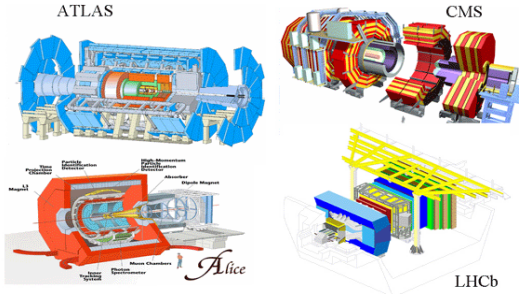
GEMSS

Luca dell'Agnello
INFN-CNAF

Barcelona, May 30 2011

INFN-CNAF

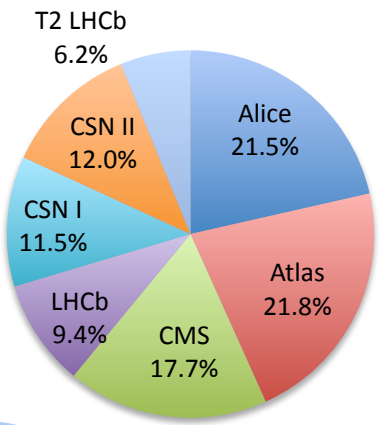
Year	CPU power [HS06]	Disk Space [PB]	Tape Space [PB]
2009	23k	2.4	2.5
2010	68k	6.6	6.6
2011	86K	9	10



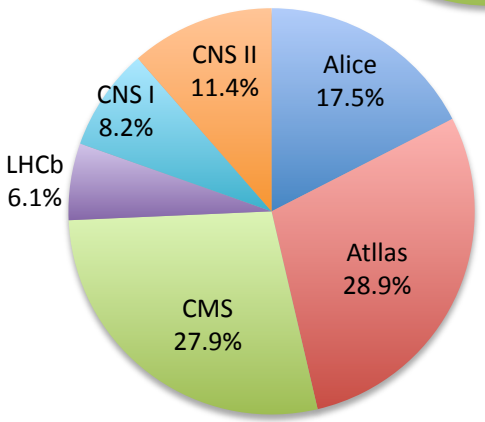
CNAF is the central computing facility of INFN

- 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) and MAGIC (Canary Islands)
  - More...

CPU shares



Disk shares



CNAF in the grid

- CNAF is part of the WLCG/EGI infrastructure, granting access to distributed computing and storage resources
  - Access to computing farm via the EMI CREAM Compute Elements
  - Access to storage resources, on GEMSS, via the srm end-points
  - Also “legacy” access (i.e. local access allowed)
- Some typical grid acronyms for storage:
  - SE (Storage Element) a Grid service that allows Grid users to store and manage files together with the space assigned to them.
  - SRM (Storage Resource Manager) middleware component whose function is to provide dynamic space allocation and file management in spaces for shared storage components on the Grid. Essential for bulk operations on tape system.

