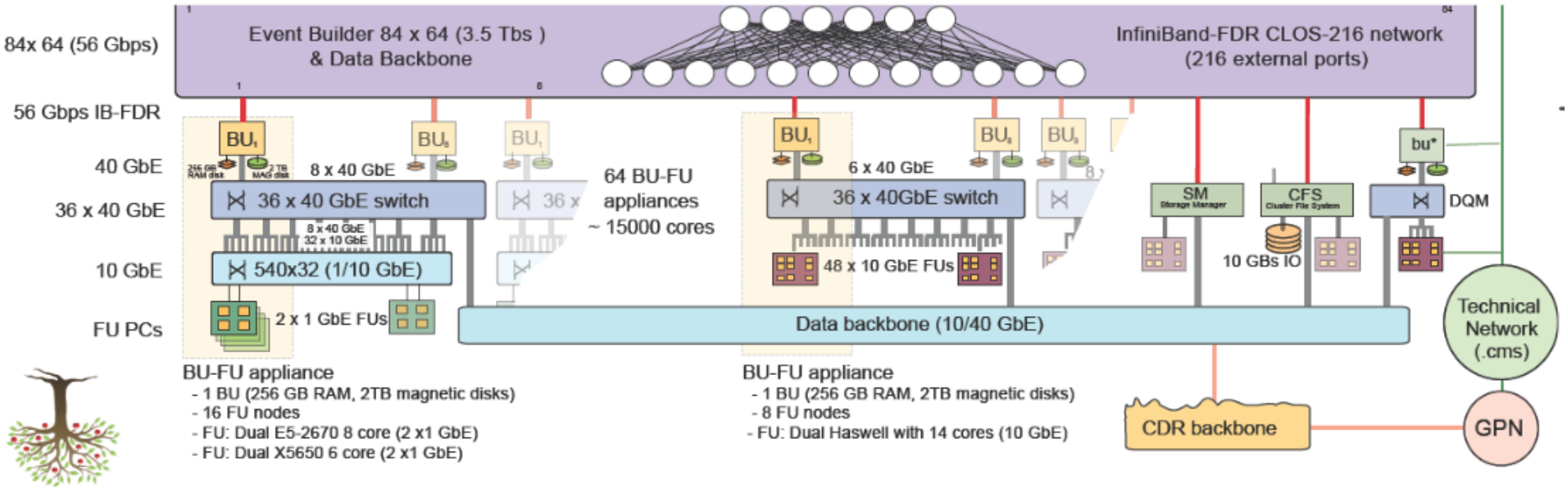
CMS experience with the deployment of Lustre

