LUX PROVIDE

HIGH PERFORMANCE
Computing & Data Analytics

Robert de Rooy, Senior HPC Engineer, Data Services

Mission

We **accelerate economic** and **social progress** through **supercomputing.**

Vision

To be recognised as **the European centre of excellence** in **HPC/HPDA** services with the **easiest onboarding** and **highest quality assistance**, in a **confidential, trusted** and **cyber-secure environment**, with a clear focus on Luxembourg's seven priority areas of sectoral expertise (Financial Services Logistics ICT Manufacturing Eco-technologies Space Health technologies).

EuroHPC
Joint Undertaking

EURO

LUX PROVIDE

MELUXINA
HIGH PERFORMANCE COMPUTING IN LUXEMBOURG

MeluXina – Storage

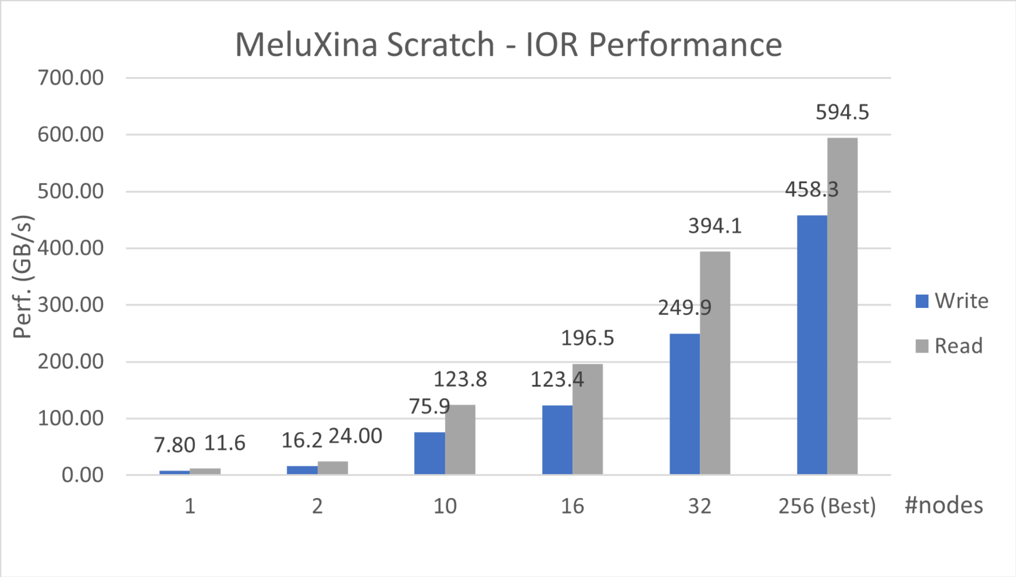
Scratch

DDN SFA400NVXE

- 12 building blocks
- 32 GB/s per block
- 50 TB addressable per block
- Lustre FS
- ~ 0.6PiB usable