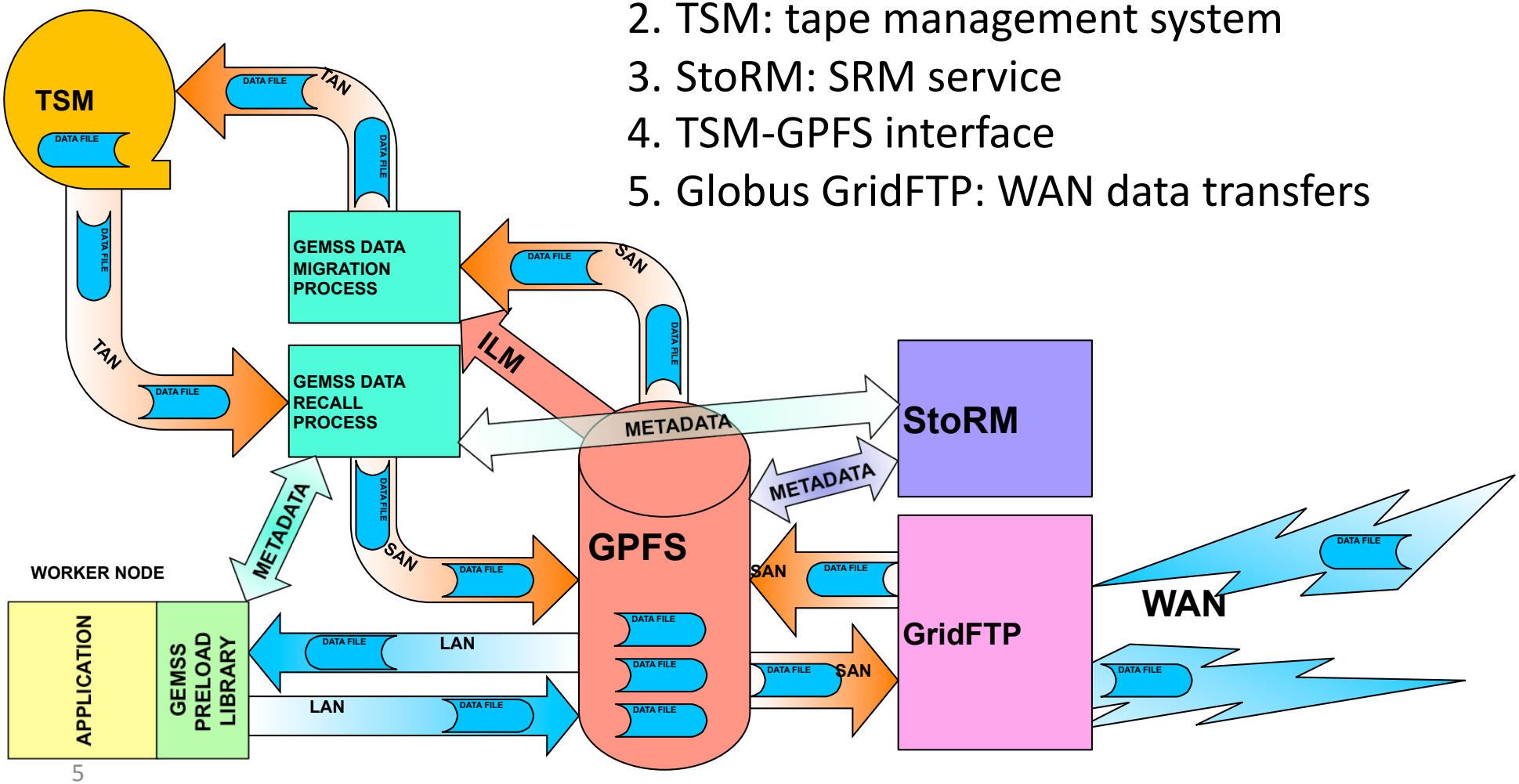
What is GEMSS?

- GEMSS is the integration of StoRM, GPFS and TSM
  - GPFS parallel file-system by IBM, TSM archival system by IBM
  - GPFS deployed on the SAN implements a full HA system
- StoRM is an srm 2.2 implementation developed by INFN
  - 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
- We combined the features of GPFS and TSM with StoRM, to provide a transparent grid-enabled HSM solution.
  - The GPFS Information Lifecycle Management (ILM) engine is used to identify candidates files for migration to tape and to trigger the data movement between the disk and tape pools
- An interface between GPFS and TSM (named YAMSS) was also implemented to enable tape-ordered recalls

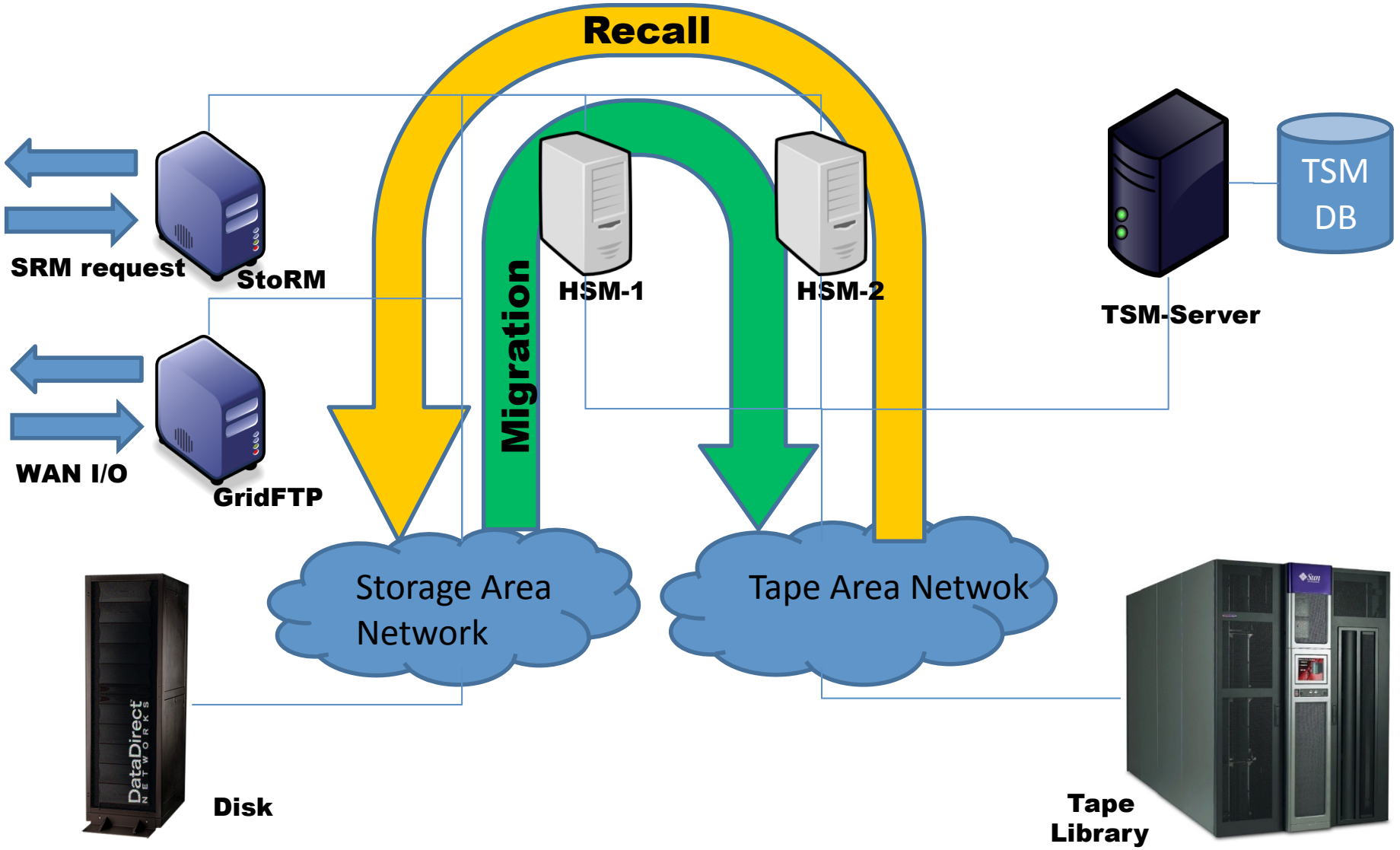
Building blocks of GEMSS system

Disk-centric system with five building blocks

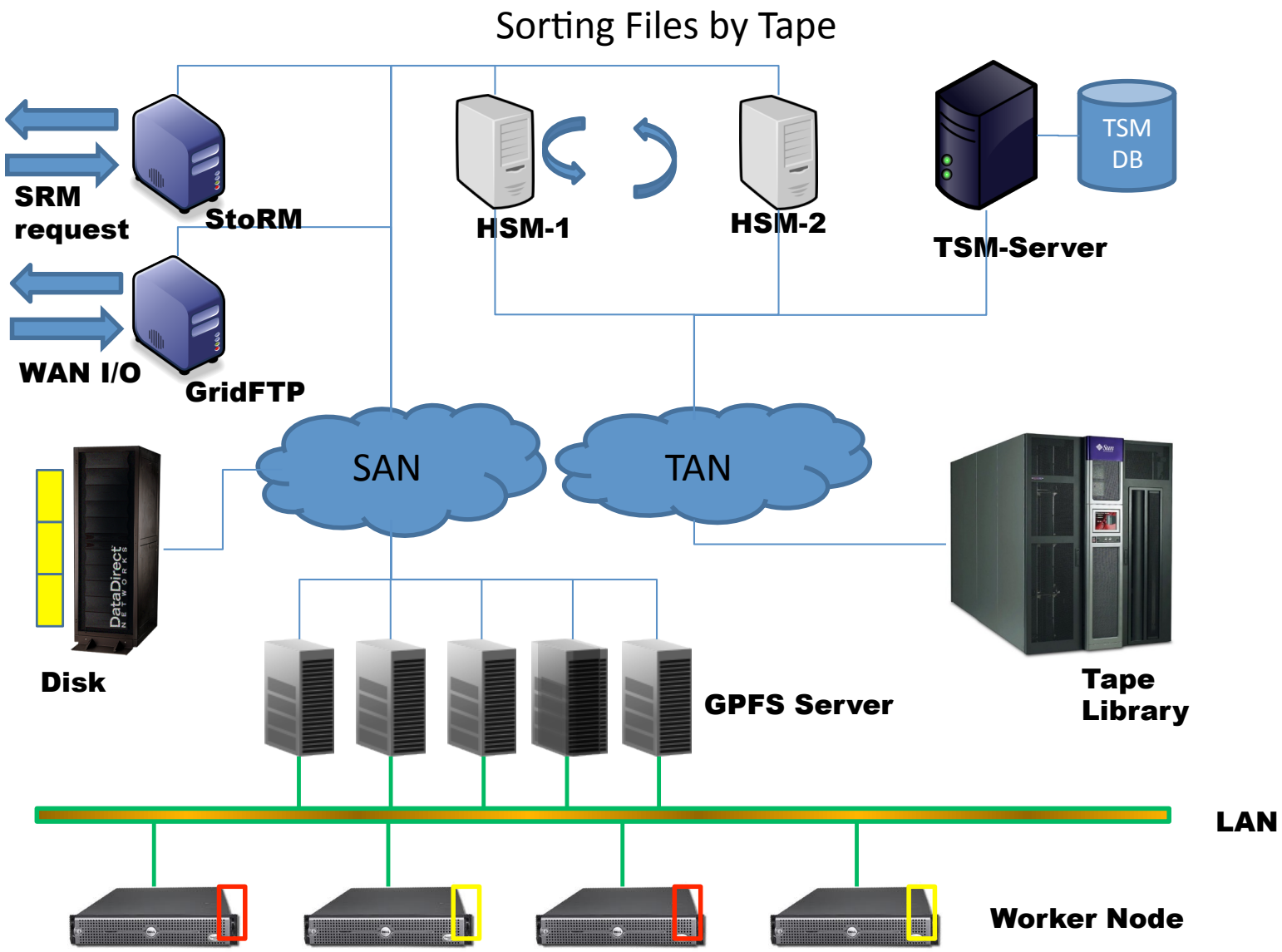
1. GPFS: disk-storage software infrastructure
2. TSM: tape management system
3. StoRM: SRM service
4. TSM-GPFS interface
5. Globus GridFTP: WAN data transfers



GEMSS data flow (1/2)



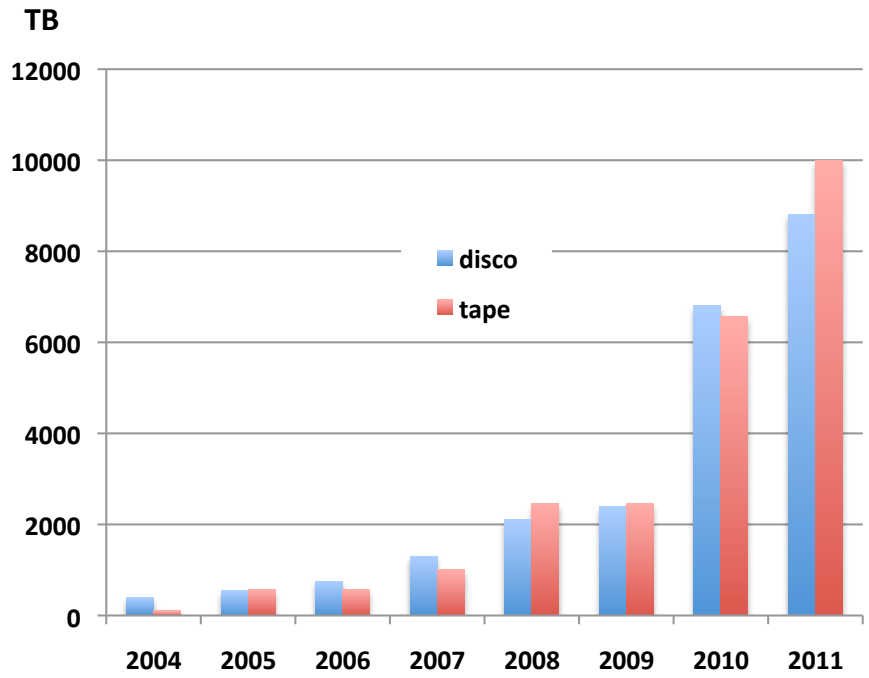
GEMSS data flow (2/2)



The requirements

- INFN Tier1 since 2002-2003
  - Some fundamental choices (later on revisited ☺)
    - batch system (torque/maui then LSF)
    - Mass Storage System (CASTOR then GEMSS)

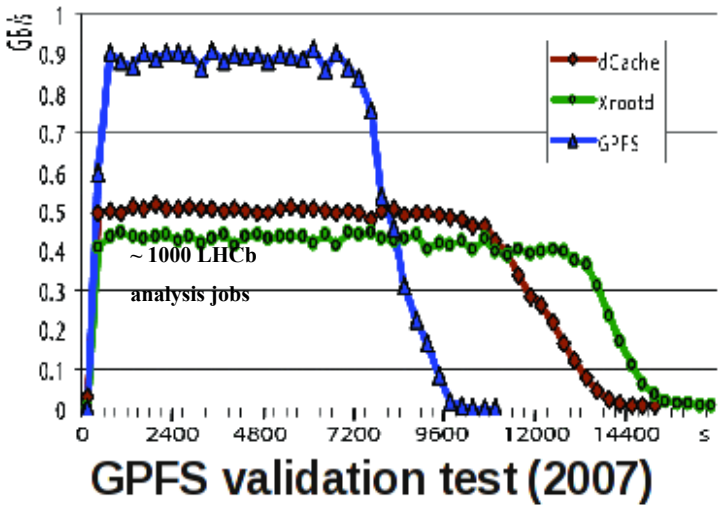
- Goal: find a common solution for all VOs
  - Fitting LHC VOs requirements…
    - Scalable up to O(10) PB
    - Offering HSM capabilities to dinamically archive and recall files from tape
    - Thousands of concurrent accesses
    - Aggregate throughput: O(10) GB/s
  - ...but also flexible for non LHC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



Storage resources at Tier1

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)



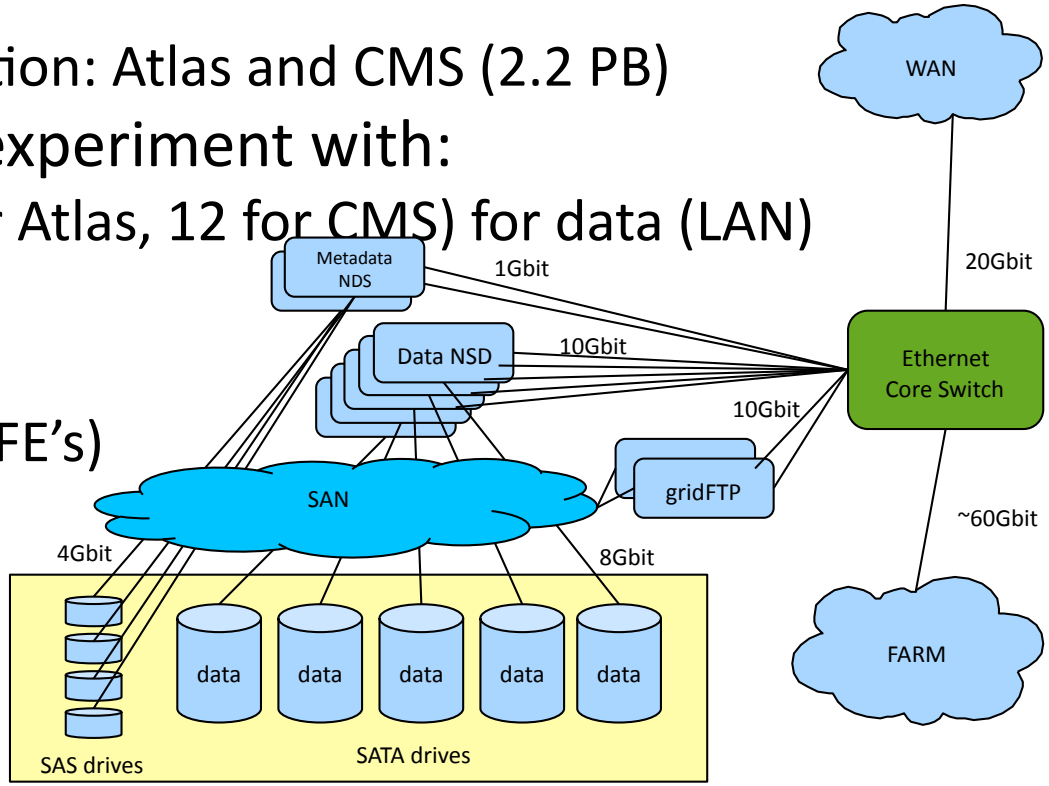
GPFS validation test (2007)

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

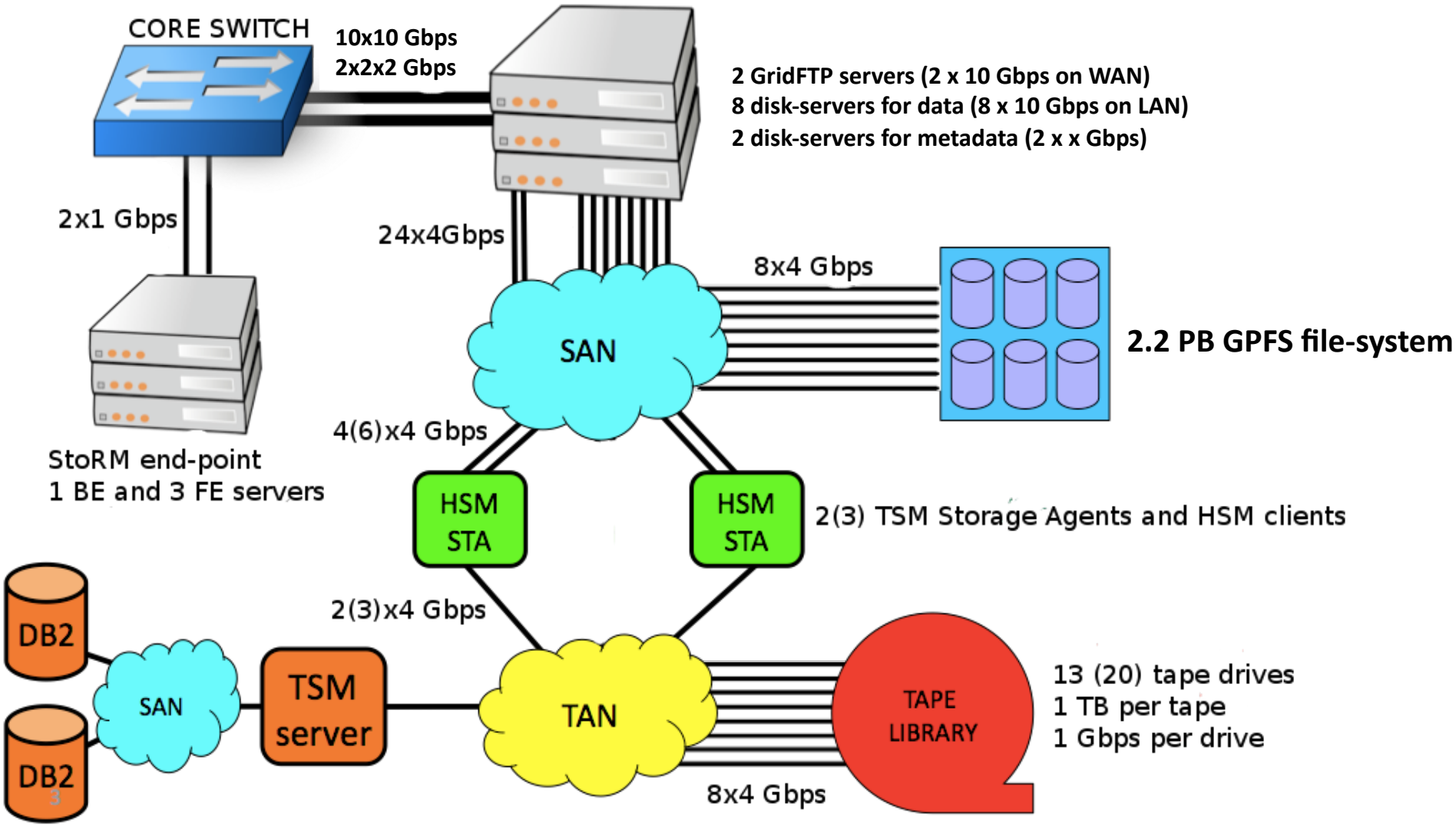
Present storage setup

- 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)



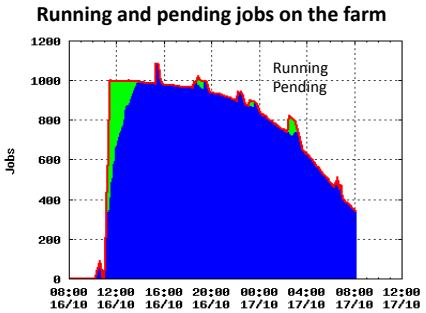
- 1 tape library Sl8500 (10 PB on line) with 20 T10Kb drives
  - 1 TB tape capacity, 1 Gbps of bandwidth for each drive
  - Drives interconnected to library and tsm-hsm servers via dedicated SAN (TAN)
  - TSM server common to all GEMSS instances
- All storage systems and disk-servers interconnected via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

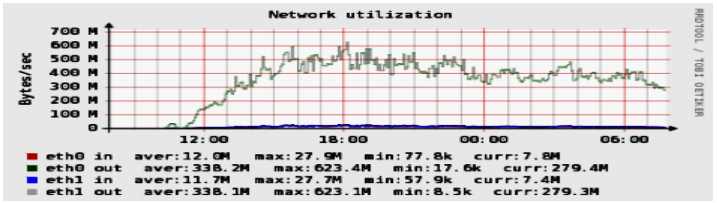
GEMSS layout for a typical Experiment at INFN Tier-1



GEMSS in production

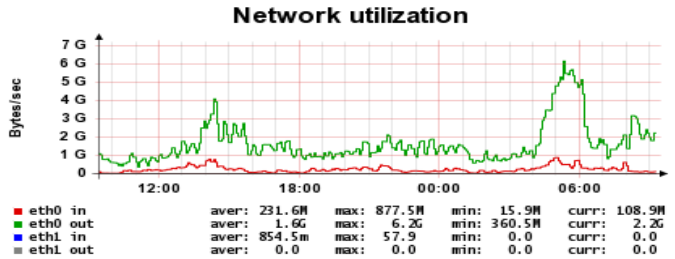
- *Gbit technology* (2009)
  - Using the file protocol (i.e. direct access to the file)
  - Up to 1000 concurrent jobs recalling from tape ~ 2000 files
    - 100% job success rate
    - Up to 1.2 GB/s from the disk pools to the farm nodes



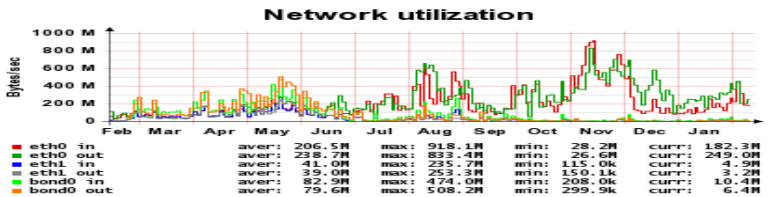


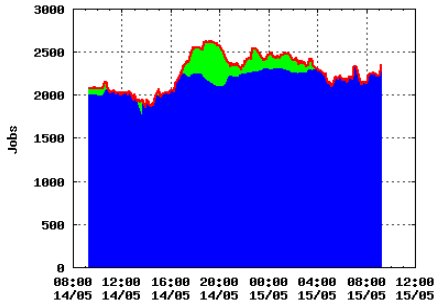
Aggregate traffic on eth0 network cards (x2)

