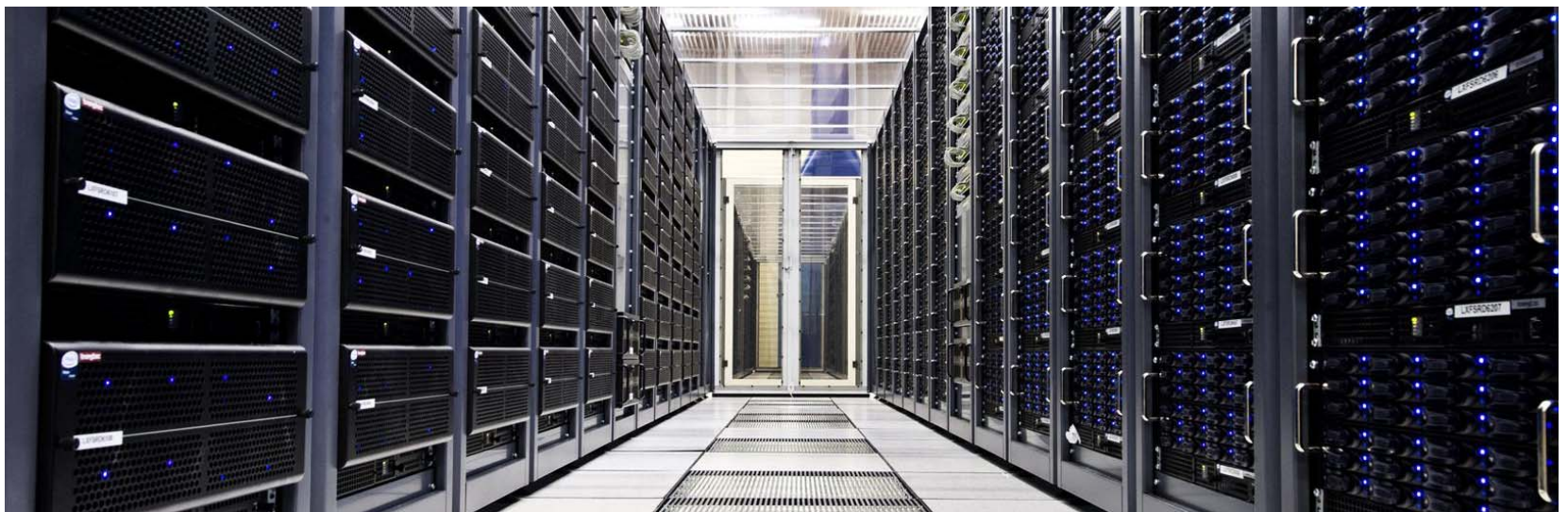


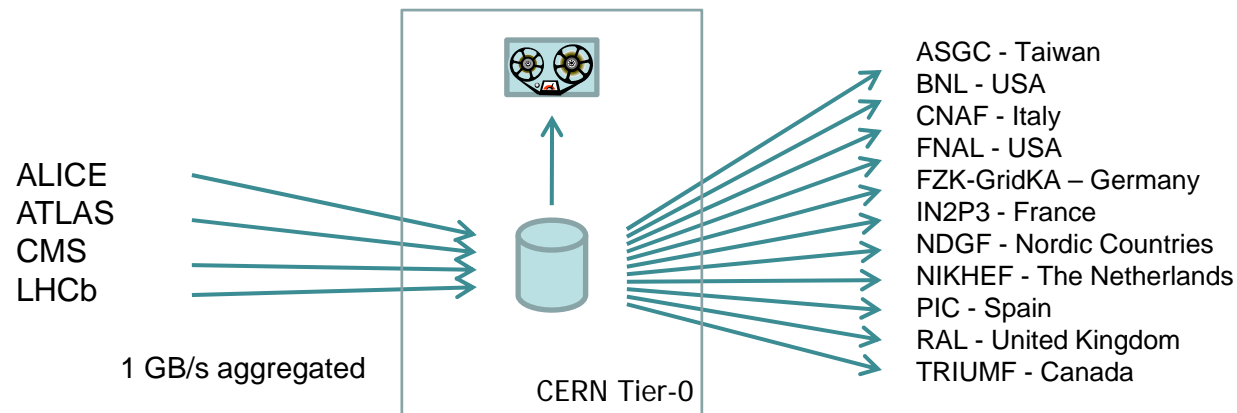
Data Management Evolution and Strategy at CERN

