

Team RACKlette

Swiss Federal Institute of Technology Zürich
Swiss National Supercomputing Centre



About Us



Our Institution. ETH Zürich is one of the world's leading universities, renowned for its research and innovation in science and technology. Established in 1855, it has produced numerous Nobel laureates and remains a driving force behind global advancements in engineering, natural sciences, and computer science.

The whole Team. RACKlette (Yes, like the Cheese!) is a student team from ETH Zürich passionate about high-performance computing. It comprises 18 active members from diverse fields like Mathematics, Computer Science, Physics, and Computational Science & Engineering, the team collaborates on optimizing computations from hardware to algorithmic performance.



Swiss National Supercomputing Centre. Our work is closely integrated with CSCS, which provides the infrastructure, ongoing maintenance for our clusters, and valuable workshops. The image depicts one of their clusters, ALPS, currently ranked 6th on the Top500 list.



Diversity of the Competing Team

Diversity of Genders:

- We actively create a welcoming environment for both women and men throughout the recruitment process and our Team.

Diversity of Major/Skills:

- Our competing Team includes students from computer science and computational science & engineering.
- CS students specialize in hardware and software optimization, while CSE students bring deep scientific insights and intuition, uncovering hidden optimization approaches.

Diversity of Origin:

- Our members come from a range of backgrounds, including German, Romanian, Chinese, Peruvian, Canadian origins.

Diversity of Experience:

- We focus on recruiting first- and second-year students based on their curiosity about HPC and motivation, rather than prior knowledge.

Competing Team



Yannik Rüfenacht (Team Captain)
Major: Computational Science & Engineering
Interests: CFD, Climate Simulation
Working on: Exascale Climate Emulator



Leonard Schmidt
Major: Computer Science
Interests: Deep Learning, AI Systems
Working on: MLPerf, Reproducibility, Mystery



Elena Aschmann
Major: Computer Science
Interests: Computer Systems, Parallel Programming
Working on: SST



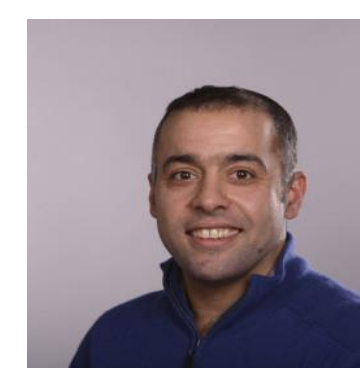
Pierre Pang
Major: Computational Science & Engineering
Interests: ML, Stochastics, PDEs
Working on: Exascale Climate Emulator



Nick Schaller
Major: Computational Science & Engineering
Interests: Parallel Programming
Working on: SST



Ioana Popa
Major: Computational Science & Engineering
Interests: Numerical Methods, Optimisation, Computational Chemistry
Working on: Reproducibility, Benchmarking



Hussein Harake (Main Advisor)
Senior Systems Engineer at CSCS



Prof. Torsten Hoefler (Advisor)
Chief ML-Architect at CSCS
Professor for Computer Science at ETHZ

Hardware & Software Configuration

Hardware

Our cluster, named *Sbrinz* after a traditional Swiss cheese variety, is designed for high performance and energy efficiency with its 4 nodes. Its architecture consists of the following components:

Nodes:

- 4 nodes, each with 2 x 96 Core AMD EPYC Turin 9355 2.4GHz - delivering high compute density and efficiency.

Memory:

- 768GB DDR5 per node – high capacity and bandwidth for CPU workloads.

GPUs:

- 16 NVIDIA H200 NVL (4 per node) – next-generation accelerators with large HBM memory and performance for AI/HPC.
- High-speed NVLink between GPUs – efficient multi-GPU scaling.

Storage:

- 3.84 TB per node NFS shared NVMe over RDMA – designed for data-intensive workloads.
- 480 GB M.2 on the headnode

InfiniBand:

- 400Gb/s NDR with ConnectX-7 – ultra-fast, low-latency communication between nodes.

Ethernet:

- 10Gb/s switch – general-purpose network connectivity.

Software

OS:

- Rocky Linux* – widely used in HPC, ensures compatibility with broad range of HPC applications.

