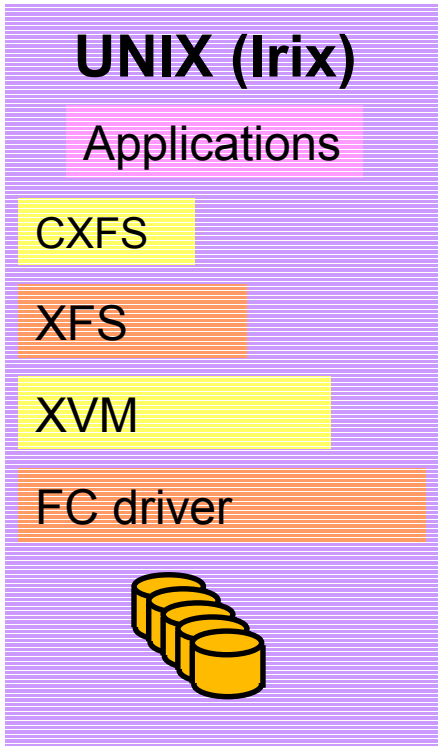
# SGI's Cluster File System - CXFS

**Brian Gaffey**

**File Systems Engineering**

# File Systems Technology Briefing




- Clustered file system features
- CXFS
- File System features: XFS
- Volume management: XVM

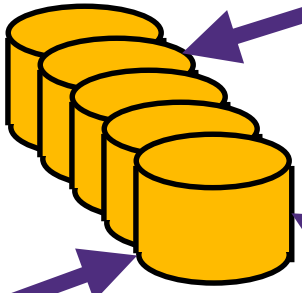

# Clustered File Systems

# CXFS

# CXFS
# Clustered SAN File System

High resiliency and availability
Reduced storage costs

Scalable high performance



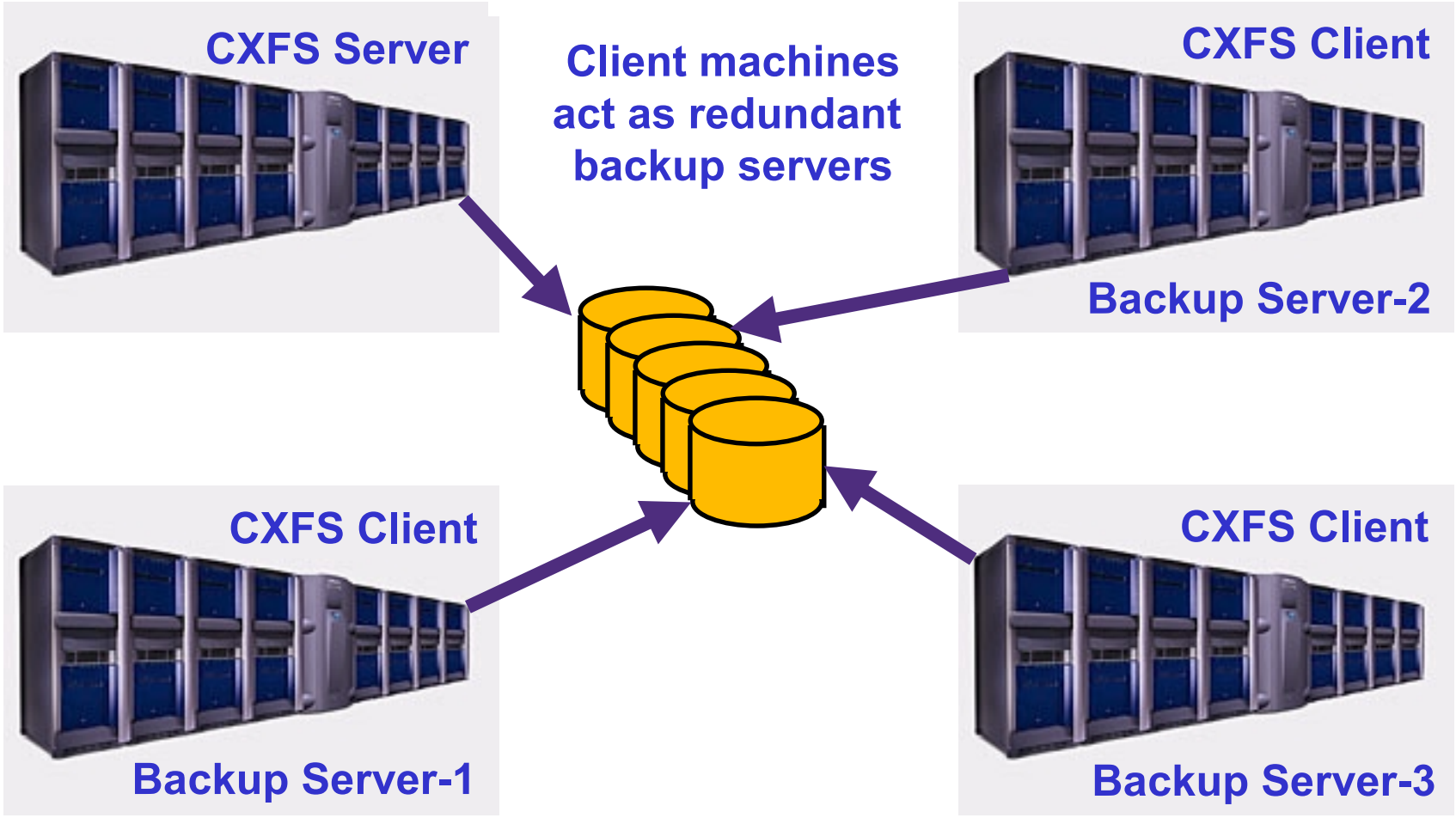
Streamlined LAN-free backups

Fibre Channel Storage Area Network (SAN)

# CXFS: Clustered XFS


- Clustered XFS (CXFS) attributes:
  - A **shareable high-performance** XFS file system
    - Shared among multiple IRIX nodes in a cluster
    - Near-local file system performance.
      - Direct data channels between disks and nodes.
  - A **resilient** file system
    - Failure of a node in the cluster does not prevent access to the disks from other nodes
  - A **convenient** interface
    - Users see standard Unix filesystems
      - Single System View (SSV)
      - Coherent distributed buffers

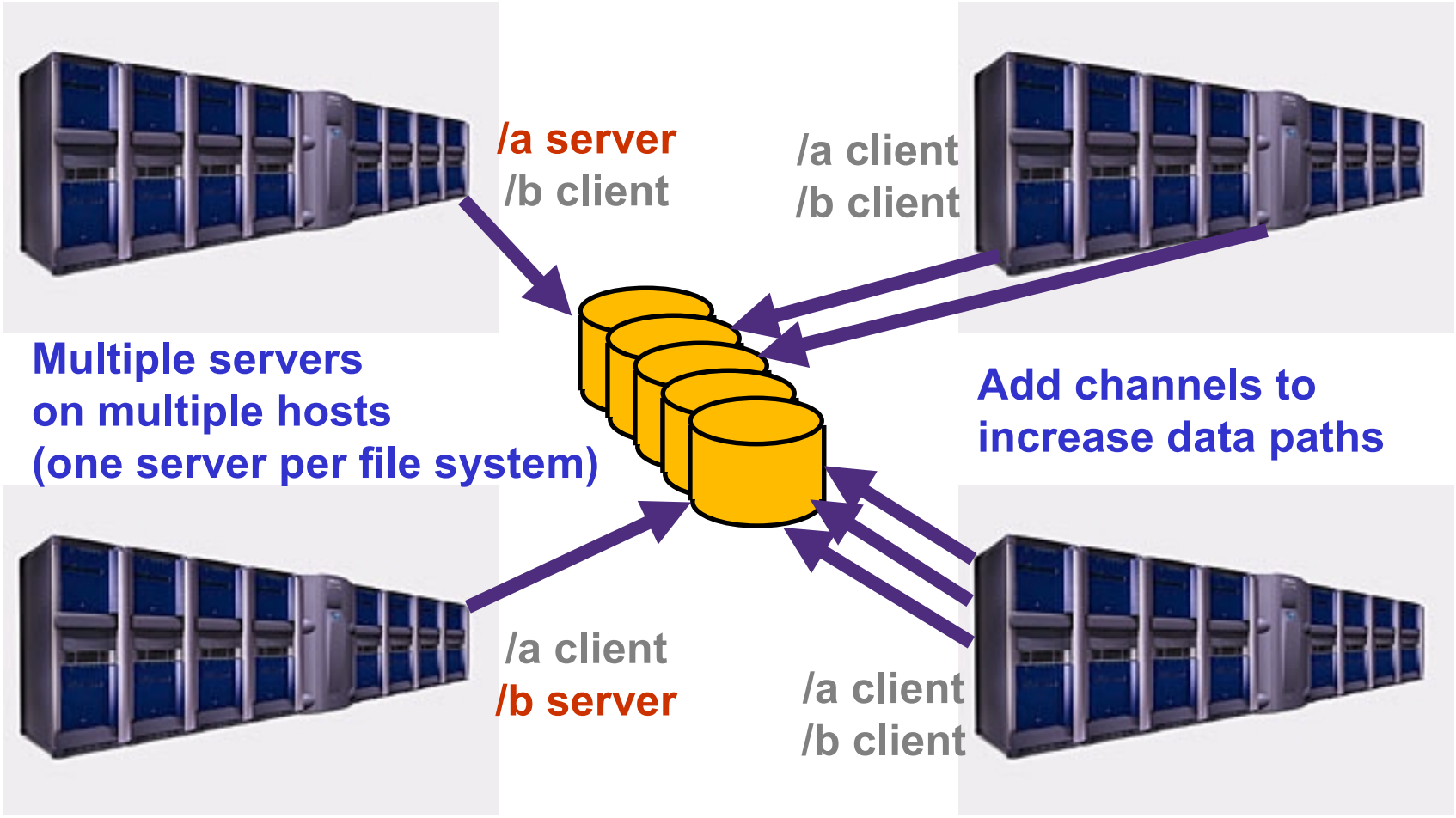
# Fully Resilient - High Availability



# CXFS Interface and Performance


- Interface is the same as multiple processes reading and writing shared files on an SMP
  - Same open, read, write, create, delete, lock-range, etc.
- Multiple clients can share files at local file speeds
  - Processes on the same host reading and writing (buffered)
  - Processes on multiple hosts reading (buffered)
  - Processes on multiple hosts reading and writing, using direct-access IO (non-buffered)
- Transactions slower than with local-files:
  - Shared writes flush distributed buffers related to that file
  - Metadata transactions (file creation and size changes)

