Overview of Data Storage, Access and Distribution Technologies

Dirk Duellmann, CERN IT

2nd Aspera Workshop

30-31 May 2011

Barcelona, Spain

Outline

- Components for File storage & distribution in the Grid
- Important Model Parameters
- Lessons learned at LHC
- Which model changes and new technologies are relevant?

CASTOR
CERN Advanced STORage manager

dCache.ORG

fts
Grid File Transfer Service

StoRM
Storage Resource Manager

BeStMan
Berkeley SRM

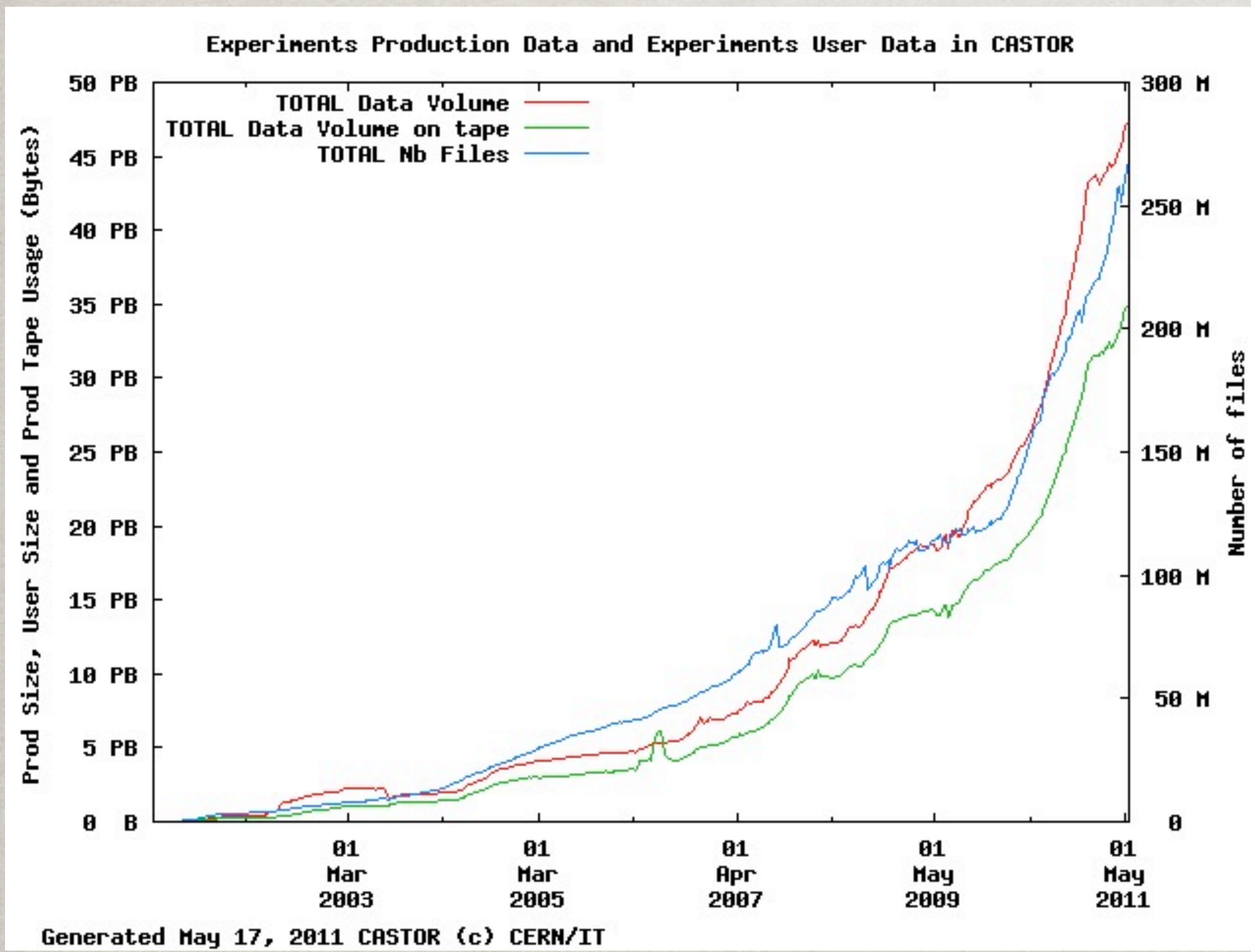
Scalla
Scalla

IBM

ORACLE

lustre

CERN - Storage Evolution

