

Introduction of LUMI supercomputer



Introduction of LUMI supercomputer

- Architecture and key parameters
- LUMI-C partition
- LUMI-G partition
- Scratch and Project storage

LUMI Component Overview

LUMI-C : CPU compute

LUMI-G : GPU compute

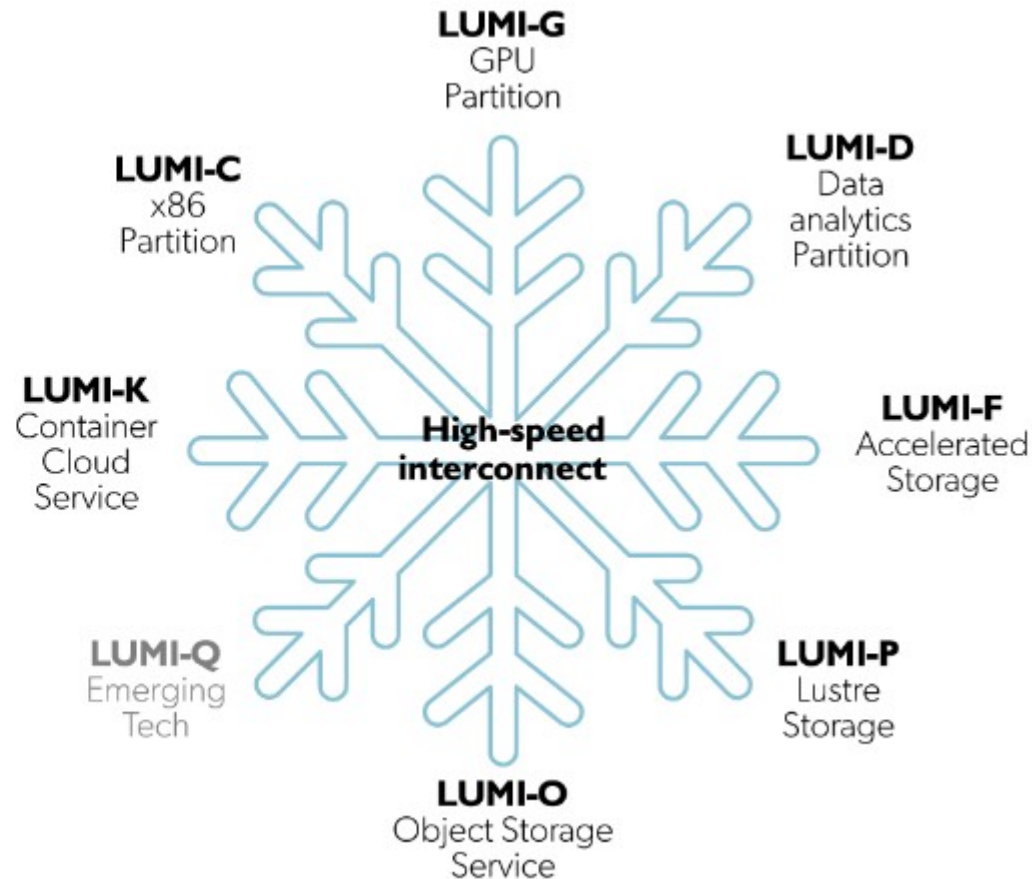
LUMI-D : Data analytics

LUMI-P : Parallel file system

LUMI-F : Flash-based parallel file system

LUMI-O : Object storage

LUMI-K : Container orchestration platform



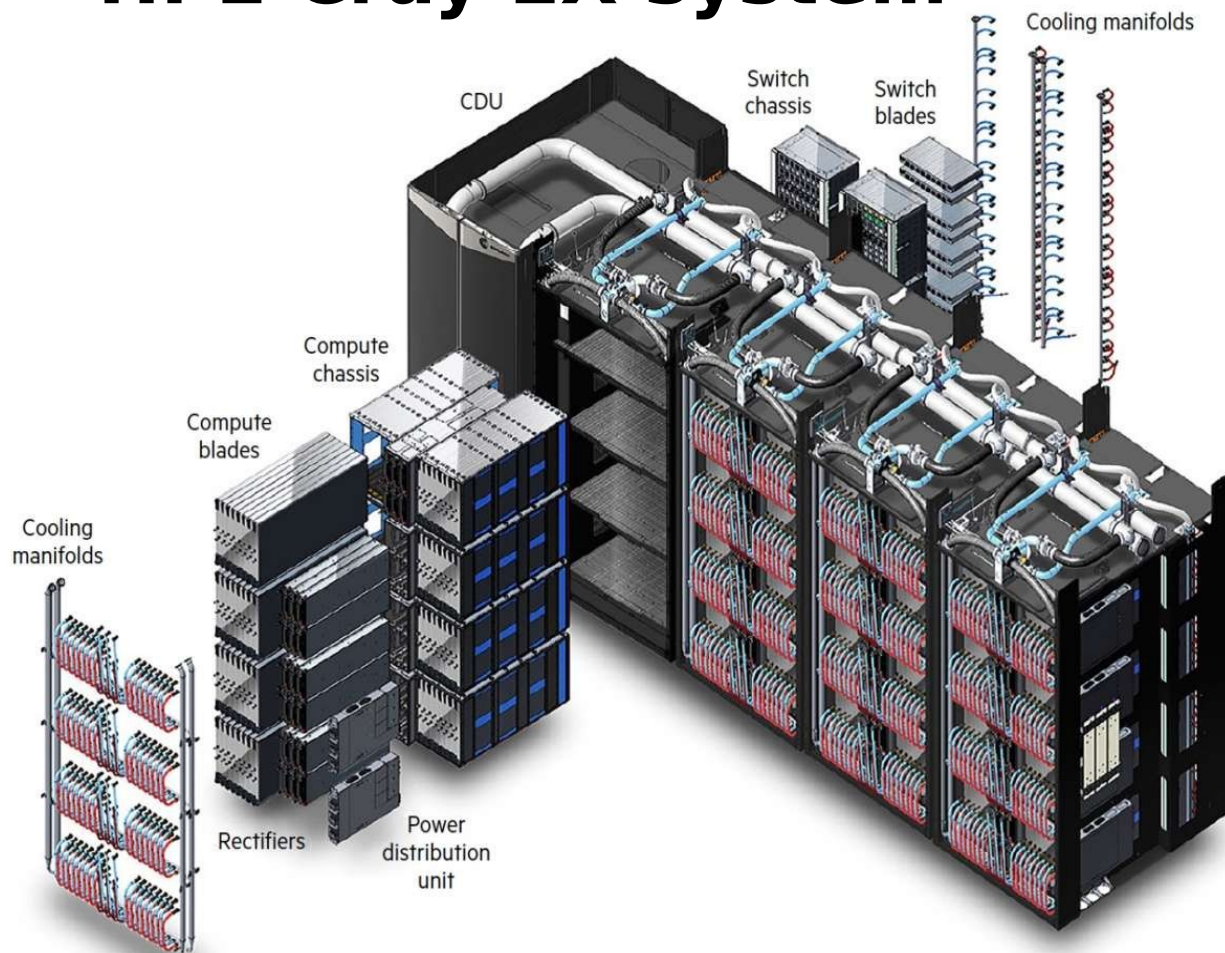
Architecture and key parameters

- LUMI-C: 1536 nodes with 2 64-core AMD EPYC 7763 CPUs (1376x 256GB, 128x 512 GB and 32x 1TB)
- LUMI-G: 2560 nodes with 1 AMD EPYC 7A53 CPU and 4 AMD MI250x accelerators (512 GB + 4x128 GB RAM)
- LUMI-F: 7 PB Lustre flash-based file storage (1740 GB/s)
- LUMI-P: 4 20 PB hard disk based Lustre file systems (4x 240 GB/s)
- Currently 4 user access nodes with an AMD Rome CPU
- All linked together with a HPE Cray Slingshot 11 interconnect
- Further developments, not yet available or fully operational:
 - Nodes for interactive data analytics: 8 4TB CPU nodes and 8 nodes with 8 GPUs each for visualisation
 - Open OnDemand environment
 - Object based file system

Slingshot interconnect

- Slingshot11
- 200 Gb/s (25 GB/s/dir) interconnect based on Ethernet but with proprietary extensions for better HPC performance
 - Adapts to Ethernet devices in the network
 - Lot of attention to adaptive routing and congestion control
 - MPI acceleration
- Dragonfly topology
 - 16 switch ports connect to nodes
 - 16 or 32 switches in a group with all-to-all connection between the groups
 - Groups are then also connected in an all-to-all way
 - Possible to build large networks where nodes are only 3 hops between switches away on an uncongested network

HPE Cray EX system



- LUMI-C
 - 1 network port/node
 - 4 nodes/compute blade
 - 2 switch blades/chassis
 - 4 nodes on a blade distributed over 2 switches
- LUMI-G
 - 4 network ports/node
 - 2 nodes/compute blade
 - 4 switch blades/chassis
 - 2 nodes on blade on other switch pair