G. Cancio, D. Duellmann, A. Pace

With input from several IT-DM developers



- For all Tier-0 data operations, CERN is using CASTOR for
 - Migration of LHC data to tape
 - Replication of the LHC data to Tier-1 sites
- CCRC'08 Data Challenges have validated this architecture which is now in production



- ... but few improvements are necessary
 - to support analysis at CERN
 - to reduce operational effort

- During 2008-09, several improvements were made to CASTOR in:
 - Monitoring
 - Security
 - SRM
 - in the tape handling area
 - in reducing latency to access files on disk and improved concurrent access

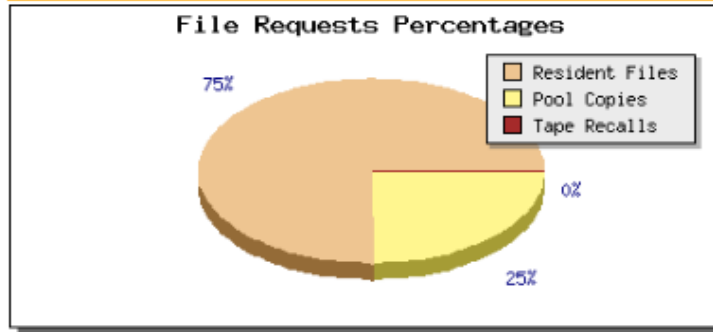
- New Castor Monitoring
 - Key performance calculated in real time (for end-users and operations)
 - New indicators: disk hit/miss, garbage collection statistics, inter-pool transaction statistics, ...
 - Immediate detection of performance issues, alarms triggering
 - Not specific to CERN monitoring infrastructure
- Web dashboard with real time alarms and drill-down options for incident diagnostics
- See poster from Pokorski, Rekatsinas, Waldron, Duellmann, Ponce, Rabaçal, Wojcieszuk
<http://indico.cern.ch/materialDisplay.py?contribId=118&sessionId=62&materialId=0&confId=35523>

CASTORATLAS: STATUS MONITOR

HOME Statistics LINKS

Requests Monitor

Requests Percentages(Total Instance)



Requests Counters per SvcClass

SvcClass	DiskHits (%)	DiskCopies (%)	TapeRecalls (%)
atical	17 (1)	0 (0)	0 (0)
atldata	18 (1)	0 (0)	0 (0)
atlprod	30 (1)	0 (0)	0 (0)
default	396 (0.72)	152 (0.28)	0 (0)
t0atlas	41 (1)	0 (0)	0 (0)
t0merge	109 (1)	0 (0)	0 (0)

Migration Monitor

Migration Counters per SvcClass

SvcClass	Files Migrated
atlasuserdisk	10
atlprod	114
atldata	24
atlt3	1
t0merge	148
t0atlas	22
default	1

Pool Transactions Monitor

External/Internal DiskCopy Counters

FROM - TO	atlasgroupdisk	default	atldata	atlprod	t0atlas
atlasgroupdisk	0	8	0	0	0
default	0	0	0	0	0
atldata	0	20	0	0	0
atlprod	0	244	0	0	0
t0atlas	0	3	0	0	0

GC Monitor

GC Files	Average Age	Average Size
112	↑ 625.55 (0.3504)	↑ 5668.648 (0.977)

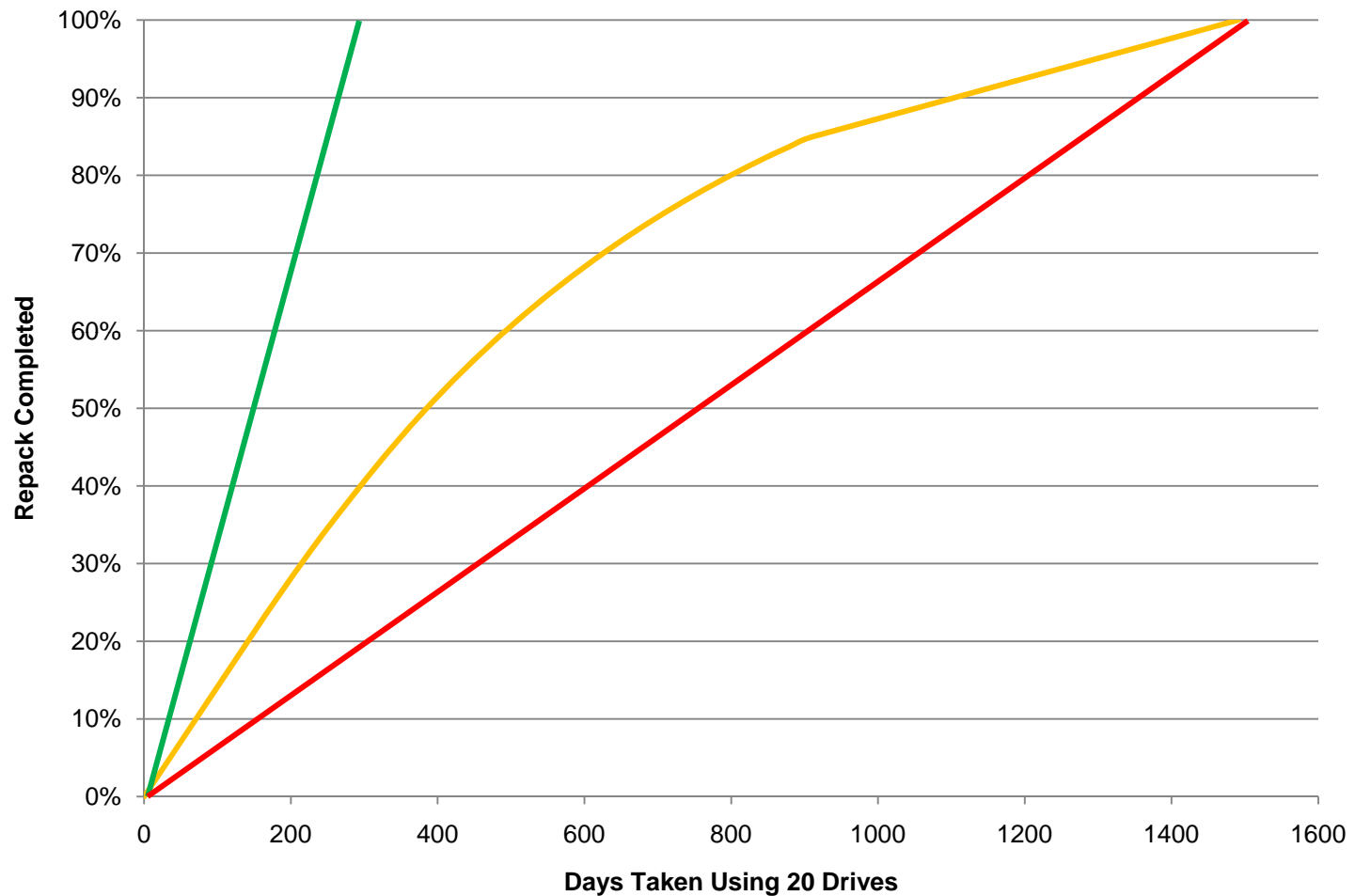
- Strong authentication now available (v2.1.8)
 - No problems identified for secure access using XROOT protocol
 - GSI/PKI (Grid certificates) and Kerberos authentication implemented in RFIO, Name Server and Request Handler
- However for RFIO, the Name Server and the Request Handler are stateless protocols
 - GSI/PKI authentication at every request **does not scale to request-intensive physics analysis activity**
 - Does not make sense to add server power to compensate this inefficiency. Currently unusable
 - The Castor server supporting Kerberos must run SL5 to have a multithreaded server (**otherwise requests will be serialized**)
 - Currently SL4 servers deployed, therefore unusable
 - Upgrade to SL5 server necessary
 - Even with SL5, the Kerberos replay cache will need to be disabled to avoid request serialization
- Options to enforce security:
 - Limit RFIO and direct Name Server access to limited trusted production cases (Analysis made with XROOT only)
 - Implement the possibility that an initial GSI/PKI authentication is used to obtain a Kerberos ticket in RFIO. Account needed at CERN, and lot of developments
- See <https://twiki.cern.ch/twiki/bin/view/DataManagement/CastorSecurity>

