Instrumental
www.instrumental.com

Trials and Tribulations of a Shared File System in a Heterogeneous Environment

Henry Newman and Nathan Schumann

Instrumental, Inc, 2748 East 82nd Street, Bloomington, MN 55425-1365

Phone: +1-952-345-2822, e-mail: hsn@instrumental.com

NASA/IEEE MSST 2004
12th NASA Goddard/21st IEEE Conference on
Mass Storage Systems & Technologies
The Inn and Conference Center
University of Maryland University College
Adelphi MD USA
April 13-16, 2004

NASA

IEEE COMPUTER SOCIETY

Notice

Product names mentioned in this document may be trademarks and/or registered trademarks of their respective companies and are the property of these companies.

Agenda

- Terminology/Definitions
- Usage Considerations
- Hardware Architecture
- Components
- HSM and Shared File systems
- Software Considerations
- Current shared file systems products
- Installation of shared file systems
- Architecture HA/RAS
- Looking toward the future

Terminology/Definitions

DAS

- ✓ *Direct Attached Storage*

SAN/DASS

- ✓ *Direct Attached Shared Storage*

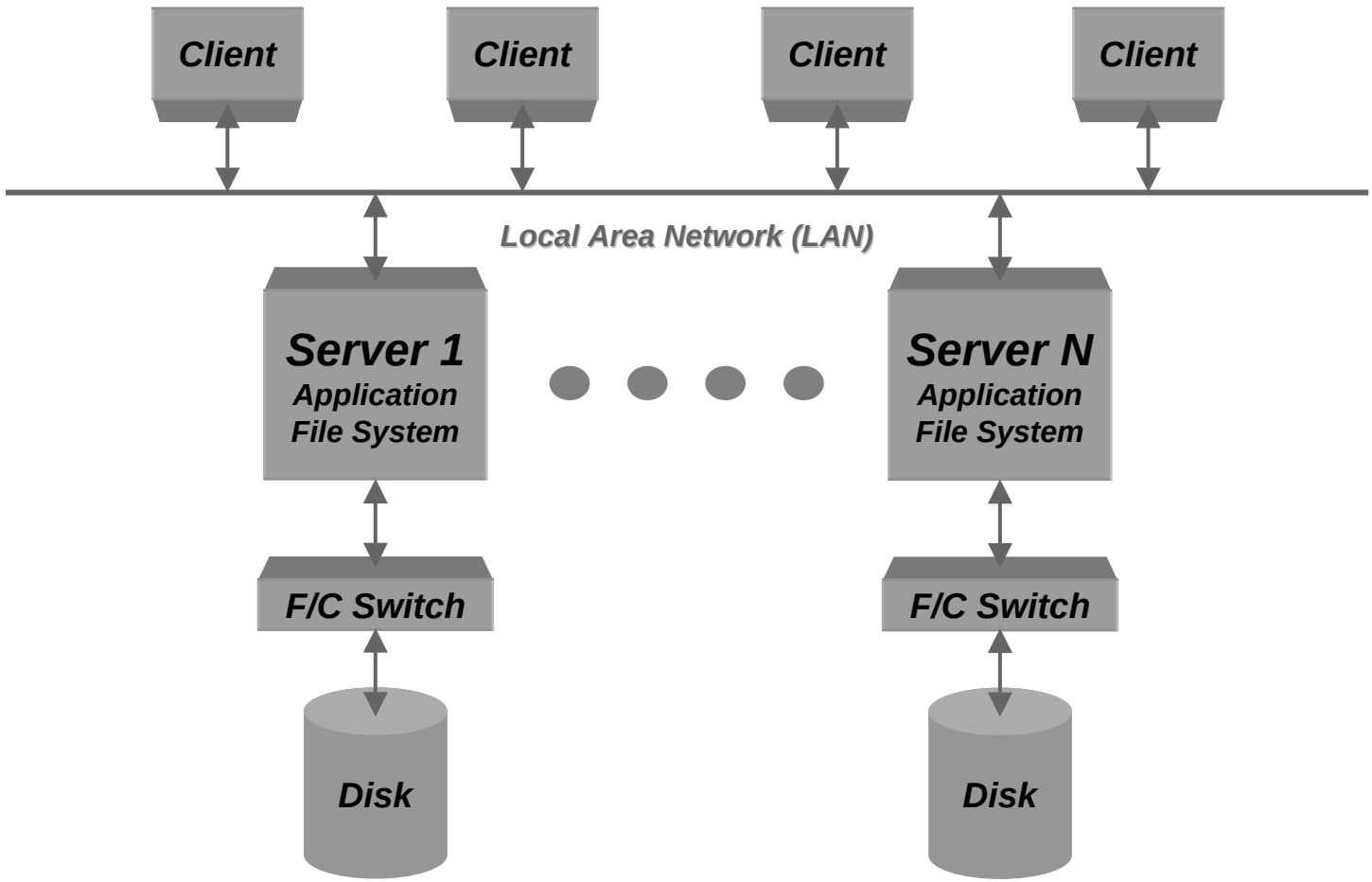
NAS

- ✓ *Network Attached Storage shared via TCP/IP*

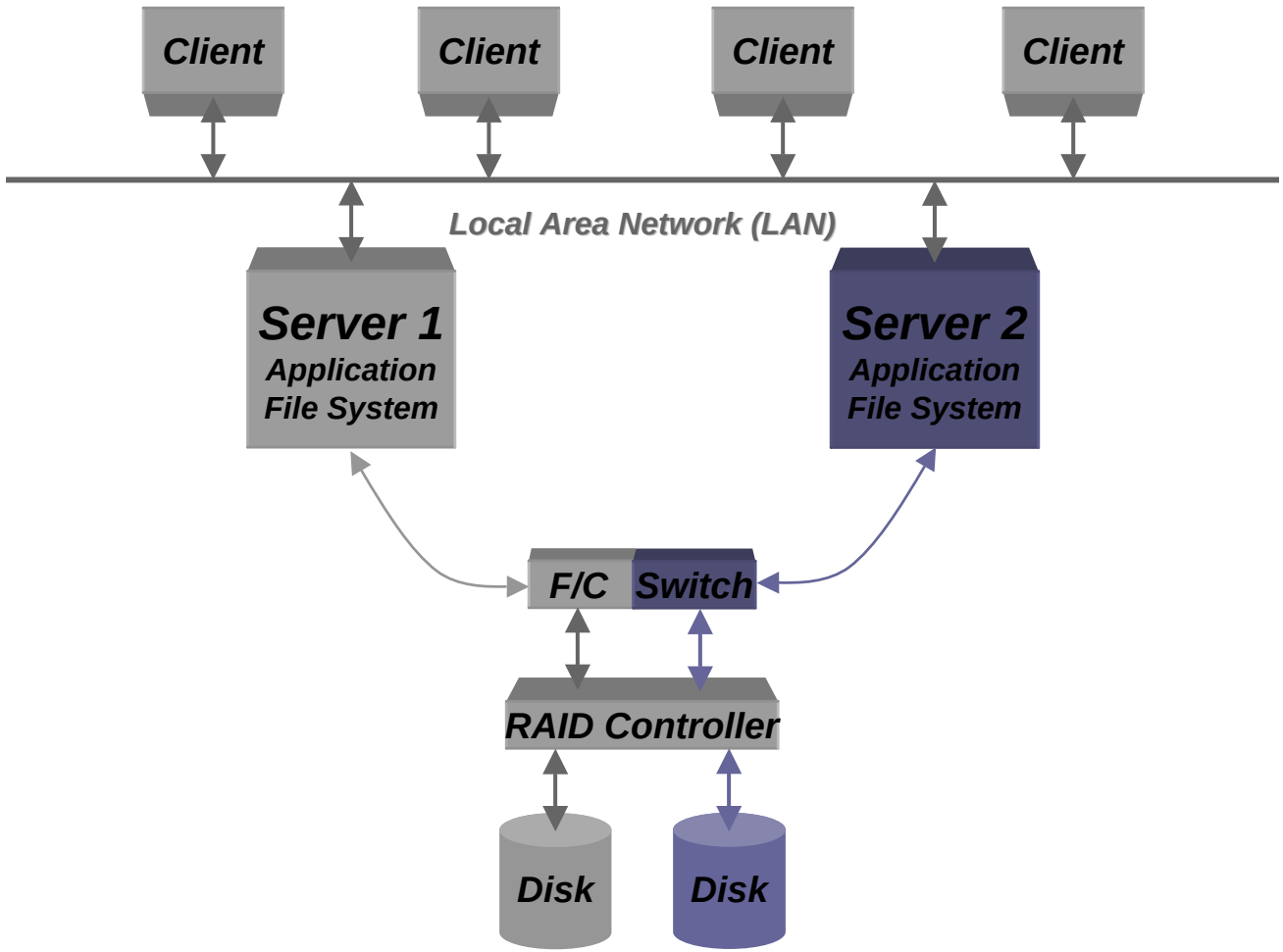
SAN Shared File System

- ✓ *File system that supports shared data between multiple servers*

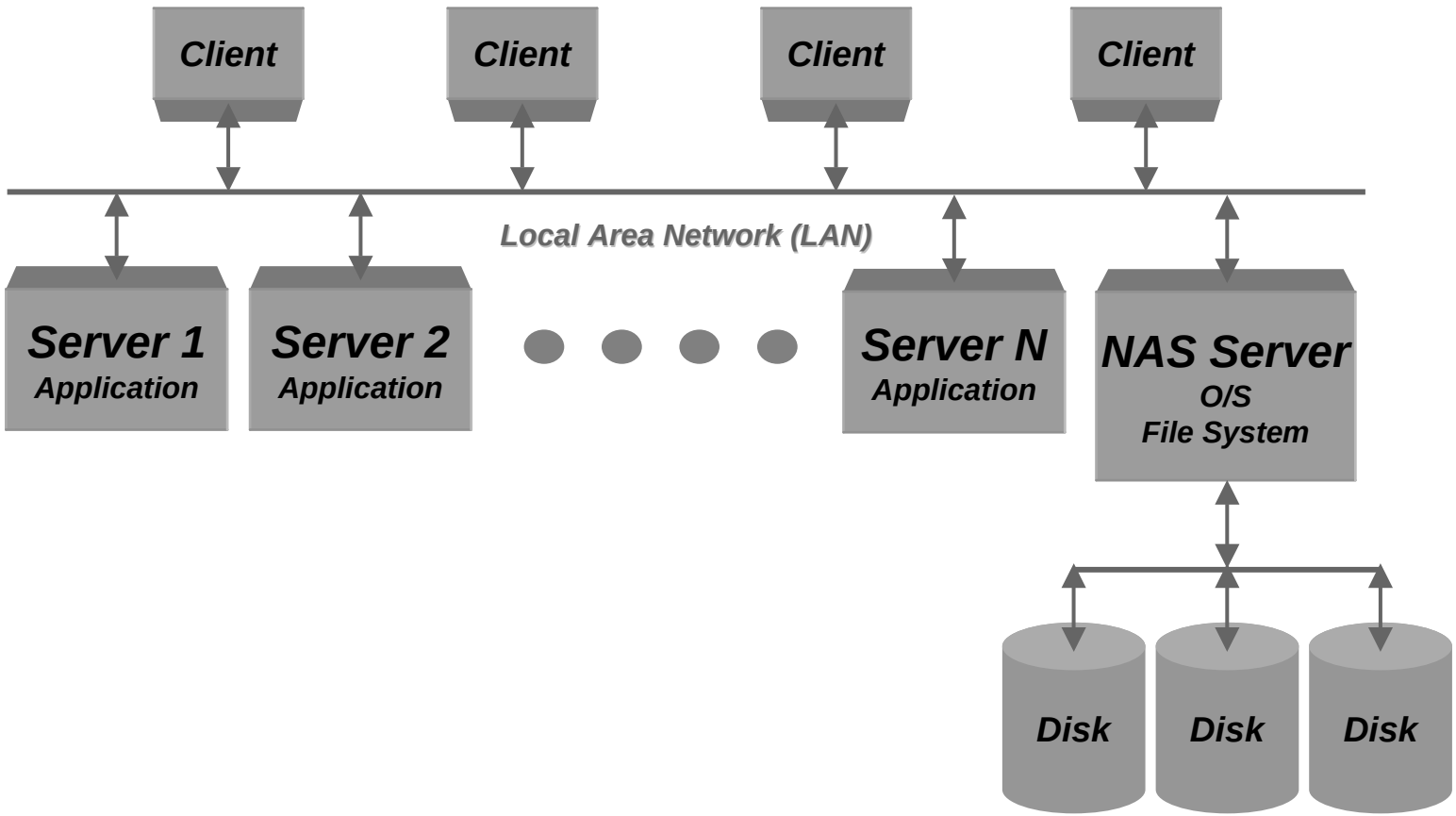
Direct Attached Storage



SAN/Direct Attached Shared Storage



Network Attached Storage

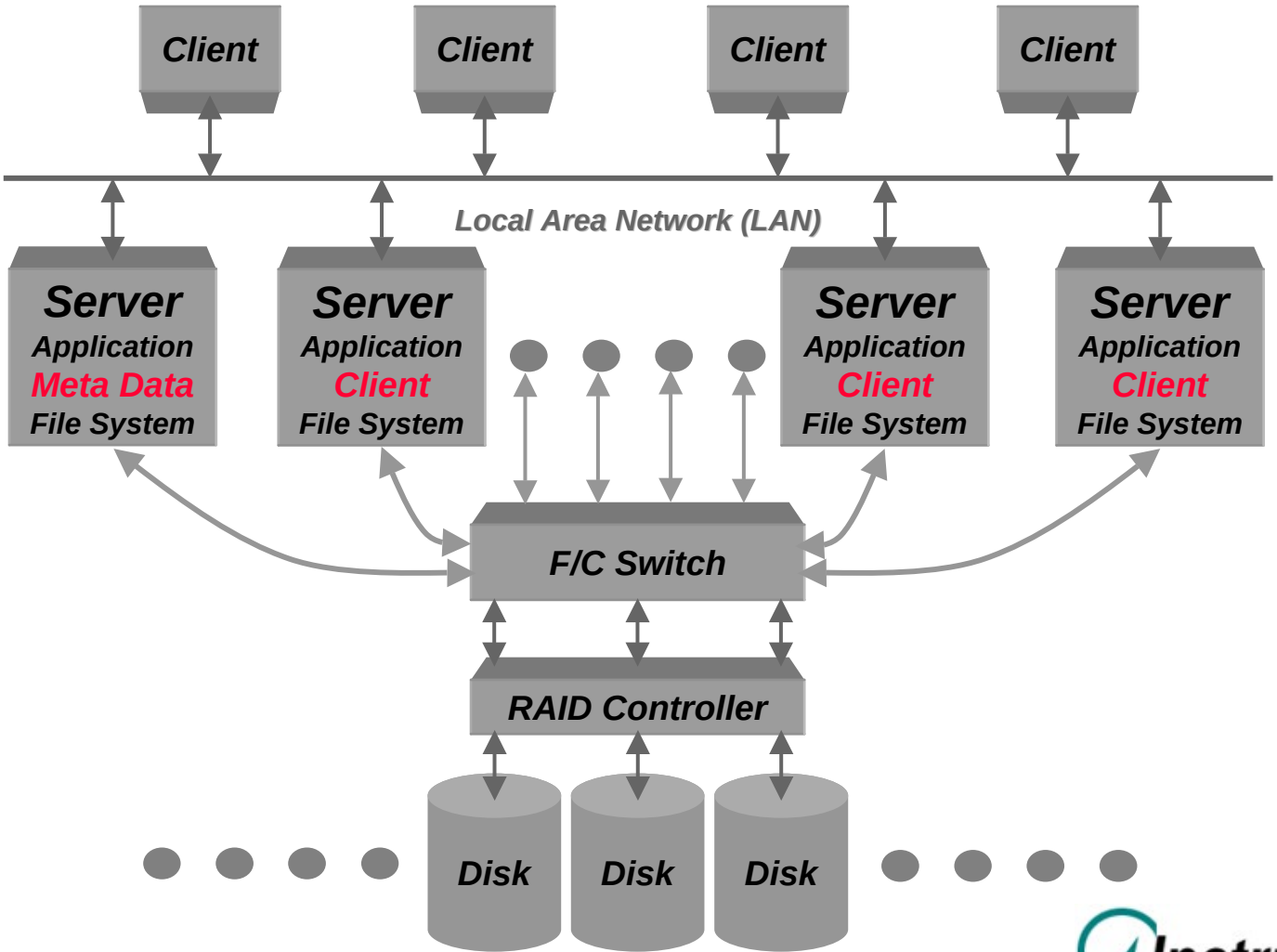


SAN Shared File System (SSFS)

The ability to share data between systems directly attached to the same devices

- ✓ *This is accomplished via fibre channel devices and a specialized file system and/or communications mechanism*
- ✓ *Different types of SAN file systems allow multiple writes to the same file system and/or the more difficult case of the same file open from more than one machine*

SAN Shared File System



SSFS versus NAS Performance

Shared file systems read/write using SCSI over FC

- ✓ *Lower CPU overhead*
- ✓ *Can use many fibre channels*

NAS is limited to GigE and IP performance

- ✓ *iSCSI is coming, but still has higher overhead*
- ✓ *IP overhead is higher than FC overhead*

Shared file systems provide higher performance

Usage Consideration

HPC Applications

Databases

Performance tradeoffs

Where to Use Shared File Systems

Where high performance data movement requires non-IP data movement