Lavinia Darlea, on behalf of CMS DAQ Group

MIT/DAQ CMS

April 12, 2016

CMS DAQ2 System

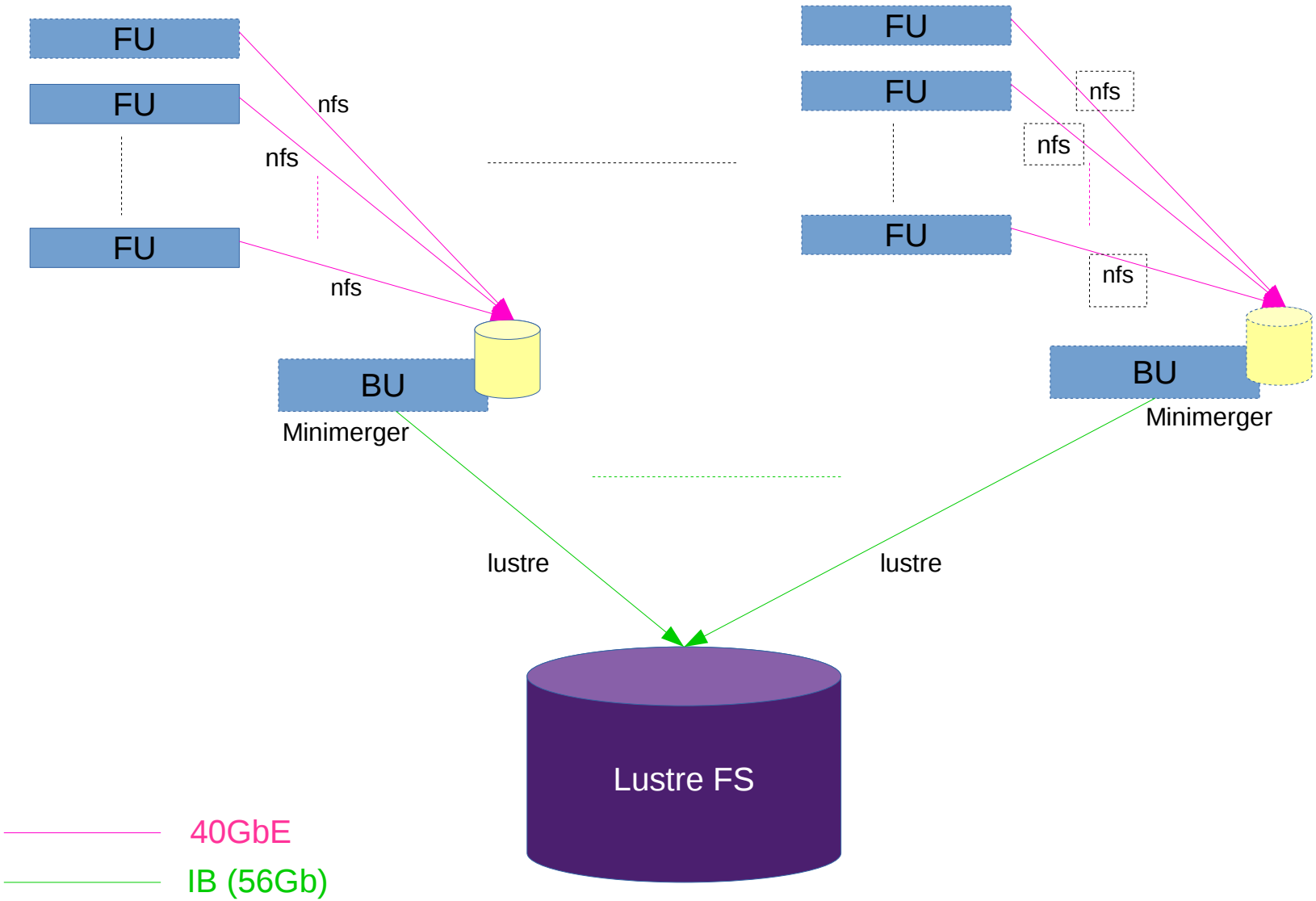


Storage Manager and Transfer System (SMTS) in the DAQ chain

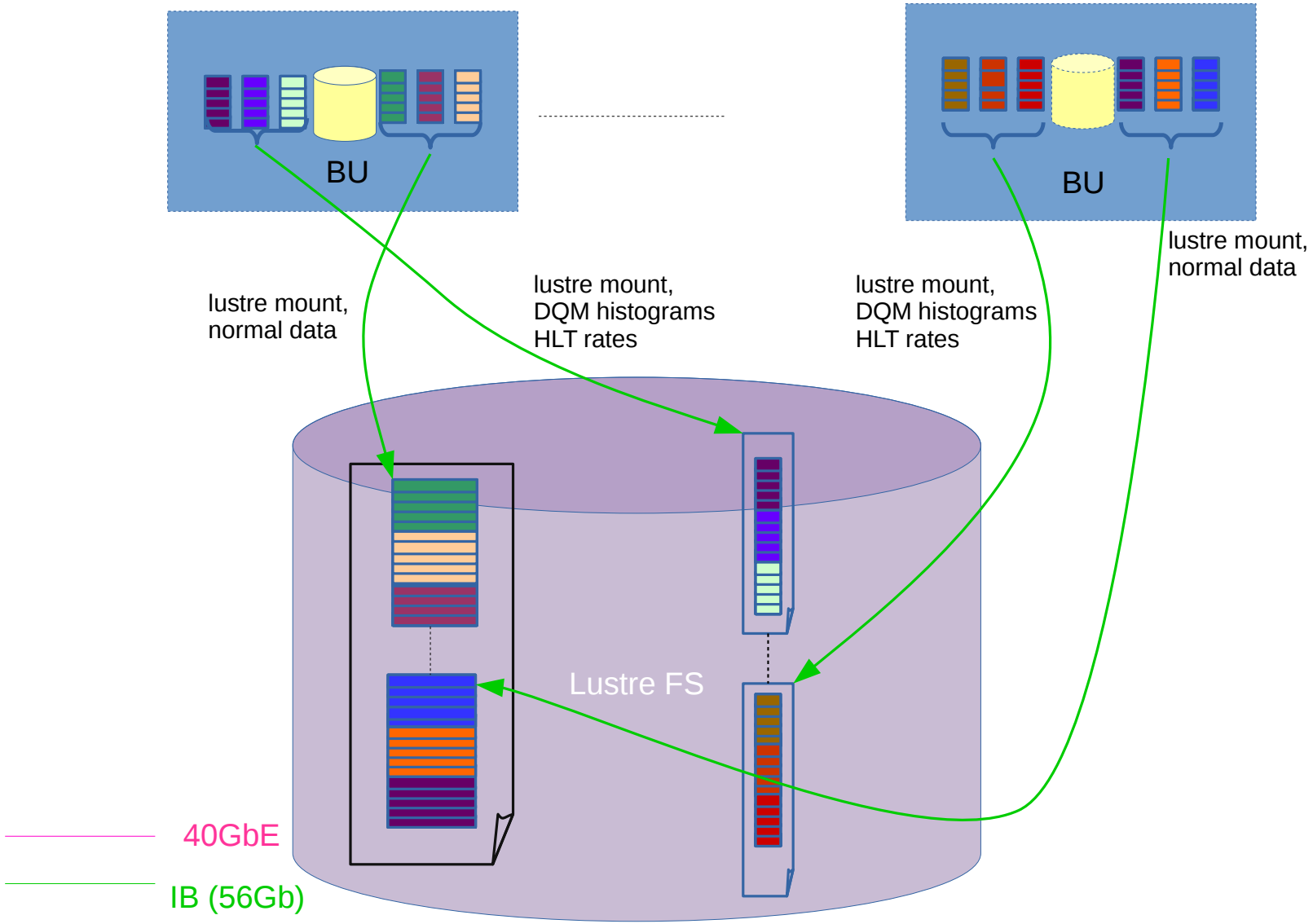
SMTS and DAQ

- input: output of the Data AcQuisition chain
- Lustre FS: ensure safe temporary storage
- output: transfer to Tier0

