
# LUMI Introduced

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

# What is LUMI?

**… a (pre-)exascale supercomputer …**

- … and not a superfast PC …
- … and not a compute cloud infrastructure…
- … and not a server farm with a fast interconnect and a job scheduler.

- Each of these infrastructures have their own trade-offs
- A supercomputer is
  - Built for scalable parallel applications in the first place
  - A large shared infrastructure with lightweight management
  - Principle: Reduce hardware costs by clever software
  - Built for streaming data through the machine at all levels, not for random access to small bits of data

# What is LUMI? (2)

… a (pre-)exascale supercomputer …

- Part of the EuroHPC ecosystem:
  - 5 "petascale" supercomputers, all operational
  - 3 pre-exascale machines:
    - LUMI, hosted by CSC in Finland
    - Leonardo, hosted by CINECA
    - MareNostrum5, hosted by BSC
  - 2 exascale machines
    - Jupiter at JSC is work-in-progress
  - 4 mid-range systems in procurement or construction
- LUMI is built by a consortium of 11 countries, including Belgium

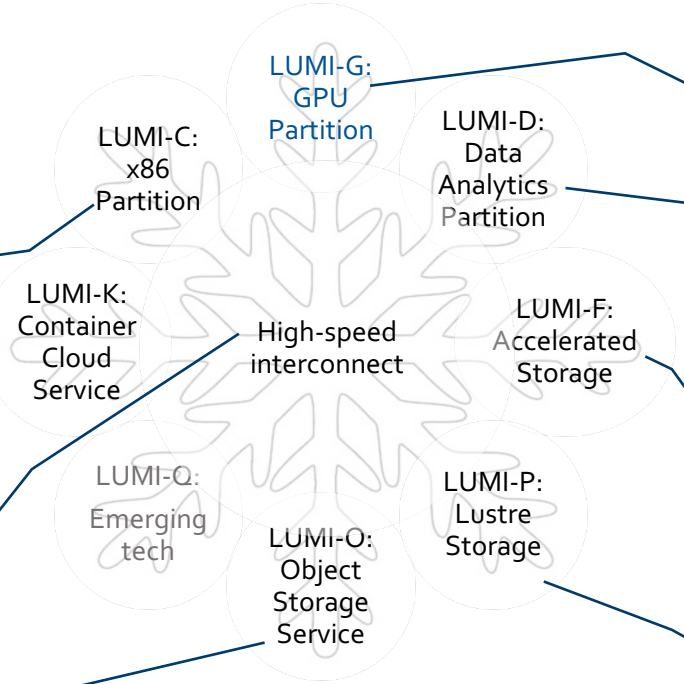
# LUMI, the Queen of the North

*LUMI is a Tier-0* ***GPU-accelerated supercomputer*** *that enables the convergence of* ***high-performance computing***, ***artificial intelligence***, *and* ***high-performance data analytics.***

- Supplementary CPU partition
- ~260,000 AMD EPYC CPU cores

HPE slingshot interconnect, based on libfabric rather then the UCX library used by NVIDIA/Mellanox.

30 PB encrypted object storage (Ceph) for storing, sharing and staging data




Tier-0 GPU partition: over 530 Pflop/s FP64 vector performance powered by AMD Instinct GPUs

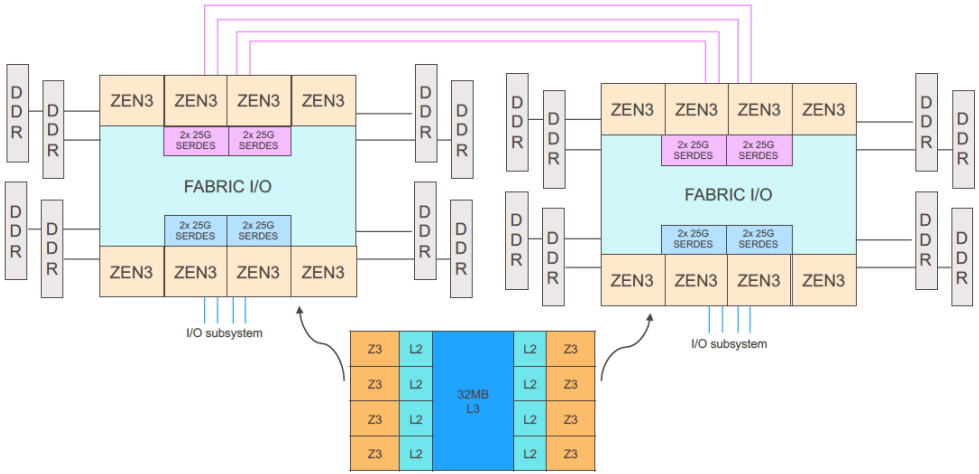
Interactive partition with 4 TB CPU-only and 2 TB NVIDIA GPU nodes for data analytics and visualization

