NFS, its applications and future

Brian Pawlowski
Vice President and Chief Architect
beepy@netapp.com

Who am I? Why am I here?

The arc of the presentation

- What is NFS?
- The evolution of NFS
- NFS Version 4
- Drill down: Linux NFS Version 4
- What about iSCSI?
- Linux compute clusters
- NFS in context
- Challenges for NFS

With occasional sidetracks…

What is NFS?

What is NFS?

- NFS is a protocol for a distributed filesystem.
  - Based on Sun's RPC version 2 protocol
  - Can export arbitrary local disk formats
- First revision, NFSv2, was published in 1985.
  - It exports basic POSIX 32-bit filesystems
  - Slow, particularly for writing files
- NFSv3, was published in 1994
  - Extended to 64-bit files & improved write caching
  - It is perhaps the most commonly used protocol for sharing files on *NIX/Linux LANs today

Remember these file systems?

- Apollo Domain
- AT&T Remote File System (RFS)
- Andrew File System (AFS)
- Distributed File System (DFS)

(NFS Version 4 is influenced by AFS)

NFS Today

- "It was 20 years ago today."
  - SCSI and NFS grew up together
- Transformed from something you turn on in a UNIX release to a well-defined storage segment
- Home directories
- Large partitionable tasks that may run as parallel threads
  - Typical applications include search engines, e-mail, animation and rendering, scientific simulations, and engineering
- Scalable databases
- GRID computing

NFS Version 4

NFS Version 4

- Openly specified distributed filesystem
  - NFSv2/v3 quasi-open with Informational RFC
  - Windows, AFS, DFS not “open”
- Well-suited for complex WAN deployment and fire-walled architectures
  - Reduced latency, public key security
- Strong security
  - Public and Private key
  - Fine-grained access control
- Improved multi-platform support
- Extensible
  - Lays groundwork for migration/replication and global naming

The IETF process and NFS

1998

Sun/IETF Agreement

BOF, working group forms

Strawman Proposal from Sun

Meetings, writing, e-mail
Prototyping by 5 organizations

1999

Working Group Draft

Additional prototyping
Six working group drafts
Working Group Last Call
IETF Last Call
IESG Review
Assign RFC number

2000

2001

Proposed Standard RFC 3010

2002

Proposed Standard RFC 3530

Two independent implementations
6+ months

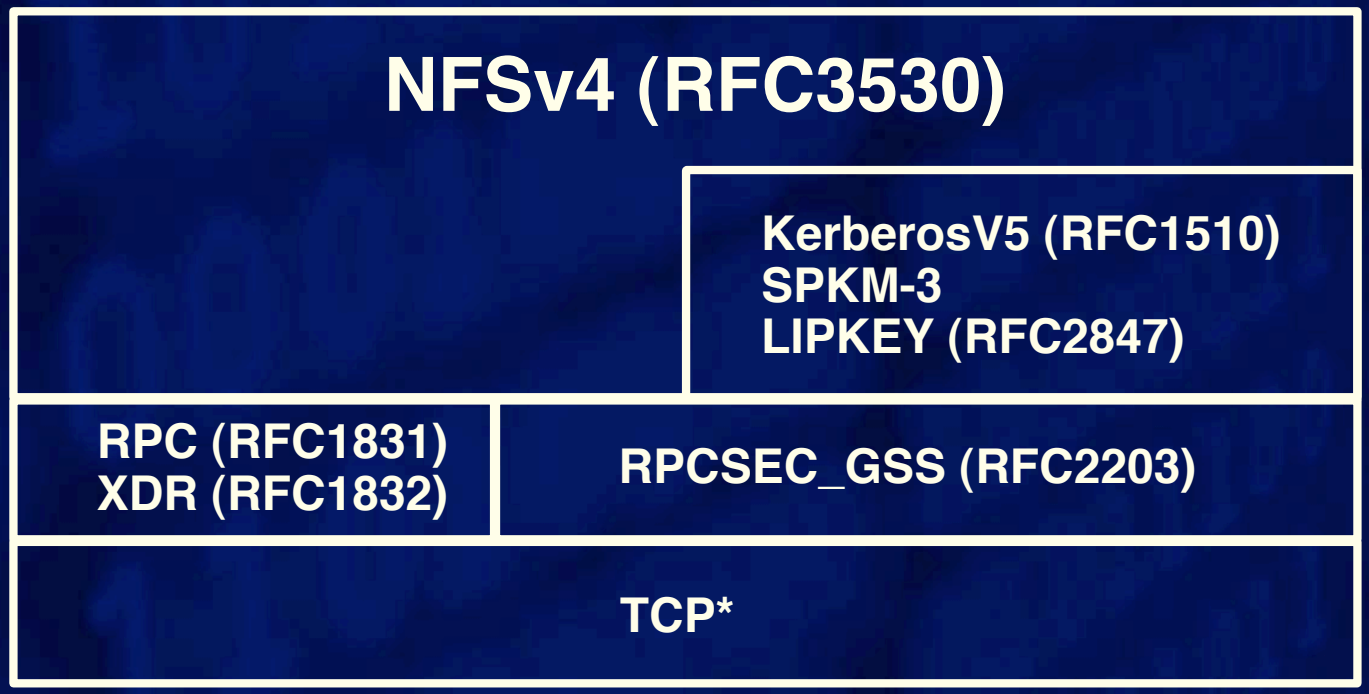
Draft Standard

Internet Standard apotheosis

Couple things we did right this time

- Open source reference implementations of NFS Version 4 were funded early (started by Sun)
- Interoperability events held 3 times a year
  - With focus of non-Connectathon events on NFS Version 4
- Huge improvements in execution and coordination over NFS Version 3

