High Performance Storage System
Roadmap to HPSS v8

HPSS

* Please review disclosure statement on last slide

In this presentation we discuss…

- Capabilities that were introduced in our current release, HPSS 7.1
- Capabilities that will be in our 2010 release, HPSS v7.3
- Significant capabilities that are now in development for our next major release, HPSS v8, tentatively scheduled for mid to late 2011

  - HPSS has undergone continual development, re-engineering, and implementation of new features since 1992 under the guidance of the Technical Committee and Executive Committee of the HPSS Collaboration.
  - HPSS development team strength, across 5 DOE labs and IBM, is as strong now as it has ever been

What was new in HPSS 7.1

- Small File Performance
  - This feature improves the create rates for HPSS small files by reducing the unnecessary overhead.
- Variable Length Storage Segment (VLSS)
  - The VLSS feature provides a new method for managing segments on disk volumes.
  - Helps to avoid allocating too many storage segments, for applications which do not know the size of their files when the files are first opened.
  - The Variable Length Storage Segment feature applies to disk storage segments only, not to tape storage segments
  - The VLSS implementation is faster and more efficient segment allocation methods and better fragmentation control.
  - The allocation method to be used for new file creations is now specified in the Class of Service configuration.

What was new in HPSS 7.1

- Tape Aggregation
  - Provides the capability to aggregate multiple small files into an aggregated tape segment when they are migrated from disk to tape
  - Significantly improve the rate at which small files are migrated in the HPSS system and will also lead to a reduction in HPSS metadata.
- Disk Affinity
  - Optimizes the placement of disk storage segments when files are being created.
  - When a new storage segment is needed, the Core Server selects the least recently allocated disk volume which is managed by the least busy Mover.
  - Reduces potential disk contentions and to improve load balancing for the Movers.

What was new in HPSS 7.1

- Other changes
  - Multiple Class of Service (COS) streams improve the performance of the `changecos` utility.
  - Migration enhancement for file family, to reduce tape mounts.
  - RTM enhancements
  - New `hpssadm` command line options
  - Couple new Client API functions.
  - Metadata conversion utility

What is coming in HPSS 7.3

- Improved high availability including heartbeat across multiple servers and failover for core servers, initially for RHEL.
- End-to-end check sums for data content integrity.
- Tools to assist in tape migration from other systems, initially DXUL.
- User Defined Attributes (UDA) provide the capability to tag files with user data to enable searches and identification of files.
- Support for Security-Enhanced Linux (SELinux), initially for RHEL.
- Support for Texas Memory RAMSAN solid state storage systems.

HPSS roadmap for HPSS v8 timeframe

- Object-based core server architecture allowing multiple core servers.
- Improved high availability including heartbeat across multiple servers and failover for movers, initially for RHEL.
- Additional end-to-end check sums for data content integrity.
- Redundant arrays of independent tapes (RAIT), developed in partnership with NCSA.
- Path Failover and Load Leveling for HPSS SAN3P
- Additional tools to assist in tape migration from other systems.
- GPFS + HPSS features are in a separate roadmap

Next GHI Feature release with GPFS 3.3

- Expected to release in 2Q2010 ←
- Supports important new GPFS 3.3 features.
  - Fast access inode extended attributes.
  - Image backup and restore.
- Other requirements that HPSS/GHI will be releasing include:
  - Enhanced load balancing to HPSS I/O Managers.
  - Support selection of HPSS Classes of Service, and File Families using ILM policies.
  - Checksum support.
  - New monitoring and administration tools.
    - Interactive control over DMAPI queue.
    - Health and status monitoring.
- HPSS 7.3 ←

GHI features beyond GPFS 3.3

- Additional scalability enhancements.
- Deletion of files in HPSS that are no longer referenced.
- Partial staging of large files.
- Administrative interface for configuration.
- Administrative interface to lock files to GPFS.
- Ability to read migrated copies of files directly from HPSS.
- Support heterogeneous clusters.
- Guarantee specific HPSS “hierarchy depth” (i.e. more than one HPSS copy) before file is removed from GPFS.
- Repack sparse aggregates.
- Partial restore of a backup to restore lost or damaged files.
- Utility to identify and reconcile any differences between GPFS and HPSS archive.
- *HPSS 7.3+ (Independent of HPSS releases)* ←

HPSS program goals for capacity and performance

	HPSS v7 2009	HPSS v7.3 time frame 2010	HPSS v8 time frame 2011	Blue Waters time frame V2012-2015
Capacity in bytes	10^{16} 10s of PB	10^{17} 100 PB	10^{17} 100s of PB	10^{18} 1 Exabyte
Number of files	10^{8} 100s of M	10^{9} 1 Billion	10^{10} 10s of B	10^{12} 1 Trillion
File creates per second	10^{2} Hundreds	10^{3} 1 Thousand	10^{4} 10s of K	10^{4} 10s of K

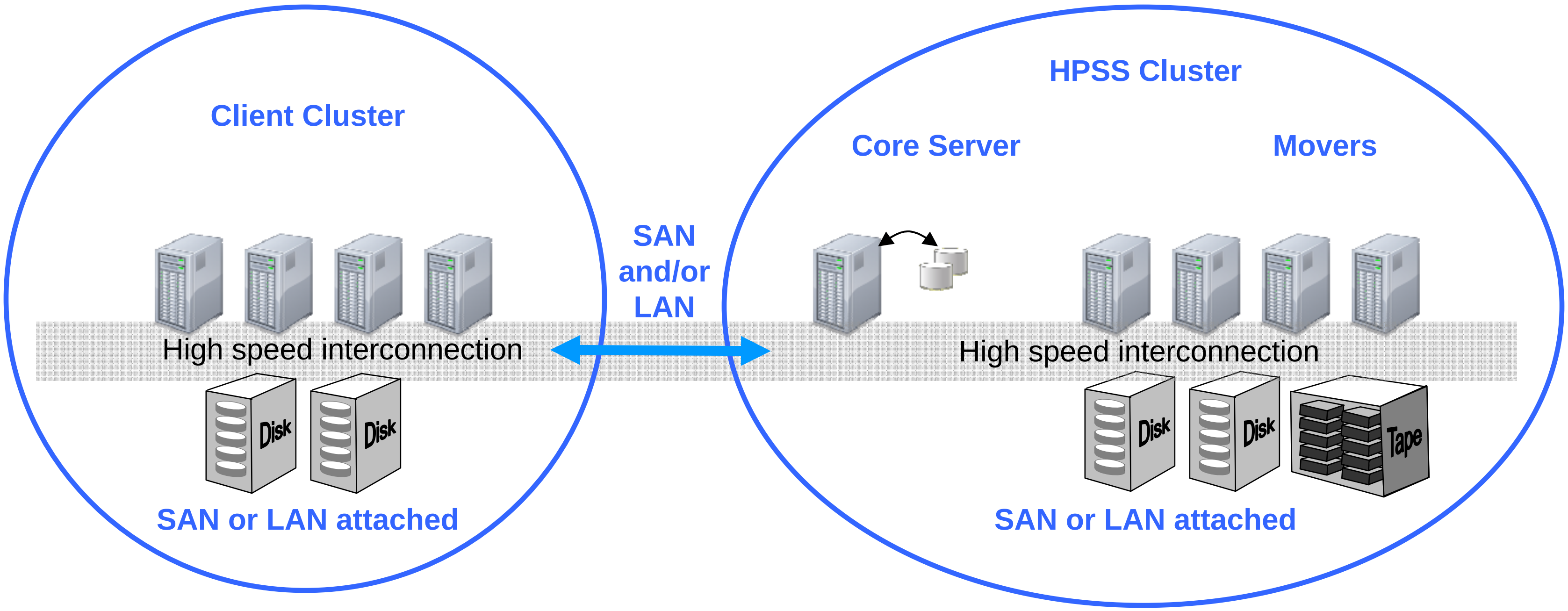
Design directions for HPSS V8

- Discussed in this presentation
  - Clustered, object-based metadata server
  - RAIT (Redundant arrays of tape)

- Other HPSS V7+ and V8 plans not in this presentation
  - Validation of content (checksum)
  - User-defined attributes (allow user metadata)
  - Secure Linux (move toward multilevel security)
  - High availability heartbeat and failover
  - HPSS for GPFS improvements in function and performance

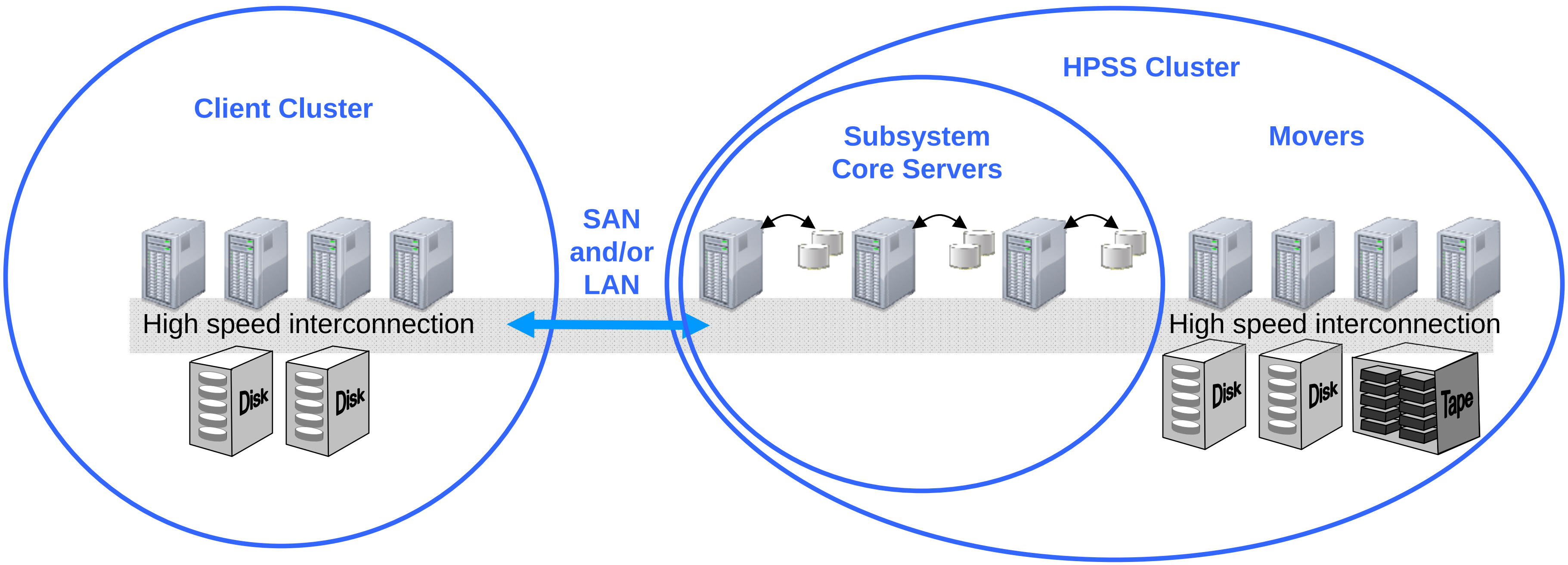
Architectural overview of current HPSS v7

