SNIA

Education

pNFS, parallel storage for grid, virtualization and database computing

Joshua Konkle, Chair NFS SIG

Pranoop Erasani, Sr. Ecosystem Architect, NFS Protocols, NetApp

SNIA Legal Notice

Education SNIA

- The material contained in this tutorial is copyrighted by the SNIA.
- Member companies and individuals may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced without modification
  - The SNIA must be acknowledged as 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 nor should be construed as legal advice or opinion. If you need legal advice or legal opinion please contact an attorney.
- The information presented herein represents the Author's personal opinion and current understanding of the 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, parallel storage for grid, virtualization and database computing
  - 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. Explain the storage layouts for parallel NFS; NFSv4.1 Files, Blocks and T10 OSD Objects. We'll conclude the session with 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 and Panasas act as co-chairs
- Deliver Panels/Sessions on NFSv4.1 when possible
  - E.g. SNW Europe in October '09, Super Computing '10

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 common 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

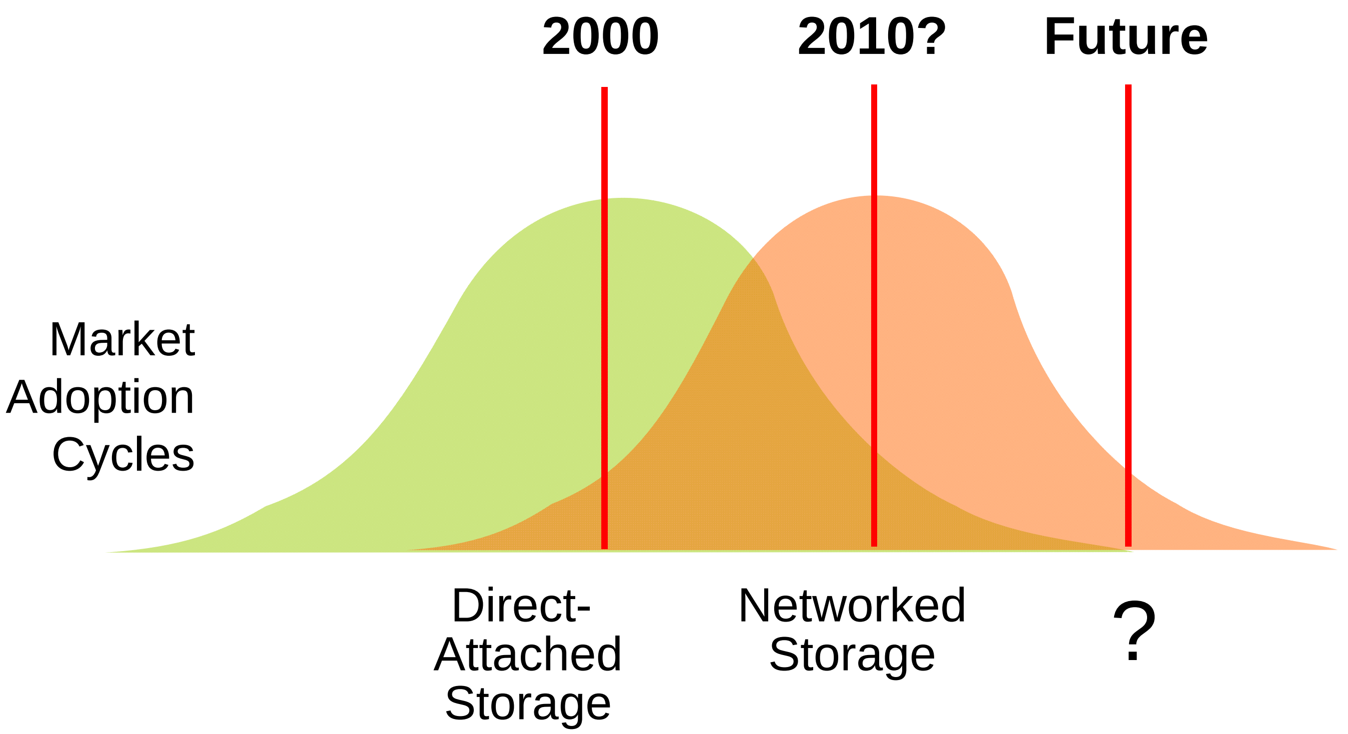
NFSv4 SHIP is sailing

	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 utf8 characters	Global file system for multi-national organizations
Performance	Multiple read, write, delete operations per RPC call Delegate locks, read and write procedures to clients	Better network utilization for all NFS 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

The Evolution of Storage

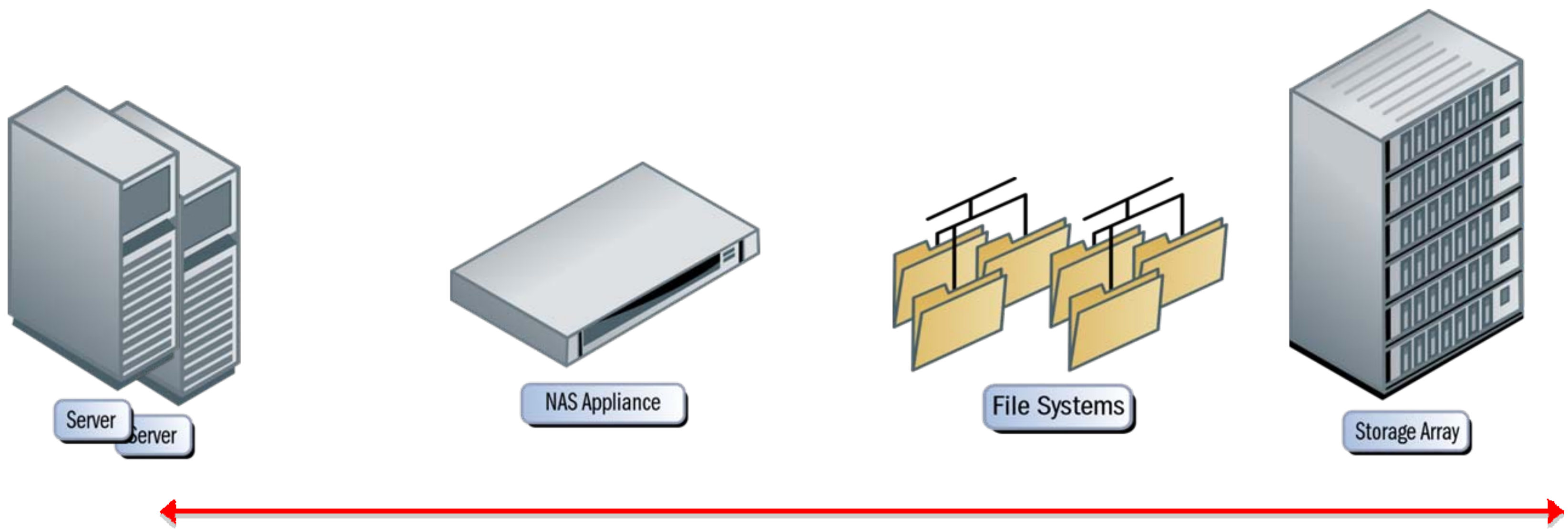


Evolving Requirements

- Economic Trends
  - Cheap and fast computing clusters
  - Cheap and fast network (GigE to 10GigE)
- Performance
  - Exposes single threaded bottlenecks in applications
  - Evolution of computing models
  - Reduced time to market, response time
- Powerful compute systems
  - Analysis begets more data, at exponential rates
  - Competitive edge (IOPS)

NFS – What's the problem?

