

Day 2 Introduction

Kurt Lust
LUMI User Support Team (LUST)
University of Antwerp

November 2026

Lessons from day 1

LUMI

- LUMI architecture
  - Both CPU and GPU nodes are very hierarchical
  - Proximity of resources in a node is important
    - Proper process and thread distribution needed
  - Powerful interconnect, but different from "the standard" and new
    - Some libraries are different from standard distributions
- HPE Cray Programming Environment
  - Essential part of the system software on an HPC cluster
  - Usually no ABI standard for packages: OpenMP runtime, MPI, …
  - Therefore bringing in binaries not always straightforward
  - Restricted Linux on the compute nodes
    - Improved scalability for large applications
    - But it also implies some software may not work

Lessons from day 1 (2)

LUMI

- Accessing LUMI
  - Not different from most HPC clusters
  - Organisation in users and projects if fairly typical for tier-1/tier-0 systems
  - But the distributed nature of allocation management is very atypical and may be confusing
- Data transfer to/from LUMI
  - Users experience it as a pain point
  - Limited protocols
  - Designers of LUMI focused strongly on the object storage as an intermediate station

Lessons from day 1 (3)

LUMI

- Modules: LMOD
  - Installed and available modules
  - Search commands for modules
- Application software on LUMI: Unconventional for many
  - Taking diversity into account
  - Need to react quickly to system updates
  - Supportability by a small central support staff that may not be familiar with your research domain
  - Evolution towards custom environments
  - EasyBuild (main tool) and Spack (secondary tool) + containers (this afternoon)

Lessons from day 1 (4)

LUMI

- LUMI support
  - Distributed across multiple organisations, and this will not improve
  - Don’t expect domain knowledge in your field of research
  - Support is there to support, not to replace you for some tasks. You have to help support also
  - 10 people can’t do miracles and we have to focus on things that are important for more users and cannot be bothered with things that are used only a few times

This morning: Running jobs

LUMI

- Slurm is not a very good resource manager/scheduler but it is the best production quality one we have…
- Slurm introduction
  - No two Slurm systems are configured the same so useful even if you are a Slurm expert!
- Process and thread distribution and binding
  - Not only the work of Slurm
  - Can have a large influence on performance of an application
  - Unfortunately there is no uniformly best way to do it so we cannot preconfigure it for you…
    - And we'll run into Slurm limitations though life is a little easier since the January 2026 system update

Afternoon Theme 1: Data

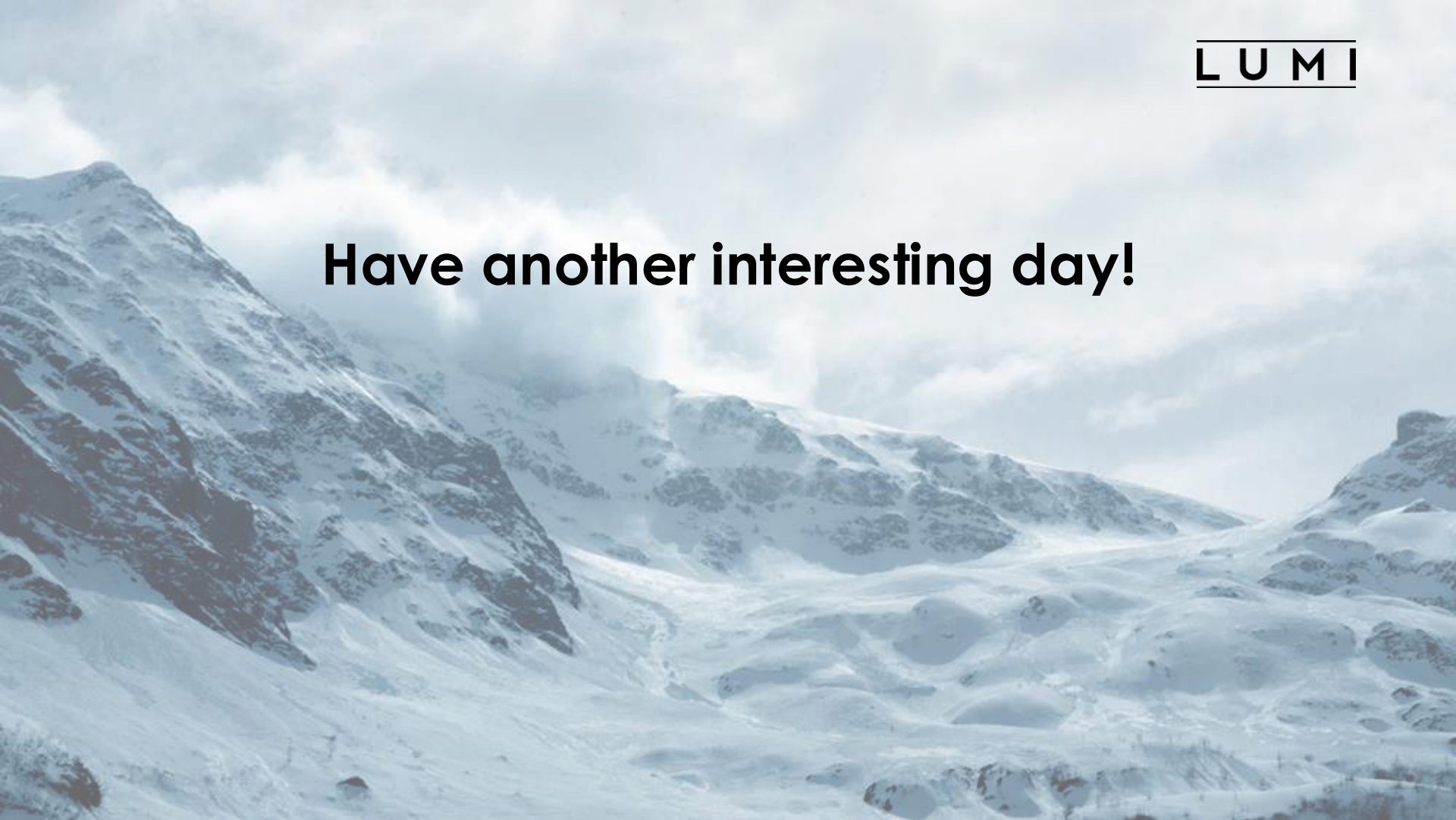
LUMI

- Large parallel file systems can be your friend and your enemy
  - Your friend if you have a good HPC-friendly code and know how to use it…
  - But your enemy in many cases.
  - Basically, some file system parameters don't scale well with the size of the machine so we need to use the file system properly
- Object storage is a very different type of storage
  - Useful for organising data transfers
  - Useful for some programs that can access data directly on object storage
  - Can in principle be used as a backup of your data on Lustre

Afternoon Theme 2: Containers

LUMI

- An important use of containers on LUMI is actually helping the file system deal with HPC-unfriendly software packaging
  - Pitfalls of containers on LUMI
  - Build and extend containers for LUMI, with some focus on AI applications


LUMI
Have another interesting day!