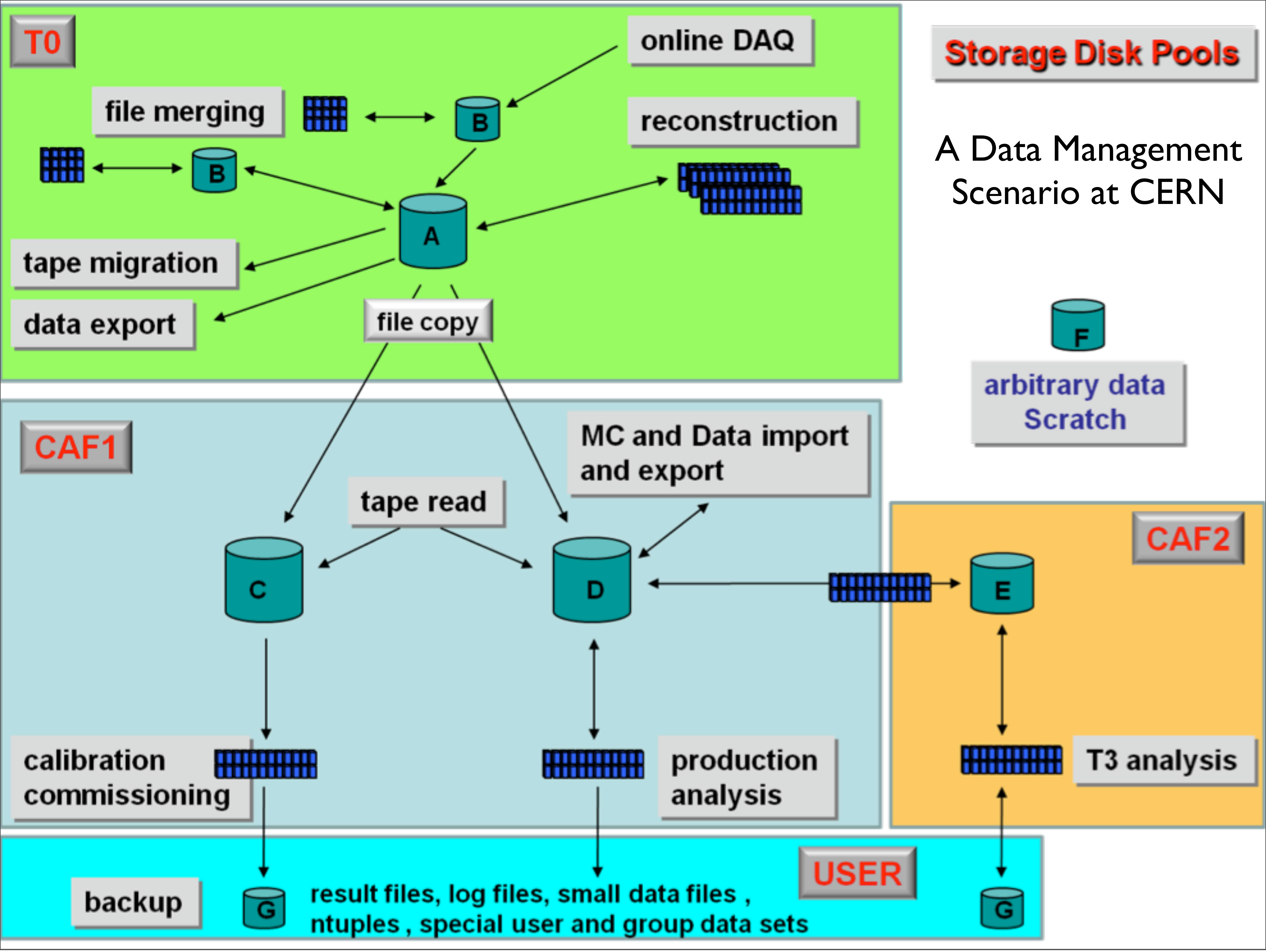


more than 20 PB stored in 2010

6 GB/s sustained (220 TB/day)

~1500 disk servers

IBM + Oracle tape libraries


T0
online DAQ
file merging
B
B
reconstruction
A
tape migration
data export
file copy
Storage Disk Pools
A Data Management Scenario at CERN
F
arbitrary data Scratch
CAF1
MC and Data import and export
tape read
C
D
CAF2
E
calibration commissioning
production analysis
T3 analysis
USER
backup
G
result files, log files, small data files , ntuples , special user and group data sets
G

