



LUMI

Introduction to
LUMI

Moving your AI training jobs to LUMI workshop

11.06.2026

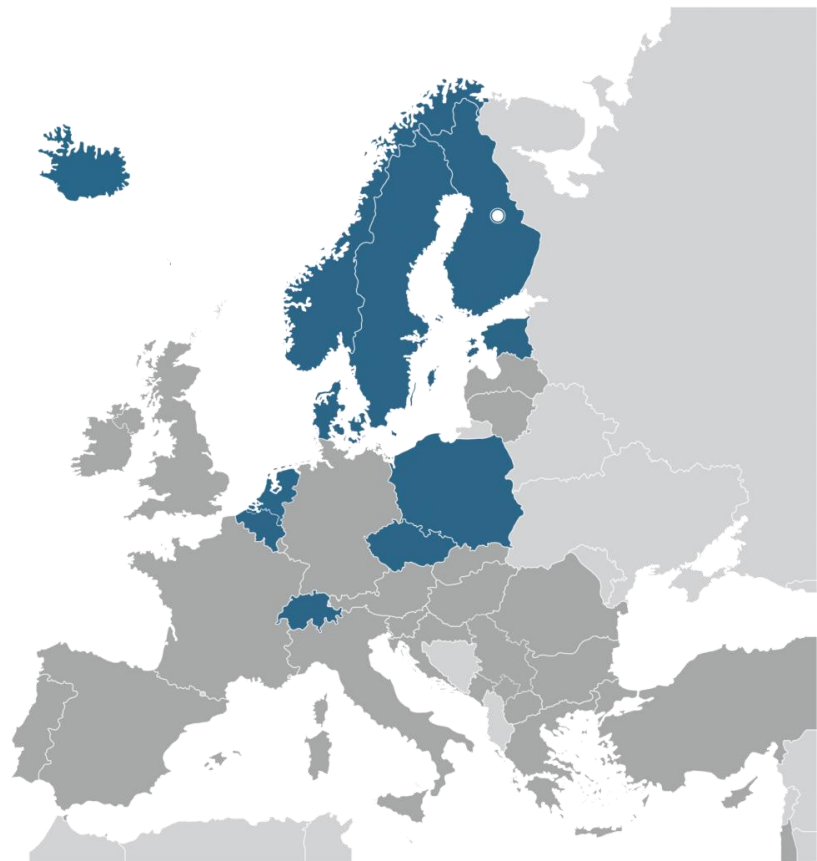
LUMI is not one single computer

It behaves quite a bit different than your local computer

LUMI is a very fast computer in Europe

LUMI

- 9th fastest computer in world (TOP500)
- Operated by LUMI consortium
 - 11 countries collaborating
 - 50 % financed by EuroHPC JU
- Located in Kajaani, Finland
- Distributed LUMI user support team (LUST)
 - One full time employee equivalent from each country
 - Offer email support, courses, workshops, ...
 - Responsible of software stack
- LUMI AI factory for AI topics



LUMI is a **cluster** of individual computers

LUMI

- LUMI is not one superfast computer
- Instead it consists of a few thousand individual computers ("nodes")
- All of them are connected by a fast interconnect
- Speed comes from parallelization



