Data Management on UL HPC

Sébastien Varrette, PhD

UL HPC Management Team,

Parallel Computing and Optimization Group (PCOG),

University of Luxembourg (UL), Luxembourg

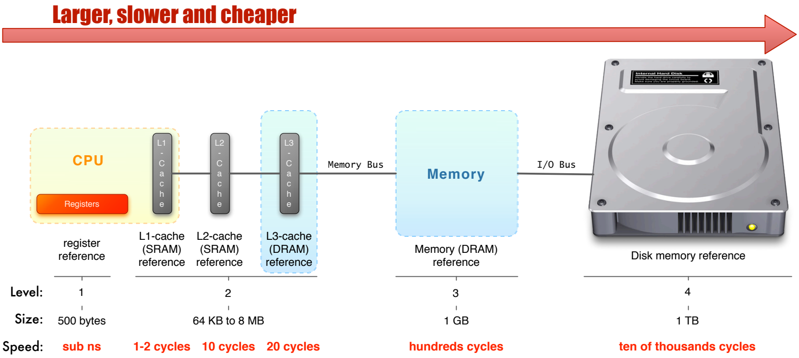
Summary

1 **Data Storage on UL HPC**

2 Storage Policy

3 Last Challenges
Effective Storage and Memory Management
Fault Tolerance

Local Data Storage



- SSD R/W: 560 MB/s; 85000 IOps **1000 €/TB**
- HDD (SATA @ 7,2 krpm) R/W: 100 MB/s; 190 IOps **100 €/TB**

Available File Systems

File Systems

Logical manner to store, organize, manipulate and access data.

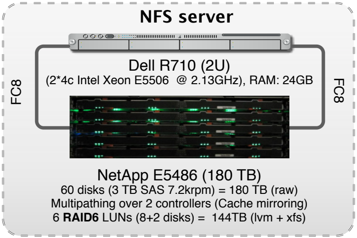
- **Disk file systems**: `ext4` (nodes), `xfs` (storage servers)
- **Network file systems**: NFS, SMB/CIFS
- **Distributed parallel file systems**: Lustre, GPFS, GlusterFS
  - ↪ data are stripped over multiple servers for high performance.
  - ↪ generally add robust failover and recovery mechanisms

Shared storage on UL HPC

- All based on disk enclosure (Nexsan or NetApp)

NFS-based Storage on UL HPC

- Enclosures configured with `xfs` over LVM
- An attached server exports the volume over `NFS`



NFS server

Dell R710 (2U)
(2*4c Intel Xeon E5506 @ 2.13GHz), RAM: 24GB

NetApp E5486 (180 TB)
60 disks (3 TB SAS 7.2krpm) = 180 TB (raw)
Multipathing over 2 controllers (Cache mirroring)
6 **RAID6** LUNs (8+2 disks) = 144TB (lvm + xfs)

Effective capacity: **109 TB**

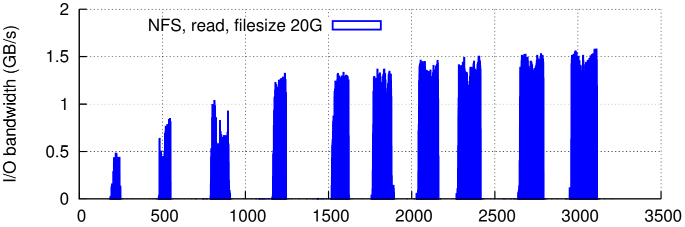
- Only available on **Chaos**:
  - ↪ 1 Netapp Enclosure (109 TB):
    - ✓ `$HOME`
    - ✓ `$WORK`

- **Note**: all NFS shared storage of **Gaia** was replaced on March 2015 in favor of GPFS