8.5 PiB Flash-based storage layer with extreme I/O bandwidth of 2 TB/s and IOPS capability. Cray ClusterStor E1000.

4 x 18 PiB parallel file system. Cray CusterStor E1000

# LUMI compute node configurations

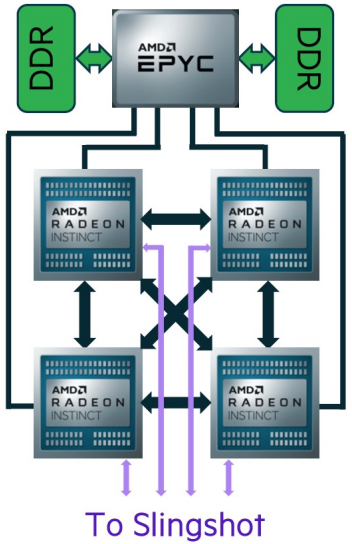
## LUMI-C




2x 64-core AMD Milan processors per node
1888 nodes with 256 GB, 128 with 512 GB and 32 with 1 TB

- But 32 GB reserved
- And 2 sockets, 8 NUMA domains, 16 L3 cache regions…

## LUMI-G (marketing view)

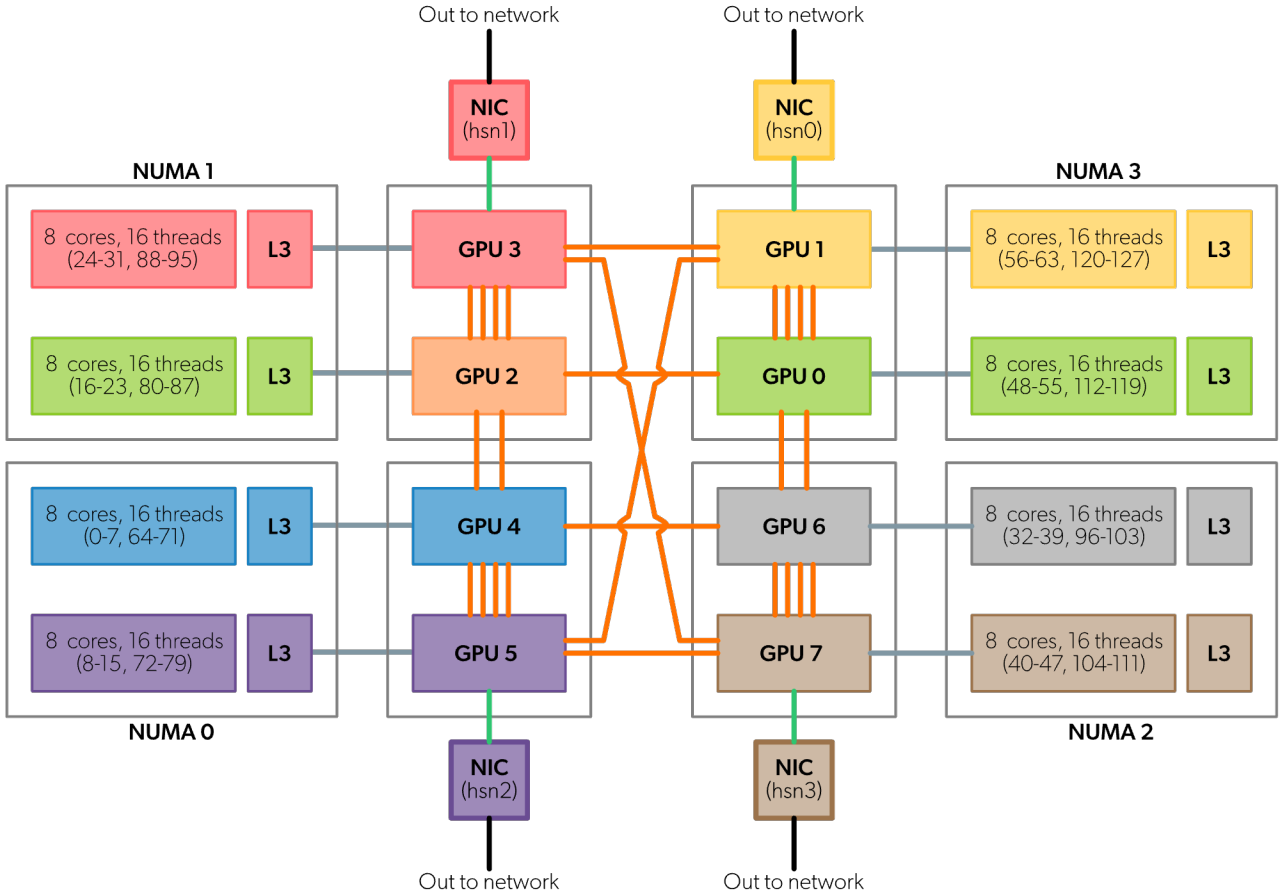



2978 nodes with 1 64-core AMD Trento CPU, 4 MI250x and 512+4x128GB memory

- But 8 cores and 32 GB reserved
- Compute GPU, not render GPU

# Real LUMI-G node

- 4 GPUs behave as 8 with 64GB each
- Bandwidth between the dies is low
- Binding to the CCDs is important for performance: Each GPU die closely associated to an L3 cache region
- Read the docs and/or take a course!



# Why care?

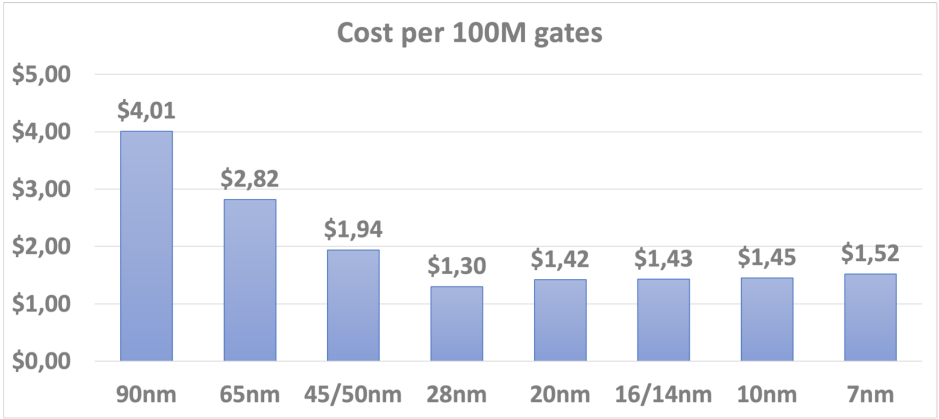
Efficiency matters!

- On 10 K€ worth of compute time – probably a big Tier-2 user: 20% gain is not a lot
  - Until you look at the scale of the whole infrastructure
- On 50 K€ – large VSC Tier-1 project: you can already spent time to increase efficiency
- On 250 K€ – large LUMI-BE project: 20% is on the order of 1 year of a Ph.D. grant
- On 2.5 M€ – a Flemish EuroHPC extreme scale project on LUMI: Do I need to say more?

# Why care?

Higher efficiency is the only way to get more resources for the money

- Data from the Marvell 2020 Investor Day:




- Processors still become faster, but little or no performance/EURO improvement
- Need to find ways to do more work per transistor
  - via (hardware) architectural innovations,
  - via better software to get more out of our hardware budget,
  - and by making better use of your current software and hardware.

# Challenges of large machines

- Topology and resource hierarchy (but already in Tier-2 clusters also)
- Not all file system properties scale
  - The larger the cluster, the more problems with metadata access
  - And any shared filesystem will have much higher file access latency than your PC…
- Schedulers don't scale well
  - More users = fewer jobs per user in flight or in the queue…
  - So, need a subscheduler to organise capacity computing workflows
- Linux may not be what you expect
  - Linux is not a supercomputer operating system,
  - it is both too little and too much
    - A large system may disable features to reduce OS jitter
- Reliability: Chance that a given node crashes on a given day 1‰, then
  - Your 1-day 1000 node job has only a 40% chance of completing successfully…

# Software stack: Challenges on LUMI

**Not your typical VSC environment**

- Very leading edge and inhomogeneous machine (new interconnect, new GPU architecture with an immature software ecosystem, some NVIDIA GPUs for visualisation, a mix of zen2 and zen3)
  - Need to remain agile
- Users that come to LUMI from 12 different channels (not counting subchannels), with different expectations
- Small central support team considering the expected number of projects and users and the tasks the support team has
  - But contributions from local support teams
- Cray Programming Environment is a key part of our system
- Users really want more and more a customised environment
  - Everybody wants a central stack as long as their software is in there but not much more
  - Look at the success of conda, Python virtual environments, containers, …

# Three-legged LUMI solution

- EasyBuild-maintained software stacks, each based on a particular release of the programming environment
  - Small central collection of base libraries and tools with a transparant system for user installation
  - But not (yet) your standard EasyBuild
    - FOSS: Wrong MPI…
    - Intel and AMD don't go together well
    - No AMD GPU support
- Spack also offered, based on releases of the PE
  - Much stronger tool at creating personal environments
- Containers are an important tool,
  - not for portability or performance portability as they tend to be promoted,
  - but as a tool to reduce the load on the file system of tools like Python and Conda,
  - and as an instrument to enable some software with conflicting requirements,
  - with AI and GUI environments currently being the main target on LUMI.

# Essential preparation for non-VSC systems: Take care of your workflows

- Separate input data, output data and intermediate data, and code.
- Separate visualisation and GUI work from the actual computations.
  And similarly for pre- and postprocessing.
- Additional motivation: Some large systems don't allow outgoing internet connections, not even from the login nodes!
  - So no access to GitHub or PyPi or …
- The main job scheduler may not be sufficient for all your scheduling needs.
  You may need a subscheduler, or a workflow management system running (partly) in a job rather than outside a job.
- On more and more clusters, access to the compute resources is only through the resource manager.
  - So no ssh to compute nodes
  - And maybe even no mpirun

# What does all this imply for AI?

- Training datasets on the system as lots of small files is not appreciated. Proper data management is required.
  - But it will help you on any system…
  - … unless you find a site that wants to invest crazy amounts of money in storage or offer you a server that is not shared by many.
- You can't run a 100 node training from a non-containerised installation
- Standard AMD containers will not work on LUMI as they are built for a different and incompatible network architecture
  - Strangely enough AMD containers support NVIDIA network technology better than anything else…
  - But the LUMI User Support Team offers some containers themselves (see later)

# What does all this imply for AI? (2)

- Not all process starters will work as much AI software is written for a cloud environment and badly adapted to HPC environments
  - Not a LUMI-specific or AMD problem though
- Much AI software is immature, even on NVIDIA hardware
  - But this is still worse on AMD hardware, even though there is a very rapid evolution
  - And good enough that NVIDIA seems to be getting a bit concerned as they rushed an H200 to the market to compete with MI300 and announced Blackwell long before it was ready
  - But some AI software is really meant for people with development skills on any platform…
    E.g., JAX README: "This is a research project, not an official Google product. Expect bugs and sharp edges."

# AI containers on LUMI

- Focus on PyTorch as this is by far the most popular package on LUMI
  - But some containers for Tensorflow and JAX and packages on top of AI frameworks, such as AlphaFold.
- These containers have a mix of conda components and specifically compiled ones
- Also contain the RCCL plugin needed for Slingshot, or link to a suitable MPI
- Extensible via a Python virtual environment
  - And some also have facilities to pack that in a file system friendly way
- Trying out an introductory AI-specific training
  - Next edition likely in early December
- Check the LUMI Software Library

# Getting access

- For Belgian users via LUMI-BE and EuroHPC
  - LUMI-BE tries to sit between the tier-1 systems and EuroHPC allocations
  - Repeat users are expected to try the EuroHPC channel also
- Separate channels for
  - Benchmarking to prepare a regular project application (4 months in most cases)
  - Software development (1 year)
  - Regular projects (1 year)
  - EuroHPC extreme scale projects (1 year) aimed primarily at projects with very scalable codes and very large simulations
- CPU-only resources are limited compared to tier-1 at the VSC
  - Belgian share basically the equivalent of a 150 node cluster

# Getting access (2)

- Access via project applications
  - Scientific and technical review for EuroHPC regular and extreme scale projects
  - Technical review for EuroHPC benchmark and development projects
  - Technical review only for LUMI-BE projects if there is a corresponding reviewed science project already
    - Procedures based on the VSC Tier-1 procedure
  - For open research only, otherwise buying compute time is possible
  - *Participation in the promotion expected as we need to show funding agencies that access to large machines is important*
- See
  - LUMI-BE: enccb.be/GettingAccess
  - EuroHPC: eurohpc-ju.europa.eu/access-our-supercomputers/eurohpc-access-calls_en