LUMI site



LUMI-C Partition

LUMI-C : CPU compute

LUMI-G : GPU compute

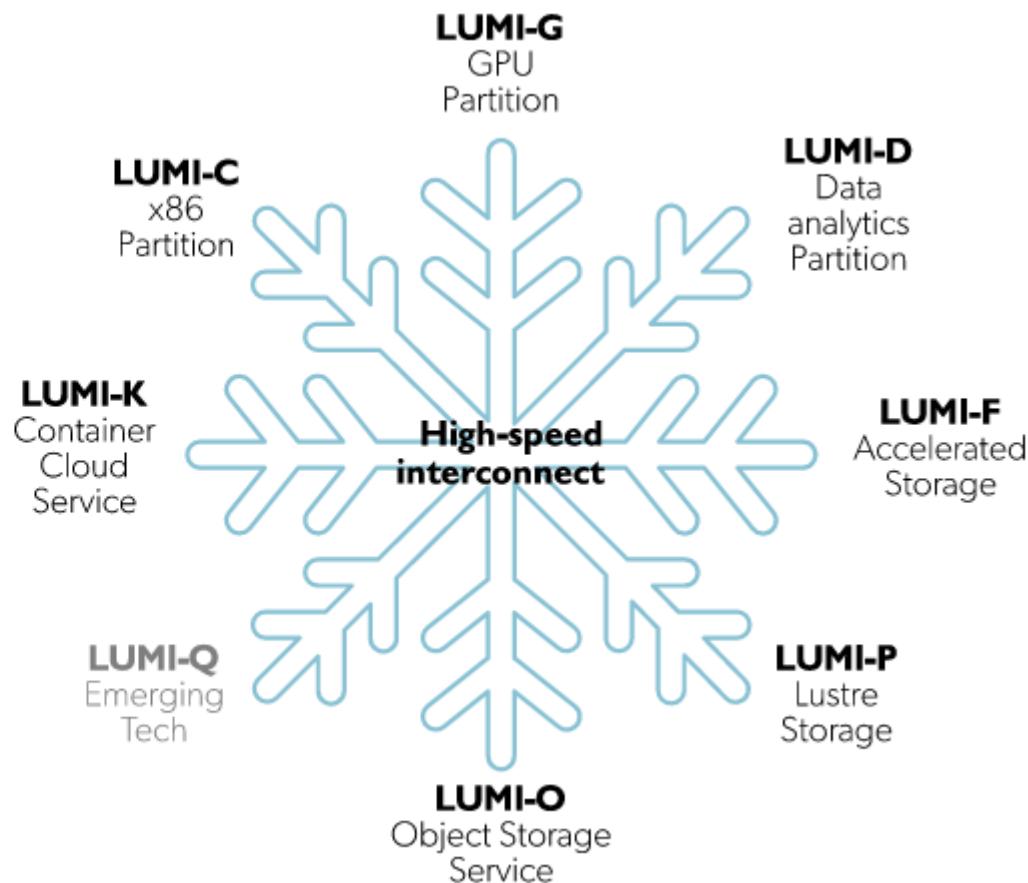
LUMI-D : Data analytics

LUMI-P : Parallel file system

LUMI-F : Flash-based parallel file system

LUMI-O : Object storage

LUMI-K : Container orchestration platform

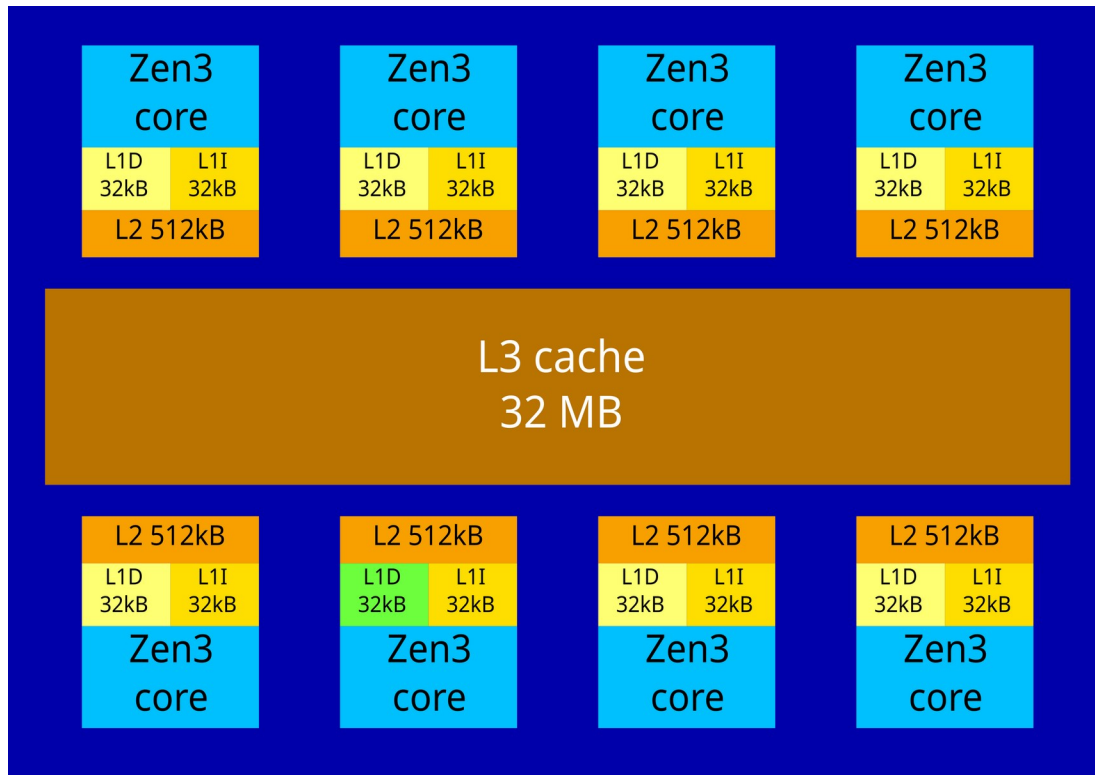


LUMI-C Partition

Nodes	CPUs	CPU cores	Memory	Disk	Network
1376	2x AMD EPYC 7763 (2.45 GHz base, 3.5 GHz boost)	128 (2x64)	256 GiB	none	1x 200 Gb/s
128	2x AMD EPYC 7763 (2.45 GHz base, 3.5 GHz boost)	128 (2x64)	512 GiB	none	1x 200 Gb/s
32	2x AMD EPYC 7763 (2.45 GHz base, 3.5 GHz boost)	128 (2x64)	1024 GiB	none	1x 200 Gb/s