Fabric connectivity

- 4x IB HDR100 per block



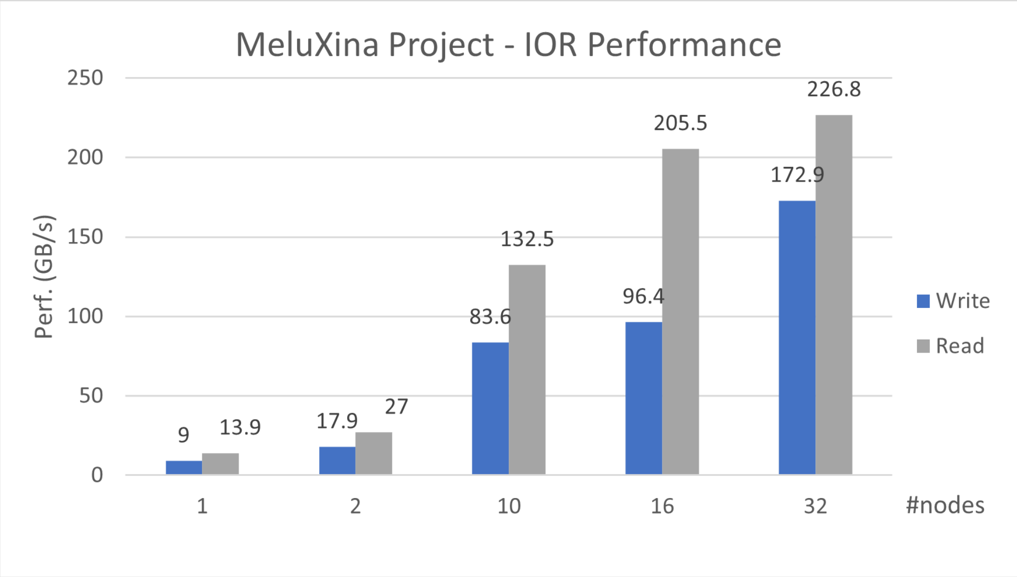
Project / Home

DDN SFA7990X + SFA400NVXE (MD)

- 10 building blocks + metadata
- 19 GB/s per block
- 1.33 PB addressable per block
- Lustre FS (EXA 5.2.4, Lustre 2.12.8)
- ~ 12PiB usable

Fabric connectivity

- 4x IB HDR100 per block



Backup

DDN SFA7990X + SFA200NVXE (MD)

- 2 building blocks + metadata
- 15 GB/s per block
- 3.5 PB addressable per block
- Lustre FS
- ~ 6.7PiB usable

Fabric connectivity

- 4x IB HDR100 per block

IBM TS4500 library

- 8x LTO8 drives
- 420x 12TB LTO8 cartridges
- ~4.5PiB Raw

Fabric connectivity

- 16x 8Gb FC

LUX PROVIDE

Lustre Snapshots

LUX PROVIDE

LAD22
LUSTRE ADMIN & DEV WORKSHOP
Paris 26-28.09.2022

Snapshot requirements and wishes

Requirements

Filesystem snapshot creation and deletion
Multiple snapshots for at least a week's worth of data
No noticeable impact to user operations during snapshots
Minimal performance overhead

Future wish list

Provide an easy way for users to access snapshot data
Allow partial filesystem snapshot

The Bad, The Ugly

Snapshot support for MeluXina capacity tier required by tender

Lustre feature originally planned by DDN for late 2020 availability
→ Various delays, missing functionality and operational issues

Accepted installation in July 2022 after co-design and testing
→ **Deep technical work & partnership essential**

The Good

After initial issues, DDN prepared an action and test plan

→ Set up a near identical system (MDT/OST count & OST size vs MeluXina) for in-lab development and testing

→ Provided remote access to LXP for testing in Jan 2022

Weekly calls with DDN developers and management

Multiple issues found by DDN and LXP, solved & improved upon:

- *lctl snapshot_destroy* now asynchronous, with new options
- *lctl snapshot_create* now defaulting to *barrier=off*
- New *lctl snapshot_stat* functionality

More performance improvements planned for future EXAOS updates

Testing & benchmark methodology

Core needs

Ensure platform **stability**: no crash especially during common operations (*create, delete*, *mount*)

Ensure platform **usability**:

→ no interruptions during user access (Home directories)

→ no application impact (crashes or corruption)

Ensure **performance**: minimal difference *with* vs *without* snapshot in place

Tests and benchmarks

Synthetic I/O using **mdtest**

Application checks using **GROMACS, OpenFOAM** & **PyTorch**

MDTest

Scripted run, using 7 nodes (560c) with Fujitsu A64FX CPUs & Mellanox interconnect

Loop 5x: *snapshot_create* + run *mdtest* at 0, 10, 30, 60, 120 min intervals

Loop 2x: *snapshot_destroy* for oldest snapshot + run *mdtest* at 0, 10, 30, 60, 120 min intervals