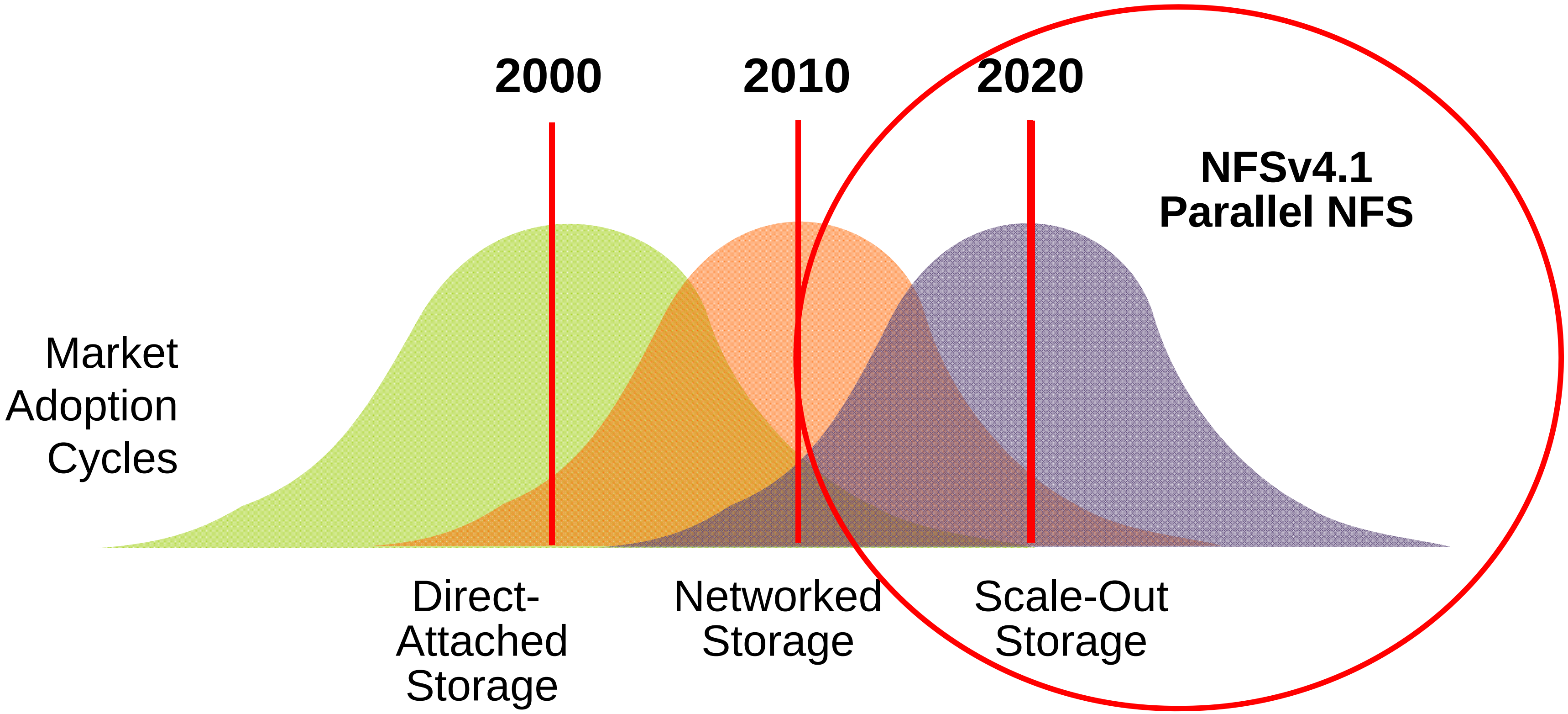
- In-band data access model
  - Easy to build, Limited in scale
  - Well-defined failure modes
  - Limited load balancing options
- Results in Limitations
  - Islands of storage
  - Server and Appliance HW
  - Networking and I/O



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
  - Limited to dual-head configuration
- 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 load balancing are typically limited to a pair of NFS server heads

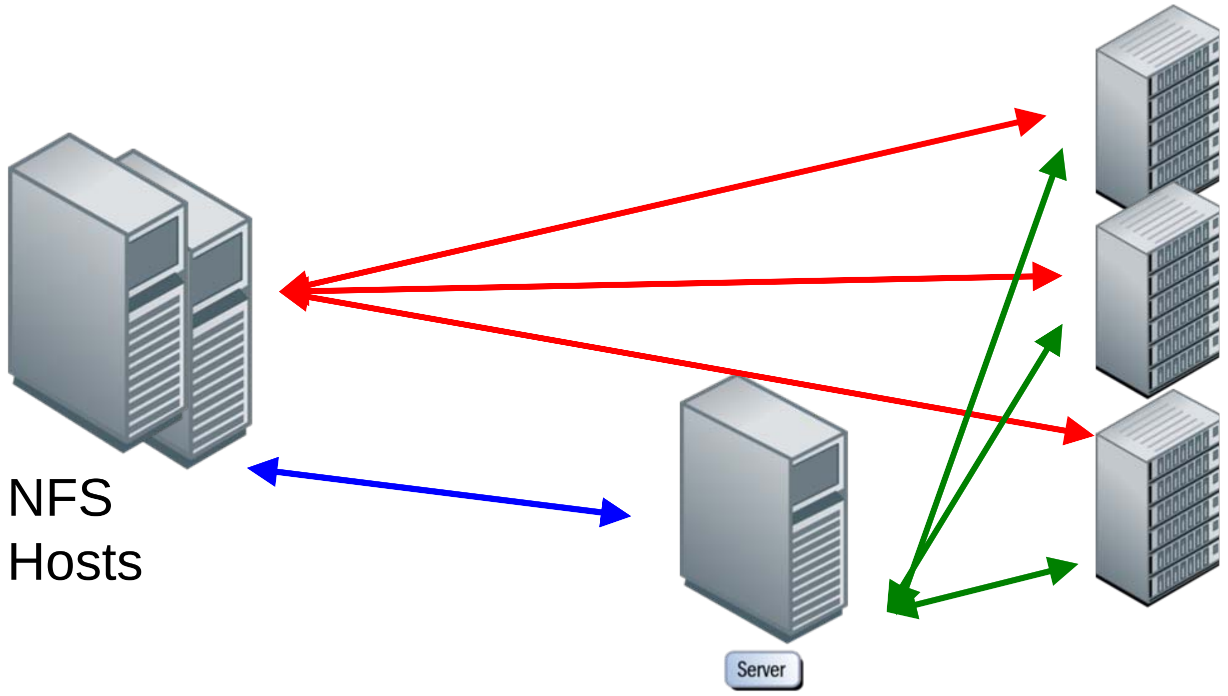
What is the Solution?



