Education

pNFS & NFSv4.2; a filesystem for grid, virtualization and database

Alex McDonald, NetApp
Co-Chair SNIA NFS SIG

Author: Joshua Konkle, NetApp

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Abstract

- pNFS & NFSv4.2; a filesystem for grid, virtualization and database
  - This session will appeal to Virtual Data Center Managers, Database Server administrators, and those that are seeking a fundamental understanding pNFS. This session will cover the four key reasons to start working with NFSv4 today, and explain the storage layouts for parallel NFS; NFSv4.1 and the upcoming NFSv4.2 standard. The session includes use cases for database access, enterprise and desktop virtualization, including deduplication options.

Tutorial Agenda

- Introduction to NFS and NFS Special Interest Group
- NFS v4 – Security, High Availability, Internationalization and Performance (SHIP)
- pNFS – Layout Overview
  - Files based access
  - Block based access
  - Object based access
- pNFS – OpenSource Client Status
- pNFS Use Cases – Virtualization, Database, etc

SNIA's NFS Special Interest Group

- NFS SIG drives adoption and understanding of pNFS across vendors to constituents
  - Marketing, industry adoption, Open Source updates
- NetApp, EMC, Panasas and Sun founders
  - NetApp, EMC and Panasas act as co-chairs
- White paper on migration from NFSv3 to NFSv4
  - "Migrating from NFSv3 to NFSv4"

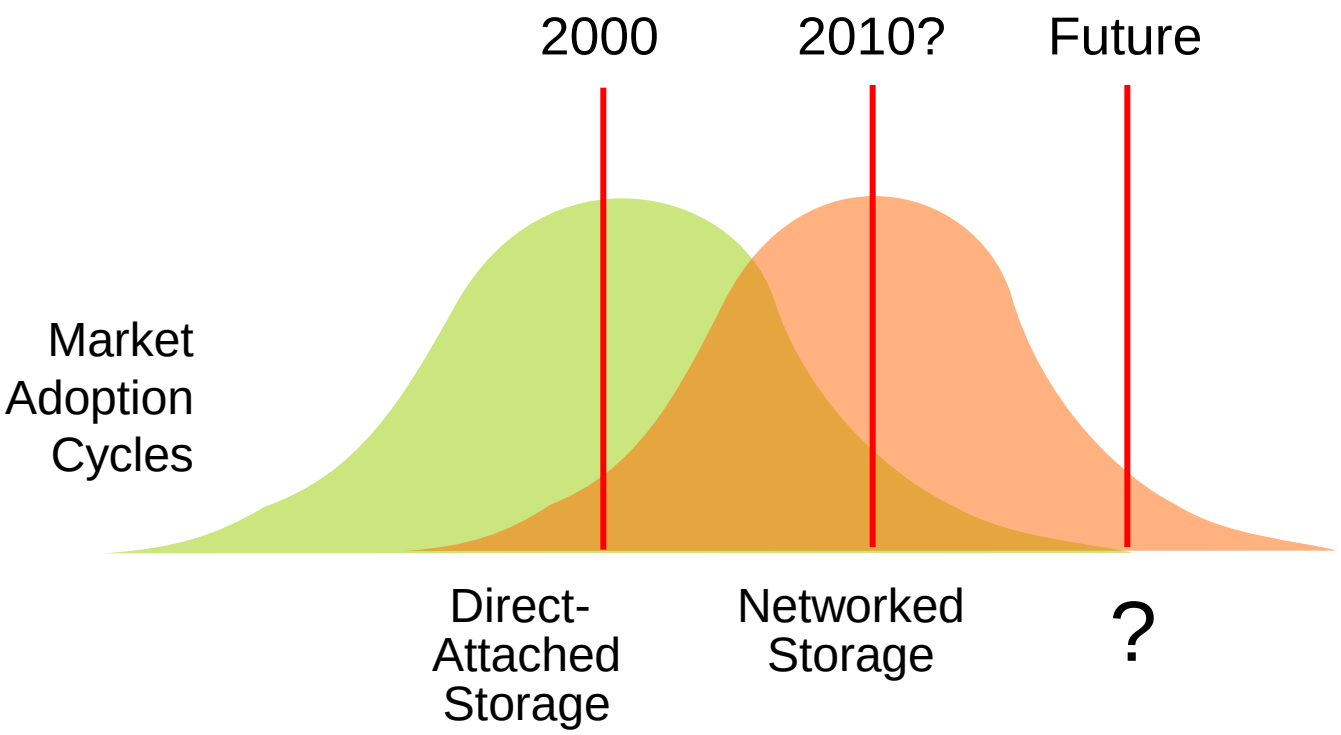
SNIA Ethernet Storage Forum

Learn more about us at: www.snia.org/forums/esf

Background Information

- Network File System
  - Protocol to make data stored on file servers available to any computer on a network
  - NFS clients are included in all commonly used Operating Systems, e.g. Linux, Solaris, AIX, Windows etc…..
  - Application and OSI layers (remote procedure calls)
- NFS Server; Inspiration to NAS and appliances
  - Commodity Operating Systems have NFS servers
  - NAS Appliance – Control, Consistency and Cadence
  - Vendors offer commodity hardware, w/ management software

The Evolution of Storage

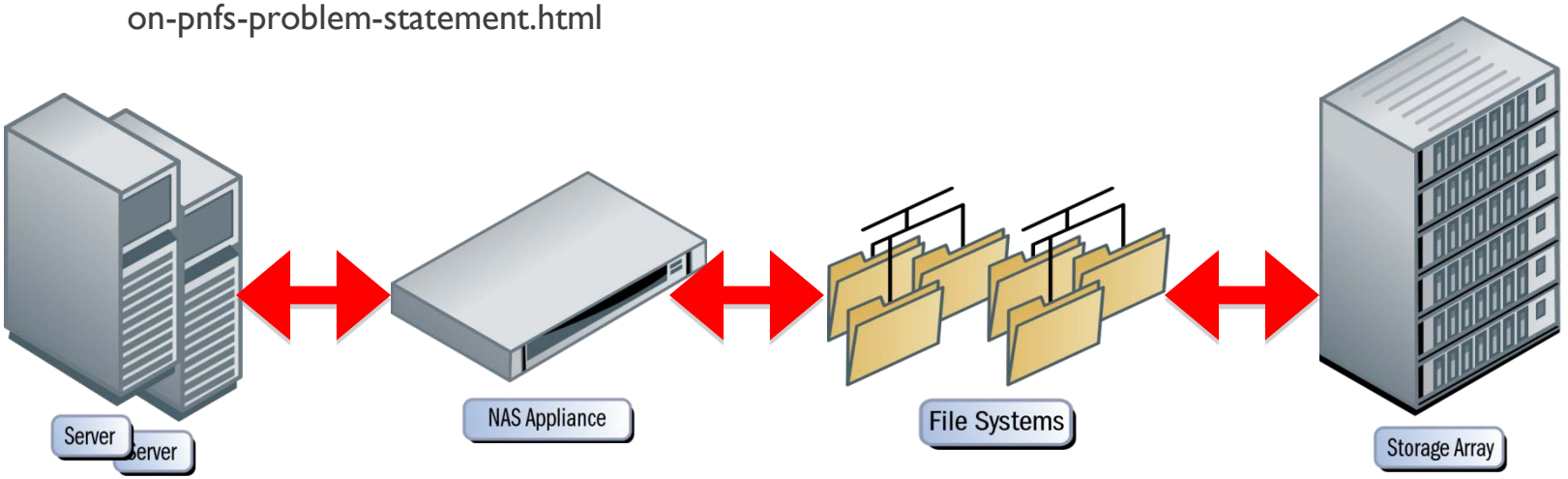


Evolving Requirements

- Economic Trends
  - Cheap and fast computing clusters
  - Cheap and fast network (1GbE to 10GbE, 40GbE and 100GbE in the datacenter)
  - Cost effective & performant storage based on Flash & SATA
- Performance
  - Exposes single threaded bottlenecks in applications
  - Increased demands of compute parallelism and consequent data parallelism
- Powerful compute systems
  - Analysis begets more data, at exponential rates
  - Competitive edge (ops/sec)
- Business requirement to reduce solution times
  - Beyond performance; NFS 4.1 brings increased scale & flexibility

NFS – What's the problem?

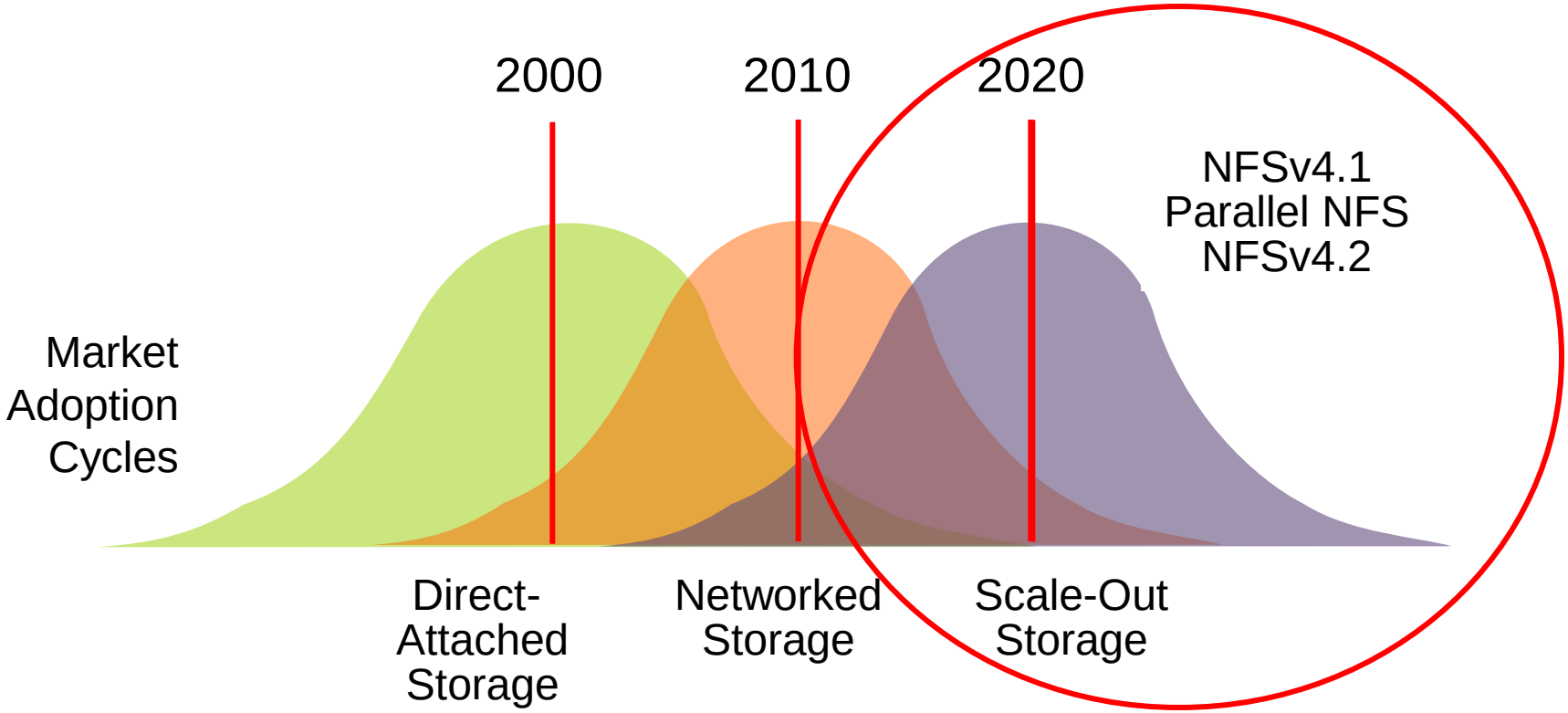
- In-band data access model
  - Easy to build, Limited in scale
  - Well-defined failure modes
  - Limited load balancing options
  - Garth Gibson (Panasas), Peter Corbett (Netapp), Internet-draft, July 2004 http://www.pdl.cmu.edu/pNFS/archive/gibson-pnfs-problem-statement.html
- Results in Limitations
  - Islands of storage
  - Server and Appliance HW
  - Networking and I/O