LUMI consists of different parts

LUMI

- Computers

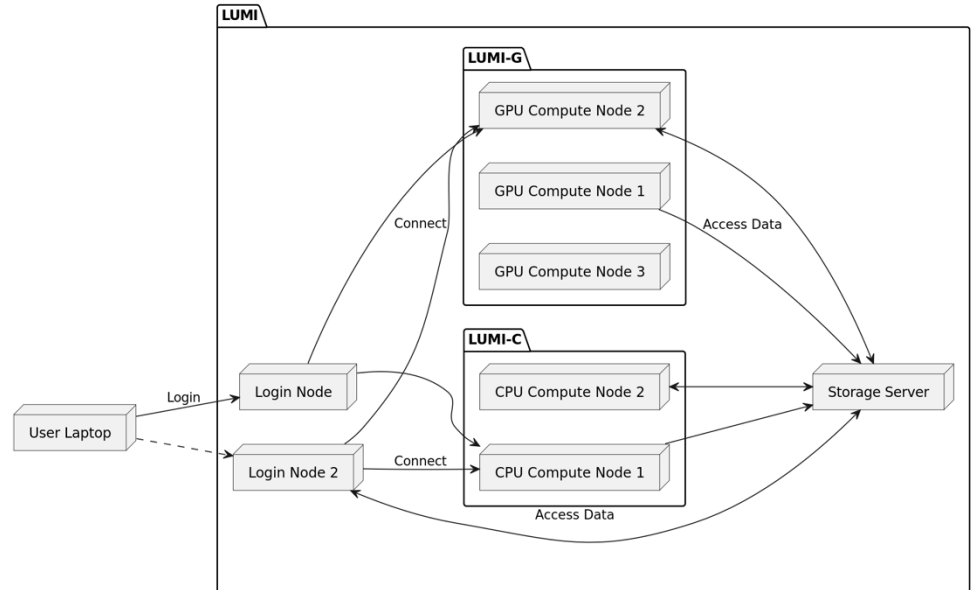
- Login nodes – UAN (user access nodes)
- CPU compute nodes – LUMI-C
- GPU compute nodes – LUMI-G
- Visualisation nodes – LUMI-D
- Container cloud platform – LUMI-K

- Storage

- 80 PB main parallel storage – LUMI-P
- 8.5 PB accelerated storage – LUMI-F
- 30 PB object-based storage – LUMI-O

- Interconnect

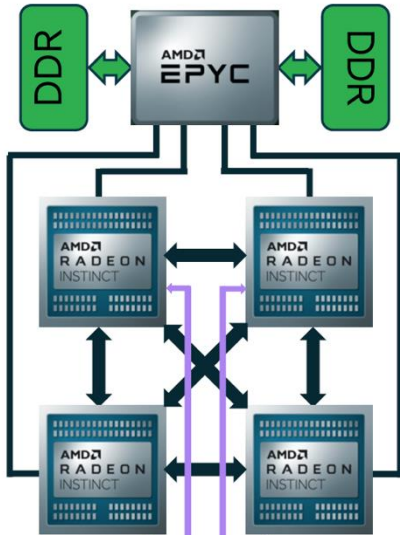
- HPE Slingshot 13
- Connects everything



LUMI-C and -G are quite different

LUMI

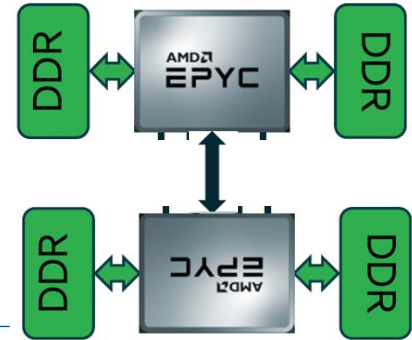
LUMI-G



2978 nodes with
4x MI250X (2 x 64GB)
1x AMD Trento CPU
512 GB RAM
4x 200 Gbit/s NIC

1888 nodes with 256 GB,
128 with 512 GB and 32 with 1 TB RAM
2x 64-core AMD Milan CPUs
1 x 200 Gbit/s NIC