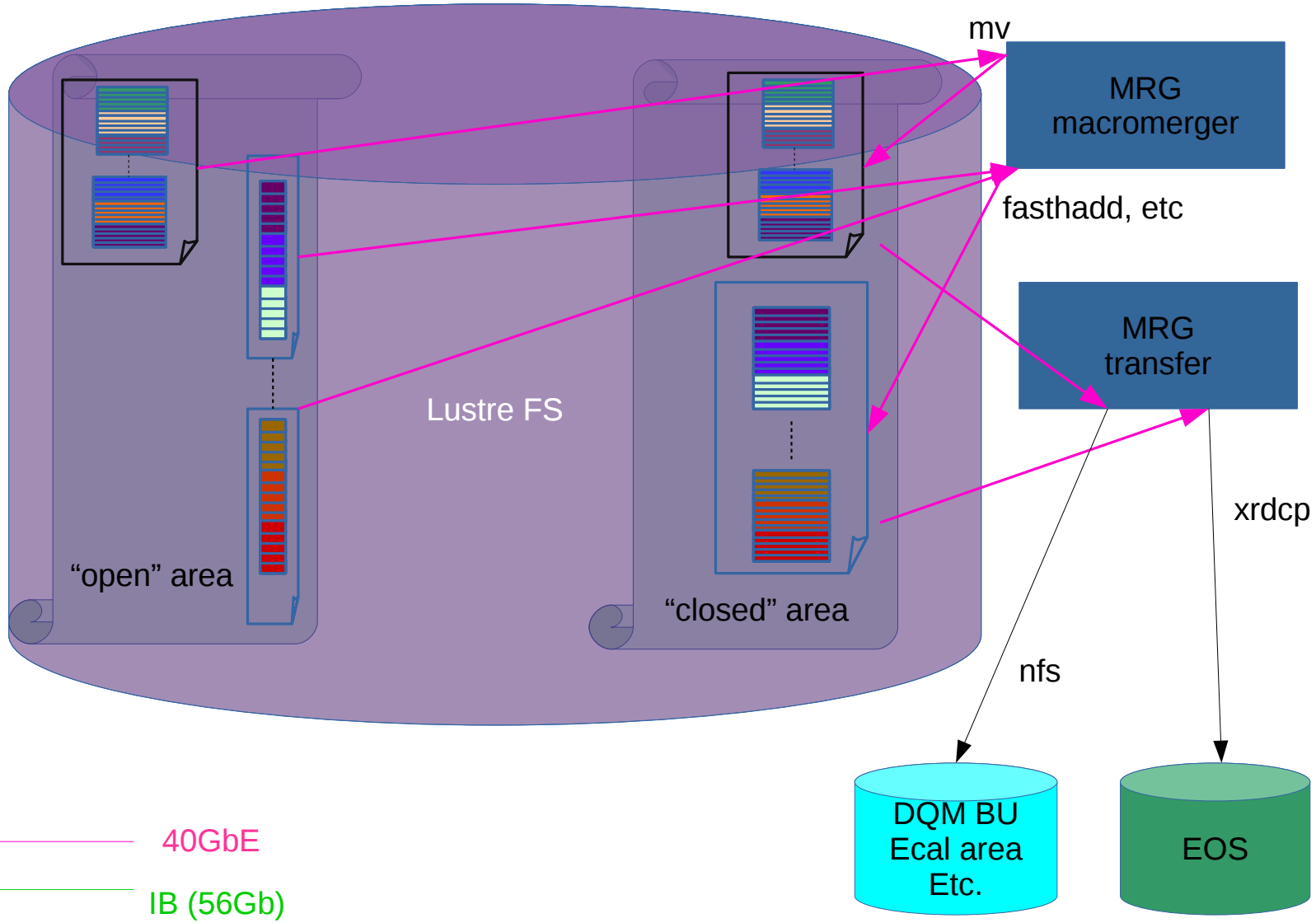
Data Flow into Lustre: overview



Data Flow into Lustre: minimergers



Data Flow into and out of Lustre: macromergers and transfer



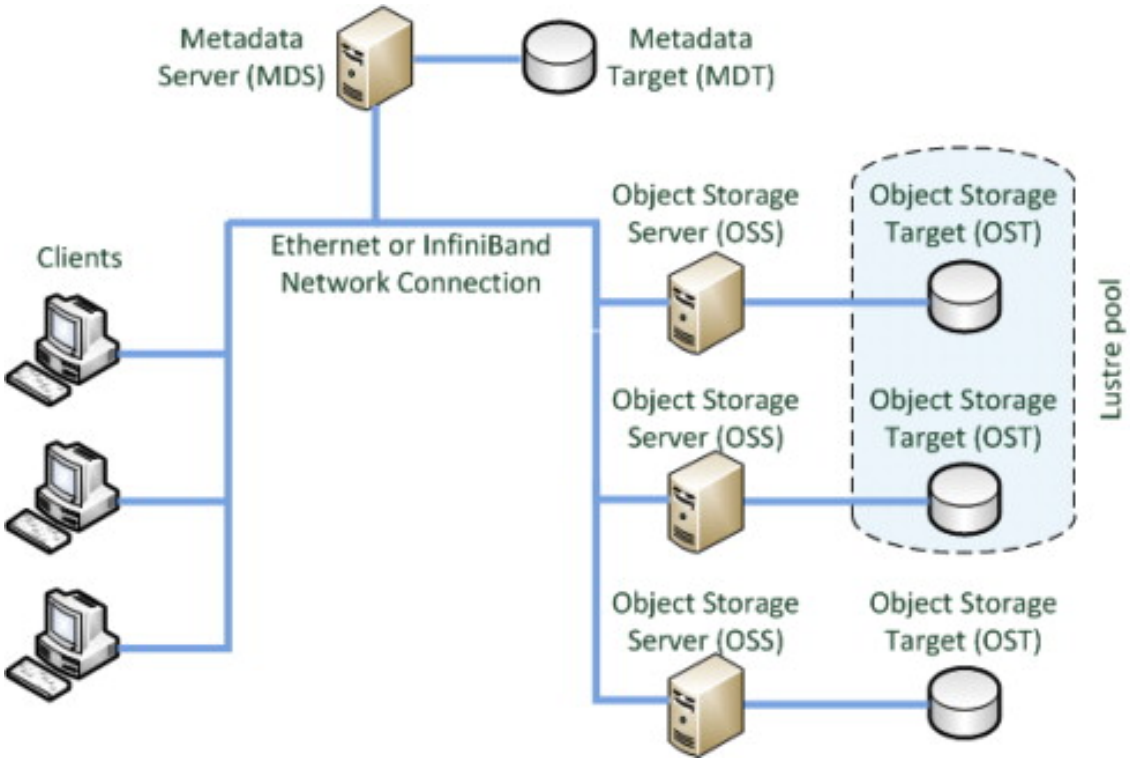
Storage and Transfer System Requirements

Storage and Transfer

Requirement	In	Out	Total
Space			250TB
Mergers Bandwidth	3GB/s	~0.3GB/s	3.3GB/s
Transfers Bandwidth	-	3GB/s	3GB/s
Total Bandwidth	3GB/s	3.3GB/s	6.3GB/s
Nb of files*	~2840 files/LS	~2780 files/min	~2840 files/min

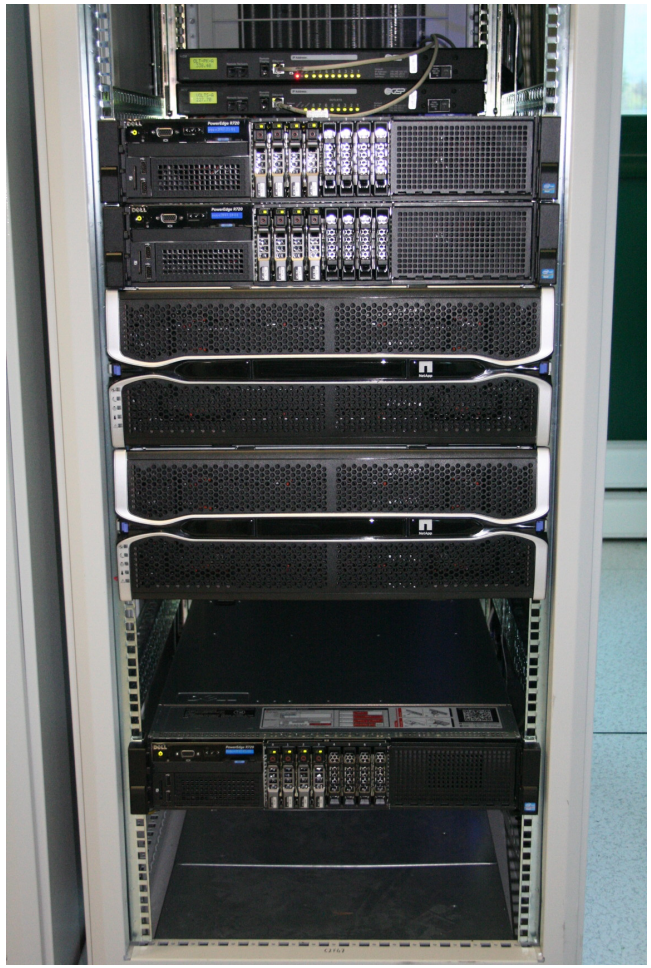
*In: create; Out: destroy
computation: (20 streams x 1 data file)/LS, (20 streams x 2 jsns x 70 BU)/LS, (1 lock file x 20 streams)/LS

Lustre File System – Implementation



Lustre FS architecture

- current Intel Enterprise Edition for Lustre version: 2.2.0.2
- servers: 6 DELL R720
  - 2 MDS nodes in active/passive failover mode
  - 4 OSS nodes, each controls 6 OSTs in pairs of active/passive failover mode

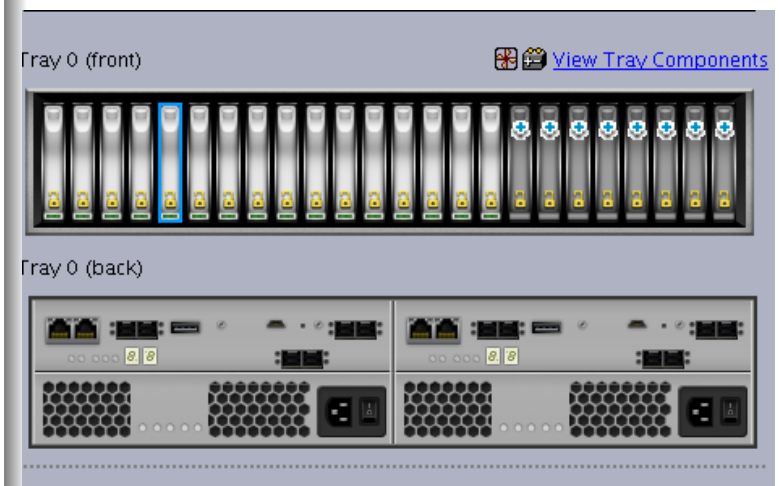


Rack view – MDT (low), 1 OST controller and 1 disk shelves expansion enclosure

Lustre File System – Implementation

Meta–Data Configuration

- 16 drives of 1TB in 1 volume group, 8 hot spares
- only 10% of the disks capacity is used in order to increase performance
- partitions: 10GB for MGT (special partition which serves as entry point for the clients connections), 1TB for MDT
- redundancy: RAID6



MDT: NetApp E2724 front and rear view

Lustre File System – Implementation

Object Storage Configuration

- 2 OST controllers: NetApp E5560
- each controller manages one disk expansion enclosure DE6600
- each disk shelf enclosure contains 60 disks of 2TB each
- total raw disk space: 240 disks x 2TB = 480 TB
- physical installation: 2 racks, 1 controller and its expansion enclosure per rack

