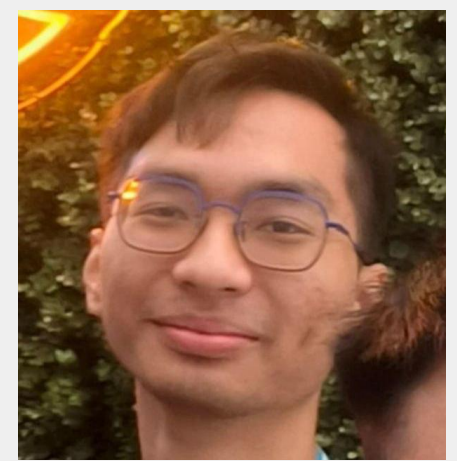


1. About Team Kent Ridge

We are a collective of highly talented students from the National University of Singapore (NUS), hailing from diverse backgrounds but sharing a passion for cutting-edge high-performance computing.

2. Our Members



Way Yan (Captain)
Likes math and digging deep into computer systems.



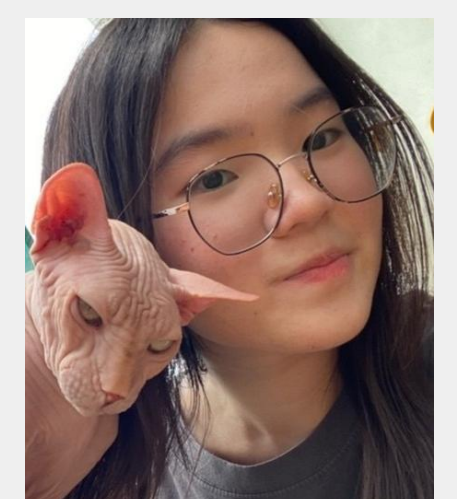
Zhi Shen
Enjoys reading up on and playing with hardware and system tuning and observability.



Taanish
Passionate about optimising AI-driven systems at scale.



Amy
Likes learning about low-level architecture and parallel computing.



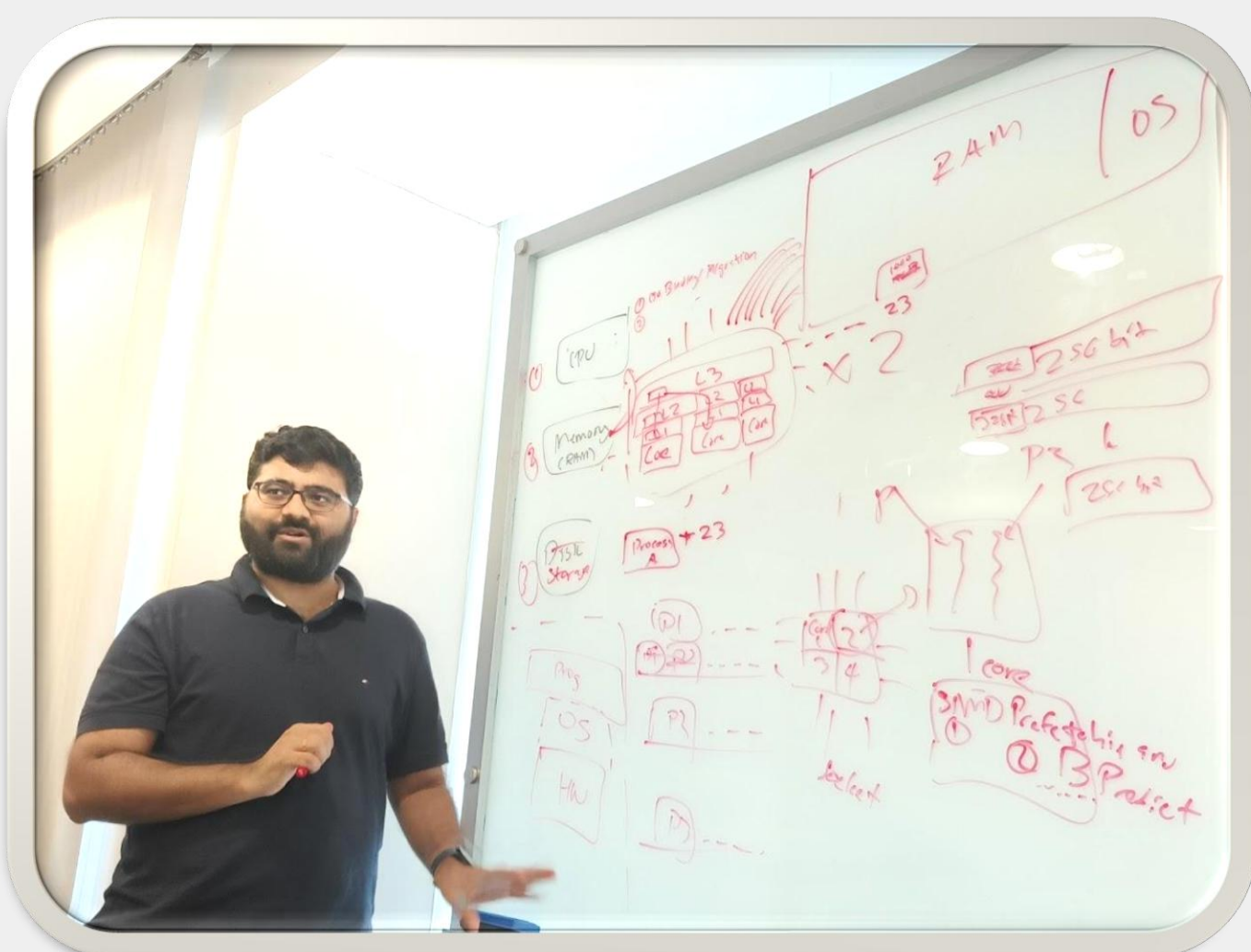
Gabriella
Enjoys tackling complex algorithmic problems and troubleshooting nasty bugs.