- SRM interface improved
 - Collaboration with RAL
 - Required robustness being reached
- Future plans
 - to improve logging, including tracing of requests through their lifetime, time-ordered processing
 - Implementing the “server busy” protocol improvement, but waiting for new clients deployment
 - Better integrate SRM database with stager database

- In production with 2.1.8:
 - New tape queue management supporting recall / migration policies, access control lists, and user / group based priorities
- Planning a new tape format with data aggregation
 - Increase current write efficiency which today drops on small files
 - allows managed tape migration / recall to match native drive speed
 - no more performance drop when handling large number of **aggregated** small files

- See Poster from Bessone, Cancio Melia, Murray, Taurelli

<http://indico.cern.ch/materialDisplay.py?contribId=118&sessionId=62&materialId=1&confId=35523>

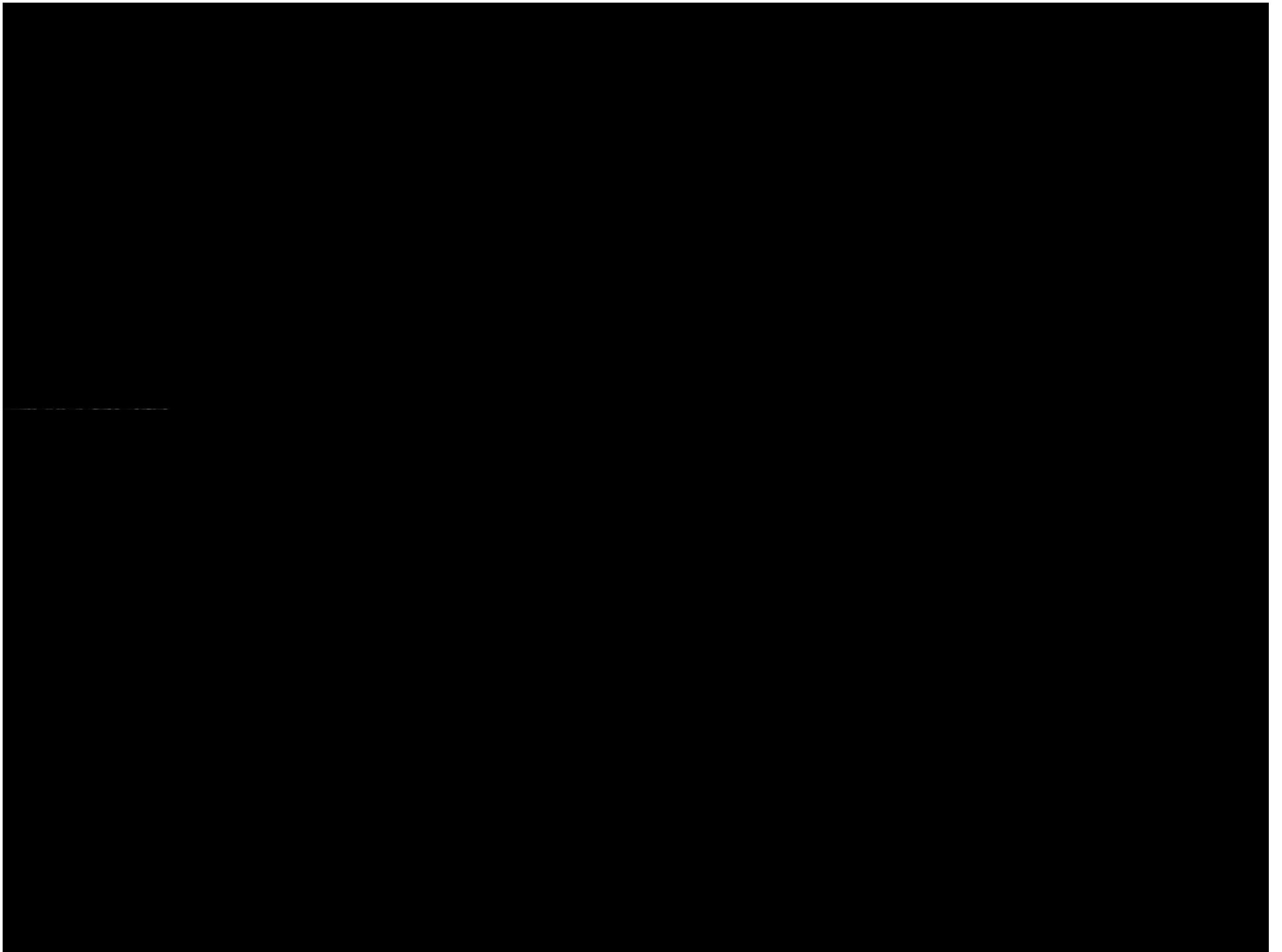


- Not a problem for the Tier-0 operation
 - Scheduled I/O guarantees Tier-0 data streaming to tape and to Tier-1 which are not affected by the high LSF latency
- ... but a **concern** for interactive data access and analysis
 - Unacceptable **to wait seconds** to open a file for read when it is already on disk

- Removal the LSF job scheduling
 - Only for XROOT clients reading from disk files. File write and tape access is still scheduled as well as all RFIO requests.
 - Latency reduces from seconds to milliseconds
 - Plan to remove job scheduling also for XROOT write operations in the future release of Castor
- Name Server performances improvements