To Slingshot



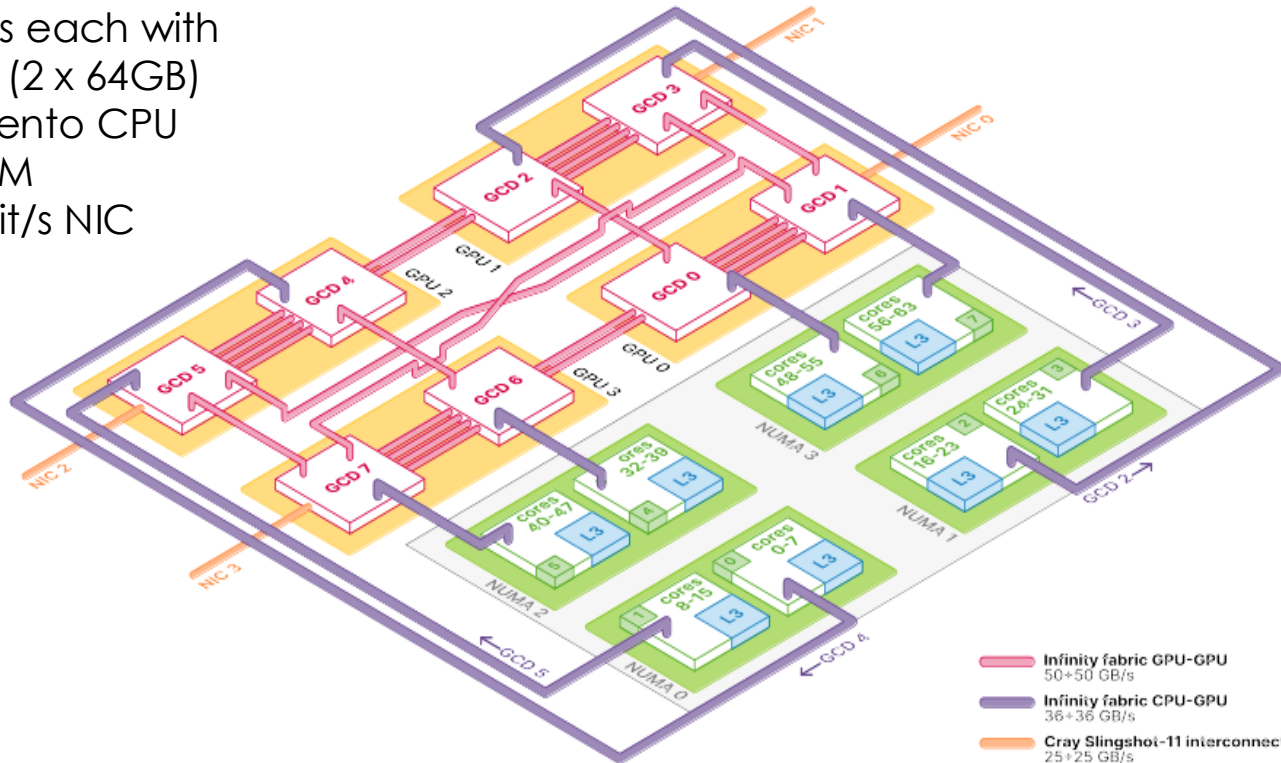
To Slingshot

LUMI-C

GPU nodes are the center of LUMI

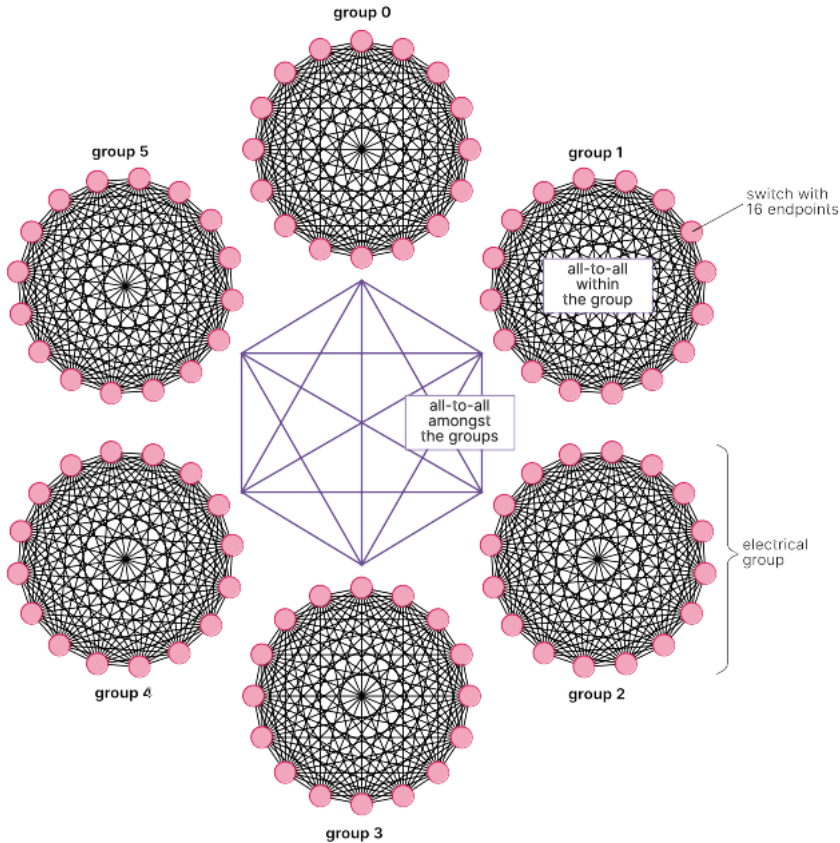
LUMI

2978 nodes each with
4 x MI250X (2 x 64GB)
1 x AMD Trento CPU
512 GB RAM
4 x 200 Gbit/s NIC



Interconnect is the fast backbone of LUMI

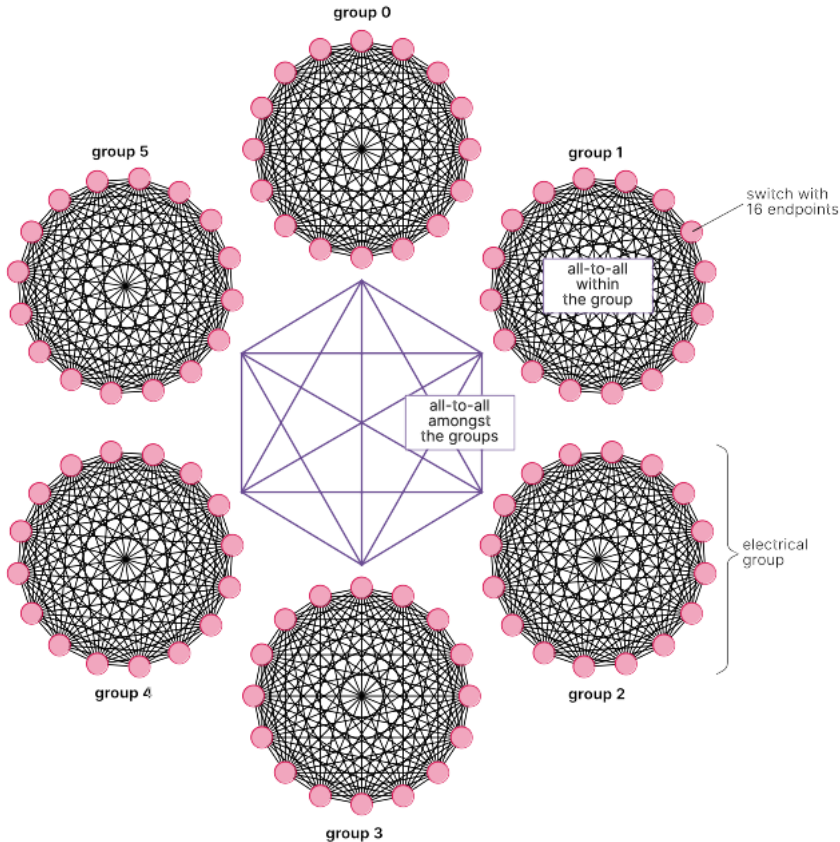
LUMI



- Connects all nodes
- Has a similar role to ethernet
- Much higher speed bandwidth

Interconnect is the fast backbone of LUMI

LUMI



- Slingshot in Dragonfly topology
 - Each G node is connected to 4 switches
 - All-to-all amongst switches in a group
 - All-to-all between groups
 - Max of 3 switch hops
- Make sure to use it