Xinyang
Very experienced at optimising real-world production systems.

3. Our Preparation

- Weekly syncs to align progress and resolve blockers.
- Weekly clinics for newer members to acquire the necessary skills.
- Experiment with system setup and tuning on our very own CPU cluster.
- Conduct training runs on NUS SoC's CPU+GPU cluster.
- CS2106 *Introduction to Operating Systems* and CS3210 *Parallel Computing* provides us with strong fundamentals for HPC.



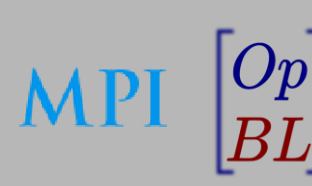
4. Our Cluster

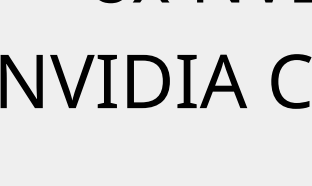


















Hardware choice:
AMD EPYC 9965 CPUs
8x NVIDIA H100 GPUs
NVIDIA ConnectX-6 NICs

5. Our Strategy

HPL, HPL-MxP

IC: Way Yan

- Extensive and efficiently automated experimentation to pinpoint the best parameters (particularly N, PxQ and NB).
- NSight profiling to better understand performance characteristics and bottlenecks on GPUs.
- Fine-tune inter-GPU communication patterns (NCCL, MPI-aware CUDA, UCX), as well as CPU-GPU communication.

MLPerf Inference

IC: Taanish

- Use lower precision within the stacks (FP4-FP8).
- Implement fused attention with TensorRT-LLM.
- Quantise the K/V cache and use segmented KV, so as to reuse allocations.
- Tune parameters to maximise tensor parallelism.

Exascale Climate Emulator

IC: Amy

- Read up on the mathematical theory and implementation details, so as to guide our optimisation.
- Familiarise ourselves with parsec.
- Get comfortable with the build process.
- Compile with architecture-specific libraries (AMD, CUDA).

Structural Simulation Toolkit

ICs: Gabriella, Zhi Shen

- Pre-discover a working set of parameters that have greatest impact on performance.
- Profile using scorep and built-in benchmarking tools to discover bottlenecks.
- Prepare batch files with predefined parameters.
- Fine-tune rank and thread allocation of the simulations to reduce communication latency.
- Draft cluster scheduling plan to allocate runs according to parameter importance.