 - Planned for the next Castor release
 - No more multiple network roundtrips to resolve names
 - Additional database optimization
 - “stat” command to obtain file information performance improved by factor of 20
 - From 400 – 500 requests / sec to more than 10K requests / sec
 - Further improvements possible by using the XROOTD cache (but limited to XROOT clients)

- For disk-based analysis, the XROOT protocol becomes strategic to benefit from the increased performances
 - Excellent integration with ROOT for physics analysis
 - Connection-based, efficient protocol for direct access to storage on disk pools. Strong authentication built in
 - Can leverage easily functionalities of the underlining file systems
 - Security, Authorization (ACLs), Quotas, Audit and logging
- Opens several possibilities
 - Mountable file system
 - Usable with FUSE in interactive mode, global namespace
 - Additional access protocols can be added (eg. NFS 4.1)
- Separation of access protocols from storage technology
 - Access protocols: RFIO, XROOT, MFS, POSIX, NFS 4.1, ...
 - Storage technology: Castor File Servers, Lustre, Xrootd / Scalla, GPFS, ...

- Data Management at CERN is ready for the LHC startup
- Fully committed to support all major Grid / experiment interfaces
 - GridFTP, RFIO, SRM, XROOT, ...
- Major effort in improving efficiency, stability and operational cost
 - Tape, SRM, Monitoring, ...
- But it is impossible to overcome intrinsic limitations of some technologies
 - Tapes will never support random access
 - Chaotic analysis requires an efficient, connection-based, secure access protocol