AMD is not Nvidia

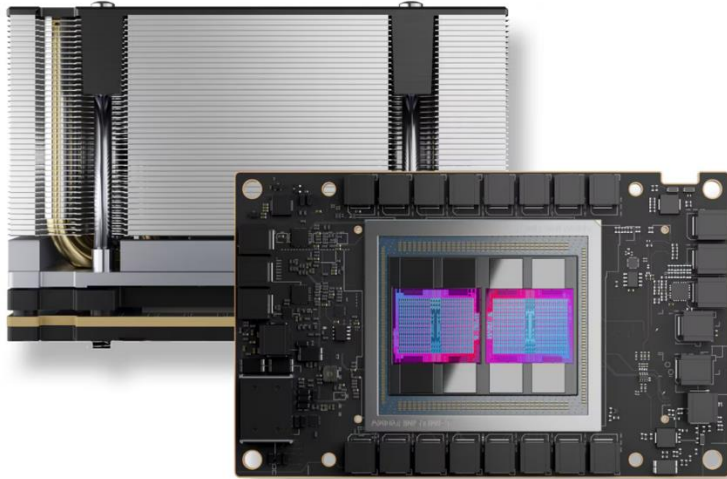
But the differences are quite small

Our GPUs are confusing

LUMI

Each AMD Instinct MI250X

- 2 Graphics Compute Die (GCD)
- 110 compute units per GCD with each 64 stream processors
- 64 GB HBM GPU memory per GCD
- Each process can only use 64GB max – not 128GB



Different names but usually **same concept**

L U M I



PyTorch

ML Training

PyTorch

Infiniband / RoCE

Networking Between Nodes

HPE Slingshot

NCCL

Cross-GPU Communication

RCCL

CUDA / CuDNN

Software Stack

ROCm

A100, H100

GPU

MI250X, MI300X

ROCm is not CUDA

- ROCm is the equivalent software stack to Nvidia's CUDA
- Basically drop-in replacement
- Very similar concept
- Some small differences
- Consists of
 - GPU drivers
 - Compilers and profilers
 - Math and communication libraries

PyTorch makes it simple

- Both CUDA and ROCm are loaded with `cuda` submodule
- Check whether you can see any GPUs with `torch.cuda.device_count()`

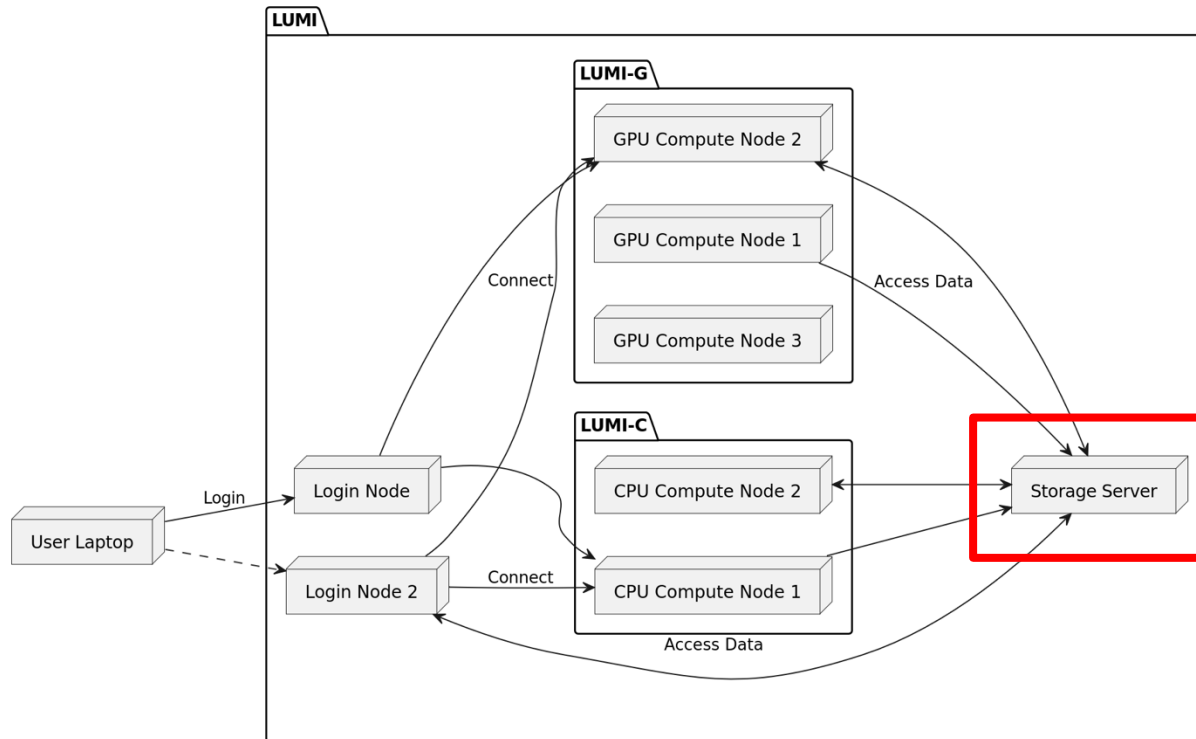
```
dietzej@nid005021:~$ singularity exec $SIF python -c 'import torch; print(f"Number of GPUs
: {torch.cuda.device_count()}"); print(torch.cuda.get_device_properties(0))'
Number of GPUs: 1
_CudaDeviceProperties(name='AMD Instinct MI250X', major=9, minor=0, gcnArchName='gfx90a:sr
amecc+:xnack-', total_memory=65520MB, multi_processor_count=110)
dietzej@nid005021:~$ █
```