- HPSS has a cluster architecture, like a cluster file system but with tape.
- HPSS presents user with a POSIX and/or extended POSIX file system.
- HPSS scales horizontally by adding Movers and devices.
- Normally there is a single core server and optional backup core server, which include name services, bitfile services, storage virtualization and management, library management, and overall system management.



Architectual overview of HPSS v7 with Subsystems

- While no HPSS site has ever stressed the limits of a single core server, HPSS Subsystem feature enables the use of multiple core servers.
- Subsystems separate the name space into regions, each with a core server.
- Client-side aggregation, provided by access brokers such as the GPFS-HPSS interface (GHI), Gleicher Enterprises HTAR, or LANL's PSI, can effectively make use of HPSS Subsystems for super scalability.



Clustered, object-based metadata server approach

- HPSS v8 prototyping effort is underway to scale metadata management beyond the limits of a single core server (name space region).

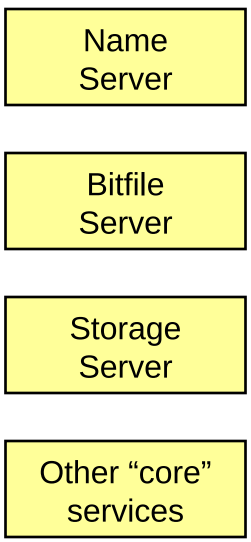
- Developing an approach to
  - minimize rework of existing core server components
  - reduce dependence on external client-side aggregation
  - leverage scalable, commercial, off-the-shelf database technology
  - provide scalable, manageable metadata backup and restore

- HPSS v8 core server will be Linux only
  - However other components such as Movers may be Linux or AIX

Evolution from server to object architecture

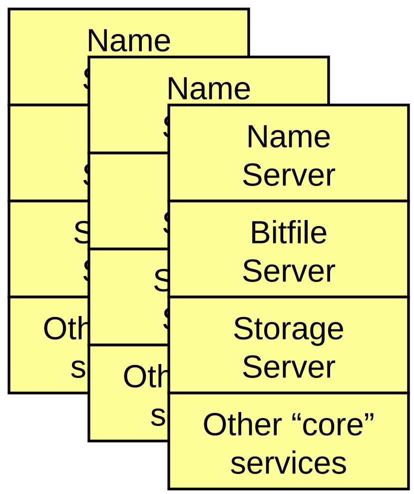
HPSS V1 – V4

- Independent servers
- connected by RPCs



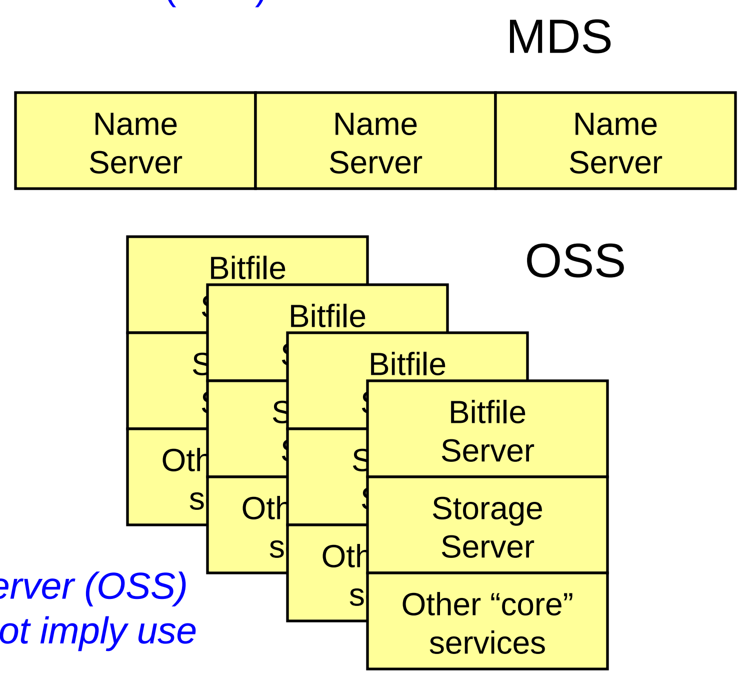
HPSS V5 – V7

- Consolidated "core server"
- Notion of "subsystems" each with a fixed subset of the name space



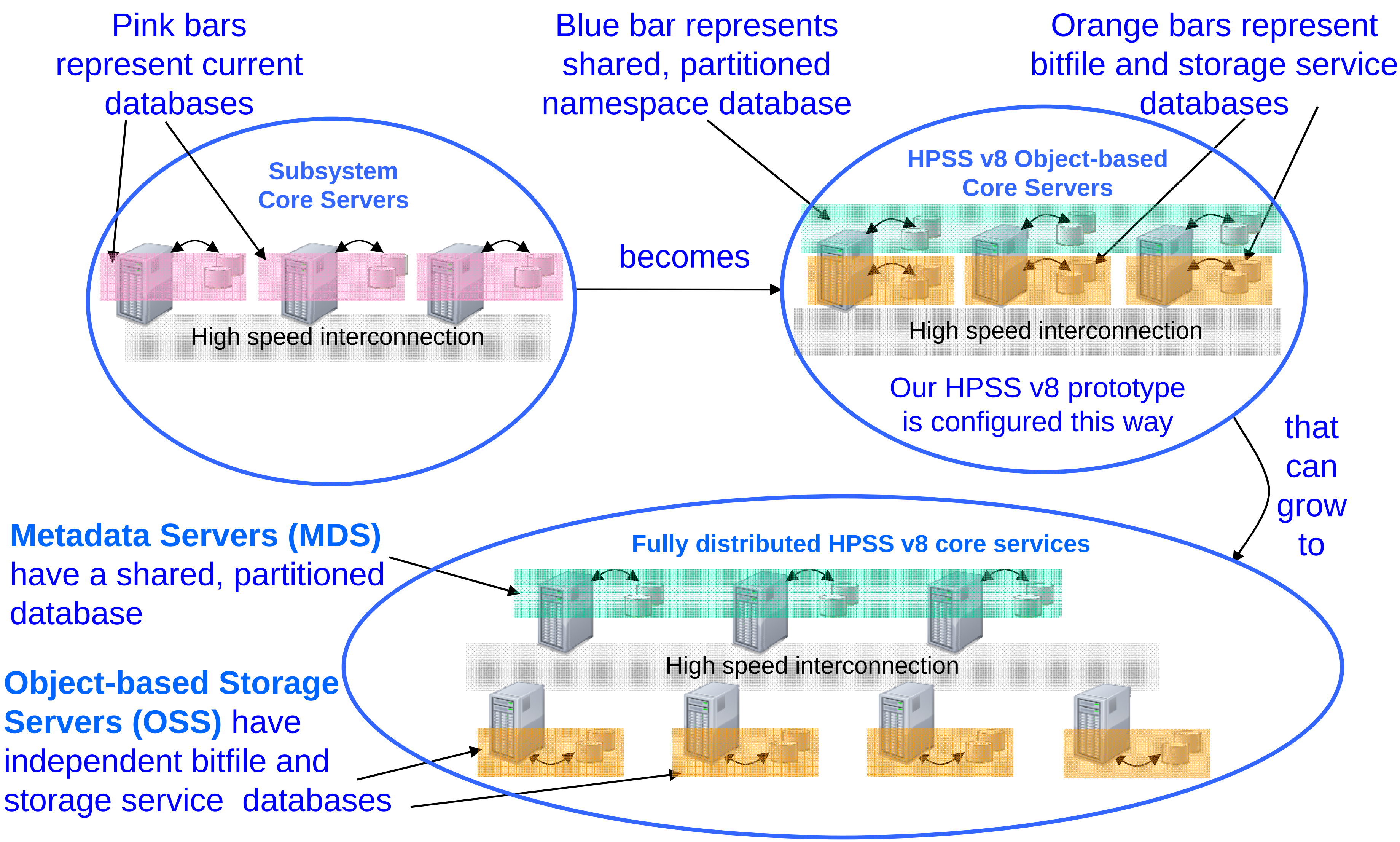
HPSS V8

- Name service cluster provides single, integrated balanced name space called Metadata Server (MDS)*
- Bitfile server and storage server become independent storage object managers called Object-based Storage Server (OSS)*



* *Metadata Server (MDS) and Object-based Storage Server (OSS) are intended to be generic, descriptive names and do not imply use of standards such as ANSI T10*

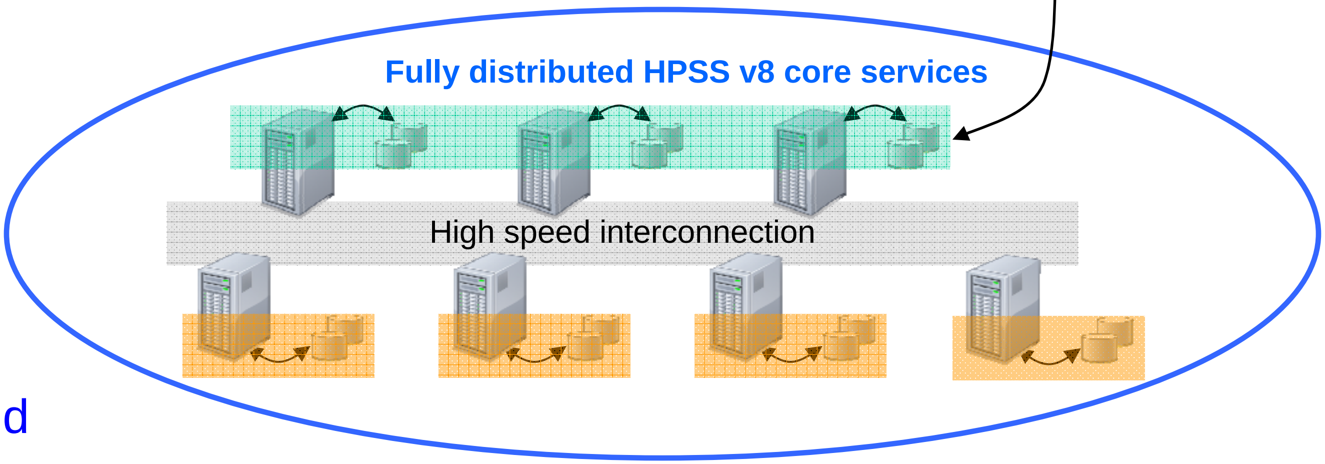
HPSS v8 core services use partitioned databases