Purge remaining snapshots

Verification of stability, usability & performance

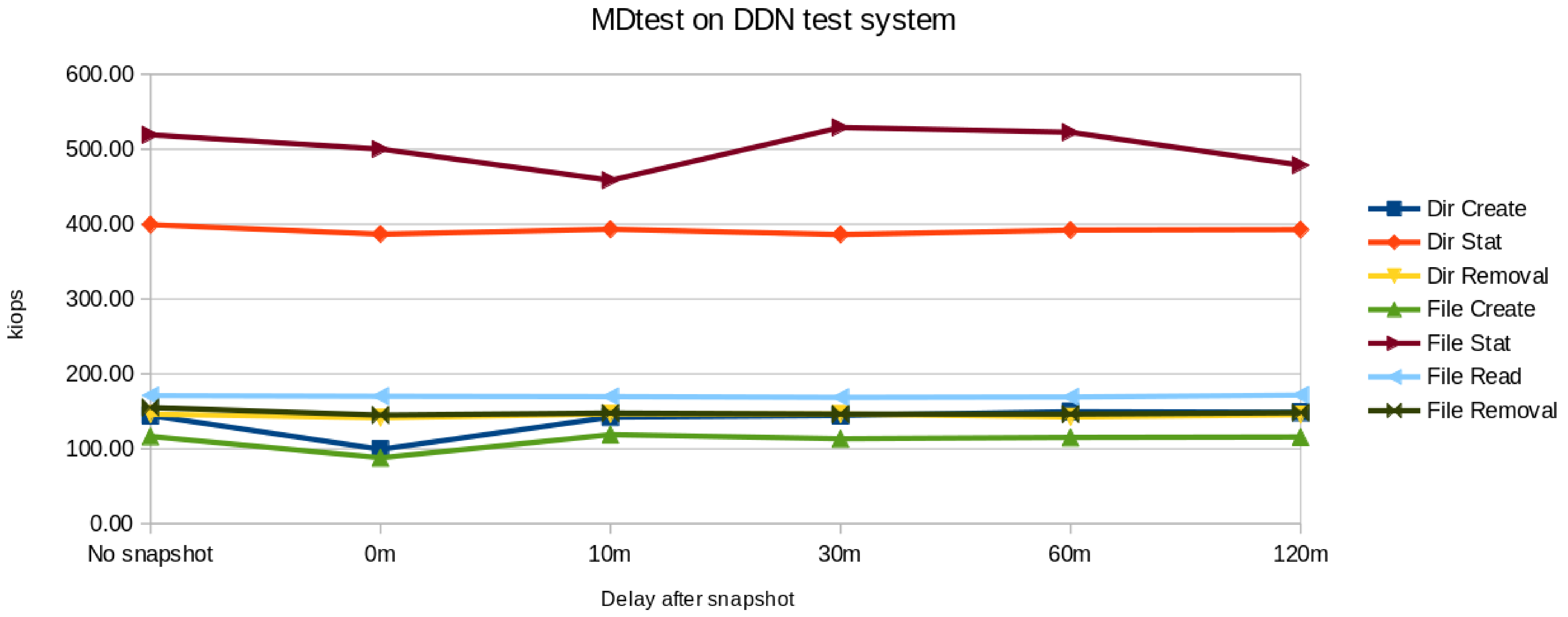
Snapshot performance – mdtest*

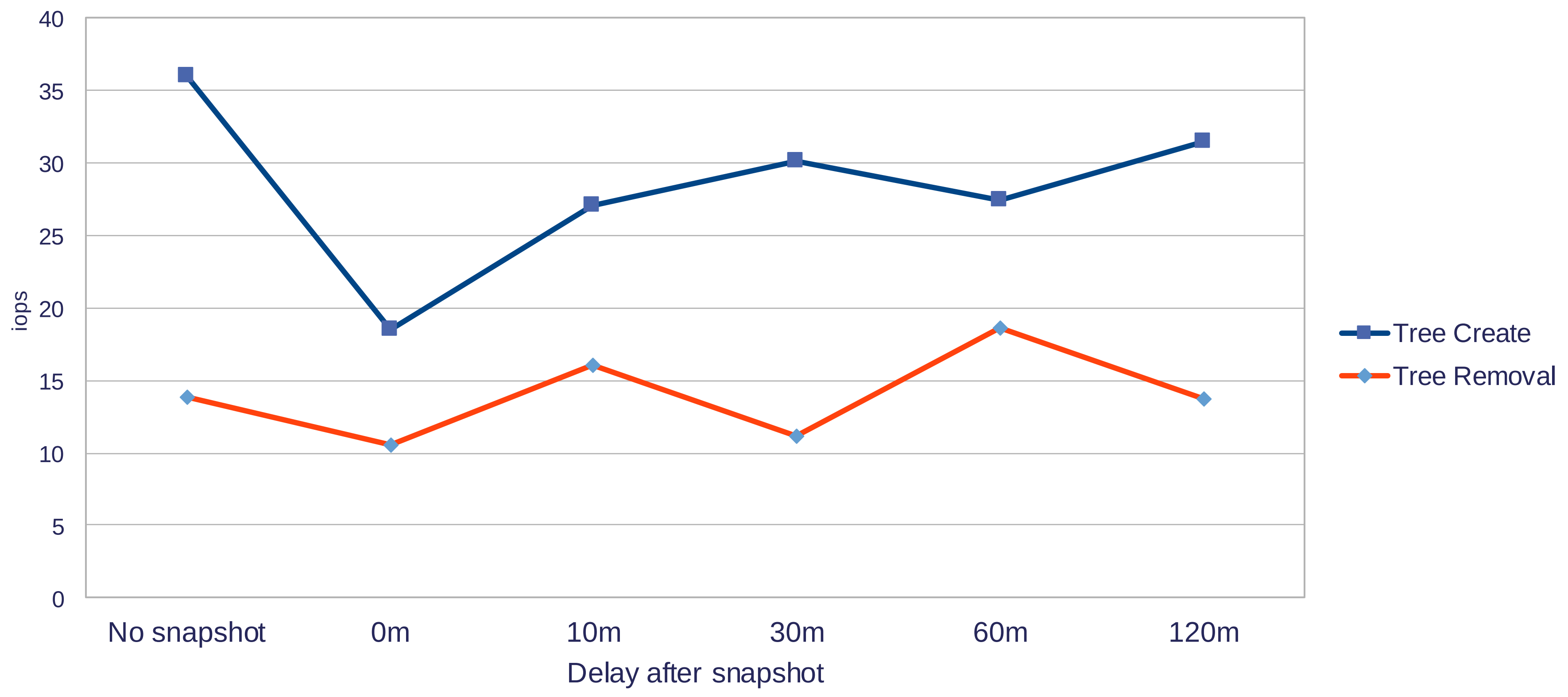
Storage / #Snapshots	Directory (kiops/s)			File (kiops/s)				Tree (iops/s)	
	Create	Stat	Removal	Create	Stat	Read	Removal	Create	Removal
No snapshot	144.42	399.12	146.23	116.61	519.03	171.09	155.12	36.06	13.91
1	139.52	393.80	143.91	111.01	525.62	170.84	146.33	31.58	12.32
2	135.96	393.76	141.96	110.42	509.48	169.91	146.78	27.17	13.83
3	141.57	390.88	145.16	111.26	501.55	170.43	146.60	29.49	13.68
4	136.91	390.03	144.50	110.31	497.72	169.95	146.75	26.95	14.02
5	138.33	389.86	142.88	110.68	514.95	168.76	146.78	30.33	12.28
1st del+create	139.57	390.19	143.29	110.17	502.71	169.56	146.73	27.11	13.61
2nd del+create	140.87	389.81	144.86	111.90	507.27	170.73	145.06	33.45	14.09
Std. deviation	2.70	3.26	1.37	2.13	9.51	0.77	3.13	3.31	0.74
Diff with/without	96.22%	98.01%	98.33%	95.03%	97.97%	99.38%	94.40%	81.66%	96.39%

*** results from DDN test lab system with only 2 MDS**

Interesting tidbits: Lustre perf. recovery

After ***snapshot_create***
→ is the FS perf. impacted?
→ how much and for how long?