- *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



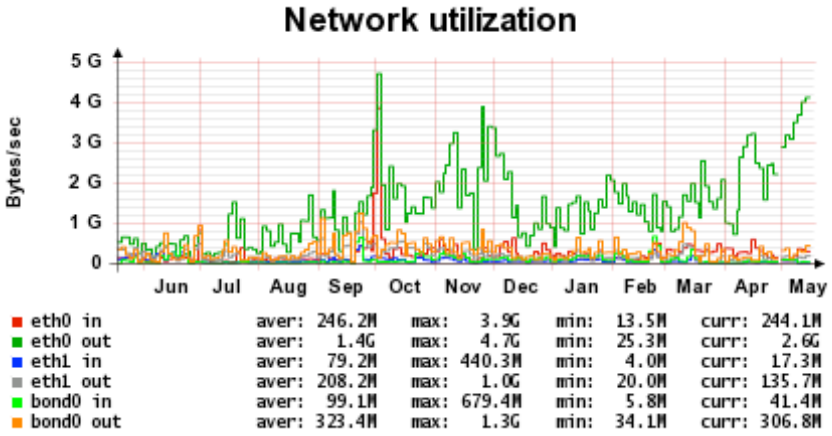
Farm- CMS storage traffic



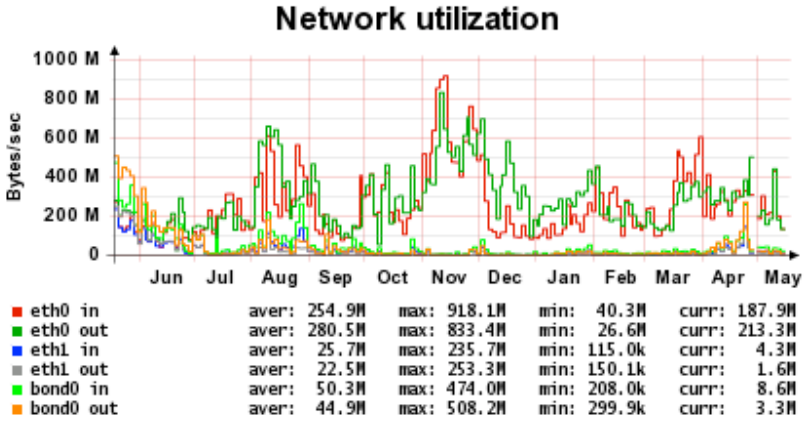


CMS queue (May 15)

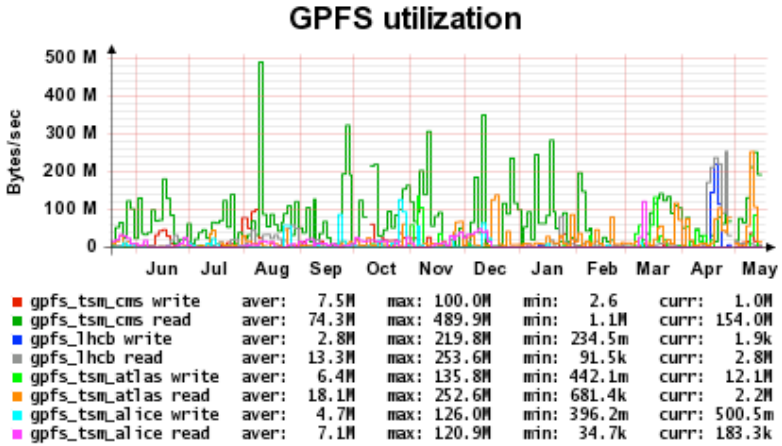
Yearly statistics



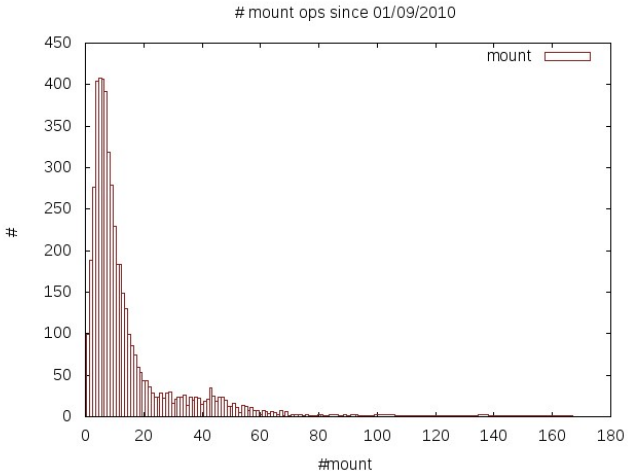
Aggregate GPFS traffic (file protocol)



Aggregate WAN traffic (gridftp)



Tape-disk data movement (over the SAN)



Mounts/hour

Summary

- Excellent stability of the system
  - Good feedback from experiments (not only LHC!)
- Reduced management effort
  - 3 FTE to manage all the system (sw + fabric)
  - 9 PB of disk + 1 10 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

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)

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

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

- OK, we decided to afford this limited piece of work → 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 filesystem metadata are saved on disk along with the data

- Data and metadata are treated simmetrically, 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 filesystem

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 filesystem export to clients (parallel NFS server solution with full HA capabilities developed by IBM)

Stateful 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 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