Performance, Management and Reliability

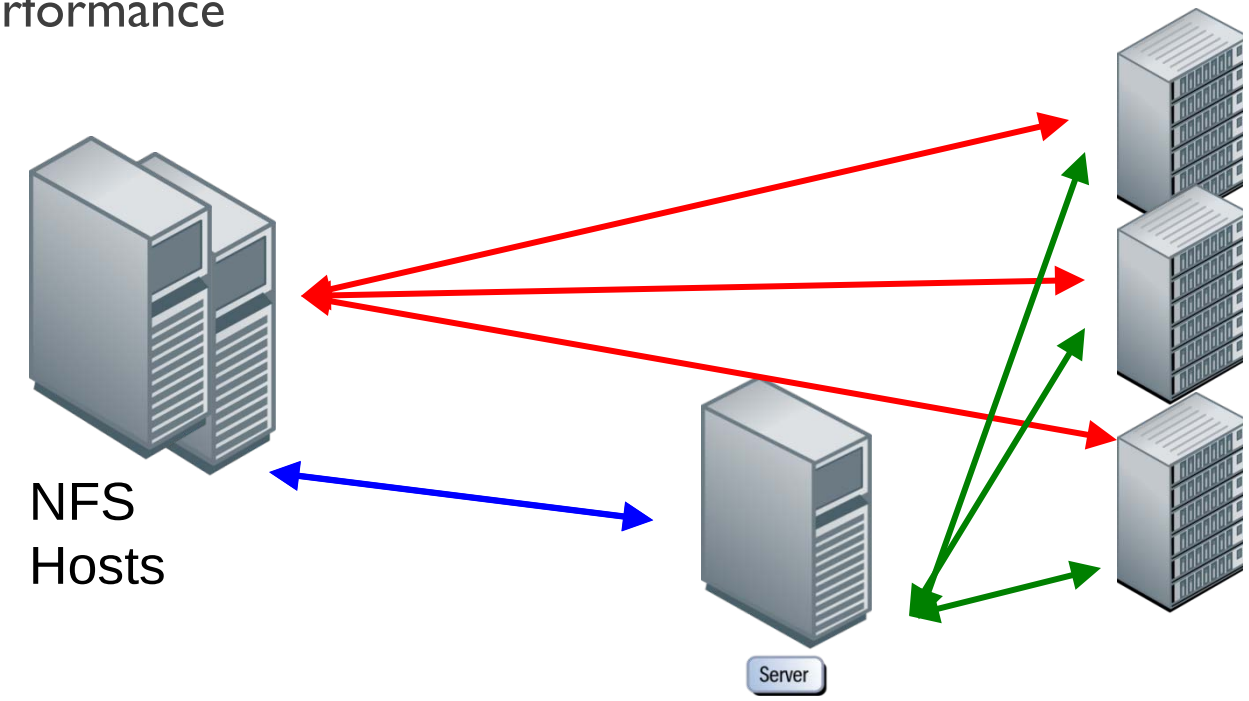
- Random I/O and Metadata intensive workloads
  - Memory and CPU are hot spots
  - Load balancing limited to pair of NFS heads; originally designed for HA
    - Not a limitation of the NFS 4.1 protocol
- Compute farms are growing larger in size
  - NFS head can handle a 1000+ NFS clients
  - NFS head hardware comparable to client CPU, I/O, Memory
  - NFS head requires more spindles to distribute the I/O
- Reliability and availability are challenging
  - Data striping limited to single head and disks
  - Non-disruptive upgrades affect dual-head configurations
  - Access and connectivity is typically limited to a pair of NFS server heads

What is the Solution?



NFSv4.1 – Parallel Data Storage

- Results in Improvements
  - Global Name Space
  - Head and Storage scaling
  - Non disruptive upgrades while maintaining performance
- NFSv4.1 – Three Storage Types
  - Files – NFSv4.1
  - Blocks – SCSI
  - Objects – OSD T10



NFSv4 SHIP is sailing

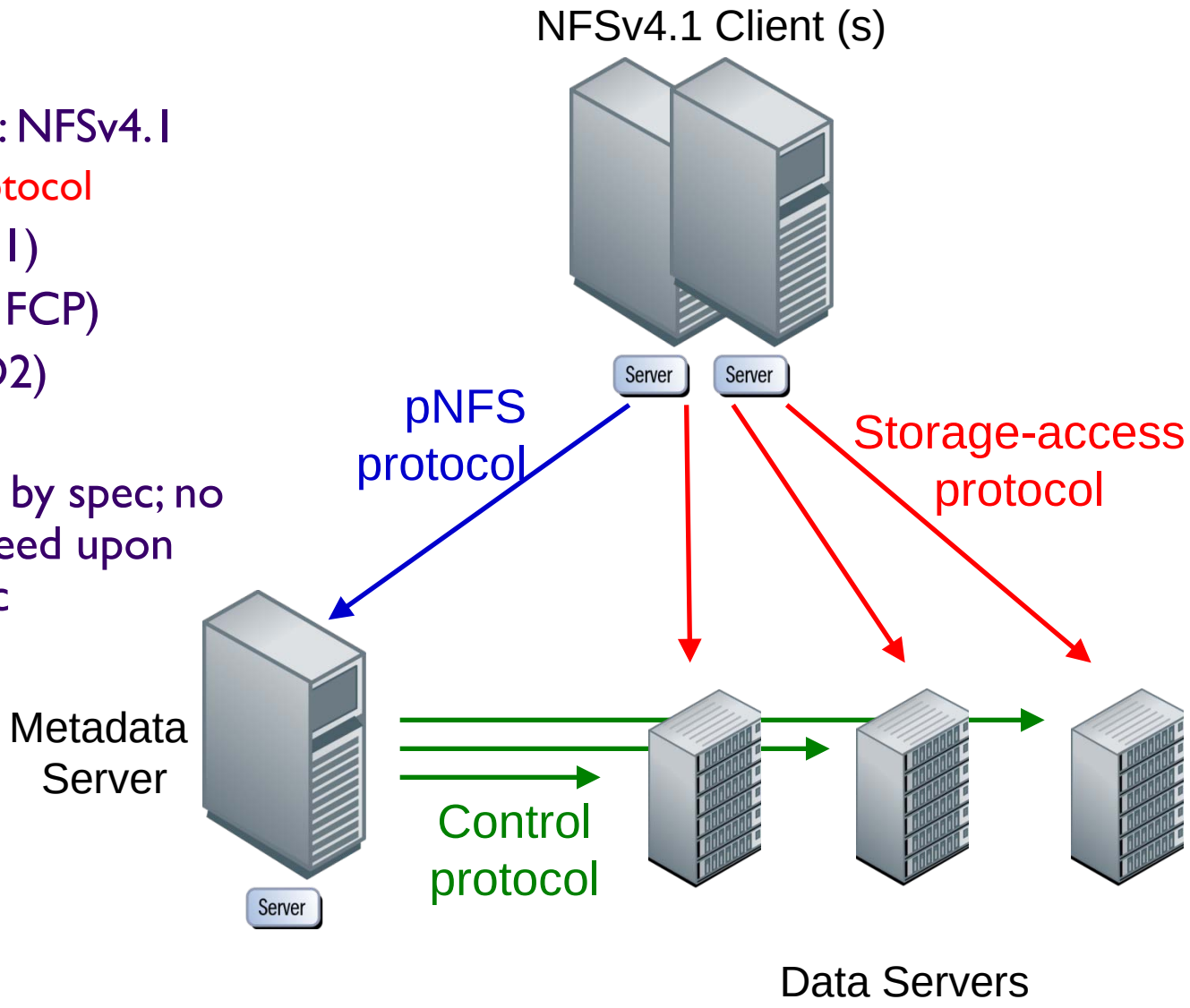
	Functional	Business Benefit
Security	ACLs for authorization Kerberos for authentication	Compliance, improved access, storage efficiency
High availability	Client and server lease management with fail over	High Availability, Operations simplicity, cost containment
International characters	Unicode support for UTF-8 codepoints	Global file system for multi-national organizations
Performance	Multiple read, write, delete operations per RPC call	Better network utilization for all NFS clients
	Delegate locks, read and write procedures to clients	Leverage NFS client hardware for better I/O

NFSv4 - HA and Performance

- High Availability via Leased Lock
  - Client renews lease on server file lock @ n Seconds
  - Client fails, lock is not renewed, server releases lock
  - Server fails, on reboot all files locked for n Seconds
    - Gives clients an n Second grace period to reclaim locks
- Performance via Delegations
  - File Delegations allow client workloads for single writer and multiple reader
  - Clients can perform all reads/writes in local client cache
  - Delegations are leased and must be renewed
  - Delegations reduce lease lock renewal traffic

NFSv4.1 - Parallel NFS 101

- pNFS protocol
  - Standardized: NFSv4.1
- Storage-access protocol
  - Files (NFSv4.1)
  - Block (iSCSI, FCP)
  - Object (OSD2)
- Control protocol
  - Not covered by spec; no generally agreed upon characteristic

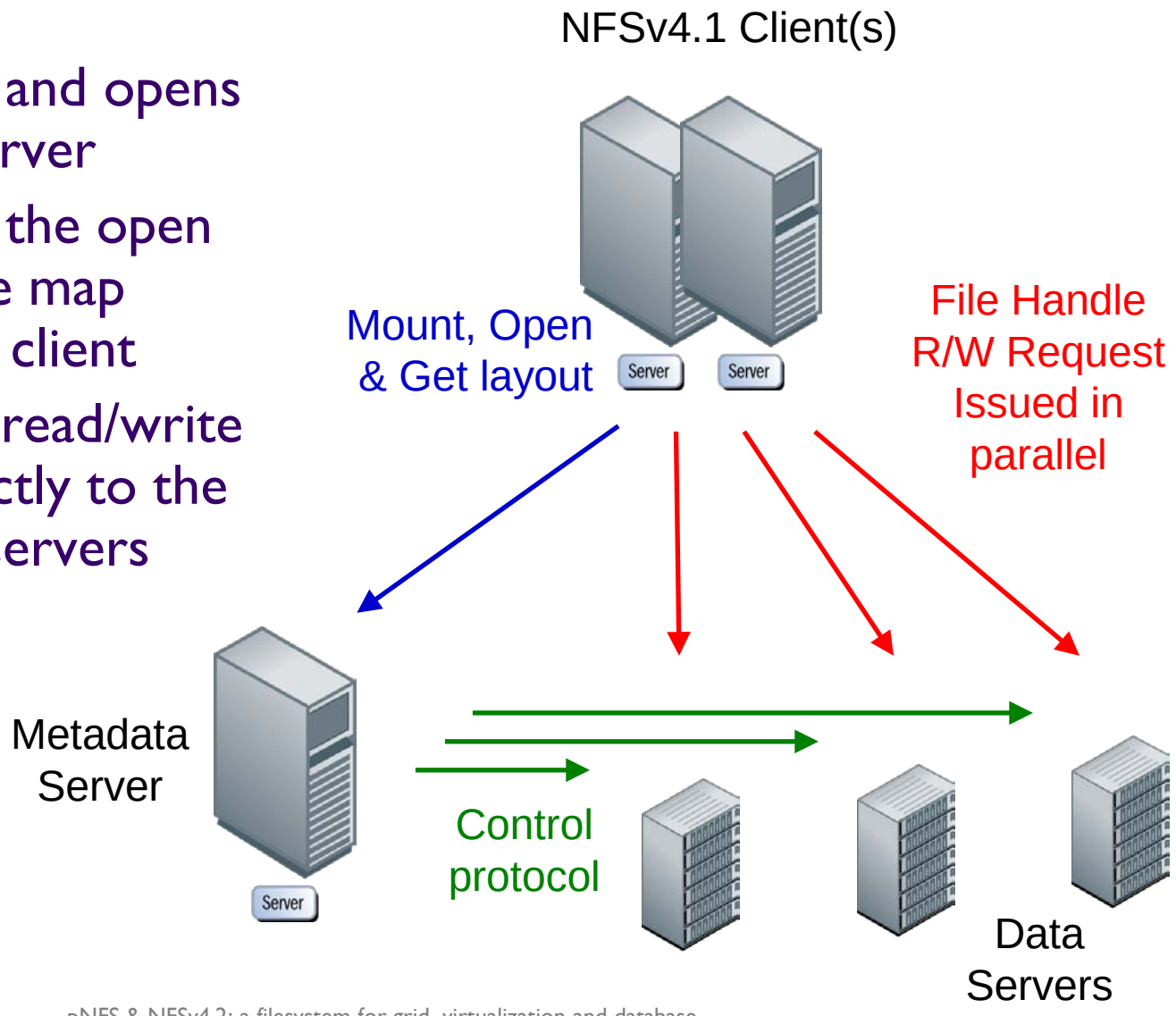


pNFS Operations

- GETDEVICEINFO
  - Client gets updated information on a data server in the storage cluster
- GETDEVICELIST
  - Clients requests the list of all data servers participating in the storage cluster
- LAYOUTGET
  - Obtains the data server map from the meta-data server
- LAYOUTCOMMIT
  - Servers commit the layout and update the meta-data maps
- LAYOUTRETURN
  - Returns the layout; Or the new layout, if the data is modified
- CB_LAYOUT
  - Server recalls the data layout from a client; if conflicts are detected