- ✓ *Often there is a crossover point between small block I/O over NFS and movement over FC*

Database applications

- ✓ *Requiring many queries as compared with updates*

SSFS are becoming far more common

- ✓ *They are a requirement for some applications*

HPC Applications

Real-time data capture

- ✓ *Often requires 100s of MBytes/sec. or more of performance and many files*
- ✓ *Process flows where one or more machines catch data and others process*

Large computational environments

- ✓ *Movement of data from HPC systems to HSM systems*

Databases

Small block writes

- ✓ *Slower in a SSFS than direct attached storage*
  - *Database updates could be a problem*

Different SSFS are likely needed for different request sizes and access patterns:

- ✓ *Index files*
- ✓ *Redo logs*
- ✓ *Tables space*

Performance Tradeoffs

In a perfect world SSFS data access would be the same as direct attached storage

- ✓ *As we all know the world is not perfect and never will be*

You need to understand your data access patterns and I/O request sizes to determine if the expected request sizes will meet the performance requirements

Metadata updates are critical to performance

Hardware Architecture

- Memory performance
- Metadata hardware
- HBAs
- Fibre Channel Switch
- IP Switch
- RAID controller(s)

System Memory

Memory bandwidth to PCI bus critical

- ✓ Memory latency to PCI bus can result in performance loss
- ✓ Distance between memory and PCI bus crucial
  - The farther data has to move the longer it takes
- ✓ HBAs use Direct Memory Access (“DMA”)
  - Allows read/write operations direct to memory subsystem
  - Requires low latency memory bandwidth for good performance

Host Bus Adapters

HBAs provide connectivity to data

- ✓ *Port count increasing (1, 2 and 4 port)*
- ✓ *Keep in mind failure rate of PCI buses and ports*
- ✓ *Whole card more likely to fail then single port of card*
- ✓ *Path fail over important to ensure data availability*
- ✓ *HBAs can only go as fast as the slot they are plugged into*
  - *2 and 4 port HBAs may not achieve full rate due to PCI bus*
  - *Full speed on single port is 400 MB/sec for 2 Gbit hardware*
    - *(200 write/200 read)*
  - *PCI (66 MHz/64-bit) - Burst 528 MB/sec*
  - *PCI-X (133 MHz/64-bit) - Burst 1,064 MB/sec*

Fibre Channel Switches

Fibre channel switches are the SAN

Port count not everything

- ✓ *New switches coming out with upwards of 1,024 ports*
- ✓ *Single switch still a single point of failure*