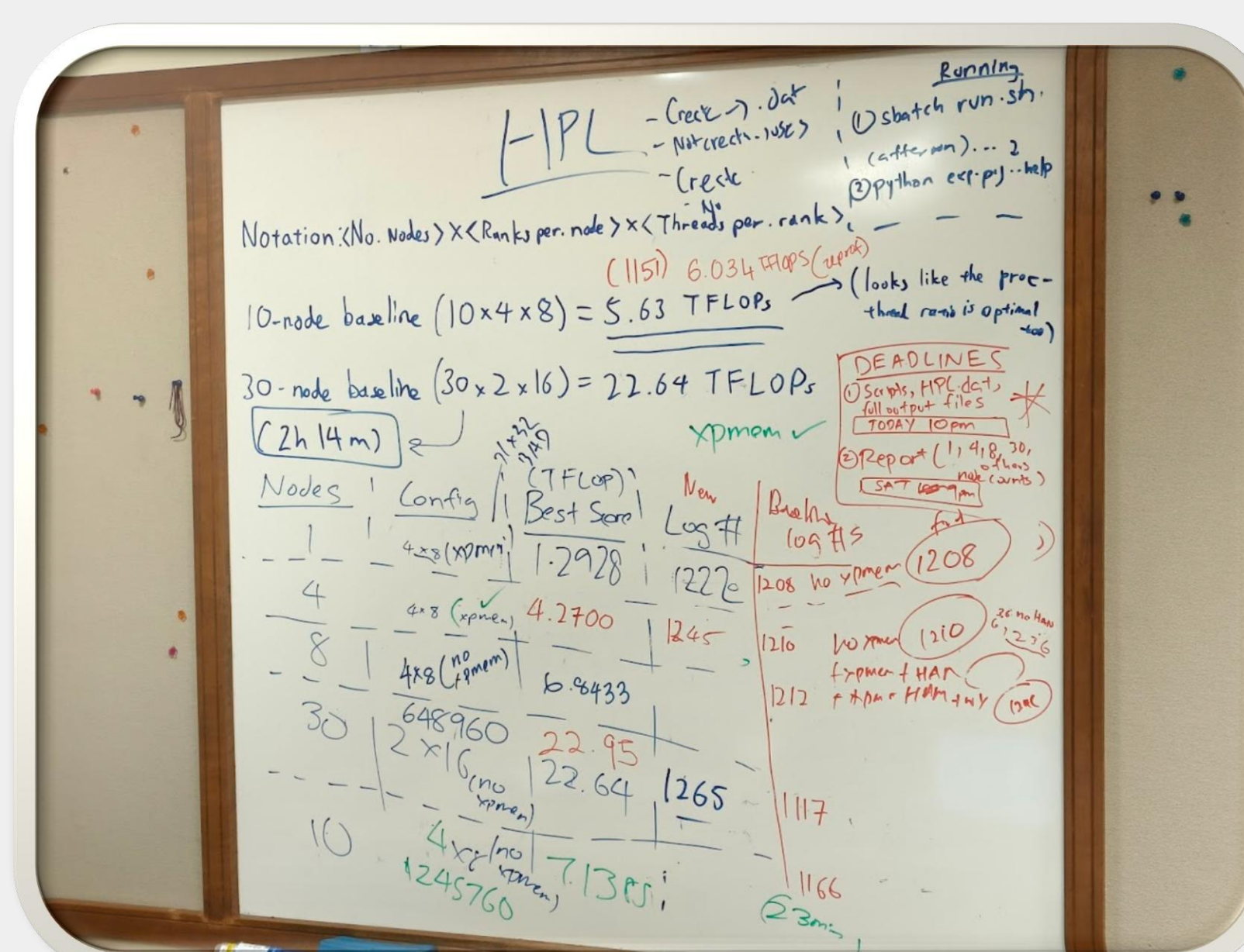
Reproducibility Challenge

IC: Xinyang

- Reading up of reproducibility paper early.
- Start experimenting with building the applications and running with the different application flags to test impact on performance.

System-Level Optimisations

- Tune process scheduling and memory management to maximize performance.
- Control process management and interrupts on CPUs.
- Conduct frequency tuning of nodes with CPU governors, pstate management.
- Disable ECC on GPUs.
- Tune the underlying filesystem to increase parallel performance.



We are grateful for the support of our sponsors: