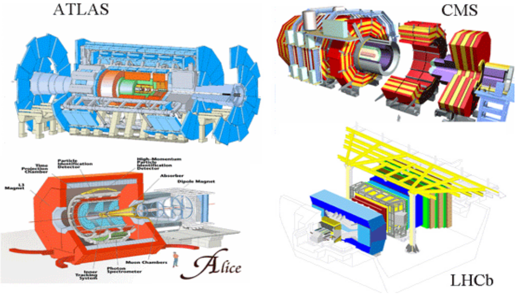
The Storage @ INFN Tier1: status and perspective

Luca dell'Agnello
INFN-CNAF

Ferrara, July 6 2011

INFN-CNAF

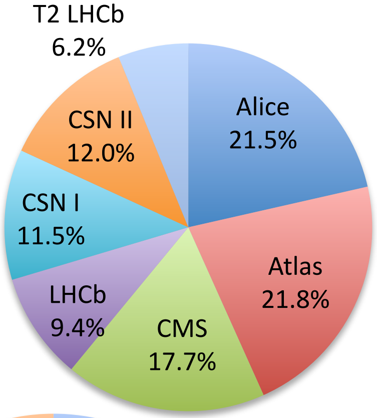
Year	CPU power [HS06]	Disk Space [PB]	Tape Space [PB]
2009	23k	2.4	2.5
2010	68k	6.6	6.6
2011	86K	9	10

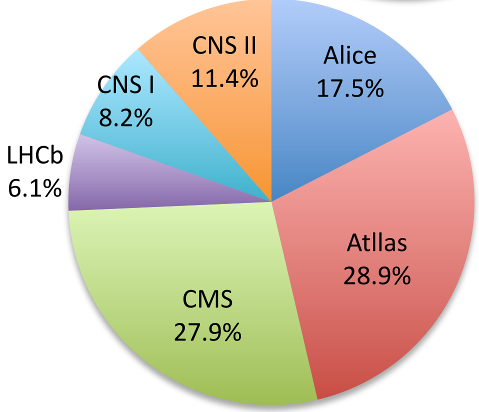


CNAF is the central computing facility of INFN

- Italian Tier-1 computing centre for the LHC experiments ATLAS, CMS, ALICE and LHCb...
- ... but also one of the main Italian processing facilities for several other experiments:
  - BaBar and CDF
  - Astro and Space physics
  - VIRGO (Italy), ARGO (Tibet), AMS (Satellite), PAMELA (Satellite), AUGER (Argentina) and MAGIC (Canary Islands)
  - More...
- Also the main computing center for SuperB ☺

CPU shares

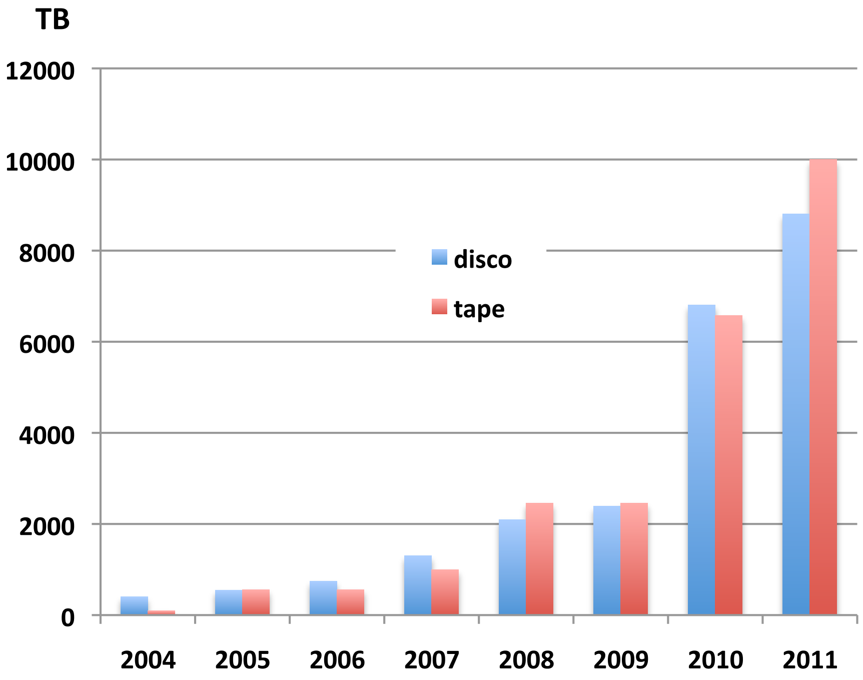




Disk shares

Our starting point

- INFN Tier1 since 2002-2003
- Goal: find a common storage solution for all experiments (VOs)
  - Fitting LHC VOs requirements…
    - Scalable up to O(10) PB
    - Offering HSM capabilities to dynamically archive and recall files from tape
    - Thousands of concurrent accesses
    - Aggregate throughput: O(10) GB/s
  - ...but also flexible for non LHC experiments requirements
  - Enabling both local and grid access
  - Overall requirements: easiness of management, stability and high availability



Storage resources at Tier1

- Our first choice was CASTOR…..
  - In our experience not very stable and easy to manage
- …. then GEMSS a new HSM system based on GPFS (parallel fs by IBM)
  - Phase out of CASTOR started in 2007 and has been completed end of 2010

Why GPFS

Original idea since the very beginning: we did not like to rely on a tape centric system

- First think to the disk infrastructure, the tape part will come later (if still needed)
- the user load is on the disk anyway

We wanted to follow a model based on well established industry standard as far as the fabric infrastructure was concerned

- Storage Area Network via FC for disk-server to disk-controller interconnections

This lead quite naturally to the adoption of a clustered file-system able to exploit the full SAN connectivity to implement flexible and highly available services

There was a (major) problem at that time: a specific SRM implementation was missing

- This lead to the development of StoRM

Basics of how GPFS works

The idea behind a parallel file-system is in general to stripe files amongst several servers and several disks

- ✦ This means that, e.g., replication of the same (hot) file in more instances is useless → you get it "for free"

Any "disk-server" can access every single device with direct access

- ✦ Storage Area Network via FC for disk-server to disk-controller interconnection (usually a device/LUN is some kind of RAID array)
- ✦ In a few words, all the servers share the same disks, but a server is primarily responsible to serve via Ethernet just some disks to the computing clients
- ✦ If a server fails, any other server in the SAN can take over the duties of the failed server, since it has direct access to its disks

All file-system metadata can be saved on disk along with the data

- ✦ Dedicated fast disks for metadata improve performances
- ✦ Data and metadata are treated symmetrically, striping blocks of metadata on several disks and servers as if they were data blocks
- ✦ No need of external catalogues/DBs: it is a true file-system

Some GPFS key features

Very powerful (only command line, no other way to do it) interface for configuring, administering and monitoring the system

- ✦ In our experience this is the key feature which allowed to keep minimal manpower to administer the system
  - ✦ 1 FTE to control every operation (and scaling with increasing volumes is quite flat)
- ✦ Needs however some training to startup, it is not plug and pray… but documentation is huge and covers (almost) every relevant detail

100% POSIX compliant by design

Limited amount of HW resources needed (see later for an example)

Support for cNFS file-system export to clients (parallel NFS server solution with full HA capabilities developed by IBM)

Statefull connections between "clients" and "servers" are kept alive behind the data access (file) protocol

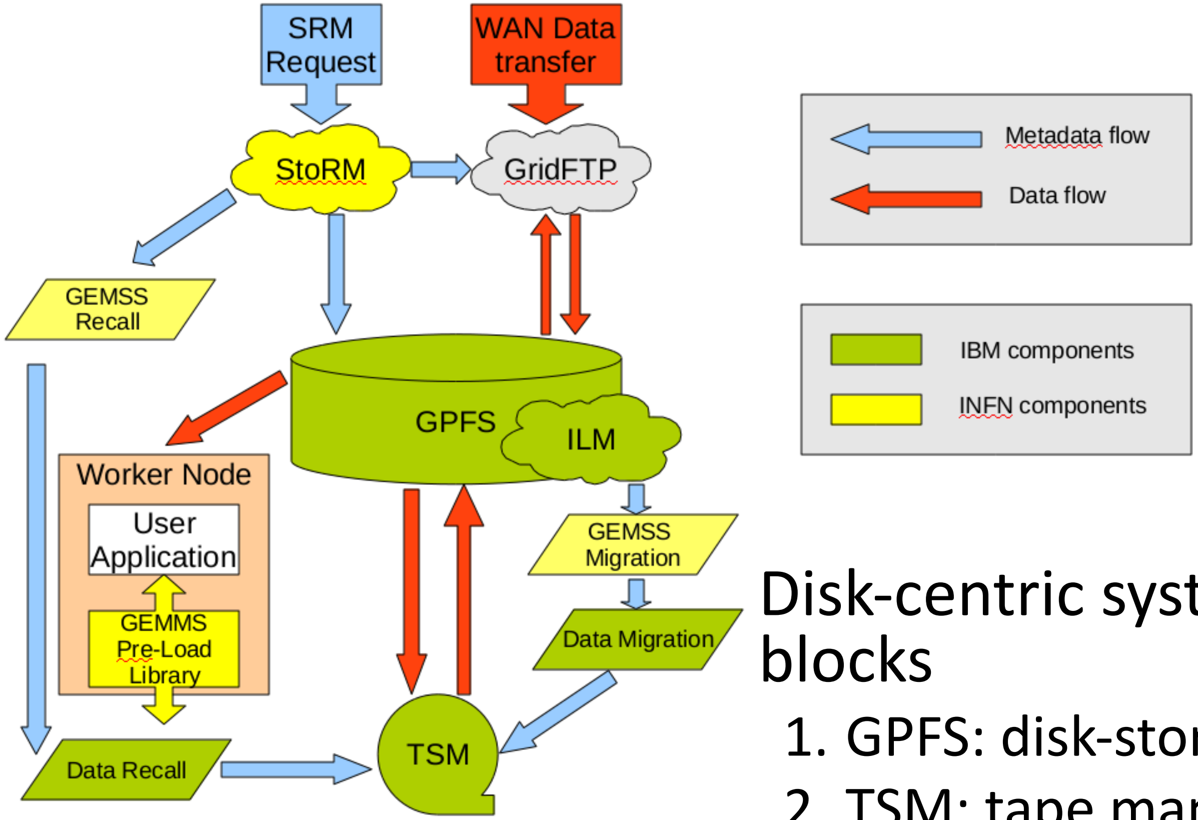
- ✦ No need of things like "reconnect" at the application level

Native HSM capabilities (not only for tapes, but also for multi-tiered disk storage)

GEMSS

- GEMSS is the integration of GPFS with StoRM and TSM providing a transparent grid-enabled HSM solution.
  - GPFS deployed on the SAN implements a full HA system
  - StoRM is an srm 2.2 implementation developed by INFN-CNAF
    - Already in use at INFN T1 since 2007 and at other centers for the disk-only storage
    - designed to leverage the advantages of parallel file systems and common POSIX file systems in a Grid environment
  - TSM is a tape back-end storage by IBM
- Native POSIX (i.e. access protocol ‘file’) for direct access from the farm
  - Possible to bypass srm for reading (speeding up the access)
- WAN access provided via gridftp
- Xrootd possible (just a protocol on top of the storage)
    - (Very) low efficiency of Alice jobs under investigation

Building blocks of GEMSS system

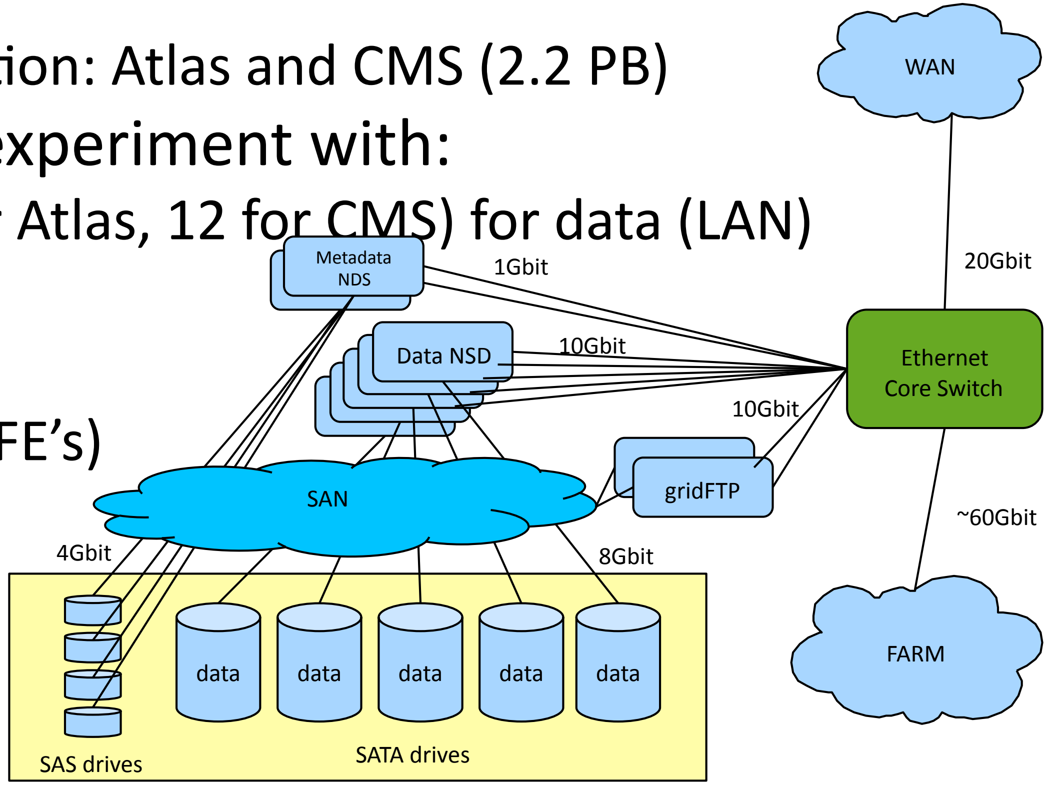


Disk-centric system with five building blocks

1. GPFS: disk-storage software infrastructure
2. TSM: tape management system
3. StoRM: SRM service
4. TSM-GPFS interface
5. Globus GridFTP: WAN data transfers

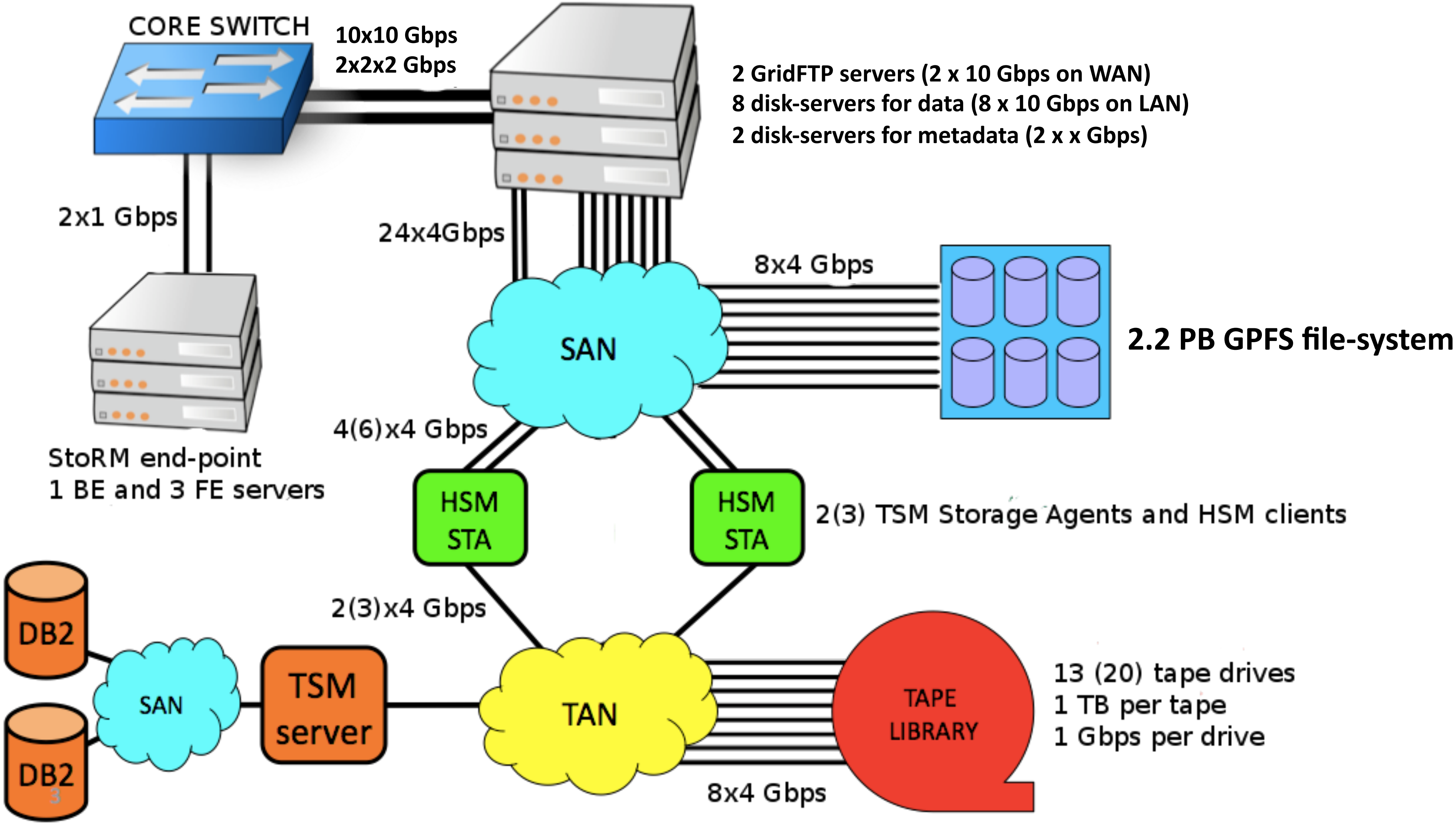
Present CNAF storage setup

- Disk storage (~ 9 PB under GEMSS) partitioned in several GPFS clusters
  - Largest file-systems in production: Atlas and CMS (2.2 PB)
- One cluster for each (major) experiment with:
  - Several disk-servers (e.g. 8 for Atlas, 12 for CMS) for data (LAN)
  - 2 disk-servers for metadata
  - 2-4 gridftp servers (WAN)
  - 1 storm end-point (1 BE + 2-4 FE's)
  - 2-3 tsm-hsm servers (for access to tape)
- Storage aggregate bw: ~ 40 GBps (10 GE servers)



- 1 tape library Sl8500 (10 PB on line) with 20 T10Kb drives
  - 1 TB tape capacity, 1 Gbps of bandwidth for each drive
  - Drives interconnected to library and tsm-hsm servers via dedicated SAN (TAN)
  - TSM server common to all GEMSS instances
- All storage systems and disk-servers interconnected via SAN (FC4/FC8)

GEMSS layout for a typical Experiment at INFN Tier-1



2 GridFTP servers (2 x 10 Gbps on WAN)
8 disk-servers for data (8 x 10 Gbps on LAN)
2 disk-servers for metadata (2 x x Gbps)

2.2 PB GPFS file-system

StoRM end-point
1 BE and 3 FE servers

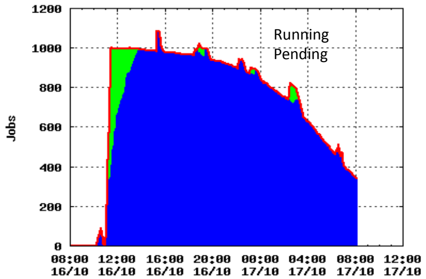
2(3) TSM Storage Agents and HSM clients

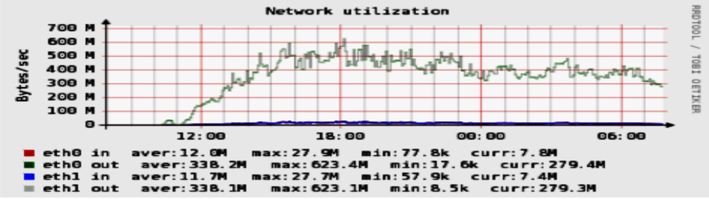
13 (20) tape drives
1 TB per tape
1 Gbps per drive

GEMSS in production

- *Gbit technology* (2009)
  - Using the file protocol (i.e. direct access to the file)
  - Up to 1000 concurrent jobs recalling from tape ~ 2000 files
    - 100% job success rate
    - Up to 1.2 GB/s from the disk pools to the farm nodes

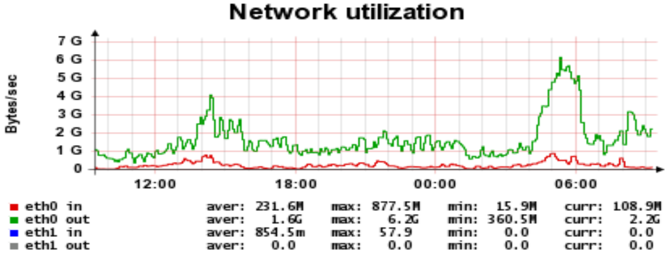
Running and pending jobs on the farm



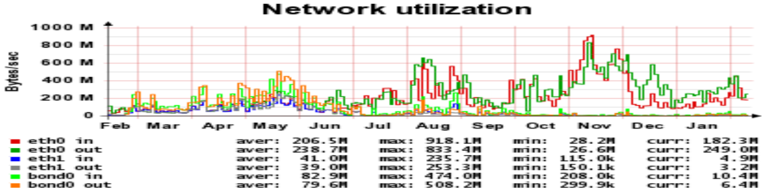


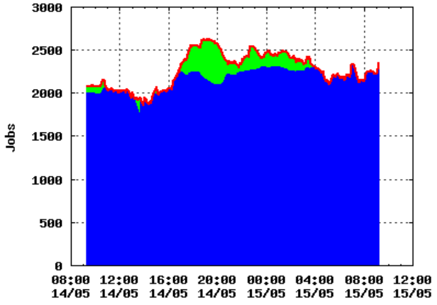
Aggregate traffic on eth0 network cards (x2)

- *10 Gbit technology* (since 2010)
  - Using the file protocol
  - Up to 2500 concurrent jobs accessing files on disl
    - ~98% job success rate
    - Up to ~ 6 GB/s from the disk pools to the farm nodes
  - WAN links towards saturation



Farm- CMS storage traffic





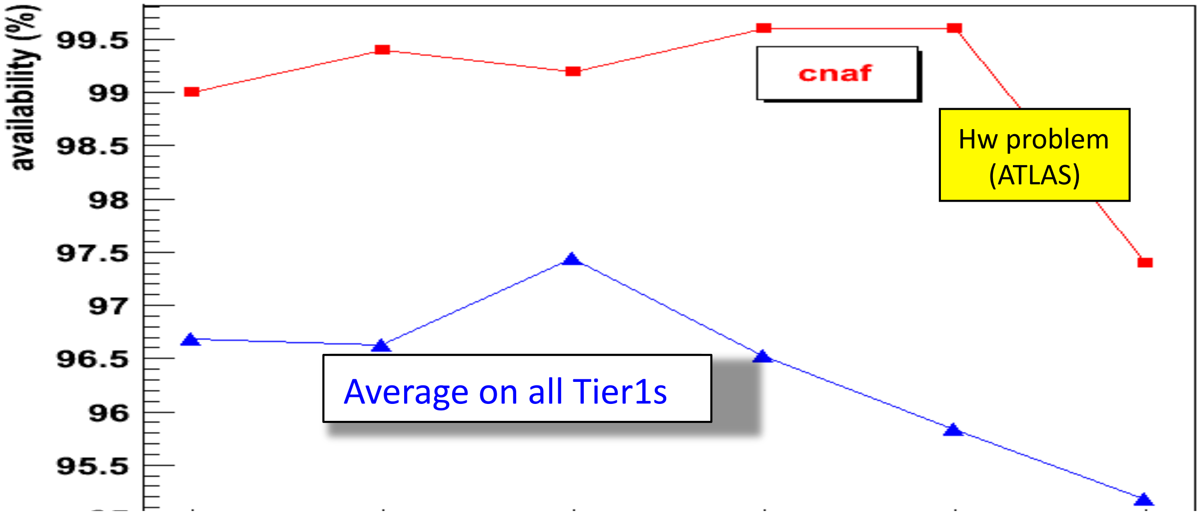
CMS queue (May 15)

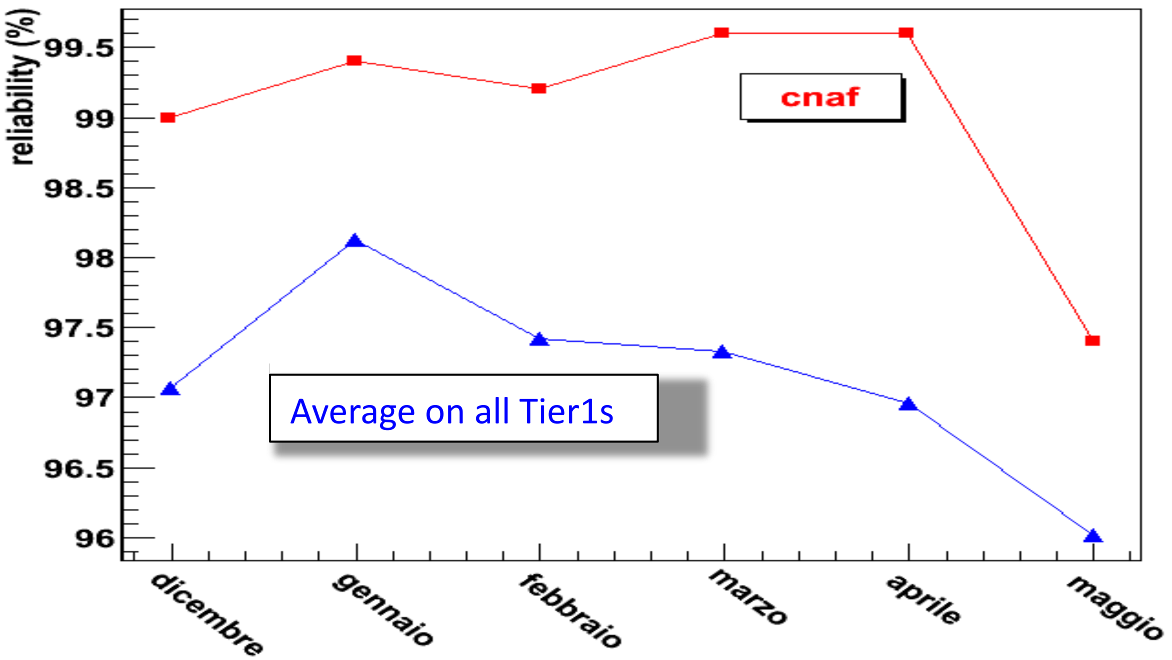
INFN T1 availability and reliability

$$availability = \frac{upTime}{totTime - unkTime}$$

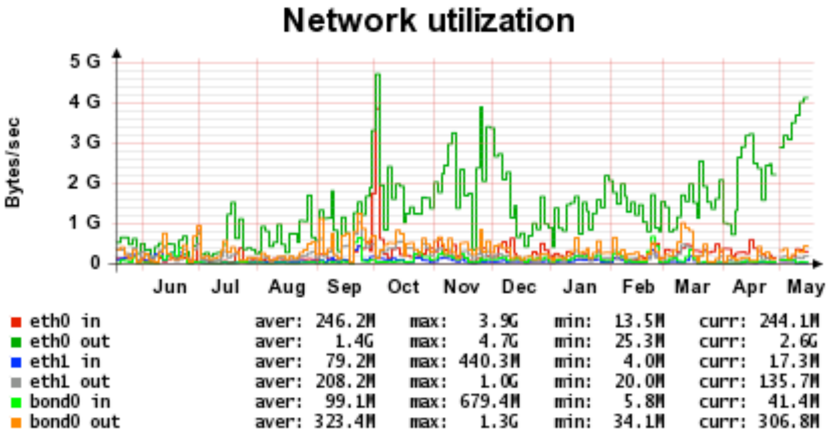
From December 2010 to May 2011.

$$reliability = \frac{upTime}{totTime - scheddownTime - unkTime}$$



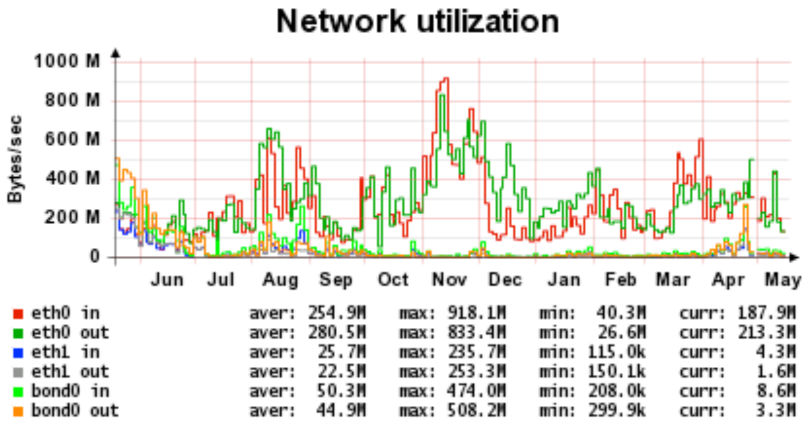


Yearly statistics



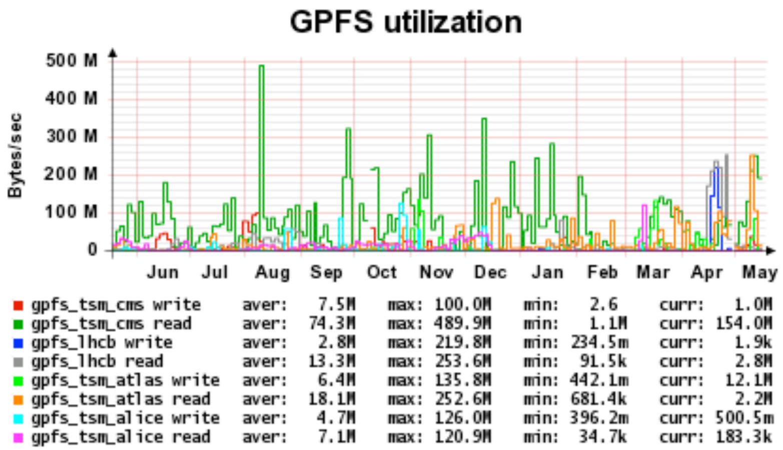
```
🟥 eth0 in        aver: 246.2M   max:   3.9G   min:  13.5M   curr: 244.1M
🟩 eth0 out       aver:   1.4G   max:   4.7G   min:  25.3M   curr:   2.6G
🟦 eth1 in        aver:  79.2M   max: 440.3M   min:   4.0M   curr:  17.3M
⬜ eth1 out       aver: 208.2M   max:   1.0G   min:  20.0M   curr: 135.7M
🟩 bond0 in       aver:  99.1M   max: 679.4M   min:   5.8M   curr:  41.4M
🟧 bond0 out      aver: 323.4M   max:   1.3G   min:  34.1M   curr: 306.8M
```

Aggregate GPFS traffic (file protocol)



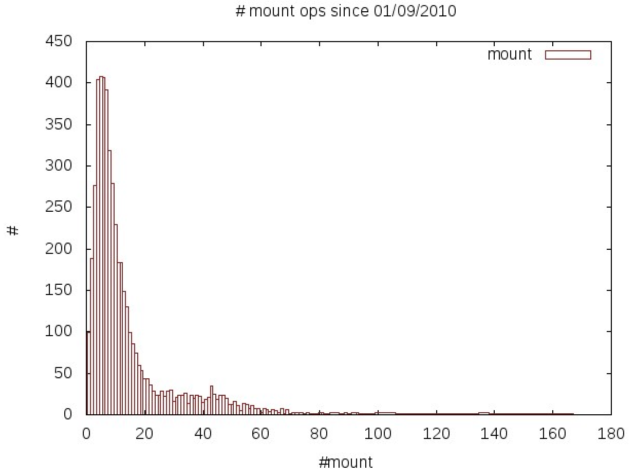
```
🟥 eth0 in        aver: 254.9M   max: 918.1M   min:  40.3M   curr: 187.9M
🟩 eth0 out       aver: 280.5M   max: 833.4M   min:  26.6M   curr: 213.3M
🟦 eth1 in        aver:  25.7M   max: 235.7M   min: 115.0k   curr:   4.3M
⬜ eth1 out       aver:  22.5M   max: 253.3M   min: 150.1k   curr:   1.6M
🟩 bond0 in       aver:  50.3M   max: 474.0M   min: 208.0k   curr:   8.6M
🟧 bond0 out      aver:  44.9M   max: 508.2M   min: 299.9k   curr:   3.3M
```