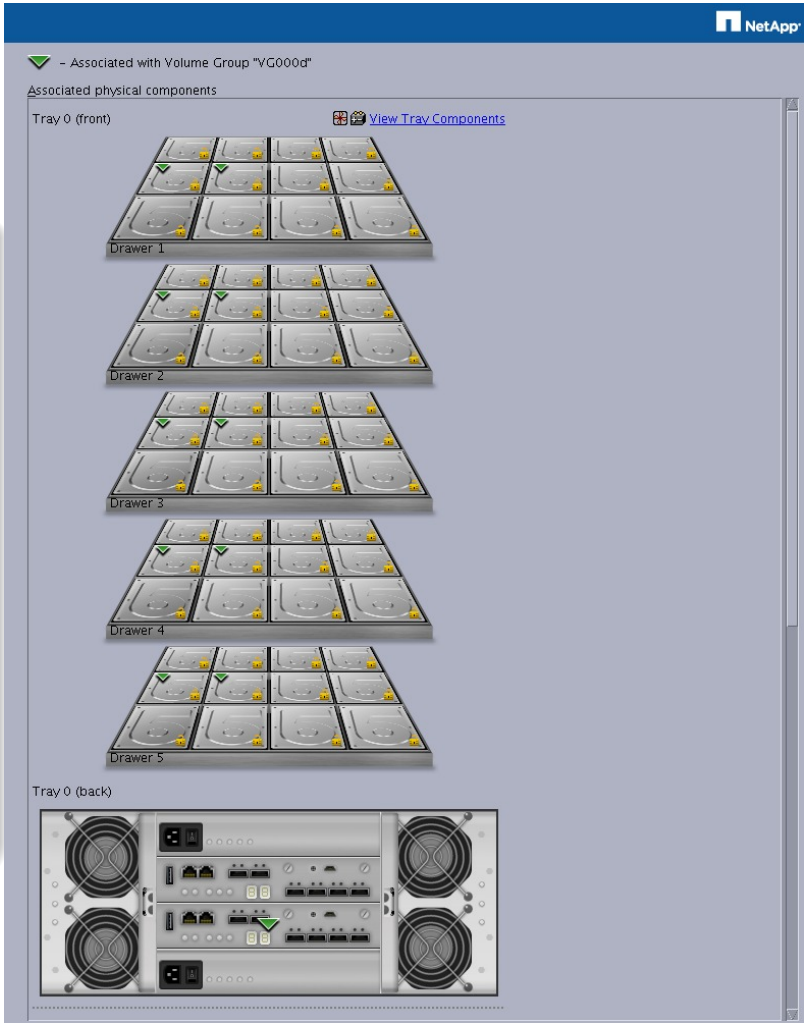
Front OST

Disk shelves

Lustre File System – Implementation

OST: Volume Configuration

- each controller/expansion shelf is organized in 6 RAID6 volume groups (8+2 disks)
- the volume groups are physically allocated vertically to ensure resilience to single shelf damage
- total usable space: 349TB



Volumes configuration

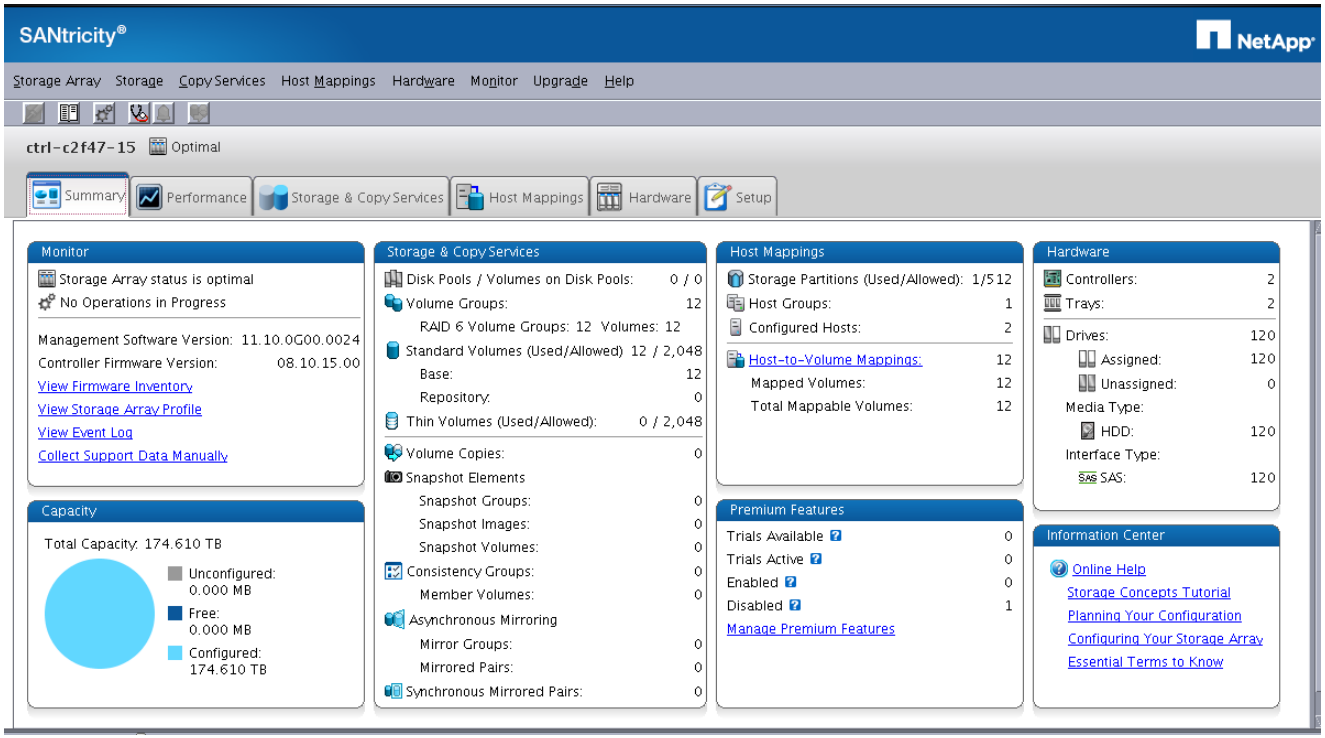
Lustre File System – Implementation

High Availability

- volumes distribution provides full shelf failure redundancy
- all volumes are RAID6
- all devices (controllers, shelves, servers) are dual powered (normal and UPS)
- all servers configured in active/passive failover mode via corosync/pacemaker: MDS in neighbouring racks, OSS within the same rack
- LFS nominal availability: 40GbE and InfiniBand (56Gb) data networks*

* not in failover mode

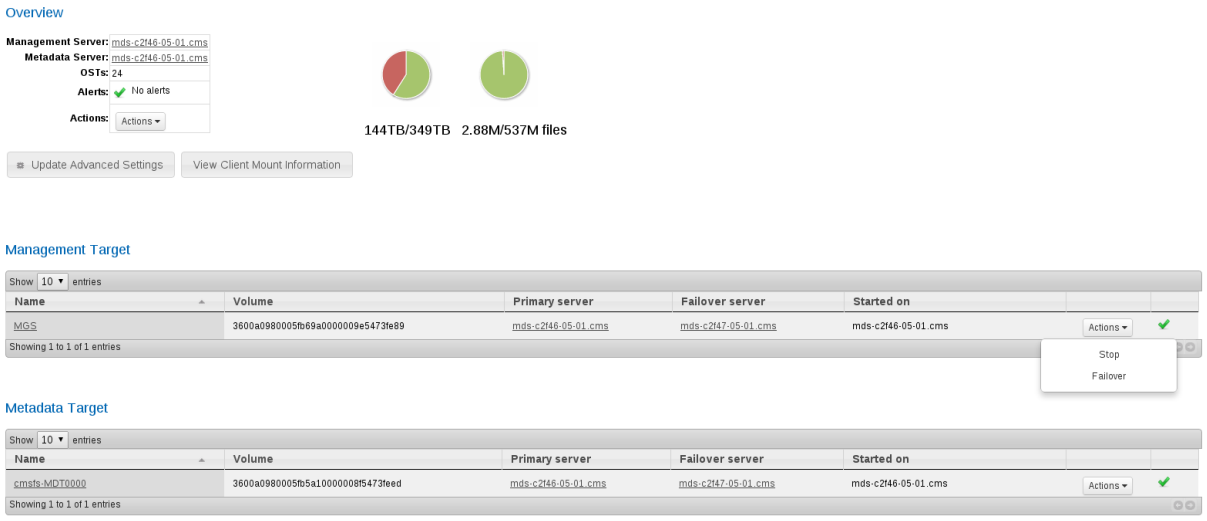
Lustre File System – Control and Monitoring