Performance dictated by backplane

- ✓ *Switches may have issues sending I/O to different boards*

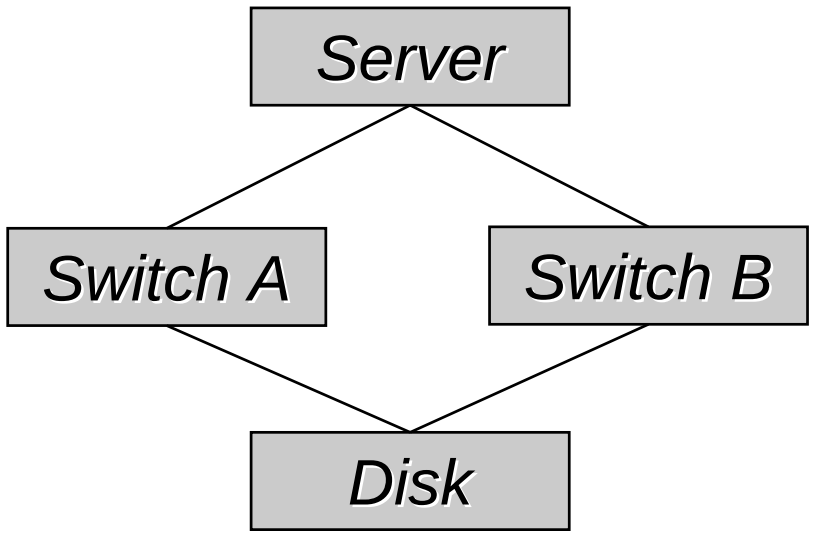
All switches are non-blocking

- ✓ *May not be efficient in routing traffic*
  - *Especially between blades*
- ✓ *Some switches may have known blocking ports*
- ✓ *FC requests always satisfied*

Fibre Channel Switches

Reliability

- ✓ Switches typically very reliable
- ✓ Ports do fail and thus so will I/O paths
- ✓ Important to build redundancy into SAN
  - By using multiple switches



Network Performance

IP networks carry the metadata

- ✓ *Some implementations move file maps around*
  - *From metadata master to the client(s)*
  - *Significant latency if the network is slow; and*
  - *The applications are poorly written*
    - ➢ *Open/Extend/Close/Open example*

"Money can buy bandwidth, but latency is forever"

- ✓ *John Mashey*

Network Switches

Again, port count not everything

- ✓ *Single switch still a single point of failure*

Performance dictated by backplane

- ✓ *Switches may have issues sending I/O to different boards*
- ✓ *Network congestion more likely with IP metadata*

Consider a redundant private network for metadata

RAID Controllers

Hardware RAID is the only HPC solution

Cache performance is key

- ✓ *Write cache mirroring important*
  - *Will slow write performance*
  - *Will cut effective cache size in half*

For reliability metadata write cache mirroring is more important than data

RAID Alignment

RAID volume configuration

- ✓ *Must be matched to the volume manager and file system allocation for high performance*
- ✓ *Large requests will not run at rate without this*
  - *Due to the extra overhead required in the controller*
    - ➢ *Especially important for stripe groups*
- ✓ *Match volume configuration to application request size*
  - *Request size = nDisks * block size*

RAID Controllers LUN Mapping

Match I/O to volume configuration

- ✓ *Large requests*
  - *More disks, larger volume block size*
  - *Require larger block sizes to efficiently use RAID volume stripe*
- ✓ *Small requests*
  - *Smaller volume sizes and thus smaller RAID volume stripe widths*

File system configuration must be understood

- ✓ *Need to know how file system will be laid out to configure RAID volumes*
- ✓ *What level of data redundancy (RAID-5, RAID-1, etc…)*

Components of a Shared File System

- Share file system architecture
- Metadata
- Clients
- Storage

SSFS Architecture

Similarities of shared file systems

- ✓ *Client/server based*
- ✓ *Metadata servers are required*
  - *Client authentication*
  - *Control access to metadata*
- ✓ *Metadata travels via network*
  - *NFS or CIFS*
  - *Proprietary protocol*
- ✓ *Clients access data directly via SAN*

SSFS Architecture

Metadata server

- ✓ Services client metadata requests
  - Open, close, delete, move files
  - Allocate data blocks for write
  - Send file maps

Client(s)

- ✓ Network attached for metadata ops
  - Open, close, delete, move files
- ✓ SAN attached for data ops

SSFS Architecture

Metadata (Inodes)

- ✓ *Many hosts cannot write metadata simultaneously*
- ✓ *Single host handles inodes and disk allocations*
- ✓ *Shared via LAN or WAN*
- ✓ *Protocol most likely proprietary, may use NFS or CIFS*

Data

- ✓ *Clients cannot modify metadata, only make requests*
- ✓ *Data read/written by client directly to storage via SAN*

SSFS Architecture

Client/Server Communications

- ✓ Client uses open(2) call to request file for write
- ✓ Client sends request to server
- ✓ Client/server authentication
- ✓ Server allocates/assigns metadata
- ✓ Server allocates space on physical data device(s)
- ✓ Server responds with file map or information
- ✓ Client may request extent when file grows past initial allocation
- ✓ Client closes file and sends request to server
- ✓ Server may truncate map on close

Metadata

Metadata is the file system

- ✓ Each file has at least on inode allocated
- ✓ Metadata critical for file system operation
  - Data may be physically on storage, but without a pointer it is useless
- ✓ May hold information about data backup/archive
- ✓ Generally extremely small writes/reads and very random
  - Should be on separate device away from data
  - Prevents head contention on physical spindles
- ✓ HA metadata servers a must in critical environment
  - New level of difficulty for some products

Clients

Heterogeneous is the future

- ✓ Some file systems currently support heterogeneous systems
  - Windows
  - Linux (RedHat, SuSE, etc…)
  - UNIX (Solaris, IRIX, AIX)
- ✓ Clients perform most if not all I/O to data devices
  - Server simply handles metadata
- ✓ Should have own, redundant connections to storage
  - File system is no good unless clients can access it

Storage

Different types and levels of storage for SSFS

- ✓ Disk for high performance data access
- ✓ Tape or optical for archive/backup
- ✓ Data may reside at several levels (HSM)
  - Online - on disk and available via SAN
  - Near line - on tape, optical or other lower performance media yet still available to clients
  - Offline - on tape, optical or other removable media
    - ➢ May require human intervention to retrieve

HSM and Shared File Systems

Tape Connectivity

Tape Compression

Tape Connectivity

Not needed on all systems

- ✓ *Metadata server typically responsible for handling tape requests and I/O*
- ✓ *Not all tape drives can be connected via switch*
  - *Switch may have a translative mode but may not be effective*
- ✓ *Switch attached allows for multiple server access*
  - *Systems go down, but tape drives may not*
- ✓ *Catalogs critical to tape access just as metadata*
  - *Holds access and capacity information*
  - *Tape has limited life span*

Tape Compression

Compression allows for effective use of tape storage

- ✓ *Hardware compression important for performance*
- ✓ *Compression buffer size extremely important*
- ✓ *Careful what kind of data is used for capacity planning*
  - *Text data extremely compressible*
  - *Binary information may compress little if any*
  - *Some binary data may expand when compressed*

Always plan for the worst case scenario

- ✓ *Little if any compression of data unless data understood*

Software Considerations

Operating Systems

Storage Software

Operating Systems

Some operating system and driver defaults do not allow large requests

- ✓ *You need to change defaults to allow requests greater than 128 KBytes in some cases*

System page I/O overhead is greater in with some systems

Linux does not support large requests

- ✓ *Over 128 Kbytes*
  - *Fixed in the 2.6 kernel*
- ✓ *Random request size 4 KBytes - 128 KBytes*

Disclaimer

The following slides were provided by each of the respective vendors. We are presenting their products features without comment or changes to the slides other than formatting. We make no claims as to the accuracy of these slides

Current SSFS Products

- Architecture
- Supported platforms
- Performance considerations

Current Products

ADIC

- ✓ StorNext

IBM

- ✓ Storage Tank

SGI

- ✓ CxFS

Sun

- ✓ QFS

IBM

- ✓ SANergy

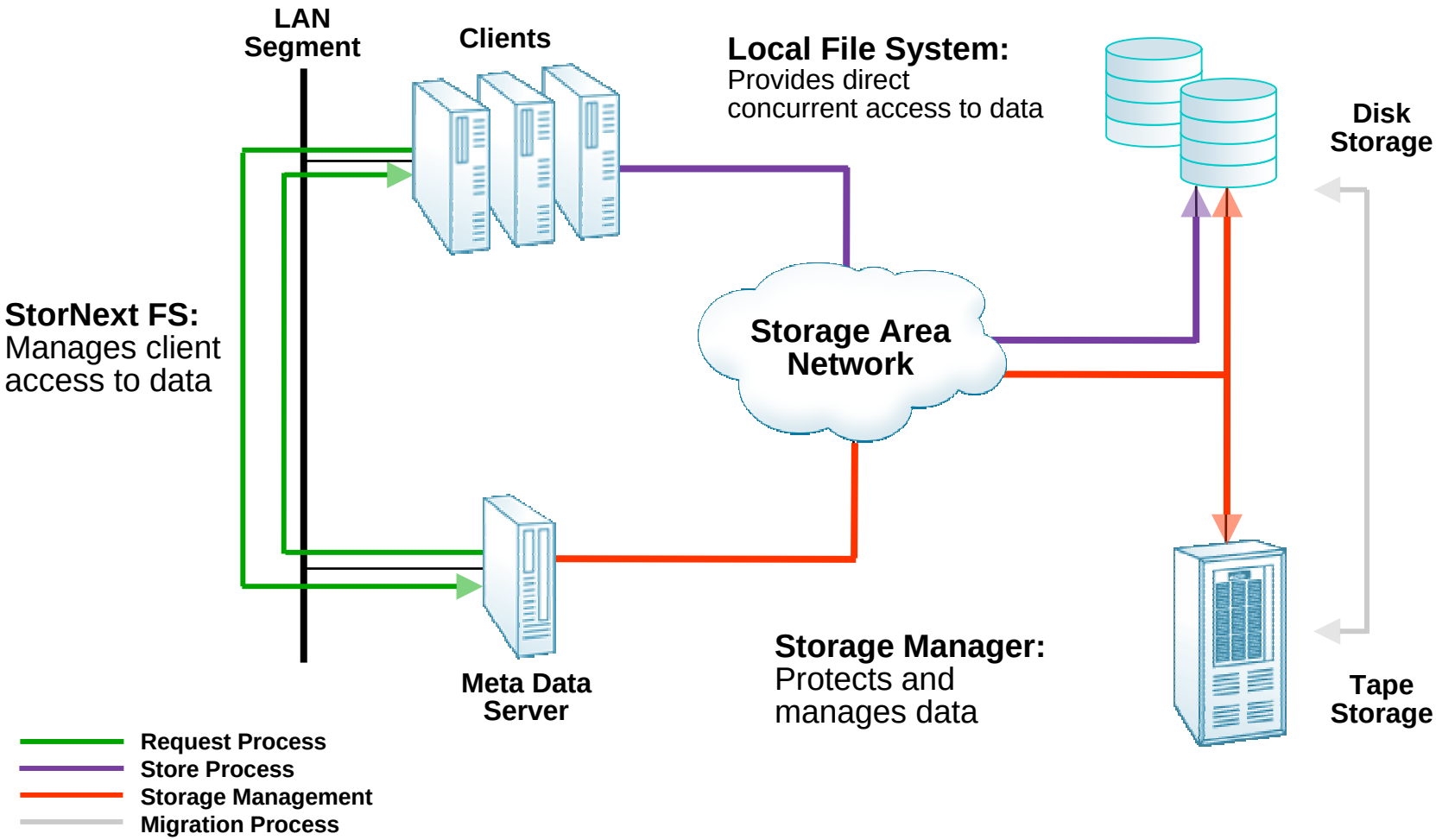
VERITAS

Instrumental
www.instrumental.com

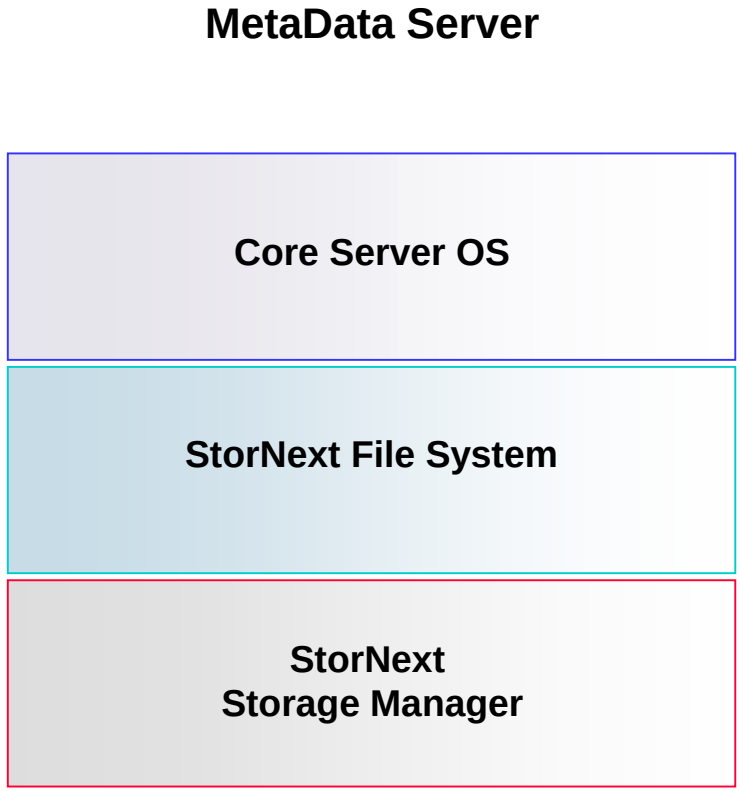
StorNext Supported Operating Systems

	StorNext File System	StorNext Storage Manager
Sun Solaris	√	√
IBM AIX	√	
SGI IRIX	√	√
Red Hat Linux	√	√
SuSE Linux	√	
Microsoft Windows NT, 2000, XP, 2003	√	

StorNext Data Flow



StorNext Metadata Server



StorNext Metadata Server

StorNext File System

- ✓ *Installable file system*
  - *Journaled*
  - *64 bit*
  - *POSIX compliant*
- ✓ *Normal UNIX administrative operations allowed*
- ✓ *Provides functionality to optimize disk access*
  - *Pre-allocation*
  - *Metadata separation*

StorNext Metadata Server

StorNext File System

- ✓ *Manages client data requests*
  - *Authenticates system access*
  - *File system is transparent to clients*
- ✓ *Allows simultaneous read / write access*
  - *Block level*
  - *Up to 128 clients*
- ✓ *Supports stripes and affinities*

StorNext Policy Classes

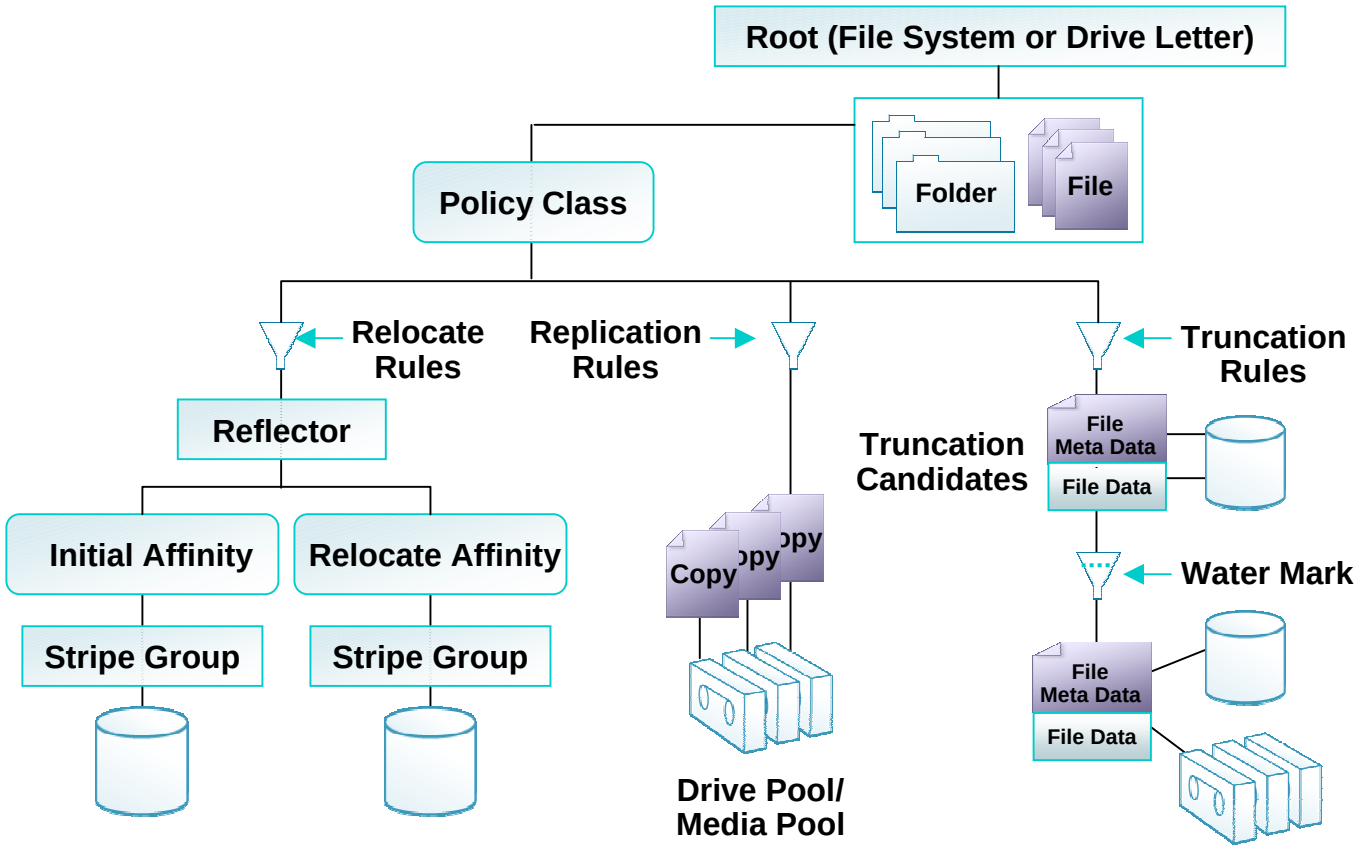
Determines

- ✓ When files are moved from disk to disk
- ✓ When files are moved from disk to tape
- ✓ When files are truncated from disk

Files to truncate / store

- ✓ Based on candidate list
  - List built by size / time since last access
  - Administrator can define other properties for determining candidates
- ✓ For truncation, a file must already be on tape
  - If multiple tape copies specified, then all copies must be on tape
- ✓ Files can be excluded from truncation
  - If disk access speeds are required

StorNext Policy Schematic



Instrumental
www.instrumental.com

StorNext User Interface

Browser-based

- ✓ *HTML*
- ✓ *JavaScript (no Java)*

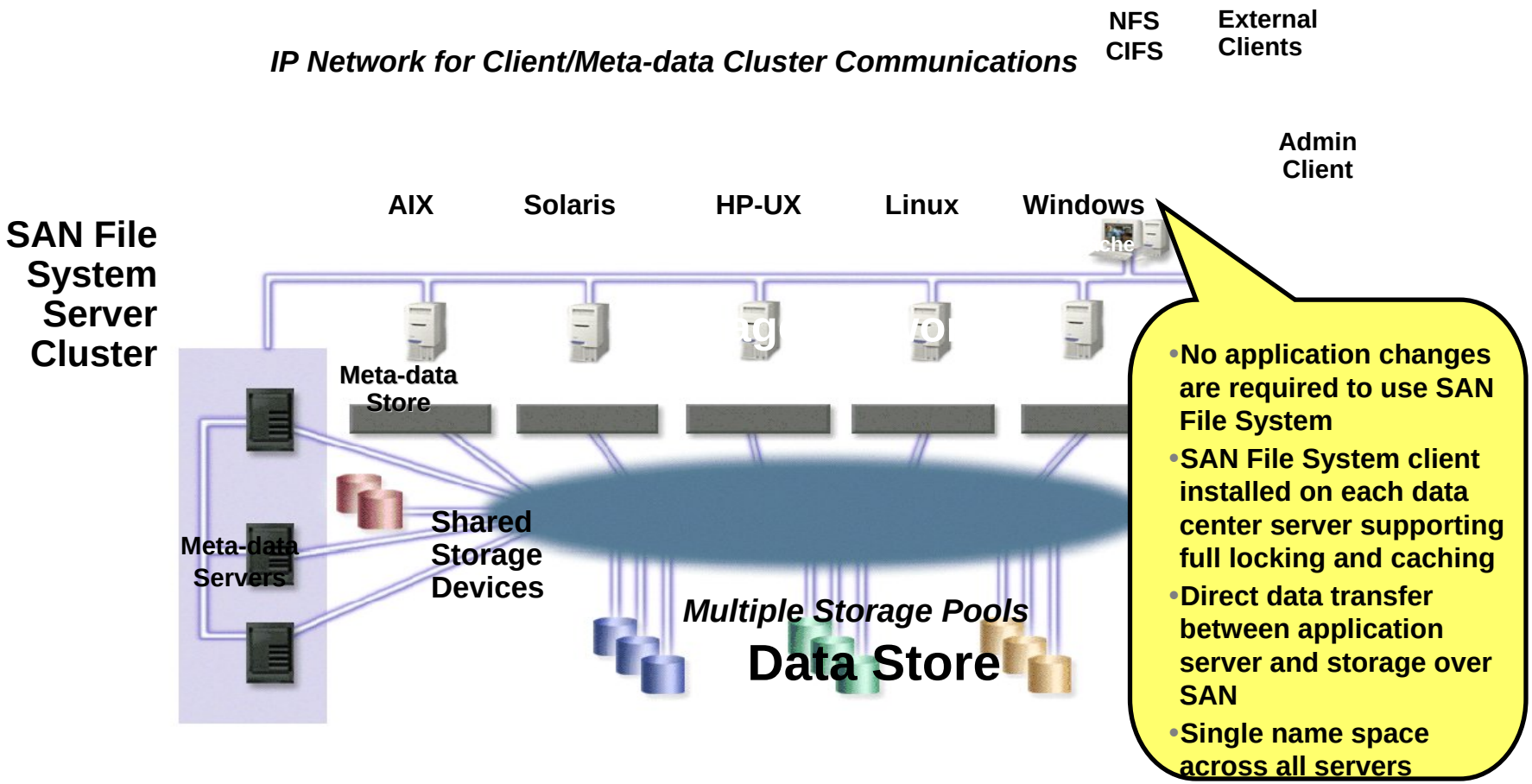
Logins with capabilities

- ✓ *Administrators*
- ✓ *Operators*
- ✓ *Users*

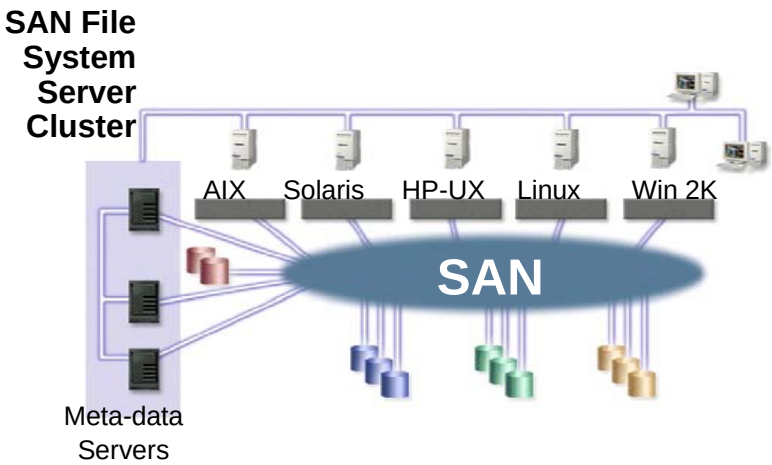
Configurable capabilities

- ✓ *For each login*

IBM SAN File System Architecture



IBM SAN File System Clients

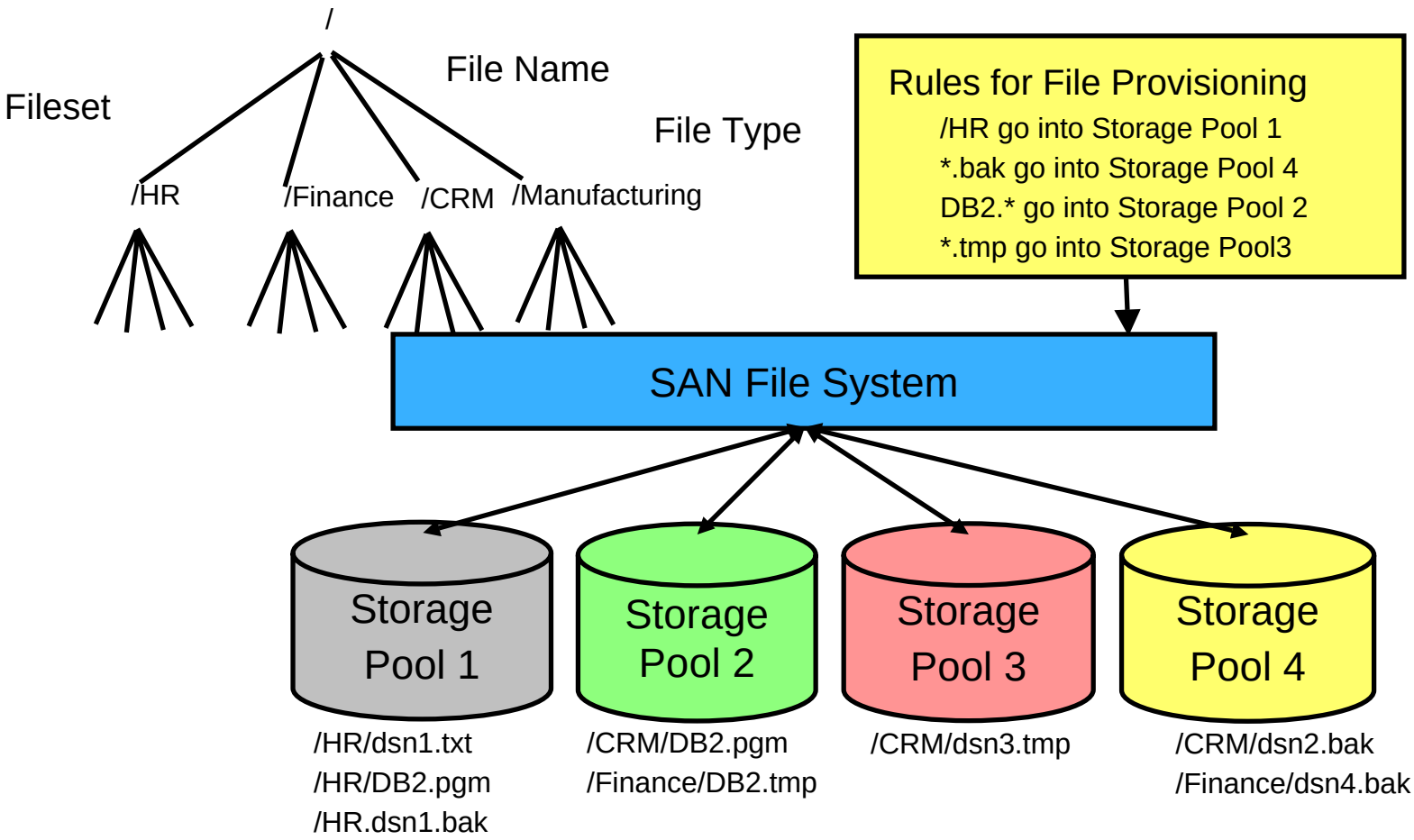


- **SAN File System Client Operating Systems**
  - Windows 2000 Server and Advanced Server
  - AIX
  - Solaris
  - HP-UX
  - Linux
  - Support current release and prior release
- **Transparency**
  - No application changes
  - POSIX compliant file interface
  - Uses native VFS or IFS interface
  - Meta-data caching
  - File locking
  - Mandatory and advisory byte range locking
  - Native operating system security

IBM Policy-based Provisioning

Automated provisioning through customizable rules

Any attribute available in the meta-data can be used to automate allocation of a file to a storage pool



IBM SAN File System Management

- Task oriented graphical user interface
- Web browser based
- Five levels of admin authority
  - Monitor
  - Operator
  - Admin
  - Backup
  - IBM Support

