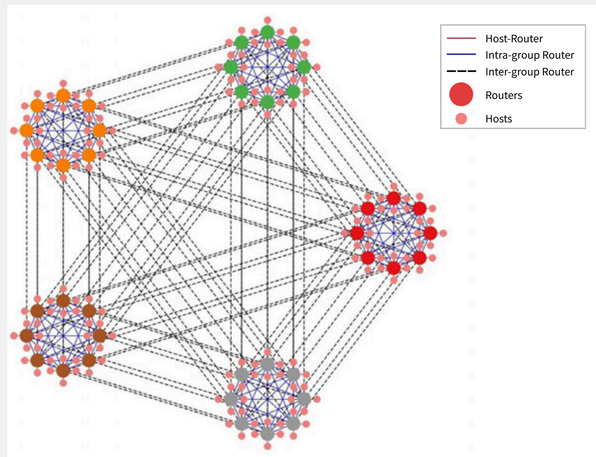

Fig.9: Dragonfly Topology