NFS Performances

- **Remember that NFS-based storage DOES NOT scale**

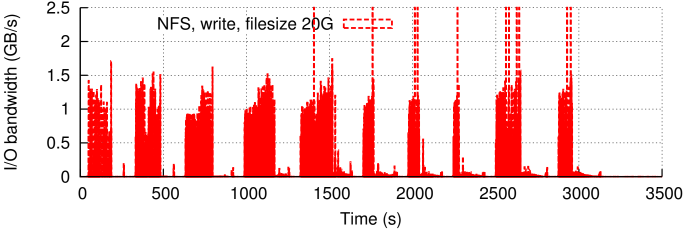
- In particular, adding a new enclosure:
  ↪ adding a new enclosure: **does not** improve the general performance
  ✓ un-like Lustre and GPFS



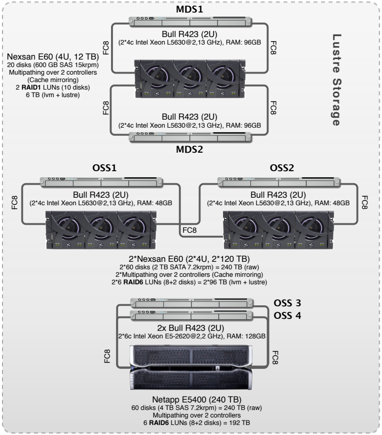
NFS Performances

- **Remember that NFS-based storage DOES NOT scale**

- In particular, adding a new enclosure:
  - ↪ adding a new enclosure: **does not** improve the general performance
    - ✓ un-like Lustre and GPFS



Lustre Storage (Gaia)



Bull R423 (2U)
(2*4c Intel Xeon L5630@2,13 GHz), RAM: 96GB

Nexsan E60 (4U, 12 TB)
20 disks (600 GB SAS 15krpm)
Multipathing over 2 controllers
(Cache mirroring)
2 **RAID1** LUNs (10 disks)
6 TB (lvm + lustre)

Bull R423 (2U)
(2*4c Intel Xeon L5630@2,13 GHz), RAM: 96GB

Bull R423 (2U)
(2*4c Intel Xeon L5630@2,13 GHz), RAM: 48GB

Bull R423 (2U)
(2*4c Intel Xeon L5630@2,13 GHz), RAM: 48GB

2*Nexsan E60 (2*4U, 2*120 TB)
2*60 disks (2 TB SATA 7.2krpm) = 240 TB (raw)
2*Multipathing over 2 controllers (Cache mirroring)
2*6 **RAID6** LUNs (8+2 disks) = 2*96 TB (lvm + lustre)

2x Bull R423 (2U)
2*6c Intel Xeon E5-2620@2,2 GHz), RAM: 128GB

Netapp E5400 (240 TB)
60 disks (4 TB SAS 7.2krpm) = 240 TB (raw)
Multipathing over 2 controllers
6 **RAID6** LUNs (8+2 disks) = 192 TB

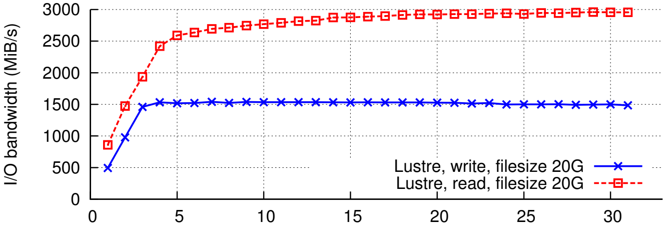
Effective capacity: **347 TB**

- Scalable Parallel FS
  - ↪ `$SCRATCH`
- Only available on **gaia**
- Current Layout:
  - ↪ 2 MDS servers,
  - ↪ 4 OSS servers,
  - ↪ 3 Nexsan E60 encl.
  - ↪ 1 Netapp E5400 encl.

