Technology Overview: Mass Storage

D. Petravick

Fermilab

March 12, 2002

Mass Storage Is a Problem In:

- Computer Architecture.
- Distributed Computer Systems.
- Software and System Interface.
- Material Flow(!)
  - Ingest.
  - Re-copy and Compaction.
- Usability.

Implementation Technologies

- Standard Interfaces
  - which work across mass storage system instances.
- Storage System Software
  - A variety of storage system software.
- Hardware:
  - Tape and/or disk at large scale.

Interface Requirements

- Protocols for LHC era….
  - Well known.
  - Interoperable.
  - At least either Simple or Stable.
- Areas.
  - File Access.
  - Management.
  - Discovery and “Monitoring”.
  - (perhaps) Volume Exchange.

Storage Systems

Storage Systems

- Provide network interface suitable for a very large, virtual organization's integrated data systems.
- Provide cost-effective implementations.
- Provide permanence as required.
- Provide availability as required.

Network Access Protocols for Storage Systems

- IP based, with "hacks" to cope with IP performance issues. (i.e. //FTP)
- Staging:
  - Domain Specific w/ Grid Protocols under early Implementation.
- File system extensions:
  - RFIO in European "DataGrid."
- Management – "SRM."
  - Prestage, Pin, Space reservations, etc.

File System Extensions to Storage Systems

- Goal: reduce explicit staging.
- Environment is almost surely distributed.
  - Storage systems are often implemented on their own computers.
- Libraries have to deal with:
  - Performance. (read ahead, write behind)
  - Security
- Implementation techniques include
  - Libraries (impact: relink software).
  - Overloading the POSIX calls in the system library.

File System Extensions to Storage Systems

- Typically, only a subset of POSIX file access is consistent with access to a managed store.
- Root Cause: conflict with permanence.
  - Modification (rm –rf /).
  - Deletion.
  - Naming.

Hardware

Tape Facility Implementation

- Expensive, specialized to set up.
- “easy” to commission in large quantities of media with great likelihood of success.
- Good deal of permanence achieved with one copy of a file.
- (formerly) clearly low system cost over the lifetime of an experiment.
- Currently -- all data written are typically read.
- Trend is for diverse storage system software, with custom lab software at many major labs.

Magnetic Disk Facilities

- Current Use: Buffer and cache data residing on tape.
- Requirements: Affordable, with good usability.
- Implementations:
  - Exterior to mass storage systems.
    - » Local or network file system.
    - » Files are staged to and from tape.
  - Internal to storage system.
    - » Provides buffering and caching for tape system.
    - » Transparent Interface:
      - DMAPI, Kernel level interfaces rare (unknown).
      - file system extensions to storage systems. (rfio dccp).

A Good Problem

- Disk capacities have enjoyed better-than-Moore's law growth in capacity.
  - Doubling each year.
  - Subject to superparamagnetic limit, market.
- Tape doubles every two years.
  - right now 60 GB tapes, ~200 GB disks.
- What sort of systems do these trend enable?
  - An Immense amount of disk comes for "free".
    - » Storage systems co-resident with compute systems?
    - » Relax the constraint that staging areas are scarce.
  - Explore disk based permanent stores.

Any Large Disk Facility - Usability

- MTBF Failures (failure of perfect items).
  - » Mechanical.
    - (perhaps) predictable by S.M.A.R.T.
  - » Electrical (failures dues to thermal and current densities)
    - Mitigated by good thermal environment
    - (perhaps) mitigated by spin down.
- Outside of MTBF failures.
  - – Freaks and Infant mortals.
  - – Defective batches.
  - – Firmware.
- BER failures (esp file system meta) data.

Disk As Permanent Storage

- Mental Schema:
  - – Many replicas ensure the preservation of information.
    - » Allows the notion that no single copy of a file be permanent.
  - – Permanent stores.
    - » Backup-to-tape model (i.e. read ~never).
    - » Only-on-Disk model.

Disk: Backup-to-tape Model

- Is Conventional.
- Each byte on disk is supported by a by a byte on tape.
- (perhaps) backup tape need not be supported in an ATL.
  - » Some technologies Slot costs ~= tape costs.
- Tape plant < ½ the size of read-from-tape systems.