# CXFS Scalability



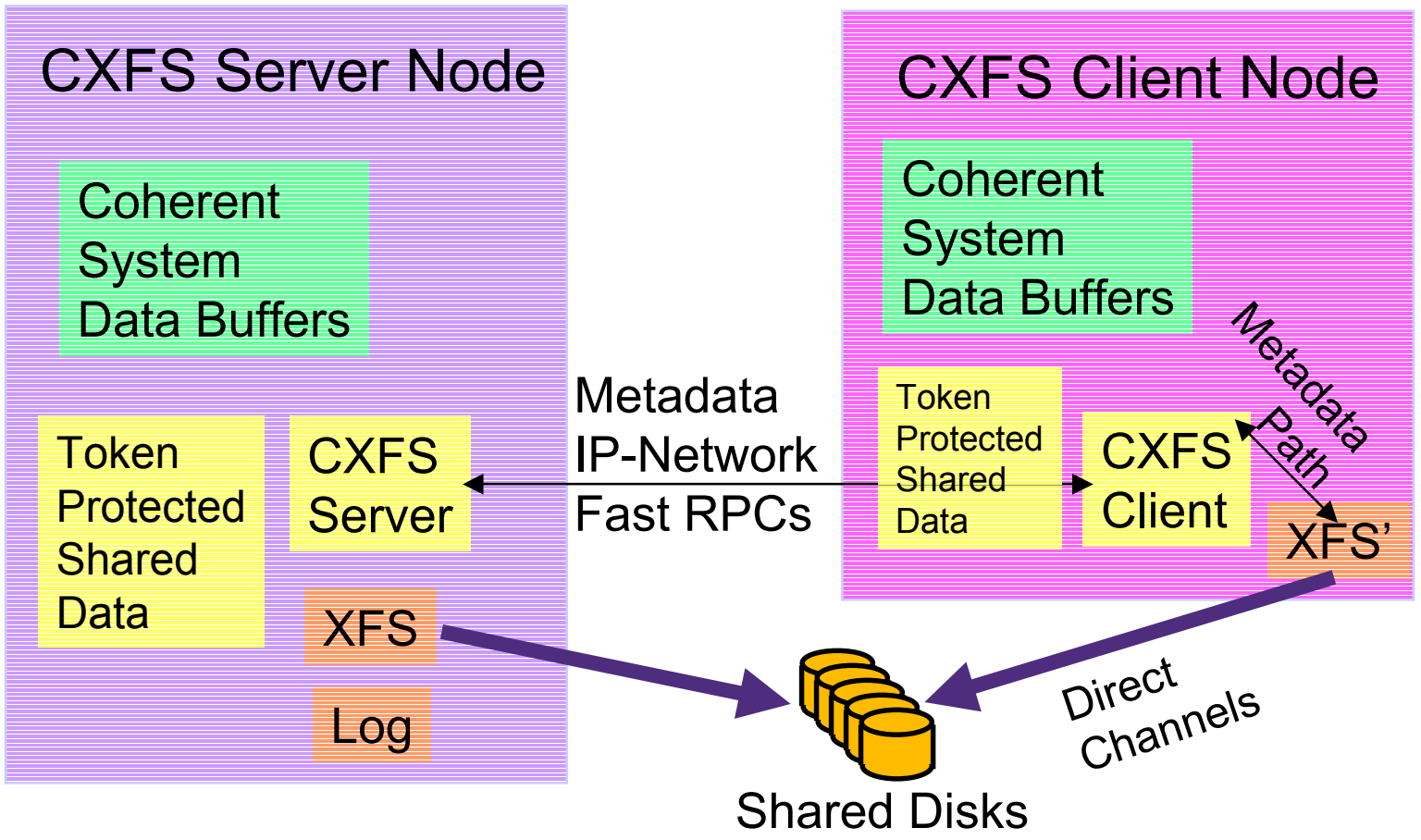
# CXFS Scalability


- Software supports up to 64 clients or servers per cluster
  - Fabric prices will tend to limit the host count to less-than 64
- Multiple CXFS servers
  - One per file system
- Normal local-host buffering for near local-file performance
  - Except when files are used for shared-reads-writes
    - Coherence maintained on a per I/O basis using tokens
- Files accessed exclusively locally on CXFS server see local XFS metadata performance (bypasses CXFS path)
- CXFS supports High-Availability (HA) environments with full fail-over capabilities
- CXFS sits on top of XFS: Fast XFS features

# CXFS Concepts

- Metadata
  - The data about a file, including:
    - size, inode, create/modify times, and permissions
- Metadata server node (a.k.a. CXFS server)
  - One machine in the cluster that is responsible for controlling the metadata of files. It also plays “traffic cop” to control access to the file.
- Metadata client node (a.k.a. CXFS client)
  - A machine in the cluster that is not the metadata server. Must obtain permission from metadata server before accessing the file.

# CXFS Client-Server Metadata Technology



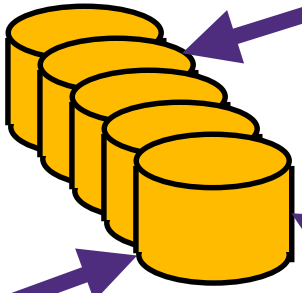
# XFS: A World-Class File System


- Speed
  - Fast metadata speed
  - High bandwidths
  - High transaction rates
  - Guaranteed-rate IO and real-time file systems
- Reliability
  - Mature log-based file system
- Scalability
  - 64 bit: 9 million terabytes
- Flexibility
  - Dynamic allocation of metadata space

# CXFS — Clustered SAN File System

High resiliency and availability
Reduced storage costs

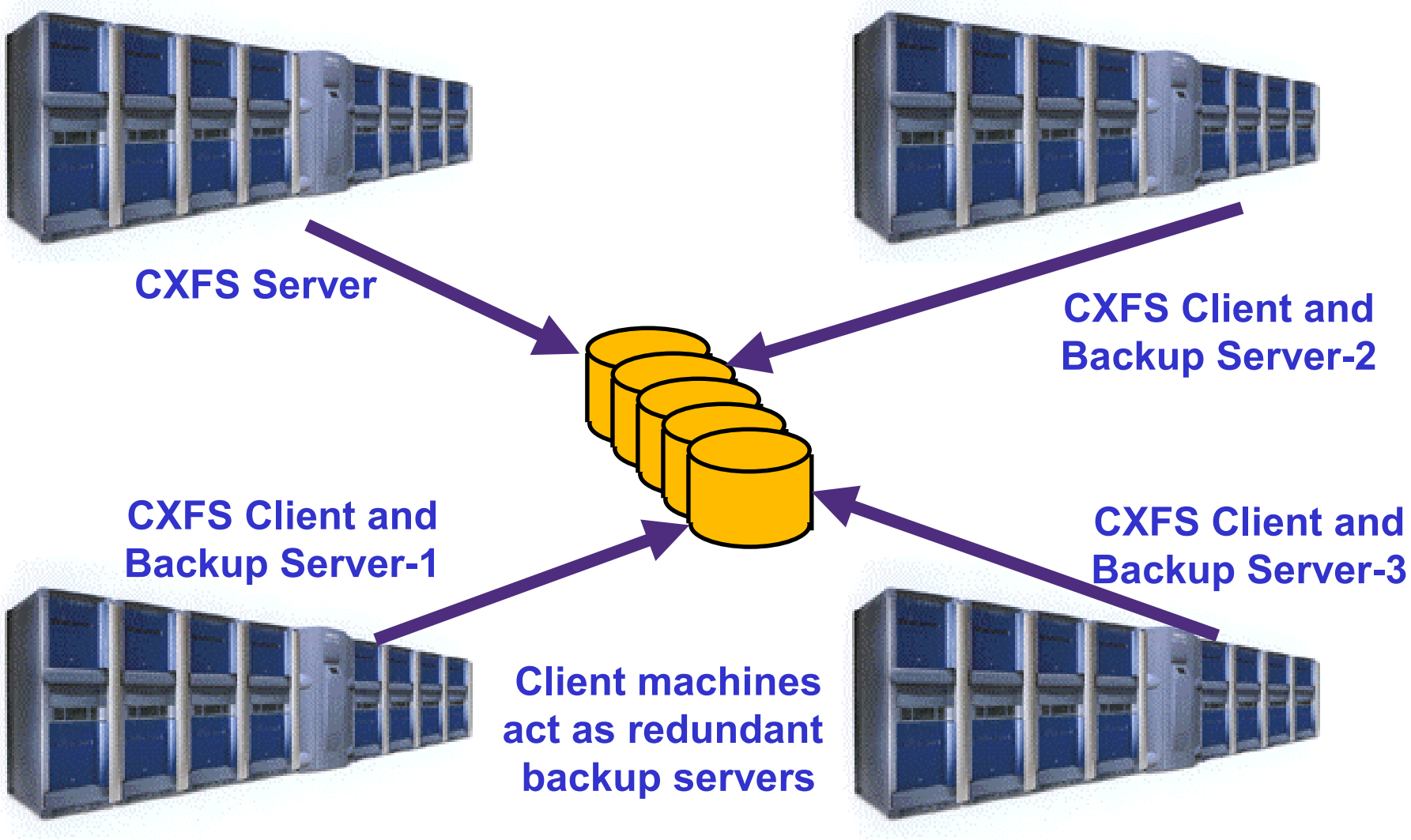
Scalable high performance



Streamlined LAN-free backups

Fibre Channel Storage Area Network (SAN)

# Fully Resilient - Highly Available



# Supports Full POSIX File System API

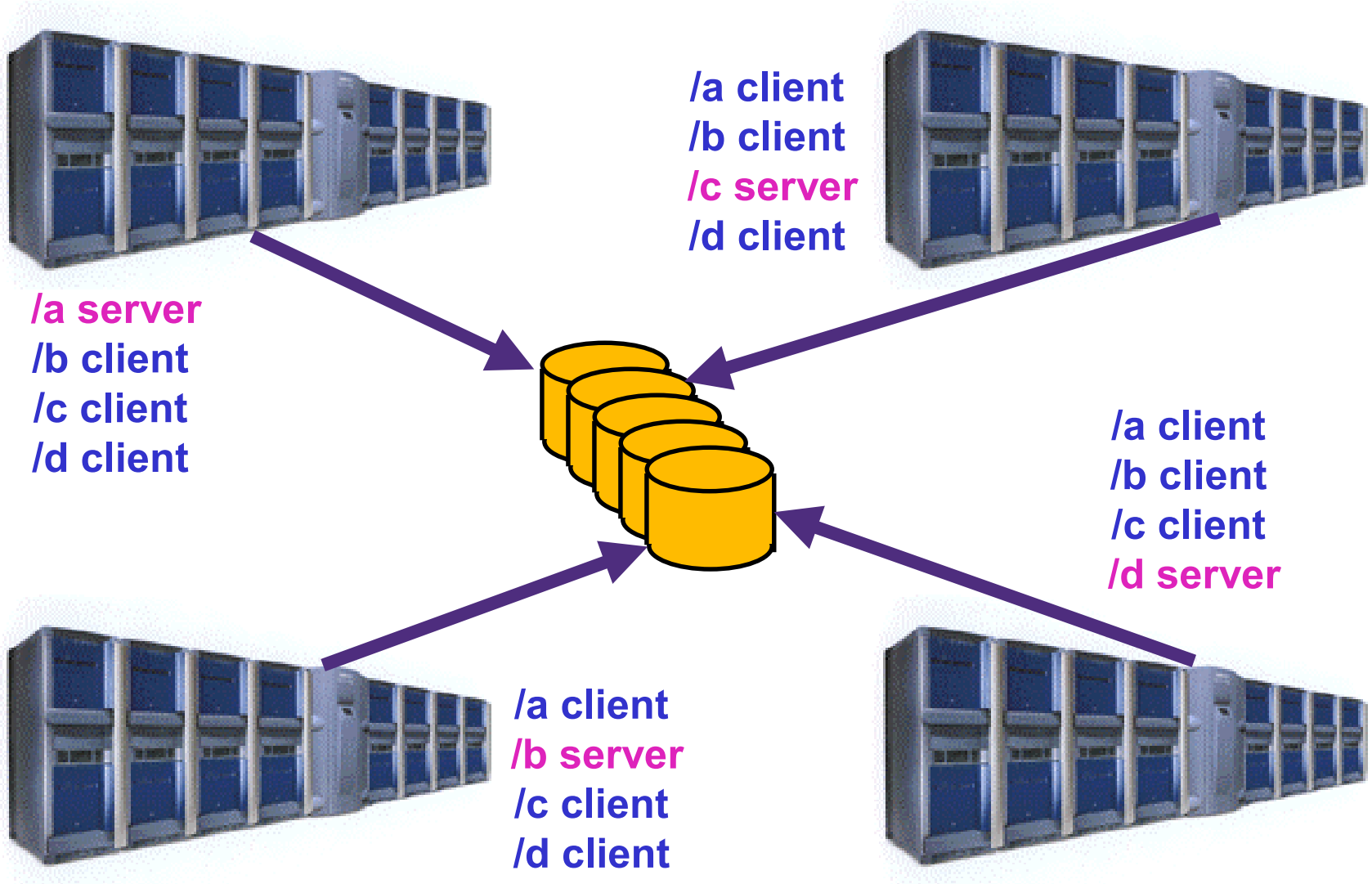

- The CXFS application programmer interface (API) is POSIX compliant
  - Fully coherent buffering, as if all processes were on an single SMP
    - Writes flush cashes on other nodes
  - Compliant with POSIX file system calls
    - Including advisory record locking
- No special record-locking libraries required
  - For example: NFS supplies a separate non-POSIX record-locking library, which is not needed with CXFS.

# CXFS Scalability

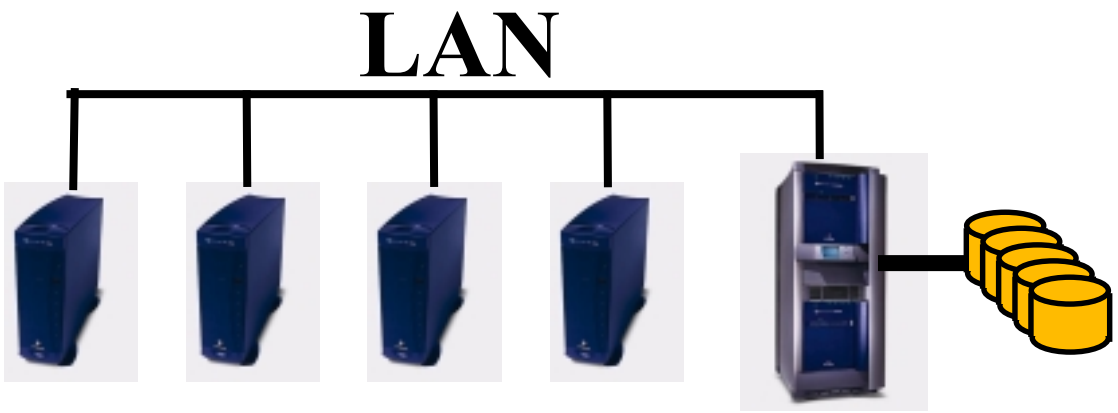
sgi

- Supports up to 64 clients or servers per cluster
  - IRIX 6.5.6 supports 8 clients
- Multiple metadata servers can exist in a cluster
  - One per file system
- Files accessed exclusively locally on CXFS metadata server see local XFS metadata performance

# Scalable: Distribute Server Load

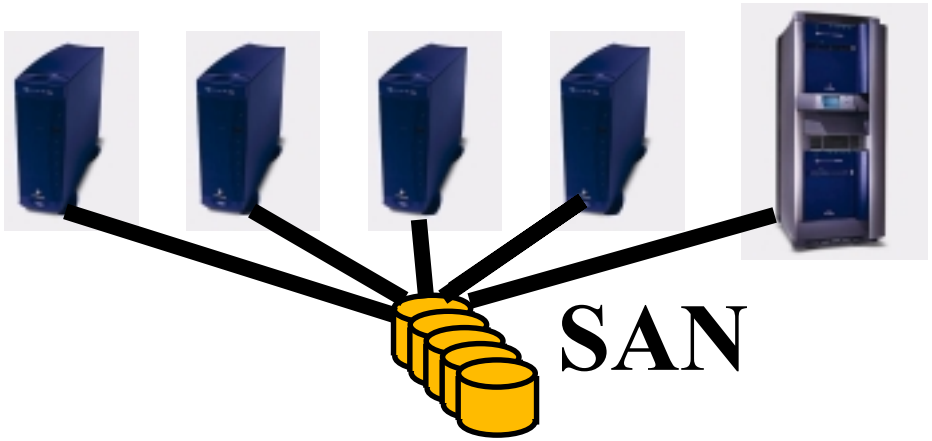


# Comparing LANs and SANs




**LAN:** Data path through server (Bottleneck, Single point of failure)

**SAN:** Data path direct to disk (Resilient scalable performance)



# CXFS vs. NFS

sgi

- *Is CXFS an alternative to NFS ?*
  - No, doesn't scale to 100s and requires direct connections to disk
  - Yes, provides high performance transparent file sharing for HPC, digital media and other markets requiring high performance

# CXFS networks

- Besides the storage area network CXFS uses the following networks
  - Metadata network - IP network (dedicated) for metadata and tokens
  - Membership network - IP network used for heartbeating
  - Reset network - non IP serial lines used to reset nodes

# CXFS Features (1/2)


- Supports guaranteed-rate IO and real-time file systems
  - for real-time and digital media applications
  - NOT on IRIX 6.5.6
- Fast recovery times: No fsck
- Avoids unnecessary writes by delaying writes as long as possible
- Contiguous allocation of disk space to avoid fragmentation
- 9 petabyte fillesystem size
  - If historical trends continue, will last 60+ years

# CXFS Features (2/2)

- Fast directory searches
- Sparse file support
  - Holes allowed in files for large direct-access addressing
- DMAPI for hierarchical file systems (HFS)
  - Interfaces to SGI's Data Migration Facility (DMF)
    and third-party HSMs: Veritas, FileServ, ADSM
  - Available on IRIX 6.5.8

# Optimal CXFS Performance

sgi

- When there are many:
  - reads from and writes to a file that is opened by only one process
  - Reads from and writes to a file where all processes with that file open reside on the same host
  - Reads from a file where multiple processes on multiple hosts read the same file
  - Reads from and writes to a file using direct-access I/O for multiple processes on multiple hosts

# Not Optimal CXFS Performance


- Multiple processes on multiple hosts that are reading and writing the same file using buffered I/O
  - direct-access I/O (e.g. databases) are okay
- When there will be many metadata operations such as:
  - Opening and closing files
  - Changing file sizes (usually extending a file)
  - Creating and deleting files
  - Searching directories

# Metadata Tokens

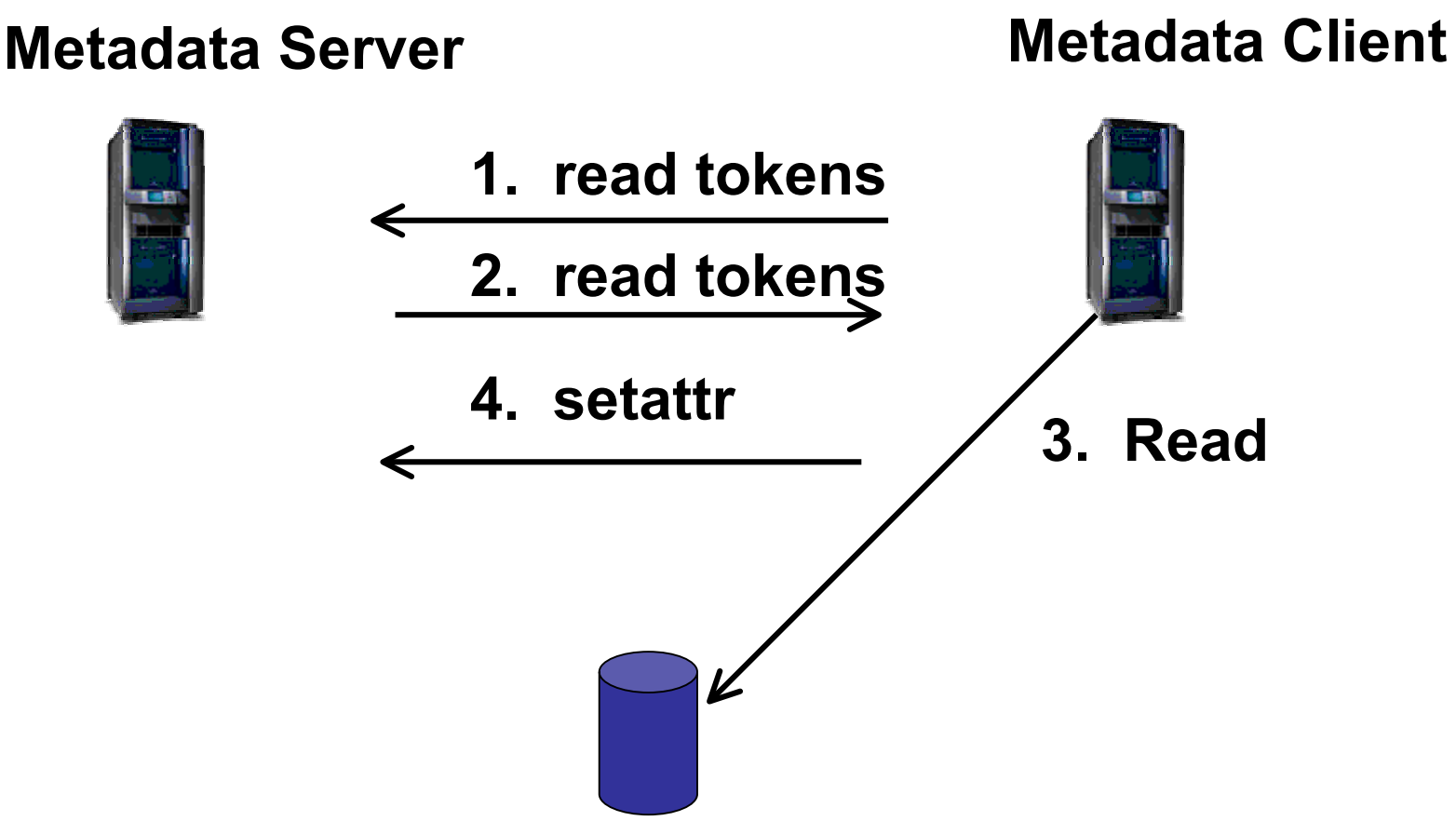
sgi

- ***Token Protected Shared Data***
  - **Read - multiple access token**
  - **Write - exclusive access token**
  - **Shared Write - multiple access token**
  - **Existence - multiple access token**

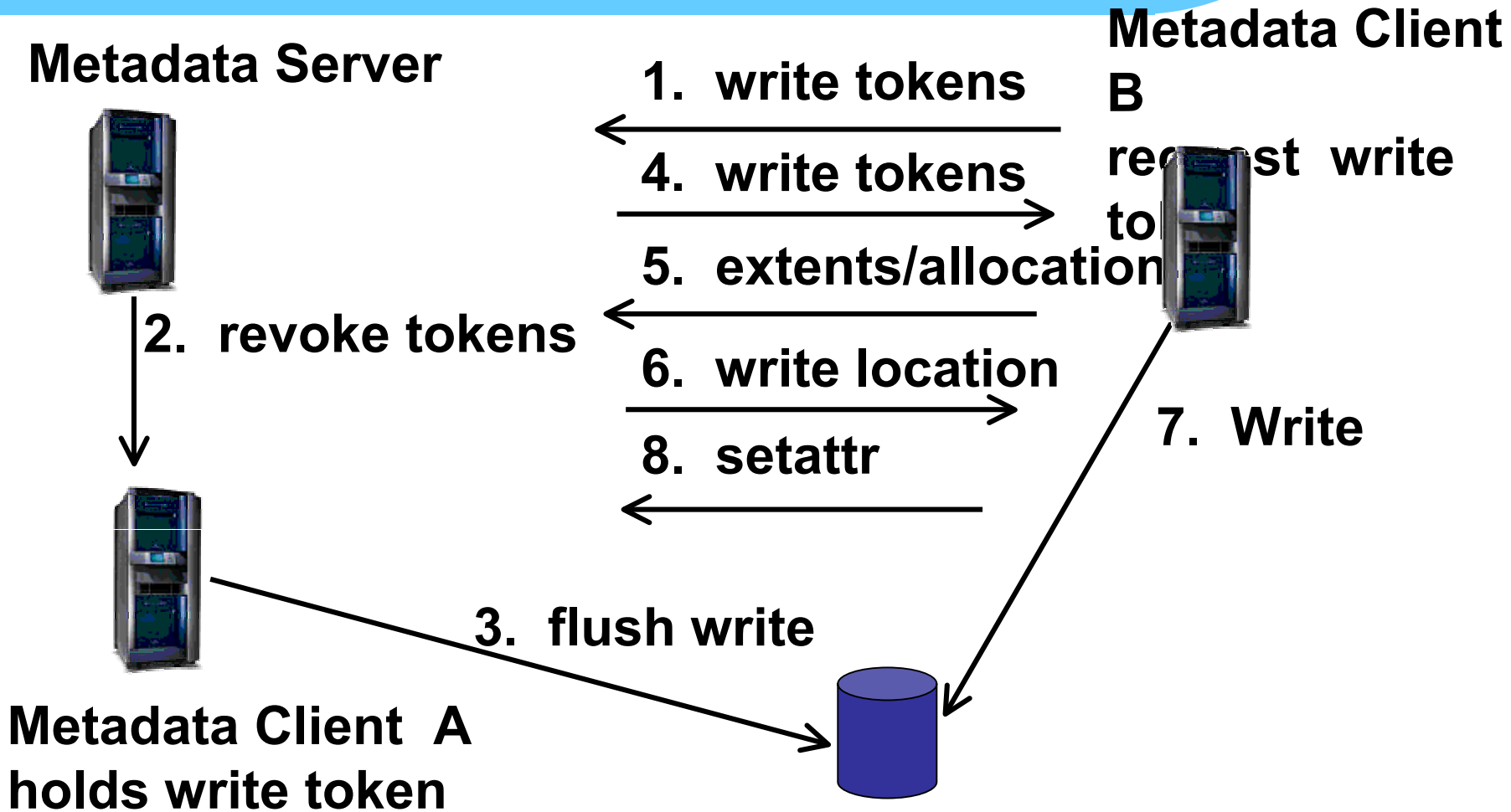
- ***Customized RPC mechanisms maximize communication speed among clients and the metadata server***

- ***In general clients will cache tokens as long as possible***

# Read Metadata Flow



# Write Metadata Flow



# Examples from our past


- Proprietary networking at Cray
- NFS on UNICOS and UNICOS/mK
- DFS fpr UNICOS, UNICOS/mK and IRIX
- SFS for UNICOS and UNICOS/mK
- Cray's Gigaring (aka SCI)

# NFS

sgi

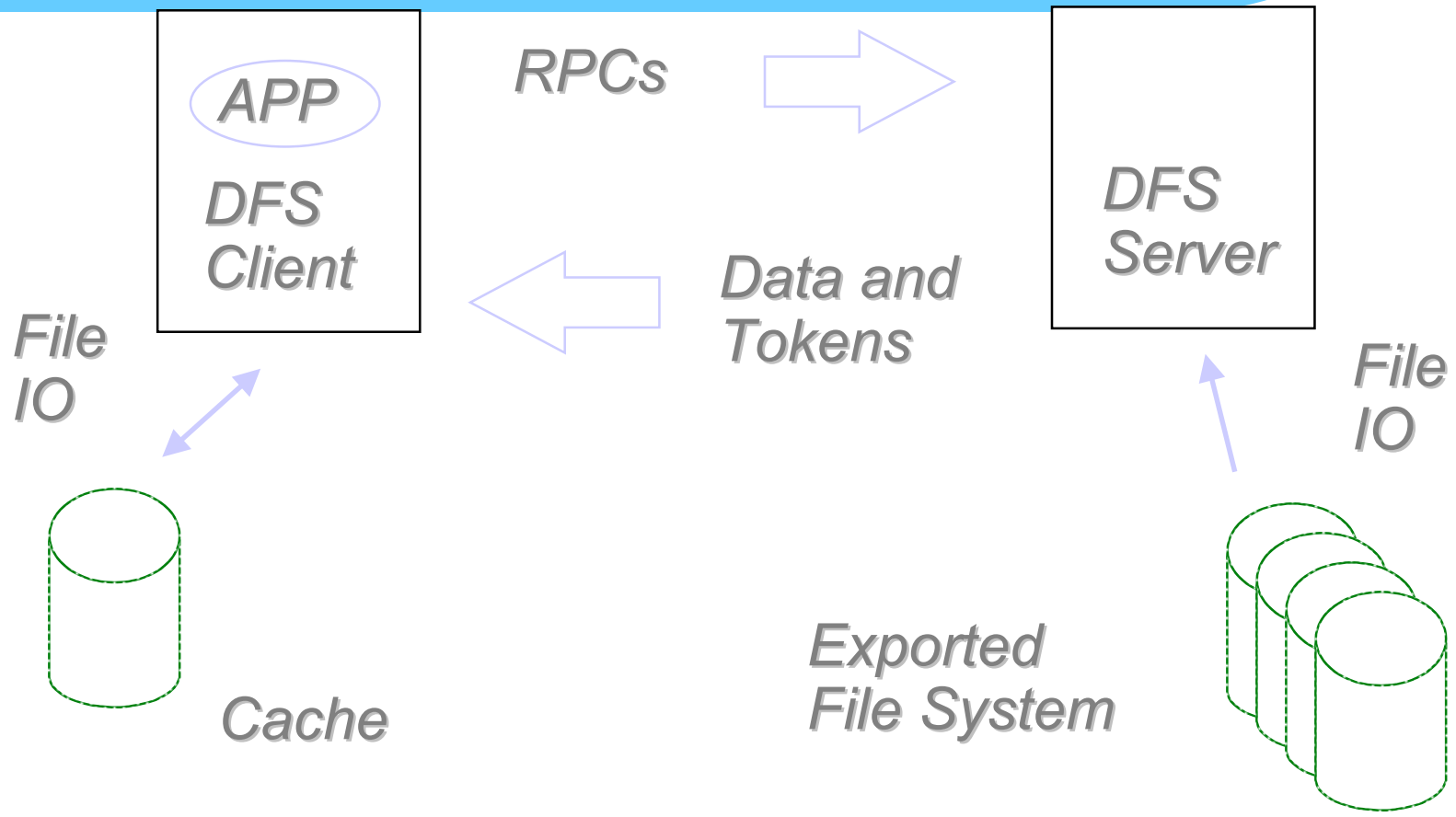
- Defacto UNIX file sharing protocol
- Performance limited by network and protocol
  - Best performance target is FTP
- Cray's extension used bigger blocks
- SGI's extension (BDS) uses bigger blcoks and changes the protocol
- NFS V3 introduces client caching

# DFS

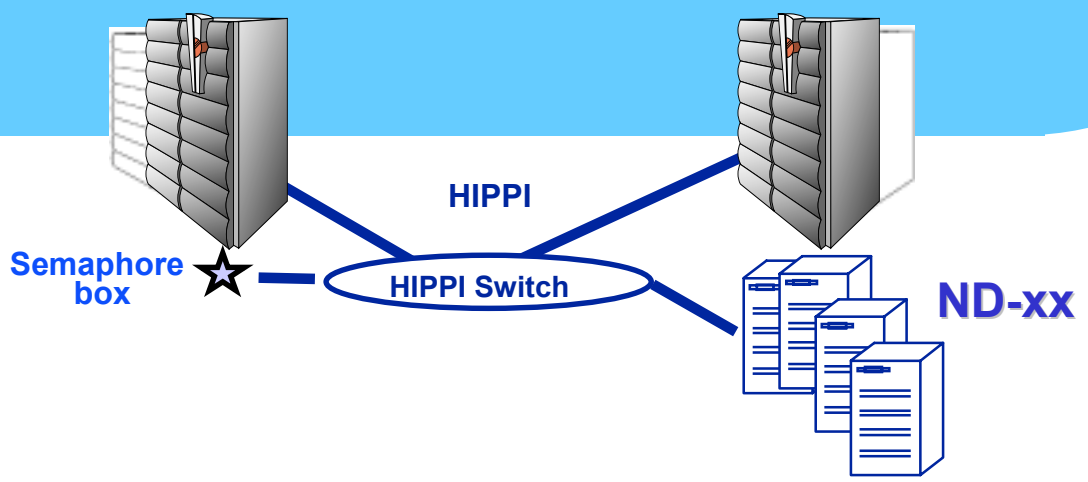
sgi

- Early adaptors of DCE/DFS were supercomputing customers
- DFS employed caching and streaming protocols for performance
- Cache sizes extended to 32GB for supercomputing customers
- Uses tokens to protect cached data
- Once cache fills then performance is limited to network speed