Partitioned databases spread the load over a cluster

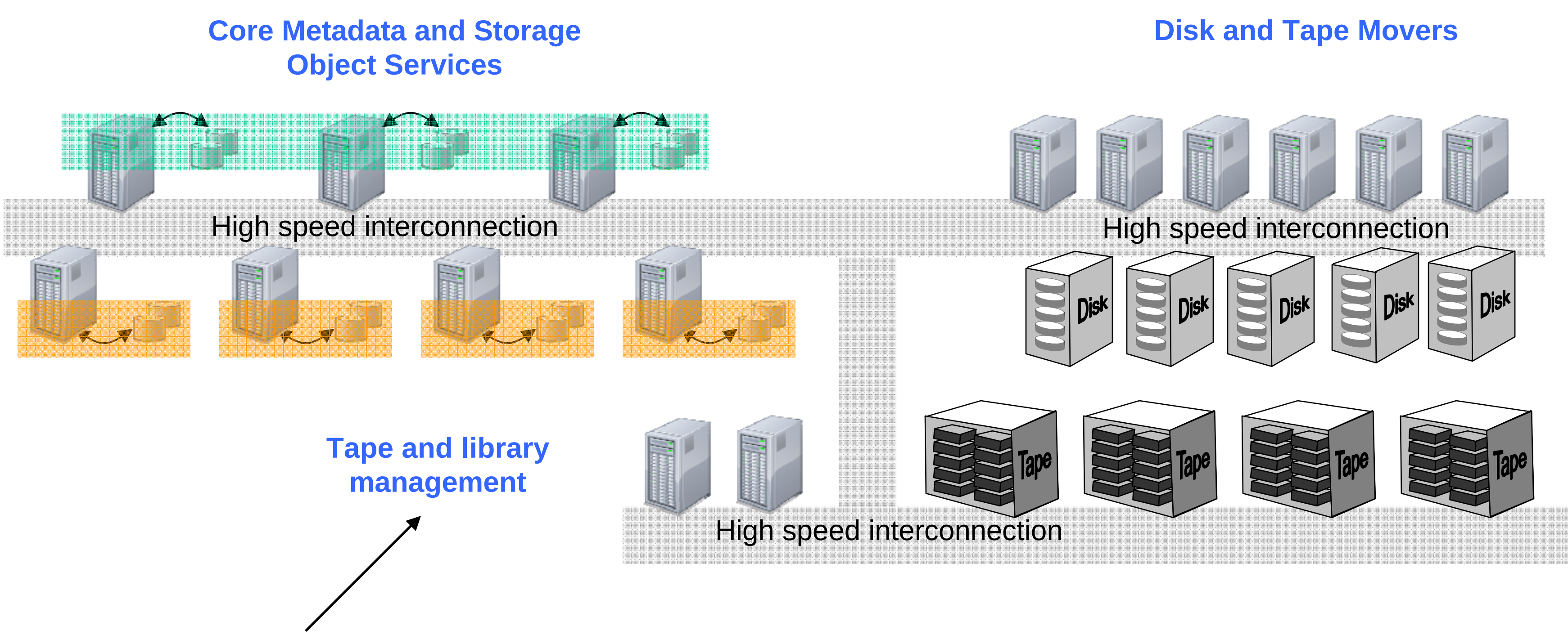
- Allows local access to most frequently used tables
  - Yet allows cluster members to share data when needed
- Balances/spreads metadata
  - Up to 1000 partitions
- Use to parallelize
  - Ingest
  - Queries
  - Backups
  - Reorgs
  - Logging
- Both partitioned and independent databases will be used in HPSS v8