Storage is not as easy as on your laptop

But if you follow some rules you will be fine

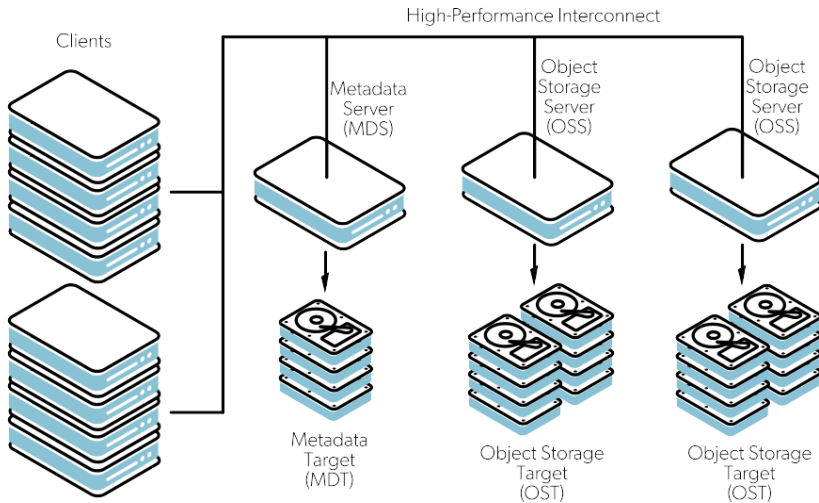
There is *more than one* storage server

L U M I



LUMI has **three** storage systems

LUMI



- LUMI-P
 - Lustre file system
 - Disk based
 - 4 independent systems with each 20 PB
- LUMI-F
 - Lustre file system
 - Solid-state (flash) based
 - 8.5 PB
- LUMI-O
 - Object storage based
 - Disk based
 - 30 PB

