HPSS – The High Performance Storage System

Storage at the Computer Centre of the IN2P3

HEPiX Spring Meeting 2006

Andrei Moskalenko

Storage team, Centre de Calcul de l' IN2P3.

What's HPSS ?

HPSS is a Highly Scalable Storage System that provides

- hierarchical storage management (**HSM**)
- Quality of Services
- **global name space**, **ACLs**, security (**DCE**, Kerberos, GSS API)
- control and programming interfaces: DMAPI, POSIX API (Extended POSIX API)
- disk and tape **data striping**; **transfer striping** (over multiple TCP connections)
- data replication (double copy, etc.)

Design-wise

- scalable architecture (achieved by adding more storage and control elements)
- network-centric (LAN and SAN).

andrei.moskalenko@cc.in2p3.fr

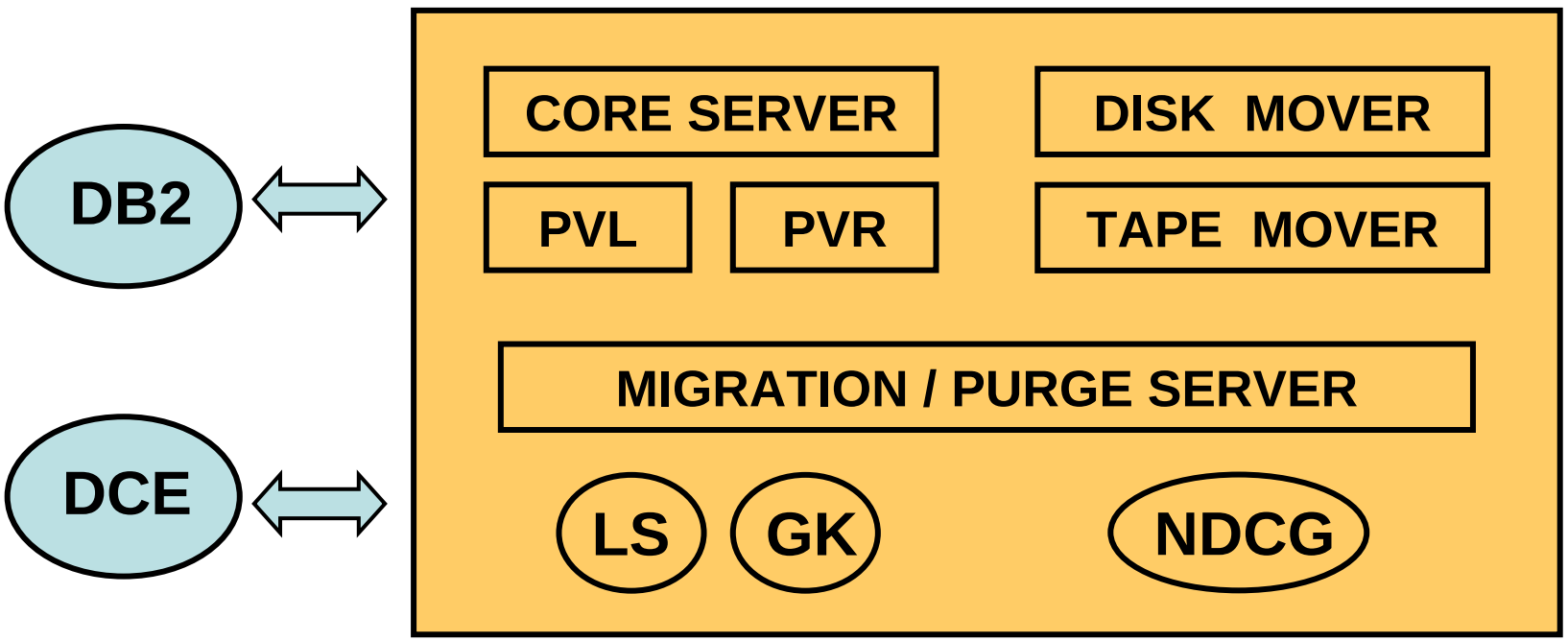
Hierarchical Storage Management provides

- **free disk spaces** through an over allocation mechanism
- **transparency** by hiding complex machinery from users