SANtricity

- GUI monitoring bandwidth usage per controller
- reports detailed text bandwidth usage per volume
- provides useful information and alerts on hardware status

Lustre File System – Control and Monitoring

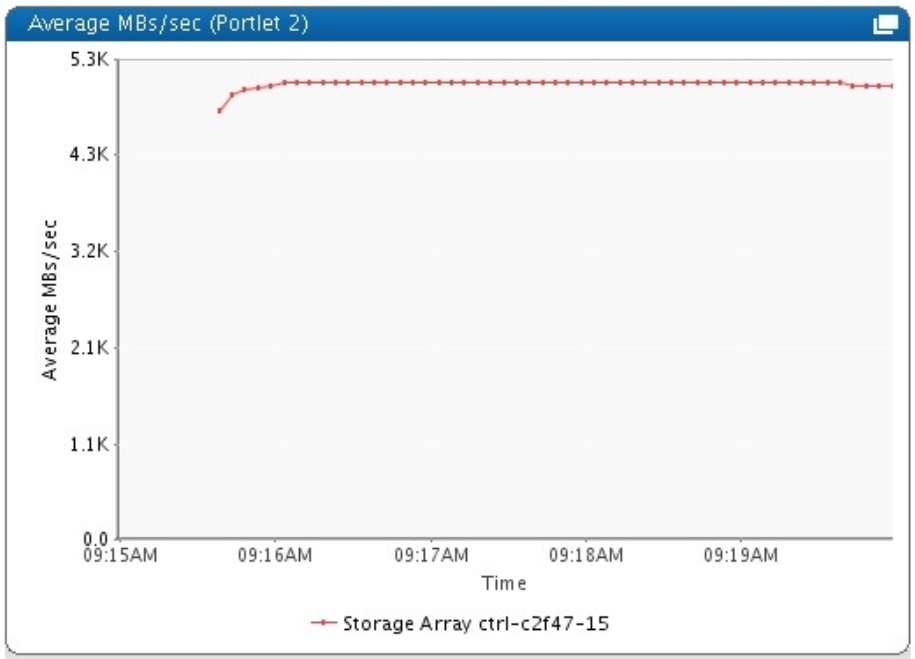


IML: Intel Management for Lustre

- (+) mostly used for control and base FS operations (failover, startup, shutdown)
- (+) the dashboard provides useful information for debugging an overloaded system
- (-) painful installation procedure
- (-) not fully reliable: fake BMC monitoring warnings, false status reports upon major FS failures

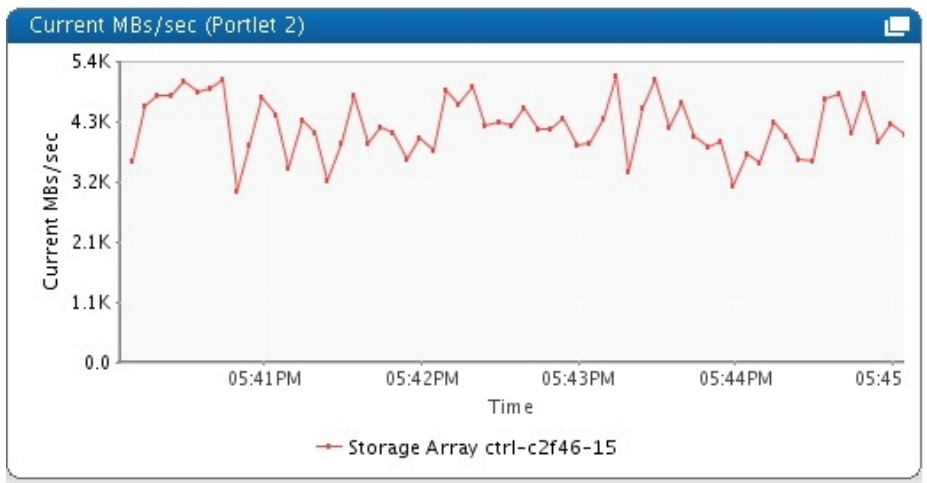
Bandwidth Validation

The plotted values are per controller. The 2 controllers were perfectly balanced.