# DFS Data Flow

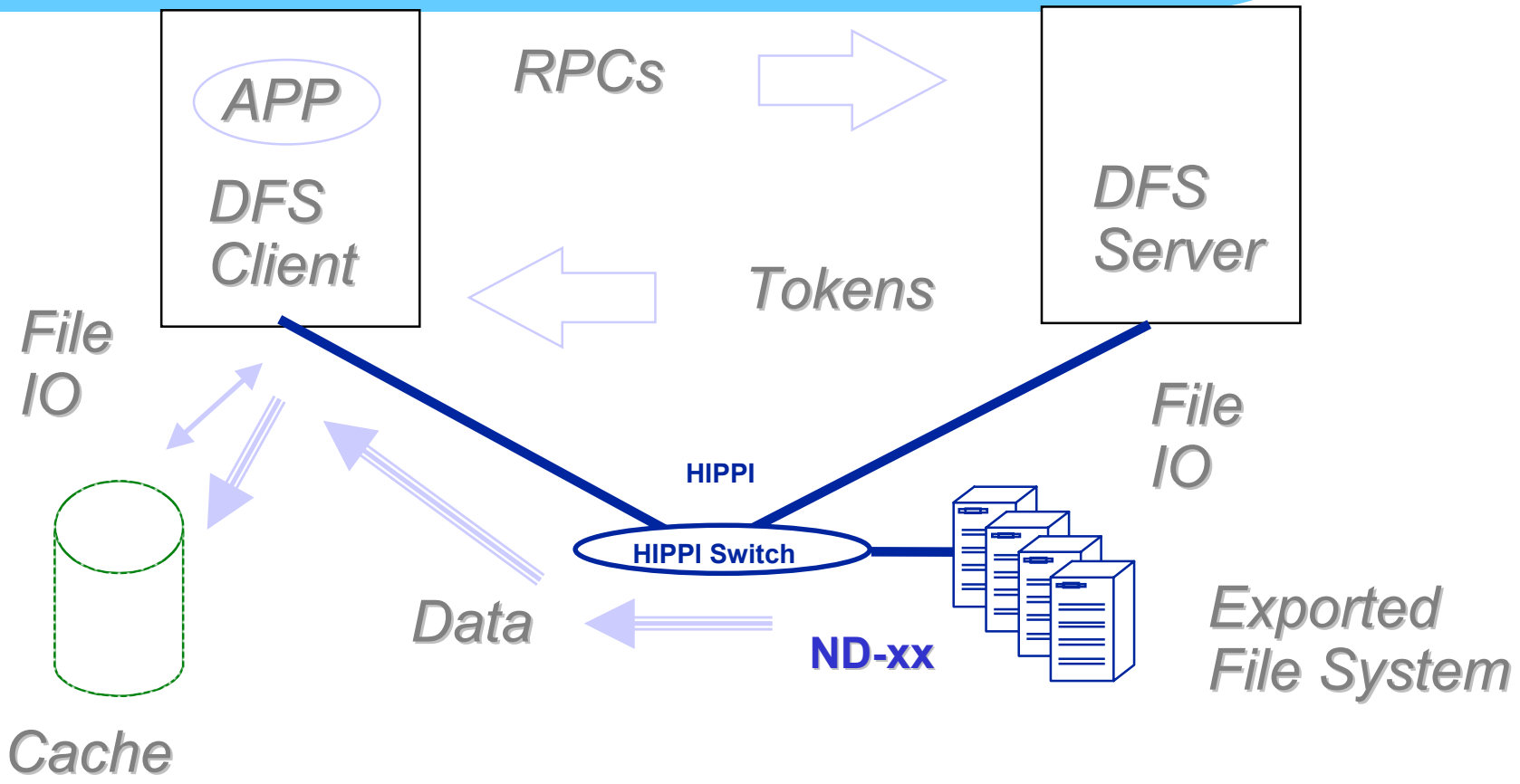


# SFS Structure



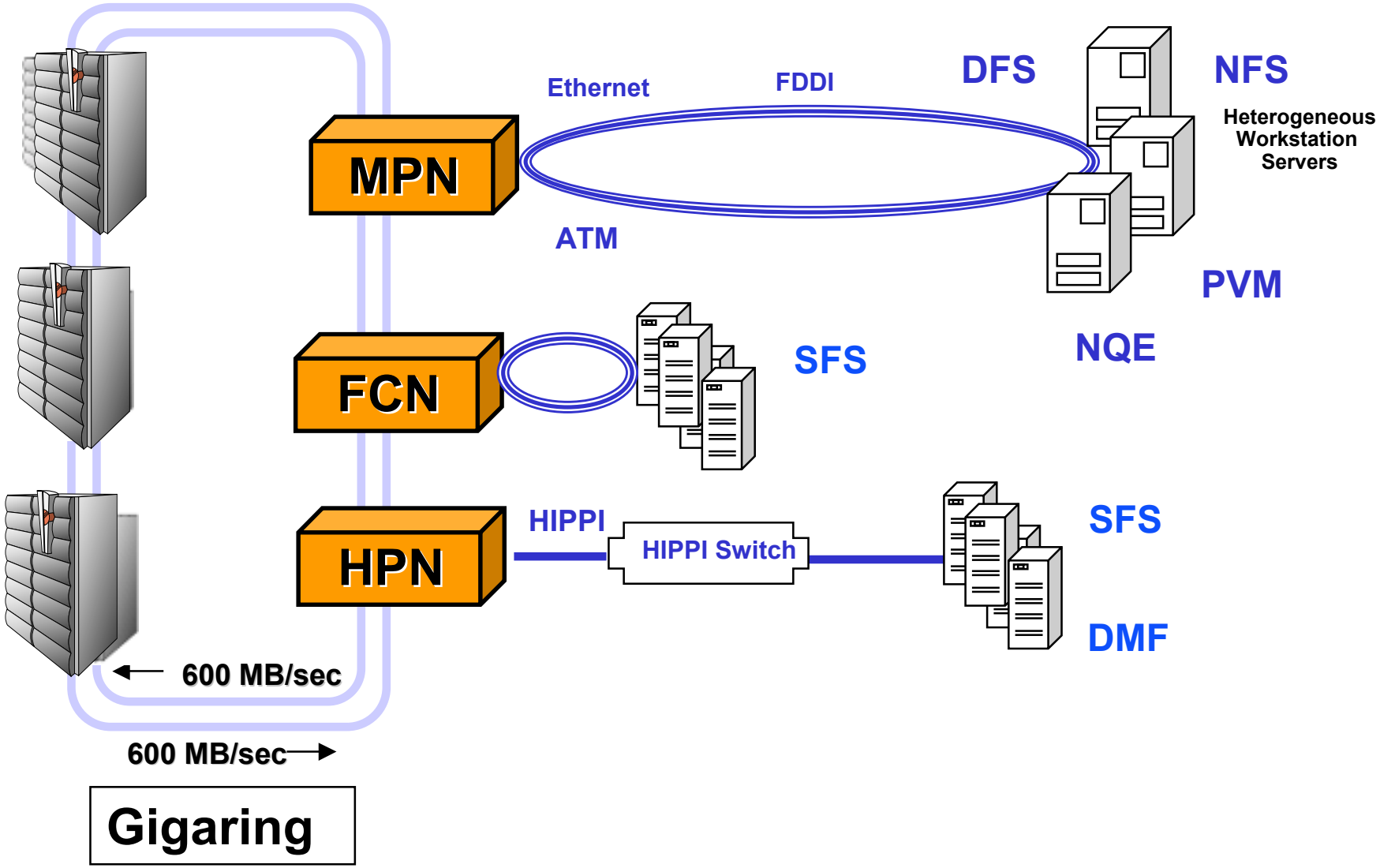

- SFS is a locking mechanism for the NC1 filesystem
  - –Read, write, and exclusive open locks
- Nodes do not directly communicate - not client/server
- Shared media restricted ND arrays
- Nodes protect metadata via semaphores
  - –Metadata is not cached
- Data flows directly to node from ND-xx
- System heartbeat for resiliency

