LUMI

An overview on storage systems

Speaker: Emanuele Vitali – CSC, LUST

Overview

- General considerations on storage for an HPC system
- Design choices
- LUSTRE
    - What is
    - How it works
    - Usage
- LUMI-O
    - What is
    - How it works
    - Why
- Comparison
- Wishes for the next system

Storage systems

- Storage is a core performance component of a machine like LUMI: it needs to be big enough to accommodate thousands of user, running in parallel.
- It is a shared resource and every user (mis)use directly impact all the others
- No single storage fits every workflow (and this is becoming more and more true with the rise of AI and python in HPC)
- There are no local disks on compute nodes, they only have /tmp ramdisk

Design:

- Back when LUMI was designed, AI was not as big as now, so it was not one of the drivers of the filesystems of LUMI
- We have 3 different storages on LUMI:
  - LUMI-P(lustre)
  - LUMI-F(lustre)
  - LUMI-O(ceph)
- They have different goals: lustre is to serve the computation, ceph to improve data transfer, offer some backup, serve the dataset optimized for object storage (.zarr and similar format)

Lustre: what is

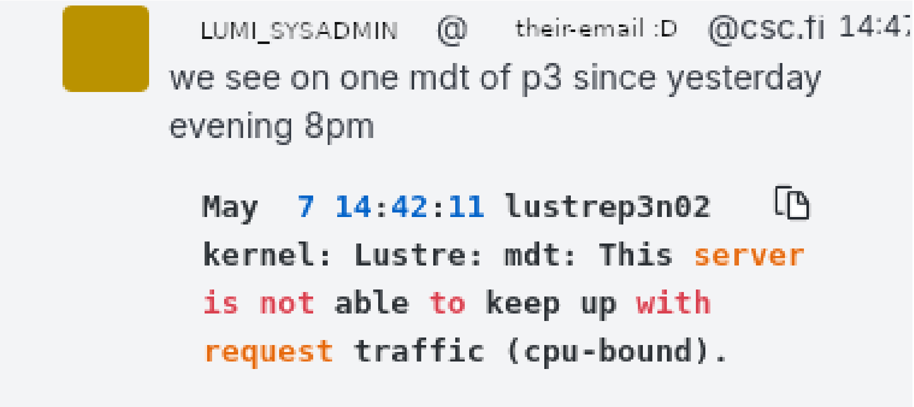
- The main filesystem of the machine.
- Well established open source middleware.
- LUMI-P: 4 * 18.2 PiB HDD @ 240GB/s
  - Divided into 4 parallel FS:
    - Mitigate MD pressure
    - Mitigate FS failures
  - The “bulk storage”
- LUMI-F: 1 * 8.5 PiB Flash @ 2TB/s
  - Workloads heavy in terms of IOPS or BW
- No backup. User responsible of saving the data they need outside of LUMI.