NFS Protocol Stack



NFS Version 4 operations

- ACCESS
- CLOSE
- COMMIT
- CREATE
- DELEGPURGE
- DELEGRETURN
- GETATTR
- GETFH
- LINK
- LOCK
- LOCKT
- LOCKU
- LOOKUP
- LOOKUPP
- NVERIFY
- OPEN
- OPENATTR
- OPEN_CONFIRM
- OPEN_DOWNGRADE
- PUTFH
- PUTPUBFH
- PUTROOTFH
- READ
- READDIR
- READLINK
- RENAME
- RESTOREFH
- SAVEFH
- SECINFO
- SETATTR
- SETCLIENTID
- SETCLIENTID_CONFIRM
- VERIFY
- WRITE
- RELEASE_LOCKOWNER

NFS operation aggregation

- The COMPOUND operation
  - NFS procedures are now groupable
- Potential for reduced latencies and roundtrip times
- Part of framework for minor versioning

Example: mount server:/test/dir

- **Client generates this COMPOUND**

PUTROOFH ← The operation formerly known as MOUNT
GETFH
LOOKUP(test)
GETFH
GETATTR
    SYMLINK_SUPPORT
    LINK_SUPPORT
    FH_EXPIRE_TYPE
    TYPE
    SUPPORTED_ATTRS
    FSID
SECINFO (dir)
LOOKUP (dir)
GETFH
GETATTR

NFS Version 4 is secure

- Mandatory to implement
  - Optional to use
  - Extensible via GSSAPI RPC
  - Kerberos V5 available now
  - Public key flavor emerging
- Security negotiated
  - Per file system policy
  - Continuous security
  - Security negotiation mechanisms
- Levels
  - Authentication
  - Integrity
  - Privacy
- ACLs (based on Windows NT)

Specification vs. implementation

- RFC 3530 defines required, recommended and optional features
  - The required features form core of interoperability
  - Recommended and optional features are negotiated
- ACLs, for example, are a “recommended” attribute
  - Not required for compliance
  - Dependent on underlying local file system support on server (on Linux - that’s a lot of file systems)
  - ACLs are ill-defined in *ix environs - mapping issues are tripping us up

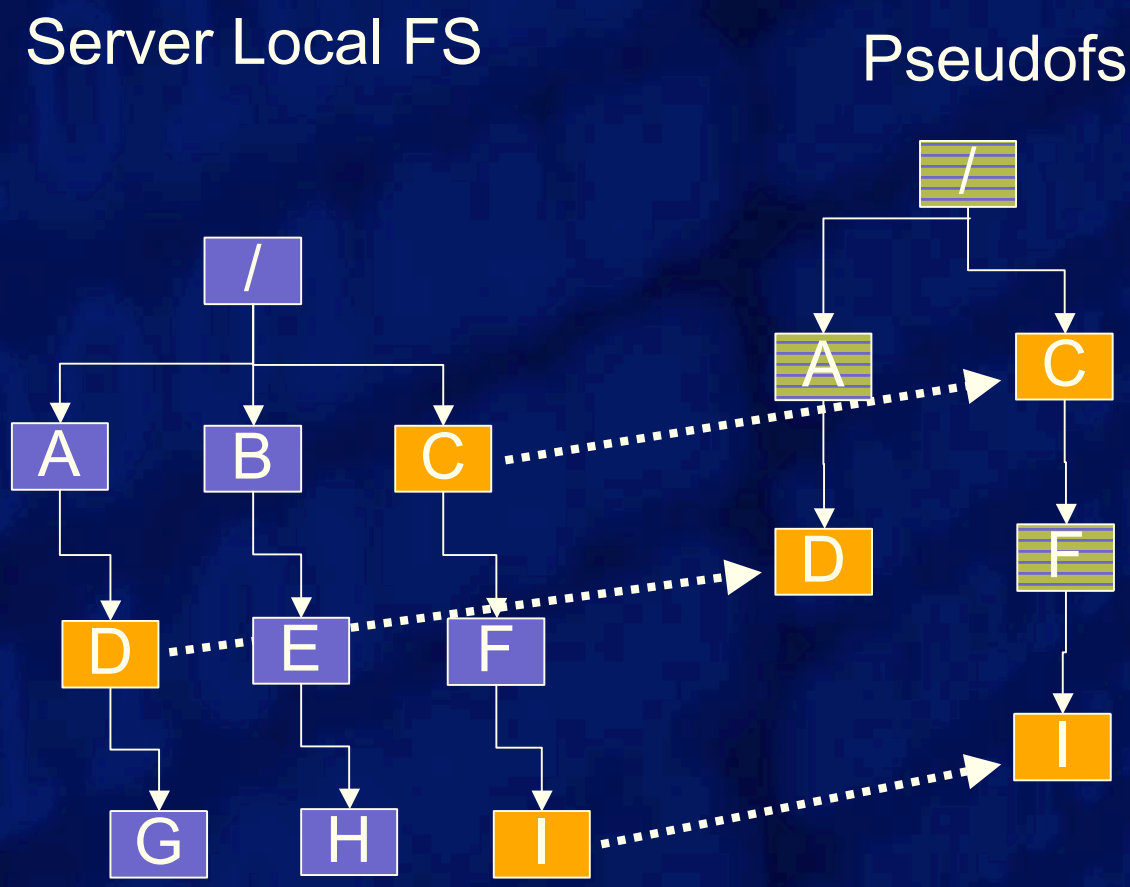
NFSv4 - Stateful

- Protocol is “session” oriented (OPEN call exists)
  - But in reality NFS Version 3 was also stateful via adjunct Locking Protocol
- Lease-based recovery of state (simplified error handling)
- File locking integrated into protocol
  - OPEN provides atomicity for Windows integration
- Addition of delegations
  - Client in absence of sharing allowed to locally cache data and locking state
  - This does not mean data sharing is defined in absence of explicit locking

Delegations (making lemonade)

- Use stateful design to enhance performance and scalability
- Enables aggressive caching on client
  - Shared read
  - Write exclusive
- Reduced roundtrips of the wire
  - Read, write, locking etc. cacheable locally
  - The fastest packet is the one never sent
- Server-based policy
  - Server manages conflicts
  - Sharing model reverts to NFS Version 3 when conflicted

The Pseudo file system



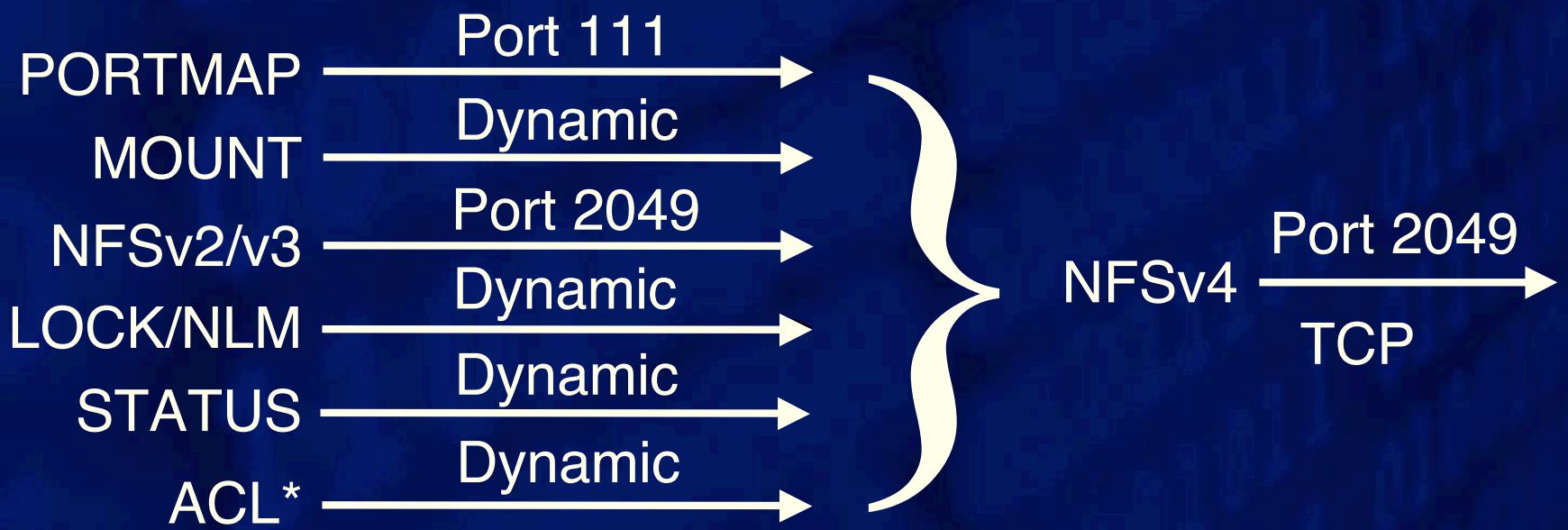
Protocol vs. Implementation II

- Administration amongst the remaining *ixes differ
- Recommended and optional features require negotiation
  - Security
  - Extensions

Simplified server namespace

- Exported file systems mountable from a single server root
  - Root filehandle for the top of the file tree
  - Still a client mount command
- For all shared/exported filesystems, server constructs pseudo filesystems to span name space
- Client can still individually mount portions of the name space
- Differing security policies can cover different parts of exported space

Firewall friendly



NFS Version 4 availability

- Network Appliance (Feb. ‘03), Hummingbird (late ‘02), Linux (via SuSE mid-’04), in RedHat Fedora (May ‘04), RHEL 4.0 Dec. 04)
  - Must be explicitly enabled
- Solaris 10 imminent (uh, yesterday)
  - On by default
- IBM AIX 5L V5.3
- BSD (Darwin) (date?)

The future of NFS Version 4

- Enhanced “Sessions” based NFS (correctness)
- CCM - session-based security for IPsec
- Directory delegations
- Migration/replication completion
  - Core protocol defines client/server failover behaviour
  - Definition of the server-server file system movement
  - Transparent reconfiguration from client viewpoint
- Proxy NFS file services (NFS caches)
- Uniform global name space
  - Features exist in the core protocol to support this
- Support for multi-realm security

Scalability: Attacking the I/O bottleneck

- Remote Direct Memory Access
  - Bypasses CPU on client and server for networking
  - Reduces the memory bottlenecks on high speed networks such as Infiniband or 10G ethernet.
    - See the NFSv4.1 RDMA extensions
- Parallel storage extensions (pNFS)
  - “Intelligent” clients are given (limited!) direct access to the storage net using block protocols such as iSCSI etc.
  - Bypasses server altogether for block file operations, but not for metadata

Drill down: Linux NFS

Linux 2.6 – Recent NFS (3 and 4) client changes

- Support for O_EXCL file creation
- Cached ACCESS permission checks
- “Intents” allow unnecessary lookups and permissions checks to be optimized away
- Asynchronous read/write improvements
  - Removed 256 page read/write request limit
  - Async support also for r/wsize < PAGE_SIZE
- DIRECT_IO / uncached I/O
- RPCSEC_GSS

Linux 2.6 NFS Version 4

- Framework added October ‘02 to Linux 2.5
- Additional cleanups and features added over past year
  - Performance work to get to V3 levels
    - Delegations
  - Framework for atomic OPEN
  - Memory management issues
  - Basic state management
    - State recovery
  - Bug fixes and code stabilization

Linux 2.6 NFS Version 4

- Goal is to stabilize basic V4 and add further advanced features
  - Will remain an EXPERIMENTAL feature dependent on testing
  - No NFS V4 ROOT (for diskless operation)
- Must address the user experience
  - State recovery under network partition
  - NFSv4 perceived as complicated to administer
    - Strong security, names spaces, migration/replication, etc.