# DFS Uses SFS To Access Data

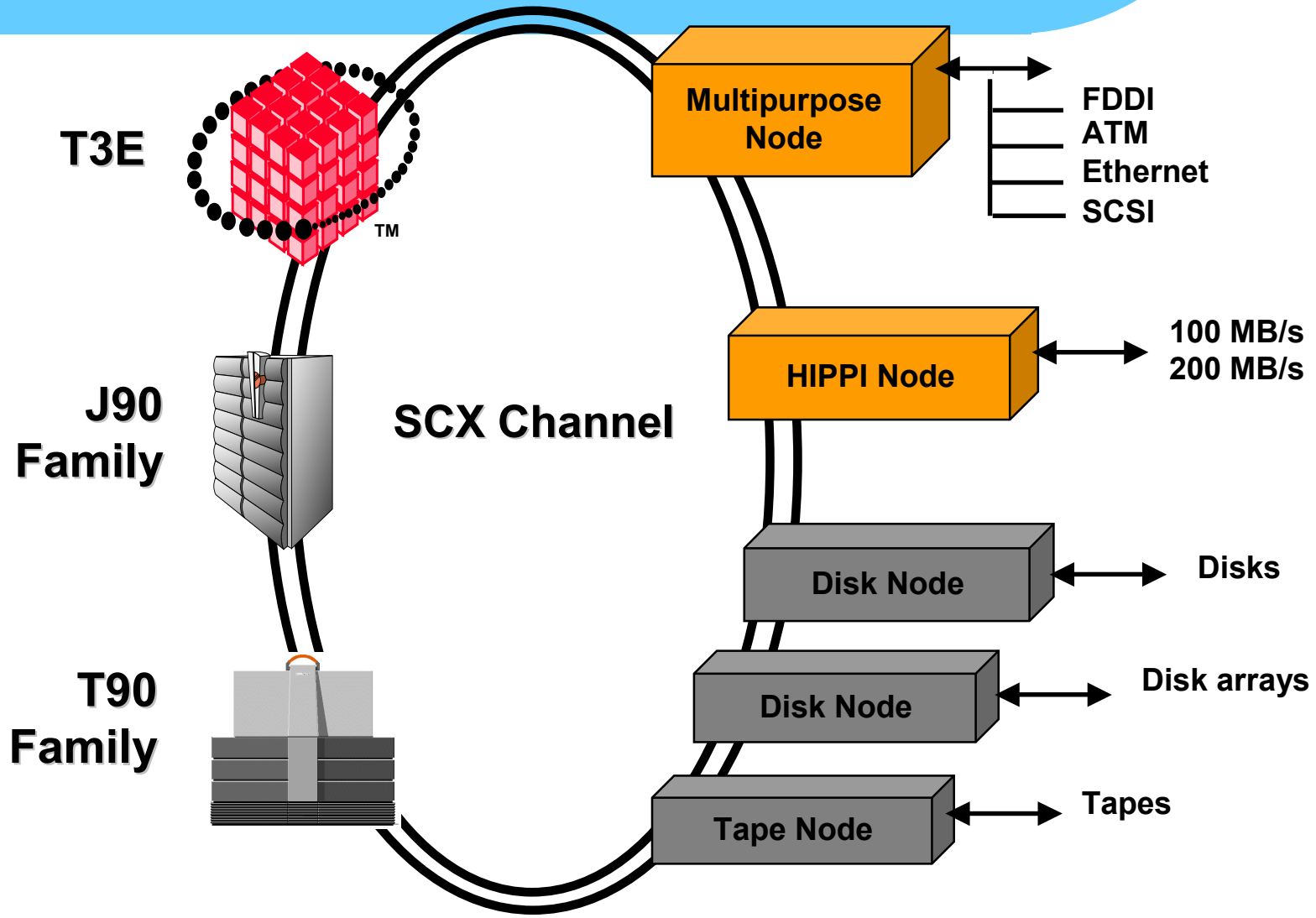



- Data Flows directly to client

# Cray's SuperCluster



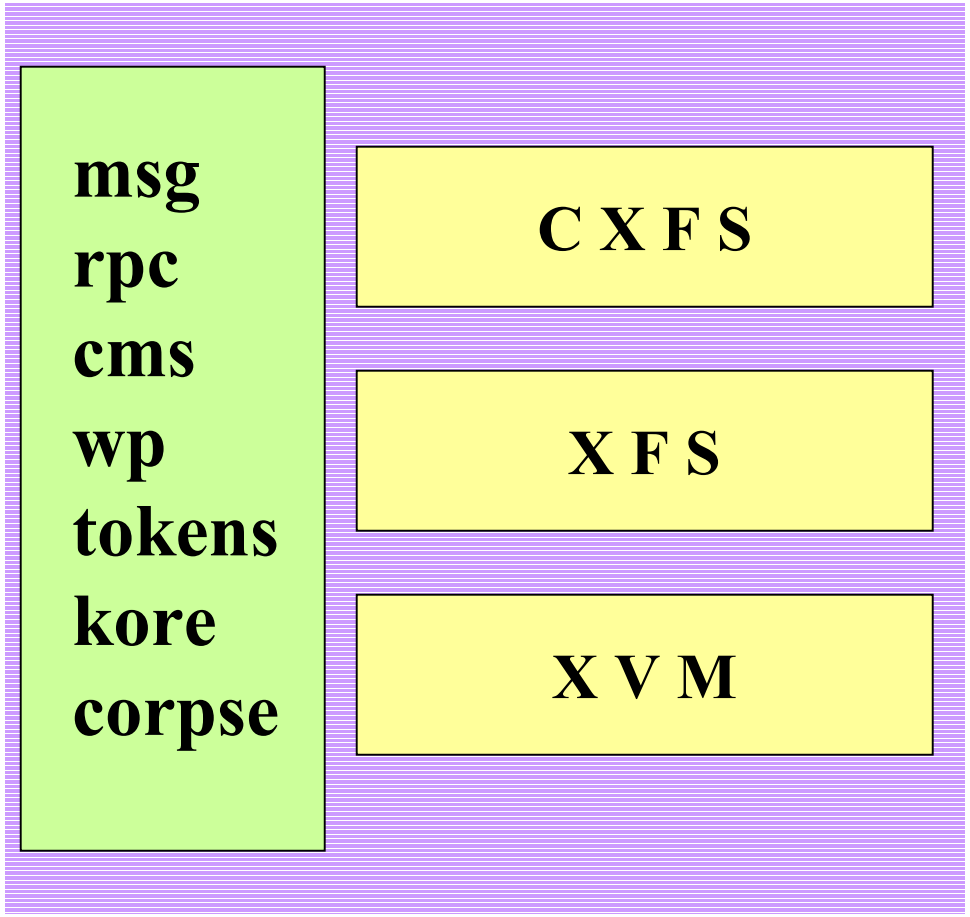
# Cray's Gigaring (aka SCI)



# CXFS design points


- Mostly read only or single writer
- Databases
- Single log per file system
- Transparency is the price of entry
- Maintain XFS features and characteristics

# Cluster Infrastructure Components



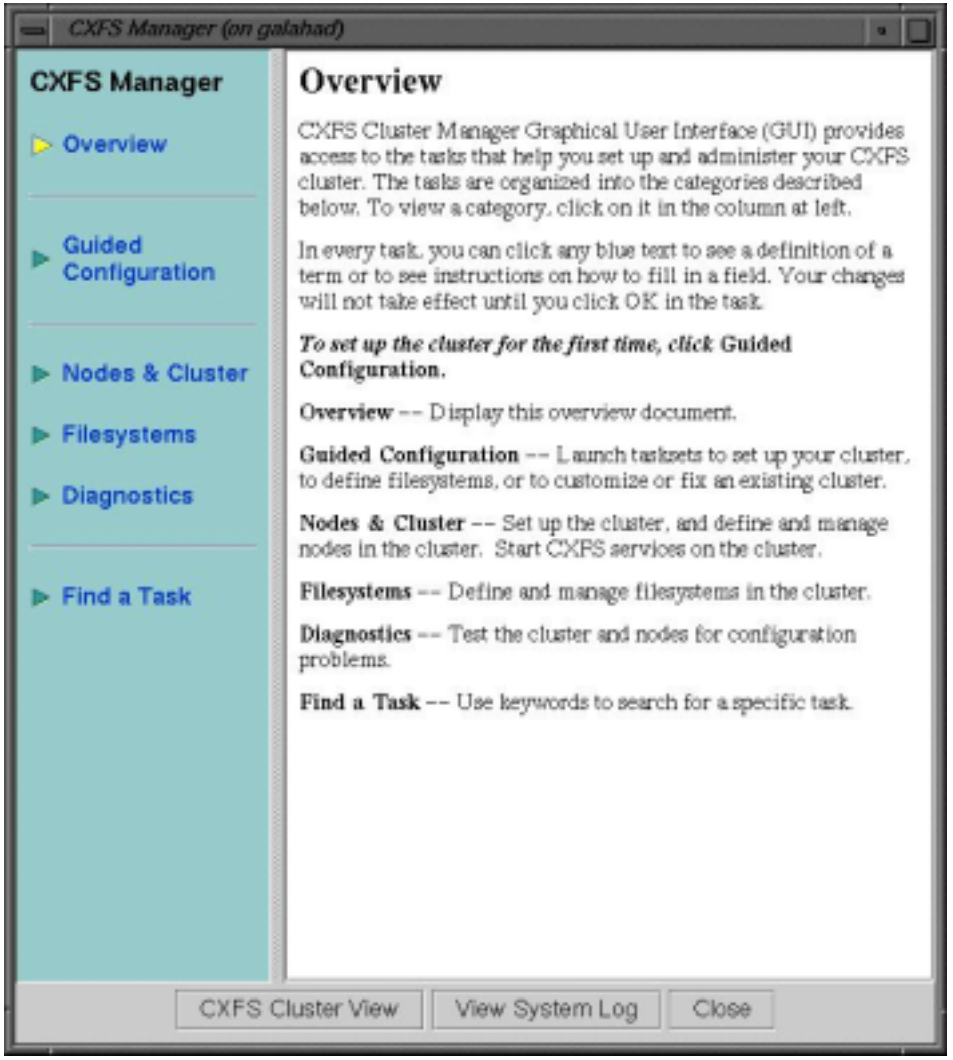

RPC and MSG
Membership services
White pages (WP)
Token module
Object relocation (KORE)
Recovery (CORPSE)

# Cluster Infrastructure Components

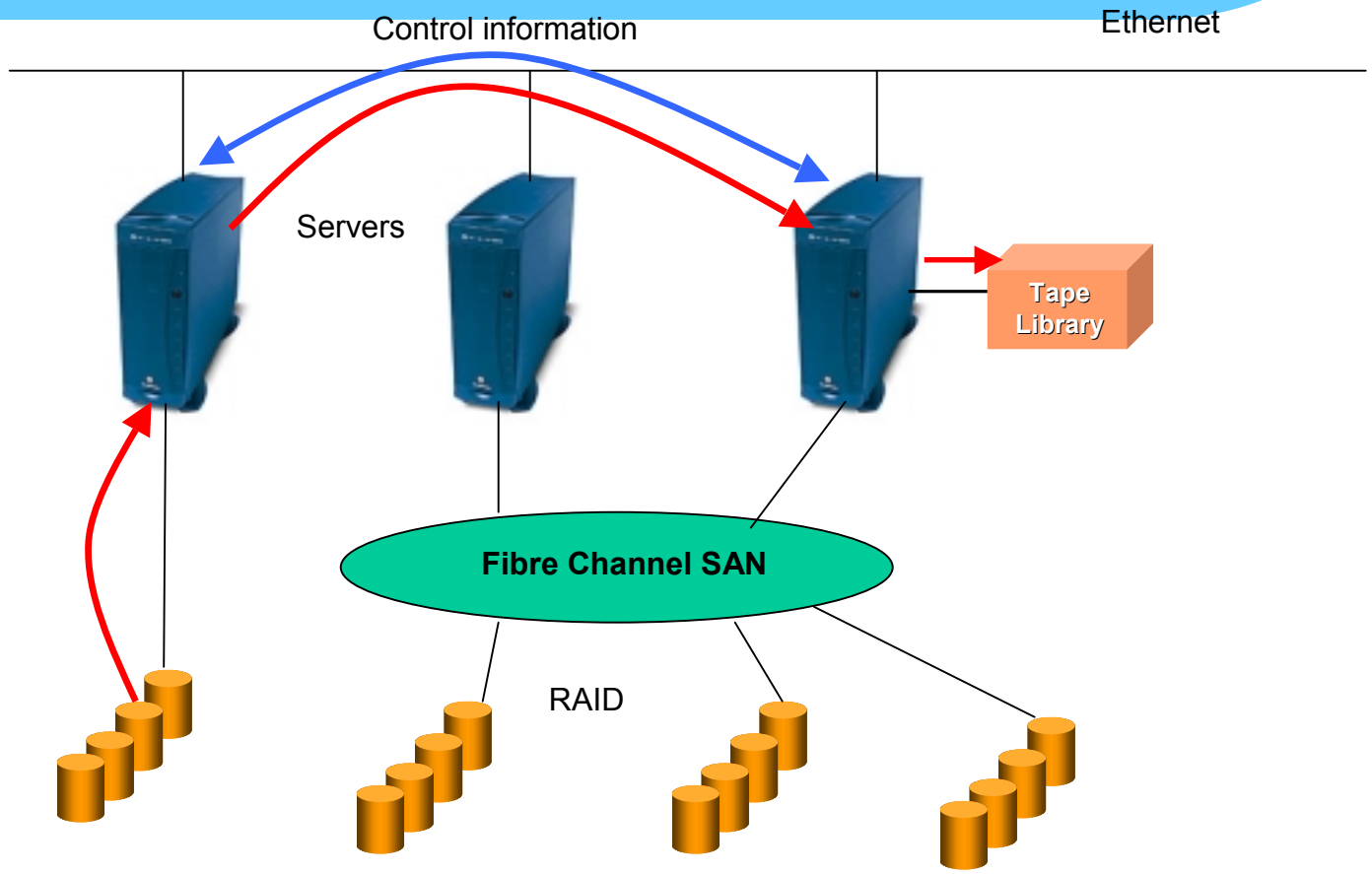
sgi

- RPC and MSG
- Membership services (CMS)
- White pages (WP)
- Token module
- Object relocation (KORE)
- Recovery (CORPSE)