HSM might also provide

- **Dynamic Resource management** : resources allocated when they needed and where they needed
- **Optimisation/Performance/Administration** (*designed to handle different types of resources)*
- Possibility to integrate **multiple storage tiers**, not just two (HPSS handles 5 levels)

HPSS 5.1 architecture



andrei.moskalenko@cc.in2p3.fr

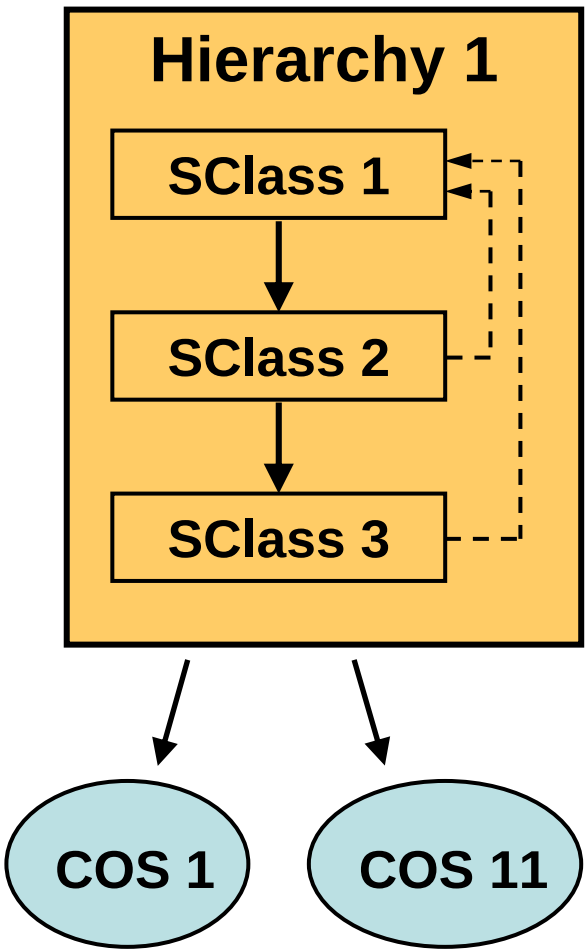
HPSS Configuration

- **Classes of Service, Hierarchies, Storage Classes**

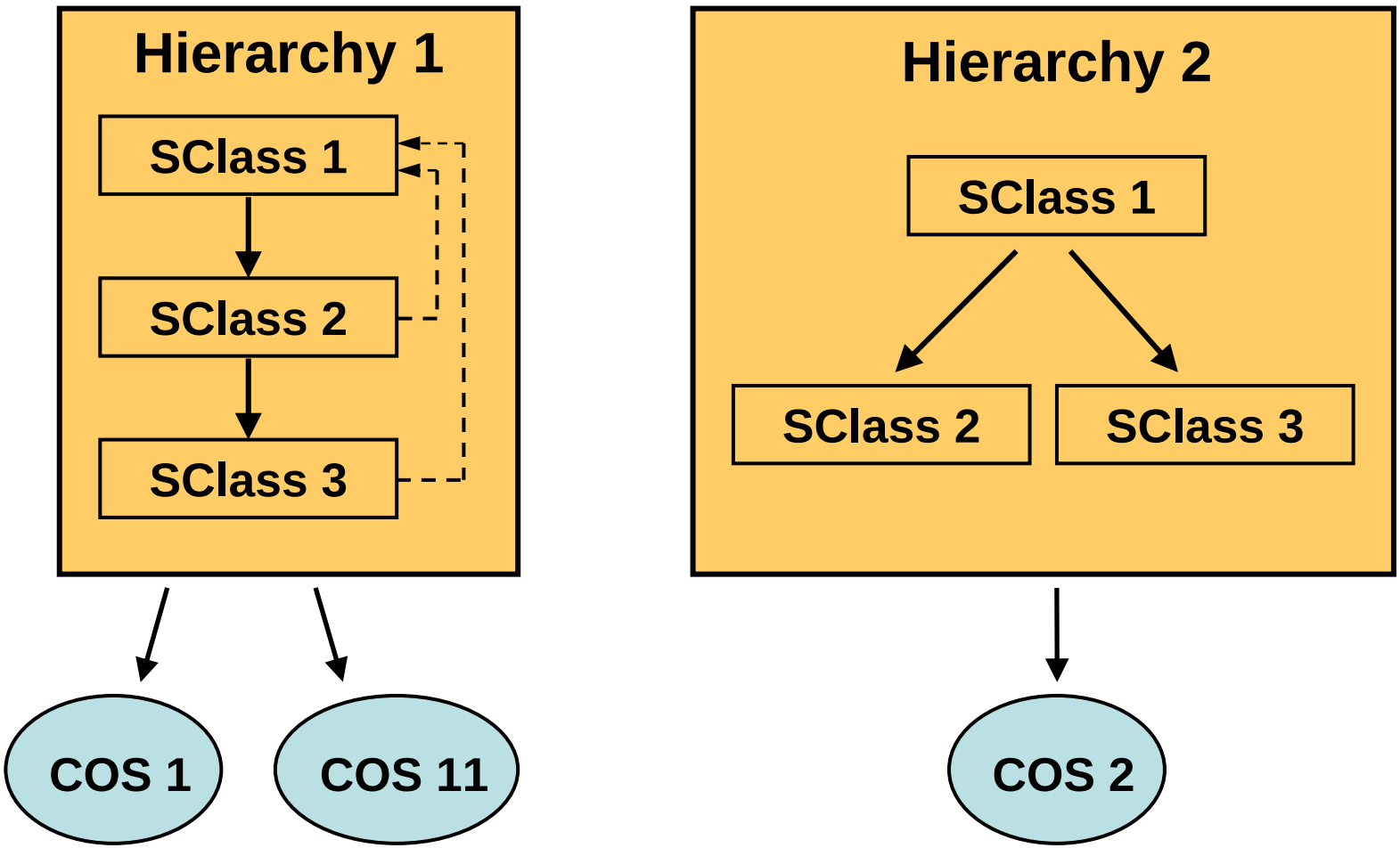
COS 10	for files	< 30 MB
COS 11		< 300 MB
COS 12		< 4 GB
COS 13		< 16 GB
COS 14		< 80 GB

- **Migration and Purge policies** (Dynamically tunable)
- **Name Space and Storage Subsystems** (each subsystem controls its part/parts of the global name space and storage resources)
- **Filesets, Junctions**
- **Families** (assure that logically aggregated data is localised on media to reduce tape mounts)
- **Machinery**: automatic repack, reclaim, etc.

Storage Hierarchies



Storage Hierarchies

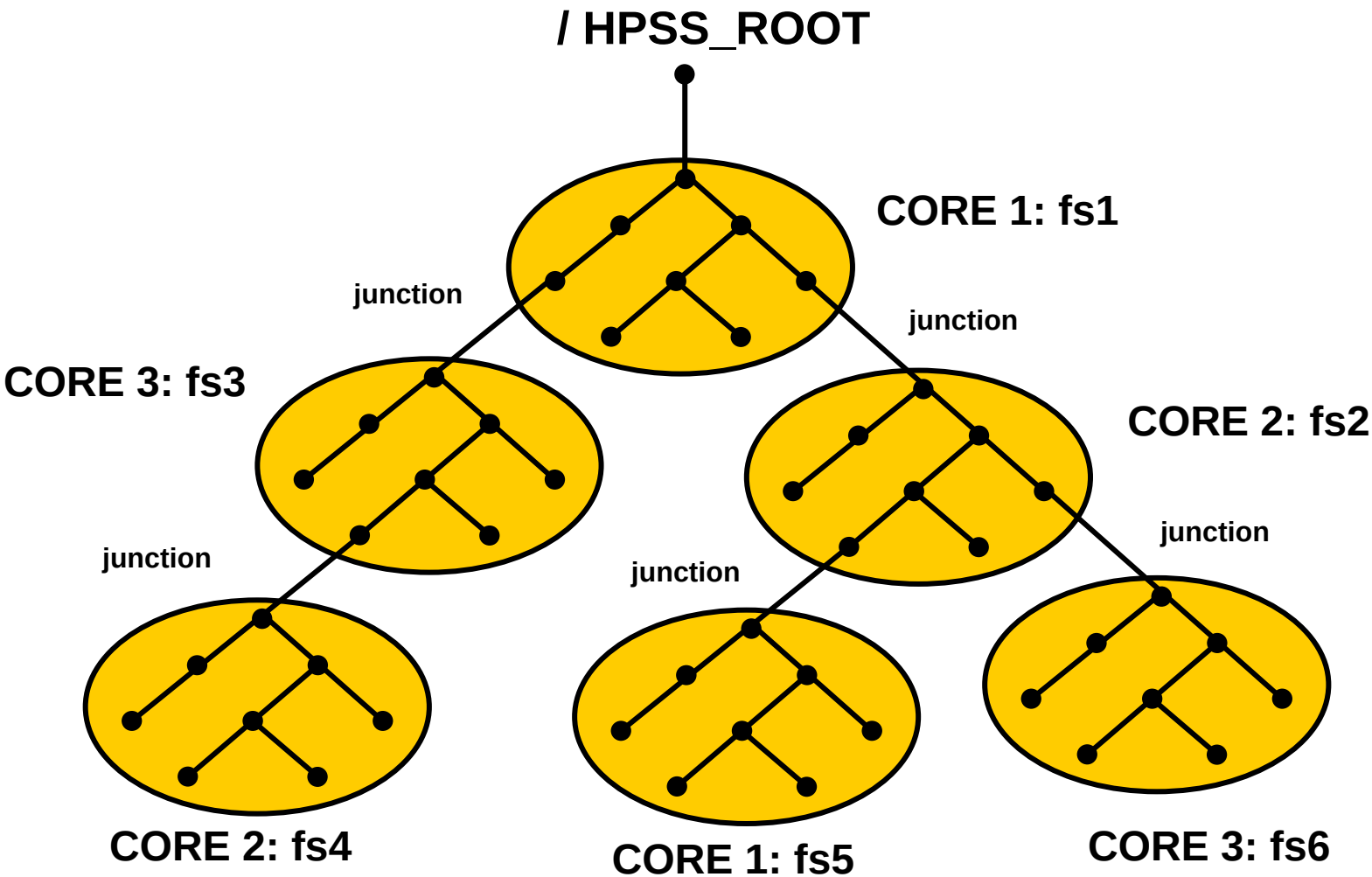


andrei.moskalenko@cc.in2p3.fr

Examples of COS definitions

- Disks only with two tiers of disks
- Tapes only
- Disks + Tapes (two tiers)
- Disks + Tapes (with double copy)
- Disks + Tapes (multiple tiers)