Cluster Management:

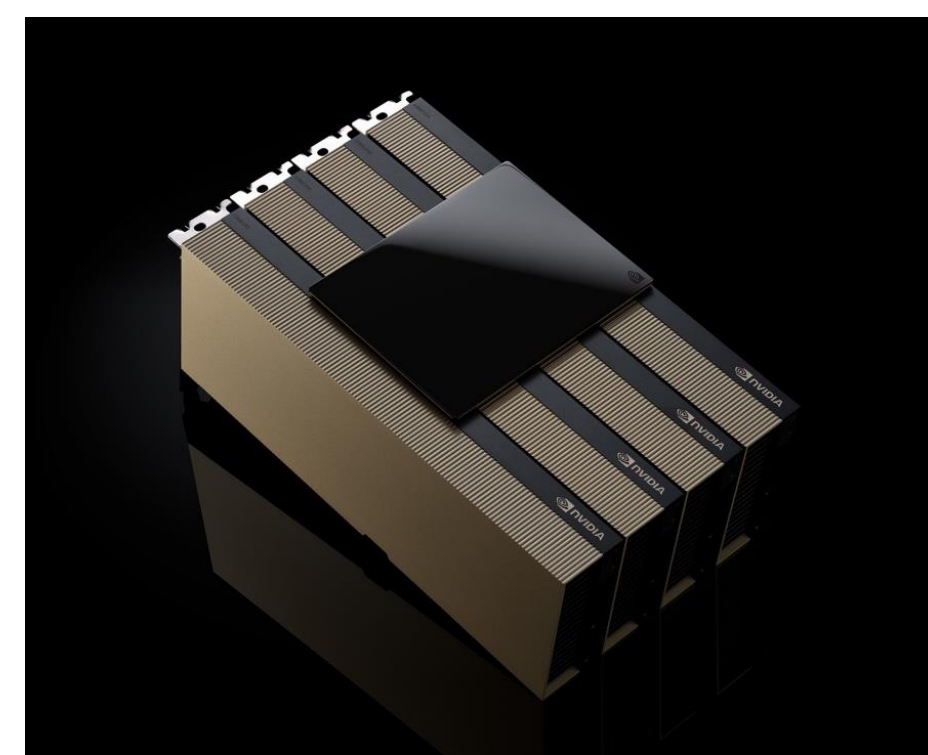
- Bright Cluster Manager* (setup: one master and compute node, four compute nodes) – streamlines cluster administration and workload, very little overhead, significantly improves efficiency in managing resources.

Package Manager:

- Spack* – handles software dependencies and environment, allows us to flexibly manage compilers and libraries

Compilers:

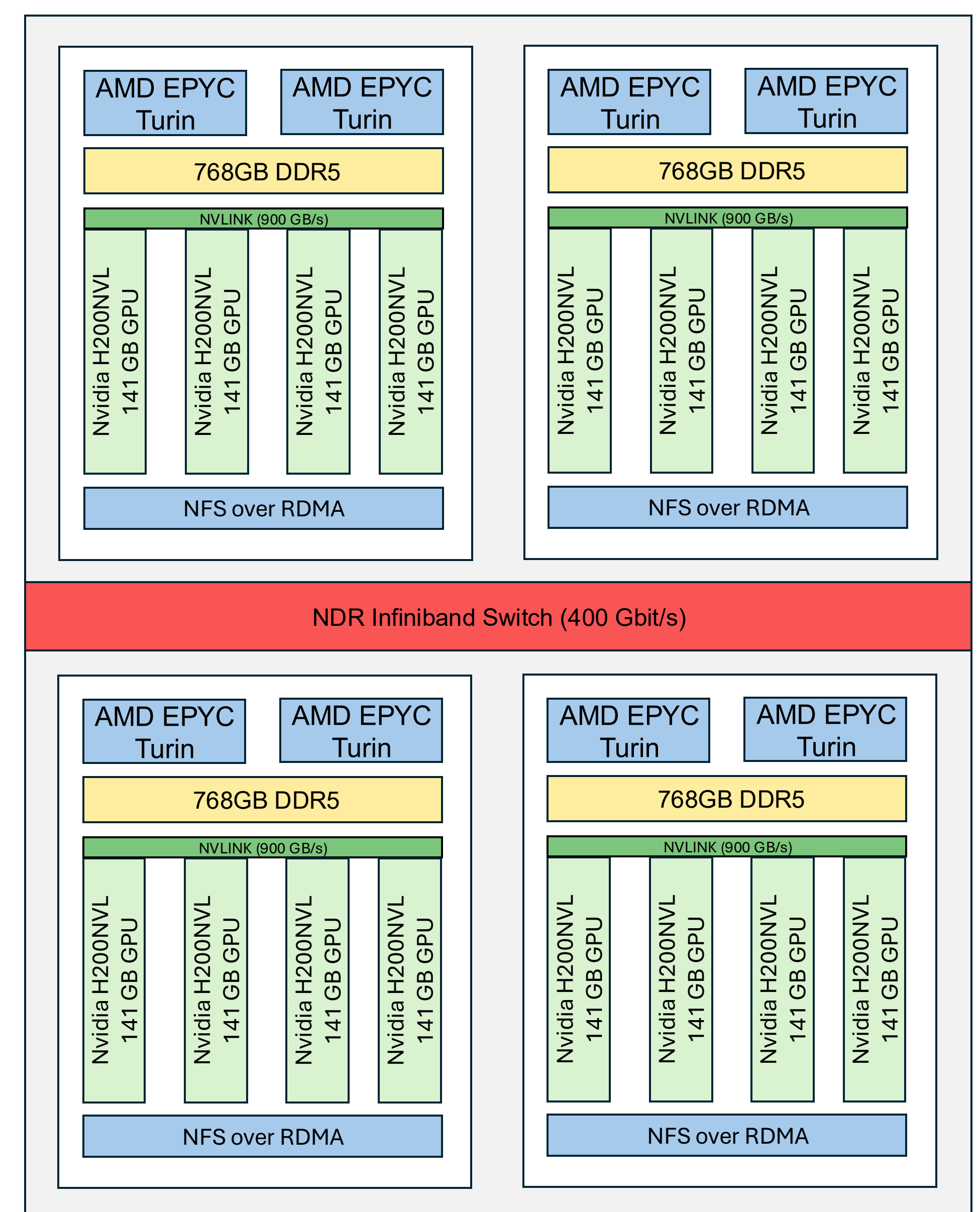
- Among others: GNU, NVHPC
- Experiment with many to compare performance on our architecture.