Linux-2.6 – Generic server changes

- Adds upcall mechanism
  - Adds RPCSEC_GSS support (version 3 and 4)
  - Id mapper
  - Improves support for exports to NIS or DNS domains.
- Zero-copy NIC support
  - Accelerates NFS READ calls over UDP+TCP
- 32K r/wsize support

Linux 2.6.10 NFS V4 - What's in

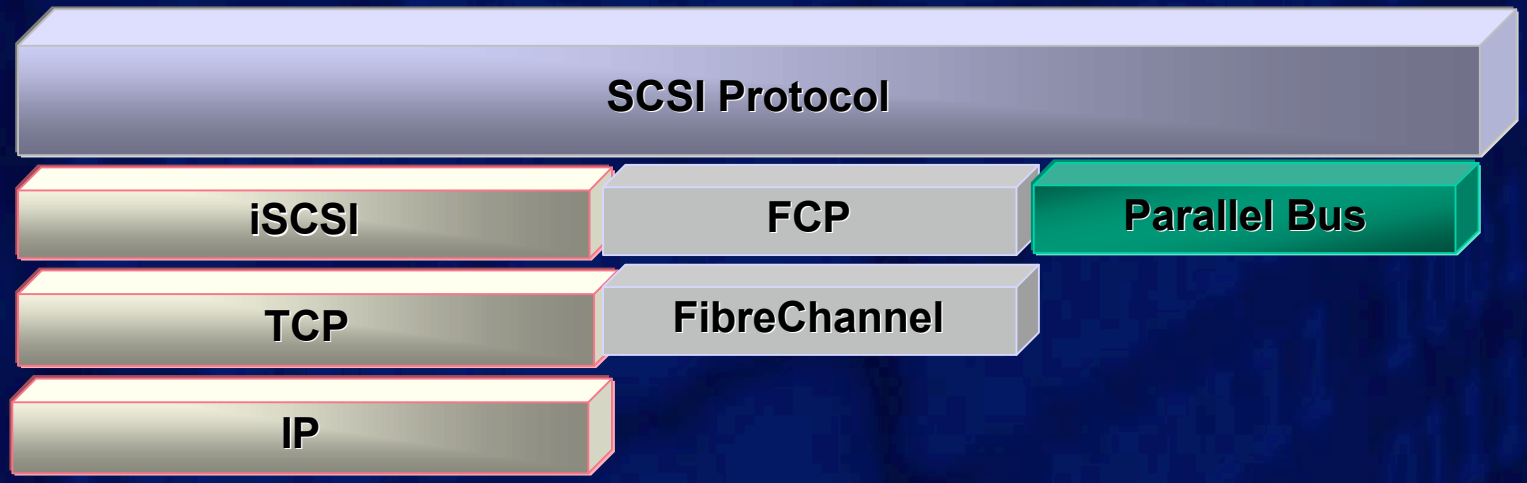
- All NFS ops and COMPOUND
- GSSAPI
  - LIPKEY/SPKM and Kerberos (authentication and integrity)
- Single server mount
- Locking
- Client side delegations (read and write)
  - Does not cache byte-range locks
- ID mapping (UID to names - user@domain)
- O_DIRECT bug fixes
- AutoFS support

Linux 2.6.10 NFS V4 - TODO

- Security
  - RPCSEC_GSS privacy support
  - SECINFO
  - Keyrings
- Server side delegation support (going in now)
- Migration/replication client support
  - “Nohide” mount point crossing - RFC 3530 compliance
- ACL support
- Named attribute support
- Non-Posix server support (missing attribute simulation)
- Volatile file handle handling
- Global name space

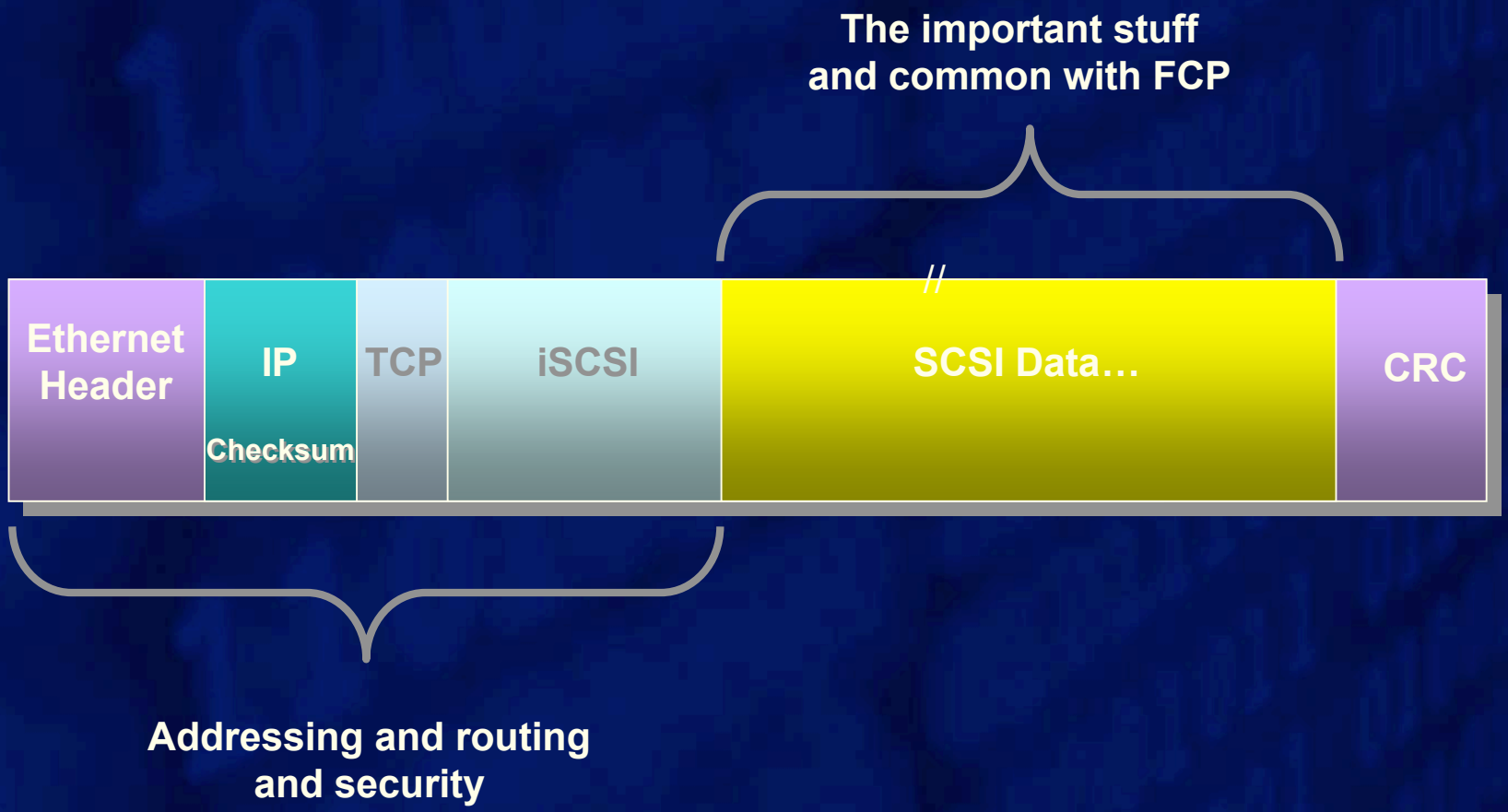
What about iSCSI?

iSCSI background



- TCP/IP Transport for SCSI command sets
  - SCSI block protocol access
- Internet standard - RFC 3720
- iSCSI is a direct replacement for FCP
  - FCP is the SAN fabric today

The important thing about iSCSI is SCSI



NetworkAppliance®

“So, iSCSI is a replacement for NFS, right?”

- In the first iSCSI presentation I made to a prospect, this was first thing out IT manager’s mouth
- I used to say “No”, but the thoughts underlying the question are interesting

iSCSI value proposition

- Leverage existing Gigabit → 10 Gigabit networking infrastructure
- Leverage existing rich set of management tools
- Leverage existing base of skilled personnel

Reducing costs

iSCSI points

- iSCSI software drivers freely and ubiquitously available
  - Windows platforms
  - Linux, and other *ixes
- HBAs and TOEs
  - Scale performance from software solution → HW assist → full offload
- Saying “performance” and “iSCSI” in the same breath though misses the point
  - Performance is not always the primary issue (else use FC SAN)
  - Many application deployments have spare (CPU and I/O) capacity
  - Optimize performance as needed

For some, iSCSI represents the path of least resistance

- It is semantically equivalent to FC SAN (SCSI)
  - But more familiar because of TCP/IP and Ethernet - so friendly outside the data center
- Application migration is trivial
  - My remote booting desktop from FC to iSCSI
- Provides a path for easily reclaiming FC port capacity by moving less critical apps to iSCSI
- With some of the important cost benefits of NAS

iSCSI is part of a solution

- Cost effective alternative to captured storage
- Windows HCL afterburner
- A place to move “less critical” SAN applications freeing up FC capacity
  - The early adopter approach
- Friendly outside the data center
  - And manageable!
- Easily envision applications like remote C: drive
  - Remind me to tell you a story
- An additional tool in the storage toolbox

Cluster computing and Linux

The Old Way

Imagine Charlton Heston in a chariot.

The New Way

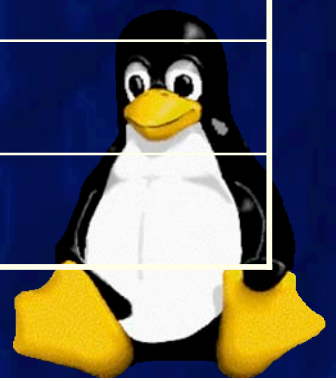
Imagine an airplane full of chickens.

“The GRID” – what is it?

- Sets of toolkits or technologies to construct pools of shared computing resources
  - Screen-savers that use millions of desktop machines to analyse radio telescope data.
  - Vertical applications in the enterprise on a large Linux cluster.
  - Middleware that ties together geographically separated computing centres to satisfy the exponentially increasing demands of scientific research.
- To many, “Grid computing” and “Cluster computing” are synonymous.

Modern numerology (this is not important)

86	The preferred architecture for commodity computing (and oddly, the number of years it took the Red Sox to win the World Series)
2	Number of physical processors in commodity pizza boxes (poor man's blade)
128	Maximum expected nodes in a Linux database cluster
2000	Typical number of Linux nodes in a render or ECAD simulation farm today
10000	Expected number of nodes in Linux compute cluster in next two years?
<9	Number of filers per 1000 Linux nodes in GRID
1	There is only one - Linus Torvalds
10?	The number of trusted minions to Linus



Scalable compute cluster

- Linux is ahead of the game
  - growing infrastructure, expertise and support
- It's all about choice!
- No! It’s all about freedom!
- Well, no actually, it’s all about cost.

