dCache in a Facility Data Storage Setting

Version 1.0
Created: 14 Apr 2006
Modified: 17 Apr 2006

Rob Kennedy
for the Fermilab Dcache Project and Administration Teams
Computing Division, Fermilab

1) Quick Introduction to dCache (see dcache.org for more)
2) FNAL Dcache Facilities: successes, challenges, NG goals
3) FNAL Experience: issues from h/w to f/s to OS to services
4) Some Dcache Configuration Issues
5) Related Dcache feature development (also interwoven)

Abstract

dCache in a Facility Data Storage Setting

Robert D. Kennedy

Computing Division, Fermilab

Stewardship of large and long-term data storage, providing a trusted data storage facility, is a far more comprehensive administrative process than maintaining smaller-scale or shorter-term data storage. Subtle issues can appear as a distributed data storage system is used to store ground-breaking dataset sizes and deliver that data every day at ever increasing sustained rates. At the same time, the expected overall data storage reliability must be maintained even as it approaches the reliability level of its individual components. These issues can be difficult or impossible to reproduce on smaller scale systems, hence data storage experience at scale is extremely valuable.

This talk discusses such issues and experience in the context of using dCache for the large-scale data storage facilities at Fermilab. While dCache is an excellent choice for distributed data storage at any scale, it has features now that are based on years of experience at O(100) terabyte data storage and delivery scales at Fermilab. In addition, enhancements and new features are being developed to support the expansion of the US-CMS data storage facility based on dCache to the petabyte scale in the very near future.

Context

- Focus: facility storage, experience, dcache.
  - Due to issues with PB-scale data storage systems, not seen or hard to reproduce at smaller scales.
  - Start the list in text... pretty pictures later.
- Not Here: Technical Talk or Feature Sales
  - Other talks and papers can fill that need (dcache.org)
- Given shared LHC data storage mission, vital to share experience, even when not ideal.
  - Issues arise independent of dCache, so talk is often not particularly dCache-specific. Scaling up x10-x100 is hard.
  - Dcache, like any service, has its idiosyncracies.
  - Some FNAL tools not yet in official Dcache RPMs yet. We are interested in perceived value, talking about them.

Dcache in a Nutshell

- Distributed Disk-based Data Storage System
- Born from Patrick Fuhrmann's team (DESY)
- Co-developed by DESY and Fermilab teams
- Started as disk buffer to tape-based HSM
- Evolved as data storage delivery demands placed on it increased. Large CDF Dcache.
- Evolved to fill more niches as LCG and Grid data models and use-cases were defined.
- Uses: large storage facilities and single desktops.
- Much FNAL Dcache feature dev is about improving facility storage since that is our in-house mission.
- http://www.dcache.org

