EasyBuild on LUMI, a pre-exascale computer

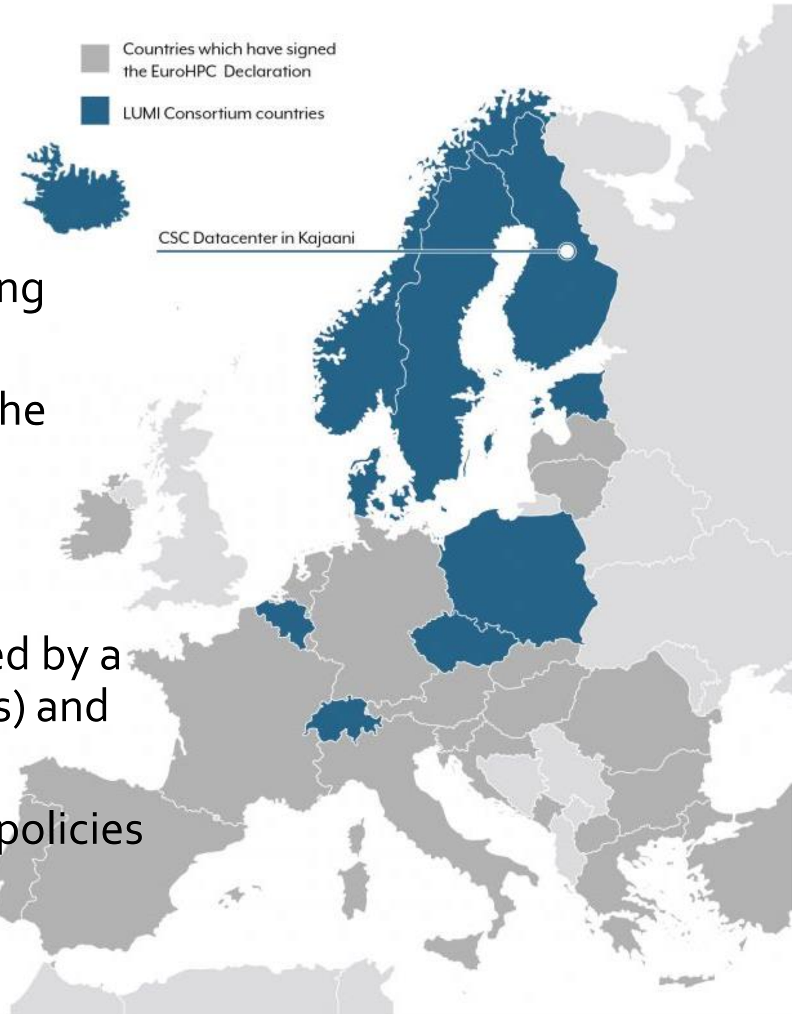
Kurt Lust
LUMI User Support Team (LUST)
University of Antwerp

What this talk is not…

- A talk about sophisticated workflows with automated quality testing
  - We've only been on the system since October 13 2021, don't have a finalised directory structure and don't have test nodes
- A talk with a lot of information about how we deal with AMD GPUs.
  - We just got some information last week and are still waiting for a functioning test system with a sufficiently final programming environment

LUMI Consortium

- Unique consortium of 10 countries with strong national HPC centres
- Joint investment of EuroHPC JU (50%) and the consortium countries
- The resources of LUMI will be allocated proportional to the investments
- The share of the EuroHPC JU will be allocated by a peer-review process (cf. PRACE Tier-0 access) and available for all European researchers
- Each LUMI consortium country sets its own policies for a national access program


Countries which have signed the EuroHPC Declaration
LUMI Consortium countries
CSC Datacenter in Kajaani

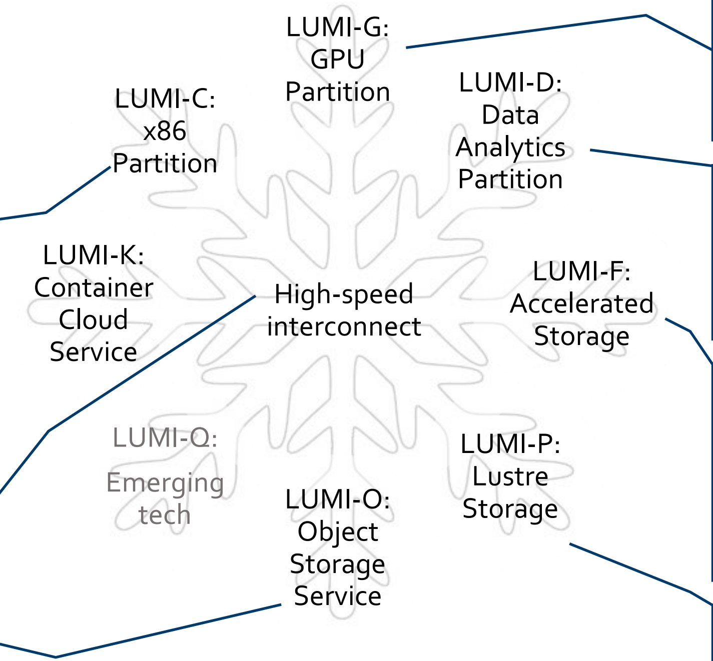
LUMI, the Queen of the North

*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
- ~200,000 AMD EPYC CPU cores

Possibility for combining different resources within a single run. HPE Slingshot technology.

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



LUMI-G: GPU Partition

LUMI-C: x86 Partition

LUMI-D: Data Analytics Partition

LUMI-K: Container Cloud Service

High-speed interconnect

LUMI-F: Accelerated Storage

LUMI-Q: Emerging tech

LUMI-O: Object Storage Service

LUMI-P: Lustre Storage

Tier-0 GPU partition: close to 500 Pflop/s (vector) from AMD MI250X GPUs

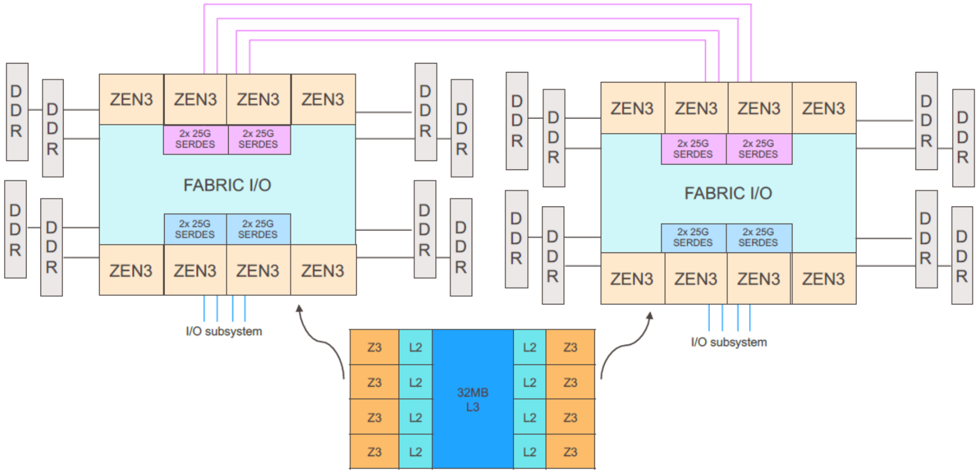
Interactive partition with 8 4 TB CPU-only nodes and 8 nodes with 8 NVIDIA A40 GPUs

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

80 PB parallel file system

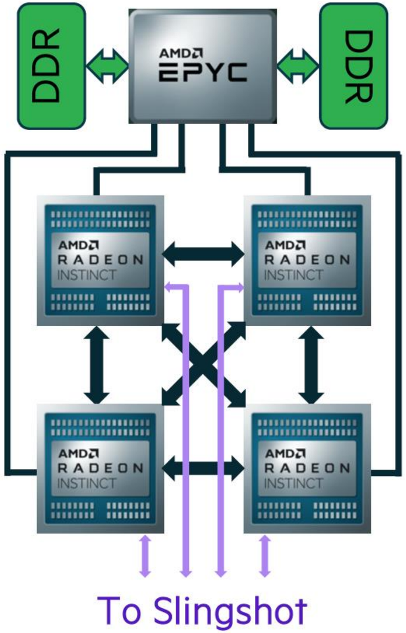
LUMI compute node configurations

LUMI-C



2x 64-core AMD Milan processors per node
1376 nodes with 256 GB, 128 with 512 GB and 32 with 1 TB

LUMI-G



LUMI user support

- Centralized virtual help-desk run by the distributed LUMI User Support Team
  - The model is based on a network of dedicated LUMI experts: each partner (except one) provides one FTE for the task
  - User Support Team will also provide end-user training, maintain the software portfolio and user documentation of the system
- "Level 3" support (e.g. application enabling, methodology support) via local centers, the EuroHPC Competence Centers and a team at HPE and AMD
- National support for issues with accounts and allocations

Given that…

- Rather experimental and inhomogeneous machine (new interconnect, new GPU architecture with an immature software ecosystem, some NVIDIA GPUs, a mix of zen2 and zen3)
- Users that come to LUMI from 11 different channels (not counting subchannels)
- Small central support team considering the expected number of projects and users and the tasks the support team has
  - But the consortium should contribute
- Cray Programming Environment is a key part of our system
- Operational: 4 copies of the software stack due to the file system setup

What are we looking for?

- A framework that allows collaboration and testing software as a regular user yet be assured that it will also work in a central installation
- Easy way to pass software installation instructions to the user
  - Don’t want everything in the central directory
  - LUST nor national support teams can write in the project directories of regular users
- An easy way for developers and application experts to contribute software to the LUMI community
- A tool with a community for support and continued development

EasyBuild or Spack?

- Spack is better supported on Cray + AMD (Frontier and El Capitan)
- EasyBuild seems more popular in LUMI consortium countries
    - Switzerland/CSCS is also in the consortium
- Strengthening the European software stack
- Community engagement is easier with EasyBuild
- Spack more a developers tool, EasyBuild more a support person tool?
- EasyConfig files offer an easy mechanism to document modules
- As long as things are done in EasyConfig files rather than EasyBlocks, two teams can easily work in parallel on different configurations without merge conflicts

Policy: Which software

- Pro-active:
  - Large set of libraries, mostly installed in the central stack
  - Some packages that proved popular in our surveys, though currently often only available as an EasyConfig file for user installation
- Re-active
  - Based on tickets, and usually for user install first

- Conda and likely much of Python and R should be containerised due to the high number of small files that may hamper the performance of the parallel file system
- Consider spack or manual install for “hard” packages

Policy: Limited central stack

- Avoid having too many variants and versions since that is confusing
- Management problem: Which software to carry on to the next version of the stack?
  - We don’t know our users the way a university tier-2 support team knows their users
  - User installs lead to duplication but also to automatic removal
- Removing a faulty package is hard, so want to restrict to software that we are fairly confident will work
- Currently also a management problem as we have 4 copies without a dedicated master

Policy: Limited central stack (2)

- Easier license management as it becomes the responsibility of the PI and members of the project
  - For most software we work with a bring-your-own-license model
  - More and more codes come with some restrictions even for academic use, e.g., first registering as a user
  - No management burden for LUST to select, e.g., who can use VASP.

Organisation: Software stacks

- CrayEnv: Cray environment with some additional tools pushed in through EasyBuild
- LUMI stacks, each one corresponding to a particular release of the PE
  - Work with the Cray PE modules as delivered, no way to install the Cray PE through EasyBuild
  - 4 versions for the 4 types of hardware: zen2, zen3, zen2 + NVIDIA GPU, zen3 + MI250X (64 sockets of zen2, 3072 + 2560 sockets of zen3)
  - Some software may be installed outside those stacks
- Future: Stack based on common EB toolchains as-is
  - MPI may be the problem

Modules: Lmod

- Strengthening the European ecosystem so you must be using Environment Modules 5 for sure?
- Well, no: Cray supports Environment Modules 3 and Lmod 8.3
- Lmod-version of Cray PE uses a somewhat unconventional partial hierarchy
- LUMI stack implemented as a 2-level hierarchy:
  - LUMI stack version
  - Partition: the 4 different hardware configurations, and three pseudo-partitions
  - EasyBuild uses a flat naming scheme on top of this
- Relocatable: Discovers its own root

Modules: Lmod (2)

- User modules integrated transparantly
  - Environment variable points to the root of the user stack
  - Structure of the module tree imposed by our setup, mirrors the central stack
  - Almost unified view of centrally installed and user/project installed modules
    - LUMI modules automatically also load the corresponding user modules
- Several presentations of the module tree, configurable via other modules
  - E.g., hide modules that are irrelevant for all but expert users
  - Labels or directories, to even hiding the Cray PE hierarchy
- Use generic implementations in Lua of the modules and Lmod introspection functions in the module to determine what the module should do
  - Easier to make consistent corrections on the whole system

EasyBuild installation on LUMI

- Fix the version of EasyBuild for a given version of the software stack
  - Bootstrapped for each version of the LUMI software stack to make those stacks fully independend of each other
- Configuration modules for EasyBuild to configure for specific tasks
  - Single module for the LUMI stack linked with different names, e.g.,
    - EasyBuild-production when installing software in the central stack
    - EasyBuild-user to let a user install software
  - Single piece of code is more complex but it is easier to ensure consistency of the settings for central and user/project install of software
  - Picks up where to install software from its name and its location in the module tree

EasyBuild installation on LUMI (2)

- 2 central repositories with various service levels:
  - LUMI-SoftwareStack: Repository for the central software stack and some other packages that we fully support and install centrally
  - LUMI-EasyBuild-contrib: Repository for software that we do not want to install centrally, e.g., because we cannot fully support the package or are not convinced that the configuration is already OK for a large enough group of users
    - Would also include "annoying" packages such as OpenFOAM or Yambo that will probably never make it to the central stack
- Installing GROMACS:

```
$ ml LUMI/21.08 partition/C EasyBuild-user
$ eb -r GROMACS-2021-cpeGNU-21.08-PLUMED-2.7.2-CPU.eb
```

```
[lumi][kurtlust@uan03-1002 ~]$ ml LUMI/21.08 partition/C EasyBuild-user

Lmod is automatically replacing "craype-x86-rome" with "craype-x86-milan".


EasyBuild configured to install software from the LUMI/21.08 software stack for the LUMI/C partition in the
user tree at /users/kurtlust/LUMI-user-appl.
  * Software installation directory: /users/kurtlust/LUMI-user-appl/SW/LUMI-21.08/C
  * Modules installation directory: /users/kurtlust/LUMI-user-appl/modules/LUMI/21.08/partition/C
  * Repository: /users/kurtlust/LUMI-user-appl/ebrepo_files/LUMI-21.08/LUMI-C
  * Work directory for builds and logs: /run/user/10012026/easybuild
    Clear work directory with clear-eb


Due to MODULEPATH changes, the following have been reloaded:
  1) craype-network-ofi

[lumi][kurtlust@uan03-1003 ~]$ eb -r GROMACS-2021-cpeGNU-21.08-PLUMED-2.7.2-CPU.eb
== Temporary log file in case of crash /run/user/10012026/easybuild/tmp/eb-gfdni031/easybuild-kzmiwnf0.log
== Running parse hook for GROMACS-2021-cpeGNU-21.08-PLUMED-2.7.2-CPU.eb...
………….
== Build succeeded for 2 out of 2
```

```
[lumi][kurtlust@uan04-1003 ~]$ module avail
---------------------------------------------- HPE-Cray PE modules ----------------------------------------------
   PrgEnv-aocc/8.1.0 (D)    cray-dsmml/0.2.0       (D)    cray-pmi-lib/6.0.13 (D)    perftools
   PrgEnv-cray/8.1.0 (D)    cray-fftw/3.3.8.11     (D)    cray-pmi/6.0.13     (D)    perftools-base/21.05.0 (L)
   PrgEnv-gnu/8.1.0  (D)    cray-hdf5/1.12.0.6     (D)    cray-python/3.8.5.1        perftools-lite
…
-------------------- EasyBuild managed user software for software stack LUMI/21.08 on LUMI-C --------------------
   GROMACS/2021-cpeGNU-21.08-PLUMED-2.7.2-CPU    PLUMED/2.7.2-cpeGNU-21.08
---------------------- EasyBuild managed software for software stack LUMI/21.08 on LUMI-C -----------------------
   Blosc/1.21.0-cpeCray-21.08                                        gdbm/1.20-cpeCray-21.08
   Blosc/1.21.0-cpeGNU-21.08                                   (D)    gdbm/1.20-cpeGNU-21.08             (D)
   Boost/1.77.0-cpeCray-21.08                                        gettext/0.21-cpeCray-21.08-minimal
…
---------------------- Infrastructure modules for the software stack LUMI/21.08 on LUMI-C -----------------------
   EasyBuild-user/LUMI    cpeAMD/21.08    cpeCray/21.08    cpeGNU/21.08
------------------------------------------ HPE-Cray PE target modules ------------------------------------------
   craype-accel-amd-gfx908       craype-hugepages1G      craype-hugepages4M      craype-network-ofi (L)
…
```

```
xpmem/2.2.40-7.0.1.0_3.1__g1d7a24d.shasta                                     (L)
------------------------------- LUMI partitions for the software stack LUMI/21.08 --------------------------------
   partition/C (S,L,CPUcompute)         partition/G (S,GPUcompute)
   partition/D (S,DataVisualisation)    partition/L (S,D:login)
------------------------------------------------ Software stacks ------------------------------------------------
   CrayEnv (S)    LUMI/21.08 (S,LTS,L)
---------------------------------------- Modify the module display style ----------------------------------------
   ModuleColour/off  (S)        ModuleLabel/PEhierarchy (S)    ModuleStyle/default
   ModuleColour/on   (S,D)      ModuleLabel/system      (S)    ModuleStyle/reset   (D)
   ModuleLabel/label (S,L,D)    ModulePowerUser/LUMI    (S)
--------------------------------------------- System initialisation ---------------------------------------------
   cray-ucx/2.7.0-1    init-lumi/0.1 (S,L)
-------------------------------------- This is a list of module extensions --------------------------------------
    Autoconf         (E)     CMake   (E)     Meson (E)     Yasm  (E)     help2man (E)     patchelf (E)
    Autoconf-archive (E)     Doxygen (E)     NASM  (E)     byacc (E)     htop     (E)     re2c     (E)
    Automake         (E)     GPP     (E)     Ninja (E)     flex  (E)     libtool  (E)     sec      (E)
    Bison            (E)     M4      (E)     SCons (E)     gperf (E)     make     (E)     tree     (E)
```

```
These extensions cannot be loaded directly, use "module spider extension_name" for more information.

  Where:
   LTS:      Long-Term Support, modules available for up to two years after release. system software permitting
   L:        Module is loaded
   S:        Module is Sticky, requires --force to unload or purge
   Aliases:  Aliases exist: foo/1.2.3 (1.2) means that "module load foo/1.2" will load foo/1.2.3
   D:        Default Module
   E:        Extension that is provided by another module

Additional ways to search for software:
* Use "module spider" to find all possible modules and extensions.
* Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
See the LUMI documentation at https://docs.lumi-supercomputer.eu/computing/Lmod_modules/ for more information on
searching modules.
If then you still miss software, contact LUMI User Support via https://lumi-supercomputer.eu/user-support/need-help/.
```

Our way of working

- Cray PE-based toolchains are a further development of the CSCS ones with several improvements
- EasyConfig files derived from CSCS and EasyBuilder ones
- Biggest problem is that EasyBlocks often don’t support Cray
  - Tend to take the CSCS approach and use generic EasyBlocks
  - Effort in adapting a custom EasyBlock may be lost in the next release of EasyBuild
- Tend to add more documentation (also local) to the module files
  - And pay attention to making software more discoverable with module spider and module keyword
- Considered using the Lmod extension function but backed down from it due to bugs in Cray Lmod (and Lmod in general)

Our way of working (2)

- Not a great fan of too many small packages
  - Do have a buildtools module which bundles various build tools for a more predictable build environment
    - And build those with the system Python
  - Bundle components remain findable but that requires manual work in the EasyConfig files
- Develop in “user mode” as a regular user would, and then import into the contributed or main software stack repository
  - Some LUSTers have a private setup of the full software stack for developement and testing

Upcoming challenges

- AMD GPUs: Immature development tools and immature applications
  - Essentially had to look elsewhere to learn about the plans of Cray for AMD support (the early access platform for Frontier)
  - And how will installation procedures of applications support AMD?
- What with SYCL (and the DPC++ dialect)?
  - Currently third party tools, how will they integrate?
  - GROMACS works on a SYCL code path…
- What with those packages that require a lot of development?
  - AI tools due to the complexity of the software
  - Bio-informatics due to the zoo of small tools with broken installation procedures

Conclusions

- EasyBuild deployment so far successful on LUMI but several challenges ahead
- A-typical setup specific for Cray systems using the Cray PE
- Very good instrument to support and deploy software in a distributed manner and communicate installation instructions
  - Already several successful installations by users based on build recipes provided by LUST
- Lmod is also an important building block of our setup

Availability

- Public GitHub repositories
  - LUMI-SoftwareStack: Lmod setup and stable EasyBuild recipes
    - Contains the technical documentation for the setup
  - LUMI-EasyBuild-contrib
- Many EasyConfig files also documented
- May add documentation about how to run on LUMI
  - More likely to maintained if it sits with the build instructions and can be updated in a single move
  - May be pulled in automatically in the regular documentation

LUMI

Kurt Lust

LUMI User Support Team

Kurt.Lust@uantwerpen.be

Follow us

Twitter: @LUMIhpc

LinkedIn: LUMI supercomputer

YouTube: LUMI supercomputer

http://www.lumi-supercomputer.eu/

contact@lumi-supercomputer.eu

EuroHPC
Joint Undertaking

The acquisition and operation of the EuroHPC supercomputer is funded jointly by the EuroHPC Joint Undertaking, through the European Union's Connecting Europe Facility and the Horizon 2020 research and innovation programme, as well as the of Participating States FI, BE, CH, CZ, DK, EE, IS, NO, PL, SE.

Leverage from
the EU
2014–2020

European Union
European Regional
Development Fund

Kainuun liitto

LUMI