# Billing

- Basic idea: You are billed for all resources that someone else cannot use.
- CPU billing units (in core hours) for the CPU-only nodes
  - Billing based on cores reserved AND memory consumption in slices of 2 GB.
- GPU billing units (in GPU hours) for the GPU nodes
  - GPU hour definition based on the idea of 4 GPUs per node rather than the 8
  - Billing based on number of Slurm GPUs, number of cores (groups of 8) and CPU memory consumption (slices of 64 GB)
- Storage use is also billed
  - Idea is that someone may need a large quota but can clean up after a run
    - Quotas limit maximum use, storage billing limits average use
  - In TB hours: Using 1 TB for one hour costs 1 TB hour on LUMI-P storage
    - x 10 for flash storage (so 10 TB hours if you use 1TB for 1 hour)
    - x 0.5 for object storage
  - Don't be too greedy: We can use 3.5-4 PB on average on LUMI-P and 0.4 PB on LUMI-F

# Pay attention to…

- Strict file quota on LUMI (block quota are flexible)
  - Not the machine to dump your big data set as millions of small files
- Strict limits on number of jobs in Slurm
  - and array job counts as many jobs,
  - so capacity computing users will need an additional level to manage their jobs.
- NoTorque wrappers for Slurm
- No ssh to compute nodes
- Software that comes as binaries does not always work on LUMI
  - MPI usually the culprit
- Only Cray MPICH, no full Open MPI support
  - And process starter is srun, no mpirun/mpiexec
- Cray PE uses universal compiler wrappers for each language, no mpicc etc.

# Pay attention to… (2)

- GUI applications can be problematic (due to COS constraints)
- GPU programming
  - HIP and OpenMP offload are the prefered models
  - OpenACC only in Cray Fortran
  - OpenCL support unclear
  - Try to support AdaptiveCpp (formerly hipSYCL) and looking at Intel DPC++ also (as we can get support for that)
  - NO CUDA, NO OpenGL on the AMD GPUs
- Check software compatibility before you apply
  - LUMI-BE and LUST are there to help advise you
- A preparatory project on LUMI is preferred as even for CPU codes LUMI is different from Hortense.
  - The interconnect can have a large influence on scalability also
  - And preparing the code may be nontrivial if it is not already on LUMI

# LUMI support

- Distributed support system
- Allocations: Support comes from the organisation that granted the allocation
  - Belgian allocation: lumi-be-support@enccb.be
- LUMI User Support Team offers L1 and L2 support (but cannot help with allocation problems)
  - Via web forms on lumi-supercomputer.eu/user-support/need-help
  - Very small team: 10 FTE for >700 projects/year
  - No access to your files!
- VSC has a tier-0 support project
  - Not yet fully staffed
  - Also via lumi-be-support@enccb.be

# LUMI support (2)

- EuroHPC has granted the EPICURE project to set up a network for advanced L2 and L3 support across EuroHPC centres
  - Belgium participates as a partner in the LUMI consortium
- In principle the EuroHPC Centres of Excellence should play a role in porting of specific applications
- But don’t forget there is a thing as L0 support:
  - Read the documentation
  - Take proper courses
  - Talk to your colleagues

# What support can and cannot do

"LUST did not support me well as they could not solve my compiler problems."

I want to simulate hydroxypropylcellulose in GROMACS. How do I set this up?

My code crashes with error message "…" but I cannot give you the code because it is proprietary and I also have no reproducer for you. Tell me what's wrong.

# What support can and cannot do (2)

Supercomputer support is there to
support you in the computational aspects of your work
related to the supercomputer
but not to take over your work.

And no support system can give you a free Research Software Engineer…

# Were can I get a training?

- System-specific trainings
  - LUST with HPE and AMD organise trainings
    - 1- or 2-day introductory trainings
    - 4-day comprehensive with more attention on how to run efficiently and on development and profiling tools
    - Announcements on lumi-supercomputer.eu/events and various mailing lists
    - Training materials on lumi-supercomputer.github.io/LUMI-training-materials
  - Local 2-day introductory training is possible if enough candidates
- Application-specific trainings
  - Should come from the EuroHPC centres of excellence and other organisations
- Combined: Application-on-system trainings
  - Small target group, but LUST is trying out an AI-on-LUMI training
- EuroHPC has some trouble creating a full replacement for the PRACE trainings

# Questions?

# Useful links

- Main LUMI website: lumi-supercomputer.eu
- Main LUMI documentation website: docs.lumi-supercomputer.eu
- LUMI Software Library: lumi-supercomputer.github.io/LUMI-EasyBuild-docs
- LUMI training materials archive: lumi-supercomputer.github.io/LUMI-training-materials
- LUMI-BE information: enccb.be/LUMI
- LUMI introductory training materials tuned for the situation in Belgium: klust.github.io/LUMI-BE-training-materials
- VSC@UAntwerpen Supercomputers for Starters course notes: klust.github.io/SupercomputersForStarters