Tools for File Management

- many different use cases and environments
- several storage element (SE) implementations have been produced
  - evolutionary rather than following an upfront design
  - use cases keep evolving and products extending
- Now consolidation is required to keep a healthy balance

Tools for File Management

- many different use cases and environments
- several storage element (SE) implementations have been produced
  - evolutionary rather than following an upfront design
  - use cases keep evolving and products extending
- Now consolidation is required to keep a healthy balance

Tools for File Management

- many different use cases and environments
- several storage element (SE) implementations have been produced
  - evolutionary rather than following an upfront design
  - use cases keep evolving and products extending
- Now consolidation is required to keep a healthy balance

File Management Components

<table>
<tr><th colspan="2">Disk</th><th colspan="2">Tape</th><th>Distribution</th></tr>
<tr><td colspan="2">High Level Storage Admin</td><td colspan="2"></td><td></td></tr>
<tr><td>Posix I/O</td><td rowspan="5">Clustered Filesystem</td><td>Aggregation & Clustering</td><td rowspan="5">Mass Storage System</td><td>Transfer Workflow</td></tr>
<tr><td>Authorisation / Authentication</td><td>Tape Scheduling</td><td>Logical Connections</td></tr>
<tr><td>Name Space</td><td>Media Migration</td><td>Error Handling / Retry</td></tr>
<tr><td>I/O Scheduling / Placement</td><td>Volume Management</td><td>Bandwidth Reservation</td></tr>
<tr><td>Disk Pools</td><td>Tape Libraries</td><td></td></tr>
</table>

Data Management For Data Productions

- Focus for many years: Data Production
  - organised access, large files, few heavy sequential accesses
    - optimising h/w setup for particular work flow pays off
      - eg dedicated disk pools to guarantee predictable storage behaviour
  - Key model parameters: volume & media cost
    - simple relationship between storage volume and I/O operations per second can be established
  - but need comprehensive monitoring and regular re-evaluation
    - hard drive volume to spindle ratio is shifting
    - relative priority / frequency of work flows is changing

Analysis Impact on Data Management

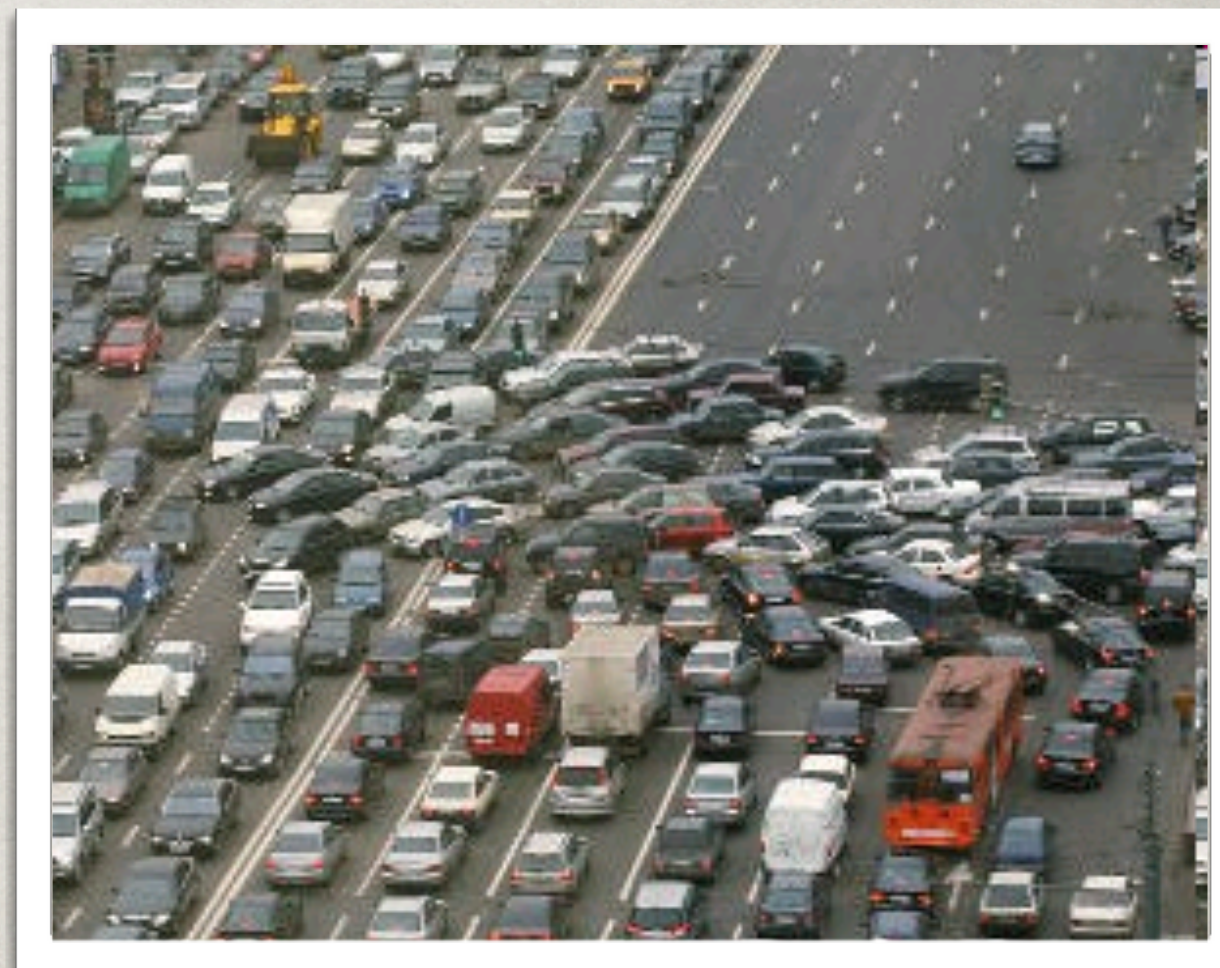
- Analysis Properties
  - many users, many (smaller) disk files, many opens and random reads
  - tuning on individual tasks is not feasible (due to larger number of them)
- Key parameters
  - File meta data access and IO/sec are more important than pure storage volume and can vary significantly for different tasks
- Additional Focus on
  - Manageability
    - accounting & quota per user/group
  - Performance
    - concurrent low latency access from many users to many files
    - computing model should provide estimates which can be compared agains measured performance - iterative process
  - Usability
    - many inexperienced users with primary interest in physics - not computing
    - preference for simple (mounted) file system view