Blue bar represents partitioned database



Architectural overview of full HPSS v8 system

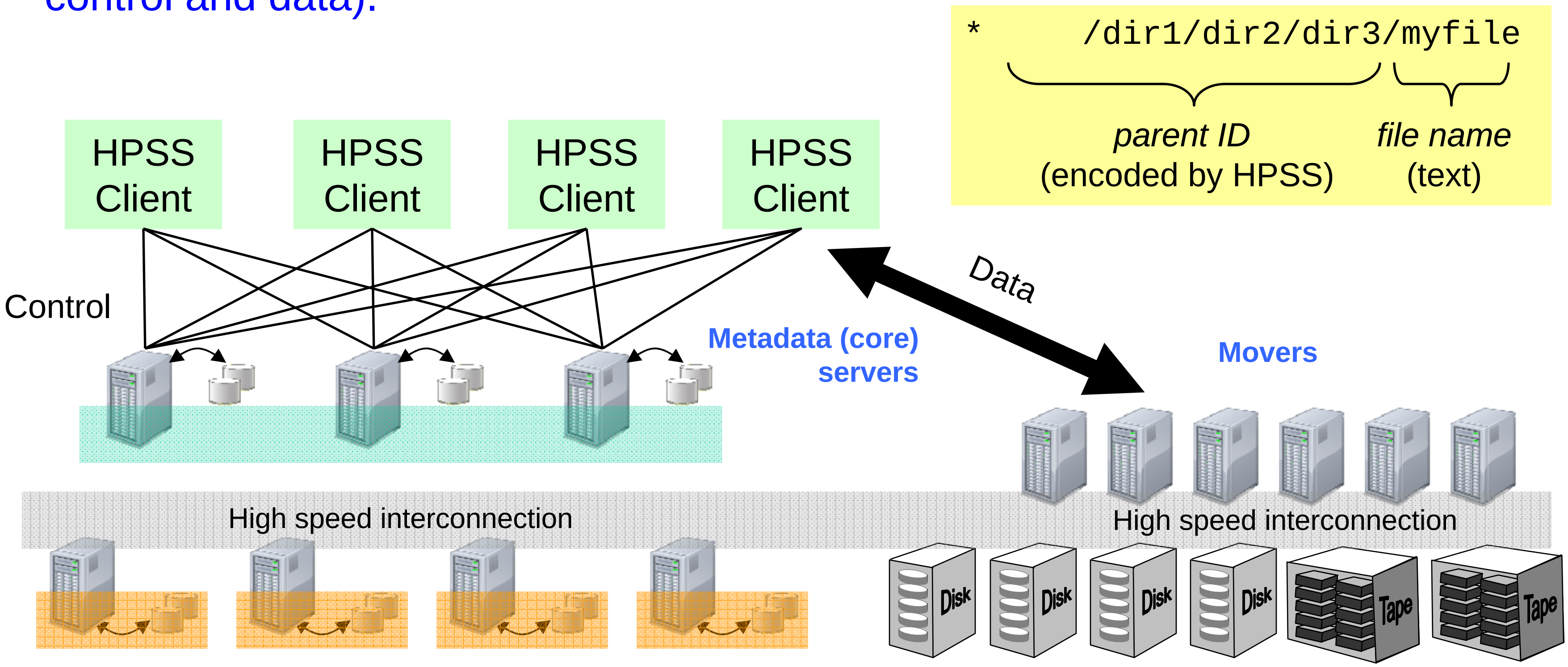
- The Movers are unaffected by the HPSS v8 change in core server architecture



- And the library managers, today often co-hosted on the core server, will be unaffected by HPSS v8 architecture but may be hosted separately

Distributing client requests to the metadata servers

- Clients will direct work to the appropriate metadata server based on a hash value.
- Hash value is created from the *file name* and *parent ID**.
- Should yield a reasonably uniform distribution of MDS accesses without hot spots.
- Data is transferred between client and HPSS mover (supporting separation of control and data).

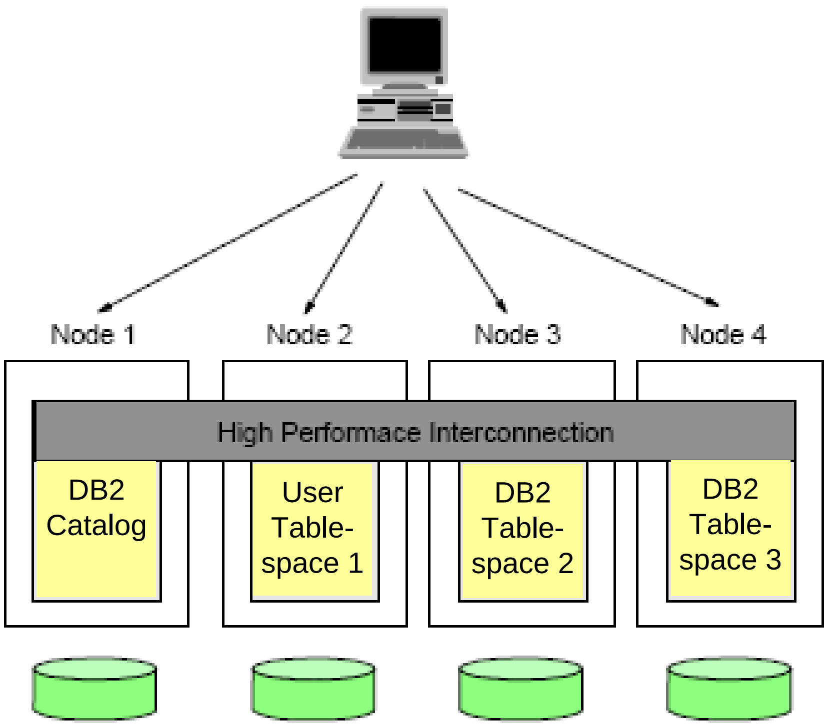


High availability for HPSS v8

- HPSS presently offers a high availability option
  - Legacy AIX only, using IBM's HA-CMP
  - Currently offering high availability core servers and movers using Red Hat Linux HA services
- HPSS v8 will require a higher degree of high availability
  - Can encompass all HPSS cluster nodes including Movers in a heartbeat cluster
  - Will support the notion of universal spare nodes that can take over any function including MDS, OSS, mover, library management
    - Which will encourage the use of uniform building blocks of similarly configured cluster nodes
  - May use DB2's "High Availability Disaster Recovery" (HADR) option to keep near-active-active metadata databases
    - Will not have to restore DBs from backup and logs)
  - Will be required and not optional for systems with clustered core services, and may be required for all v8 installations