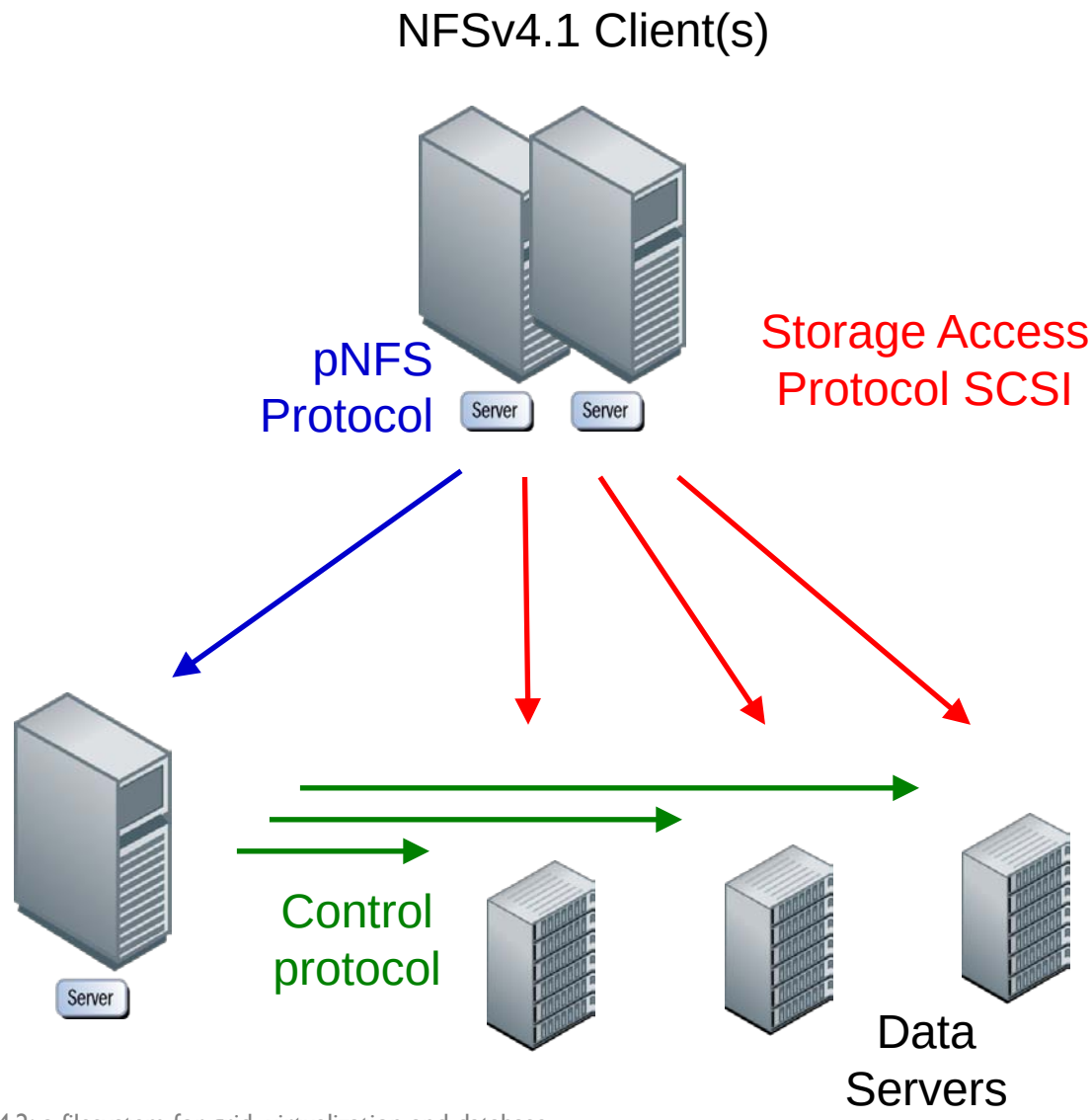
pNFS – NFSv4.1 files access

- Client mounts and opens a file on the server
- Servers grants the open and a file stripe map (layout) to the client
- The client can read/write in parallel directly to the NFSv4.1 data servers



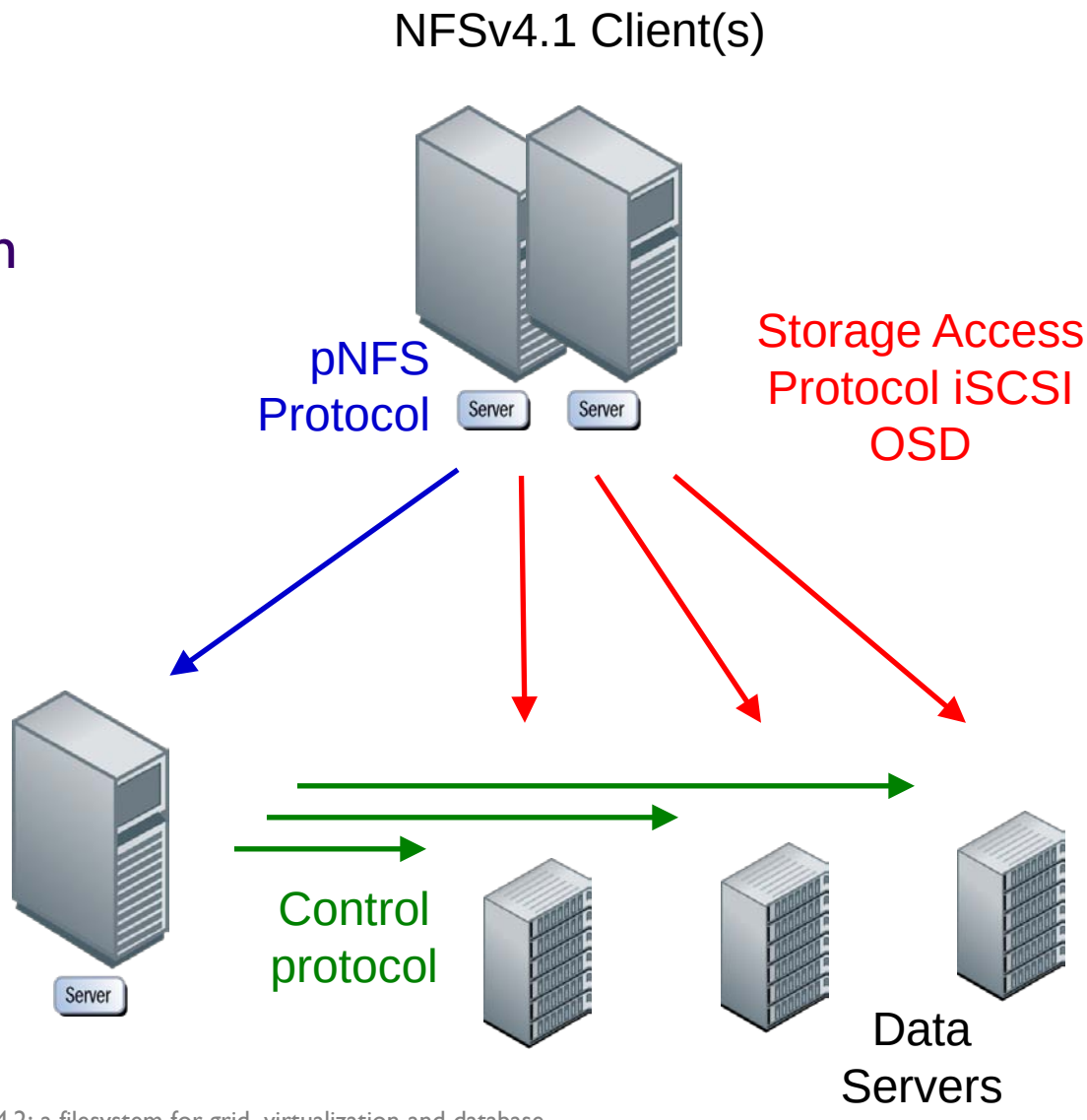
pNFS – Blocks Access Model

- Client mounts and opens a file on the server
- Server grants the open and a block map (layout) to the client
- Based on the layout obtained (read or write); the client can read/write in parallel directly to the SCSI targets



pNFS – Objects Access Model

- Client mounts and opens Object
- Server grants the open and an object stripe map and object capabilities (layout) to the client
- Based on the layout obtained (read or write); the client can read/write in parallel directly to the OSD targets

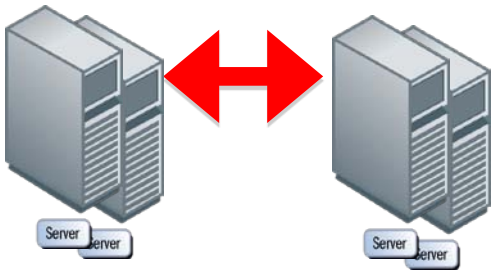


NFSv4.1 – OpenSource Status

- Two OpenSource Implementations
  - OpenSolaris and Linux
- Upstream (Linus) Linux NFSv4.1 client support
  - Basic client in Kernel 2.6.32
  - pNFS support (files layout type) in Kernel 2.6.39
  - Support for the 'objects' and 'blocks' layouts was merged in Kernel 3.0 and 3.1 respectively
- Full read and write support for all three layout types in the upstream kernel,
  - O_DIRECT reads and writes are not yet supported.
- pNFS client support in distributions
  - Fedora 15 was first for pNFS files
  - Kernel 2.6.40 (released August 2011)
- Red Hat Enterprise Linux version 6.2
  - "Technical preview" support for NFSv4.1 and for the pNFS files layout type
- Other Open Source
  - Microsoft NFSv4.1 Windows client from CITI

NFSv4.2 Major features: SSC

- Server-side copy: (SSC) Removes one leg of the copy.
  - If we have a client, src, and dest, then:
    - `cp /src/foo.db /dest/foo.db`
  - Involves two network traversals for each packet; read from the source and write to the destination
- With Server-Side Copy, destination reads directly from the source



