Durham University

CIUK Student Cluster Competition Challenge
Durham University Challenge

Karen Bower Wanqing Tu Tobias Weinzierl

October 13, 2025

1 Introduction

The purpose of this CIUK student cluster competition challenge is to encourage students to explore the interplay between bandwidth and compute bounds. By combining theory with hands-on activities, students will showcase a firm understanding of how the bandwidth and compute capabilities of a compute node in a cluster perform, as compared to the theoretical specifications published by vendors. This will motivate students to improve the source code, i.e., the first steps towards performance engineering.

2 The machines

We will use nodes from the DINE and DINE2 clusters in the HPC Hardware Lab @Durham, which are part of the COSMA HPC facility hosted in Durham.

- COSMA documentation
- DINE
- DINE2

Access to the machines will be remote. In order to access the hardware lab you will each need to follow the online instructions to register for DiRAC SAFE and then request a machine account on COSMA using project code do022.

Once your account has been created, login to Cosma8 (login8.cosma.dur.ac.uk and submit jobs to DINE/DINE2 from there using the Slurm queues ciuk-dine and ciuk-dine2. You will need to specify "#SBATCH -A do022" in your jobs.

You can log in and make preparations as soon as your account is created. The job submission queues will open on Wednesday 15th October at 3pm and you will be able to submit jobs for ONE WEEK after that time. Jobs that are submitted within that time but have not completed will still run. The maximum duration for a job is 1 hour.

To help you share data within your team, you will have access to a shared directory `/cosma/home/do022/shared/your_team_name`. We recommend groups to rely on an external service (e.g. github) to archive, share and version control their data. If you use shared directories other than above, be aware that other groups might see your data. Please note that all submitted material should list your team name and all user names associated with it.

3 Task 1: Likwid

Familiarise yourself with LIKWID and its application `likwid-bench`. Find out what the maximum compute capability of each node and its maximal bandwidth are. Measure the throughput and MFLOPS/s you obtain with the benchmark at each node and compare the results to develop your insights.

4 Task 2: STREAM Triad

The STREAM Triad is a kernel tool for measuring a computer's memory bandwidth. It is a vector operation that the benchmark uses to assess a system's efficiency in moving data between the CPU and main memory. Download, compile and run STREAM Triad. You will vary problem sizes to find out the achievable throughput for various problem sizes. This might require you to alter the source code and re-compile. Compare the resulting data with the results from Task 1 to develop further insights. Please test for the problem sizes within the appropriate range.

5 Submission

Each team is required to submit a four minute mp4 or avi video to YouTube. In your video submissions, you should summarise your observations and insights. Please share your YouTube links with us.

6 Scoring

Description of findings about node capability with `likwid-bench` (node 1)	20
Description of findings about node capability with `likwid-bench` (node 2)	20
Critical evaluation of node 1 vs. node 2	10
Explanation of observed behaviour with STREAM Triad (node 1)	10
Explanation of observed behaviour with STREAM Triad (node 2)	10
Description of findings comparing Triad to LIKWID benchmarks	10
Best-case throughput obtained with STREAM Triad	10
Time efficiency in using the testbed	10
Sum	**100**

7 Hints

- All tools/benchmarks are open source. You might want to install them somewhere locally and play around with them, before you hit the queues on Judgement Day. You can also work with your own installation on the test node if you prefer this, but you have to document what exactly you've been doing.
- Familiarise with SLURM and use the test node exclusively.
- Be careful with the chosen software stack (compilers) and document all optimisation settings.
- You might want to look up what the Roofline model is (cmp. SHAREing training links below) and use this as a framework to argue about your results.

Links & Material

- HPC Hardware Lab @ Durham: https://durham.readthedocs.io/en/latest/hardwarelab
- LIKWID - Like I Knew What I Am Doing: https://rrze-hpc.github.io/likwid/Doxygen/index.html
- STREAM: Sustainable Memory Bandwidth in High Performance Computers: https://www.cs.virginia.edu/stream
- SHAREing training: https://shareing-dri.github.io/training
- DiRAC Training Academy: https://training-academy.dirac.ac.uk