



# Storage Feature Requests

Jon Bakken  
GDM Coordination Meeting  
October 18, 2005



# A good meeting!



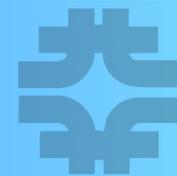
There are weekly CCF integration meetings where we discuss CMS storage and networking feature requests, problems, concerns, and future directions and goals.

- These meetings have been very valuable in making rapid progress on high priority items and on making sure issues important to CMS are being tracked
- Excellent communication between CMS and CCF
- Nothing on the rest of the slides should come as a surprise.

CMS is planning a 2.5 PB dCache, so this coordination is vital to the success of the project.



# PNFS



Need PNFS to scale to expected data rates and access patterns.

- CMS already uses companion to increase throughput
- Also mount namespace without atime, diratime to decrease load
- lots of other pnfs deployment and encp 'tricks'

PNFS is the most vulnerable component in the architecture.

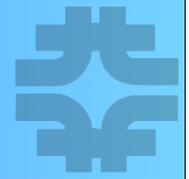
- work in progress at DESY (Chimera) and FNAL

Need frequent reverse pnfs scans to verify Enstore database and pnfs database consistency

- can only be done on backup pnfs server to avoid user impact



# SRM



## Version V2.1 implementation needed

- V2 is the LCG proposed baseline for SC4 and we have even not started testing it yet
- We need V2 to be compatible, including the same ports, with Version V1 so current deployment doesn't stop working

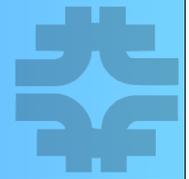
## Automated LamdaStation transfer capability

- Caltech is especially eager to have this working

Need information provider compatible with OSG and LCG GIP



# SRM



## Explicit space reservation

- This is becoming more important as users are filling Tier2 data disks. Physics coordinators need to be able to reserve space for higher priority work
- Implicit space reservation (per transfer) is working well now

## Explicit pinning of files in pools

- Some files always need to be in pools (calibration files) and we need a way for physics coordinators to specify this.



# SRM



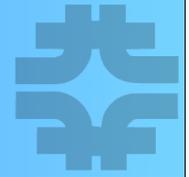
More easily understood web displays of transfers

- How long active, or pending
- How much data transferred, or to be transferred?
- How long have transfers been in the queue?
- What are transfers waiting for before they start?

Need a way to split problems in half - is the problem on the client or server end? Otherwise we need a transfer nanny who monitors failures and makes a judgment on where it failed.



# GridFTP



Performance Markers - to allow upper levels to determine transfer progress

Need NIO implementation of movers to allow for better rates and to kill ongoing transfers

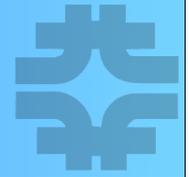
Gridftp doors should be started automatically when needed by SRM on the pool nodes - really needed? We do this manually now.

Bidirectional adapter must be tested

- needed by IP-limited Tier2 centers for scalability



# Authorization

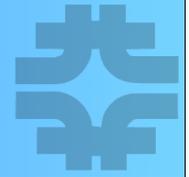


Gplazma project must be put into production

- initially just a dcache.kpwd file replacement
- but really need full integration with Privilege project so storage can use role feature of certificate



# Resilient Pools



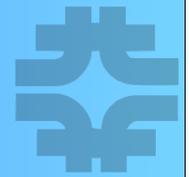
Need this feature to be more robust - no one will look at any errors or exclusion lists - it has to 'just' work

Need to be able to specify that 1 copy of replicated file must always be in certain type of pool (such as raid disk) and the rest should be on worker disks

Needs to take into account space already allocated by SpaceManager



# Enstore-dCache



Most urgent missing functionality in architecture is the HSM-pool flow control and traffic shaper in dCache data pools.

- Get precious files to tape more quickly by sending new transfers to other pools thereby increasing IO rate to tape
- Do this in some type of round-robin fashion
- First version due in December



# Monitoring

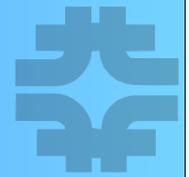


FNAL has deployed extensive monitoring. We need to choose the best parts and make it available to other sites.

- For example, plots of transfer amounts, pool space
- Should be a package on top of core packages



# General



Need checksum to be available for all possible transfer modes

Need better error messages to the users to help diagnose trouble

Need to fix dccp so it doesn't loop on server shutdown

We should to start using gridpp rpms, at least for dCache pools

We need to experiment with cell message routing to find a reliable way for 2 PB of pools to communicate without overwhelming the primary dcache domain

Better logging of all cells, correct details in logs, backup of logs