NFSv4.2 Major features: ADB

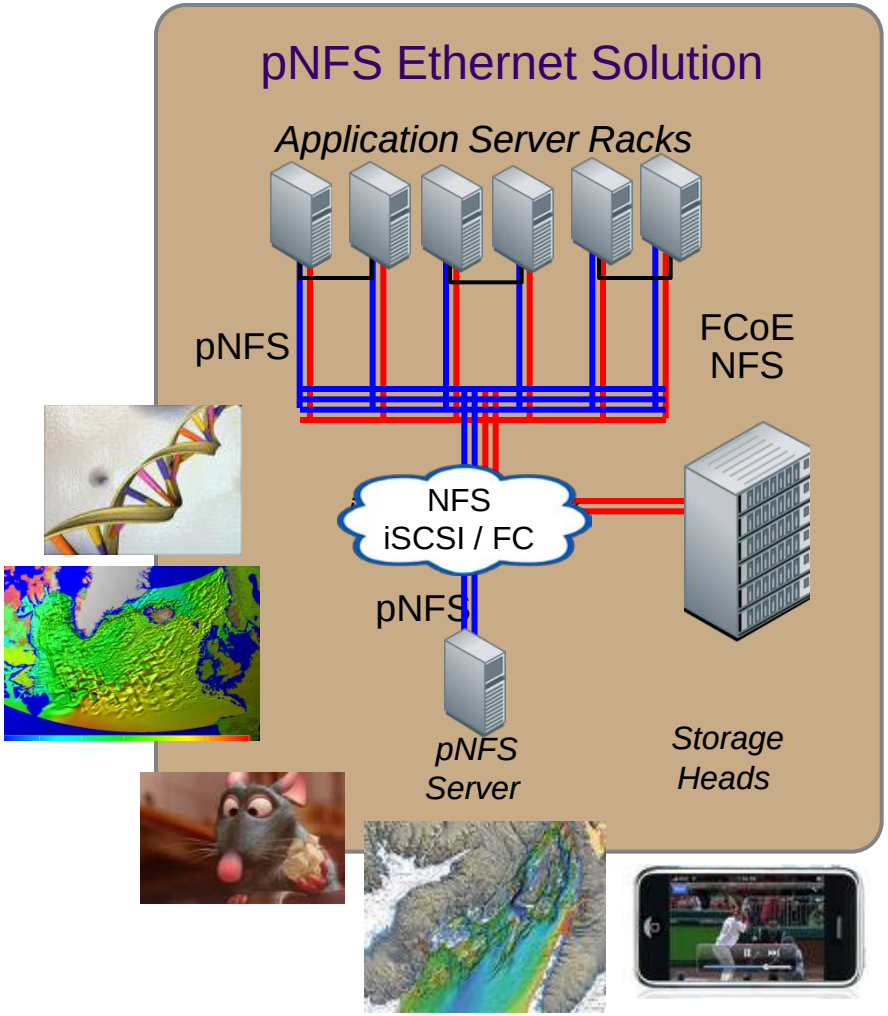
- Application Data Blocks:
  - ADB is means to allow the definition of the format of file which is being used by an enterprise application
  - Examples: database or a VM image.
  - INITIALIZE blocks with a single compound operation
    - Initializing a 30GB database takes a single over the wire operation instead of 30GB of traffic.
  - ADB describes where a logical block number is located and where a state string is located
  - Based on both of these, applications can detect corrupt blocks

NFSv4.2 Major Features

- Space reservation
  - Ability to ensure a file will have storage available to it
- Sparse file support
  - “Hole punching” and the reading of sparse files.
    - Example: If there is a 10GB hole, report with a single READ_PLUS operation.
- Labeled NFS: (LNFS)
  - MAC checks on files
- IO_ADVISE
  - Client or appl inform the server of the expected caching requirements of the file

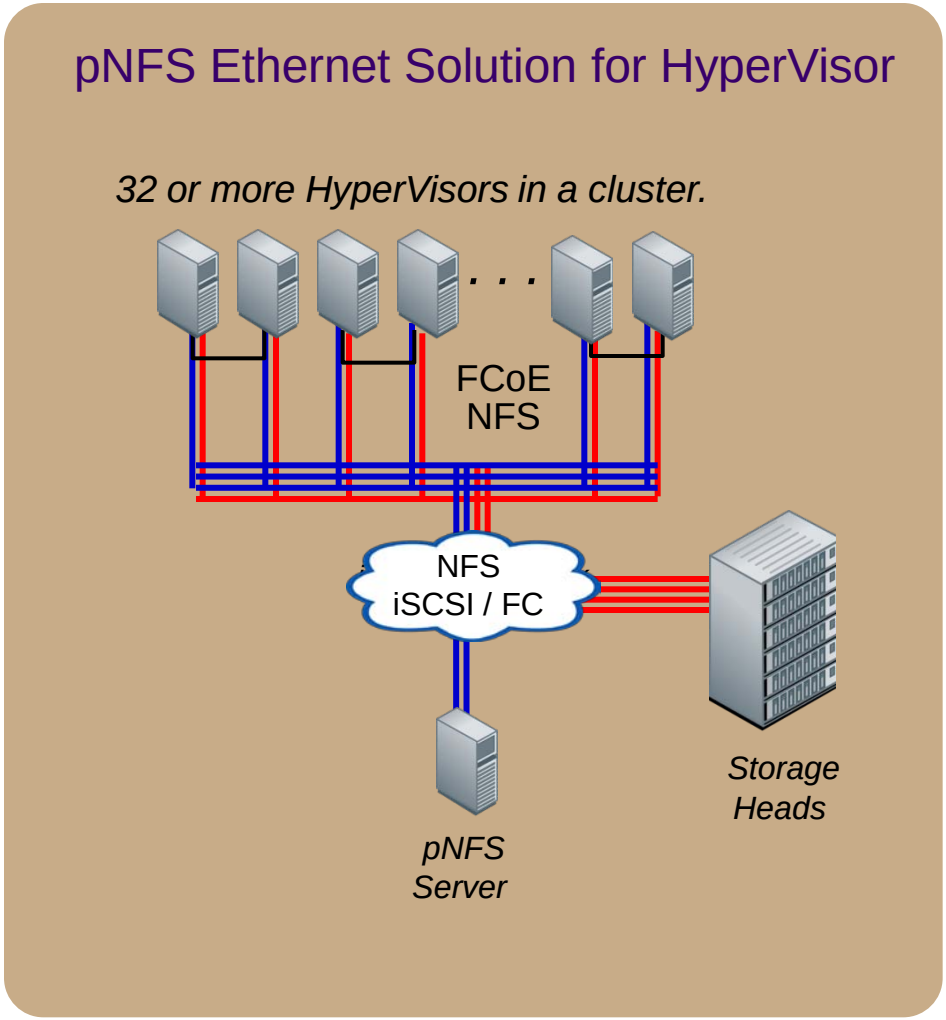
Traditional HPC Use Cases

- Seismic Data Processing / Geosciences' Applications
- Broadcast & Video Production
- High Performance Streaming Video
- Finite Element Analysis for Modeling & Simulation
- HPC for Simulation & Modeling
- Data Intensive Searching for Computational Infrastructures



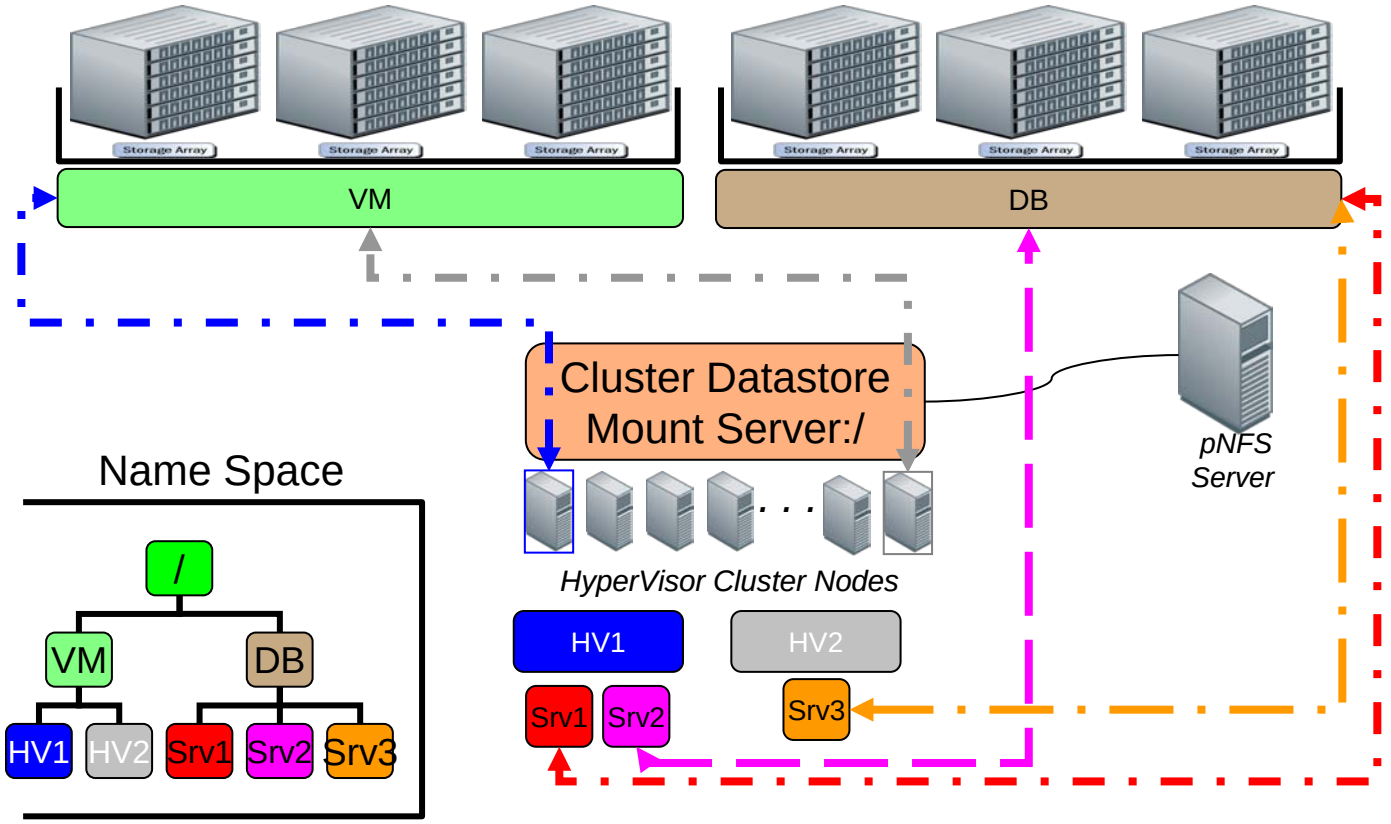
pNFS – Virtualization and Databases

- Original pNFS use case
  - 100's of hosts to storage
- 16+ Cores in future
- Single NFS Datastore
- Multiple-heads across multiple disks
- Trunking
- Directory/File Delegations
- Block pNFS Caveat
  - Limit on VMs per LUNs