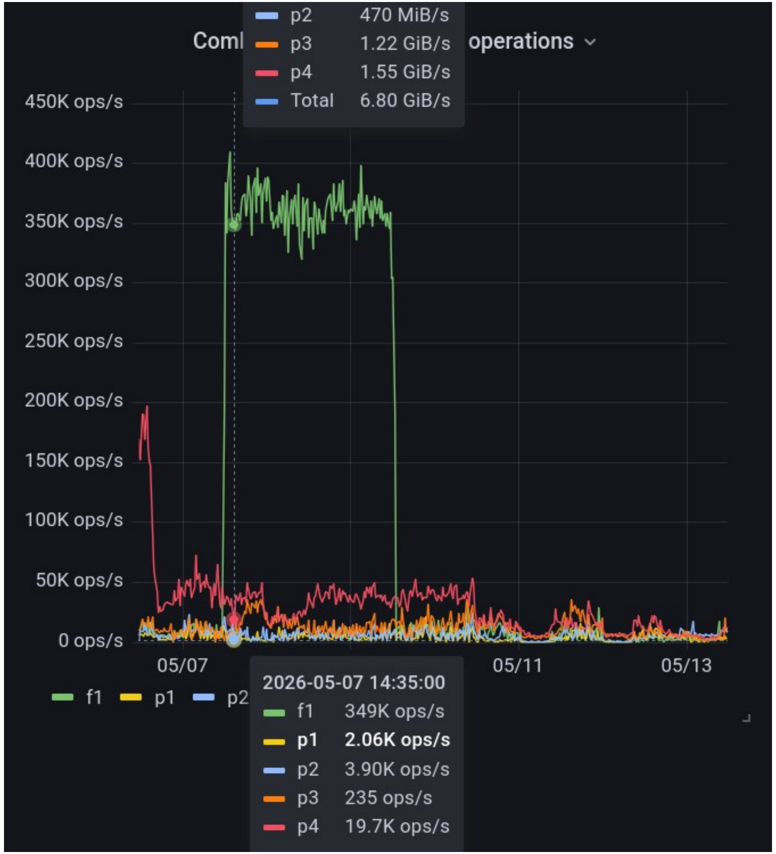
Lustre: how it works

- Lustre separates data and metadata as both are used differently
- Metadata servers (MDSes) with one or more metadata targets (MDTs) each store namespace metadata (filename, access permissions, …) and file layout.
- Object storage servers (OSSes) with one or more object storage targets (OSTs) each store the actual data.
  - Capacity of Lustre is the sum of the capacity of the OSTs
- Lustre clients that access and use the data and makes the whole Lustre setup look like a single large file system
  - Transparent in functionality: You can use it as any regular Linux file system
  - But not transparent in performance: How you use Lustre can have a huge impact on performance
- All linked together through the high performance interconnect.

Lustre usage

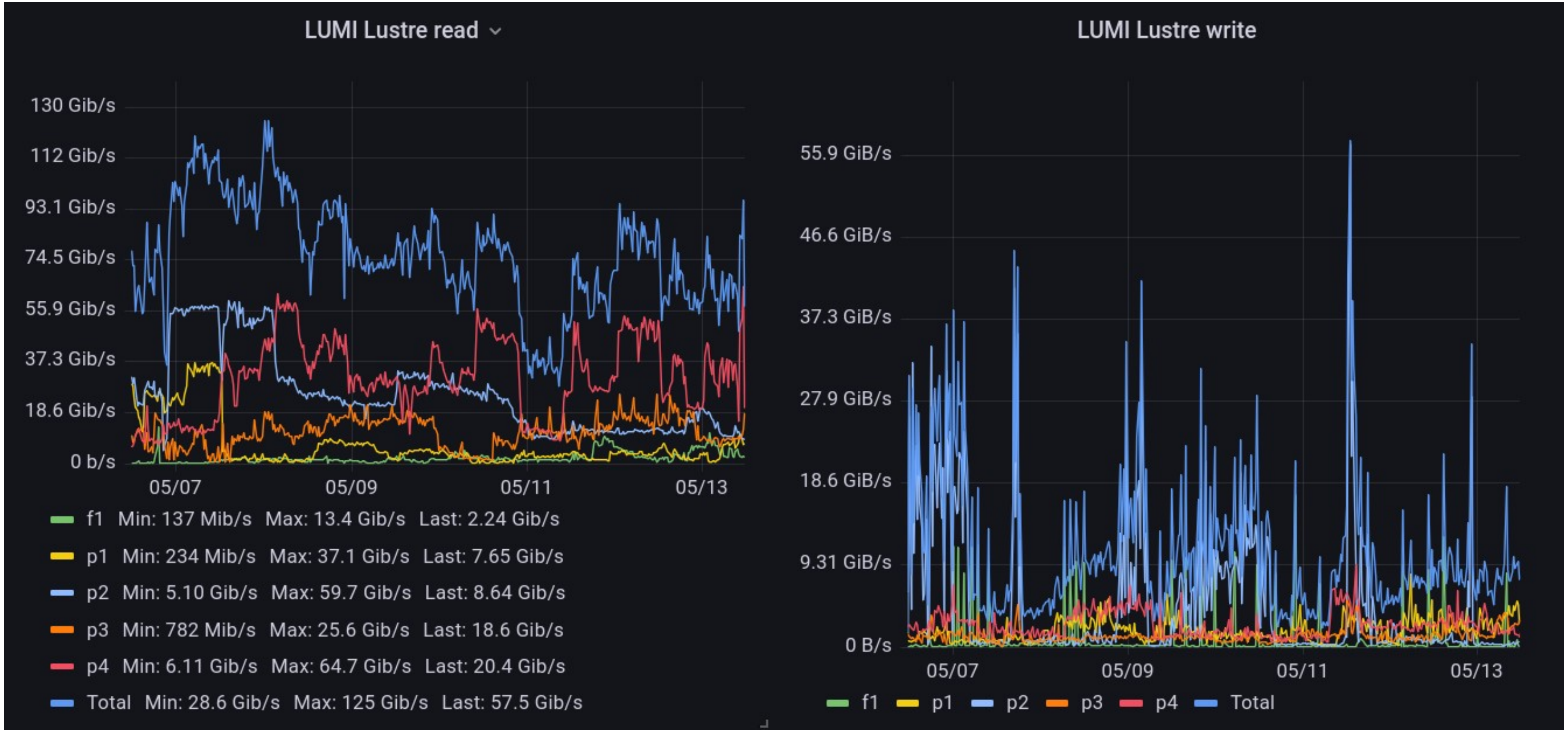
- Typically users are not well behaved with Lustre: huge pressure on MDS
- This is more common than what you may think