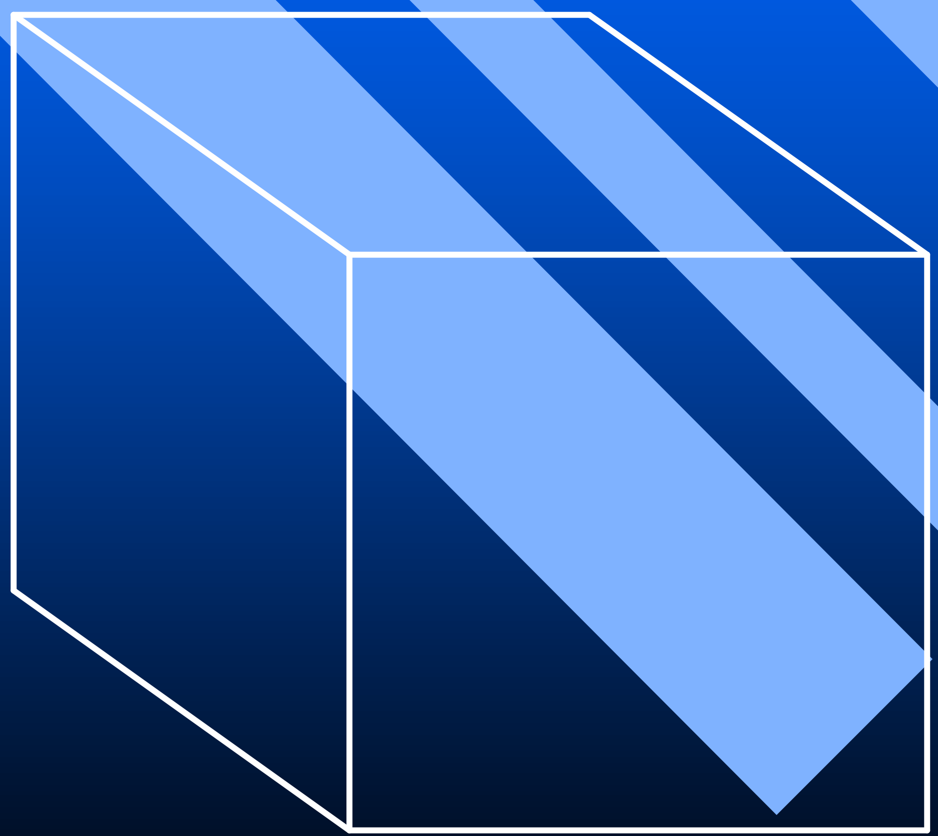
Disk: No Backing on Tape

- Requires R&D.
  - Who else is interested in this?
- Conflicts w industry's model that important data is backed up anyway.
  - This is built into the assumption for quality of raid controllers, file systems, busses, etc.
- Market Fundamentals (guess).
  - Margin in highly integrated disk > Margin in tape > Margin in commodity disk.
- Material flow problem – would you rather commission 200 tapes/week or 100 disks/week?