Dcache: Terms and Actors

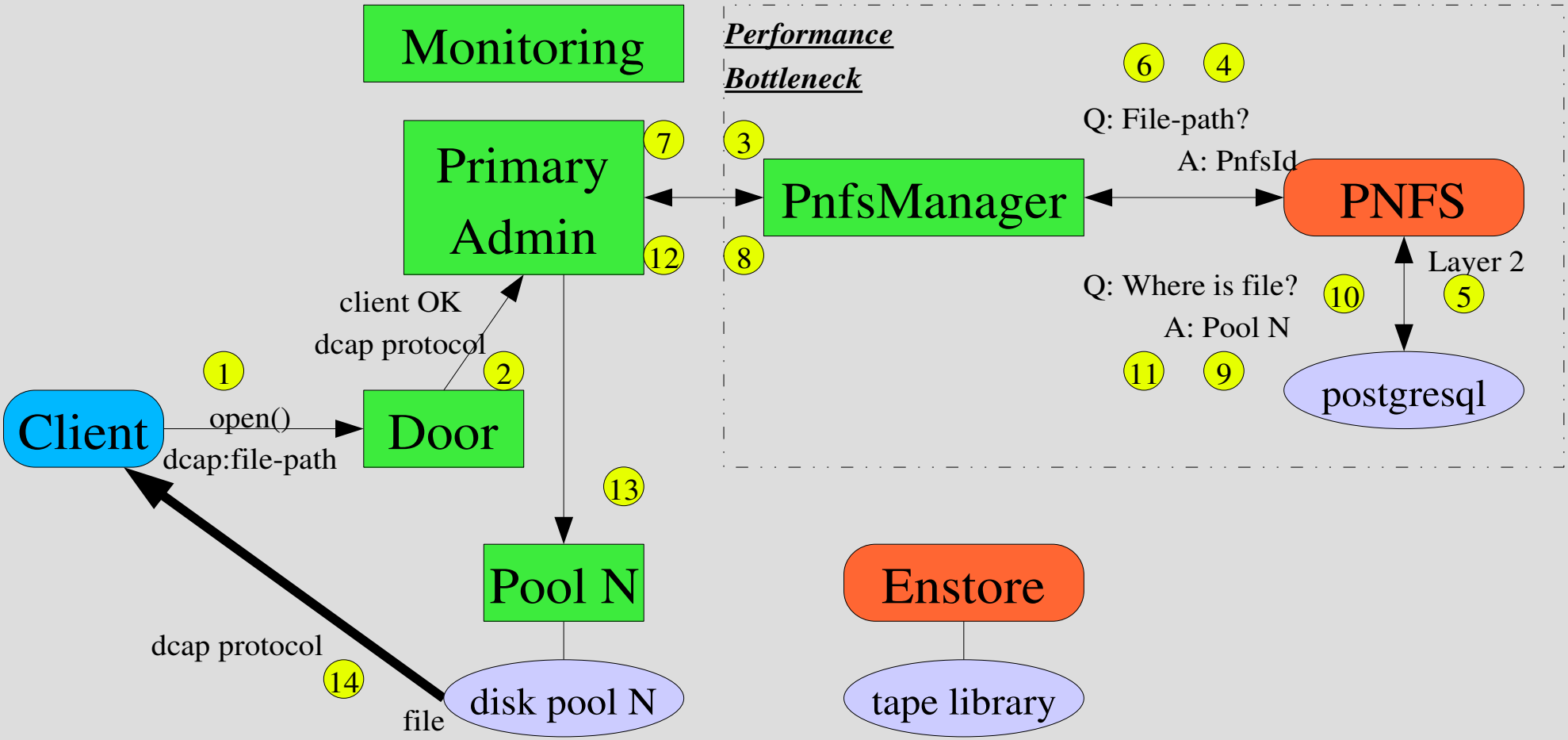
- SPOF: Single Point of Failure (gives the flavor of this talk 8^))
- Admin node: Host to dCache services. (*head, door, monitor*)
- Head node: Host to all singleton dCache sub-services. SPOF.
- Door: portal to authenticate and authorize a client for a protocol
- Monitor node: a host performing various monitoring tasks
- Pool: a virtual data partition in the dCache storage space
- Mover: process receiving or sending data to client.
- Restore: restoring data from MSS (tape) into cache.
- Store: storing data in cache to the MSS (tape).
- P2P: pool-to-pool file transfers inside of a dCache system.
- PNFS: name space, replica catalog for dCache & Enstore. SPOF.
- Domain: Collection of sub-services, “cells”, running in one JVM.
- Cell: An individual sub-service in dCache.

Dcache Storage Models

- Classic: Pools are rate-adapting cache buffers in front of tape-based data repository. New space freed by deleting LRU files. Still most common.
- Volatile: Pools used as simple cache. Same as “Classic” but without tape back-end. Example application – Short-term volatile space for farm I/O.
- Resilient: N copies of each files are maintained in different pools to insure data existence and availability. Like a virtual RAID 0. No tape back-end. Pools can become “full” since no files deleted.
- Hybrid: Mix of the above in a single system.