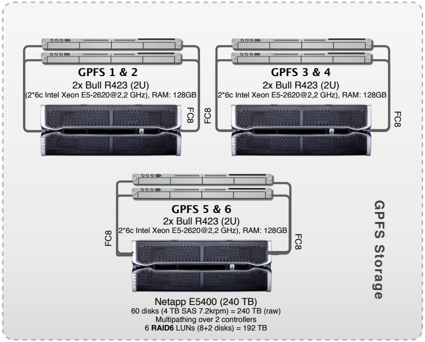
Lustre Performances

- **Remember that Lustre-based storage DOES scale**

- In particular, adding a new enclosure:
  ↪ increase the global capacity accordindly
  ↪ **adds** the performance to the global perf. of the system
  ↪ Note: below measures were done **before** the recent extension



GPFS Storage (Gaia)


GPFS 1 & 2
2x Bull R423 (2U)
(2*6c Intel Xeon E5-2620@2,2 GHz), RAM: 128GB
FC8
FC8
GPFS 3 & 4
2x Bull R423 (2U)
2*6c Intel Xeon E5-2620@2,2 GHz), RAM: 128GB
FC8
GPFS 5 & 6
2x Bull R423 (2U)
2*6c Intel Xeon E5-2620@2,2 GHz), RAM: 128GB
FC8
FC8
GPFS Storage
Netapp E5400 (240 TB)
60 disks (4 TB SAS 7.2krpm) = 240 TB (raw)
Multipathing over 2 controllers
6 **RAID6** LUNs (8+2 disks) = 192 TB

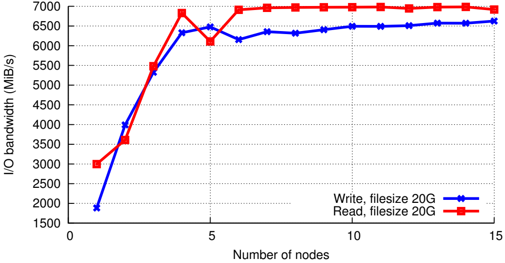
Effective capacity: **524 TB**

- Scalable Parallel FS
  - ↪ `$HOME`
  - ↪ `$WORK`
- Only available on **gaia**
- Current Layout:
  - ↪ 6 servers,
  - ↪ 3 Netapp E5400 encl.

GPFS Performances

- **Remember that GPFS-based storage DOES scale**

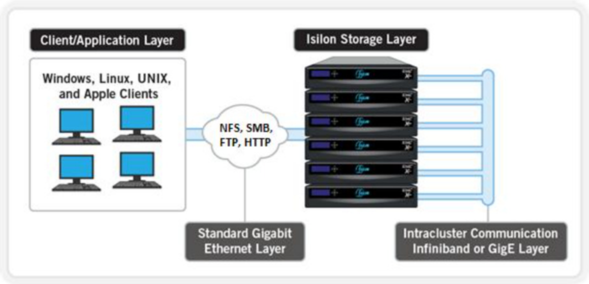
- In particular, adding a new enclosure:
  - ↪ increase the global capacity accordindly
  - ↪ **adds** the performance to the global perf. of the system



Isilon

Effective capacity: **1460 TB**

- Mounting point on the **gaia** cluster: `/mnt/isilon`
  - ↪ Performance evaluation in progress
  - ↪ **Obj**: projects data go on it


Client/Application Layer
Windows, Linux, UNIX, and Apple Clients
NFS, SMB, FTP, HTTP
Isilon Storage Layer
Standard Gigabit Ethernet Layer
Intracluster Communication Infiniband or GigE Layer

UL HPC Backups

Total **Effective** (split) capacity: **1365 TB**

- Based on bontmia and backupninja
  - ↪ Backup Over Network To Multiple Incremental Archives
  - ↪ `ULHPC/puppet-bontmia` puppet module
- **NFS-based** targets:
  - ↪ **Chaos**: 1 Netapp Enclosure (130 TB) `cartman`
  - ↪ **Gaia**:
    - ✓ 1 Netapp Enclosure (130 TB) `stan`
    - ✓ 1 Nexsan Enclosure (189 TB): former `nfs.gaia`
- **GlusterFS-based** targets (**Gaia** only) (916 TB)
  - ↪ `highlander` server exports the volumes
    - ✓ `bertha` and the others Certon are storage enclosures