Data
Management

for(tp = m, tpre

Questions / Discussion

CERN IT
Department



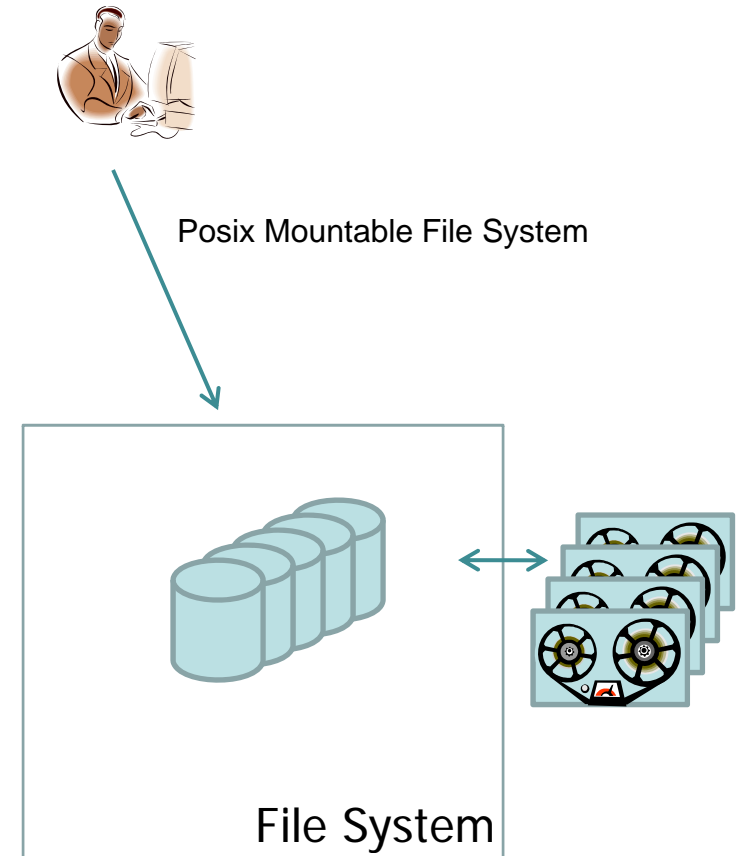
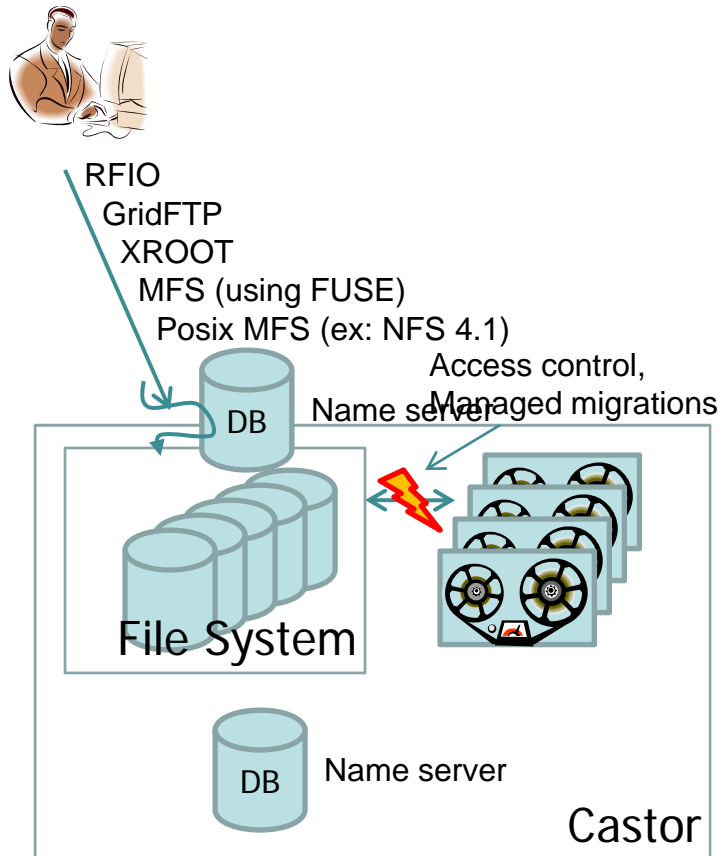
- Role of file systems and mounted file systems for physics analysis
- Role of data “Repack” process
- Role of tapes
- Hardware and Storage reliability

- Working groups on analysis requirements led by B. Panzer (Jul – Oct '08) for Tier-0, by M. Schulz (Nov'08 -) for Tiers-1/2
- (simplified) requirements collected so far
 - **Analysis made on disk pools only**
 - Tape access limited to managed and aggregated recall / migration requests from the experiments
 - No end-user recalls or direct access to tapes
 - **Demand for direct file access**
 - No unnecessary latencies
 - Mountable file system accessible from both interactive desktops and batch systems
 - Integration with the existing physics analysis tools (grid middleware and root)
 - Secure / manageable (Authentication, ACLs, Quota, Accounting, etc ...)
 - **File catalogue and name server consistency**