The AMD EPYC 7xx3 (Milan/Zen3) CPU

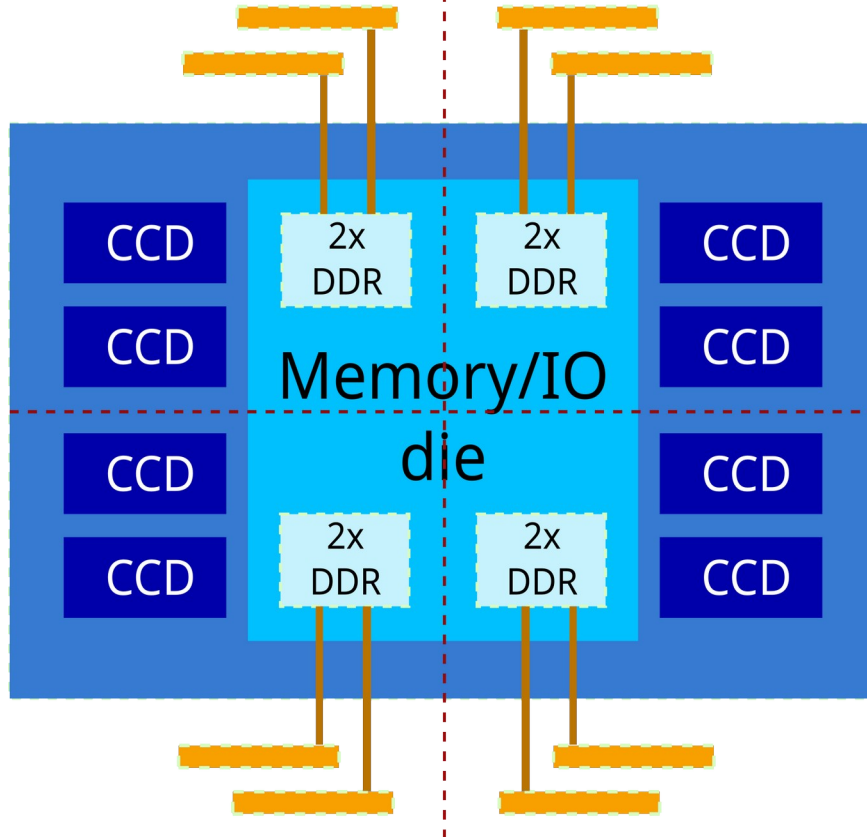
L U M I



- Building block: a Compute Complex Die (CCD)
- 8 cores
 - Each core has private L1 and L2 caches
 - L3 cache shared
- Instruction set equivalent to Intel Broadwell generation
 - AVX2+FMA, no AVX-512

The AMD EPYC 7xx3 (Milan/Zen3) CPU

NUMA node

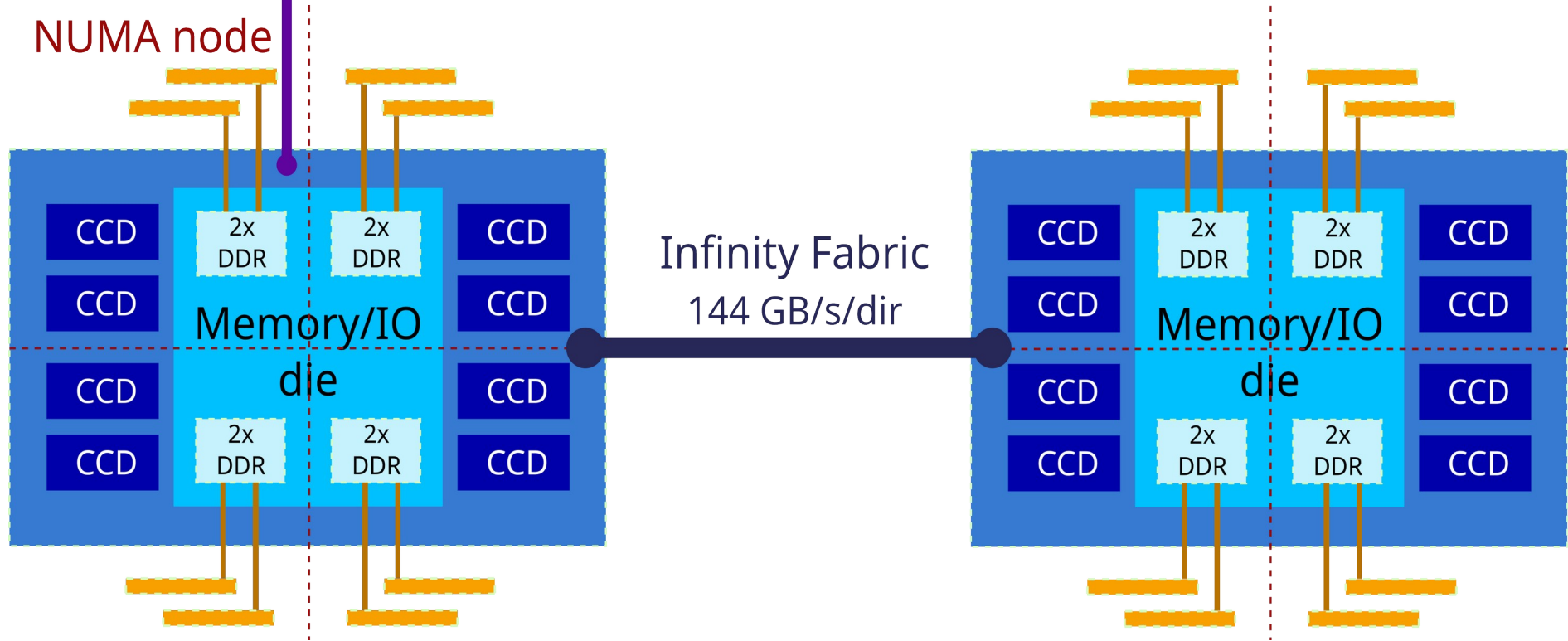


- 8 CCDs or 8 L3 cache regions
- Memory/IO die logically split into 4 NUMA domains with
 - 2 CCDs (16 cores)
 - 2 DDR4 controllers
- Memory/IO die also provides the PCIe links and intersocket links

LUMI-C node

200 Gbps Slingshot interconnect

NUMA node



Strong hierarchy

hierarchy layer		per	sharing	distance	data transfer delay	data transfer bandwidth
1	2 threads	core	L1I, L1D, L2			
2	8 cores	CCD	L3 Link to I/O die			
3	2 CCDs	NUMA node	DRAM channels (and PCIe lanes)			
4	4 NUMA nodes	socket	inter-socket link			
5	2 sockets	node	inter-node link			

Delays in numbers

		NUMA nodes CPU 1				NUMA nodes CPU 2			
		0	1	2	3	4	5	6	7
NUMA nodes CPU 1	0	10	12	12	12	32	32	32	32
	1	12	10	12	12	32	32	32	32
	2	12	12	10	12	32	32	32	32
	3	12	12	12	10	32	32	32	32
NUMA nodes CPU 2	4	32	32	32	32	10	12	12	12
	5	32	32	32	32	12	10	12	12
	6	32	32	32	32	12	12	10	12
	7	32	32	32	32	12	12	12	10