Backing up petabytes of metadata

- An exabyte of data will require petabytes of metadata,
  - which must be backed up
  - Which may itself become hierarchical to some degree
- Parallelize backup of partitioned database
  - Single command from catalog node
  - First backs up catalog node itself
  - Then backs up tablespace nodes in parallel
- Parallelize backup of independent, non-partitioned databases
- Archival of seldom-used metadata
  - Investigating techniques for moving “seldomly” accessed metadata out of production database(s).



Why RAIT?

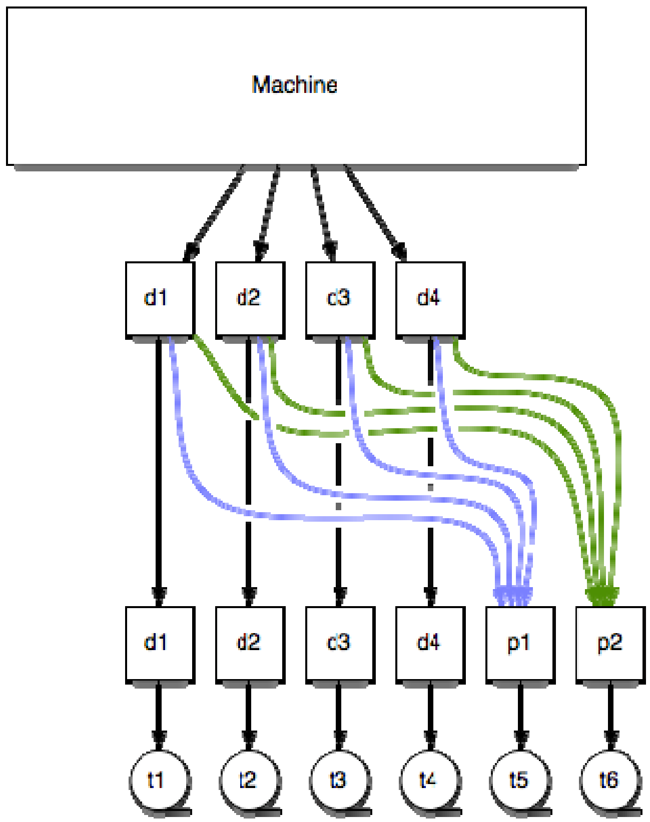
- An ever increasing amount of data is being written to tape with a generally constant error rate.
- RAIT saves cost and increases availability as compared with tape mirroring.
  - For example, a 6+2 RAIT group uses 8 tapes, while comparable mirror would require 12, a 33% savings
  - A 6+2 RAIT group always survives the loss of 2 tapes, whereas 6 tape mirrored could lose data if a tape and its mirror were lost.
- Because of ample provisioning at NCSA, a 6+2 RAIT group will be read or written as fast as a 6-way stripe, but this is not part of the justification.
- RAIT is required and funded by NCSA Blue Waters supercomputer project at University of Illinois Urbana-Champaign.

HPSS RAIT Functionality

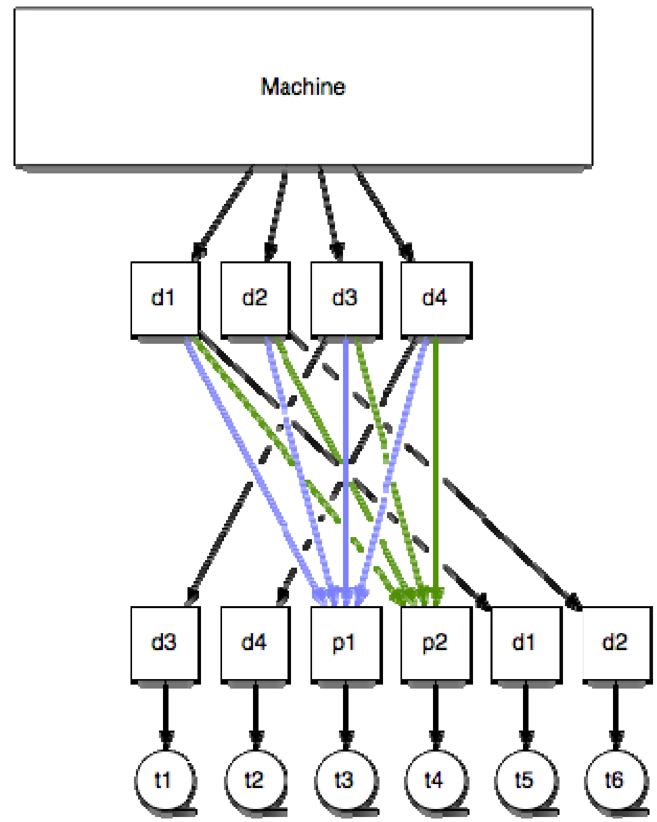
- Built on existing HPSS tape striping support and parallel transfer to/from client
- Adds parity stripes
  - one for “RAIT 5”
  - or two for “RAIT 6”
- Parity spread across all stripes (RAID 6 like)
  - Processing of parity records gets distributed across all the movers, reducing the chance a mover gets behind.
- On the fly read error recovery
- Repack RAIT virtual volumes

Writing data and parity to tape

For the first block of data, four stripes d1 through d4 are written to tape, and two parity stripes p1 and p2 are computed and written.