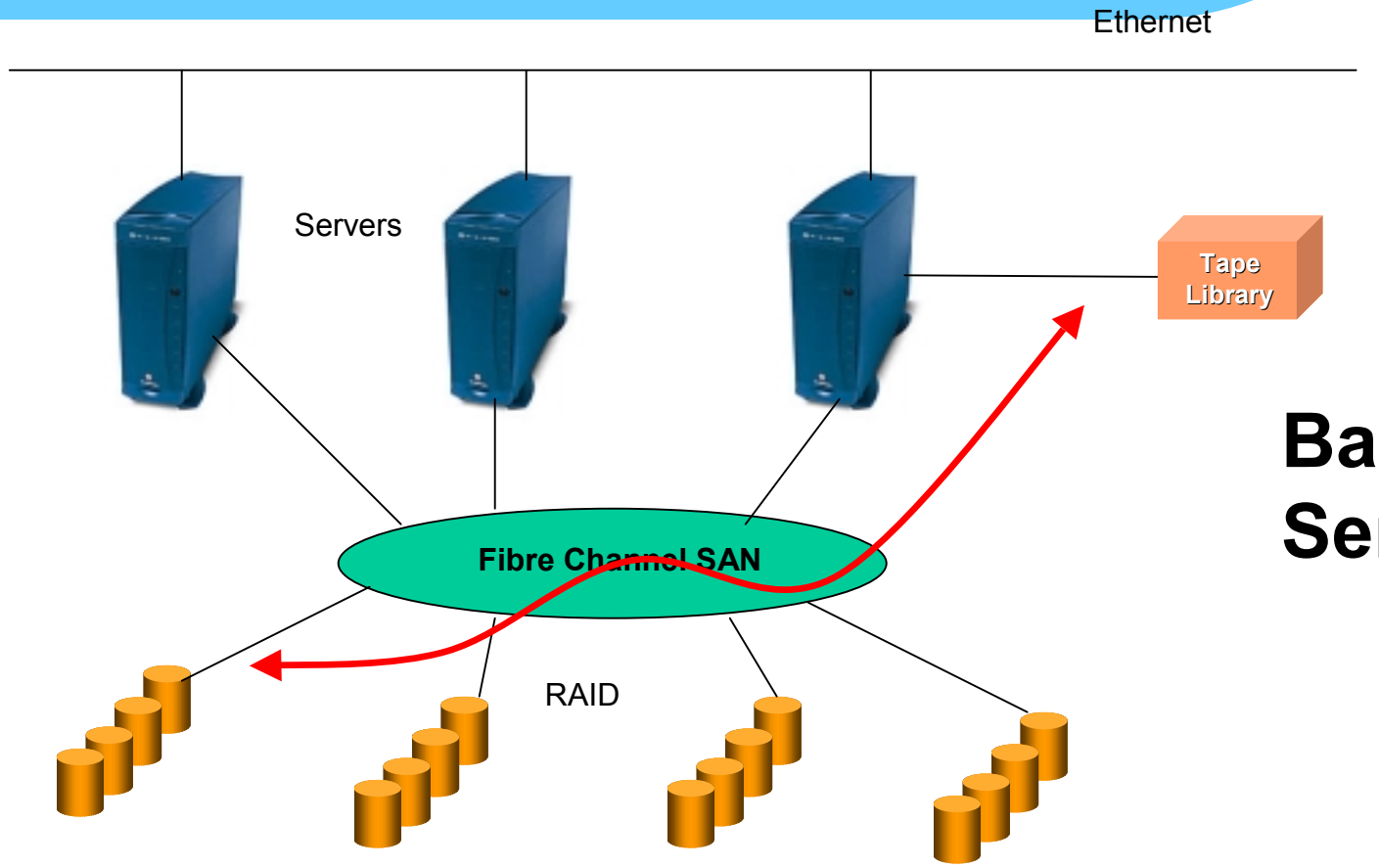
# CXFS Cluster Manager



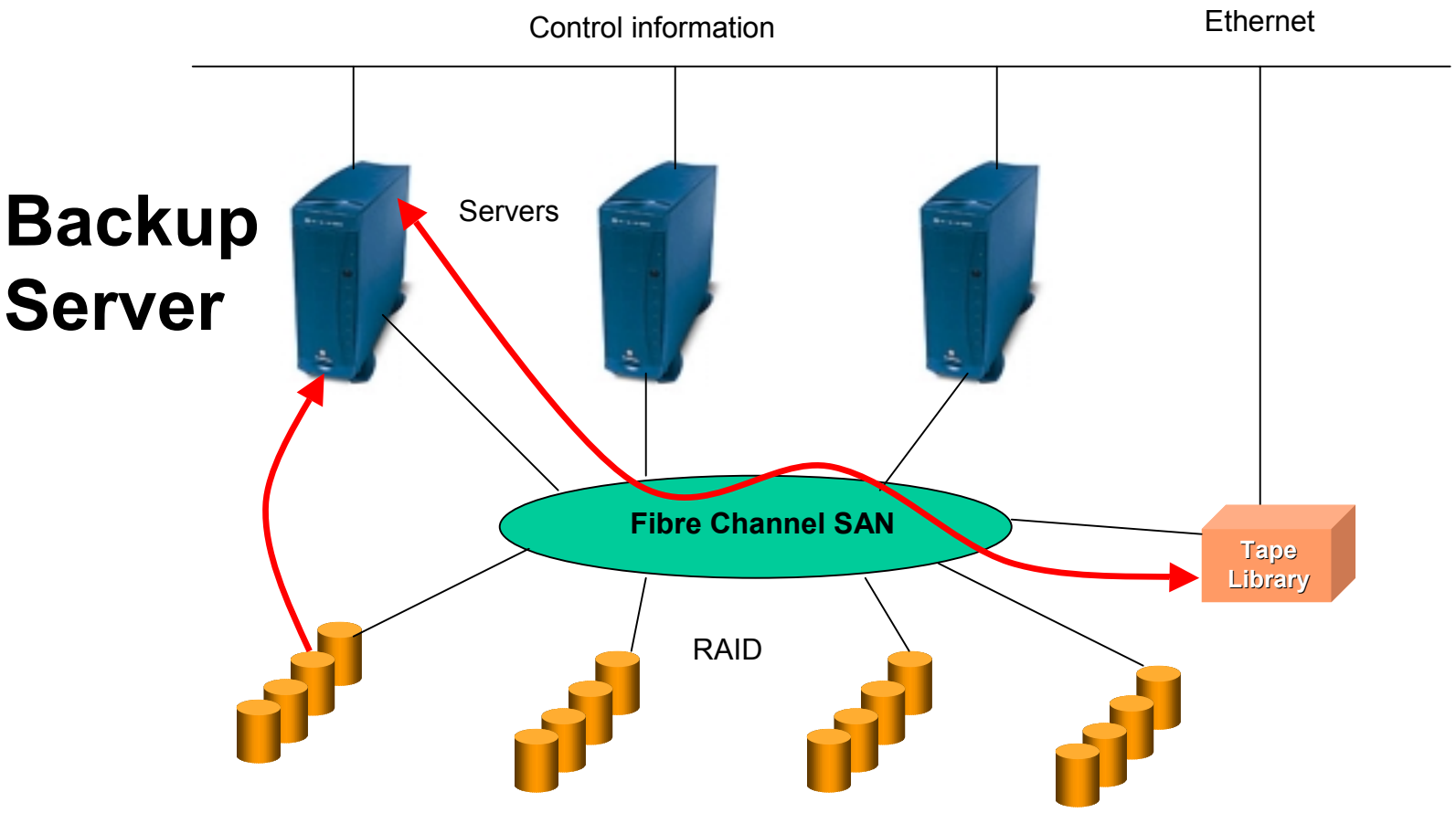
# Regular Backup



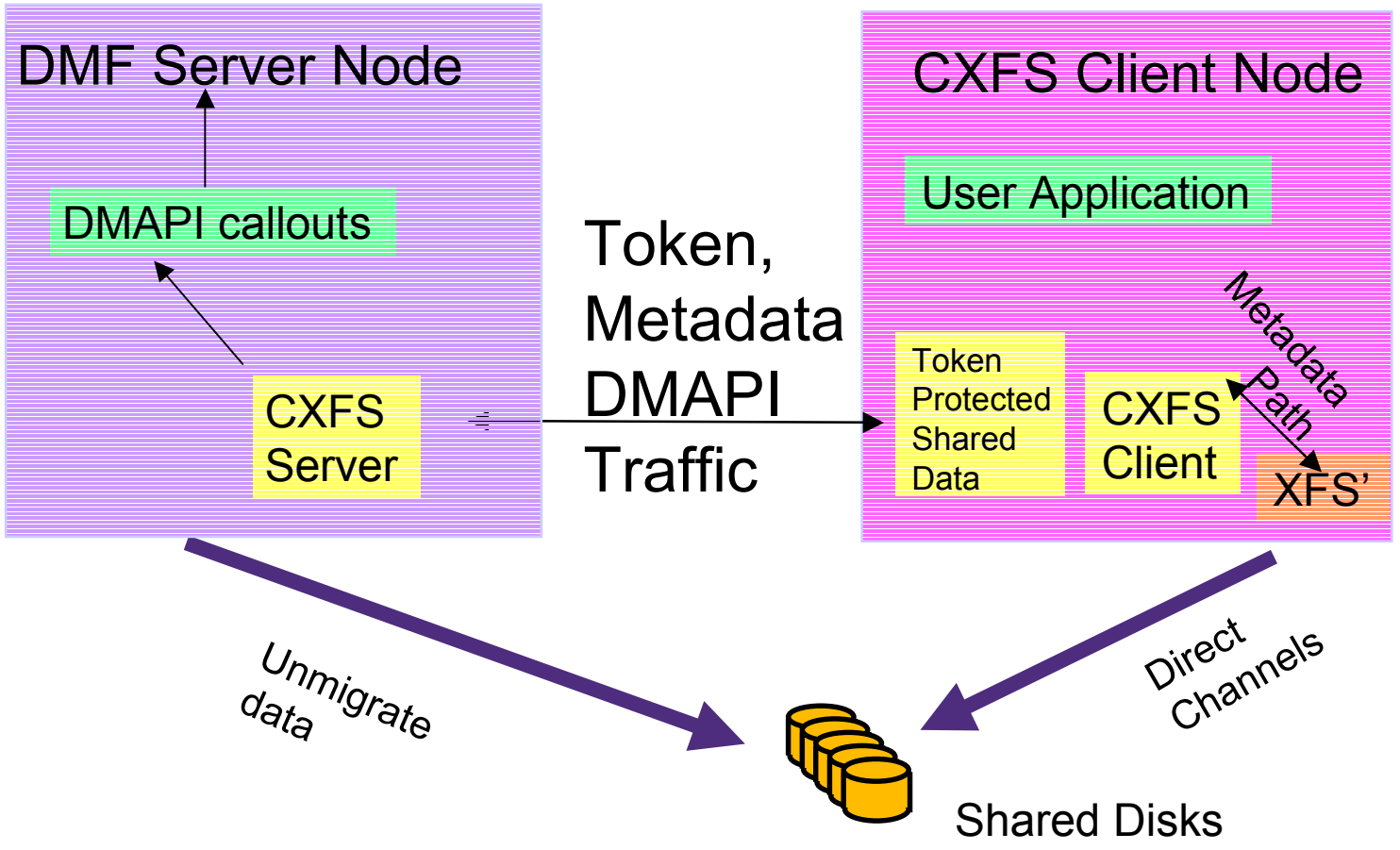
# LAN-free Backup



# LAN-free Fiber Channel tape Backup



# CXFS DMAPI Data Flow



# Reset hardware and CXFS

sgi

- Same as Failsafe
- Used to stop systems that are not in quorum
- CXFS membership can stop IO to cluster nodes in 6.5.7 (patch 3873)
- Reset hardware provides extra level of reliability. Useful for the rare times when the kernel is stuck AND IO is still possible.