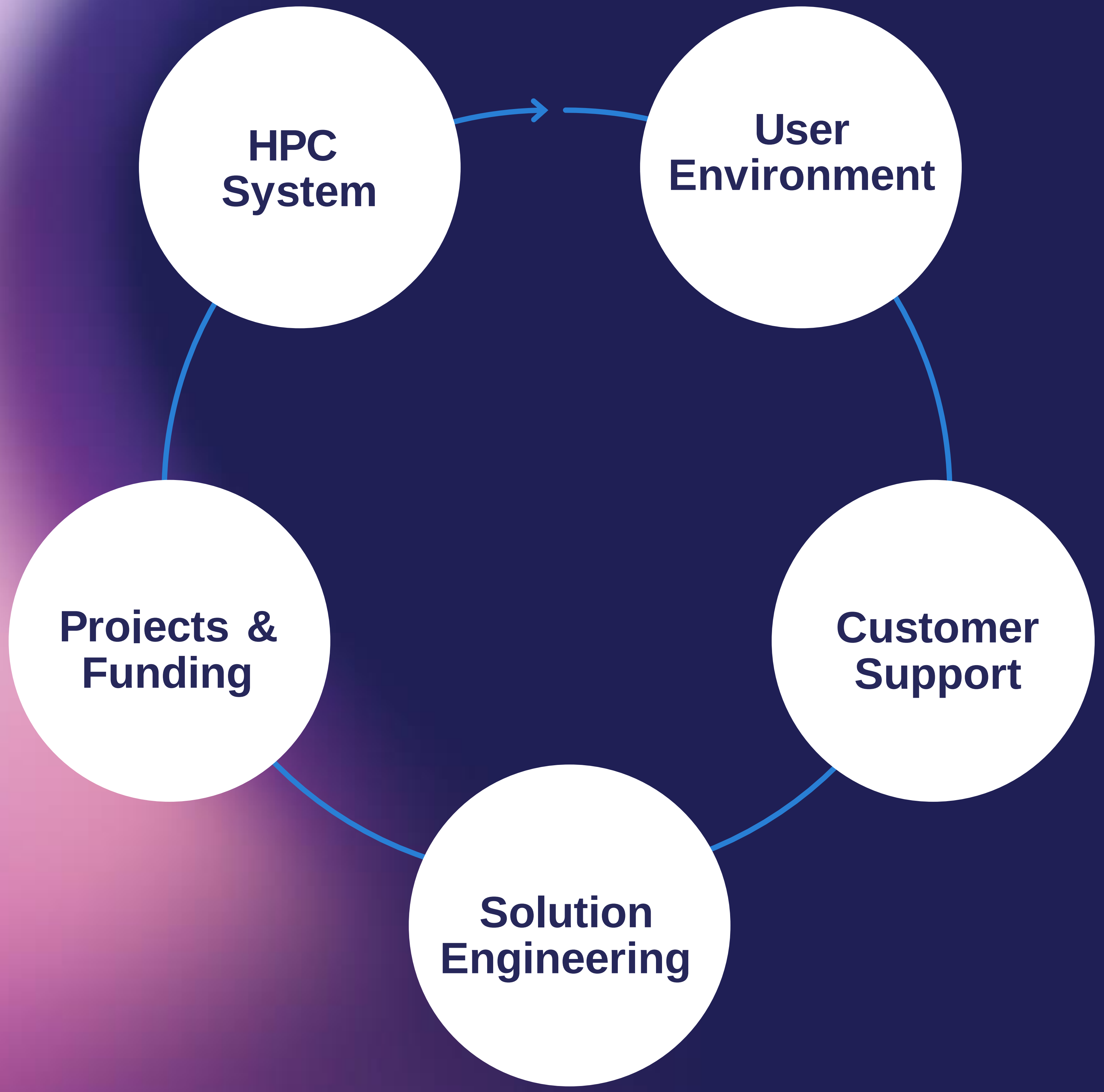
How much and for how long…
→ on which operations?

LUX PROVIDE

Luxembourg's one-stop shop for

high performance

computing and data analytics



Backup slides / Tech details

Test configurations

- Configuration differences

	#MDS	#MDT	#OSS	#OST	#OST/OSS	OST Size(TB)
Tier2 (Melluxina)	4	4	20	40	2	300
Test system(DDN)	4(collocate)	4	4	33	8or9	540

- Preparations
  - 50% capacity (8.1/17.4PB) and inodes (340/560M inodes)
- Test workloads
  - 32 jobs (10 takes per job) write/read 5 x 50GB files and remove old files randomly
  - 4 jobs (50 tasks per jobs) create new 1M files with various file size and remove 1M old files randomly
  - Jobs are continuously repeating
  - New snapshot is taken every 20mins up to 14 snapshots. After 14 snapshots, remove the oldest snapshot and take a new snapshot (We are assuming daily snapshot for two weeks)!