Aggregate WAN traffic (gridftp)



```
🟥 gpfs_tsm_cms write     aver:   7.5M   max: 100.0M   min:   2.6     curr:   1.0M
🟩 gpfs_tsm_cms read      aver:  74.3M   max: 489.9M   min:   1.1M    curr: 154.0M
🟦 gpfs_lhcb write        aver:   2.8M   max: 219.8M   min: 234.5m    curr:   1.9k
⬜ gpfs_lhcb read         aver:  13.3M   max: 253.6M   min:  91.5k    curr:   2.8M
🟩 gpfs_tsm_atlas write   aver:   6.4M   max: 135.8M   min: 442.1m    curr:  12.1M
🟧 gpfs_tsm_atlas read    aver:  18.1M   max: 252.6M   min: 681.4k    curr:   2.2M
🟦 gpfs_tsm_alice write   aver:   4.7M   max: 126.0M   min: 396.2m    curr: 500.5m
🟪 gpfs_tsm_alice read    aver:   7.1M   max: 120.9M   min:  34.7k    curr: 183.3k
```

Tape-disk data movement (over the SAN)



Mounts/hour

What's next?

- Strategy: stay on standards (and keep it simple!)
- Parallel file-systems (and SAN) are in our opinion the right choice
  - GPFS is **now** the only viable solution to have also an HSM
    - Easiness of coupling with a tape system
  - But in the long term tapes will be used as a pure archive
- Looking for NFS 4.1 based solutions
  - Possibly integrated in the hw itself
  - Extreme simplification of the infrastructure
- http as a possible alternative to gridftp
  - This is part of EMI working plan for StoRM

Summary of our experience

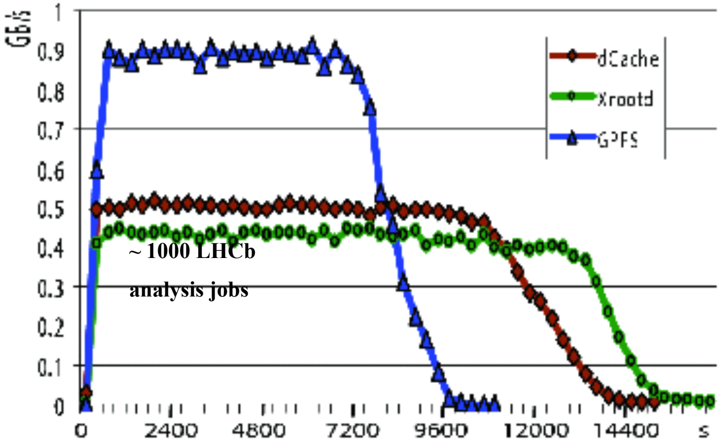
- Excellent stability of the system
  - Good feedback from experiments (not only LHC!)
- Reduced management effort
  - 4 FTE to manage and maintain all the system (sw layer, SAN, library, servers,…)
  - 9 PB of disk + 1x10 PB library
- Fabric infrastructure based on industry standards
  - Storage Area Network via FC for disk-server to disk-controller interconnections
  - clustered file-system (GPFS) to be able to fully exploit the SAN
  - Flexibility and HA by design
- Focus on standards also for data access…..
  - File protocol for local access
  - Gridftp for remote access
- ….but also flexible for legacy protocols
  - xrootd available (for Alice), bbftp for VIRGO etc..
- Looking now at new emerging standards for storage access
  - NFS 4.1 for parallel file-systems
  - http (webdav) for remote access

Backup slides

Mass Storage System at CNAF: the evolution (1)

- 2003: CASTOR chosen as MSS (and phased out Jan 2011)
  - Large variety of issues both at set-up/admin level and at VO's level (complexity, scalability, stability, support)

- 2007: start of a project to realize GEMSS, a new grid-enabled HSM solution based on industrial components (parallel file-system and standard archival utility)
  - StoRM adopted as SRM layer and extended to include the methods required to manage data on tape
  - GPFS and TSM by IBM chosen as building blocks
  - An interface between GPFS and TSM implemented (not all needed functionalities provided out of the box)