NVIDIA H200 NVL



Sbrinz at ISC24



Cluster Hardware Topology

Preparation & Optimization

General Preparation

- Weekly meetings to discuss progress and address questions or issues
- Regular presentations from team alumni
- Professionally organized workshops by CSCS experts
- Teams of two, pairing new and experienced members, to foster diverse perspectives and ensure a valuable learning experience
- Computational Science and Engineering and Physics students provide scientific context to CS students

Exascale Climate Emulator

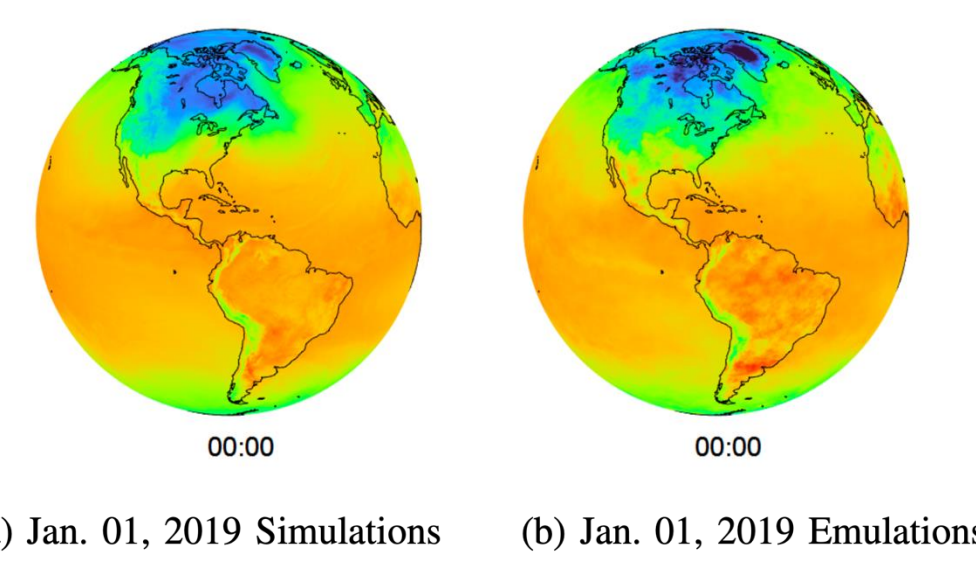
- Build ECE across compilers/MPI/GPU stacks; enable mixed-precision modes
- Profile SHT and linear-algebra kernels; tune tiling, fusion, occupancy
- Scale collectives and tasking across nodes; overlap compute-communication
- Improve parallel I/O for HDF5/NetCDF climate data
- Validate statistical fidelity and reproducibility; automation

SST

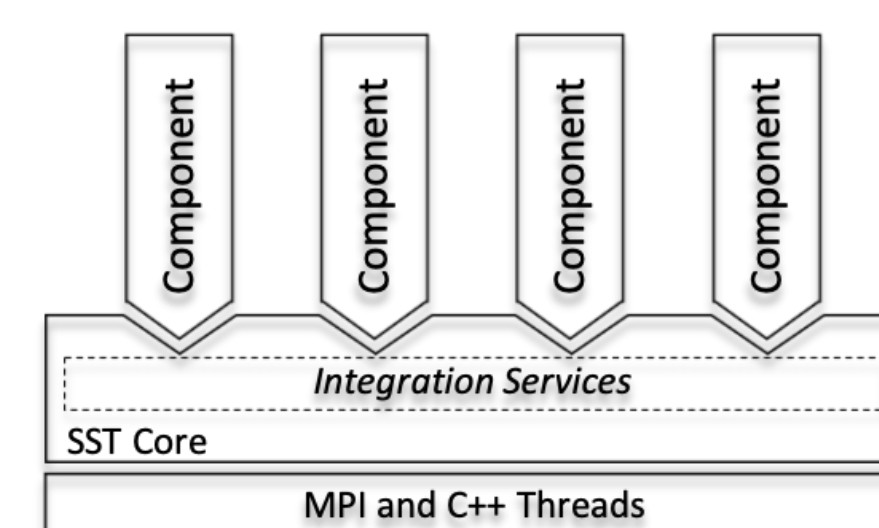
- Build with different compiler/MPI combinations
- Profile initialization and runtime behaviour
- Analyse performance of selected SST elements (e.g. Ariel, Miranda, MemHierarchy)
- Scale benchmarks across varying node counts
- Investigate impact of simulation configuration parameters
- Validate correctness across different platforms
- Automate builds and runs with Spack