Commissioning Acceptance

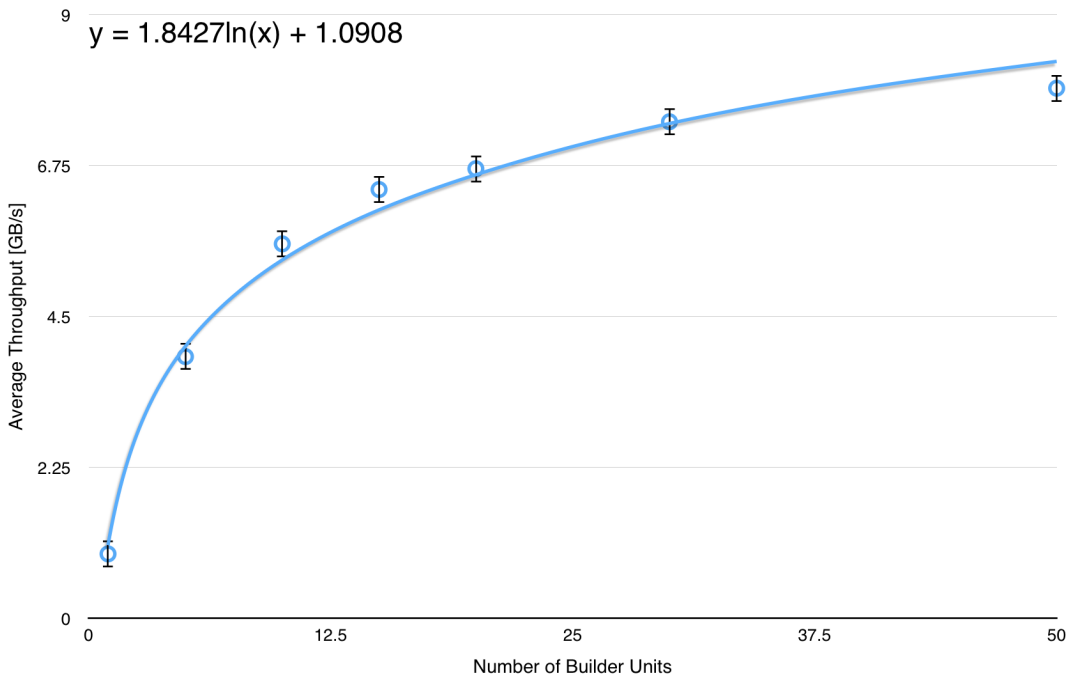
Proven steady 10GB/s rate in r/w mode



Merger emulation

Proven steady 7.5GB/s rate

Validation

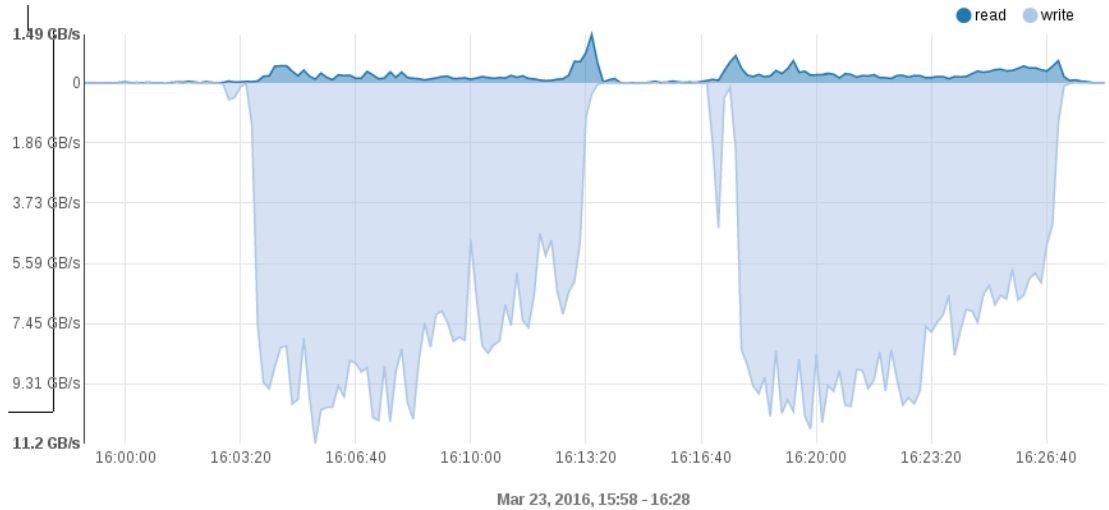


LFS bandwidth benchmarking

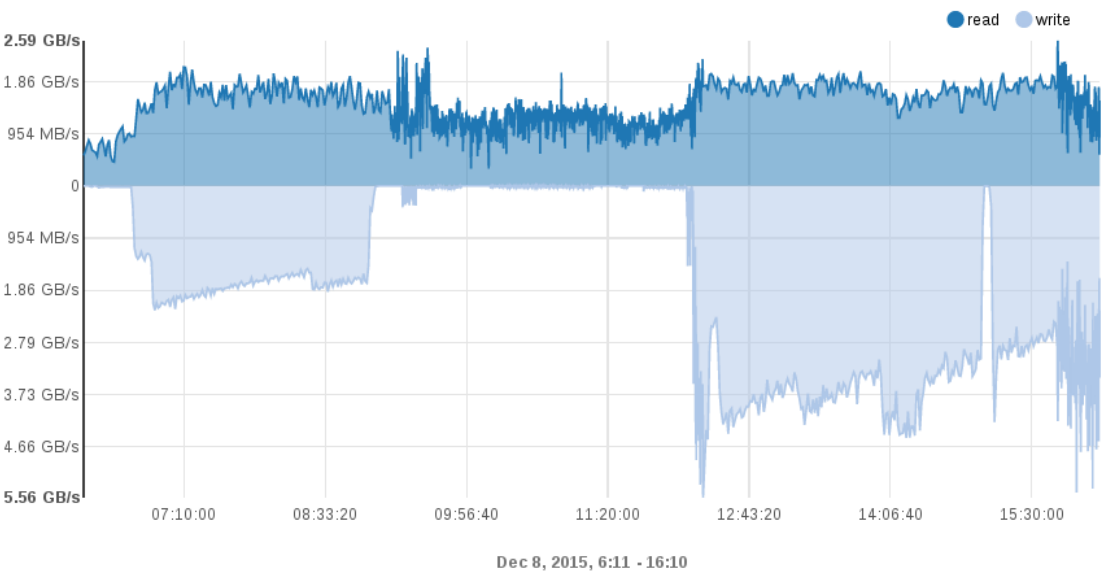
Emulation tests using the production computing cluster

- tests performed using different fractions of the available computing farm
- obvious non–linear behaviour with the number of BUs
- transfer system (read operations) were not considered during the tests
- saturation is expected around 8.5GB/s

Production Usage



DAQ tests,
March 2016



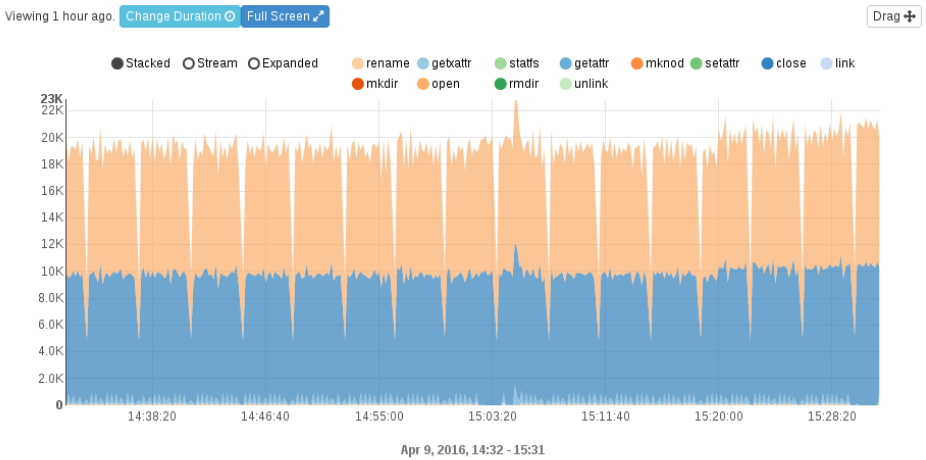
Heavy Ions runs,
December 2015

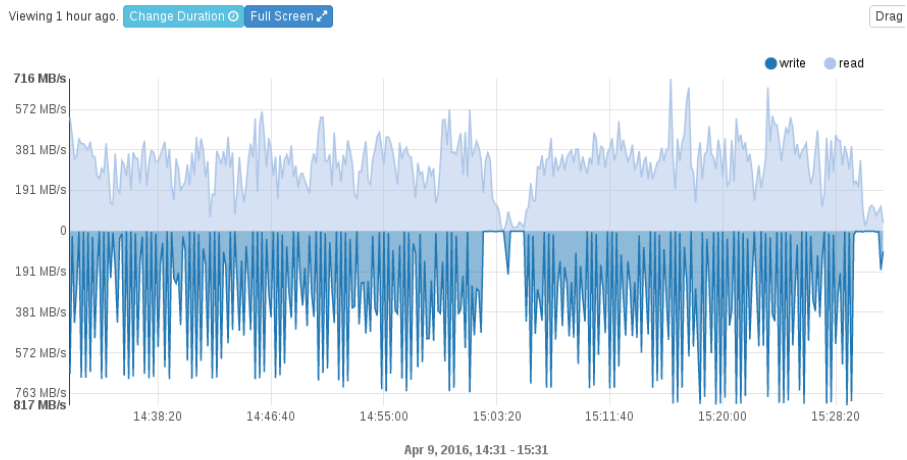
Important lessons

Sensitive points

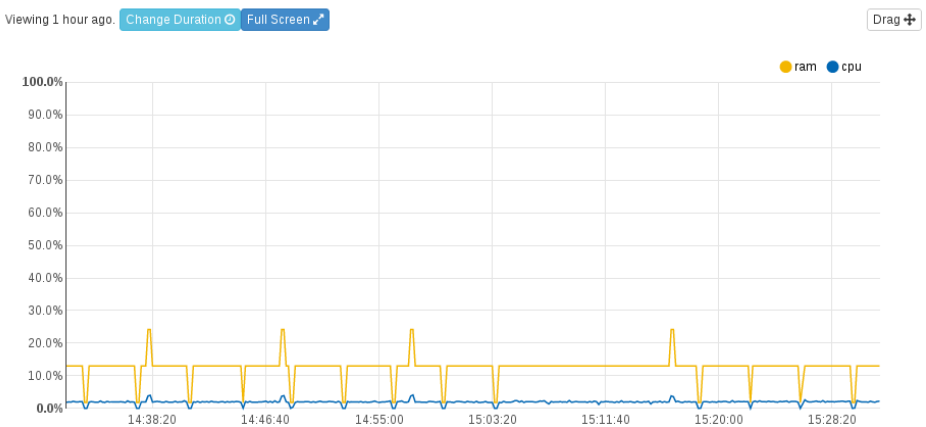
- Lustre is extremely sensitive to network glitches. It needs a very stable and reliable network. Adverse effects can go from individual clients being evicted to the entire FS shutting down
- Lustre is very greedy in terms of resources on the clients
  - unless nominally limited it will take up to 75% of the total RAM for its caching
  - unless nominally limited it will take a huge amount of slab memory to cache its objects (ldlm locks)
- Lustre is very greedy in terms of resources on the servers
  - MDS ideal setup: the MDT should fit entirely in the RAM
  - OSS: by default cache everything. They should be prevented from doing so

