alcesflight

Technical White Paper

Delivering an HPC Service Under Real-World Constraints

Lessons from the CIUK Student Cluster Challenge

Cristin Merritt, Alces Flight

December, 2025

Table of Contents

Introduction

The CIUK Student Cluster Challenge (CIUK SCC) has evolved over the years to reflect a realistic view of supercomputing, exposing students to the technical, operational, and user-facing perspectives of HPC systems and services. In 2025, following an online preliminary round focused on the effective use of existing local and national HPC services, Alces Flight’s contribution to CIUK SCC deliberately shifted emphasis towards the end-to-end simulation of creating an HPC service for a new user community.

This challenge was structured to reflect the real lifecycle of a modern HPC service. In the first stage, teams designed and deployed a functional four-node HPC cluster from bare metal, validated system integrity, and demonstrated multi-node capabilities through application benchmarking and batch-scheduled workloads. The second stage focused on operating and extending the service, introducing multi-user access, workload policies, resource constraints, application enablement, security controls, and support for diverse domains. In the final stage, teams drove adoption and usability by producing user documentation, onboarding external users, and communicating system capabilities and design choices through their presentations.

Across all stages, teams worked under conditions that mirror real-world HPC delivery: limited time, unfamiliar hardware, evolving requirements, and the need to balance performance, reliability, security, and the user experience. The technical environment incorporated industry-standard components, including Rocky Enterprise Linux, diskless node provisioning, Slurm workload management, MPI-based parallel applications, container workflows, and interactive user graphical desktop sessions.

This document examines the design rationale, reference architecture, and operational discipline demonstrated by the students during this simulation. This competition highlighted how a compact, education-focused environment mirrors many of the same architectural decisions, trade-offs, and risks encountered in delivering and operating a production HPC service. It may be a helpful training resource for teams delivering an HPC cluster capability.

Designing for Operational Reality

Small-scale HPC environments are often where operational risk is at its highest. Rapid deployments, constrained resources, heterogeneous users, and a limited margin for error quickly put pressure on any architectural decisions made. The Alces Flight challenge intentionally recreated these exact conditions in a controlled yet realistic setting.

As an industry leader in HPC and AI cluster integration, management, and toolsets, Alces Flight contributes to CIUK SCC challenges that emphasise service delivery over static configuration. For 2025, this meant building on their experience with both National and Regional UK HPC facilities and challenging the students to take on the full lifecycle of platform ownership. Teams were required to not only deploy a functioning system, but to operate it as a shared service - supporting users, enforcing policy, managing change, and justifying decisions made under pressure.

These core principles guided the simulation design:

- **Service outcomes over technical competency:** Demonstrable use and reliability were the measures of success, not just the configuration.
- **Progressive responsibility:** Teams advanced from bringing the system online, through multi-user operation, to workload onboarding.
- **Constraints as design drivers:** Limited time, unfamiliar hardware, and incomplete information were provided and reflective of real-world environments.
- **Decision-making and trade-offs:** Optional guidance was available (sometimes at a scoring cost), challenging teams to consider their own judgment, prioritisation, and risk awareness.

Together, these principles ensured the challenge would assess the team's maturity in thinking through how the service would operate alongside their technical capability to bring it into production.

Building and Operating the Service

Teams deployed and operated a four-node HPC cluster from bare metal using industry-standard technologies, including Rocky Enterprise Linux, diskless node provisioning, and the Slurm workload manager. Teams achieved their first success through delivering a functioning, multi-user platform within time and resource constraints.

To support the rapid, reproducible deployment while preserving operational transparency, teams built their clusters using ***GHPC*** - an open source HPC/AI stack deployment developed by Alces Flight. Built on Enterprise-grade tools and methodologies, GHPC is designed to adapt, extend, and customise to specific use cases. Within the challenge, GHPC provided a structured foundation for cluster deployment while still requiring teams to understand, configure, and operate every layer of the service built.