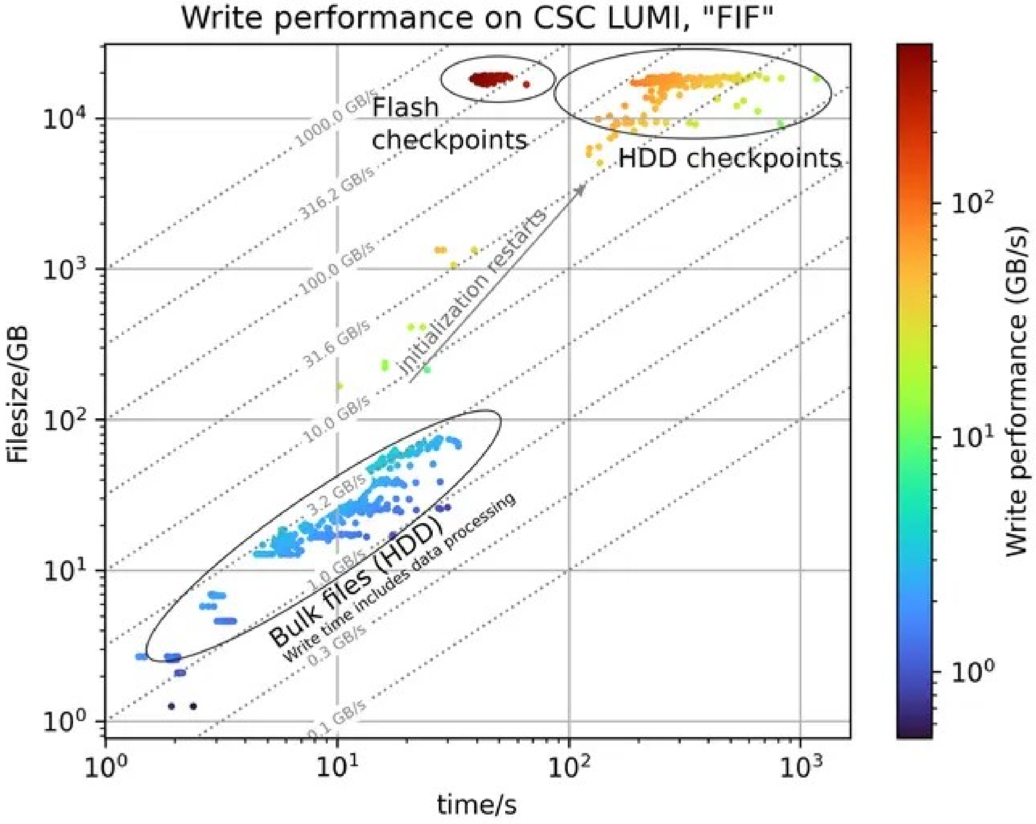
Lustre usage

- Average usage (in terms of BW) is not close to limits:



Lustre usage

- User who can correctly set parameters can reach very good throughput with HPC workloads
- This image comes from Vlasiator team: they are able to reach ~1/4 of nominal bandwidth on both LUMI-F and LUMI-P



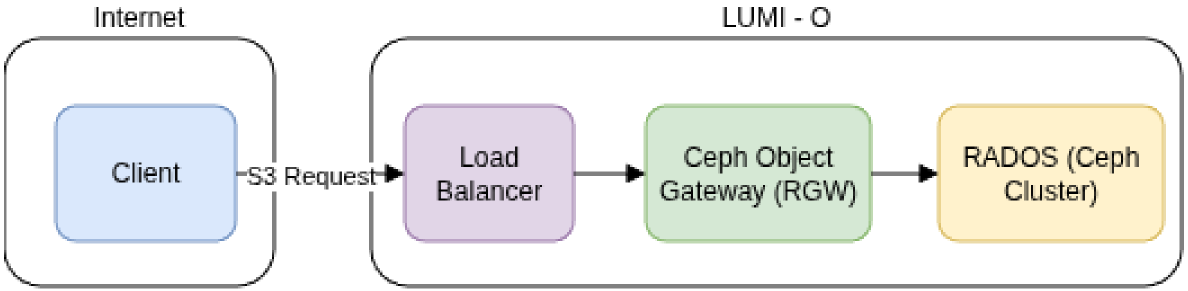
Throughput achieved with Vlasiator on LUMI (credits: Alho&Suni - Vlasiator team)