Web Browser
Consistent User Console

Scripting Tool
Consistent CLI Syntax

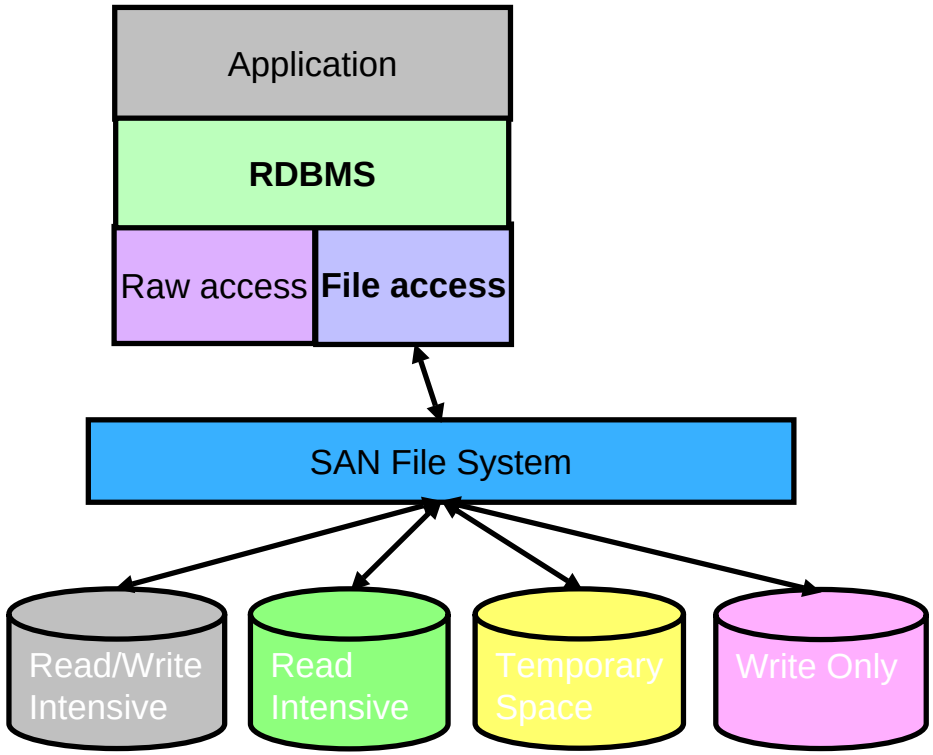
Management Application
Industry Standard Object API

Web Pages *CLI* *xmlCIM*

CIM Agent

- **Monitor system**
  - System overview
  - View logs, alerts
- **Manage meta-data servers, clients, and cluster status**
- **Manage filing**
  - Create fileset
  - Create policy
- **Manage storage**
  - Define storage pools, volumes, available LUNs
- **Maintain systems**
  - FlashCopy images
  - Meta-data checking and recovery
- **Administer users**

IBM Database Support



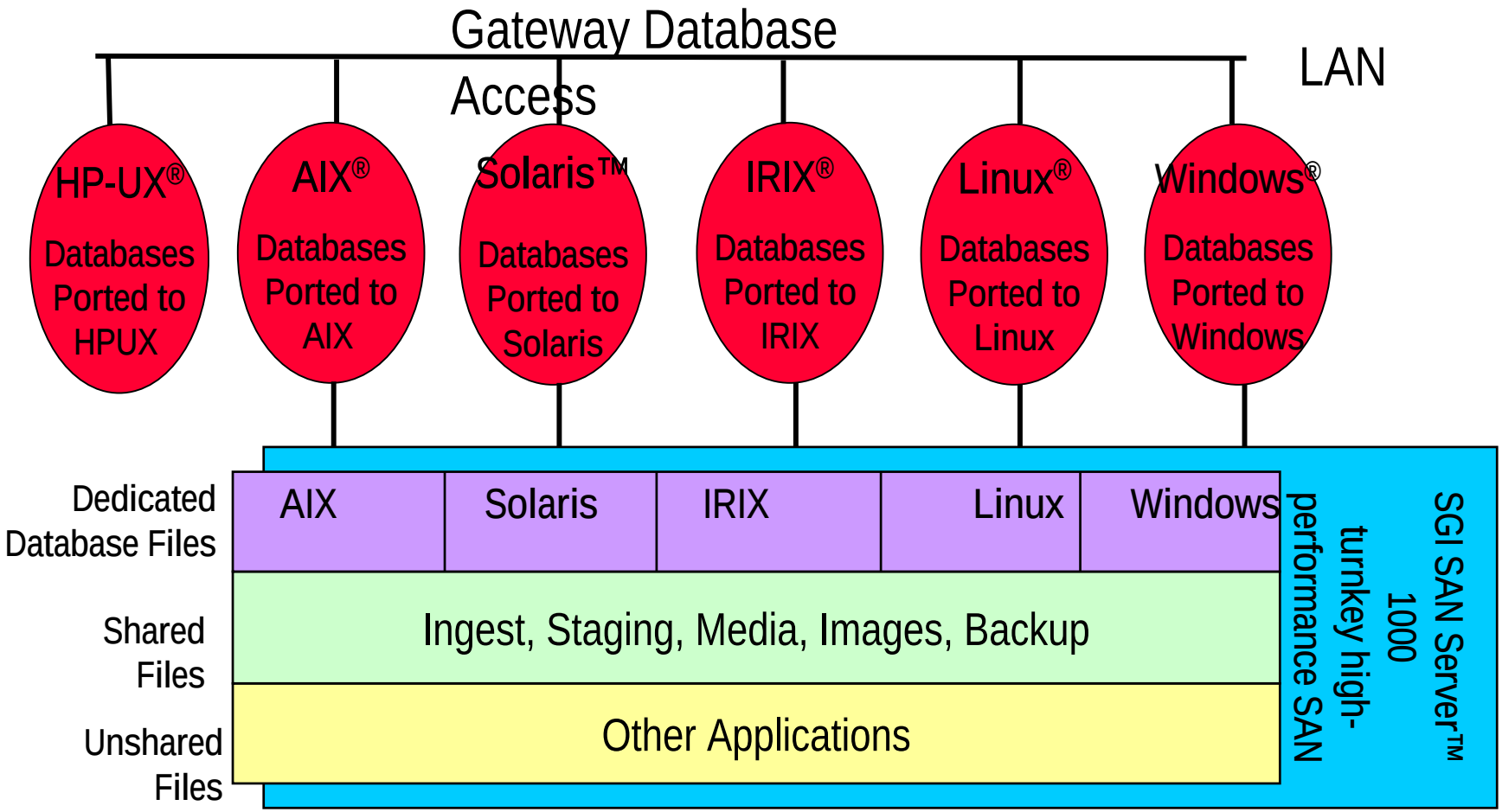
- **Support for databases that use local system file system interfaces**
- **Support for high performance database requirements**
  - Direct I/O
    - Direct memory mapping
    - No memory copy
    - No context switch
  - Parallel I/O
  - Mapping of different access patterns and requirements to storage pools with the appropriate service class

SGI CXFS

A high-performance file system for sharing data across a SAN:

- ✓ *Industry's most scalable 64-bit shared file system; supports file size up to 9 million TB and file systems up to 18 million TB*
- ✓ *Heterogeneous client support for IRIX, Solaris, Windows*
- ✓ *Allows multiple computer systems direct access to read/write the same file on the same disk at the same time*
- ✓ *Proven, time-tested technology running mission-critical applications at hundreds of sites*
- ✓ *High availability with automatic failure detection and recovery*
- ✓ *Based on the 64-bit SGI XFS file system, widely recognized as the most scalable, highest-performing file system in the industry*

CXFS - Access Any Data Anywhere



Instrumental
www.instrumental.com

CXFS: Clustered XFS

CXFS attributes

- ✓ *Shareable high-performance XFS file system*
  - *Shared among multiple nodes in a cluster*
  - *Near-local file system performance*
  - *Direct data channels between disks and nodes*
- ✓ *Resilient file system*
  - *Failure of a node in the cluster does not prevent access to the disks from other nodes*
- ✓ *Convenient interface*
  - *Users, applications and developers see standard UNIX file systems*
    - ➢ *Single System View (SSV)*

CXFS Supports Full POSIX API

POSIX compliant API

- ✓ API = Application Program Interface
- ✓ Fully coherent buffering
  - As if all processes were on an single SMP
  - Writes flush caches 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

Supports up to 64 nodes per cluster

- ✓ *48 nodes now*
- ✓ *64 clients in 3QCY03*

Multiple metadata servers can exist in a cluster

- ✓ *One per file system*

Files accessed locally on CXFS server

- ✓ *See local XFS performance*

CXFS Scalability

CXFS client nodes are "lightweight"

CXFS is a layer on top of XFS

Fast recovery times

- ✓ *No fsck*

QFS Addresses Data Growth

- **Unlimited number of files**
- **Fast file system recovery**
- **Up to 252 TBytes of online data**
  - ✓ *In one file system*
- **PBytes of nearline data**
- **Minimizes operational expenses**
- **A hierarchical file system**
  - ✓ *Bounded only by the size of your storage*
    - *Nearline & offline*

Instrumental
www.instrumental.com

QFS Delivers Performance

QFS has built in volume management

Metadata and data are separated

Striping and round robin allocation

User selectable allocation

- ✓ *Stripe width and LUN selection*

Variable block size to match hardware

QWRITE

- ✓ *Simultaneous writes to the same file*

Instrumental
www.instrumental.com

Shared QFS File Consistency Model

File consistency

- ✓ *Managed with read/write/append leases*

Only one host can write to a file at any one time

- ✓ *Default setting*
- ✓ *however multi-host write access can be enabled with the mount parameter “mh_write”*

Append hosts increase filesize

- ✓ *Write hosts write within filesize*

Memory mapped I/O disables “mh_write”