- NUMA behaviour not that pronounced within a socket
- but definitely something to take into account between sockets

LUMI-G Partition

LUMI-C : CPU compute

LUMI-G : GPU compute

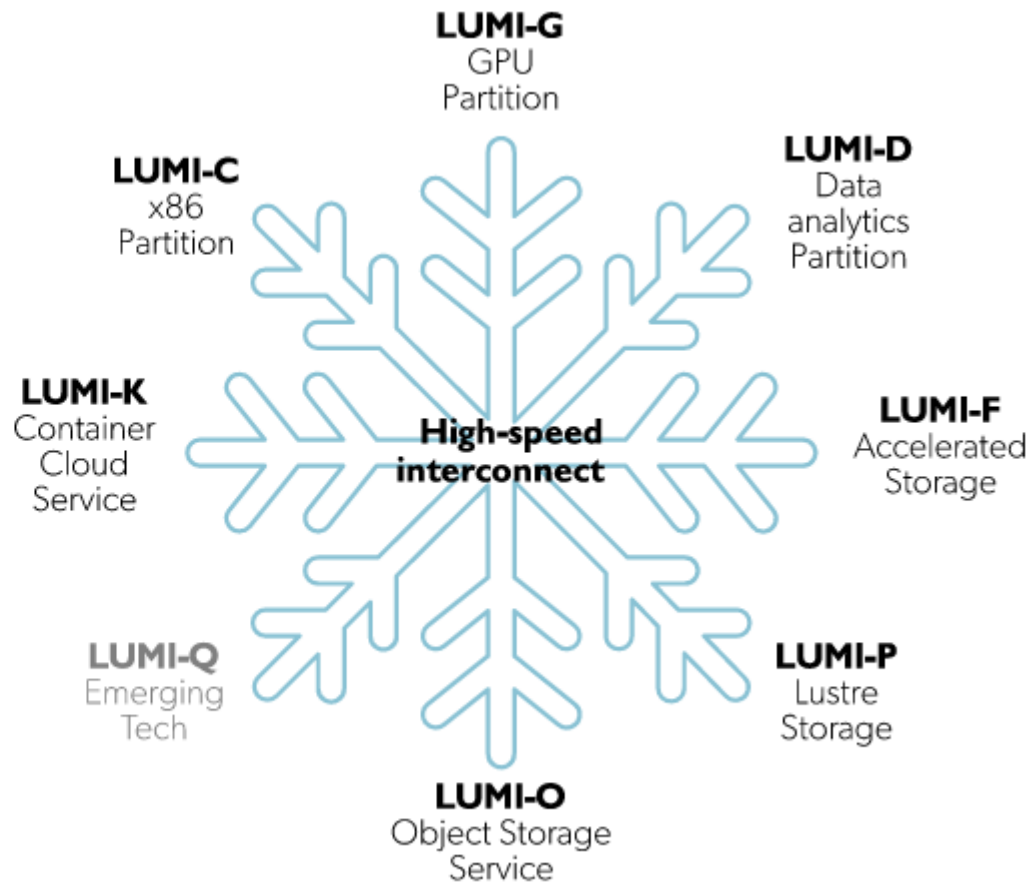
LUMI-D : Data analytics

LUMI-P : Parallel file system

LUMI-F : Flash-based parallel file system