Subsystems and Global Name Space



HPSS IBM web resources

- **HPSS Technology**
  http://www.hpss-collaboration.org/hpss/about/tech.jsp

- **HPSS Brochure**

- Introductory Presentation for the **HPSS Administrators Course**

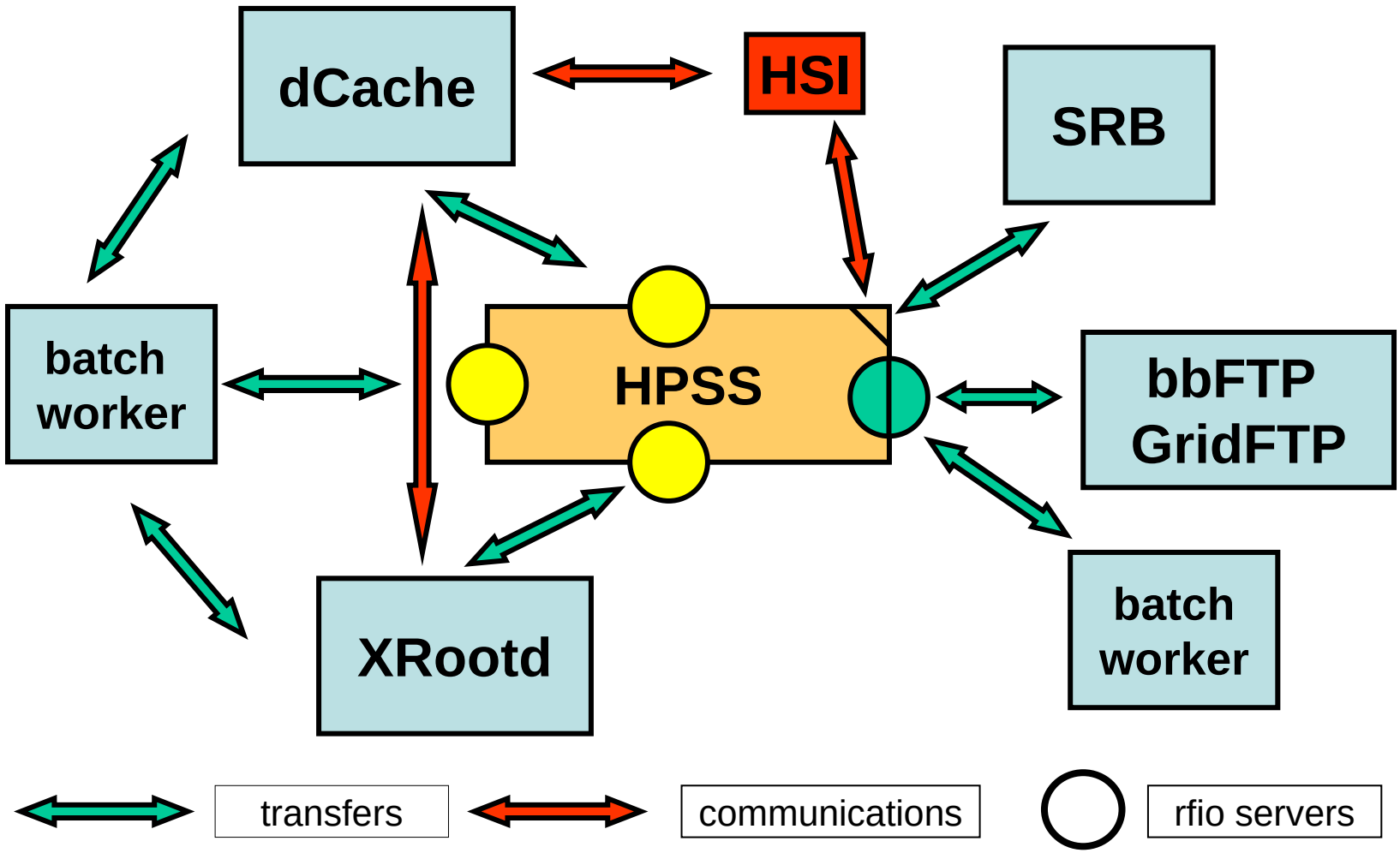
- **High Performance Storage System Scalability**: Architecture Implementation and Experience

- **Storage Area Networks** and the High Performance Storage System

- **High Availability**, etc

HPSS – The High Performance Storage System

MSS and Computing Infrastructure at the CC-IN2P3



andrei.moskalenko@cc.in2p3.fr

at our site:

RFIO

- **RFIO API POSIX** like interface + extensions (**readlist, writelist, setcos**)
- **RFIO commands**: **rfcp** uses the readlist/writelist inteface that allows the data to flow directly from the client to the allocated hpss server (disk or tape mover)

Clients

- **Xrootd, dCache** (via RFIO commands)
- **bbftp, gridftp** (via RFIO API)
- **SRB** (via HPSS-API, NDCG)

andrei.moskalenko@cc.in2p3.fr

HPSS at the CC (IN2P3). Facts.

- **Data Volume doubles annually**: 1.5 **PB** now, 2.5 **PB** at the end of 2006
- **Up to 30TB of data transfers per day**: 100MB/s with rfcp
- **Up to 18500 tape mounts per day**
- **20000 cartridges**; 8000 - 9940B/200GB; 14000 - 9840/20GB
- **3 Subsystems**. More in the future + a cluster for the Core Servers
- 32 disk servers: **36 TB/1100 disks/ 238 movers** (600 movers);
- 28 tape servers: **43 9940B** and **27 9840** drives

andrei.moskalenko@cc.in2p3.fr

Evolution...