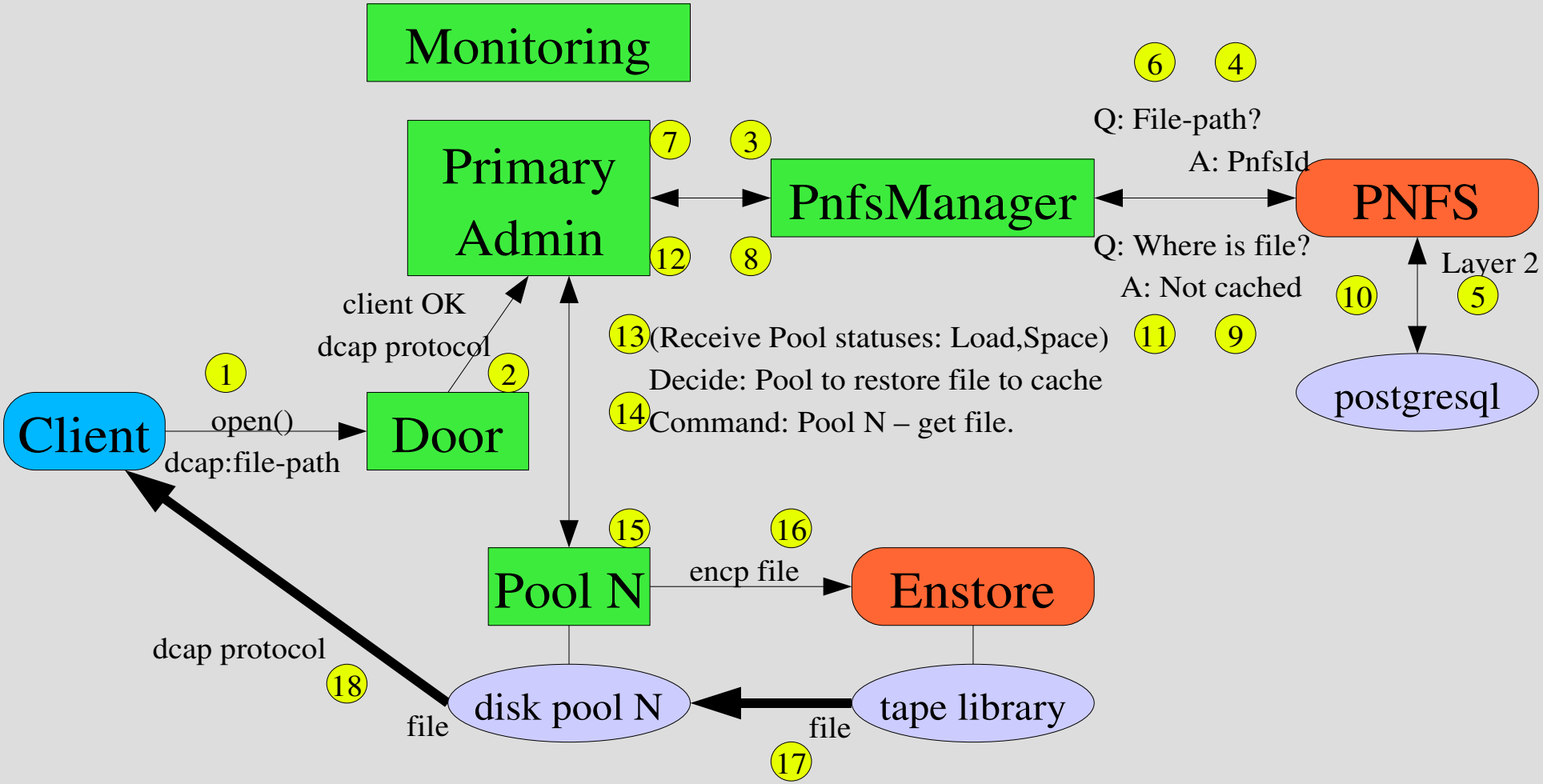
Client Reads a File in Cache

(simplified)



Client Reads a File not in Cache

(simplified)



Contributors
Fermilab Dcache Storage Facilities

- Dcache Project
  - Desy
    - Patrick Fuhmann
    - Tigran Mrktchyan
    - Mathias deRiese
  - Fnal
    - Rob Kennedy
    - Alex Kulyavtsev
    - Vladimir Podstavkov
- Proj, Dev, Mgmt Contrib
  - FNAL
    - SRM: Timur Perelmutov
    - Past,Present: Jon Bakken
    - Eileen Berman
    - Don Petravick
    - Enstore project team...
  - Gplazma: Abhishek Singh-Rana
- Storage Administration
  - FNAL IA Group
    - Gene Oleynik
    - Wayne Baisley
    - David Berg
    - Terry Jones
    - George Smutzka
  - CDF/Run2 Systems Group
  - CMS-T1 Admin Group
- FNAL Expts & Friends
    - CDF
    - US-CMS-T1
    - US-CMS-T2 sites
    - Minos, KteV, MIPP, Miniboone, ...
    - Many others...
- Many others...

FNAL Facility Dcaches

(the basic list. more are here and more are coming.)