The Initial validation focused on stability and usability rather than on system configuration and performance. Teams were required to bring all compute nodes online, configure Slurm, and demonstrate execution of workloads as standard users. Early tasks included running batch jobs, containerised workloads, and graphical desktop sessions, ensuring the platform supported multiple access and execution models commonly expected by today's HPC users. After establishing baseline functionality, the challenge shifted towards operating the cluster as a shared service. Teams introduced multiple users, configured user group policies and access controls, and implemented policies using Slurm partitions, priorities, and resource limits. These activities required users to think beyond the basic level of administrative access and consider the user's need for a fair, predictable service - a primary concern of any production environment.

Application enablement accounted for a substantial portion of the operational workload. Teams installed and executed a diverse range of applications spanning life sciences, computational fluid dynamics (CFD), machine learning, and embarrassingly parallel workloads. Managing compilers, supporting both native and container-packaged workflows, and ensuring consistent application exposure were all requirements for making this possible. In some cases, teams installed the same application using alternative compilers, reinforcing considerations around performance tuning, portability, maintenance, and operational consistency.

The main priority throughout was the user experience. Teams enabled secure access methods, graphical workflows, data movement between client systems and the cluster, and user-facing documentation to support onboarding. More advanced tasks required consideration of change management scenarios, such as configuring multi-factor authentication (MFA), scaling compute resources, and modifying scheduling policies on a live system. Often, these tasks arise as a service evolves.

Most importantly, scoring prioritised demonstration of service operation over declared capability. To earn points, real users had to complete tasks on the system. To achieve this, teams needed to host competitors, attendees, and CIUK event organisers on their services. This exercise aimed to reinforce the principle that an HPC system's value comes through reliable, repeatable use, and that service delivery - not configuration alone - is the ultimate measure of success.

What This Simulation Demonstrates

Although delivered in an educational context, the challenge design mirrors issues directly applicable to real HPC projects. Architectural risk can be exposed quickly in small clusters and cluster environments, making them adequate proving grounds for service models, policies, and workflows.

Demonstrable outcomes include:

- **Operational realism:** Participants encountered the same types of issues observed in production systems: misconfigurations, user errors, scaling decisions, and technical trade-offs.
- **Service-oriented thinking:** Teams learned to prioritise users as stakeholders, with documentation, onboarding, and communication becoming integral parts of service success.
- **Understanding risk in decision-making:** Time pressure and scoring make-up forced teams to balance speed, correctness, and sustainability - mirroring real delivery trade-offs.
- **Broad applicability:** This model maps directly to live clusters in operation, pilot systems, research platforms, and training environments where expectations are high despite limited resources. The Enterprise-grade software tools used in this simulation are identical to those used in National and Regional HPC facilities across the UK, ensuring that the skills gained are relevant and transferable to students' professional careers.

By emphasising delivery and operation over isolated technical tasks, this challenge demonstrates how realistic, service-led HPC best practices can be taught, exercised, and evaluated at a small scale.

Developing the next generation of professionals

These experiences and lessons outlined within this document directly reflect how Alces Flight approaches real-world HPC systems. If you are exploring how service-led design, operational reliability, and user-focused delivery apply to your HPC or research computing initiatives, we would welcome an opportunity to speak with you.

Email: info@alces-flight.com
Website: www.alces-flight.com

GHPC OpenFlight: https://ghpc.openflighthpc.org/

Technical Addendum:

Architecture and Operational Considerations

This addendum provides additional technical information and context for HPC practitioners seeking deeper insight into the architecture, operational mechanisms, and decision points during the challenge.