- File system advantages
 - High performance file system access
 - Mainstream software and services
 - No end-client software necessary (posix) for basic file operations
- File system drawbacks
 - File catalogue and name server consistency
 - Tools for data management need to be re-thoughts
 - Long-term dependencies if advanced (specific) file operations features are exposed to end users

- Areas of research
 - Integration with existing infrastructure
 - Wide Area Transfer / Global namespace
 - Data replication (replicated pools / hot spots)
 - Tape integration
 - integration with the file catalogue
 - SRM interface (based on StoRM ?, BeStMan ?)
- Realistic timeline for a complete solution
 - End 2010 for the file system
- Mountable file system is considered a strategic direction
 - Either with Castor or with a new file system

- Direct file system access to Castor files with similar performance in term of requests/sec and latency
 - Name server lookup generates less than 10% performance overhead
- Standard grid access protocol supported, compatibility ensured
 - SRM, RFIO, GridFTP (available now)
 - Mountable file system possible using Fuse (available now)
 - NFS 4.1 (as a future option)
- Name server / file catalogue consistency ensured
 - Global namespace
 - Tape integration, with same logical filename across disk pools and tapes
 - data management possible
- Separation between access protocols and underlying storage technologies
 - No long-term dependencies:



- Technology independent data protocol
- Centrally managed data solution

Storage technology change transparent (Lustre / GPFS / ...)
 Internal architecture change transparent (Tape / Disk / Flash ...)
 End-to-end consistency (eg checksums)

- Higher performance
- Data managed by the experiments

Namespace on tape differs from namespace on disk
 Two files with same name can have different content
 Two files with different name can have same content
 IT-organized data migrations more heavy

- Traditionally, the “Repack” process has been run whenever there was a “media change”
 - Triggered by hardware changes. Otherwise data were only read back on (user) requests
 - Repack could be run with direct tape-to-tape copy
- In the CERN recent repack exercise, repack has been “the” tool to discover several issues unrelated to the tape technology itself
 - Software errors appearing only under heavy loads in recalling data
 - Errors in tape data format that dated back several years that were not noticed until the tape was read back
- Repack is a measure of our ability to read data and must be constantly exercised
 - Plan to run repack constantly, independently of hardware changes
 - Hardware changes happen asynchronously from repack process that populates new hardware as it becomes available
 - Repack exercise applied to all storage types, disk in particular
 - Ideally, no data should be left on a media for more than one year. Ideally, 3 months
 - Software issues, (slow) data corruption are detected in time

- Tapes are a “forever dying” technology
 - This means that we will have tapes for the next 10 – 15 years – For the whole lifetime of the LHC project
- Tapes are more reliable than disks
 - True, but tapes also fail
 - recovery procedures must be foreseen for both tapes and disks
- Tapes are cheaper than disk
 - media itself becomes more expensive than disk
 - Tapes remains cheaper if the reader infrastructure is shared among a large number of tapes
- Tapes do not consume electrical power
 - Also disk do not consume power when powered off

- Why not using the same tape architecture with tapes ?
- “Drive” to “Tape” ratio 1:300, can read only one tape at the time
- Why not an architecture of 1 server for 300 disks ?
 - Faster to power on one disk than mount and seek a file on tape
 - 1 server can read 20 – 30 disks simultaneously

- Traditionally, reliability is provided by the hardware
- Hardware can fail, so we duplicate the hardware
 - Double the cost to compensate failing hardware
- Some ideas
 - Implement “arbitrary reliability” in software
 - Data redundancy can be dynamically reconfigured to compensate measured hardware failure rate
 - Have an infrastructure resilient to hardware failures
 - Requires separation of CPUs from storage.
 - SAN, iSCSI, ...

- Every file is split in N chunks, which are encoded with redundant information so that only $M < N$ chunks are necessary to reconstruct the original data
 - Similar to Raid-6 with Reed–Solomon error correction
- All chunks are saved on different media
- You can lose $N-M$ devices at any time without losing data.
- $N-M$ is arbitrary
- An ongoing “repack” process allows to re-tune the parameters to match the measured hardware reliability