Summary

1 Data Storage on UL HPC

2 Storage Policy

3 Last Challenges
Effective Storage and Memory Management
Fault Tolerance

Multi-Tier Environment

Tier-structure of storage space

1. **Tier-1** (GPFS): high performance, high reliability
   ↪ put there frequently processed data only
2. **Tier-2** (Certons): low performance
   ↪ storage and backup disks (≃ archiving)
3. **Tier-0** (LUSTRE): Scratch
   ↪ ultra high performance, (potentially) low reliability

Storage Policy

- $HOME (**NFS** or **GPFS**) **is** under a regular backup policy.
- $WORK (**NFS** or **GPFS**) **is not** backed up
  - ↪ Avoid massive parallel writes under NFS
  - ↪ Use cdw to quickly change your current directory to $WORK
- $SCRATCH (**Lustre** *) **is not** backed up
  - ↪ designed for **temporarily need**, with fast I/O
  - ↪ Use cds to quickly change your current directory to $SCRATCH
  - ↪ **On Chaos, $SCRATCH is /tmp thus NOT Shared**

Directory	Max size	Max #files	Backup
$HOME	50 GB	500.000	YES
$WORK	3 TB		NO
$SCRATCH	10 TB		NO

Project Management

- In case the regular storage limits **do not** match your expectations
  - ↪ quotas extension for project folders can be granted
  - ↪ this comes with additional fees

Project Storage Request Form

Contact: `joanna.smula@uni.lu`

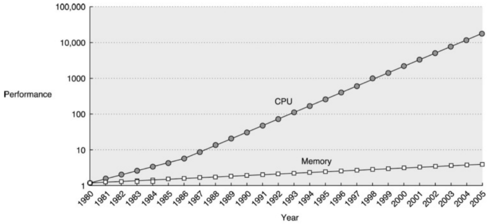
Summary

1 Data Storage on UL HPC

2 Storage Policy

3 **Last Challenges**
Effective Storage and Memory Management
Fault Tolerance

Memory bottleneck



- A regular computing node have at least 2GB/core RAM
  - ↪ Do 12-24 runs fit in the memory?
  - ↪ If your job runs out of memory, it simply crashes
- Use fewer simultaneous runs if **really** needed!
  - ↪ **OR** request a big memory machine (1TB RAM)

```
$> oarsub -t bigmem ...
```

  - ↪ **OR (better)** explore parallization (MPI, OpenMP, pthreads)

Understanding Your Storage Options

Where can I store and manipulate my data?

- **Shared** storage

  ↪ `NFS` – **not scalable** ≃ 1.5 GB/s (R) $\mathcal{O}$(100 TB)

  ↪ `GPFS` – **scalable** ≃ 6 GB/s (R) $\mathcal{O}$(500 TB)

  ↪ `Lustre` – **scalable** ≃ 5 GB/s (R) $\mathcal{O}$(400 TB)

- **Local** storage

  ↪ local file system (`/tmp`) $\mathcal{O}$(200 GB)

  ✓ over HDD ≃ 100 MB/s

  ✓ over SDD ≃ 400 MB/s

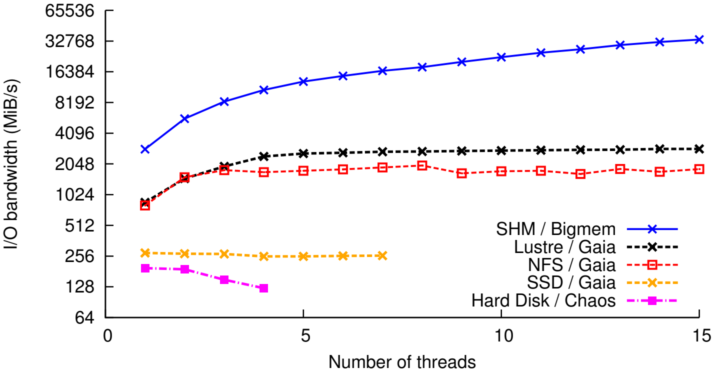
  ↪ RAM (`/dev/shm`) ≃ 30 GB/s (R) $\mathcal{O}$(20 GB)