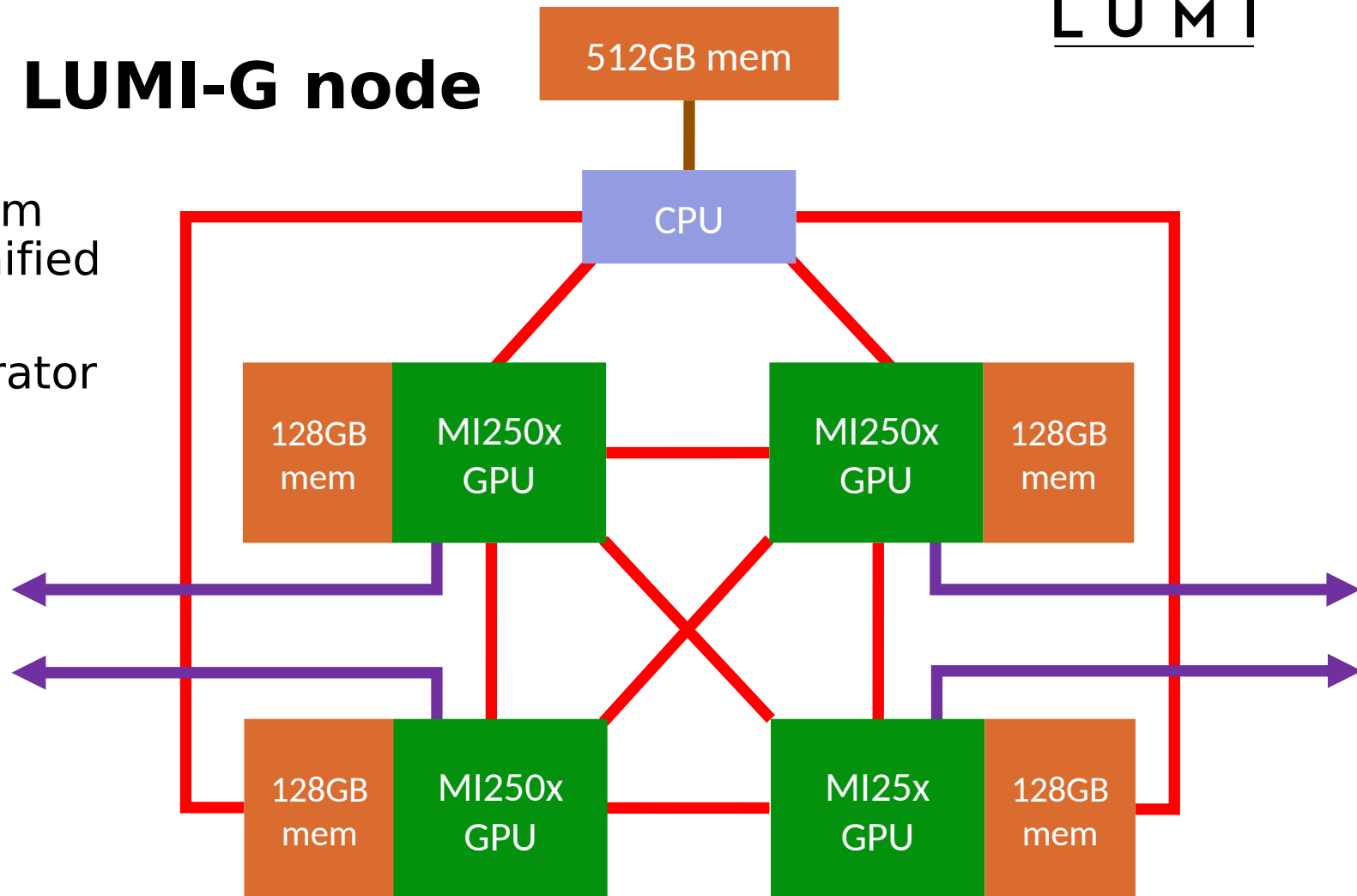
LUMI-O : Object storage

LUMI-K : Container orchestration platform



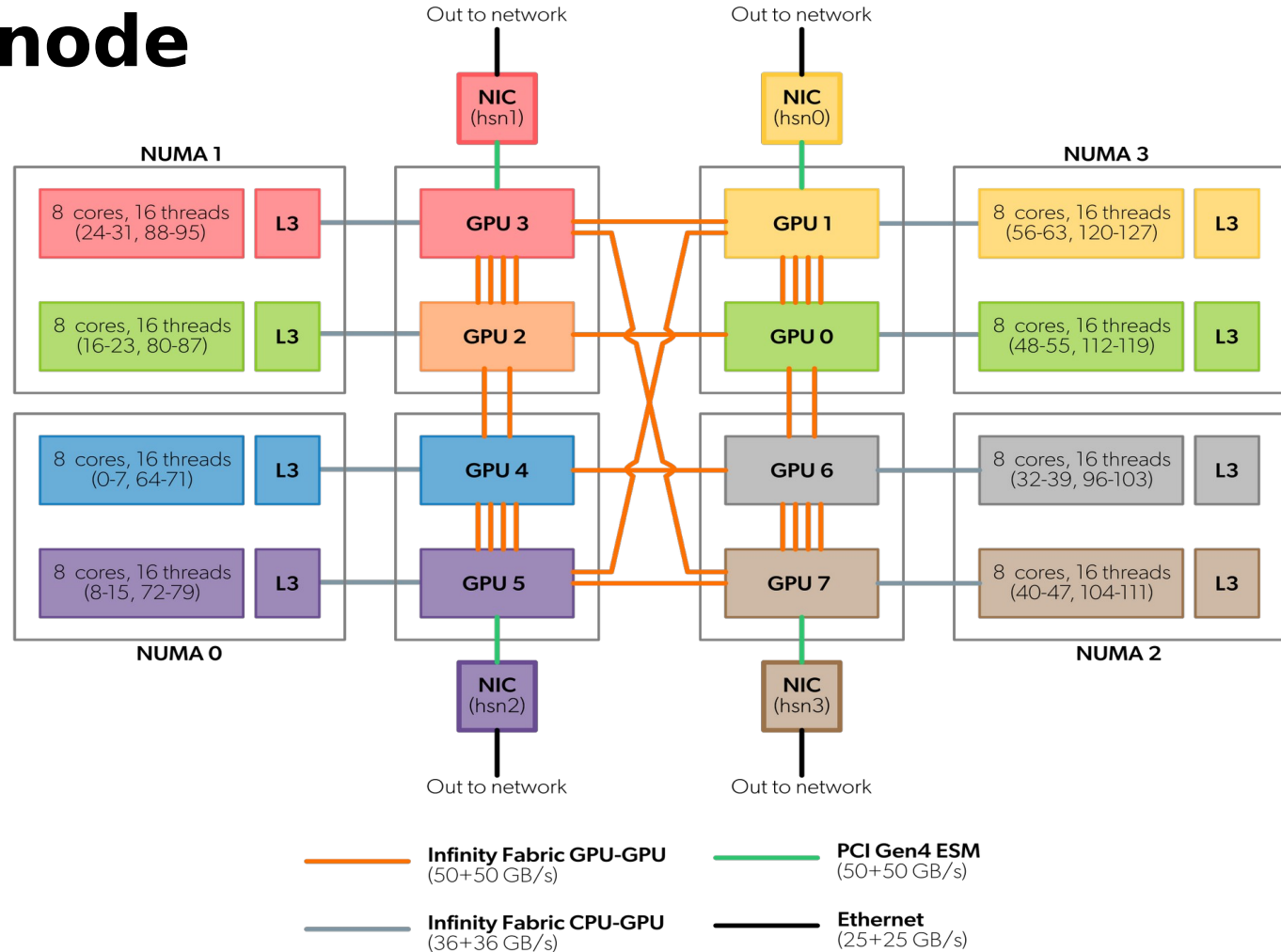
Concept LUMI-G node

- “GPU first” system with coherent unified memory
- Compute accelerator (CDNA2), not a rendering 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