(Too?) Many Protocols

- Focus on two areas
  - remote data access
  - storage management
- Key metrics
  - scalability
    - use of server resources
    - round trips / latency
  - protocol clients
    - kernel / user space
    - standards / HEP specific
  - long term maintainability
    - do we need control or trust other s/w providers?

(Too?) Many Protocols

- Focus on two areas
  - remote data access
  - storage management
- Key metrics
  - scalability
    - use of server resources
    - round trips / latency
  - protocol clients
    - kernel / user space
    - standards / HEP specific
  - long term maintainability
    - do we need control or trust other s/w providers?



(Too?) Many Protocols

- Focus on two areas
  - remote data access
  - storage management
- Key metrics
  - scalability
    - use of server resources
    - round trips / latency
  - protocol clients
    - kernel / user space
    - standards / HEP specific
  - long term maintainability
    - do we need control or trust other s/w providers?

(Too?) Many Protocols

- Focus on two areas
  - remote data access
  - storage management
- Key metrics
  - scalability
    - use of server resources
    - round trips / latency
  - protocol clients
    - kernel / user space
    - standards / HEP specific
  - long term maintainability
    - do we need control or trust other s/w providers?

(Too?) Many Protocols

- Focus on two areas
  - remote data access
  - storage management
- Key metrics
  - scalability
    - use of server resources
    - round trips / latency
  - protocol clients
    - kernel / user space
    - standards / HEP specific
  - long term maintainability
    - do we need control or trust other s/w providers?

LAN Access Protocols & Server Implementations

	Server scaling	Fail-over / redirect	Client available	Comments
	remote access API / user space mount (eg FUSE)			
RFIO	O(10-100) clients	no	even two	GPL/CERN - being phased out
XROOT	O(1000) clients	yes	via ROOT	BSD/xroot consortium (SLAC, CERN, Duke Univ.)
	direct mount / kernel module			
Lustre	O(100-1000) clients	yes	End of 2009?	GPL/SUN -> Oracle ->?? file system implementation used by SE
NFS 4.1	prototype by dCache	yes	with RedHat 6	protocol defined in RFC 3530, one server implementation per storage backend