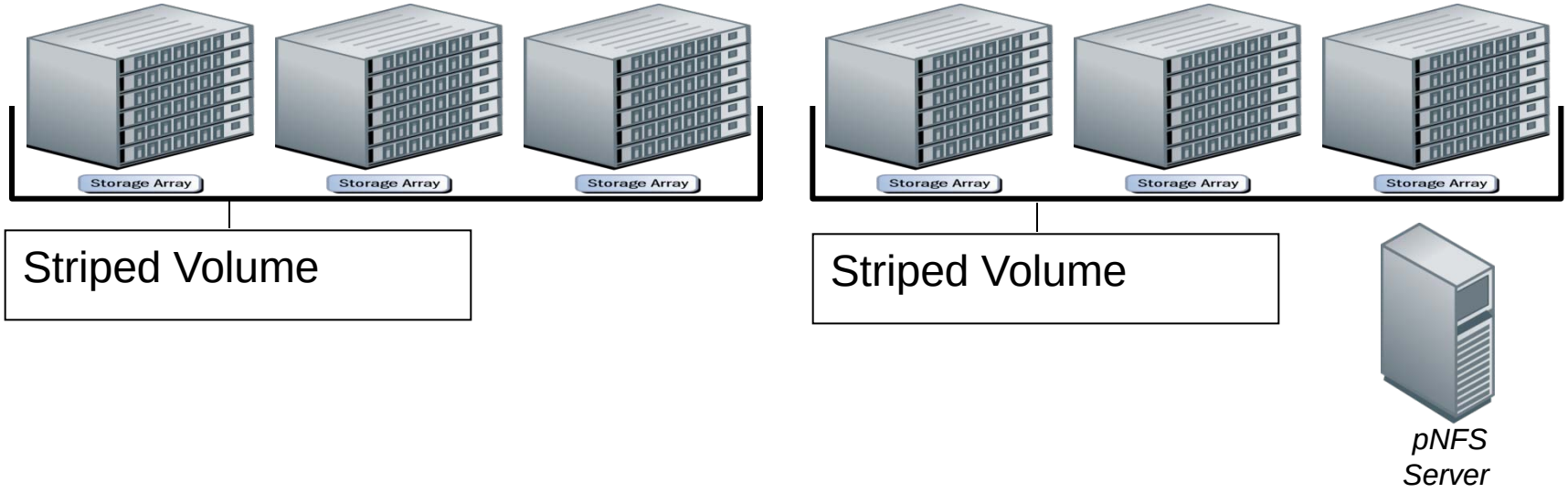


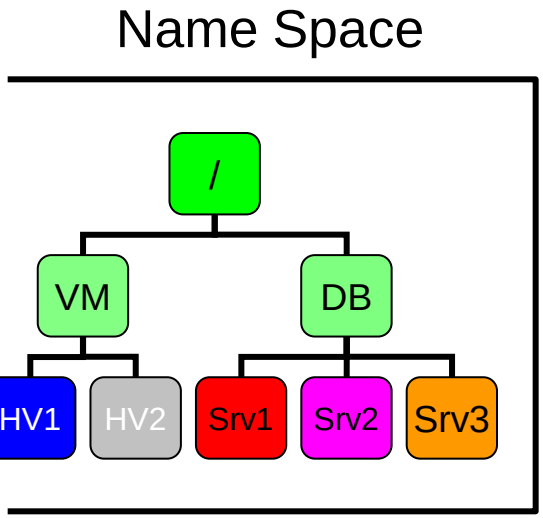
NFSv4.1 – Virtualized Data Center

- Desired destination:



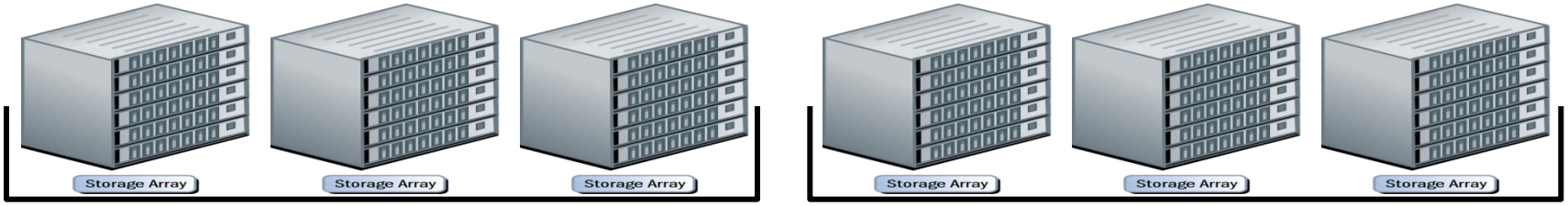
Single NFSv4.1 namespace





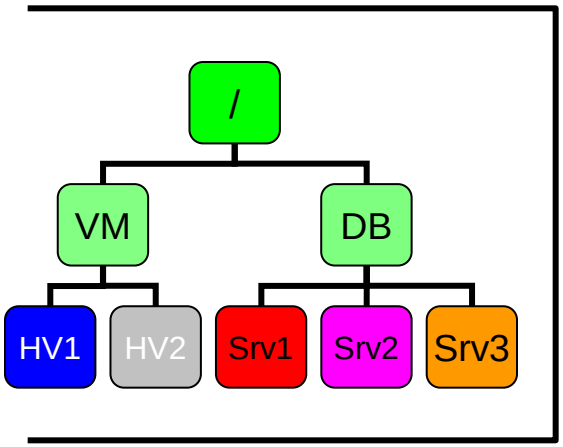
HyperVisor Cluster Nodes

Single NFSv4.1 datastore

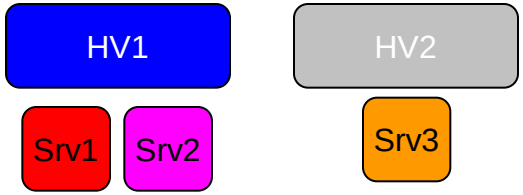


pNFS Server

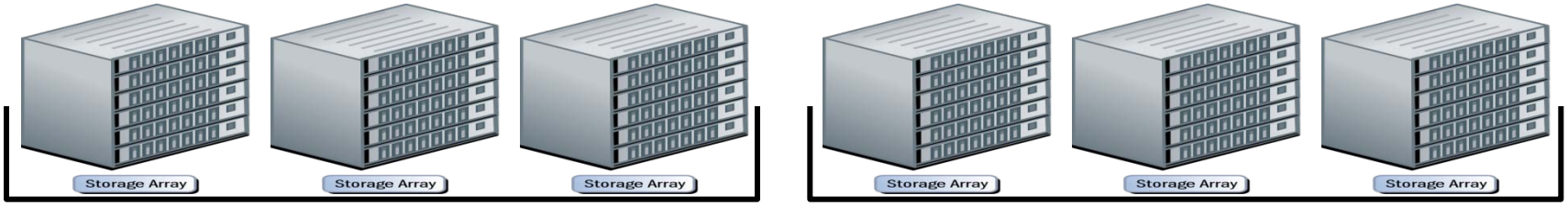
Name Space

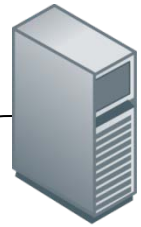


HyperVisor Cluster Nodes



VM Cluster Datastore





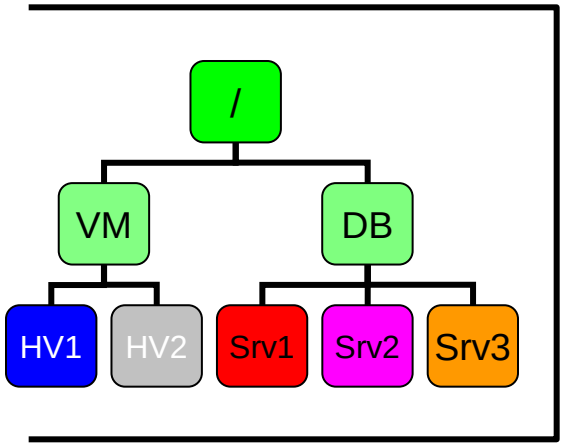
pNFS Server

Cluster Datastore
Mount Server:/

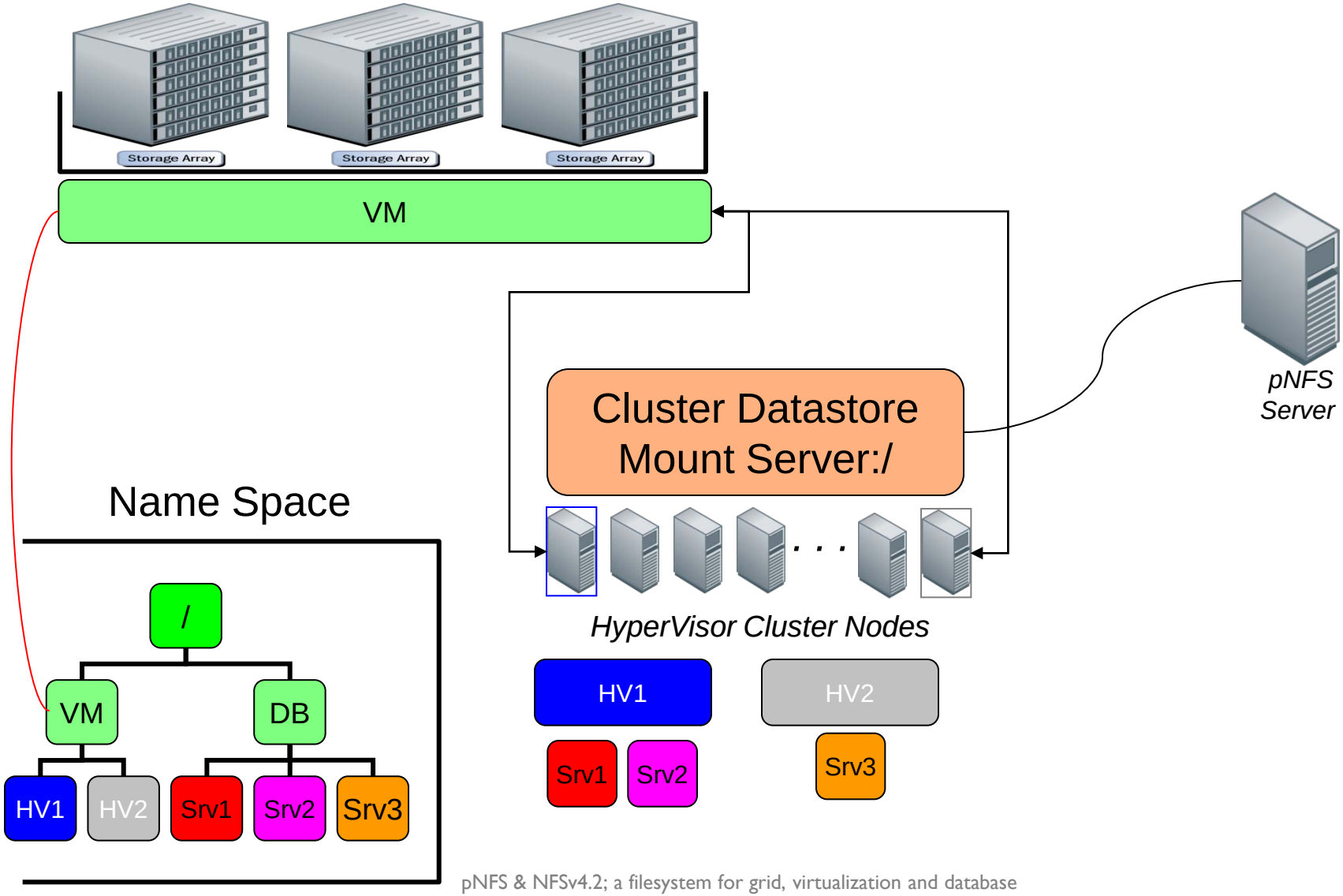
HyperVisor Cluster Nodes

HV1 | HV2 | Srv1 | Srv2 | Srv3

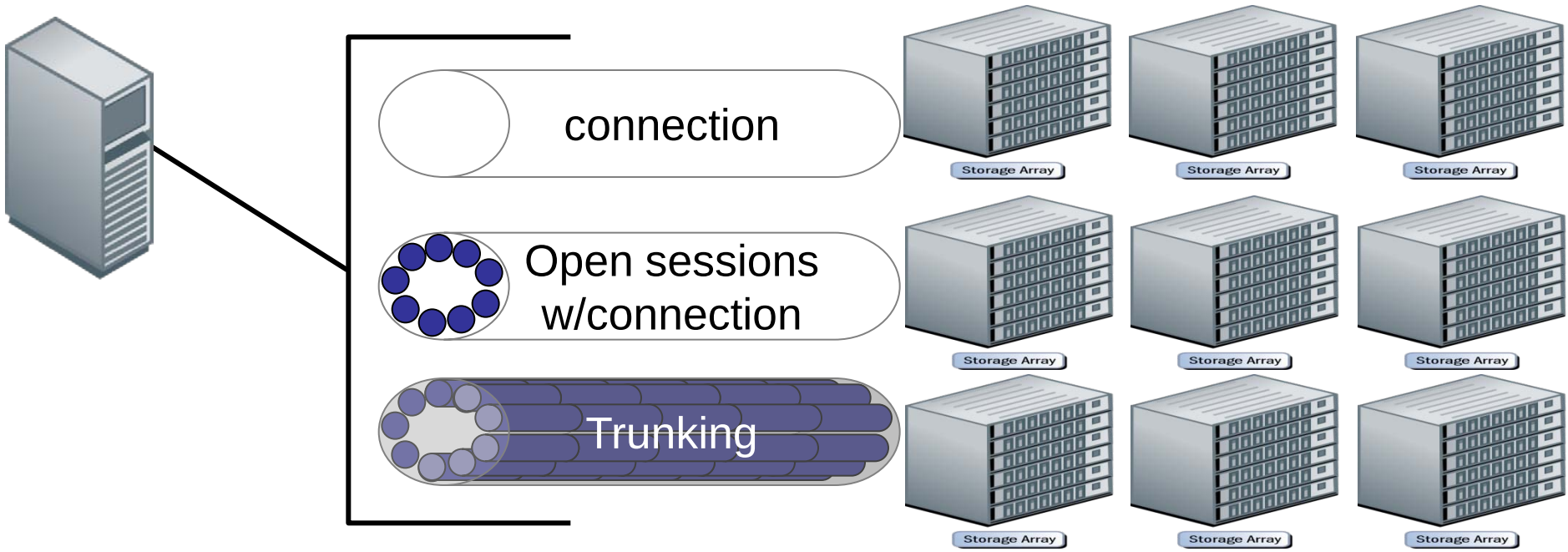
Name Space



VMs accessing volume w/layout

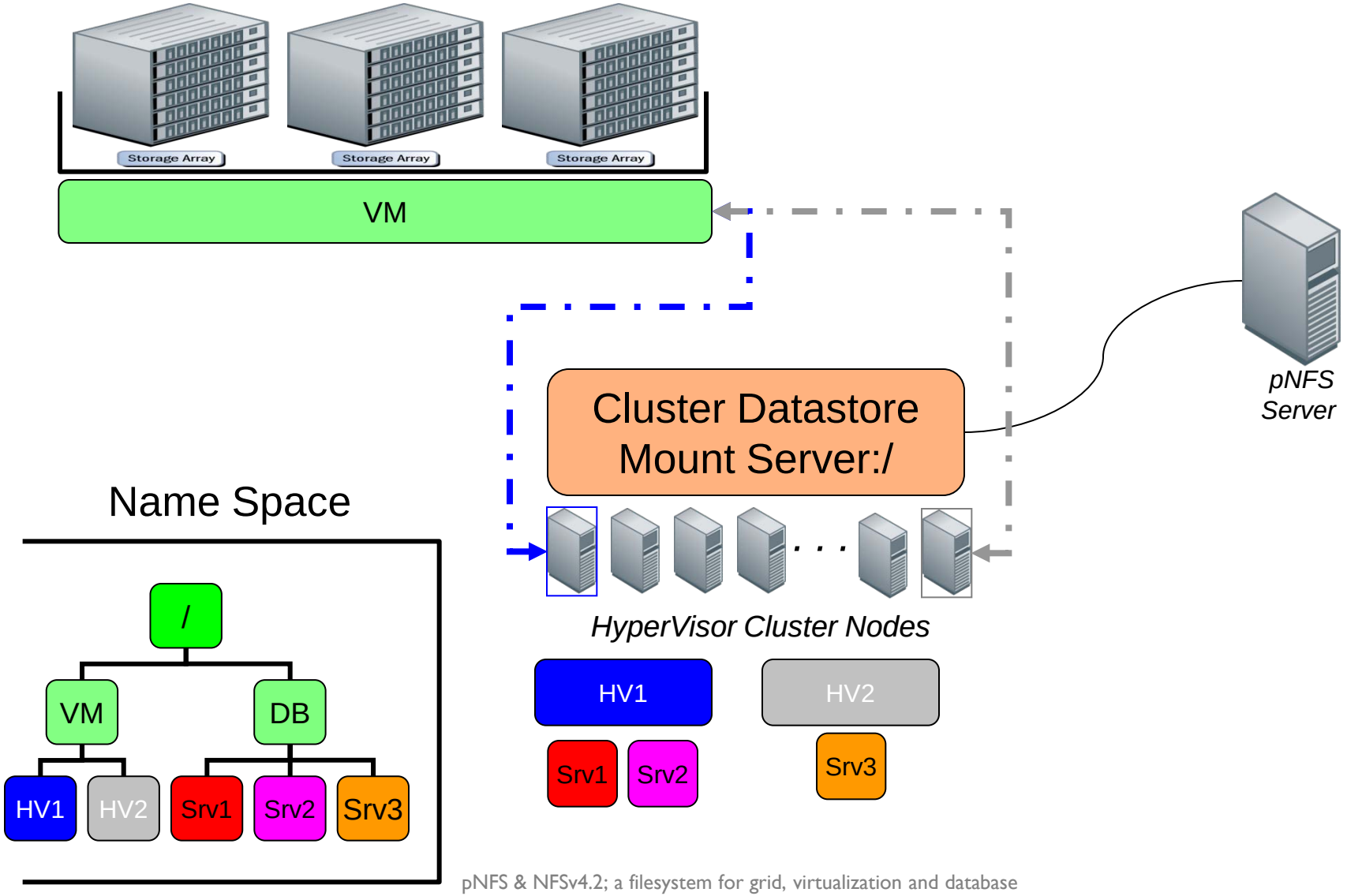


NFSv4.1 Trunking/Sessions

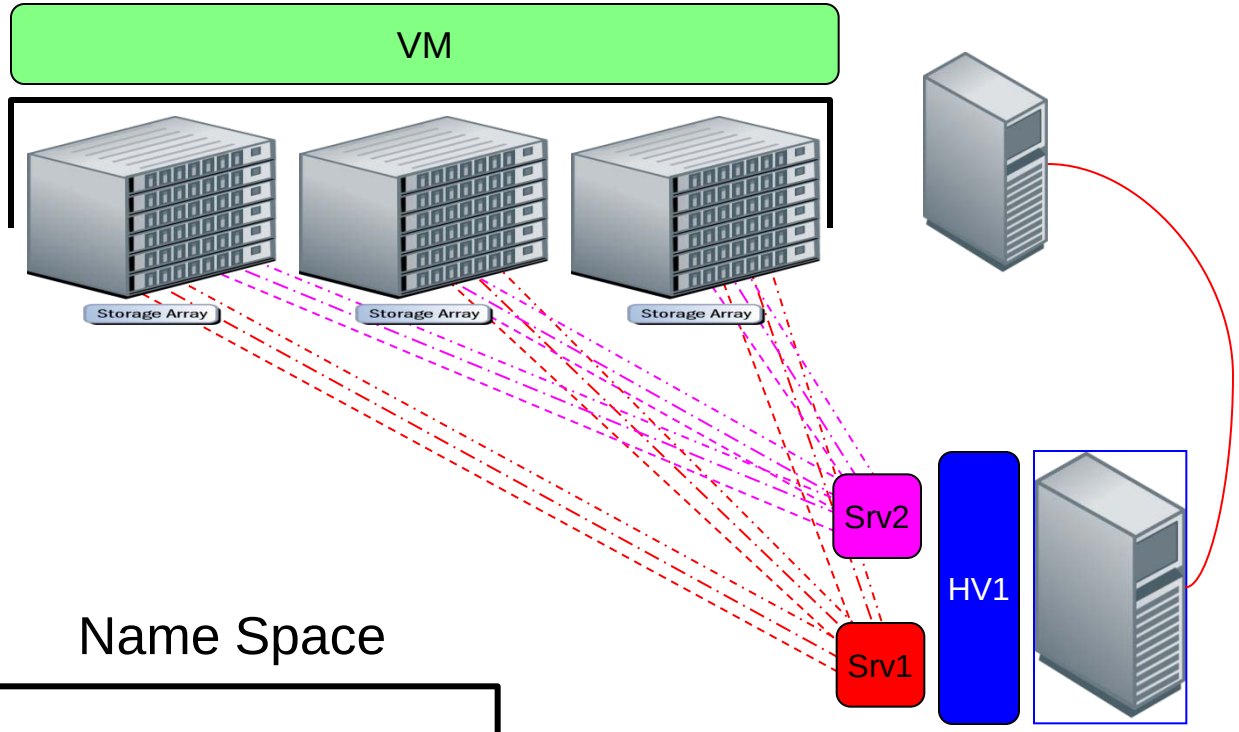


1. A single connection limits data throughput based on protocol
2. Trunking expands throughput and can reduce latency by opening multiple sessions to the same file handle/server resource