Weekend operation

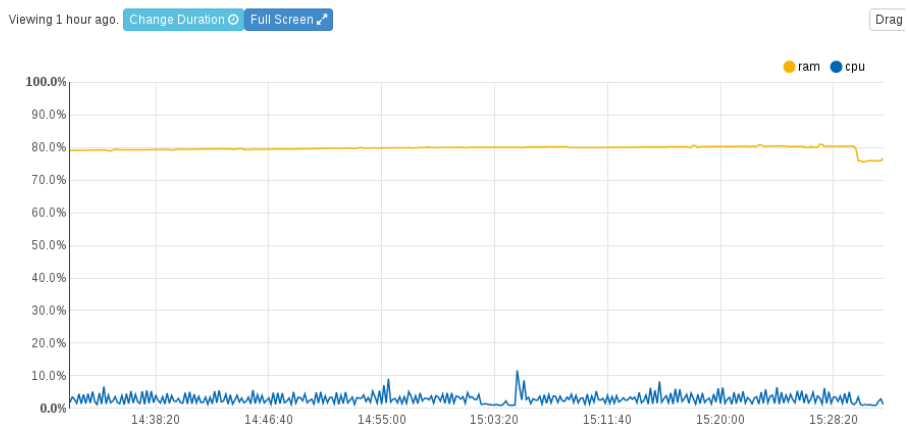




Metadata Servers



Object Storage Servers

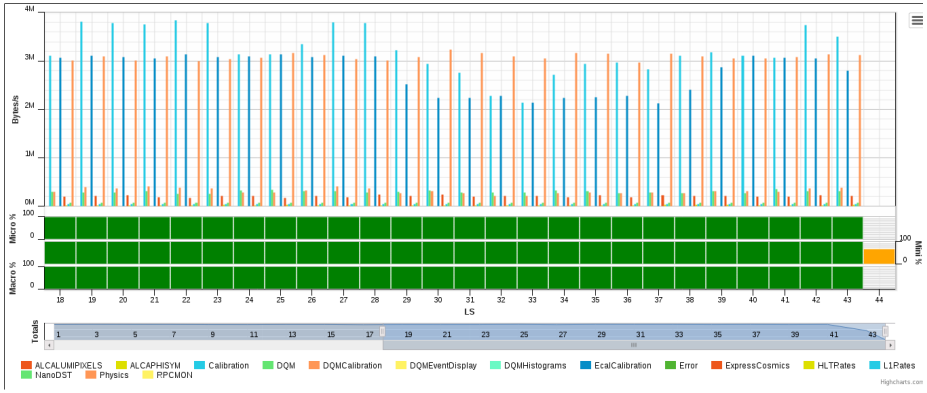


Conclusion

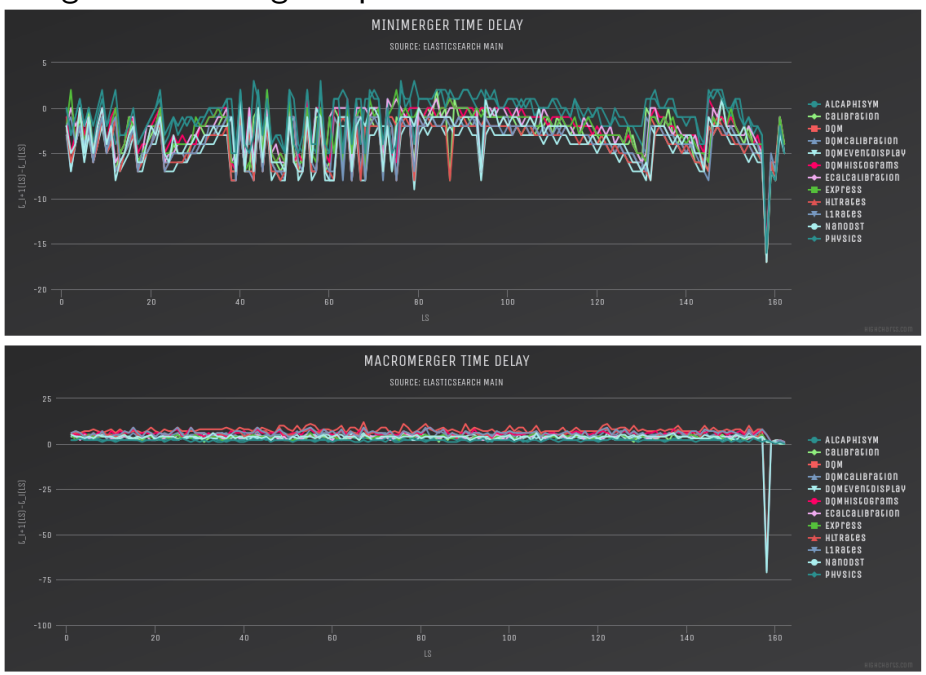
SMTS team interaction with Lustre

- IML can be misleading, but provides very intuitive ways of controlling the FS
- sub–optimal application architecture can artificially increase the load on the FS. Continuous tuning is being performed both at the application and FS level
- a few FS issues have been identified, but they have been mostly fixed
- clients recover pretty fast and painlessly after FS unavailability
- lustre and NetApp's E-Series seem to play nicely together and they deliver the required bandwidth performance
- Intel Lustre support team is reliable, knowledgeable and patient. But located mostly on a different continent

Conclusion



Mergers monitoring sample



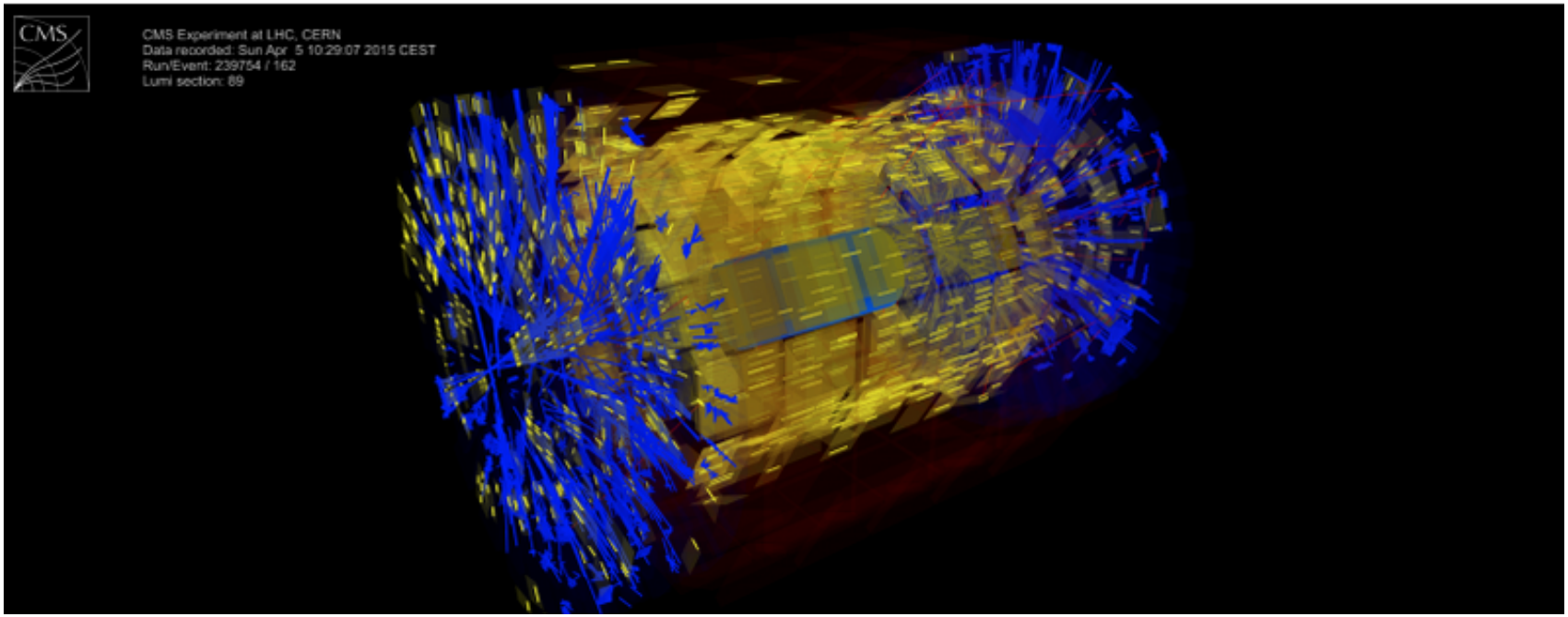
Mergers delays sample

SMTS Behaviour

- mostly stable behaviour in 1 year of production running mode
- general latencies within the requirements
- a few notable glitches, have been followed up and mostly solved