NFSv4.1 – Parallel Data Storage

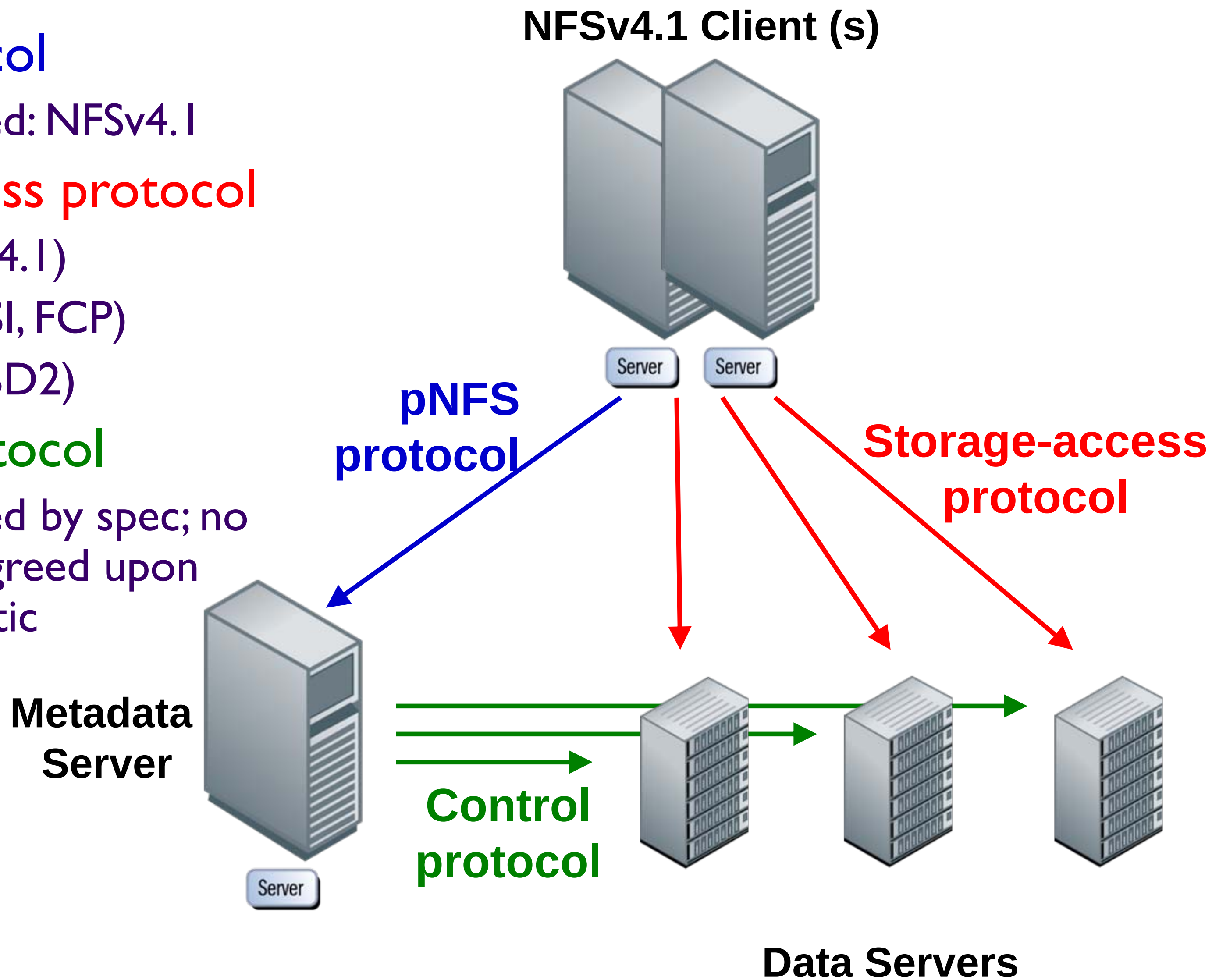
Education SNIA

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



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

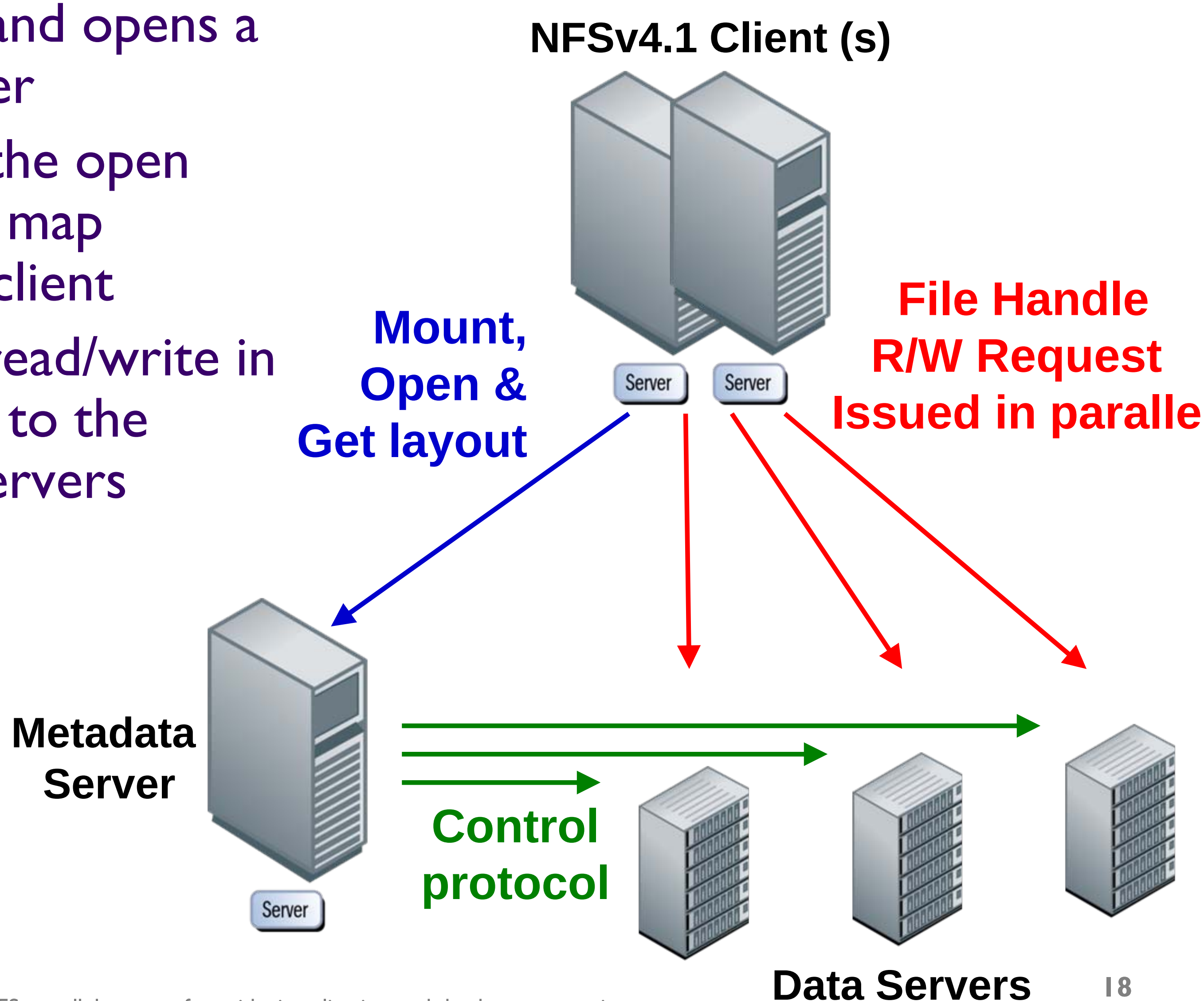
- LAYOUTGET
  - Obtains the data server map from the meta-data server
- LAYOUTCOMMIT
  - Servers commit the layout and update the meta-data maps
- LAYOUTRETURN
  - Returns the layout; Or the new layout, if the data is modified
- GETDEVICEINFO
  - Client gets updated information on a data server in the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