Reproducibility Challenge

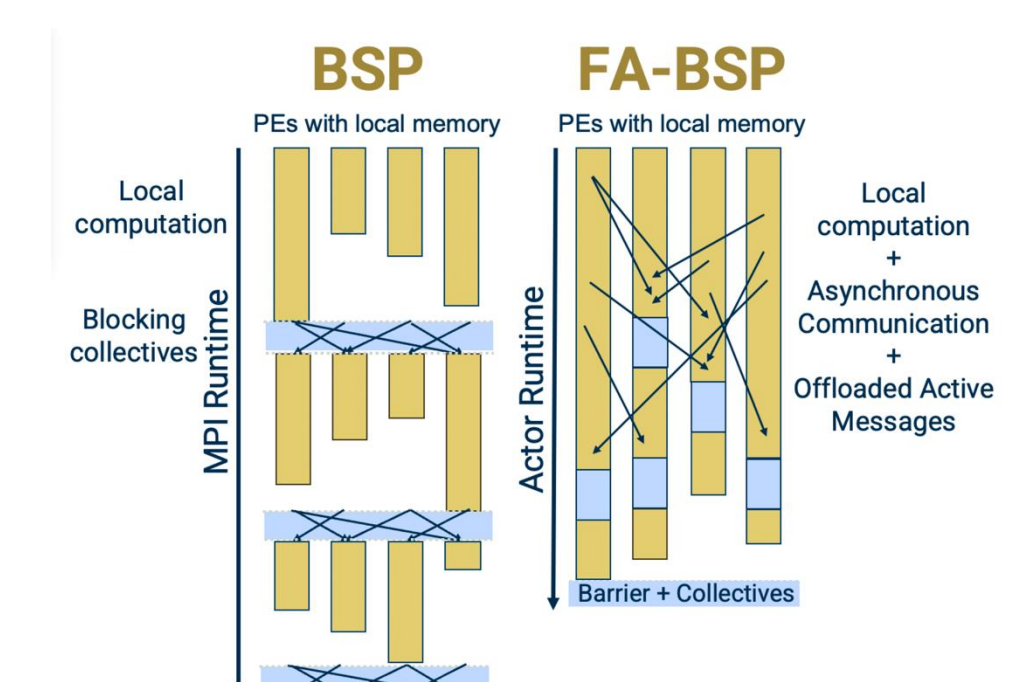
- Distributed IMM with async FA-BSP, no graph replication.
- Contribution: New GenerateRR & SelectSeeds kernels, scalable & memory-efficient.
- Results: Up to 29.6x faster; first on Twitter-2010 (41M nodes, 1.4B edges).
- Our Setup: Ran Actor IMM on cluster (HCLib + OpenSHMEM, SNAP datasets).
- Reproduction: Replicated scaling plots, verified speedups & influence quality.



Global temperature: simulation vs. ECE emulation (Jan 1, 2019)



Example SST scheme (sst-simulator.org)



Basic structure of bulk synchronous parallel (BSP) and fine grained asynchronous bulk synchronous parallel (FA-BSP)

High-Performance Linpack (HPL)

- Use the latest pre-optimized NVIDIA container
- Choose problem size that fully utilizes available GPU memory
- Determine optimal blocking size so that matrix blocks fit within the cache, improving data locality
- Leverage SLURM's built-in performance flags
- Optimize NUMA locality by applying CPU and GPU masking to minimize data transfer latencies
- Manually clear caches before running HPL to avoid cache thrashing overhead

MLPerf Inference

- Tune batch sizes to maximize throughput while meeting accuracy requirements.
- Utilize NVIDIA-optimized models tailored for the Hopper architecture.
- Leverage MLCFlow automation, which supports NVIDIA's models
- Employ the MLPerf 5.1 release.
- Focus on multi-GPU submission
- implementations

Mystery Application

- Prepare different ML frameworks, compilers and MPI implementations to quickly adapt to the given application



Team RACKlette benchmarking during SC24

Our Sponsors



CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre



HASLERSTIFTUNG