The rise of the Linux compute cluster

- Driven by cost
- Particularly the cost of cheap commodity CPUs
  - Monolithic application servers have no chance competing with the Tflop/price ratio
- Choices exist
  - *BSD development marches on
  - Sun is renewing its investment in Solaris x86
  - Even Windows has a market share

Compute cluster points

- The x86 platform won
  - Any questions?
- Support costs may still be significant
  - ...but largely offset by the hardware cost savings - it's about leveraging small MP commodity x86 hardware
  - Some customers choose to pay more for better quality in order to lower support costs and improve performance
  - Maturation of "free software" - paying for support
- For Unix environments, NFS is the cluster file sharing protocol of choice
  - Customers simply want storage solutions that scale as easily as their compute clusters

Applications: Batching

- Large partitionable tasks that may run as parallel threads
  - Clusters may include up to several thousand nodes
  - Often uses LSF and other queueing tools to manage jobs
  - Read-only data may be shared, but writing is partitioned to avoid locking issues.
  - Typical applications include search engines, e-mail, animation and rendering, scientific simulations, and engineering

Other applications

- Scalable databases
  - Fewer nodes than the batch case: up to a hundred or so.
  - High degree of write sharing necessitates heavier use of locking
  - Data integrity is often supported by specialized fencing and recovery techniques that may again need support in the underlying filesystem.
- Extremely I/O intensive high performance computing
- GRID computing

Cluster-friendly features in NFSv4

- Stateful model
  - ...but recovery remains client driven!
  - in case of a recoverable server outage, clients just retry operations until they succeed
- Leases solve NFS Version 3 state leakage problems due to client outages
  - Cures the “lost locks” syndrome
  - But introduces new issues on network partition
- Delegations allow for aggressive caching
  - Stops short of a full coherency model, though
  - Callback mechanism is firewall-unfriendly.

Cluster-friendly features in NFSv4

- GRID friendly
  - Obligatory support for strong security makes NFS deployment over the internet possible
    - Including public key
    - Includes support for data encryption
- Firewall-friendly: only port 2049 needs to be opened
    - there are no side-band mount/lock/... protocols
    - callbacks/delegations are not mandatory

NFS in the short term future

- Aim to provide robust global name spaces
- Adaptations to work with GRIDs
  - GRIDNFS – project to adapt NFS to the Globus GRID toolkit
  - In the long run, we need improved caching models for use with high latency environments
- Performance improvements using hardware assisted networking
  - NFSv4.1 includes support for RDMA

Storage in a clustered environment

- "Scale my storage as easily as I can scale my CPUs."
- Data sharing
  - Data must be accessible to all compute nodes
- (high) data availability
  - No single point of failure
- Reliable data handling
  - No data corruption
- Security
  - In particular secure transport of data between compute and storage nodes
- Support commodity TCP/IP and ethernet networking
- Performance

Scaling yet further

- Virtualization techniques permit horizontal scaling of storage using the current protocol
  - See NetApp/Spinnaker
- Parallel NFS – NFSv4 extensions to scale beyond storage network bandwidth limits
  - Allow for striping files across several storage units
  - Explore SAN and object storage integration (NFSv4 as metadata server)

Putting NFS in perspective

Let's put this in perspective

- "Wow. Michaelangelo, great statue - was that a 7 inch chisel you used?"
- "Great flick Welles, what camera did you use?"
- "Great quarter you guys had! Did you use NFS to access your financial data?"

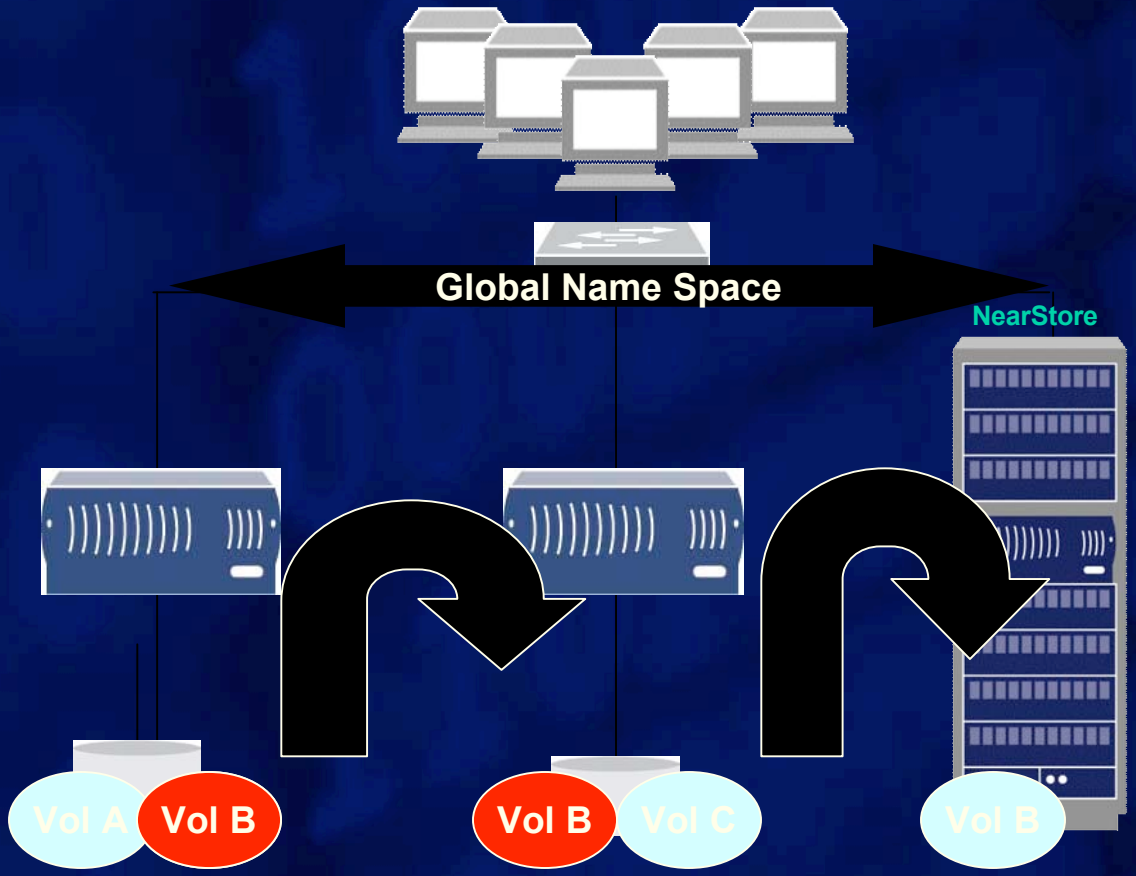
It's about applications

- Applications drive storage choices
  - What does the application vendor support?
  - What do they recommend?
  - For example, Exchange is driving iSCSI in the Windows environment
- Mix of applications in a single enterprise
  - There is no one perfect storage approach
  - There's likely more than one vendor

It's about data management

- Integration of applications with data management
  - Key applications like Exchange - application-driven backup/restore
  - Fertile ground for virtualization - blurring line between client application and storage
- Disaster recovery
- Finding data when you need it
  - Higher level data organization and grouping?