There are **no** local disks

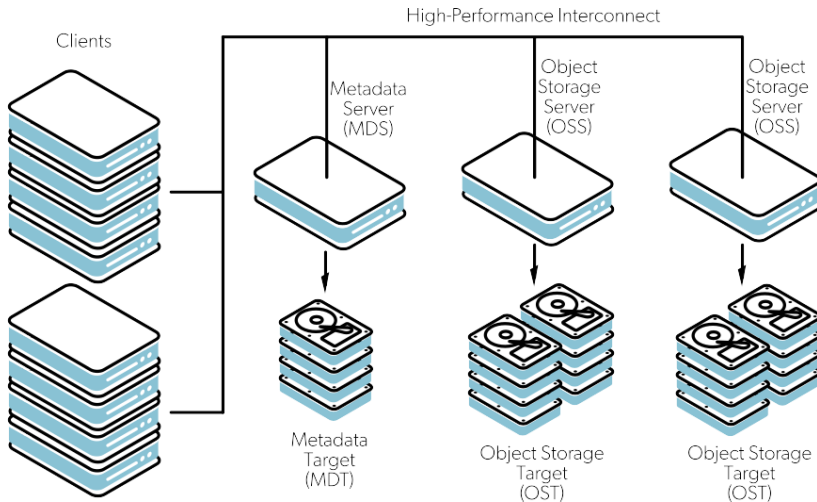
- Compute nodes have no local disks
- Instead network storage (LUMI-P & -F) has to be used
- 4 storage areas

Area	Path	Usage
User home	/users/<username>	Configuration files
Project persistent	/project/<project>	Installations + final results
Project scratch	/scratch/<project>	Input + Intermediate results
Project flash	/flash/<project>	Input if high bandwidth is needed

Lustre doesn't like **many** small files

Lustre consists of 3 parts

- **Client**
 - Compute or login node that wants to access a file
- **Metadata Server**
 - Doesn't store file content
 - Just metadata like location, size, ...
 - Tells client where to find file
- **Object Storage Server**
 - Stores actual file content
 - Either complete file or parts
 - Sends and receives data to/from client

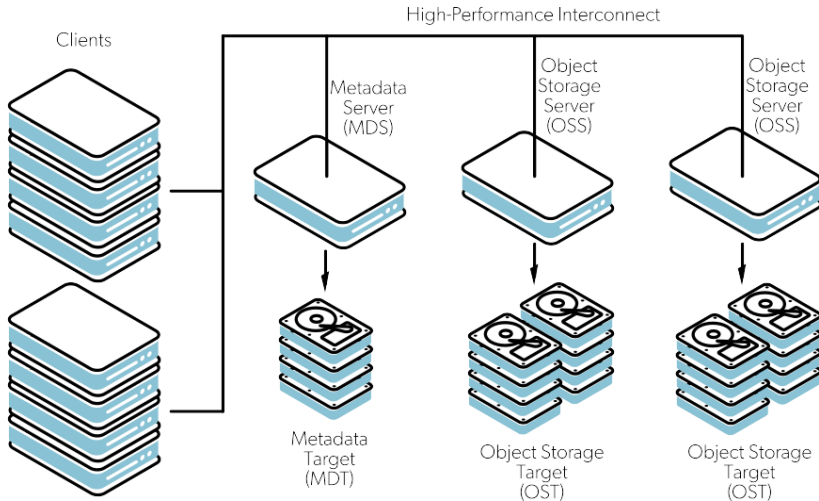


Lustre doesn't like **many** small files

LUMI

Problem with many small files

- For each file the client queries the Metadata server (MDS)
- Many object storage servers but only one MDS
- MDS can get overloaded by queries if many clients ask for lots of small files each