- From 1 to 3 **Subsystems** (more in the future)
- Extra **external control** (sophistication of **BQS resource** definition: hpss, u_hpss_cms_xxx, u_rfio_cms_xxx). MSS-BQS autoregulation mechanism?
- **RFIO** connections **watch-dog**
- **Internal control**: fewer disk per software mover/ more movers per node
- Development of **more sophisticated repack** mechanism/policy
- Introduction of a 3rd tier (**Disk** to **Tape** to **Tape**)
- **Tests**: **crash scenarios**, **system limits**, **error messages reproduction/correlations**
- 9840 to 9940B migration

andrei.moskalenko@cc.in2p3.fr

HPSS 5.1 – Daily routine and technical skills

Skills

- **UNIX** + **DCE** + **LAN** + **SAN** (administration, optimisation, etc.)
- **DB2 DBA** (backup, recovery, **optimisation** + optimisation + optimisation)
- **Robotics** (tons of tape mounts per day, incidents, earthquakes, etc.)
- Some knowledge of Java, CAs, etc.
- special: data modelling, batch system resource modelling, etc.

Routine

- **550 Tape/Drive incidents** (prem. EOMs and worse) for 2005
- RFIO log analysis (always something **new**)
- Troubleshooting occasional anomalies due to complexity (rfio bugs, hidden timeouts, etc.)
- User/Client Support + **Supervision** (But admins also need support, doughnuts and coffee…
- **Small files** ……
- **Resources planning and allocation**
- **Repacks**:10500 - 9840s (9840 ->9940B); ~600 9940Bs

andrei.moskalenko@cc.in2p3.fr

Bottlenecks, traps and inefficiencies

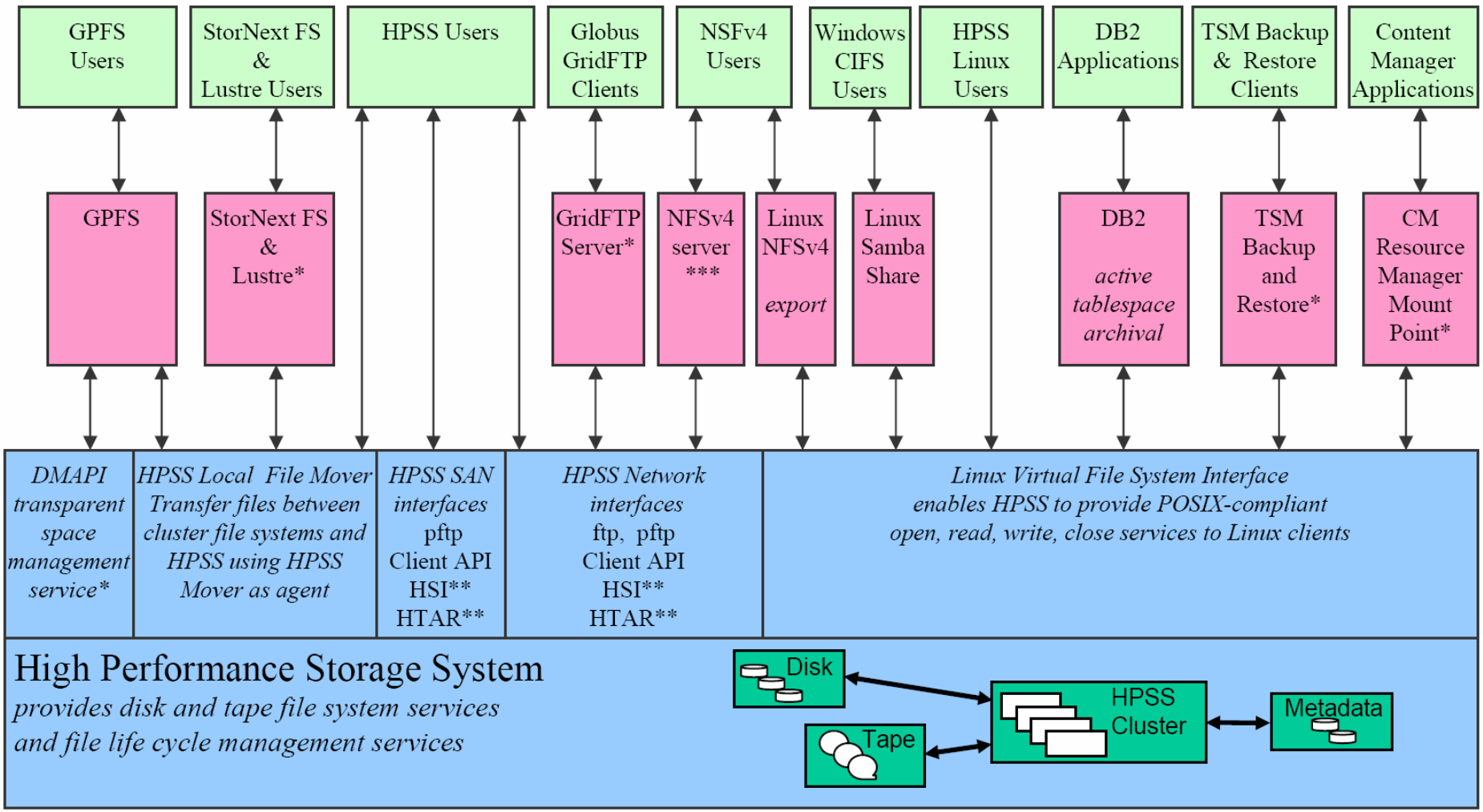
Data **isolation/clustering/aggregation** (ideally on a per user/function/usage/type basis…) is the most crucial task.