Scratch and Project storage

LUMI-C : CPU compute

LUMI-G : GPU compute

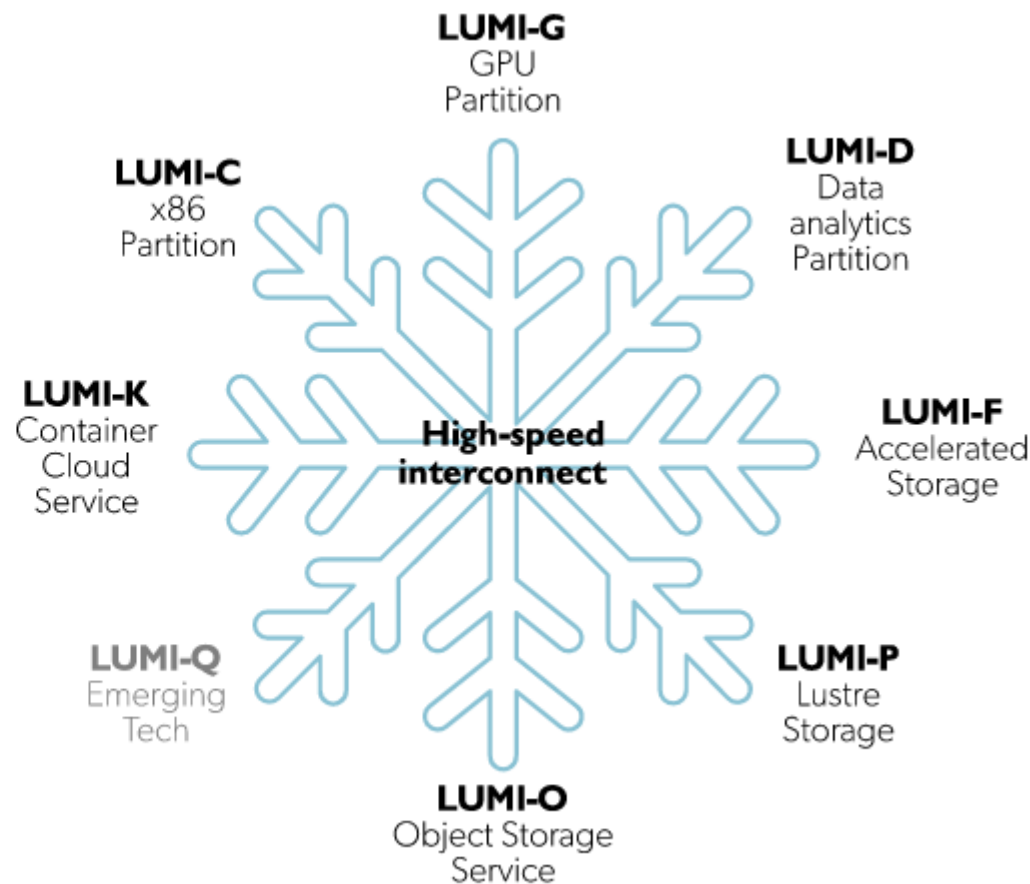
LUMI-D : Data analytics

LUMI-P : Parallel file system

LUMI-F : Flash-based parallel file system

LUMI-O : Object storage

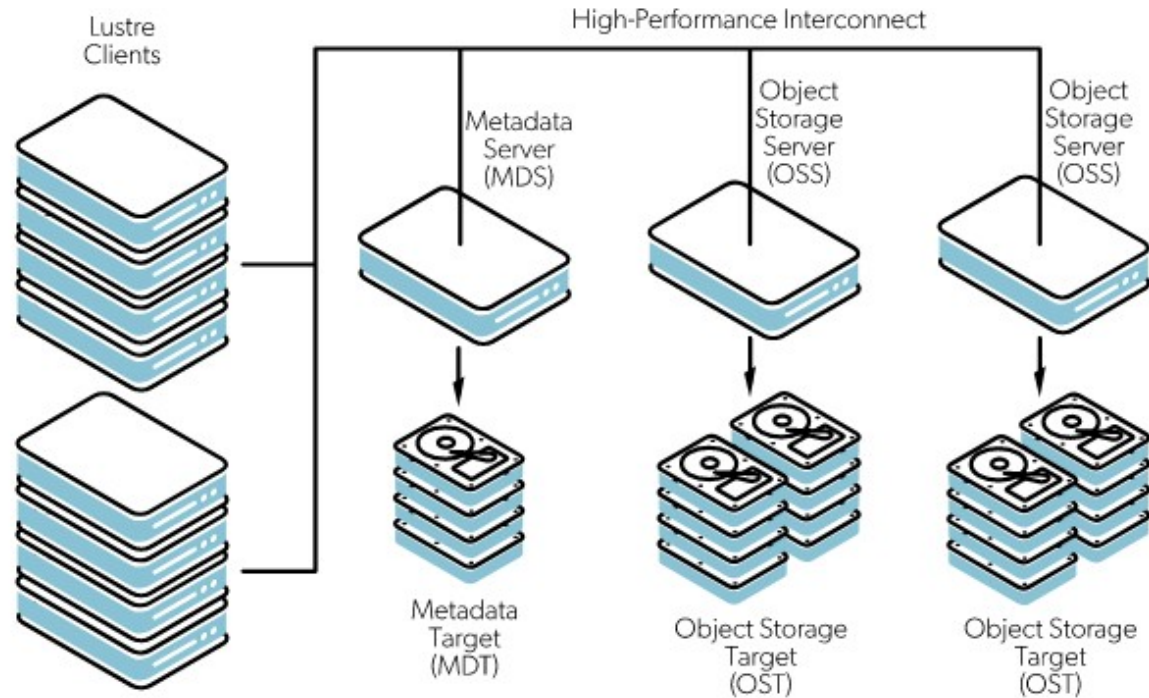
LUMI-K : Container orchestration platform