LUMI-O: what is

- Object storage system, based on Ceph
- Organisation: 3 levels:
  - Projects: LUMI projects
  - Buckets: “Containers” used to store objects.
    - Flat structure: Buckets cannot contain other buckets
  - Objects: Any type of data, stored in a bucket
    - Atomic access to objects: Put, get, copy, delete, …, but no partial write
- It is not integrated in the cluster via slingshot, but is a separate entity.
- Specs:
  - Capacity: 30 PB
  - Current quota: 150 TB capacity, 1K buckets and 500K objects per bucket
  - Persistent for the duration of the project

LUMI-O: how it works

- Ceph Object storage is S3 compatible Object storage built on top of a Ceph cluster.
- A Ceph cluster is built out of a RADOS (Reliable Autonomic Distributed Object Storage) cluster and a set of RADOS clients.
- The Ceph Object Gateway (also called RADOS Gateway) is a server that acts as an S3 API server towards the internet and as a RADOS client towards the storage cluster, i.e. a gateway between the Internet and RADOS
- A load balancer in front of this handles the incoming requests and directs them to the Gateway



LUMI-O: why

- Great for transporting data between different research sites and for sharing data with other users
  - LUMI-O is open to the internet and provides configurable access policies for your data
  - S3 is a web API: in many cases a browser is enough for fetching shared/public data
- (Temporary) back up of project research data
  - Data stays for the lifetime of projects – no cleanup operations for data that sits there while the project is active
  - S3 backends supported by many backup-programs (e.g. restic)
- Works as a storage backend and data streaming framework for big data analytics, machine learning, AI
  - Supported by Hadoop, Spark and many others
  - Workflows moving from cloud often already use S3 so there is less reworking the code

Lustre vs Lumi-O

Lustre	LUMI-O object storage
Integrated in the authentication of the supercomputer	Separate key-based authentication mechanism
Seen as any other POSIX file system	Separate range of access tools/APIs, some tools can provide a filesystem view
Closely integrated with compute nodes	Separate system
Upgrades with the system	Separate upgrade cycle and on-the-fly
No external access	External access integrated, includes web
Parallelism for performance: Access a file in parallel from multiple processes	Parallelism for performance: Different processes access different objects
Fairly expensive to very expensive hardware	Cheaper hardware

Lustre vs Lumi-O (cont.)

Lustre	LUMI-O object storage
Organisation: Hierachical directory structure and files	Organisation: Triple flat space of projects, buckets and objects
Files can be read, written, modified, appended, …	Simple atomic operations on objects: put, get, copy, delete
Optimised for bandwidth to the compute nodes	Optimised for reliability, resilience, global access
Simpler schemes for redundancy	Very complex internal redundancy setup

LUMI next: wishes

- Storage system where the user needs to know less to get good performance.
  - supercomputer filesystem to look and behave more like the SSD in users laptops
- Size similar to current
- Separate storage tier targeting specific usecases:
  - IOPS Intensive
  - Fast Objects
  - Huge Bandwidth
- Buying it 9 months ago :(

Follow us

LinkedIn: LUMI supercomputer

**Bluesky: **@lumi-supercomputer.eu

YouTube: LUMI supercomputer

www.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 Participating States FI, BE, CH, CZ, DK, EE, IS, NL, NO, PL, SE.

Leverage from the EU 2014–2020

European Union European Regional Development Fund

REGIONAL COUNCIL OF KAINUU

LUMI-O strength and weaknesses

- Strength:
  - Durability and fault tolerance: data is erasure-coded, the algorithm automatically distributes it across available nodes and there is no single point of failure
  - Not part of the system: It can be maintained online, no need for downtime to add/remove nodes, upgrade nodes, …
  - No metadata bottleneck

- Weakness:
  - Not part of the system: latency to retrieve data is higher
  - No POSIX: user need to use s3 protocol to retrieve the data. S3fs can mount it as a POSIX compliant filesystem, but the performances won’t be optimal. In general, user needs to do some extra work to use it “normally”.
  - Doesn’t perform well with “few” huge files

Lustre strength and weaknesses (NTS keep or not?)

- Strength:
  - Extremely high throughput for large I/O operations; ideal for large files and sequential access patterns.
  - Scales to thousands of nodes writing concurrently.
  - Users can tune striping parameters for performance.

- Weakness:
  - Not good for many small files since the metadata operations do not scale well.
  - Certain I/O operations are slower compared to a local SSD.
  - Misconfigured striping or too many simultaneous small-file operations can degrade service for other users.