Filesystems - two Functional Roles

- 1) the “client protocol” used to access data
  - Should provide
    - support secure authentication (incl. X509, Kerberos)
    - client side data cache, support for vector reads
    - redirect clients in case one access path is (temporary) unavailable
  - Examples: NFS4.1, XROOT/FUSE, AFS, {GPFS}

- 2) the software used to access/manage cluster storage
  - Should provide
    - high performance namespace, quota system
    - scalability in aggregate performance (eg file replication, striping)
    - support for online storage re-organisation
    - storage availability through media redundancy
  - Examples: GPFS, Lustre, AFS, XROOT
- **For the moment: no system can claim to implement both functional areas**
  - **but clustering storage is an attractive starting point for several T1 sites**

Protocols & Grid Security

- Grid Certificates and CPU
  - The LCG Grid uses decentralised identity system based on X.509 proxy certificates with role annotations
- Naive certificate evaluation for each request is often too CPU intensive
  - few tens of authentications can saturate a core
  - applies to file, database, catalogue and SRM requests
- Session concept (as eg in xroot) can help to significantly reduce the security overhead
  - Agreement on use of X509 underway between main stake-holders providing xroot access

Scalable File Namespaces

- Frequent operations:
  - obtain file meta data (stat), get directory content (ls)
    - but also: which files are hosted on machine / disk XYZ
- Name space is traditionally kept in a database
  - number of round-trips often limits the name space performance of larger storage systems
  - coalesce requests & cache results close to the client
    - inside the disk layer or in front of database
- Active name space today fits into main memory
  - New EOS development at CERN is based on in memory namespace with very significant performance gains
  - DB role changes from an efficient access layer for large volume data to a recoverable store

Optimising I/O System Efficiency

- End-to-end performance review of the full s/w stack
  - Experiments: data model & integration with persistency s/w
  - Application Area: ROOT use of storage access protocols (significant gains even after 10y)
  - Storage providers: resulting meta data and data rates
  - Sites: CPU<->storage connectivity balance
- This review is not a task for end-users!
  - Need to instrument code and services with appropriate monitoring and build up working groups with user and site involvement to analyse results

Storage Request Manager

- SRM is a complex standard with many stake-holders
  - Goal: isolate users from implementation details of a particular storage element
  - Only a subset implemented by WLCG SEs
    - Approach seems different from other standards
      - eg SQL: extend a consistent core provided by all
- Is the implemented subset still consistent/used?
- Is the effort for the SRM abstraction smaller than a direct integration with storage elements?

File Catalogs

- Exists within each storage element (local name space)
  - and globally at experiment level
  - in some cases on the level of datasets (complete set of files)
- Issues
  - reliable synchronisation between different name space providers
  - related: temporarily or permanently unavailable files
- Current practice
  - comparing dumps of all files in an SE with experiment catalog
    - neither scalable nor consistent
- Message based synchronisation scheme under development

File Set Support

- Current storage systems provide a convenient filename space to experiments
- but do not really aid several of their main work-flow primitives
  - change disk/tape state for a complete set of files
  - check if a file set is complete on-disk/on-tape/at-a-site
- from the service perspective
  - file-set knowledge would help in more efficient dataset placement on disk & tape
  - garbage collection on disk
- File set concept would allow for more efficient support of production workflows

Hierarchical Storage Management

- Hierarchical Storage Management (HSM) systems promise to hide the storage hierarchy from users.
  - simple file level (posix) interface
  - system manages/optimises movement between tape and disk.
- Is the HSM model still used / useful?
  - Production
    - Experiment work-flow system have to insure (pre-stage) dataset on disk
    - Disk-only pools play an ever increasing role
  - Analysis - also here HSM seems of limited utility
    - input data must be on-disk, volume is managed by physics WGs
    - most users don't have access to tape
- Over the last years we have largely given up on using HSM
  - we just use automatic archiving of new data
- A direct access to disk cache and a decoupled archive with transfers managed by an experiment work-flow system re-gained transparency.