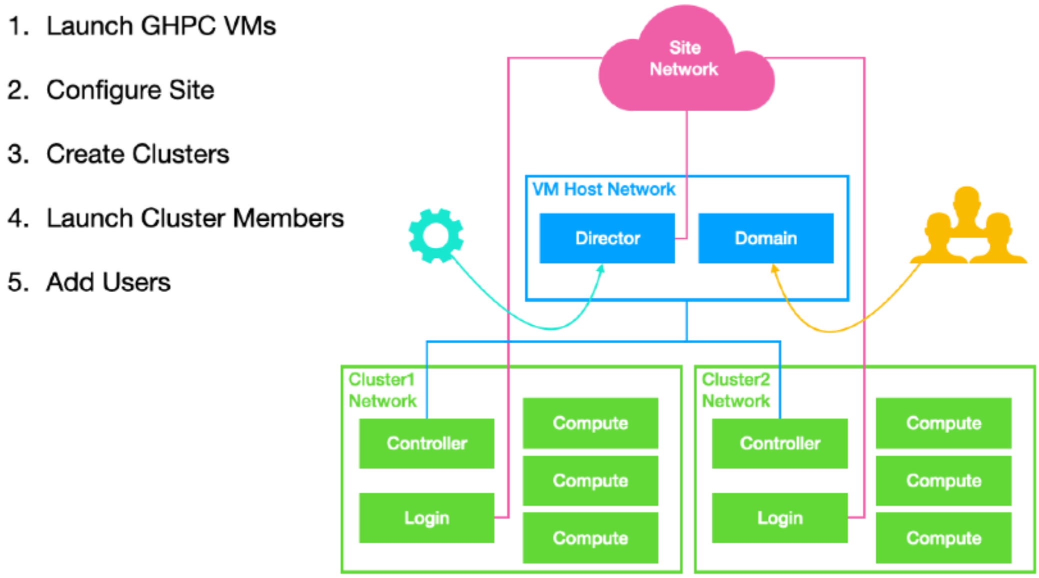
The focus is on service delivery and operational realism, rather than isolated performance optimisation or synthetic benchmarking.

Reference Architecture Overview

Each team deployed and operated a compact HPC environment designed to mirror standard small-to-medium scale production systems:

- Four-node cluster created utilising on-site hardware
- Network segmentation delivers a clear security separation of compute nodes and roles
- Diskless provisioning model for compute nodes using NFS-root technologies
- Standard open-source Rocky Enterprise Linux as a base operating system
- Slurm workload manager for job scheduling, accounting, and policy enforcement

To enable rapid, transparent deployment, the cluster build utilised GHPC, an open-source HPC/AI stack deployment framework developed by Alces Flight. GHPC provided a reproducible baseline while preserving complete visibility and control over system configuration, allowing teams to customise, extend, and troubleshoot their environments as challenge requirements evolved.



1. Launch GHPC VMs
2. Configure Site
3. Create Clusters
4. Launch Cluster Members
5. Add Users

Core Operation Capabilities

Once deployed, each cluster operation was a shared, multi-user service. Core capabilities exercised during the challenge included:

- Multi-user authentication and access control
- Job scheduling via Slurm, including partitions, priorities, and resource limits
- Support for batch, interactive, containerised, and graphical workloads
- Environment Modules for consistent application exposure and user experience

Demonstration and assessment of operational success was through use, with workloads requiring standard users to execute rather than administrators.

System Validation

System validation was intentionally integrated into the regular service operation rather than treated as a standalone benchmarking exercise. Validation of clusters happened through:

- Successful scheduling and execution of jobs across multiple nodes
- Concurrent use by multiple users
- Execution of diverse workload types via the scheduler
- Live demonstrations carried out by external users, including fellow competitors, CIUK attendees, and event organizers

This approach prioritised service correctness, usability, and reliability over synthetic performance metrics.

Application Enablement

Teams enabled a range of applications representative of real HPC usage, including:

- Native and containerised application workflows
- Domain-specific workloads spanning life sciences, CFD, machine learning, and embarrassingly parallel jobs
- Exposure to application build and deployment considerations across different software stacks

Emphasis was placed on repeatability, maintainability, and clarity of exposure, reflecting production expectations over one-off builds.

