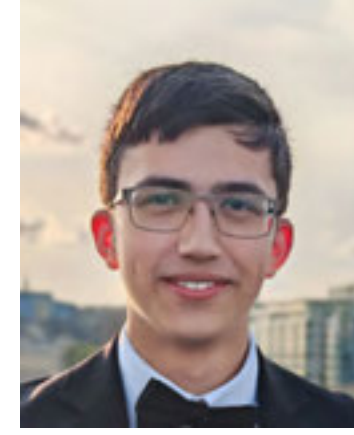


About Our Team

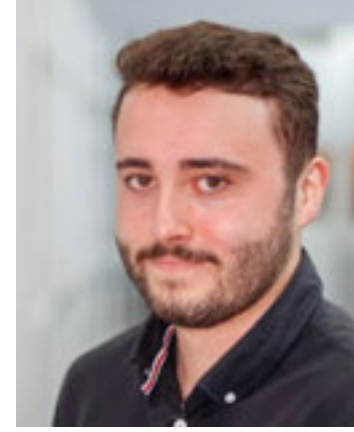
Our Student Cluster Competition team, hosted by the Bristol Centre for Supercomputing has members from across the UK, with students from Imperial College London, the University of Warwick, the University of Durham and the University of Bristol. We have a strong and varied background, with some of our members having no Student Cluster experience and others having won IndySCC24, competed in ISC25 and taken part in CIUK. What's more, four of our team members hold multiple nationalities, including Australia, America, Canada and Germany.



Joseph Glynn
Computer Systems Engineering @ Warwick
HPC Experience: New
Interests: **Experimental cooking / Gardening**



Katie Pambakian
Computer Science @ Bristol University
HPC Experience: New
Interests: **Cycling**



Tarrin Curtis
Computer Science @ Cardiff University
HPC Experience: New
Interests: **Climbing**



Quentin Duff
Electrical Engineering @ Imperial College London
HPC Experience: New
Interests: **Running / Quadcopter Racing**



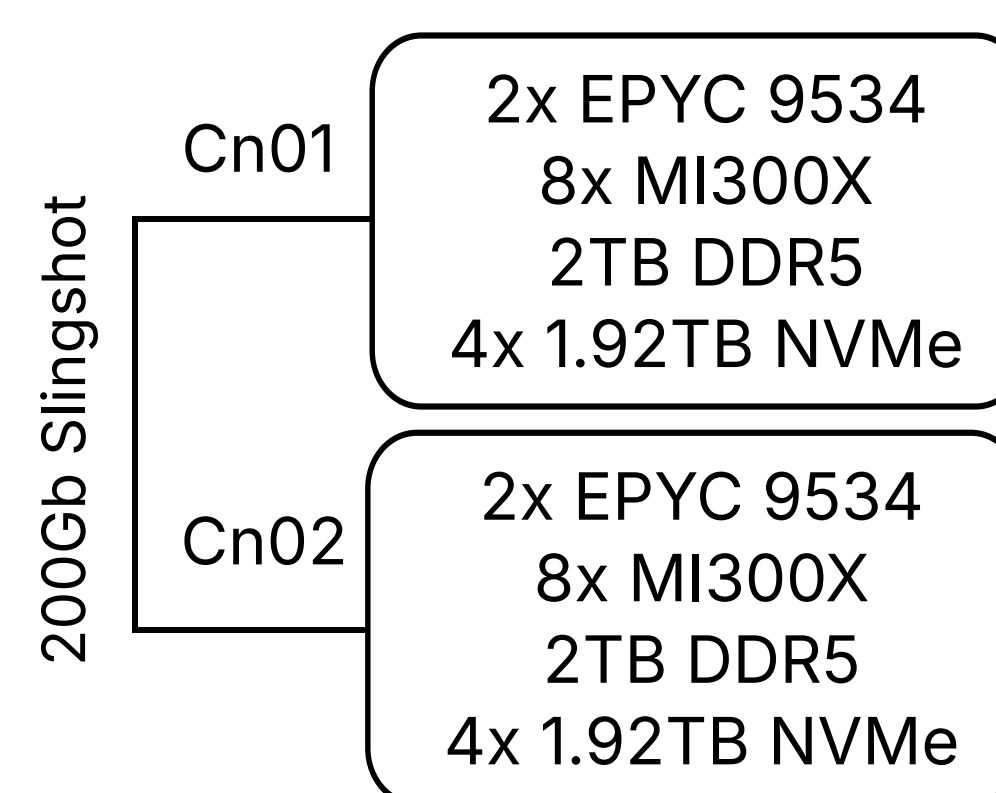
Toby Davis
Computer Science @ University of Durham
HPC Experience: **IndySCC24, ISC25, CIUK**
Interests: **Mountain Biking / Photography**



Tom Shtasel
Computer Science @ Imperial College London
HPC Experience: **ISC25**
Interests: **Metro Maps**