Tape Media Repack

- CERN: every 2-3 years tapes are copied to new format, drives or media
  - economy: recycle existing media at higher density
  - spot potential media or s/w problems
- Significant effort
  - h/w investment (dedicated drives)
  - s/w development & deployment
- Review gain/effort with statistics from current repack round

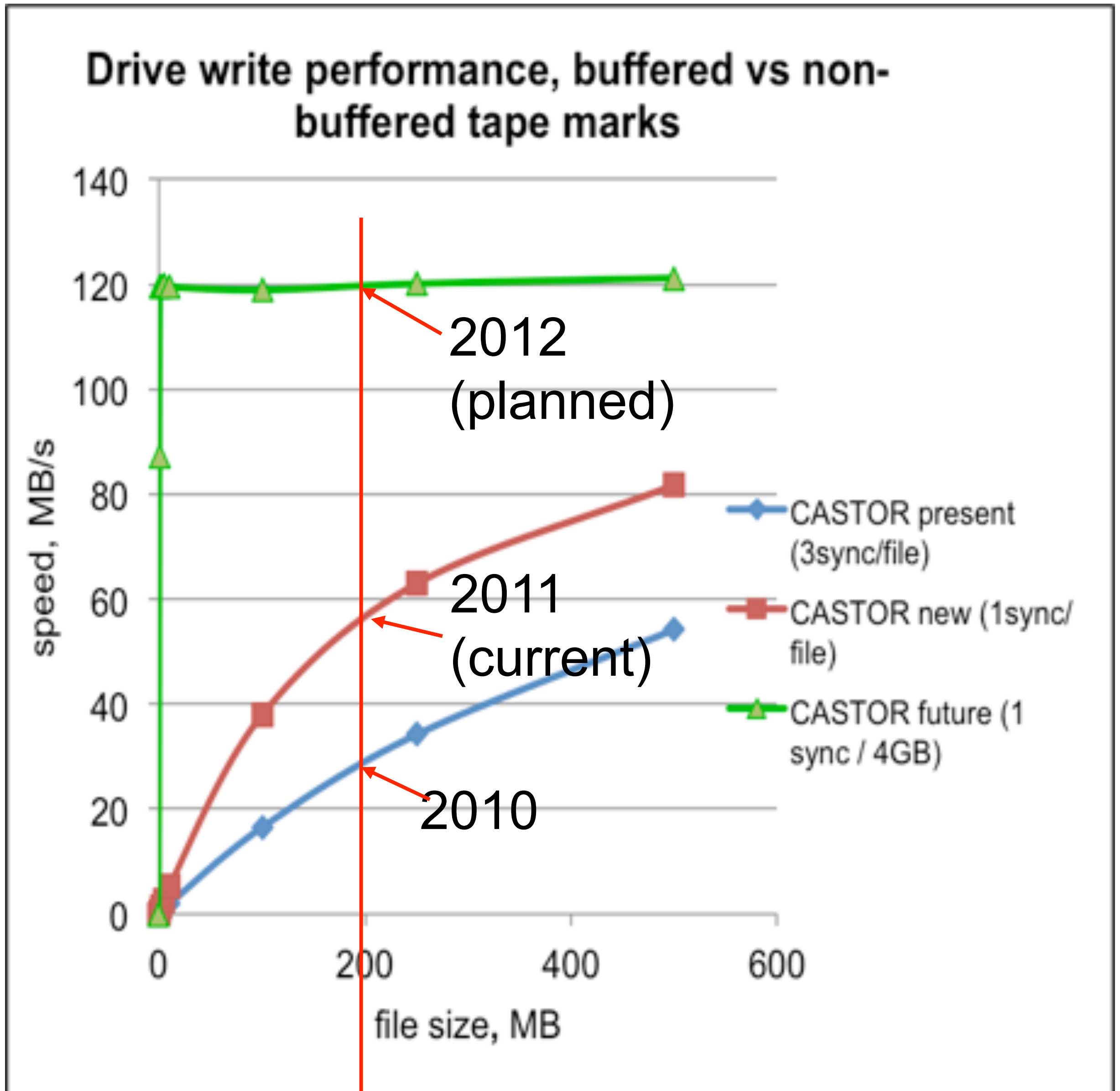
Tape Media Repack

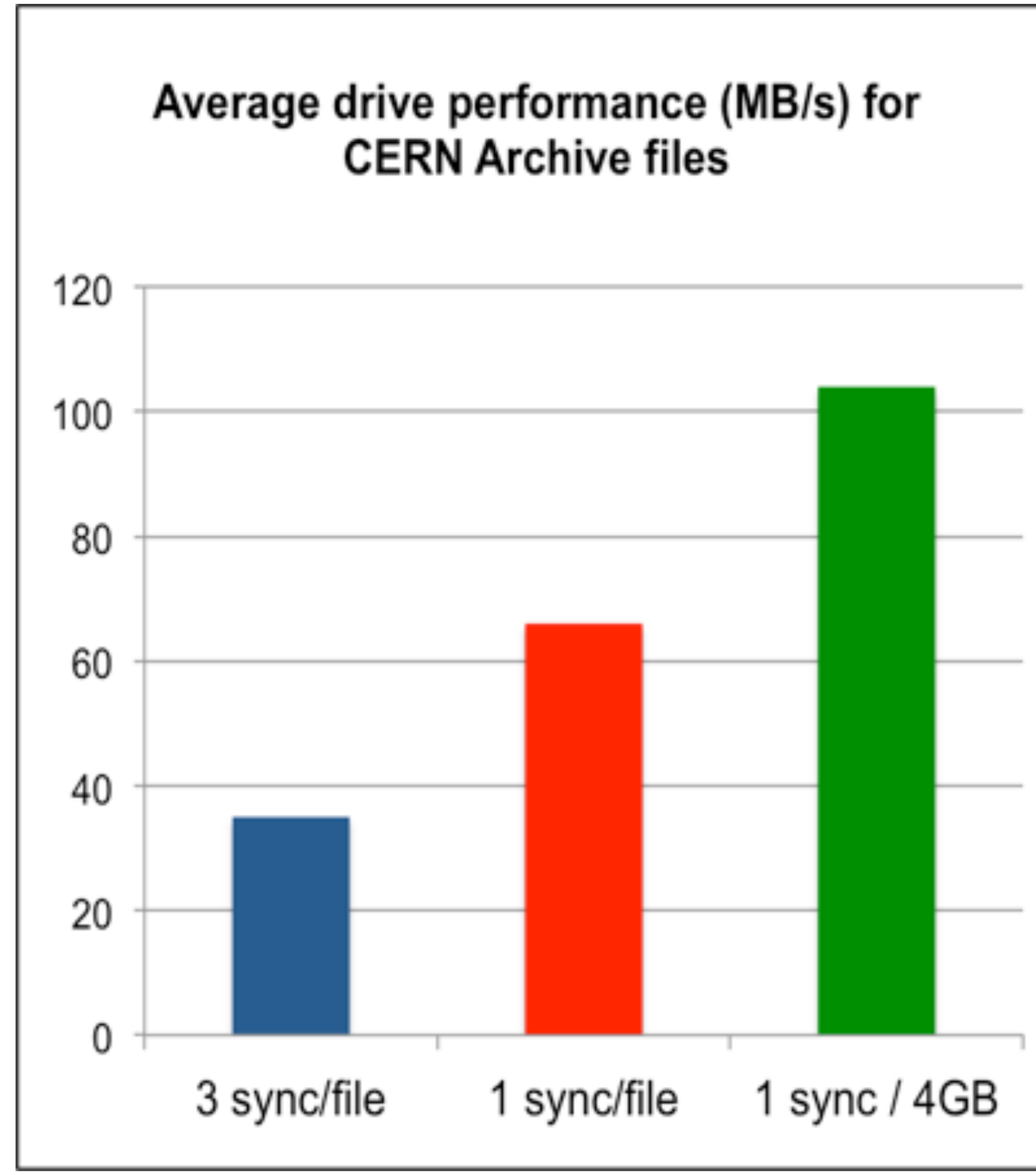
- CERN: every 2-3 years tapes are copied to new format, drives or media
  - economy: recycle existing media at higher density
  - spot potential media or s/w problems
- Significant effort
  - h/w investment (dedicated drives)
  - s/w development & deployment
- Review gain/effort with statistics from current repack round

Efficient Tape Use

- Write aggregation (eg Castor)
  - Independence of I/O unit from user file size
  - Write at tape speed, independent from file sizes
  - Main challenge: risk management as underlying tape format will change
- Read clustering
  - Data set is granule of experiment data management
  - Can we exploit the data set concept?
    - insure file clustering on minimal number of volumes
    - by predictive caching on disk

Tape Writing - Impact of Tape Marks





Average file size of
Currently written files
200 MB

CERN IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Summary

- Distributed data management components for LHC have been successfully tested in data production use cases
  - They work well for LHC production!
    - But the deployment effort is high.
  - Development driven by consolidation and stability
- Focus has moved to analysis use case - main changes
    - low latency performant protocol and file meta data
    - decoupled disk only pools managed by experiments
- Medium term: prepare to integrate new technologies
  - Large in-memory "DB"s and clustered file systems are beginning to change the storage landscape