- FNDCA: FNAL "Public" Dcache (http://fndca.fnal.gov)
  - First and oldest FNAL dCache facility.
  - Modest size, expanding. Classic model, but to include volatile.
  - Preparing it to become a Grid (OSG) storage facility
- CDFDCA: FNAL CDF Dcache (http://cdfdca.fnal.gov)
  - First O(100 TB) Dcache facility anywhere. Classic model.
  - About 160 TB in size, mostly read pools with sub-caches.
  - Peak data movement: 90 TB/day (dcap transfers to cdf farms)
- CMSDCA: US CMS Tier 1's Dcache (http://cmsdca.fnal.gov)
  - Soon to be the largest Dcache facility. Hybrid model.
  - 160 TB in size, soon to be much larger, widely varying pools.
  - Peak pre-SC4 data movement: 350 TB/day per network plots
  - 4 GB/sec delivered to users – 04 April 2006

Facility Storage Goals

- Stewardship
  - the position of one who “manages another's property, finances, or other affairs”... or their vital scientific data.
- Facility: Offers more than storage space
  - Trusted service – facility assumes many risks.
  - Simple for user – facility hides tech details.
- Some Basic Facility Service Goals
  - Reliable
  - Manageable
  - Secure
  - Cost-effective
  - Predictable
  - Teachable
- Challenges
  - Rapid increase in scale to LHC levels: space and rates
  - Limited funding, staffing, and experience to draw upon
  - Largest industrial data stores tend to be in RDBMSs

Facility Stack View

- Not just “let's run a distributed storage app”.
- Consider facility service stack (no official schema)
  - Disks, disk controllers, controller drivers
  - File system, Operating system, Network
  - Namespace Service (PNFS, Chimera in development)
  - Data File Service (Dcache) seen by customers
  - Service Monitors and Watchdogs
  - Management and Administration
- Facility should perform risk assessment across all of these layers, assuming nothing.
- Monitoring and Metrics should be cross-checks across layers: early warning, quick diagnosis.

Disks, Controllers, Drivers

- Disk Firmware – self-diagnotic led to high-rate of RAID drop-outs that made CDFDCA unusuable (old issue, closed)
- Disk Controllers – weakest link?
  - CRC checksums of files seen to spontaneously change (read directly from platter) - “ spontaneous” corruption
  - Latest drivers from vendor may help. Based on RH or kernel.org tree? Admins into kernel patch&build biz....
- Activities – theme: take nothing for granted
  - Before: dc_bench, test loads to stress W-R-E pattern
  - CRC scans of all files on disk 1/week
  - To Do: plots of service errors by hardware to detect flaky controllers, complement SMART and other diagnostics.
  - (FNAL) Centralized file server procurement and testing?
  - RAID striping performance plus or minus if 25+ readers?

File System, OS, Networking

- File System
  - XFS: best performance for large files, best resilience to file fragmentation, fine robustness... 3 years ago. Still true. Lacks Irix XFS 64 MB blocks for large files.
  - XFS: fine on RAID, but block map-out support issue make it less attractive to run on bare IDE disks.
  - Could do: run XFS defrag on file servers 1/month
- Linux kernel issues
  - 2.6 kernel: some kind of TCP/IP stack “hang”
  - Getting stable MoBo + kernel.N can take time
  - As with: misc. drivers requiring kernel patching?
- Networking – can complicate debugging
  - Host-specific firewall holes, as well as time-outs

Namespace Service

- PNFS Performance Scaling
  - Known bottleneck to next x10 scale in Dcache service
  - Can now scale by running >1 PNFS/Dcache
  - Chimera in testing, to replace PNFS, x10 “faster” than PNFS, can exploit RDBMS scaling solutions directly
- PNFS Operations
  - Lacks monitoring: eg. Request service time (disks slow?)
  - Produces many harmless “NFS i-node mismatches” messages in /var/log/messages. Can hide real problems
  - PnfsManager (PNFS-Dcache interface) request timeouts
- Activities underway
  - Test and deploy Chimera to replace PNFS
  - Consider namespace monitoring, interface monitoring

Dcache Layer

- Dcap reconnect – clients reconnect to alternate pool if original pool fails. So long in Dcache, sometimes overlooked
- Errors in Dcache billing – not necessarily a client request failure. This makes metric tracking awkward since delays in tape system can lead to false “errors in billing.
- Logging content – Some exceptions are normal events, oddities tolerated by (us) developers, some misleading. Much improved over 2 years ago. To do: Log4j.
- Metadata – Content of PNFS, Enstore (tape), and Dcache meta-data diverge if a transfer failure or a service faults. FNAL monitoring under development now to improve response time to divergence appearing. Reminding clients to check their dccp status return codes or ftp equivalent.