NFSv4.1 – OpenSource Status

- Two OpenSource Implementations
  - OpenSolaris and Linux
- OpenSolaris Client and Server
  - Support only file-based layout
  - Support for multi-device striping already present (NFSv4.1 + pNFS)
  - "Simple Policy Engine" for policy-driven layouts also in the gate
- Linux Client and Server
  - Support files (NFSv4.1)
  - Support in progress blocks (SCSI), objects (OSD T10)
  - Client consists of generic pNFS client and "plug ins" for "layout drivers"
- Predicted timeline for Linux:
  - Basic NFSv4.1 features 1H2009 – On Target; delivered Sessions/Trunking
  - NFSv4.1 pNFS and layout drivers by 2H2009 - Behind
  - Linux distributions shipping supported pNFS in 2010 - Behind

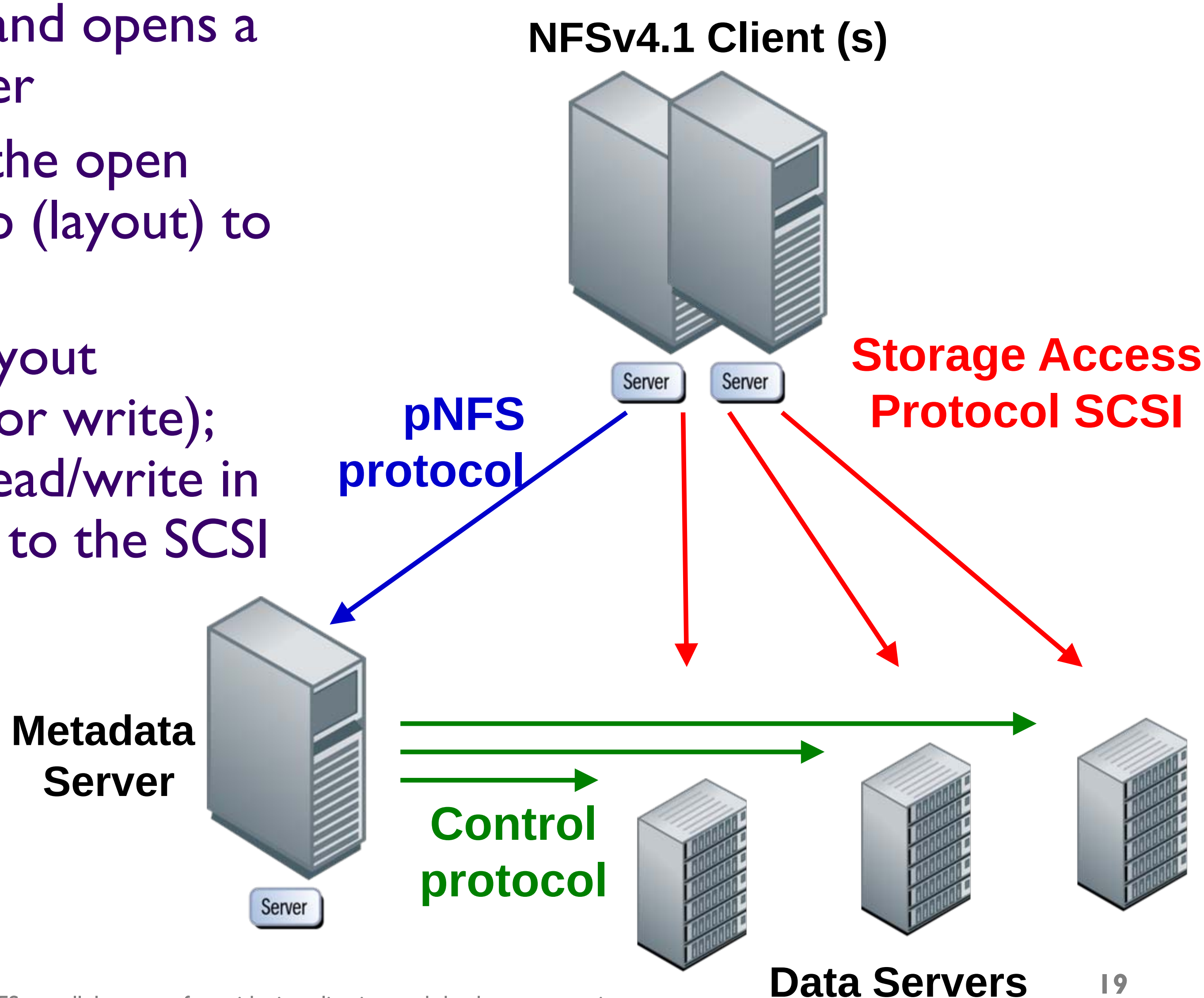
pNFS – NFSv4.1 files access

- 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



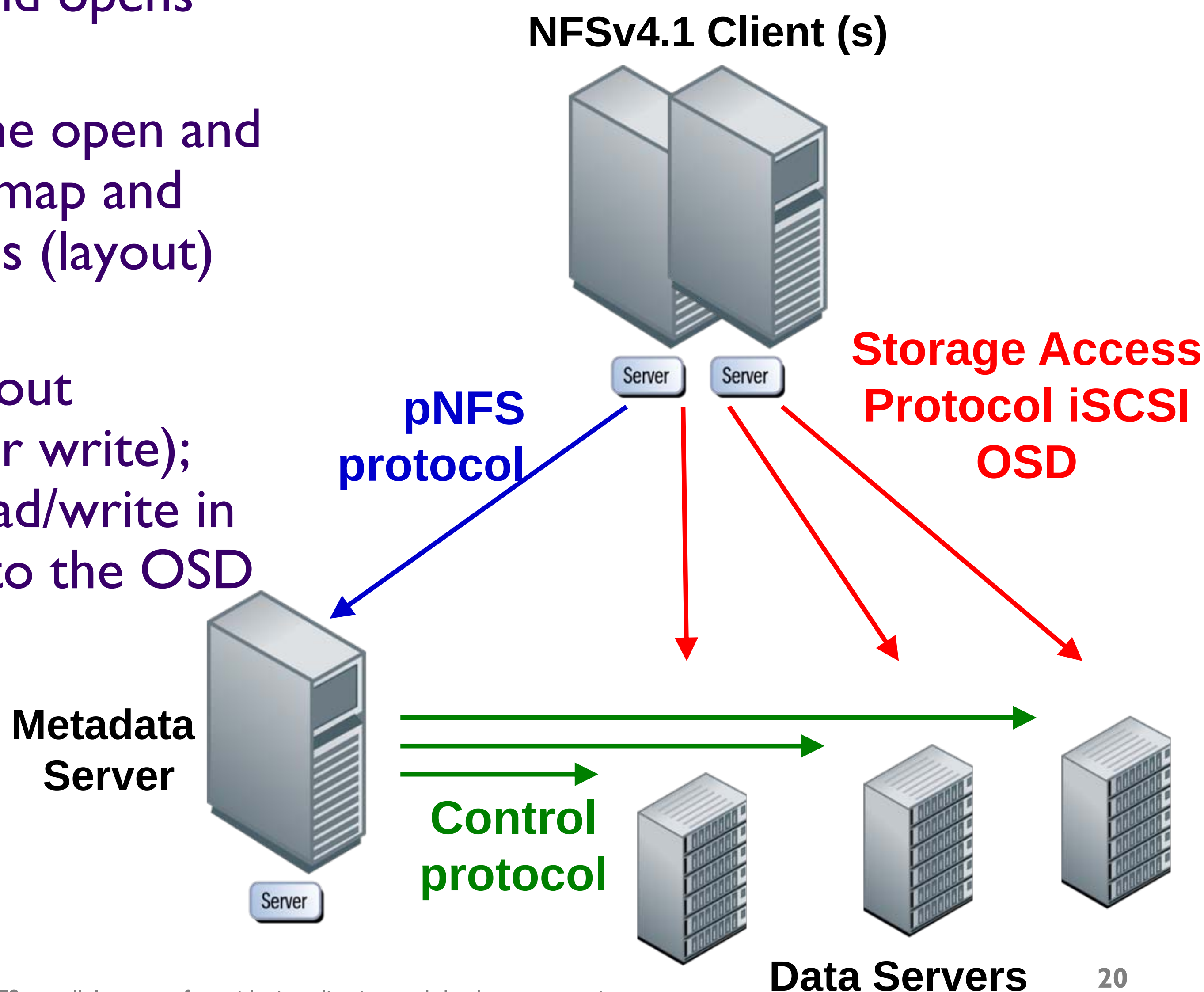
pNFS Blocks Access Model

- Client mounts and opens a file on the server
- Servers 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 target's



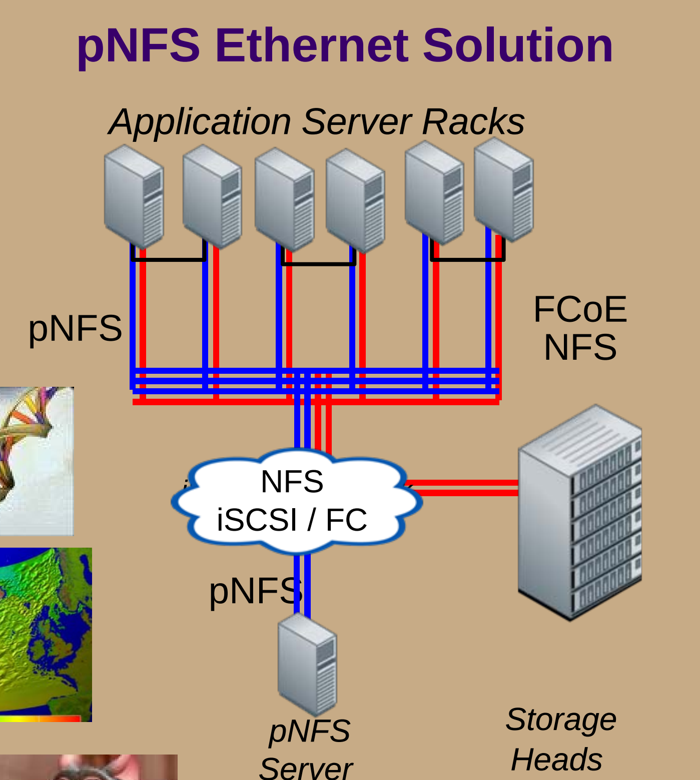
pNFS Objects Access Model

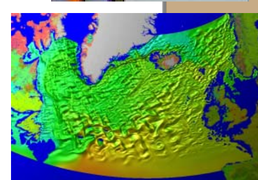
- Client mounts and opens Object
- Servers grants the open and an object stripe map and object capabilities (layout) to the client
- Based on the layout obtained (read or write); the client can read/write in parallel directly to the OSD targets

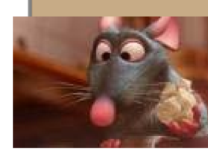


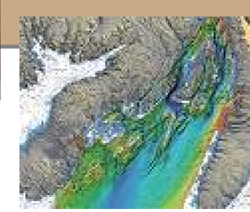
Traditional HPC Use Cases

- 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



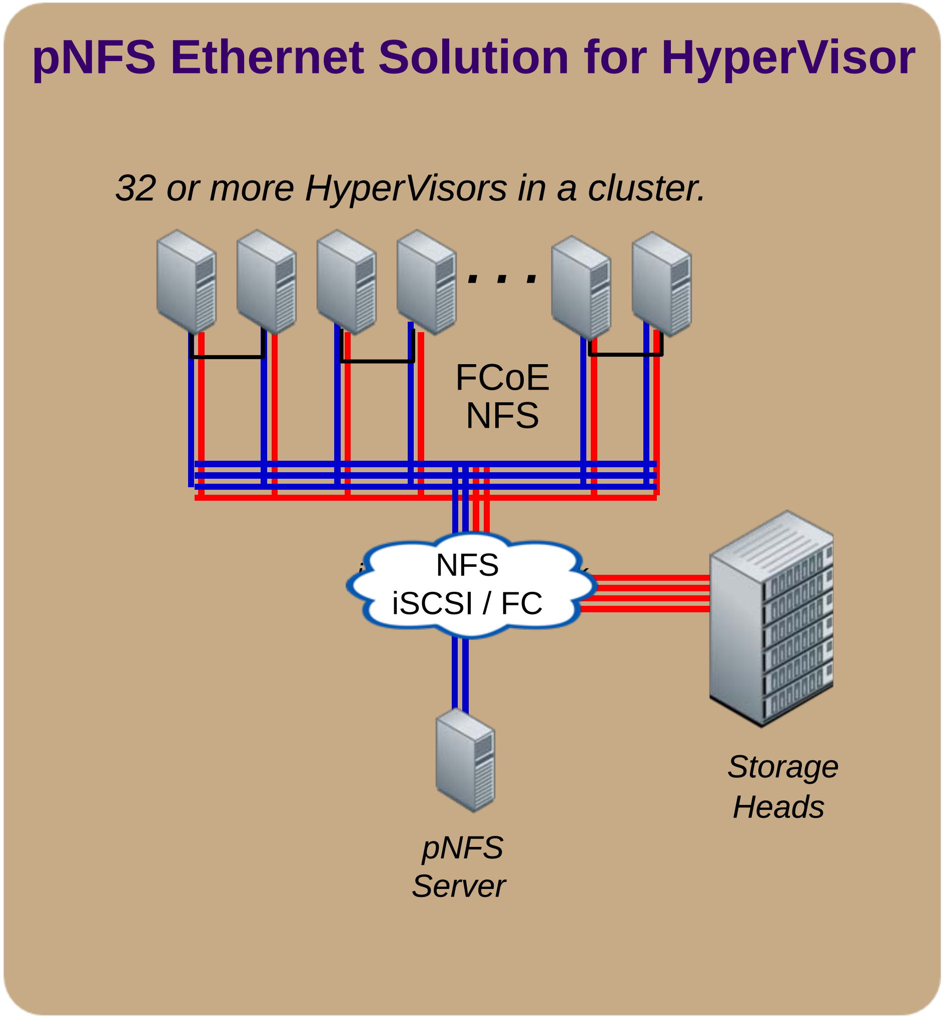




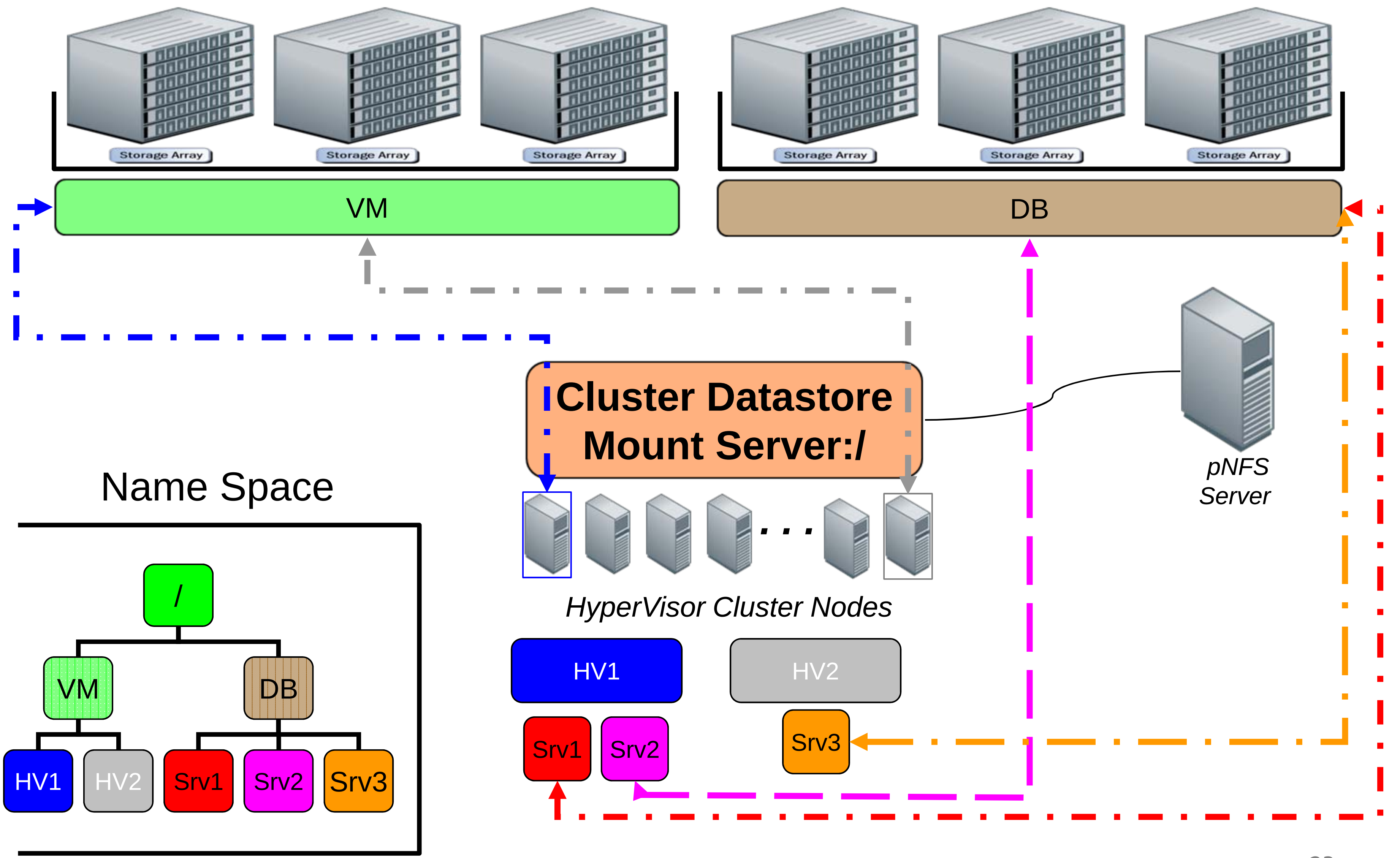


pNFS for Virtualization and Databases

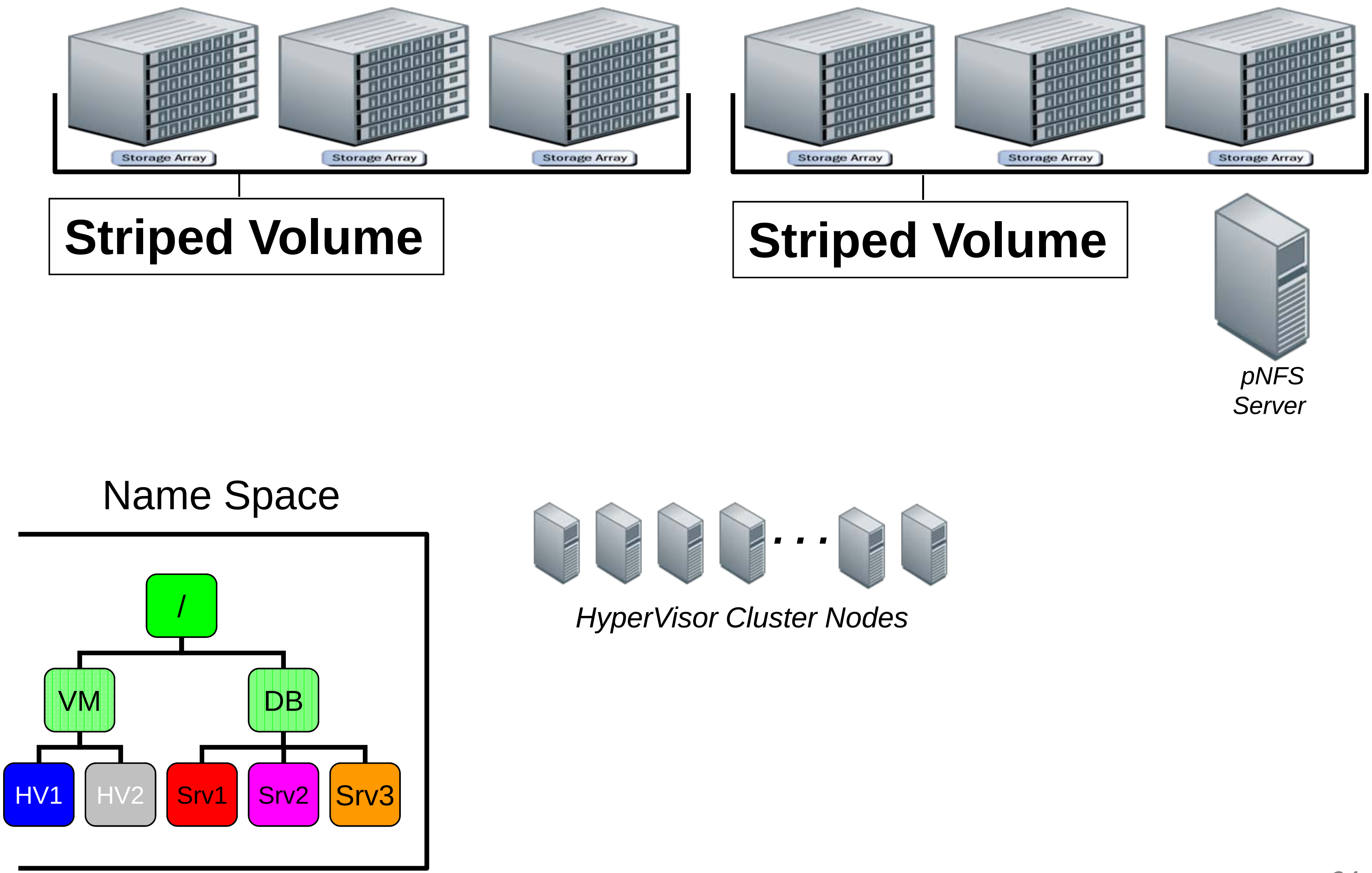
- Original pNFS use case
  - 100's of hosts to storage
- 16+ Cores in future
- Single NFS Datastore
- Multiple-heads across multiple disks
- Trunking
- Directory/File Delegations
- Caveat
  - Limit on VMs per LUNs