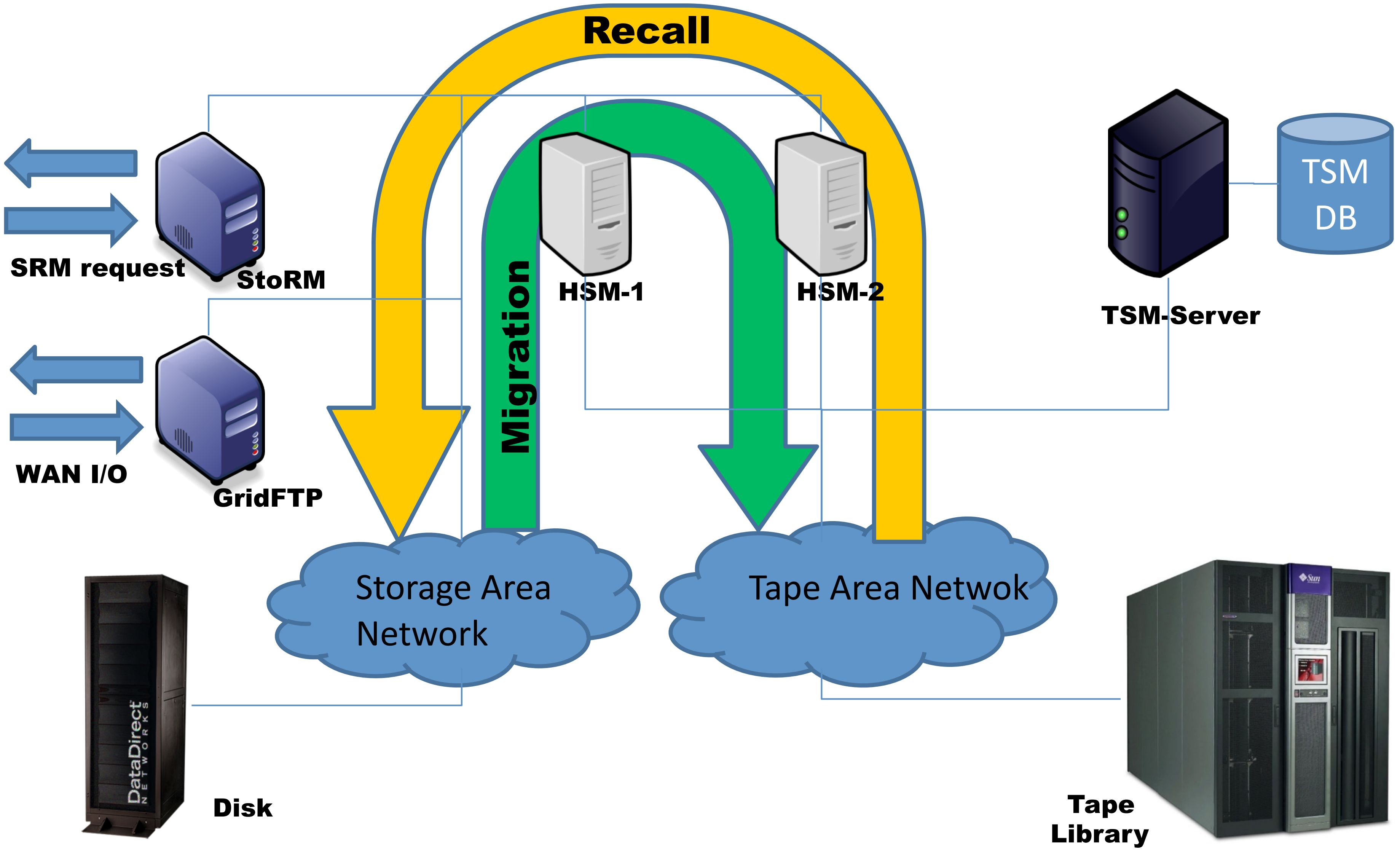
~ 1000 LHCb analysis jobs

GPFS validation test (2007)

Mass Storage System at CNAF: the evolution (2)

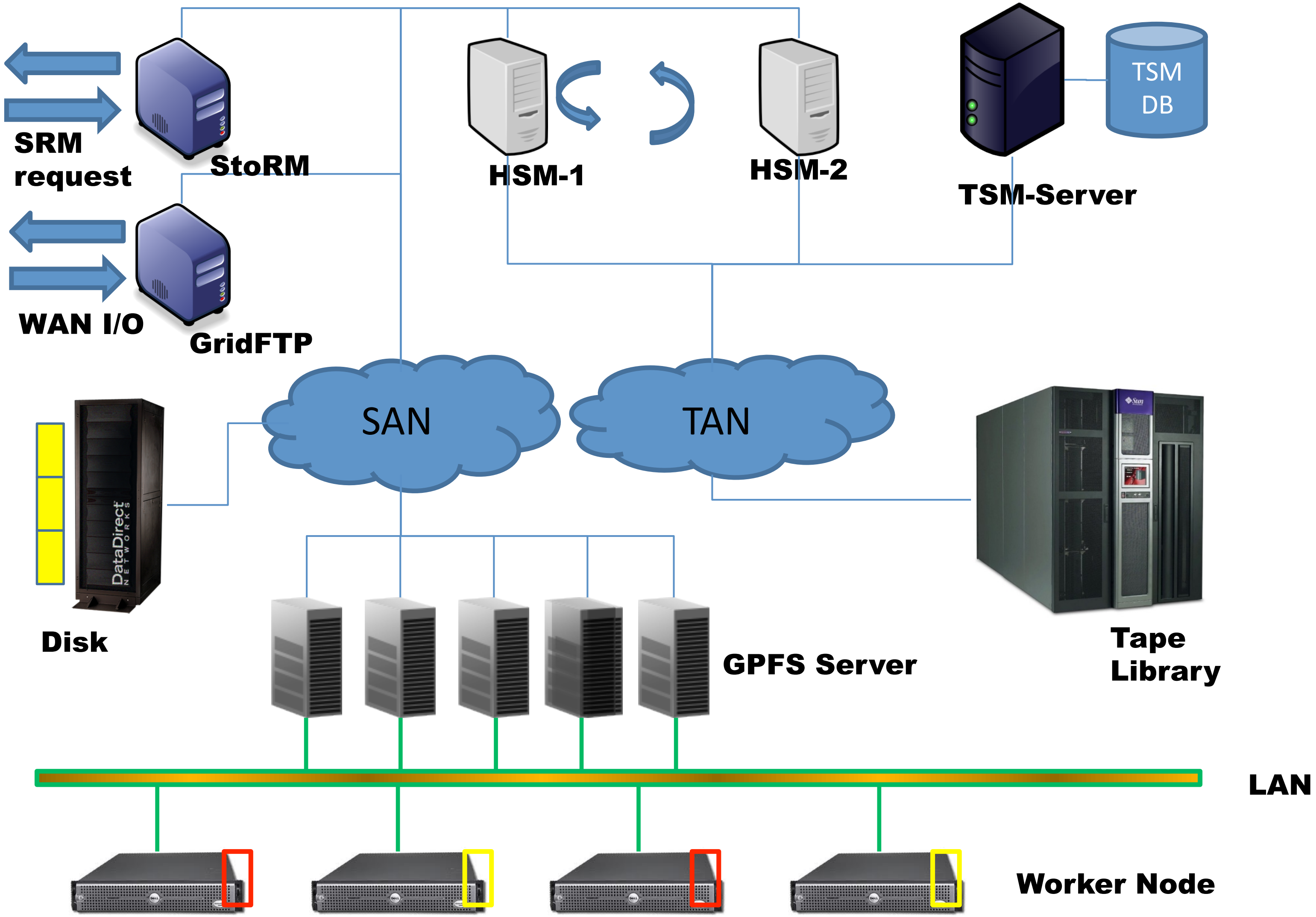
- Q2 2008: First implementation (D1T1, the easy case) in production for LHCb (CCRC'08)
- Q2 2009: GEMSS (StoRM/GPFS/TSM), the full HSM solution, ready for production
- Q3 2009: CMS moving from CASTOR to GEMSS
- Q1 2010: the other LHC experiments moving to GEMSS
- End of 2010: all other experiments moved from CASTOR to GEMSS
    - All data present on CASTOR tapes copied to TSM tapes
    - CASTOR tapes recycled after data check

GEMSS data flow (1/2)