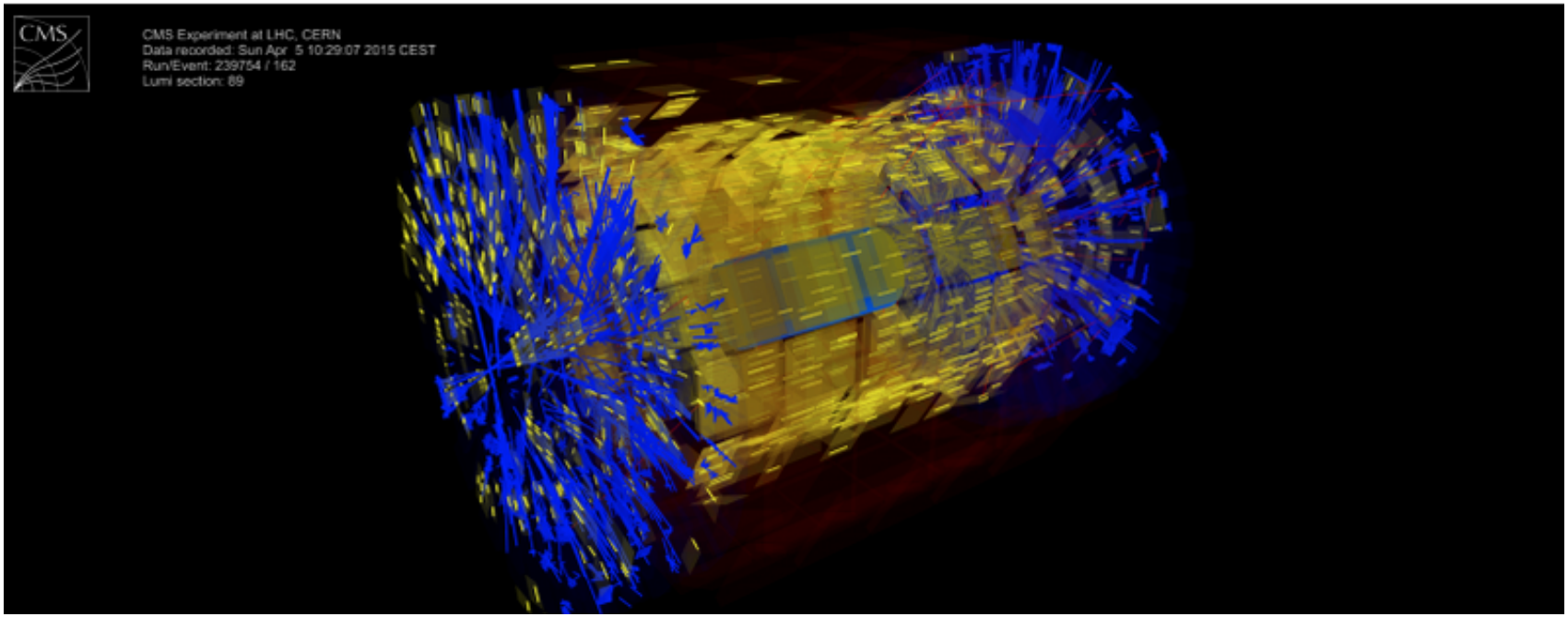
Questions?

Stories...



Event display of one of the first particle splashes seen in CMS during Run2

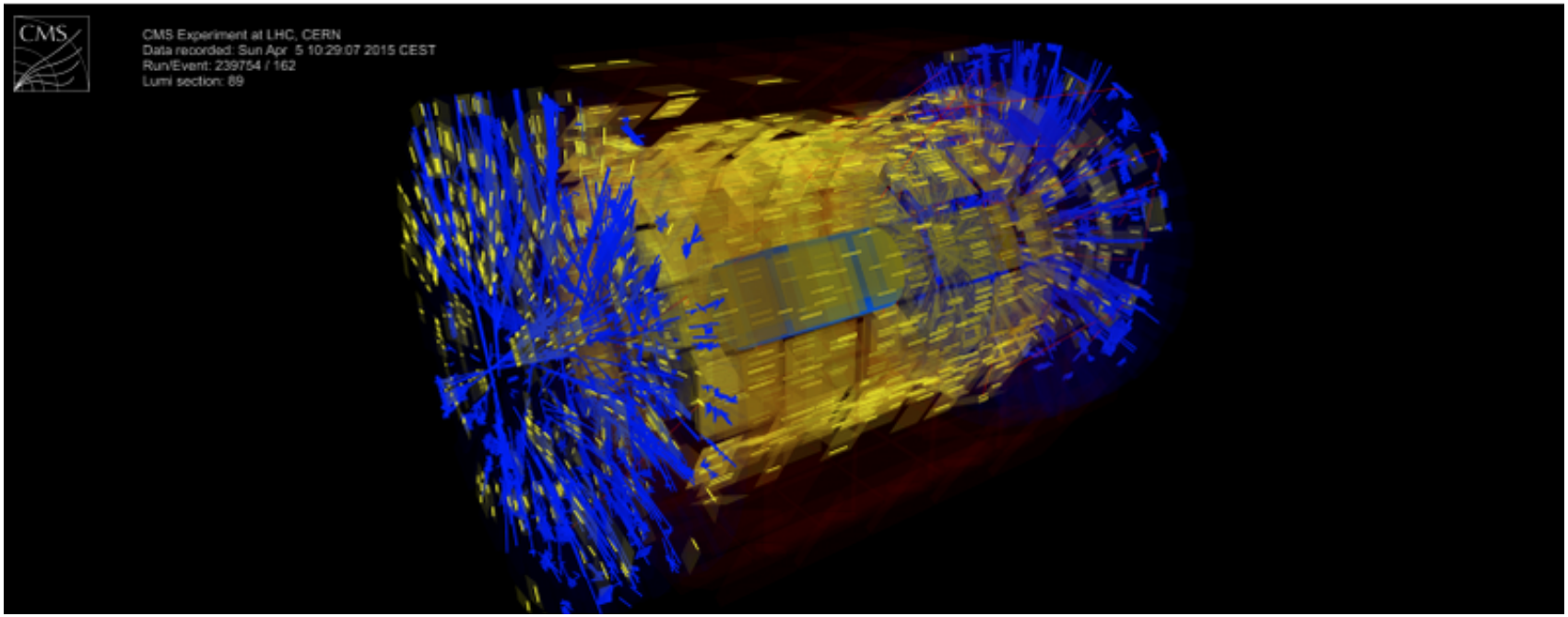
Stories...



Event display of one of the first particle splashes seen in CMS during Run2

... only a few minutes before one of the OSS servers crashed...

Stories...



Event display of one of the first particle splashes seen in CMS during Run2

... only a few minutes before one of the OSS servers crashed...
... and the failover mechanism failed ...