- **New data** should not get on the same storage media as **old data** (repacks)
- Easy to write, but difficult to read (but **writes/reads** = **15/85**)
- Repacked Volume ~ Stored Volume (if data badly clustered)
- Repacks and Deletes make DB2 tables **volatile**. Volatile tables with 20 million entries are an optimisation nightmare.
- Badly isolated data => too many mounts
- If we let users choose they tend to make a lot of mistakes (**wrong COS**, **small files**, **MD5 integrity tests** run by users, etc.)
- Weak support for tape errors, insufficient error message reporting, not so many administrative tools, non dynamic configuration...

andrei.moskalenko@cc.in2p3.fr

HPSS – The High Performance Storage System

HPSS Enterprise HSM Services. HPSS 6.2 Client Access Overview



.

A slide from Harry Hulen's presentation of HPSS, hulen@us.ibm.com

Interfacing with HPSS

- **RFIO** at our site
- **HPSS API** = POSIX Client API (CLAPI) extends POSIX API to include COSs, striping, etc. (LINUX VFS will provide access to the HPSS CLAPI)
- Hierarchical Storage Interface. **HSI** can provide information about file locations (ex., tape id and the exact position)
- FTP, Parallel File Transfer Protocol (**PFTP**) Interface
- Data Management API (**DMAPI**) (will be interfaced with Linux XFS,GPFS)
- Linux **NFSv4** and SAMBA
- **GridFTP native** support, etc

at our site:

RFIO

- **RFIO API** POSIX like interface + extensions (**readlist, writelist, setcos**)
- **RFIO commands**: **rfcp** uses the **readlist/writelist** inteface that allows the data to flow directly from the client to the allocated hpss server (disk or tape mover)

- **Xrootd**, **dCache** (via **RFIO commands**)
- **SRB** (via **HPSS-API**, NDCG)
- **bbftp**, **gridftp** (via **RFIO API**)

andrei.moskalenko@cc.in2p3.fr

Conclusions

HPSS is an excelent HMS system that

- provides highly scalable storage and archival services (**SU**s)
- provides (SAN-centred) **global file system functionalities**
- is capable to host **10s** of **PB** of data (100s PB?)
- provides support for scalable, parallel I/O operations
- scales to **10s** of **TB** daily throughput (100s TB)
- does not impose any unreasonable restriction on your storage models
- is highly **modular** (new technology, evolution)
- is very **robust** (data replication, HA)

andrei.moskalenko@cc.in2p3.fr

Conclusions

HPSS is an excellent HMS system that

- provides highly scalable storage and archival services (SUs)
- provides (SAN-centred) global file system functionalities
- is capable to host 10s and 100s PB of data
- provides support for scalable, parallel I/O operations
- scales to 10s and 100s TB daily throughput
- does not impose any unreasonable restriction on your storage models
- is highly modular (new technology, evolution)
- is very robust (data replication, HA)

What one would possibly like to see is

- more sophisticated **migration/purge** policies (**dCache-like**)
- tools/utilities for data migration (exploiting **meta-data**, not just copy)
- more sophisticated inter-subsystem name space partitioning
- take advantage of all **SAN functionalities** as soon as possible
- a better incident and error message control

andrei.moskalenko@cc.in2p3.fr