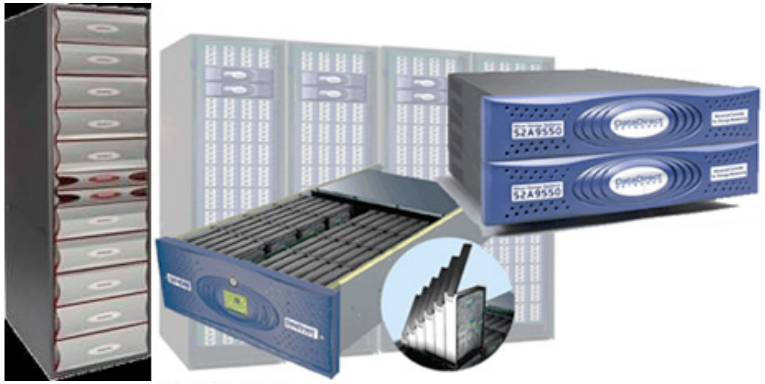


GEMSS data flow (2/2)

Sorting Files by Tape



Storage resources



- 9 PB of disk on-line under GEMSS
  - 7 DDN S2A9950 (2 TB SATA disks for data, 300 GB SAS disks for metadata)
  - 7 EMC 3-80 + 1 EMC 4-960
- Max storage aggregate bw: ~ 40 GBps
  - LAN based on 10 Gbps Ethernet
    - ~ 40 10Gbps servers connected to core switch
    - ~ 60 1Gbps servers to aggregation switches
  - WAN: 2 x 10 Gbps links to OPN + 1 10 Gbps to GIN
    - ~ 10 10Gbps gridFtp servers + ~ 10 1 Gbps gridftp servers

- 1 tape library Sl8500 (10 PB on line) with 20 T10Kb drives
  - 1 TB tape capacity, 1 Gbps of bandwidth for each drive
  - Drives interconnected to library and tsm-hsm servers via dedicated SAN (TAN)
  - TSM server common to all GEMSS instances
- All storage systems and disk-servers interconnected via SAN (FC4/FC8)

GEMSS in production for CMS

GEMSS went in production for CMS in October 2009

✦w/o major changes to the layout

- only StoRM upgrade, with checksum and authz supportbeing deployed soon also

Good-performance achieved in transfer throughput

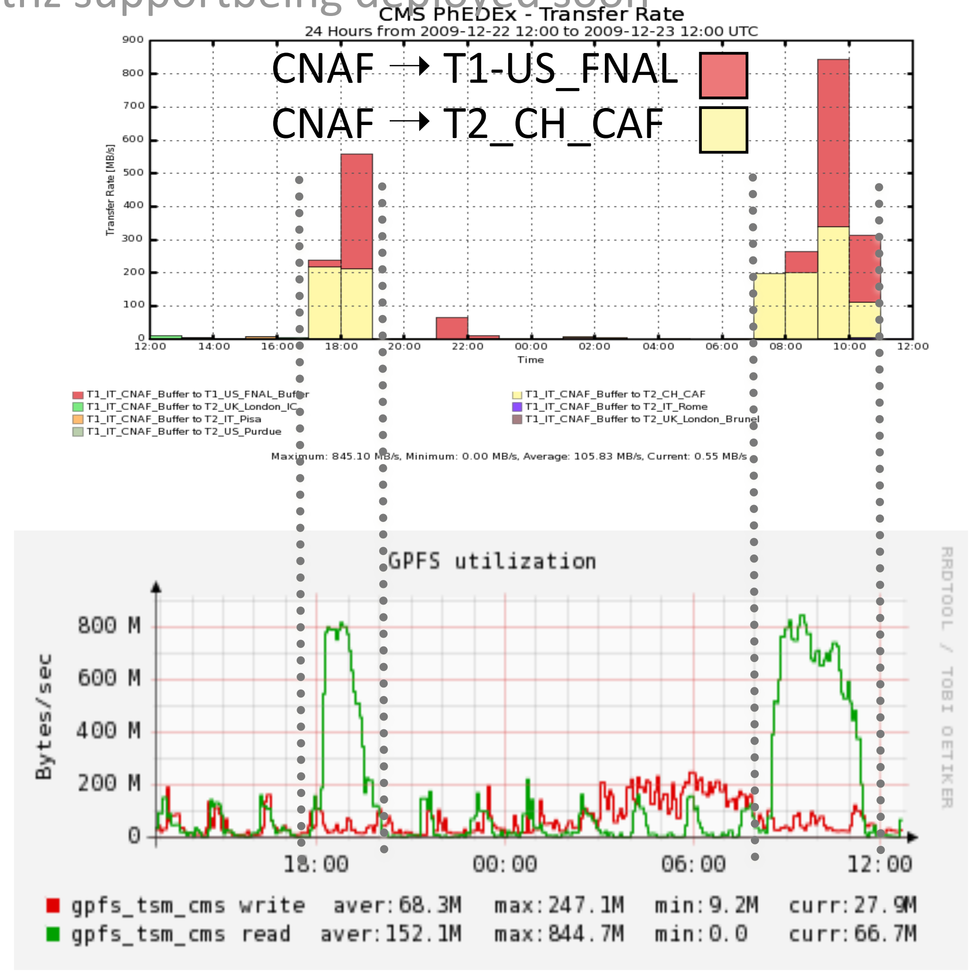
- High use of the available bandwidth
- (up to 8 Gbps)

Verification with Job Robot jobs in different periods shows that CMS workflows efficiency was not impacted by the change of storage system

- “Castor + SL4” vs “TSM + SL4” vs “TSM + SL5”

As from the current experience, CMS gives a very positive feedback on the new system

- Very good stability observed so far


CMS PhEDEx - Transfer Rate
24 Hours from 2009-12-22 12:00 to 2009-12-23 12:00 UTC
CNAF → T1-US_FNAL
CNAF → T2_CH_CAF
Transfer Rate [MB/s]
12:00
14:00
16:00
18:00
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
12:00
Time
T1_IT_CNAF_Buffer to T1_US_FNAL_Buffer
T1_IT_CNAF_Buffer to T2_UK_London_IC
T1_IT_CNAF_Buffer to T2_IT_Pisa
T1_IT_CNAF_Buffer to T2_US_Purdue
T1_IT_CNAF_Buffer to T2_CH_CAF
T1_IT_CNAF_Buffer to T2_IT_Rome
T1_IT_CNAF_Buffer to T2_UK_London_Brunel
Maximum: 845.10 MB/s, Minimum: 0.00 MB/s, Average: 105.83 MB/s, Current: 0.55 MB/s
GPFS utilization
Bytes/sec
800 M
600 M
400 M
200 M
18:00
00:00
06:00
12:00
RRDTOOL / TOBI OETIKER
gpfs_tsm_cms write aver:68.3M max:247.1M min:9.2M curr:27.9M
gpfs_tsm_cms read aver:152.1M max:844.7M min:0.0 curr:66.7M