Shared QFS File Consistency Model

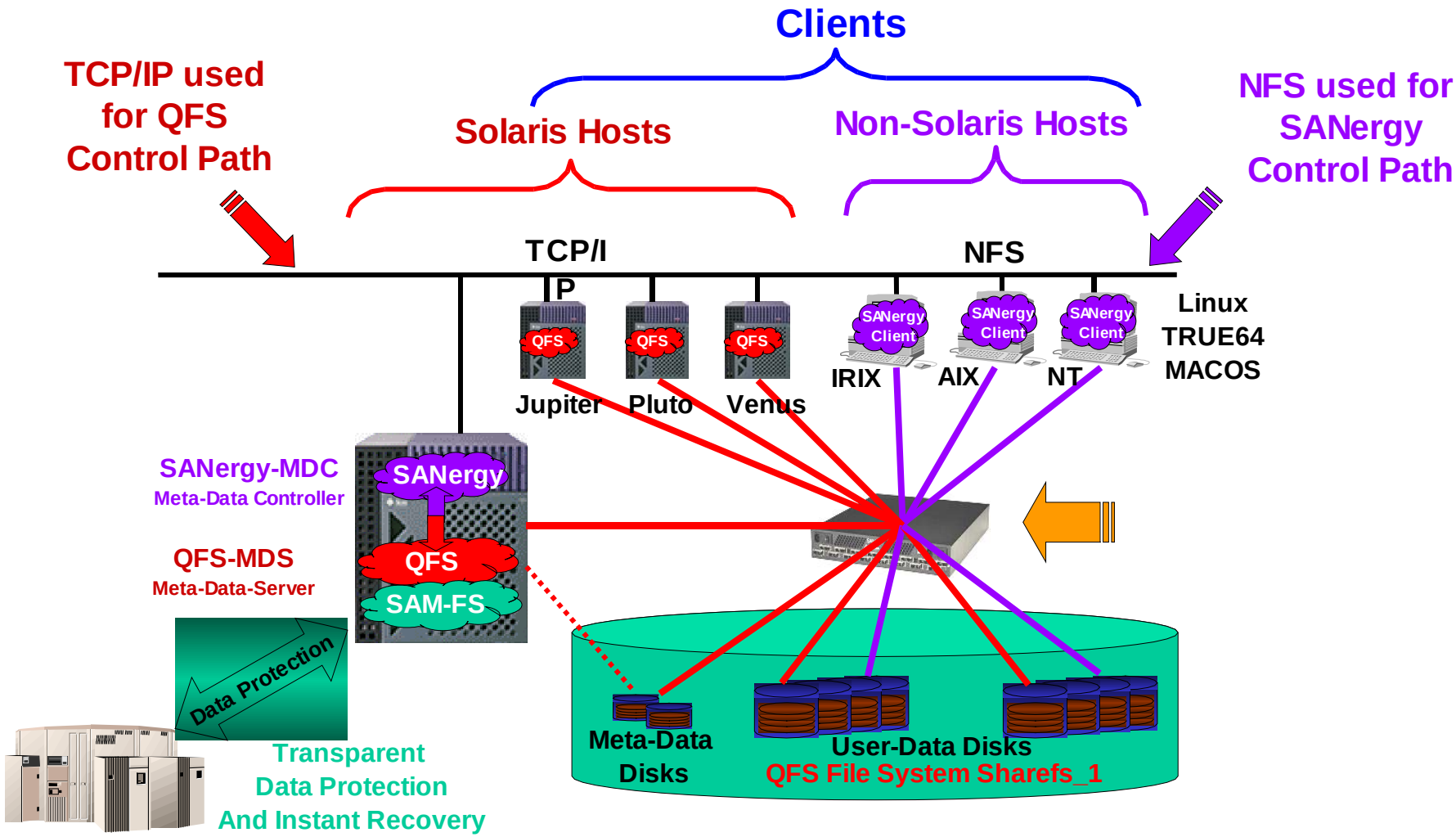
Lease Type	Multi-Host Write Disabled	Multi-Host Write Enabled
Read	•Multiple reader hosts allowed •Can use paged I/O	•Multiple reader hosts allowed •Can use paged I/O
Write	•Only one writer host allowed •Writer can use paged I/O •All other hosts wait	•Multiple readers and writers allowed •If other hosts are accessing the file, all I/O is direct
Append	•Only one append host allowed •All other hosts wait for write •Other hosts can read; all I/O is direct	•Only one append host allowed •If other hosts are accessing the file, all I/O is direct

QFS 4.0 File Sharing with SANergy

Configuration Requirements

- ✓ *All "Basic Configuration Requirements" and:*
  - *QFS shared file system is required for SANergy*
  - *SAM-FS can run in addition to QFS (SAM-QFS)*
  - *SANergy Meta-Data Controller (MDC) must run on the QFS Meta-Data Server*
  - *All Non-Solaris clients to share the file system must have SANergy Client software installed*
  - *The file system to be shared must be NFS exported on the QMS and NFS mounted on each SANergy client*
- ✓ *For Failover*
  - *All "Failover Configuration Requirements" and:*
  - *HA application must failover SANergy MDC*

Data Sharing in a SAN with QFS 4.0



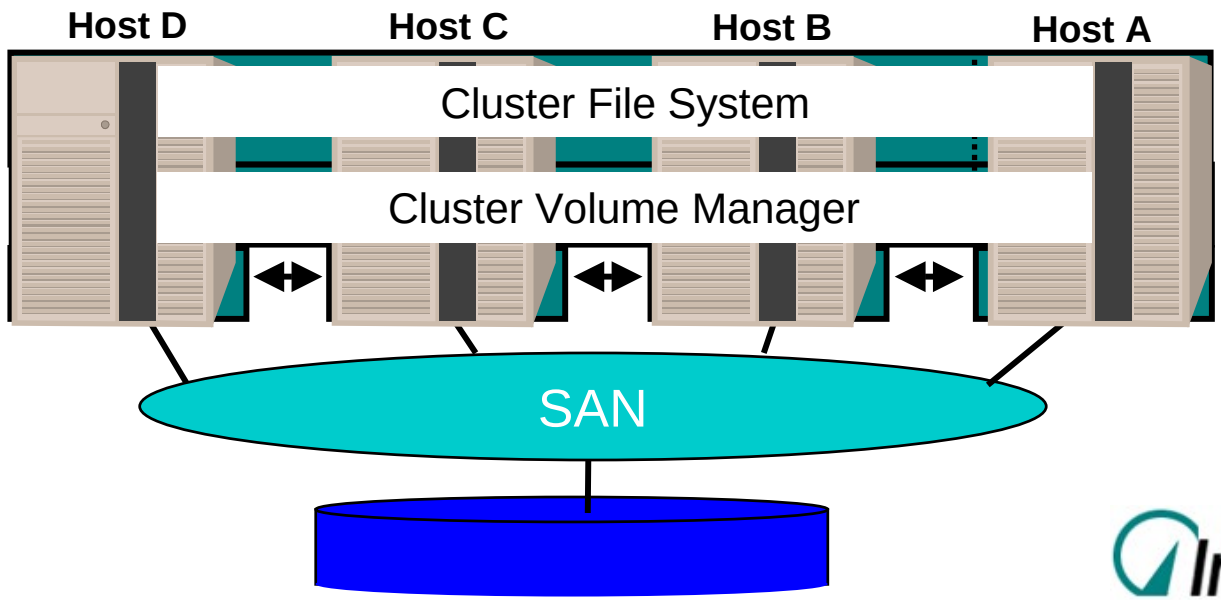
Instrumental
www.instrumental.com

VERITAS SF Cluster File System

Shared, online storage management & POSIX compliant file system

Tight integration of CFS, CVM & VCS

Available for Solaris, HPUX (Linux & AIX coming)



VERITAS SF Cluster File System

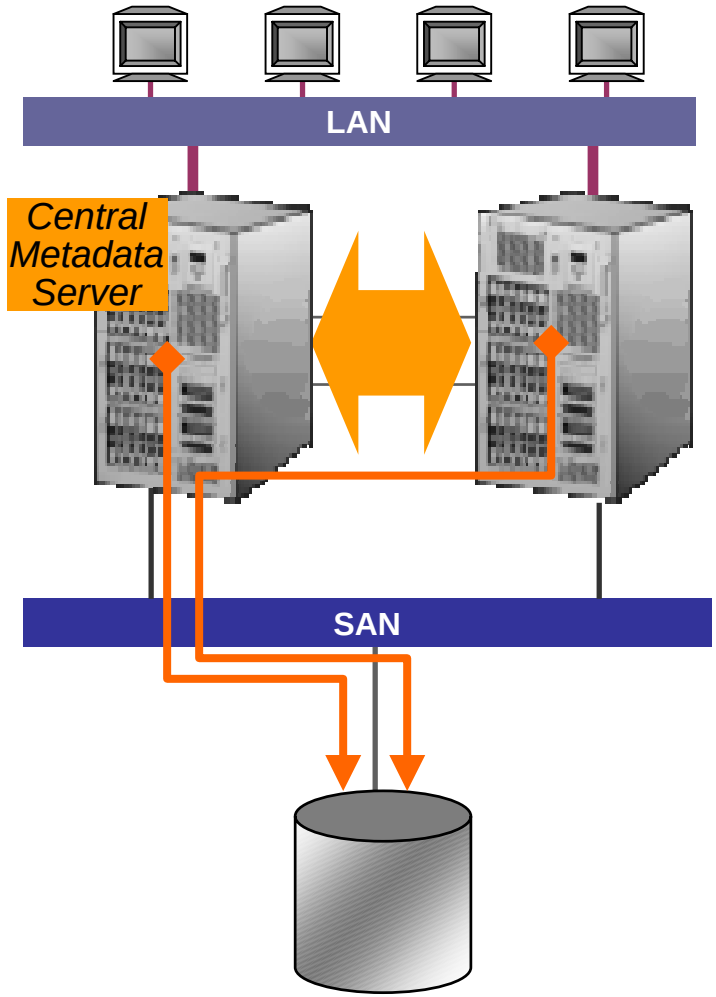
Extends local file system functionality

Asymmetric topology

- ✓ *Servers access data directly*
- ✓ *Any server can update most metadata*
- ✓ *The master updates the log*
- ✓ *The master can failover at any time to another node*