Non Disruptive Migration



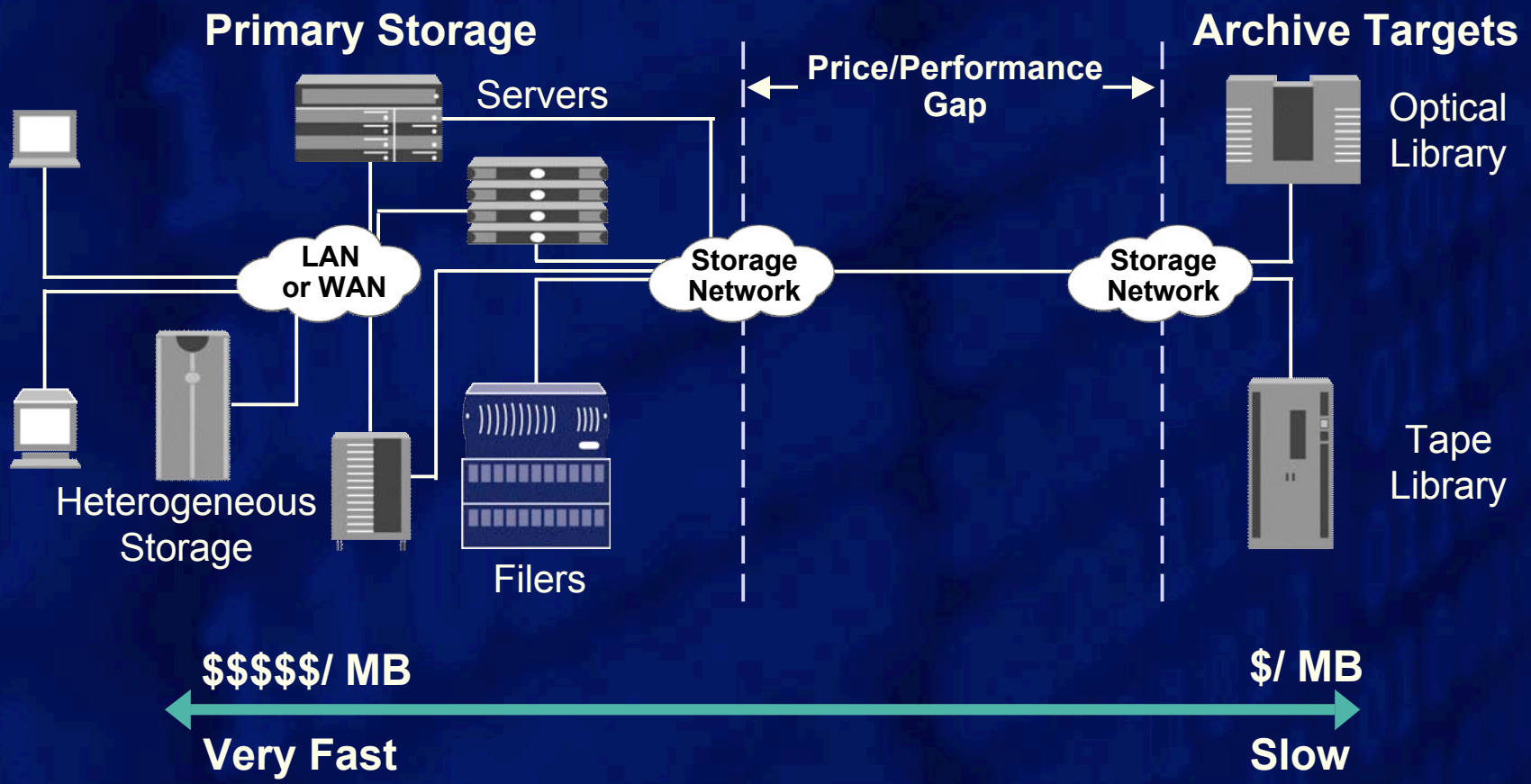
Migration:

- Fast (on-demand)
- Transparent to users
- Migration of Name Space or backend Volumes
- Migrate aged data to NearStore

It's about cost

- Ability to (re)provision, expand and manage storage to maintain high utilization will most affect overall cost long term
- Leveraging commodity networking
  - iSCSI and NFS are similar here
- Primary storage and Nearline support for all storage access - transparently
  - Migration and replication
- Consolidation to reduce management costs

Existing Storage Hierarchy



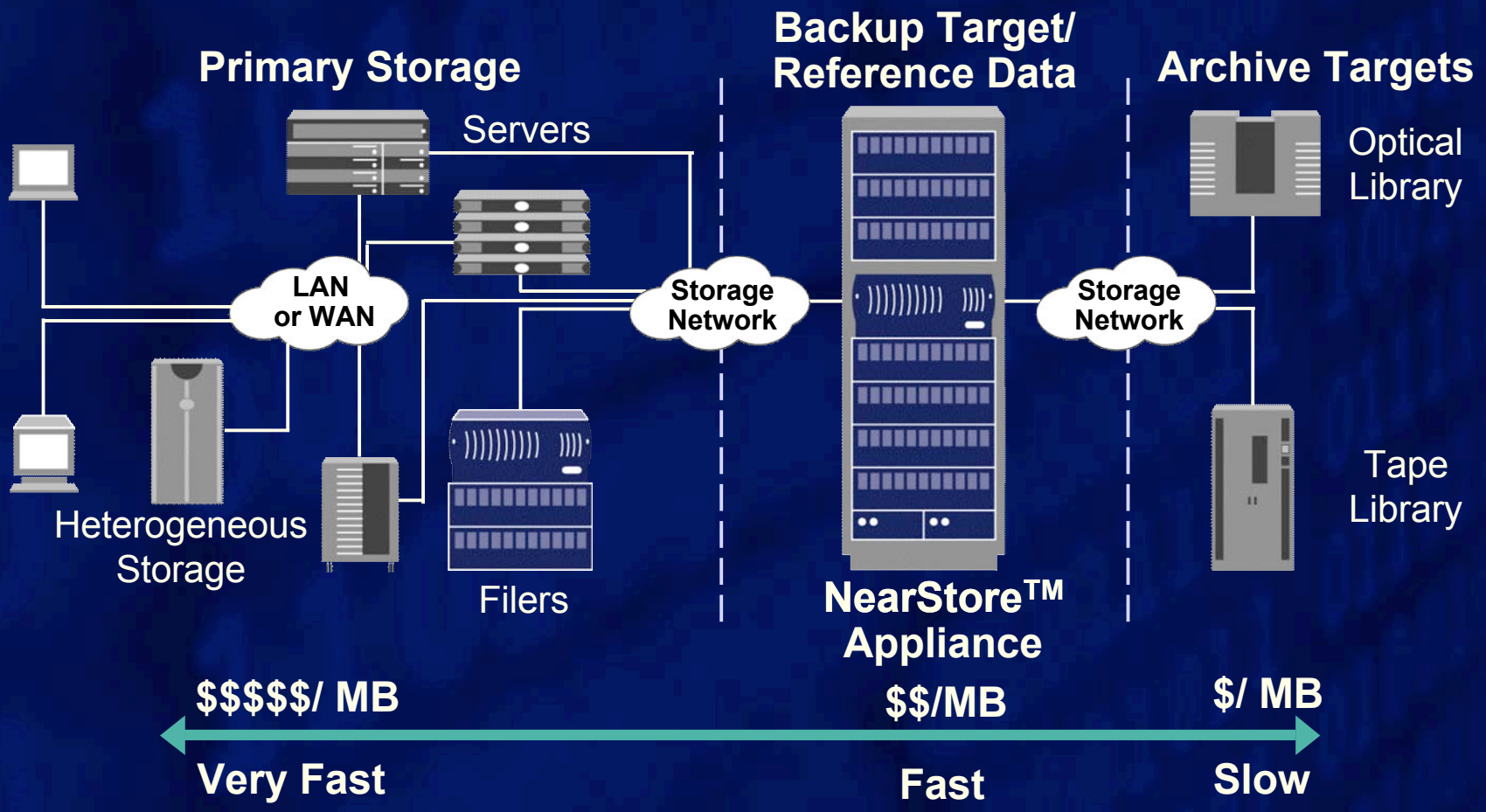
- The traditional two-tier hierarchy creates a large price/performance gap
- Current challenges need a storage solution that fills the gap

NetworkAppliance®

Economics of recovery

- Acceptable downtime is application dependent
- Simplest – but most costly - approach is full mirror online
- ATA drives are cheap

Emerging Storage Hierarchy



NetworkAppliance®

Understanding the context around NFS

- Regardless of data access protocol, similar issues and solutions in data management
  - Common management regardless of access method
- That other operating systems (besides Linux) drive fundamental architecture decisions
  - Blade provisioning via NFS is a non-starter perhaps - because of multi-OS support
  - Enter iSCSI - the least common denominator - the Windows splash effect
- People don't buy NFS servers
  - They buy Oracle or other applications
  - They build application compute clusters
  - And manage the data around it - with NFS perhaps

beepy, are you saying there is no difference between storage architectures?

Differences are important

- NAS protocols define a file view - higher level organization and semantics
  - Enables sharing
  - Enables large compute clusters (>5,000 nodes)
- iSCSI, like FC SANs, provides simpler SCSI block interface
  - Higher level semantics via explicit file system encapsulation
  - Sharing via layered cluster file system (complexity and cost?)
- Customers will use and continue to explore a variety of approaches

The challenge for NFS

Other than that Mrs. Lincoln…

- NFS = Network File System
- NFS = Not For Speed
- NFS = Not For Security

But an NFS vendor may sit there and think “My side of the boat is dry!”

Exactly.

First impressions last a long time, and hard to turnaround once set.

So, back to NFS - what do customers want?

- No surprises
  - Customers really want a better NFS Version 3
  - Are we prepared to provide support for NFS Version 4?
- Reliability
  - Testing
  - Scalability
- Playing well with others
  - Agreeing on common administration models
  - Agreeing on common features (else we will drop things from spec in IETF)
- Security
  - Administration needs to be simplified simplified simplified
- Performance is at bottom of list I think

Additional Resources

- NFS
  - http://www.netapp.com/tech_library/ftp/3183.pdf (Linux NFS)
  - http://www.netapp.com/tech_library/hitz94.html (NFS Version 3)
  - http://www.beepy.com/SANEnfs.pdf (NFS and Linux clusters)
- NFS Version 4
  - http://www.ietf.org/rfc/rfc3530.txt
  - http://www.nluug.nl/events/sane2000/papers/pawlowski.pdf (OLD)
  - http://www.redbooks.ibm.com/abstracts/SG247204.html
  - http://www.nfsv4.org
- SAN and iSCSI case studies
  - http://www.netapp.com/case_studies/index_prod.html#san-nas
- SAN Performance Technical Report
  - http://www.netapp.com/tech_library/3321.html

Questions