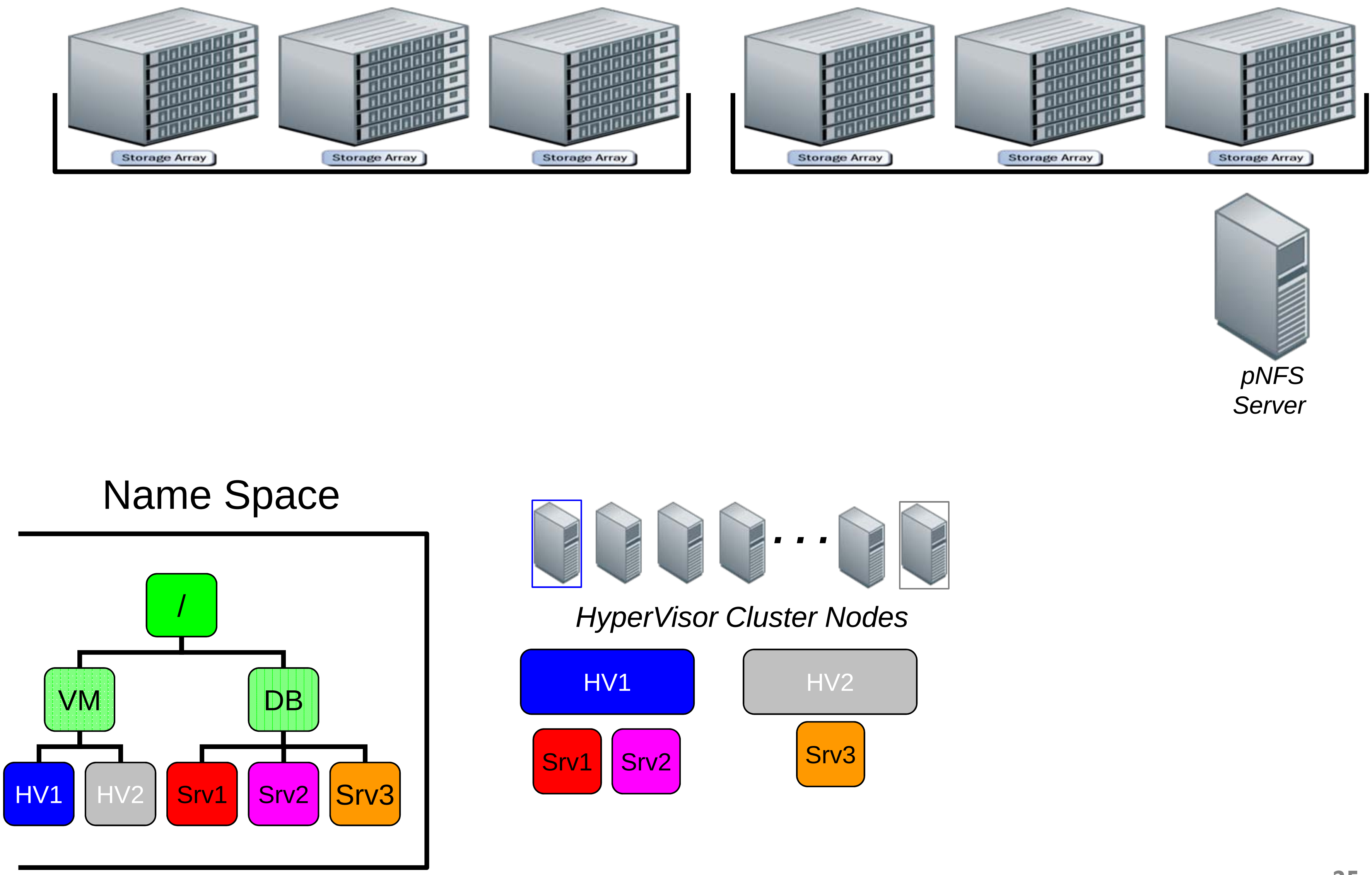
NFSv4.1 – Virtualized Data Center



Single NFSv4.1 namespace