⇒ **In all cases:** small I/Os really **kill** storage performances

Storage performances

- Based on IOR or IOZone, reference I/O benchmarks

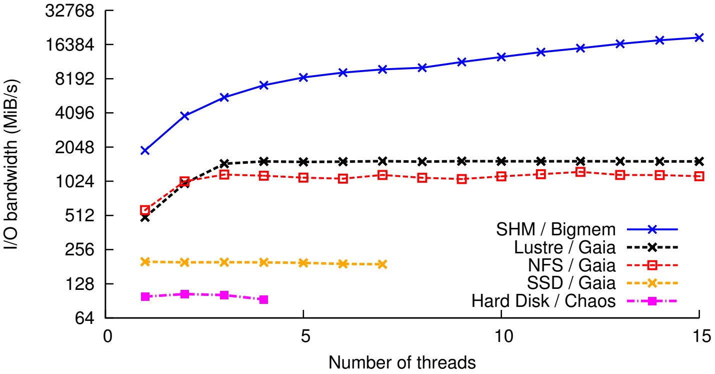
Read



Storage performances

- Based on IOR or IOZone, reference I/O benchmarks

Write



Speed Expectation on Data Transfer

http://fasterdata.es.net/

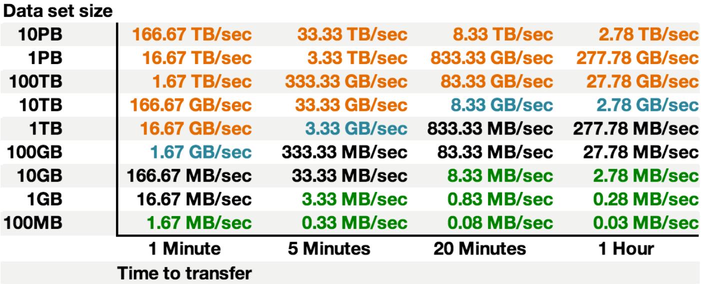
- How long to transfer **1 TB** of data across various speed networks?

Network	Time
10 Mbps	300 hrs (12.5 days)
100 Mbps	30 hrs
1 Gbps	3 hrs
10 Gbps	20 minutes

- **(Again)** small I/Os really **kill** performances
  - ↪ **Ex**: transferring 80 TB for the backup of `ecosystem_biology`
  - ↪ same rack, 10Gb/s. 4 weeks ⟶ 63TB transfer...

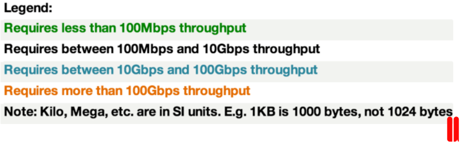
Speed Expectation on Data Transfer

http://fasterdata.es.net/



Data set size	1 Minute	5 Minutes	20 Minutes	1 Hour
10PB	166.67 TB/sec	33.33 TB/sec	8.33 TB/sec	2.78 TB/sec
1PB	16.67 TB/sec	3.33 TB/sec	833.33 GB/sec	277.78 GB/sec
100TB	1.67 TB/sec	333.33 GB/sec	83.33 GB/sec	27.78 GB/sec
10TB	166.67 GB/sec	33.33 GB/sec	8.33 GB/sec	2.78 GB/sec
1TB	16.67 GB/sec	3.33 GB/sec	833.33 MB/sec	277.78 MB/sec
100GB	1.67 GB/sec	333.33 MB/sec	83.33 MB/sec	27.78 MB/sec
10GB	166.67 MB/sec	33.33 MB/sec	8.33 MB/sec	2.78 MB/sec
1GB	16.67 MB/sec	3.33 MB/sec	0.83 MB/sec	0.28 MB/sec
100MB	1.67 MB/sec	0.33 MB/sec	0.08 MB/sec	0.03 MB/sec

Time to transfer



Legend:

Requires less than 100Mbps throughput

Requires between 100Mbps and 10Gbps throughput

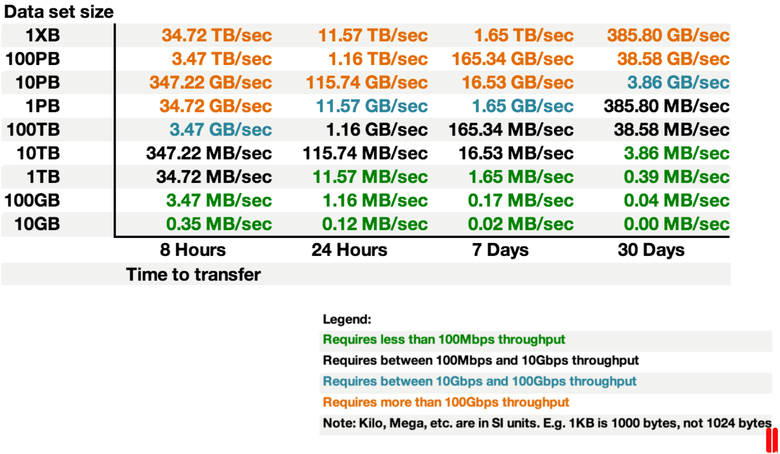
Requires between 10Gbps and 100Gbps throughput

Requires more than 100Gbps throughput

Note: Kilo, Mega, etc. are in SI units. E.g. 1KB is 1000 bytes, not 1024 bytes

Speed Expectation on Data Transfer

http://fasterdata.es.net/



Data set size	8 Hours	24 Hours	7 Days	30 Days
1XB	34.72 TB/sec	11.57 TB/sec	1.65 TB/sec	385.80 GB/sec
100PB	3.47 TB/sec	1.16 TB/sec	165.34 GB/sec	38.58 GB/sec
10PB	347.22 GB/sec	115.74 GB/sec	16.53 GB/sec	3.86 GB/sec
1PB	34.72 GB/sec	11.57 GB/sec	1.65 GB/sec	385.80 MB/sec
100TB	3.47 GB/sec	1.16 GB/sec	165.34 MB/sec	38.58 MB/sec
10TB	347.22 MB/sec	115.74 MB/sec	16.53 MB/sec	3.86 MB/sec
1TB	34.72 MB/sec	11.57 MB/sec	1.65 MB/sec	0.39 MB/sec
100GB	3.47 MB/sec	1.16 MB/sec	0.17 MB/sec	0.04 MB/sec
10GB	0.35 MB/sec	0.12 MB/sec	0.02 MB/sec	0.00 MB/sec

Time to transfer

Legend:

Requires less than 100Mbps throughput

Requires between 100Mbps and 10Gbps throughput

Requires between 10Gbps and 100Gbps throughput

Requires more than 100Gbps throughput

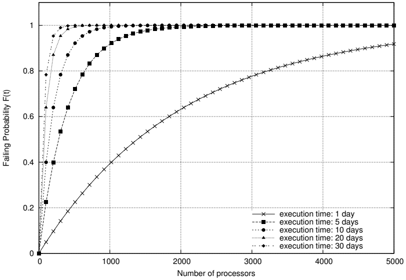
Note: Kilo, Mega, etc. are in SI units. E.g. 1KB is 1000 bytes, not 1024 bytes

Fault Tolerance

- Cluster maintenance from time to time

Fault Tolerance

- Cluster maintenance from time to time
- Reliability vs. Crash Faults in Distributed systems



Fault Tolerance

- Cluster maintenance from time to time
- Reliability vs. Crash Faults in Distributed systems
- Fault Tolerance general strategy: checkpoint/rollback
  - ↪ assumes a way to save the state of your program
  - ↪ hints: OAR `-signal -checkpoint -idempotent...`, BLCR
  - ↪ combine best-effort jobs with checkpointing (`http://git.io/c-dn1A`)

Thank you for your attention...

Questions?

Sébastien Varrette, PhD
mail: sebastien.varrette@uni.lu
Office E-007
Campus Kirchberg
6, rue Coudenhove-Kalergi
L-1359 Luxembourg

UL HPC Management Team
mail: hpc-sysadmins@uni.lu

1. **Data Storage on UL HPC**
2. **Storage Policy**
3. **Last Challenges**
   - Effective Storage and Memory Management
   - Fault Tolerance