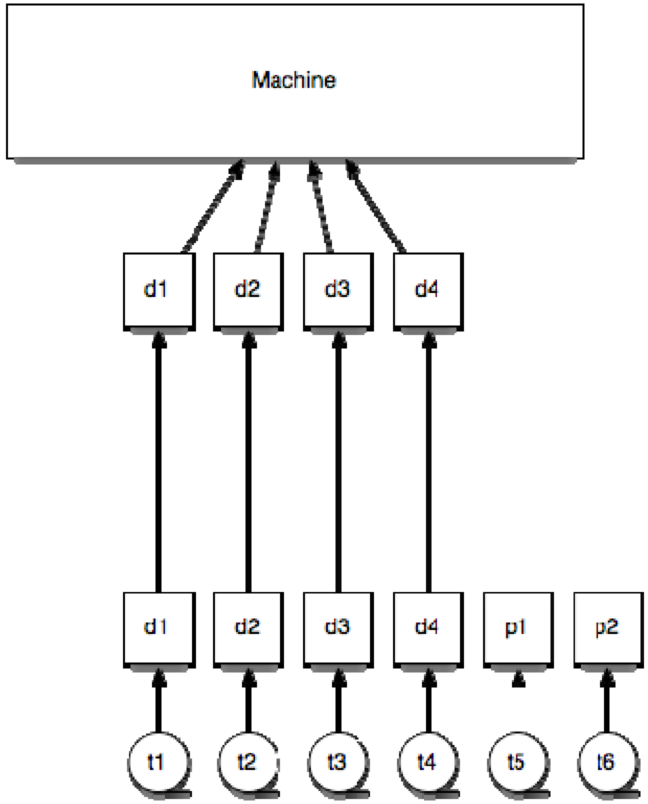


For the next block of data, the tapes selections are rotated left by two tapes, and so on. This helps keep all tapes about the same length.

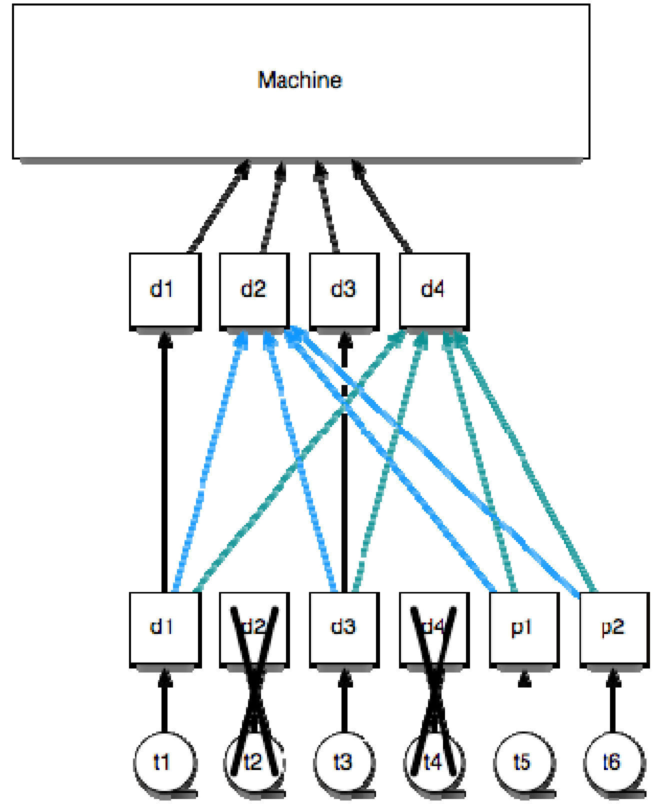


Reading the data

If there are no errors, the parity stripes are not used.



If there are errors, parity stripes are used to re-create the lost stripes. We anticipate this can be done at tape streaming rates.



HPSS Contacts

Bob Coyne – coyne@us.ibm.com

Jim A. Gerry – jgerry@us.ibm.com

Harry Hulen – hulen@us.ibm.com

Patrick Schaefer – pschaef@us.ibm.com

Disclaimer

- Please obtain and read product documentation before deciding to acquire HPSS.
  - Documentation includes the HPSS License Agreement, the Statement of Work, and HPSS and other product manuals.
  - In case of conflict between information herein and product documentation, the documentation shall take precedence.
- Forward looking information including schedules and future product capabilities reflect current planning that may change and should not be taken as commitments by IBM.
- IBM may at its sole discretion discontinue, add, or change HPSS features and function without notice.