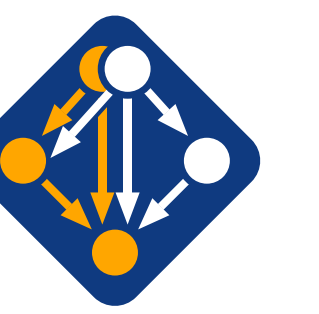
Hardware

- **AMD EPYC Turin CPUs** for high core counts and library/application support
- **AMD MI300X GPUs** to balance FP64 perf (HPL) with mixed precision (ECE, MLPerf)
- **HPE Slingshot** for high bandwidth, low latency communication
- **NVME SSDs** for parallel, high-bandwidth storage
- **Tune CPU frequency, GPU power and fan speeds** per application in advance
- Targets just over power limit and downclock below 10kW for optimal FLOPS/W



Software

- **Ansible scripts** for fast, scalable provisioning ensuring reproducibility
- **SLURM** for efficient node, resource scheduling
- **NFS** for consistent centralised storage across all nodes
- **BeeGFS** for parallel IO bandwidth
- **Rocky Linux v9.6** for performance and stability, as well as better NFS support
- **Prometheus + Grafana** for monitoring and visualisation of cluster performance
- **Environment Modules** for customisable build environments and easy library changes
- **Spack** for efficient, reproducible builds of both programs and dependencies
- **Internal build tools + documentation** for an efficient workflow and accessible team documentation



Preparation

- **Internal competitions** using past applications on a training cluster provided by BriCS
- **Provisioned** our cluster's software infrastructure (Slurm, environment modules, drivers, NFS etc.)
- **Optimised at all levels** (BIOS settings, kernel-level tuning, network parameters)
- **Microbenchmarked** the system (STREAM, OSU), and stress-tested (HPL) for power/performance expectations
- **Time windows** (with backup time) for pairs to execute each application in the 48 hours
- **Regular meetings** both online and in person, ensuring consistent teamwork in the lead up to the competition
- Attended a multi-day **in-person workshop** with BriCS staff, hardware vendors, and other industry professionals
- **Backup, document, and script everything** we can - learning from experience



Optimisation Strategy

Each application is split into three phases: introductory, exploratory, optimisation.

Introductory: Assign two people per application. Compile and run the program to get an initial visualisation of the results. This provides a good baseline to compare against and allows us to verify the program is working as expected.

Exploration: Research the mathematical background and domain. Understand possible bottlenecks and benchmark performance impacts. Use internal build tools for automation and reproducibility.

Optimisation: Profile the program. Applying insights from exploration, tune program options, compiler and library choice, compiler flags and system-level parameters in that order. NUMA affinity, OpenMP/MPI balancing and GPU communication patterns are also important considerations.

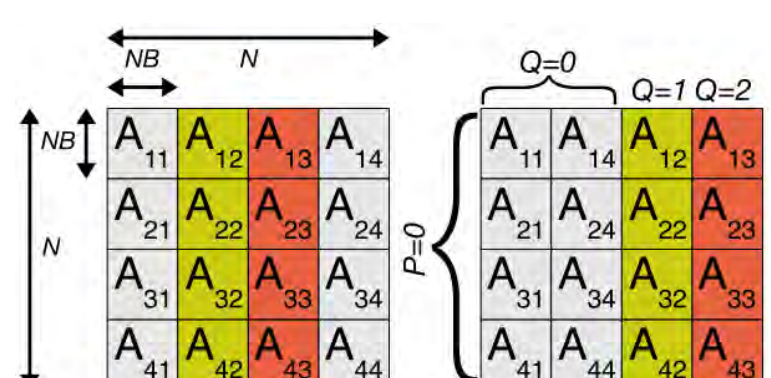
Support

Our group is supported by our mentor, Joseph Moore, together with postgraduate researchers in HPC. Our primary computing system is provided through the generous sponsorship of HPE, while cloud and on-prem resources, funded by BriCS, are dedicated to training activities and hands-on skill development.



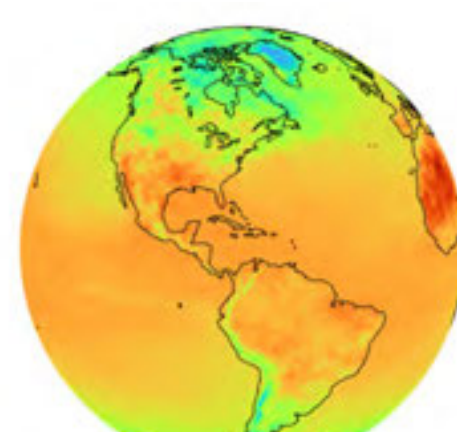
HPL / HPL-MxP/ MLPerf

- Parameter sweeps, scripts and heuristics to find the optimal parameters
- Use vendor-provided binaries for optimized GPU implementations
- Tune MPI and network configuration to minimise communication bottleneck and maximise bandwidth



Exascale Climate Emulator

- Tune ParSEC runtime
- Efficiently map dependency graph to all our GPUs
- Minimise cross-node transfers
- Investigate MPI rank/thread distribution + placement
- Experiment with OpenMP:MPI ratio



Structural Simulation Toolkit

- Optimise partitioning
- Put components connected by low latency links on same rank
- then balance load across nodes
- Investigate MPI rank/thread distribution + placement



Mystery Application

1. Read the documentation
2. Compile with standard, stable libraries such as GCC, OpenMPI and OpenBLAS
3. Start tuning program inputs and options; compiler and library selection and compiler flags.
4. Simultaneously visualise the output.