VM Access using single mount

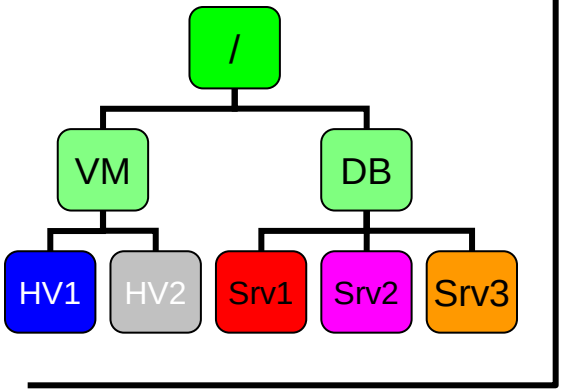


VM access using pNFS + Trunking

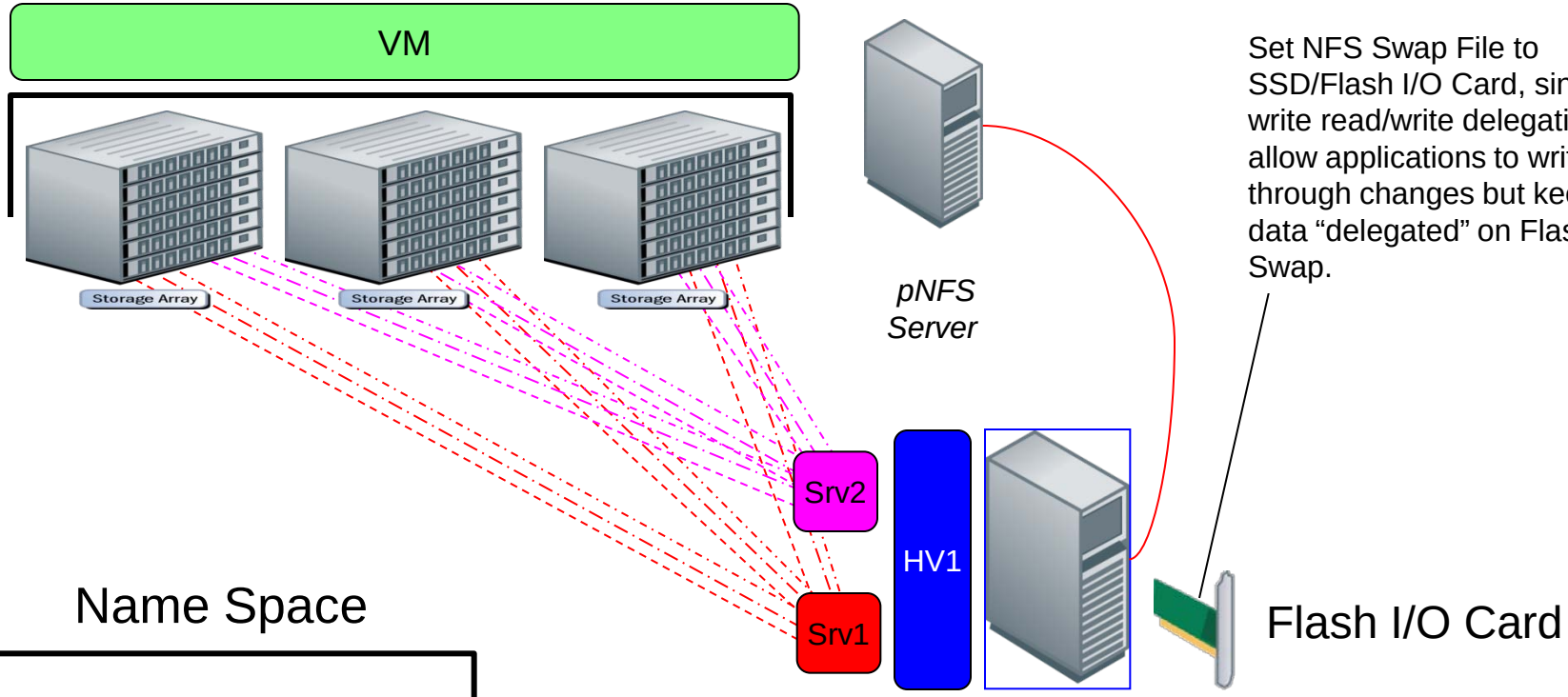


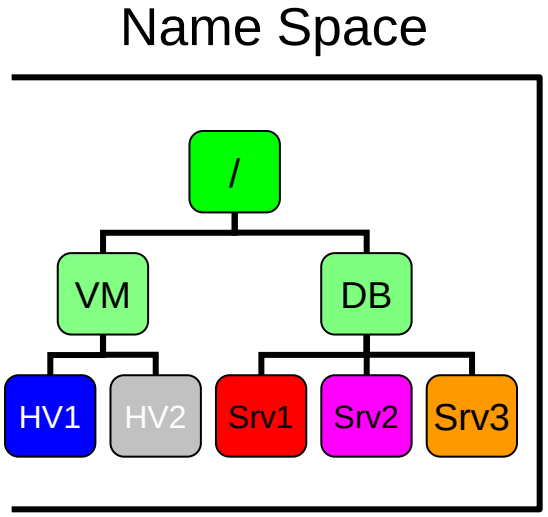
pNFS Server

Name Space



NFSv4.1 Directory/File Delegations

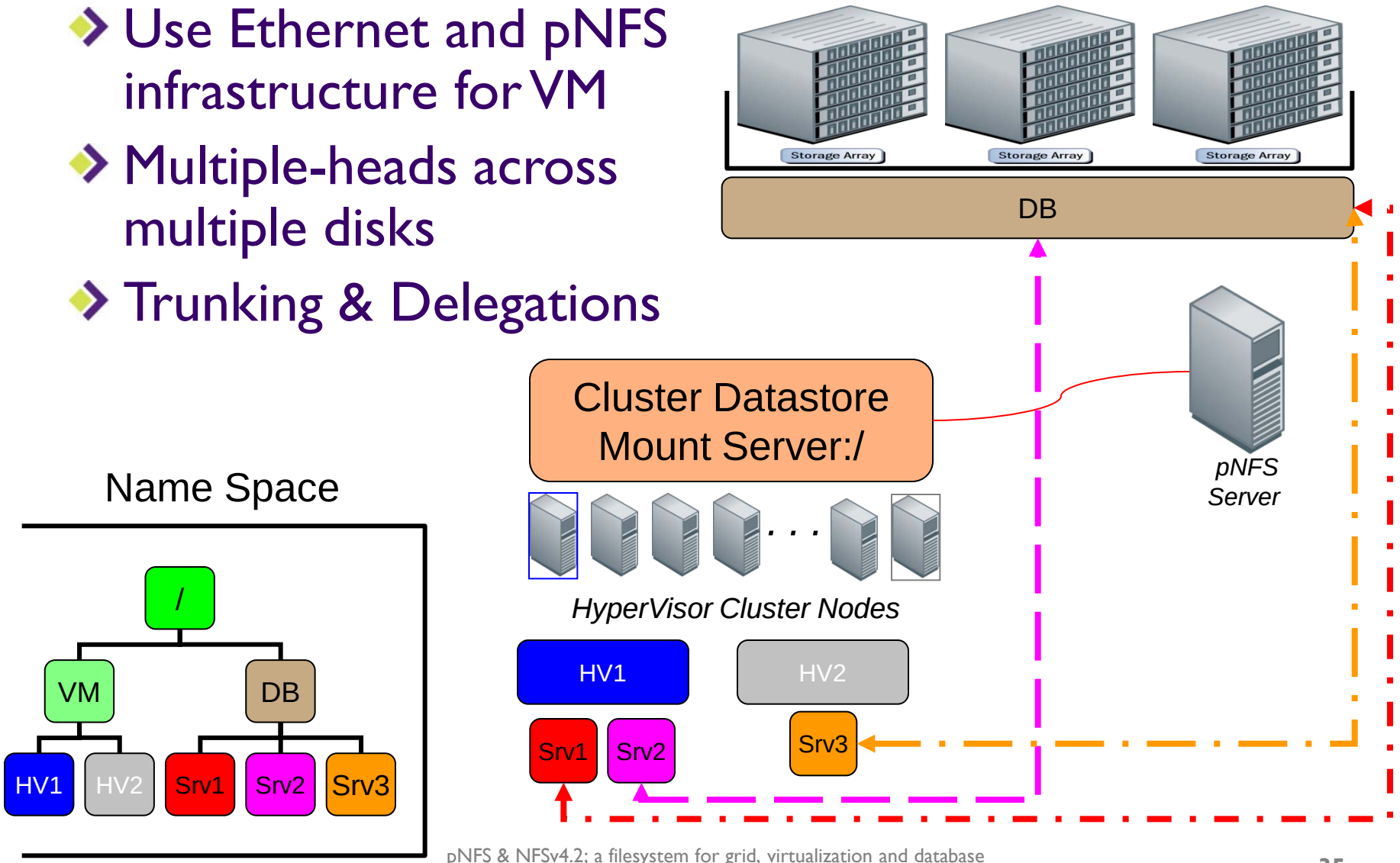




- Delegations available in NFSv4
  - Reduce renewals for locks
  - Improve R/W performance
  - Remove getattr storms

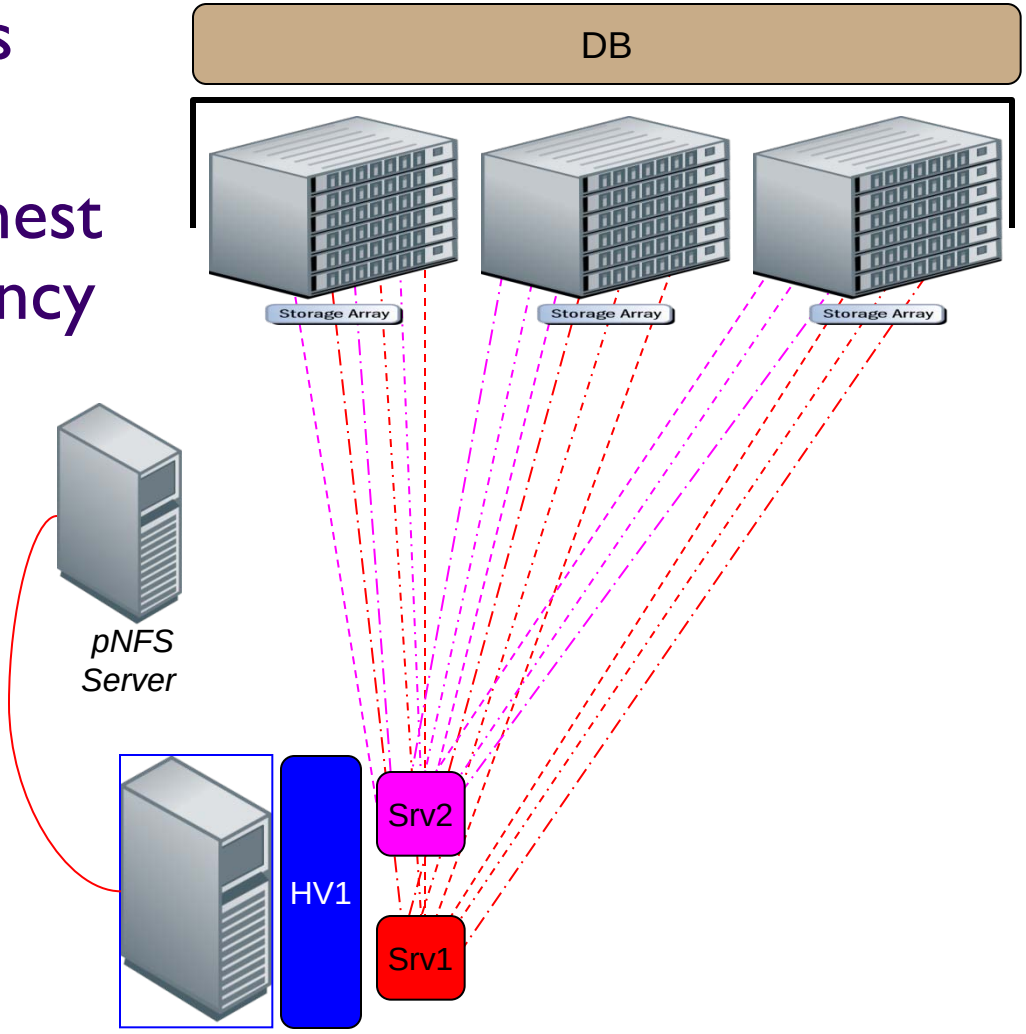
NFSv4.1 – Database enhancements

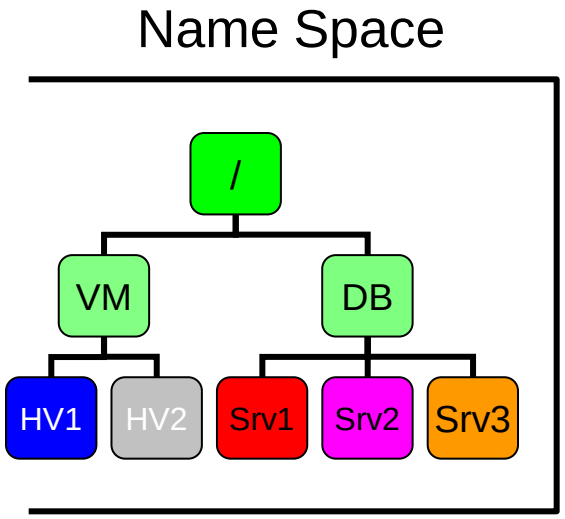
- Use Ethernet and pNFS infrastructure for VM
- Multiple-heads across multiple disks
- Trunking & Delegations