Operational Stressors

The challenge environment intentionally compressed common operational stressors into a short timeframe:

- Time-bound deployment and configuration
- Incomplete or changing requirements

- Real users executing real workloads
- Configuration changes applied to live systems

These conditions required teams to manage risk, recover from errors, and make pragmatic trade-offs, behaviours directly transferable to production environments.

Why This Matters in Production Environments

Small-scale HPC systems often introduce disproportionate risk due to rapid deployment, limited staffing, and diverse user needs. This challenge recreates these conditions in a controlled environment, exposing the same architectural and operational decision points as those faced by production teams.

For departmental leaders, this demonstrates how open, reproducible platforms can accelerate delivery without sacrificing control. For team leads, it provides a concrete reference for onboarding, training, and assessing operational readiness in environments where reliability, usability, and stability matter as much as raw performance.

Acknowledgements and Warranties

THIS MATERIAL IS DESIGNED TO ASSIST CAPABLE USERS TO CUSTOMISE OUR SERVICE SUBSCRIPTIONS AND SOFTWARE PRODUCTS, EXTENDING THEM TO PERFORM OPERATIONS BEYOND ORIGINAL DESIGN PARAMETERS. EXAMPLES OF POSSIBLE EXTENSIONS ARE PROVIDED TO HELP CUSTOMERS REALISE FUTURE POTENTIAL OF THE SOFTWARE AND ARE PROVIDED WITHOUT WARRANTY AND ARE NOT SUPPORTED OR GUARANTEED BY ALCES FLIGHT LTD OR OUR LOCAL SOFTWARE RESELLERS. THIS DOCUMENTATION HAS BEEN CREATED TO INFORM AND SHARE KNOWLEDGE WITH CUSTOMERS FOR REFERENCE PURPOSES ONLY; SOFTWARE FEATURES AND PERFORMANCE ARE NOT GUARANTEED. THIS DOCUMENTATION IS NOT DESIGNED AS A STAND-ALONE TRAINING TOOL – EXAMPLE COMMANDS AND SYNTAX ARE INTENDED TO DEMONSTRATE FUNCTIONALITY IN A TRAINING ENVIRONMENT AND MAY CAUSE DATA LOSS IF EXECUTED ON LIVE SYSTEMS. REMEMBER: ALWAYS TAKE BACKUPS OF ANY VALUABLE DATA. THIS DOCUMENTATION IS PROVIDED “AS IS” AND WITHOUT ANY WARRANTY; WITHOUT EVEN THE IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NO COMPUTER SYSTEM CAN BE COMPLETELY SECURE - FOLLOW GOOD SECURITY ADVICE AND ALWAYS HAVE YOUR COMPUTER SYSTEMS CHECKED BY COMPETENT SECURITY PROFESSIONALS BEFORE BEING USED WITH LIVE DATA. PLEASE SEE THE EULA INCLUDED WITH THE PROVIDED SOFTWARE PACKAGES FOR FULL USAGE TERMS AND CONDITIONS. WE WELCOME CUSTOMER FEEDBACK AND SUGGESTIONS FOR FUTURE ENHANCEMENTS - PLEASE VISIT THE COMMUNITY SUPPORT SITE FOR THE RELEVANT OPEN-SOURCE SOFTWARE PROJECT FOR MORE DETAILS.

FOR MORE INFORMATION ON ALCES FLIGHT, PLEASE VISIT: HTTP://WWW.ALCES-FLIGHT.COM/. PLEASE SUPPORT SOFTWARE DEVELOPERS WHEREVER THEY WORK – IF YOU USE OPEN-SOURCE SOFTWARE, PLEASE CONSIDER CONTRIBUTING TO THE MAINTAINING ORGANISATION OR PROJECT, AND CREDITING THEIR WORK AS PART OF YOUR RESEARCH AND PUBLICATIONS.