Advantages

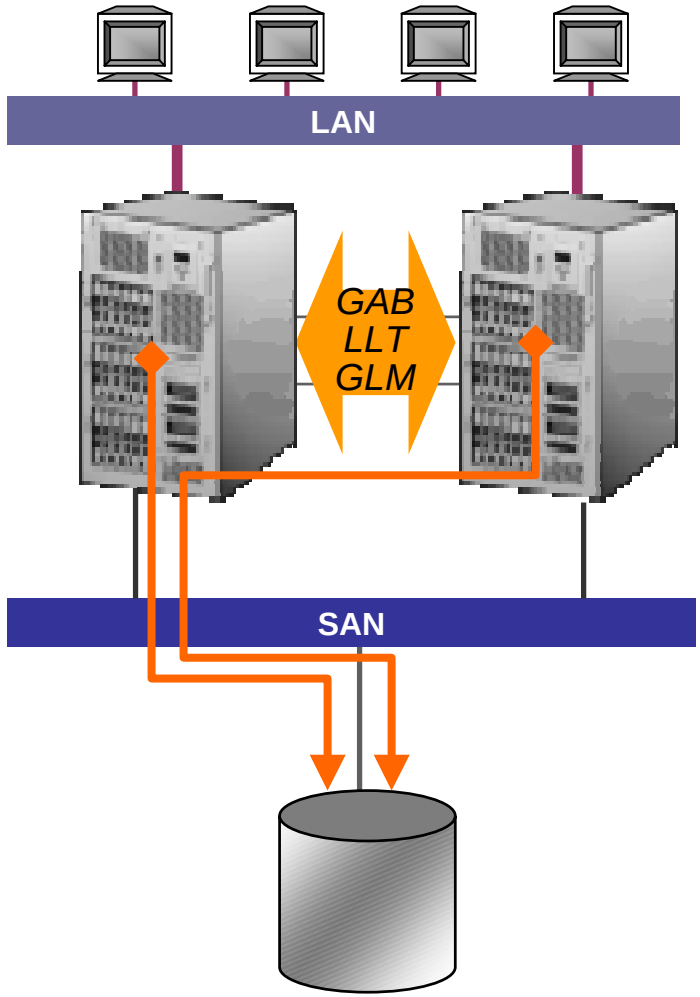
- ✓ *Less metadata traffic*
- ✓ *More scalability*



VERITAS SF Cluster File System

"Cluster Awareness"

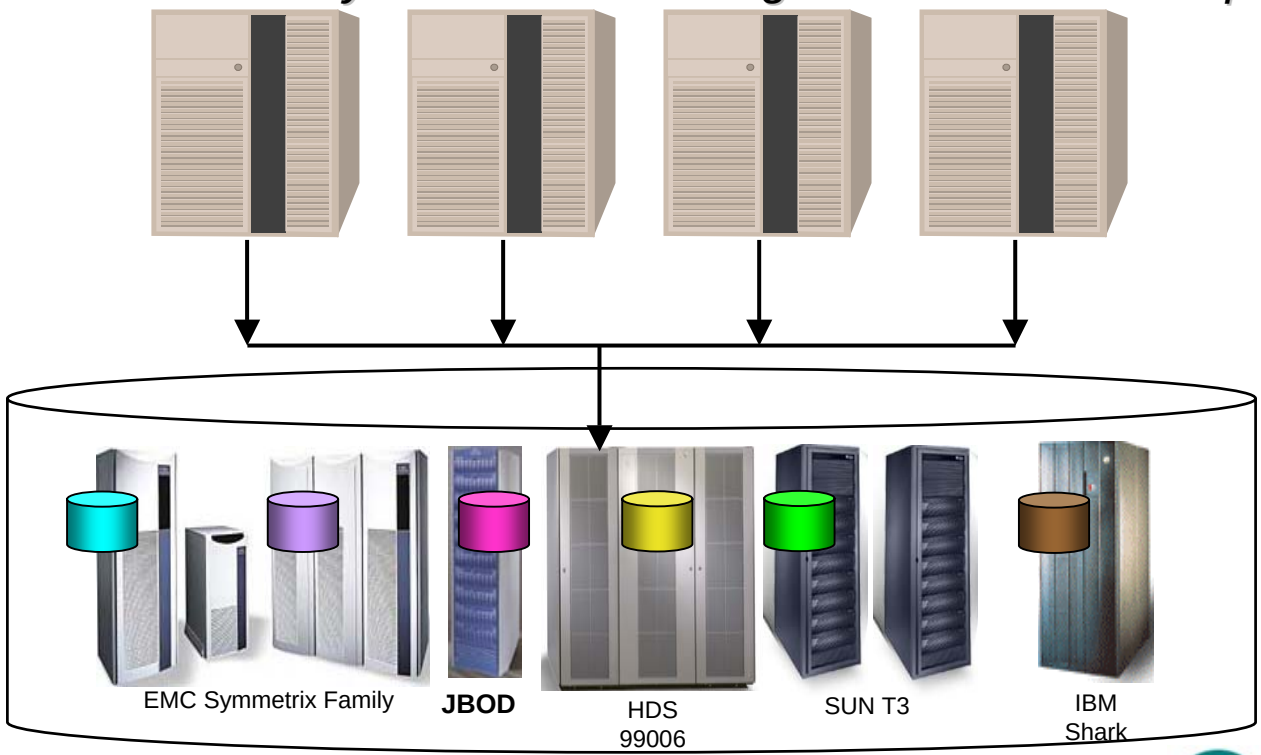
- ✓ *Determining cluster membership and inter-node communication*
  - *Global Atomic Broadcast protocol (GAB)*
  - *Low Latency Transport protocol (LLT)*
- ✓ *Failover of CFS Master, CVM Master, and Applications*
- ✓ *Node lockout for split-brain (cluster partitions)*



VERITAS SF Cluster File System

Cluster Volume Management

- ✓ *Enables servers to view and share disk groups across a cluster*
- ✓ *Simultaneous access to volumes from multiple servers*
- ✓ *Clustered Dynamic Multi-Pathing notifies all nodes of path state*



VERITAS Discovered Direct I/O

HPC for applications with mixed I/O sizes

For large FS writes, it is much faster to write directly to the disk

- ✓ *Direct I/O skips the FS cache*
- ✓ *BUT, for smaller writes, direct I/O is slower!*

We developed... Discovered direct I/O

- ✓ *DDI/O looks at the size of each write & gets the best performance from each I/O model*

Example:

If write is <=256KB, uses the FS cache

If write is > 256KB, uses Direct I/O

Instrumental
www.instrumental.com

VERITAS Cluster File System

Concurrent access

- ✓ *To file systems from multiple nodes*

All nodes can directly read/write

Dynamically resize file systems

64-bit file system support

Journaling file system

Installation of Shared File Systems

Choosing a metadata server

Metadata communication

Choosing a Metadata Server

Platform choice limited by applications

- ✓ *Operating systems and thus hardware choice depend greatly on the applications and what they need to run on*

What is the server doing?

- ✓ *If more than sharing metadata then platform decision more important for…*
  - *PCI bus (PCI-X versus PCI)*
  - *Number of HBA slots*
  - *Size and type of memory*
  - *Not least of all processor*

Metadata Network

The network is as important as metadata

- ✓ *Metadata network should be separate from other traffic if possible*
  - *Removes possibility of network overloading by users and other applications*
- ✓ *GigE versus 100 BASE-T not that important*
  - *GigE if using metadata on in-band network*
  - *GigE may be necessary to sustain large number of requests and large file maps*
- ✓ *Multiple networks for all server, potential servers and clients key to file system availability*
  - *Includes using network multi-pathing software*

Architecting for HA/RAS

- HBAs
- Switches
- RAIDs
- MDC
- Clients

Architecting for HA/RAS

Try to eliminate single points of failure

- ✓ Requires redundancy on all levels
- ✓ Hardware
  - HBAs, RAID controllers and switches
- ✓ Service redundancy
  - Metadata and archive
- ✓ Data redundancy
  - Multiple copies, streams and RAID protection

Architecting for HA/RAS

HBA fail over difficult in heterogeneous environment

- ✓ *Not all platforms support HBA fail over*
- ✓ *Fail over mechanisms may cause problems with other platforms*
  - *Especially difficult with asymmetric RAID controllers*
  - *Enterprise RAID controllers are generally symmetric and do not cause similar issues*

Architecting for HA/RAS

Fibre Channel Switches

- ✓ Single switch = single point of failure
- ✓ Most switches have fully redundant components
- ✓ What about…
  - Firmware upgrades
  - Power outages
  - Human error
- ✓ Dual switches allow for;
  - Greater reliability
  - Load balancing
  - More ports

Architecting for HA/RAS

RAID Controllers

- ✓ *Controllers typically in redundant configuration*
  - *Asymmetric - Active/passive volume access through one or the other controllers*
  - *Symmetric - Active/active volume access through multiple controllers*
- ✓ *RAID levels 1, 1+0 and 5 are typical for redundancy*
  - *RAID 1 - Mirrored volume*
  - *RAID 1+0 - Striped mirrored volume*
  - *RAID 5 - Striped volume with parity*

Architecting for HA/RAS

RAID Controllers

- ✓ *Single controller still single point of failure*
- ✓ *Software volume mirroring can help, but may be costly in performance*
- ✓ *Most vendors have block level remote copy available on controller*
  - *Can replicate within SAN or long-distance across network or extended SAN*

Architecting for HA/RAS

Metadata Servers

- ✓ *Cluster, cluster, cluster*
- ✓ *Metadata controllers are crucial to file system availability*
- ✓ *Some vendors have built in fail over for metadata services*
- ✓ *Cluster software of some sort most likely a requirement*
  - *Applications are just as critical as metadata and need to be available for use*
  - *Supporting resources need to be available as well*
- ✓ *File systems can still only have one metadata server*
  - *May be able to load balance multiple file systems over multiple servers*

Architecting for HA/RAS

File System Clients

- ✓ Typically not configured into clusters
- ✓ Do need to know how to attach to HA metadata pair
  - Single host name that can fail over between metadata servers
- ✓ Fail over should be as transparent to clients as possible
  - Clients will most likely receive interruption of service during failover
  - Time of interruption greatly varies depending on fail over mechanism and other resources required for full operation

Looking Toward the Future

4 Gb interconnect

- ✓ *Is farther out than we wish*

Server centric file systems

- ✓ *Are here for the foreseeable future*

SAN/WAN performance and security

- ✓ *Presents many issues and concerns*

Object bases storage

Object-based Storage Devices

Storage groups blocks into “Objects”

- ✓ *All I/O done to objects*
- ✓ *Similar to a file system but at the device level*
- ✓ *Basically each drive has a file system*
- ✓ *T10 committee www.t10.org*

Advantages

- ✓ *Simplified security*
- ✓ *Allocation could be improved based on*
- ✓ *Object management easier than block management*
  - *New applications for object mirrors and copies*

Thank You

Instrumental