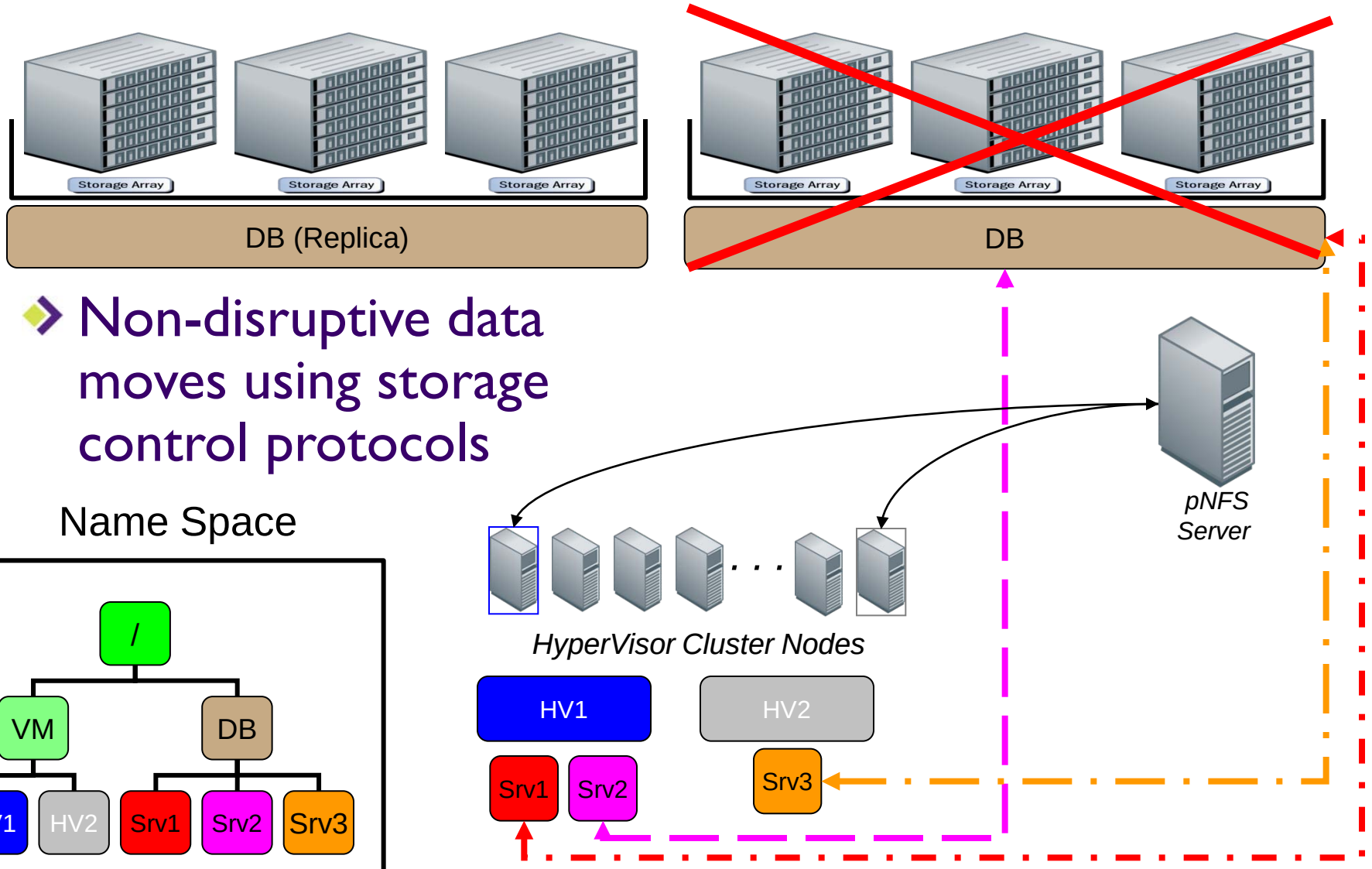
DB access using pNFS + Trunking

- Multiple-heads across multiple disks
- Trunking enables highest IOPS and lowest latency

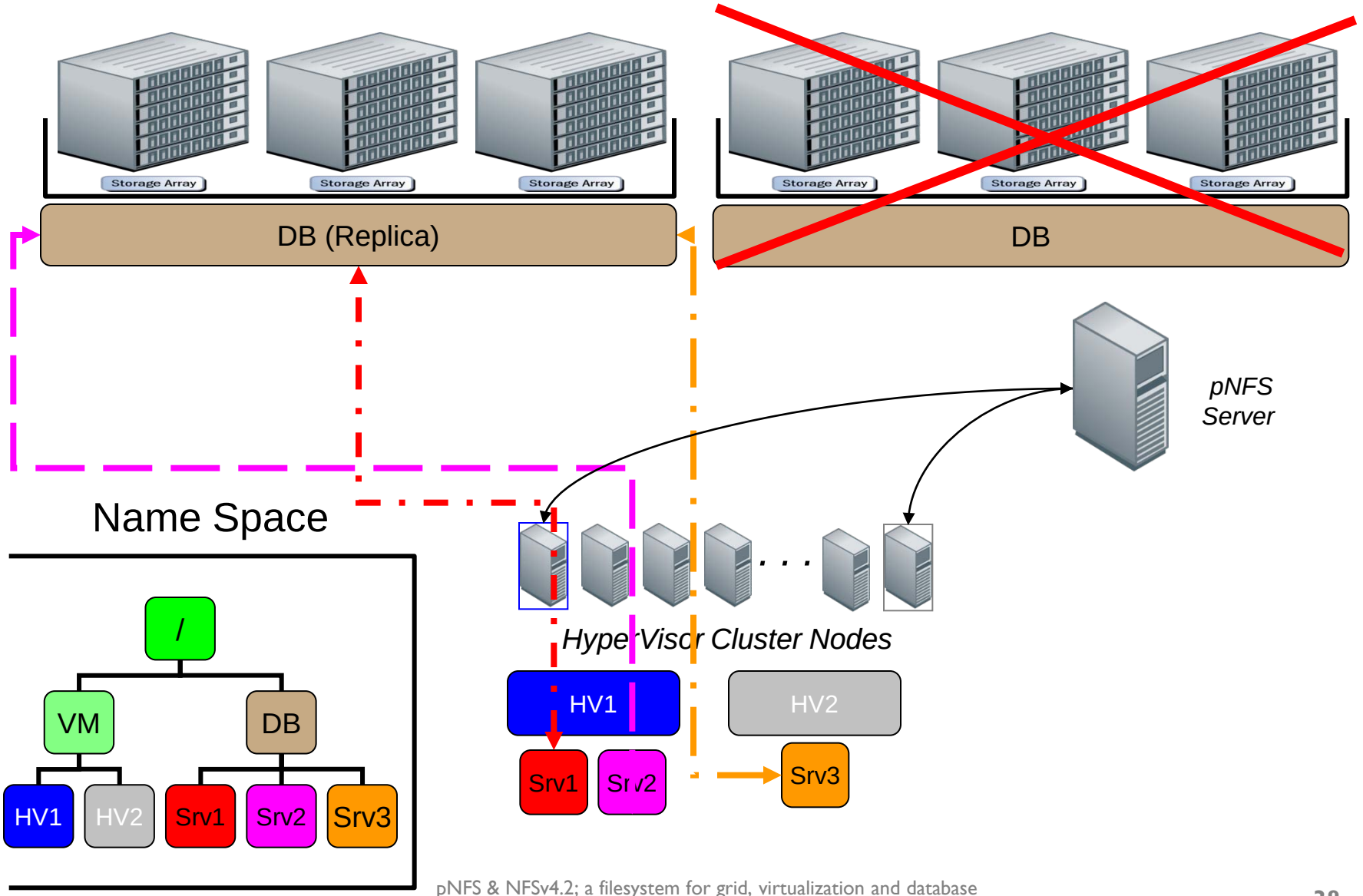




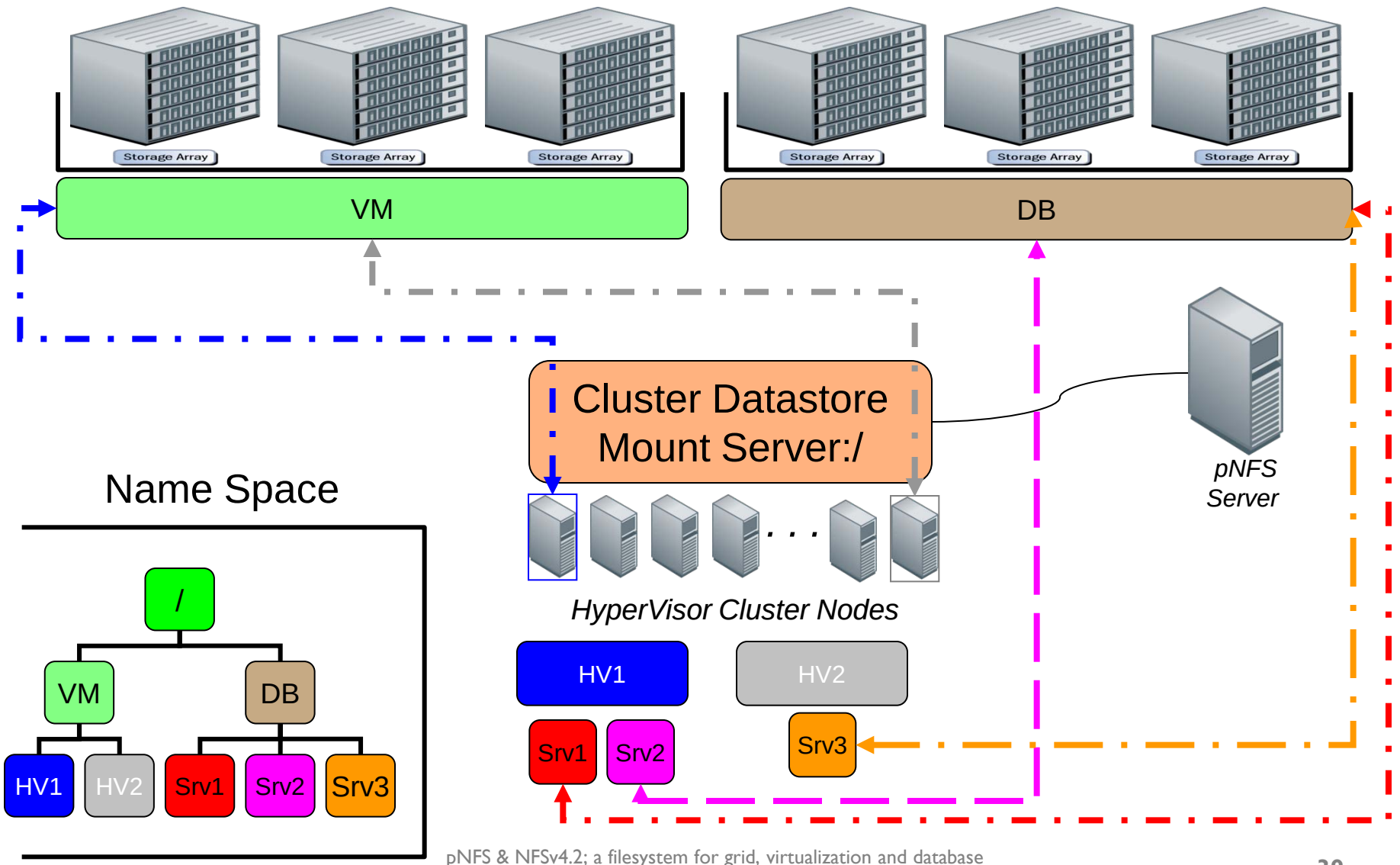
NFSv4.1 – Layout Callbacks



NFSv4.1 – Layout Callbacks



NFSv4.1 – Virtualized Data Center



Summary/Call to Action

- pNFS is the first open standard for parallel I/O across the network
  - Ask vendors to include NFSv4.1 support for client/servers
- pNFS has wide industry support
  - commercial implementations and open source
- Start using NFSv4.0, NFSv4.1 today
  - NFSv4.2 nearing approval

Q&A / Feedback

- Please send any questions or comments on this presentation to SNIA: tracktutorials@snia.org

Many thanks to the following individuals for their contributions to this tutorial.

- SNIA Education Committee

Joshua Konkle (author)
Mike Eisler, Co-Editor of NFSv4.1
J. Bruce Fields
Brian "Beepy" Pawloski, (Co-Chair, NFSv4.1)
Joe White,
Howard Goldstein,
Ken Gibson
Omer Asad
Sachin Chheda
Jason Blosil
Sorin Faibash
Rob Peglar
Dave Hitz
Dave Noveck
Peter Honeyman
Brent Welch
David Black
Piyush Shivam
Mark Carlson
Andy Adamson
Pranoop Ersani
Ricardo Labiaga
Tom Haynes

Backup slides.

- http://wiki.linux-nfs.org/wiki/index.php/Main_Page
- NFS Version 4.1
  - RFC 5661 - Network File System (NFS) Version 4 Minor Version 1Protocol
  - RFC 5662 - Network File System (NFS) Version 4 Minor Version 1
- External Data Representation Standard (XDR) Description
  - RFC 5663 - Parallel NFS (pNFS) Block/Volume Layout
  - RFC 5664 - Object-Based Parallel NFS (pNFS) Operations
- http://tools.ietf.org/html/
- pNFS Problem Statement
  - – Garth Gibson (Panasas), Peter Corbett (Netapp), Internet-draft, July 2004
  - http://www.pdl.cmu.edu/pNFS/archive/gibson-pnfs-problem-statement.html
- Linux pNFS Kernel Development
  - http://www.citi.umich.edu/projects/asci/pnfs/linux