Improvements

- Performance improvements for snapshot delete
  - Prefetch truncate indirect blocks (EX-5068)
    - Speedup deletion process 512sec to 140sec (with active client IOs)
    - Patch completed and merged
  - Asynchronous snapshot delete operation (EX-5068)
    - ‘lctl snapshot_destroy’ returns immediately and trigger truncate process in separate thread as a background process
    - lctl snapshot_destroy returns in few seconds (previously 140 sec)
    - Feature implemented and started to test. Patch going to be merged.
  - User space tool (‘lctl snapshot_destroy’ command) improvements for snapshot delete
    - Support new command “lctl snapshot_stat” and few options in ‘lctl snapshot_destroy’
      - Show background truncate progress
    - Patch implemented and under review.

Improvements cont’d..

- Speedup snapshot mount
  - Parallelize snapshot mounts in ’lctl snapshot_mount’ (EX-4911)
    - Patch completed and merged.
    - Reduced mount time from 2760 sec to 137 sec (Total of 33 OSTs; number of OSTs matters less due to parallelization)
- Eliminating performance impacts at snapshot creation
  - Speed up snapshot creation on large targets (EX-5082/EX-5178)
    - Patch implemented and under review
    - Snapshot creation time down to 37sec from ~300 sec without patch
    - Under performance testing
    - Continue to look for further optimizations

Updates (May 6th)

- Update user interface for snapshot destroy
  - New sub command “lctl snapshot_stat” introduced

```
# lctl snapshot_stat
 lustre-OST0000:
   pending_delete_kb: 88
   delete_paused: 1
   delete_delay: 10
   ms used_kb: 132
```

  - Added new options to ”lctl snapshot_destroy”
    - "-D | --delay": add delay in ms between the truncation in each loop to control the destroy rate
    - “-P | --pause": set --pause on or off to pause or resume destroy operation

Updates (May 6th)

- EX-5082/EX-5178: Eliminating performance impacts at snapshot creation
  - Patch that significantly reduces the performance impact of snapshot creation
- Identified a Regression in Testing
  - Unable to mount snapshot due to incorrect GDT (Group Descriptor Table) of the snapshot file (EX-5194)
  - EX-5082/EX-5178 patches change the way how the GDT is handled
  - Investigation: Created a debug patch to collect more information
- Performance impact of large OSTs regardless of whether snapshot are enabled (EX-5183)
  - We have started to use very large OSTs a few years ago
  - Typically, large OST consists of massive amounts of blocks in ldiskfs.
  - Block allocators in the linux kernel finds suitable free blocks and allocates data to them.
  - DDN has fixed and optimized the block allocation for large OST, but optimizations might not be sufficient
  - Various other improvements exist in upstream ext4 in the Linux kernel
    - Ported these patches to Lustre/EXAScaler and started performance tests and analysis
    - Comparing performance with/without patch

Updates (May 13th)

- New system is running up and accelerates tests
  - Setup a system for development
  - Setup a large system (same size of current test system)
- Performance investigations(EX-5183)
  - Might or might not be related to co-located MDS and OSS configuration
  - Isolating the problem in conditions (if the same problem exists in dedicate MDS and OSS configuration)
- An issue was identified in async delete operation
  - Delete thread wasn’t triggered if backend device has some errors conditions
- Fix exclude bitmap errors
  - Snapshot blocks are marked as exclude blocks to avoid COW in snapshot shrink/merge/removal
  - ldiskfs and fsck couldn’t handle them in some cases
  - Patch for ldiskfs and e2fsprogs (fsck) are under review

Updates (May 13th)

- Continue to work on user interface improvements
  - Added printing used capacity per snapshot
    - lctl snapshot_list -a
  - Support table format support for visibility
- (Minor) Fix loading loop backup device module automatically when it's not laoded

Updates (May 20th)

- Code Freeze
  - All features except GDT patch were merged into snapshot branch and built a stable release
    - Optimizations in snapshot delete. Implemented asynchronous snapshot delete and readahead for Indirect blocks.
    - Allows parallel snapshot target mount to speedup snapshot_mount
    - Various improvements of user interface
      - "lctl snapshot_stat" support for background jobs
      - Added new table format to print snapshot summary
  - GDT patch is still under investigation and re-work required
  - Continue to work on the performance improvements for large OST
- Start QA processes on two large systems
  - One system starts tests on empty filesystem and another system starts tests at 50% full
    - Make sure all snapshot operations works properly
    - Observe performance impacts on clients
    - Fault injections trigger occasionally