Monitoring and Watchdogs
Accumulated FNAL Dcache Add-ons

- Monitoring package – newly revised plotting package from FNAL, based on content of internal Dcache billing database. To do: RPMs for dcache.org as optional package.
- Some watchdogs – CRC scans, door login inactivity (kill if >3 days), short-term checks until issue fixed in code, pool file listings, gather ftp transfer log information for web display, clean-up old logs, create queue plots, login plots,...
  - To do: save to Dcache logs and FTP logs to tape
  - Exploit logfiles. Detect increases in error frequency
  - Long-term hardware-CRC failure plots as early warning
- FNAL: eventual watchdog package(?) – depending on generality of task, may create to avoid duplication of effort and capture best practices.

Management & Administration

- Support for ever-increasing scale means constant stress on components and people.
- Balance what seems like urgent needs with continued “investment” in tasks like planned software dev and administrative procedures.
  - This investment in infrastructure prevents diminishing returns in kluge-upon-kluge work
- Crucial to keep planning 3 months ahead, even as “fires”b reak-out with scale-ups.
  - Track time spent in “fires” – staff being overwhelmed?
  - Meet after each “fire” to write lessons learned.
  - Improve procedure if possible to simplify tasks next time.

Dcache Pool Configuration

- File family affinity and “sub-caches”
  - Multi-VO space accounting
- Pool size choice – arbitrary, but some ideas:
  - Map pool space to a disk controller for errors
  - 1 to 3 pools per controller... low number
  - Pools small enough for sub-cache granularity
- Write-only pool nodes vs read-write
- Compute nodes can be problematic as storage servers. Dedicated RAID preferred.
  - Often loaded by client jobs. Worker node does not respond to file requests or dcache pings.

Dcache Cost Model Config

- http://www.dcache.org/manuals/experts_docs/PoolCost.html
- Example of extremes
  - Fill all empty space first (load factor = 0)
  - Spread load evenly. File space may be reused in some pools before empty space consumed (space factor = 0)
- Cost Model issue: has proven awkward at times
  - Particular values may not produce same behavior as system changes, eg. Add many new empty pools.
  - Or behavior *desired* changes as system changes
- Practice tuning test stand for effect, then ***tune*** production
- Work in progress (DESY): partitioning in the cost model
  - Set different parameters for each “partition” in system
- Work under discussion: how to simplify the translation of desired gross system behavior into a few parameters.

Admin Procedures

- Work in progress (DESY): Simpler upgrade procedure to preserve localized configuration
- Client jobs generally lack check-pointing, so storage downtimes have big impact on farms
  - Eg. long jobs must be stopped well before downtime
- Desirable to simplify administration
  - Simple administrative interface with feed-back
  - “Shift crew” monitoring: service up or not, period
  - Simplify common administrative tasks done on a live system to avoid downtime, add more such tasks
  - Expand “standard” models for installation and configuration to support more variations out of the box

FTP-Related Development

- FTP Scalability
  - Configure FTP door on each pool node (US-CMS-T1)
  - SRM on layer on top of FTP to spread load
  - New gridftp X-Mode may also help (work in near future)
- FTP and Firewalls & NATS
  - Arbitrarily hard use-cases may be requested
    - NATs commonplace outside facility environment
  - Should expand supported use-case examples to cover more known-to-work configurations
  - FTP vs. HTTP vs. TFTP vs. in-house protocol
- FTP Performance
  - Re-implement with Java NIO (work in progress)

More Development

- “Billing”s calability (under consideration)
  - Anticipated bottleneck when scale x4
  - Have observed cpu,memory usage in detail
- HSM and User I/O bandwidth competition
  - So many user read/writes that HSM tape drives cannot be kept busy with data from file in pool
  - DESY work in progress, to be tested soon
- SRM (explicit) space reservation ~ Oct 2006
  - Separate project, but impacts Dcache facilities
  - Complex feature: effectively an internal quota system, works even with unstable resilient pools
  - Not a global space accounting system

Conclusion

- Incomplete discussion, but its a start.
  - Modest list of experiences out in the open.
- Facility storage, whatever implementation, requires more consideration than CPU Farm
  - Not a disk space farm expanded by adding disk
  - Storage facility must preserve state long-term
- Dcache has been successful in role at FNAL

  Years of experience and recent US-CMS-T1 scale-up have led to many ideas on how to improve. Working through list. Do not be discouraged – this is thoroughness, not concern.
- This talk can be followed in more depth on specific topics, or in more technical detail....