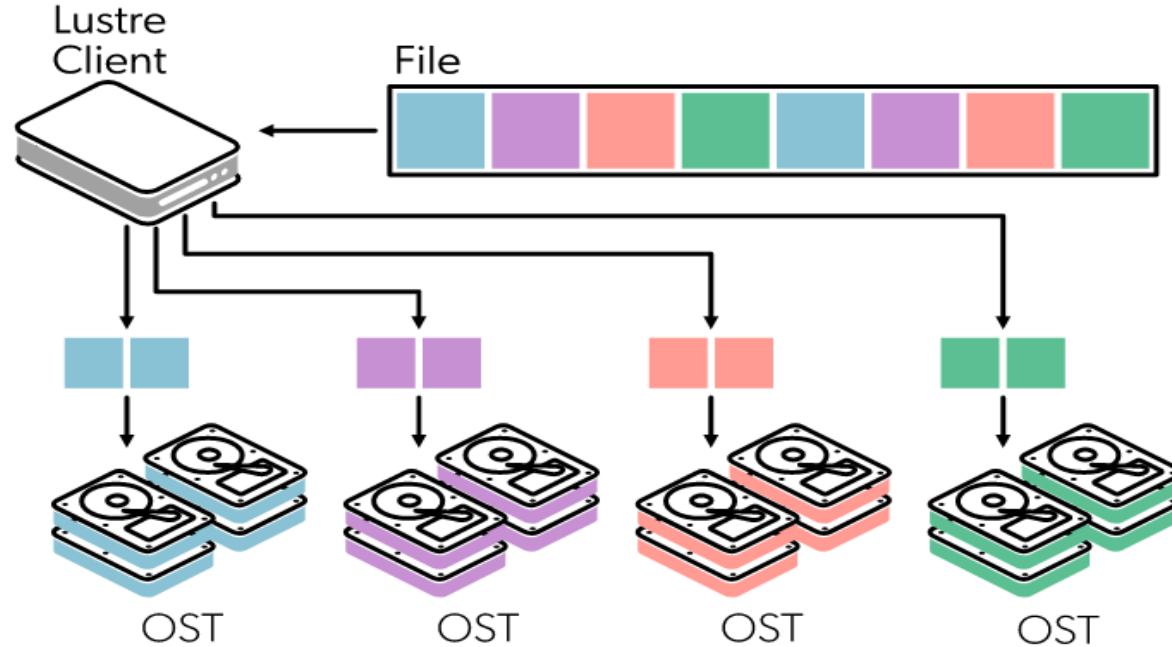
LUMI-P Overview

- 4 independent Lustre file systems
- Storage capacity of 20 PB each
- Aggregate bandwidth of 240 GB/s
- Each is composed of
 - 1 MDS (metadata server)
 - 32 Object Storage Targets (OSTs)(spinning disks)
- Avoid using a large number of small files

Lustre Building Blocks



LUMI-P striping



striping of a 8MB file over 4 OSTs (stripe count = 4). Each stripe is 1MB (stripe size = 1m) in size. Each OST store 2 stripes.

LUMI-P striping example

- `your_lumi_username@uan02:~>`
- `TESTDIR=/scratch/project_465000NNN/\`
`your_lumi_username/test`
- `mkdir -p $TESTDIR`
- `lfs setstripe --stripe-count=4 --stripe-size=2m`
`$TESTDIR`
- `touch $TESTDIR/file.txt`
- `lfs getstripe $TESTDIR`

LUMI-P striping example

lmm_stripe_count: 4
lmm_stripe_size: 2097152
lmm_pattern: raid0
lmm_layout_gen: 0
lmm_stripe_offset: 9

obdidx	objid	objid	group
9	41467721	0x278bf49	0
11	41450840	0x2787d58	0
13	41460249	0x278a219	0
15	41496704	0x2793080	0

LUMI-P storage areas

	Path	Intended use	Hardware partition used
User home	<code>/users/<username></code>	User home directory for personal and configuration files	LUMI-P
Project persistent	<code>/project/<project></code>	Project home directory for shared project files	LUMI-P
Project scratch	<code>/scratch/<project></code>	Temporary storage for input, output or checkpoint data	LUMI-P
Project flash	<code>/flash/<project></code>	High performance temporary storage for input and output data	LUMI-F

LUMI-P Quotas and Retention

	Quota	Max files	Expandable	Backup	Retention
User home	20 GB	100k	No	Yes	User lifetime
Project persistent	50 GB	100k	Yes, up to 500GB	No	Project lifetime
Project scratch	50 TB	2000k	Yes, up to 500TB	No	90 days
Project fast	2 TB	1000k	Yes, up to 100TB	No	30 days

LUMI-P as personal \$HOME

- Up to 20GB of data
- Data is deleted once account expires
- Backup is provided, but only until account expiry
- Set your own reminders to transfer data away before account expiry

LUMI-P as project home

- Shared data among the members of a project
- E.g. to share applications and libraries compiled for the project
- Located at /project/project_<project-number>
- Data is purged once the project expires

LUMI-P project scratch

- Main runtime data storage, such as:
- Temporary storage for input, output, or checkpoint data of your application
- Do not use as long-term storage
- Purged after 90 days

LUMI-P Billing

- $\text{TB-hours-billed} = \text{storage-volume} \times \text{time-used}$
- $1.2 \text{ TB} \times 24 \text{ hours/day} \times 4 \text{ days} = 115.2 \text{ TB-hours}$
- Use `lumi-allocations` cmd to check current state of available and used resources
- Cost of LUMI-F storage units is 10x

Resources

- More information:

<https://docs.lumi-supercomputer.eu/firststeps/getstarted/>

- Helpdesk: LUMI User Support Team (LUST) :

<https://docs.lumi-supercomputer.eu/helpdesk/>

Introduction of LUMI supercomputer



Presented by Jan Vicherek, IT4I