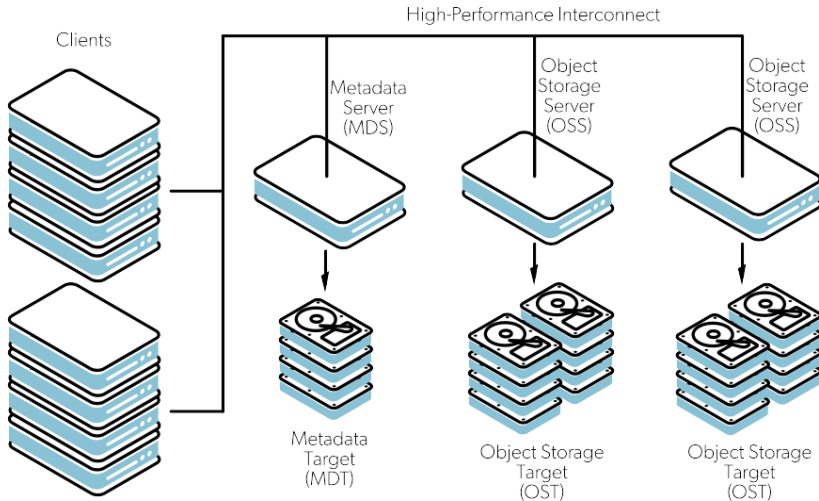


Lustre likes **few** large files

LUMI

To avoid **overloading** MDS

- Avoid many (thousands) small files
- Avoid opening/closing many files in short time
- Bundle files together
- Python environments can be a problem → discuss later

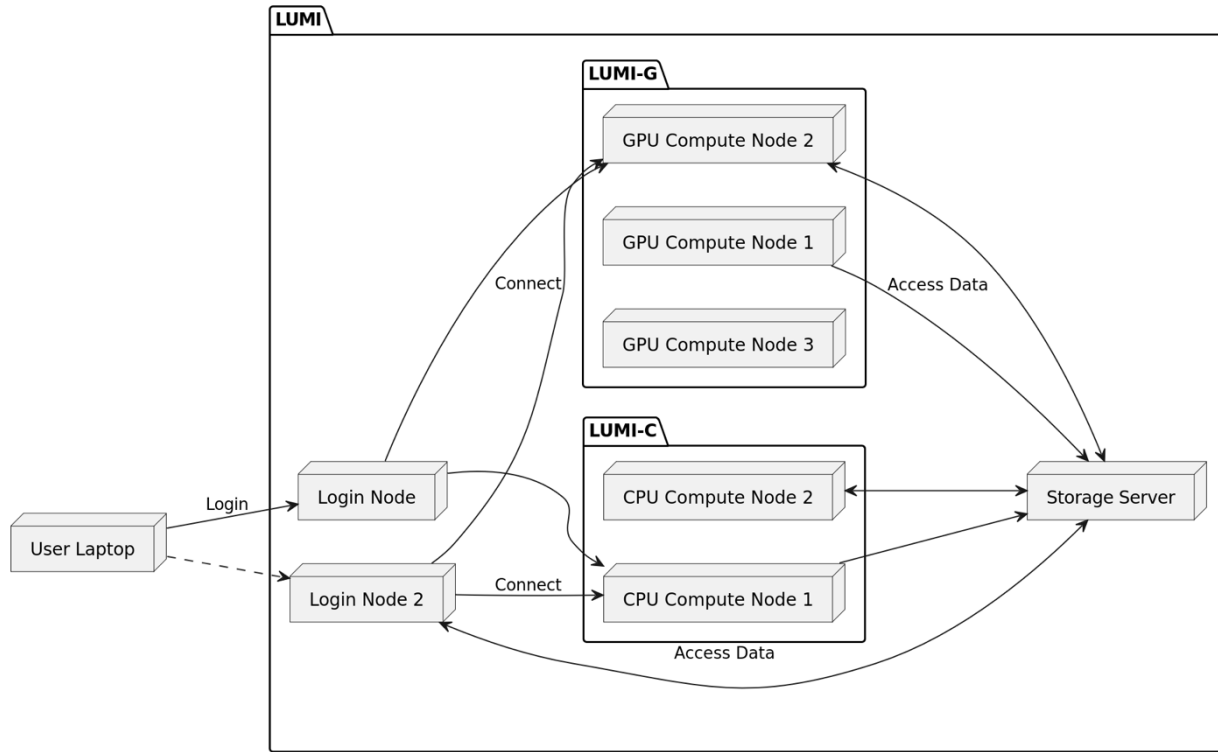


What about `/tmp`?

- Compute nodes don't have local disks/flash
- `/tmp` resides in memory
- Consumes space of your memory allocation
- Remember to allocate enough memory if you want to use `/tmp`

LUMI consists of different parts

LUMI



Use them well and you will get great results

Questions?