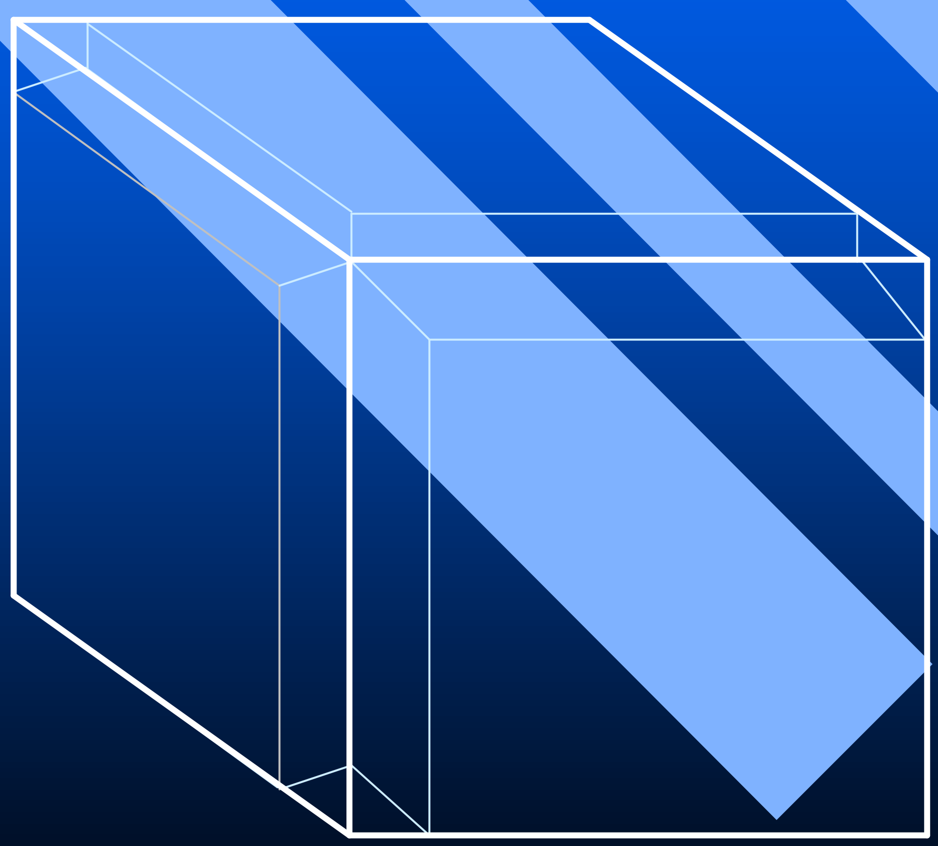
No Backing on Tape – Technical

- At large scale users see the MTBF of disk.
- Need to consistently commission large lots of disk.
- Useful features (spin down, S.M.A.R.T) can be abstracted away by controllers.
- Would like a better primitive than a RAID controller.

Imagine Disks Arranged in a Cube (say 16x16x16)

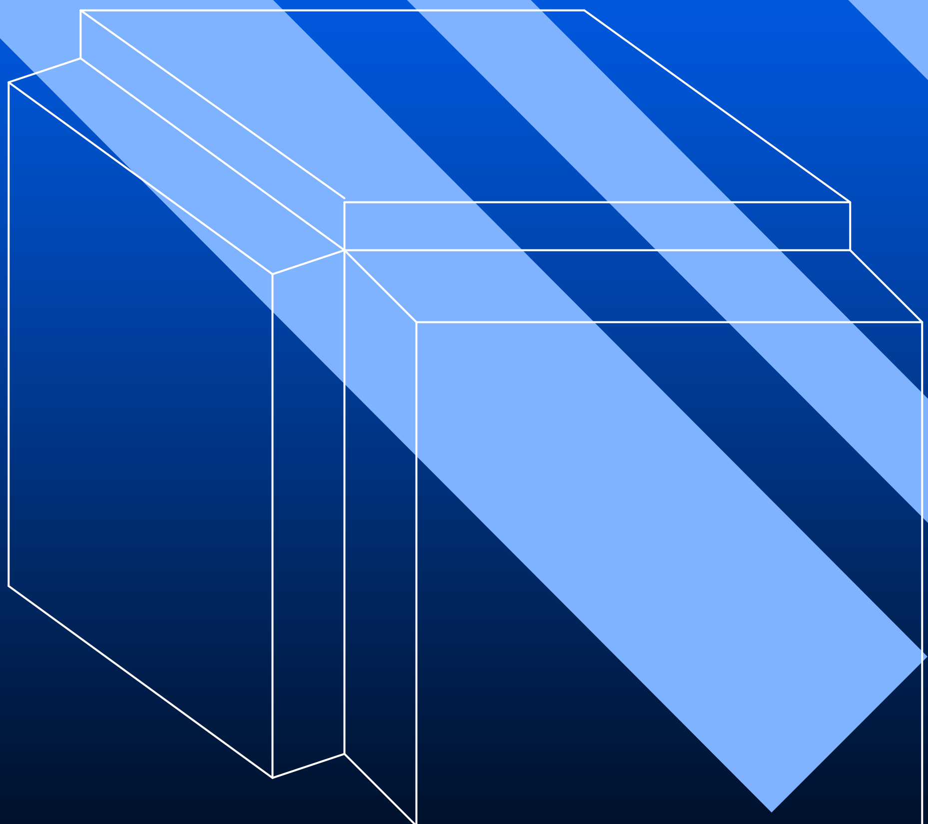


Add Three Parity Planes



Very High Level of Redundancy:

3/19 Overhead For 16x16x16 Data Array.



Handles on Cheap Disk in the Storage System

- Program oriented work in progress today…
- TB commodity disk servers. (Castor, dCache….).
- "small file problem" -- Today's tapes are poor vehicles for < 1 GB files.
  - Some concrete plans in storage systems.
    - » HPSS, small files as meta data, backup to tape.
    - » FNAL Enstore permanent disk mover.
- Exploit excess disk associated with farms.

Summary

- Quantity has a Quality all of its own.
- Low cost disk:
  - Is likely to provide significant optimizations.
  - is potentially disruptive to our current models.
- At the high level (well for a storage system…) we must implement interoperation with experiment middleware.
  - Any complex protocols must be standard.
  - Stage and file-system semantics are both prominent.