# CXFS Memberships and quorums

sgi

- CXFS product memberships :
- kernel based - used by CXFS and XVM
- user space - used for configuration
- Membership must have a majority of nodes to form a quorum.
- No quorum will stop cluster
- Weights are used to break 50-50 ties

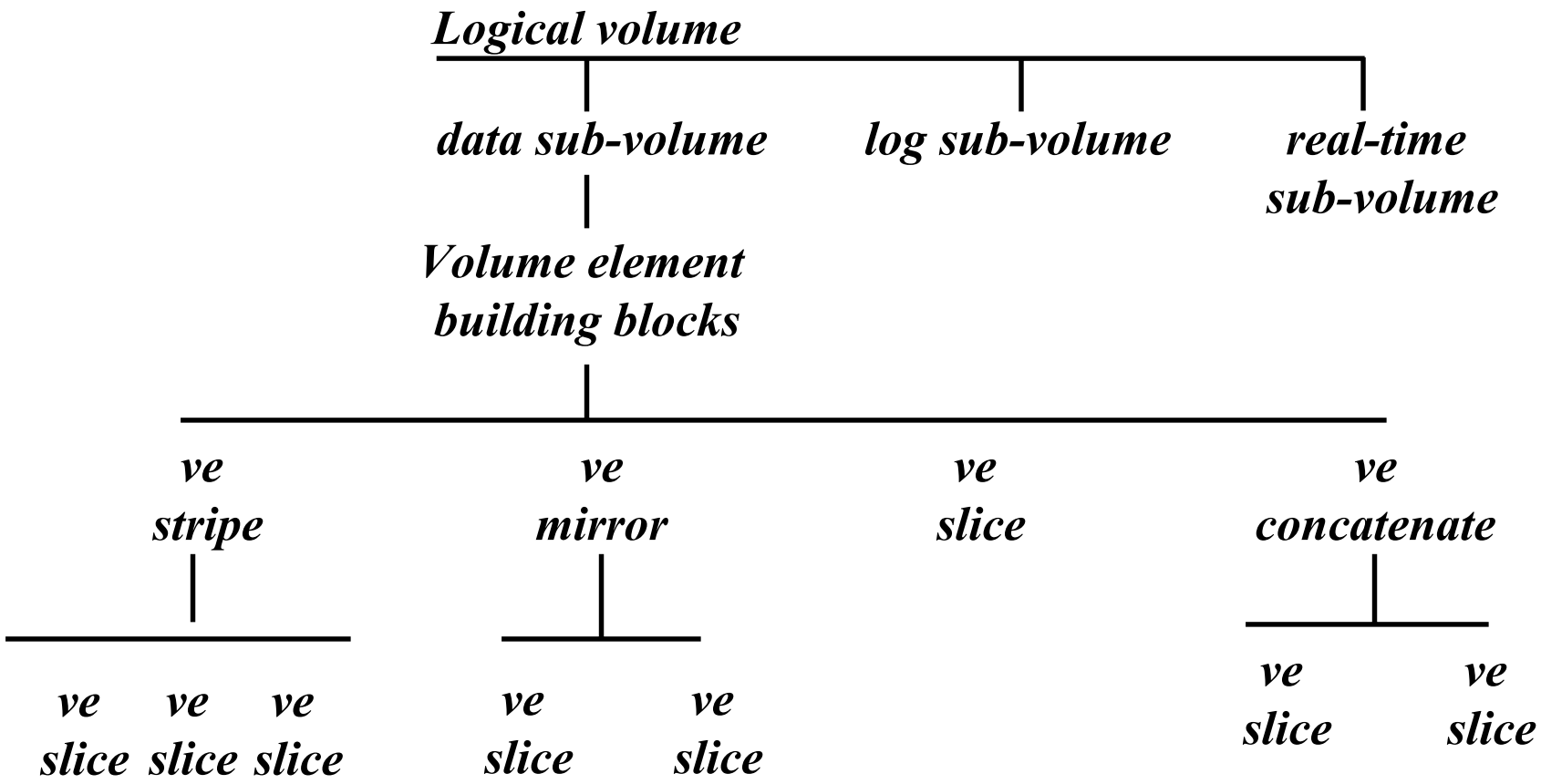
# Heterogeneous CXFS

- IRIX servers and clients in second half of 1999
  - IRIX-XFS/XVM performance and file-system features
- Clients for Windows NT, Linux and other major UNIX system in 2001
  - Performance and features may be limited by particular OS interfaces
  - Evaluating other OSes


# Volume Management

# XVM

# XVM: Volume Management



# XVM - Volume Management


- Striping, mirroring, and concatenation of volume elements
  - Flexible combinations of mirroring and striping
- Thousands of disks:  E,g,. 64K stripe width
  - Practically unlimited
- Self identifying volumes
- Subvolumes separate data, log, and real-time information
- On-line configuration changes

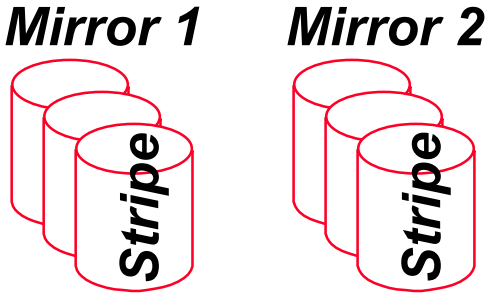
# XVM Features


- Next generation of XLV
- Unlimited stripes
- Unlimited number of logical partitions
- Can hot plug disks
- Fast mirror revives (uses region logging)
- Shared/Private volumes
- Root and swap volumes (can’t be shared)

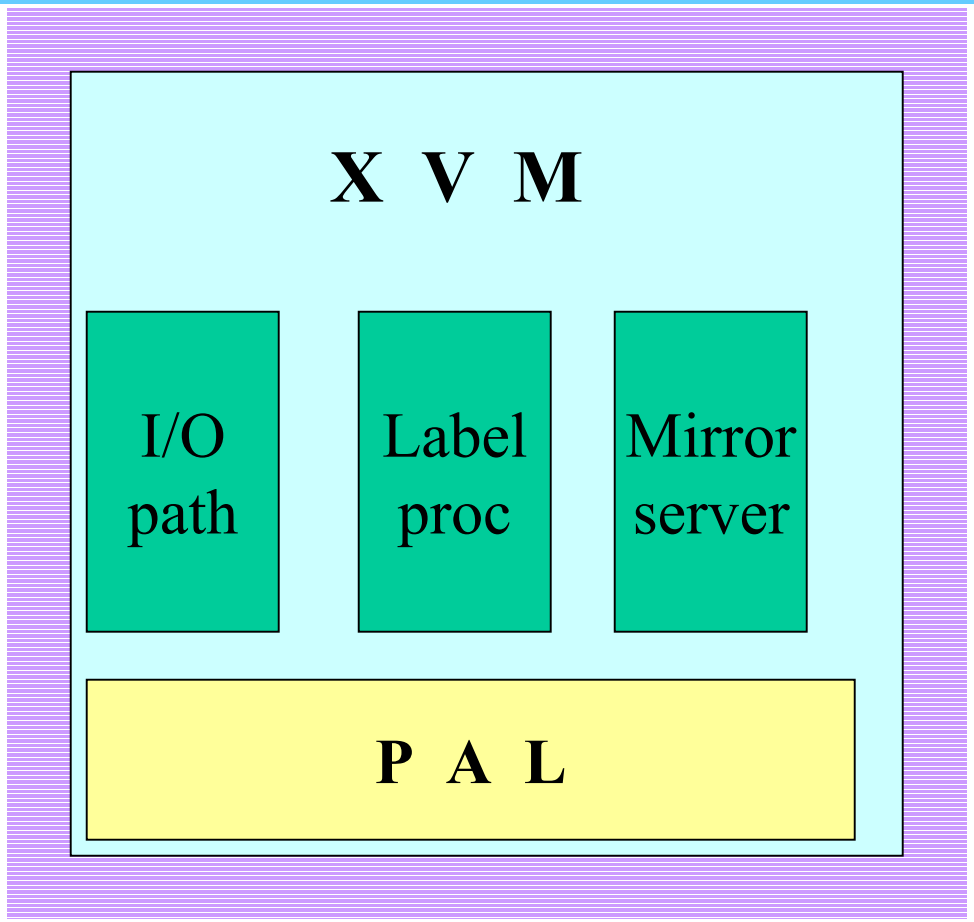
# XVM Flexible Combinations

## Striped Mirrors


## Mirrored Stripes



# X V M Components




Physical Abstraction Layer
Label Processing
IO breakout
Mirror server

# Supported Hardware

- Origin2000 or Origin200 or Octane platforms
- One of the following:
  - QLogic HBAs
  - Brocade switches or Emulex hubs (not both)
  - All supported disks work with Emulex
  - Disks supported with Brocade
    - Clarion RAID or Ciprico RAID
    - Max 10 per loop and 120 per fabric
  - Adaptec XIO HBA on Emulex hubs with CLARiiON or Ciprico RAID disks or JBOD

# Competition

sgi

- SUN - Parallel File System is MPI-IO
- IBM/Mercury - NT server, control via NFS
- limited scaling, no backup server
- HP/Transoft - volume locking
- DataDirect - customers ?, NFS based
- ADIC/Mountain Gate - good digital apps
- Veritas - have CVxVM, file system due out

..

# CXFS configuration database

sgi

- cdb is from Failsafe
- Used to define the pool of nodes, nodes in the cluster and file systems to share
- Commands entered from GUI update the database
- Other nodes detect the database has changed and try to execute the required commands locally


sgi
TM