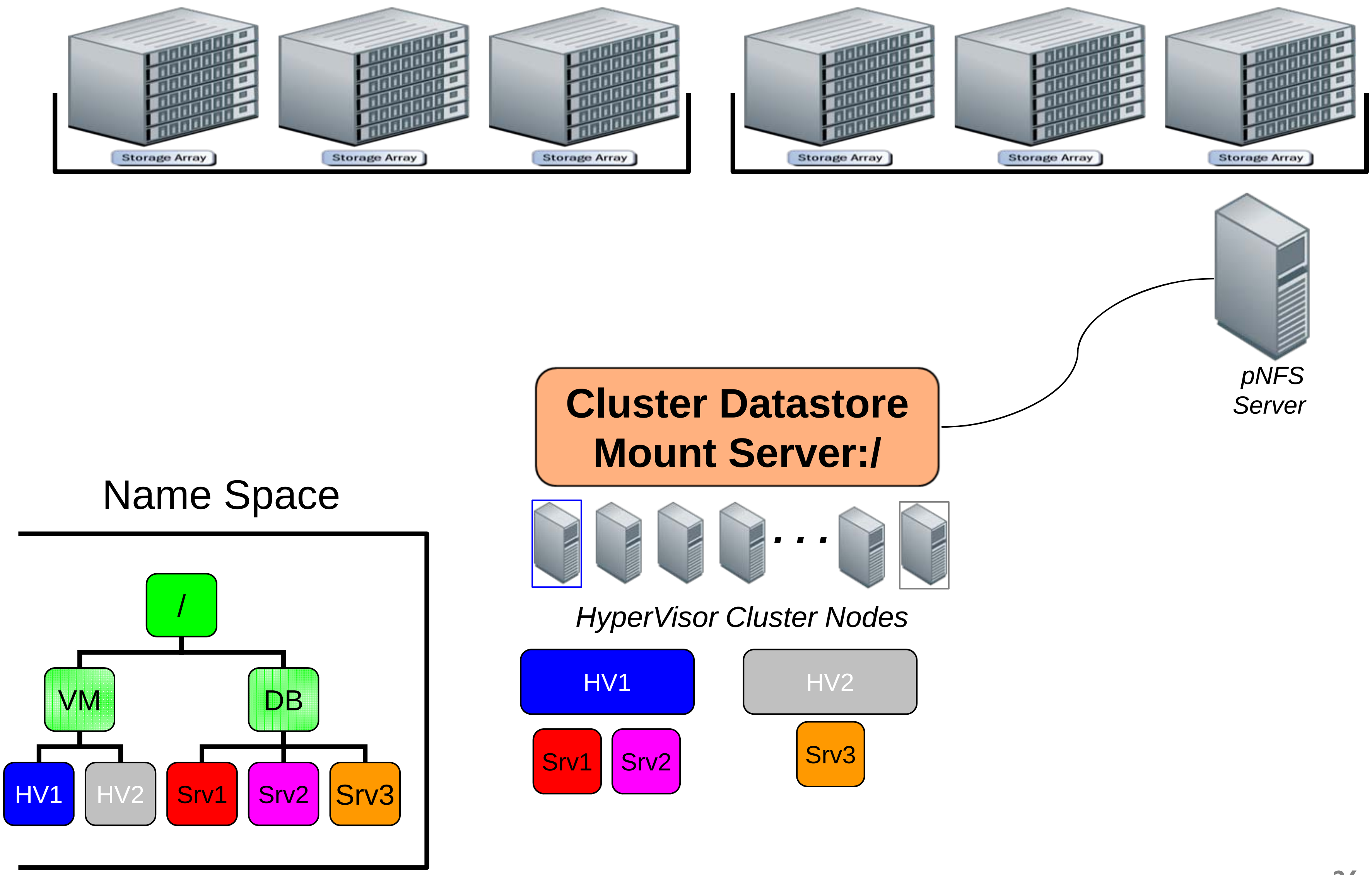


Single NFSv4.1 datastore

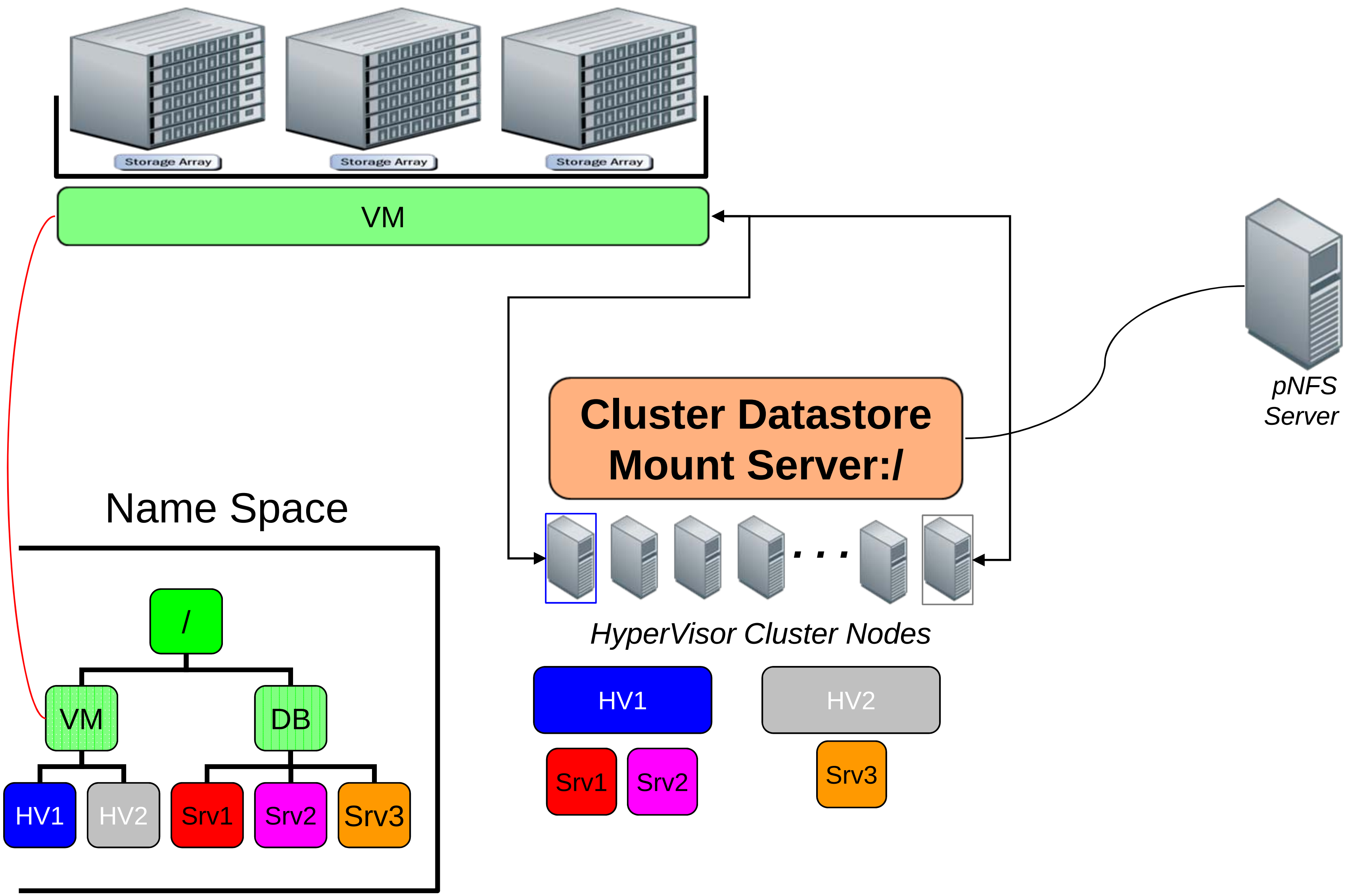


VM Cluster Datastore

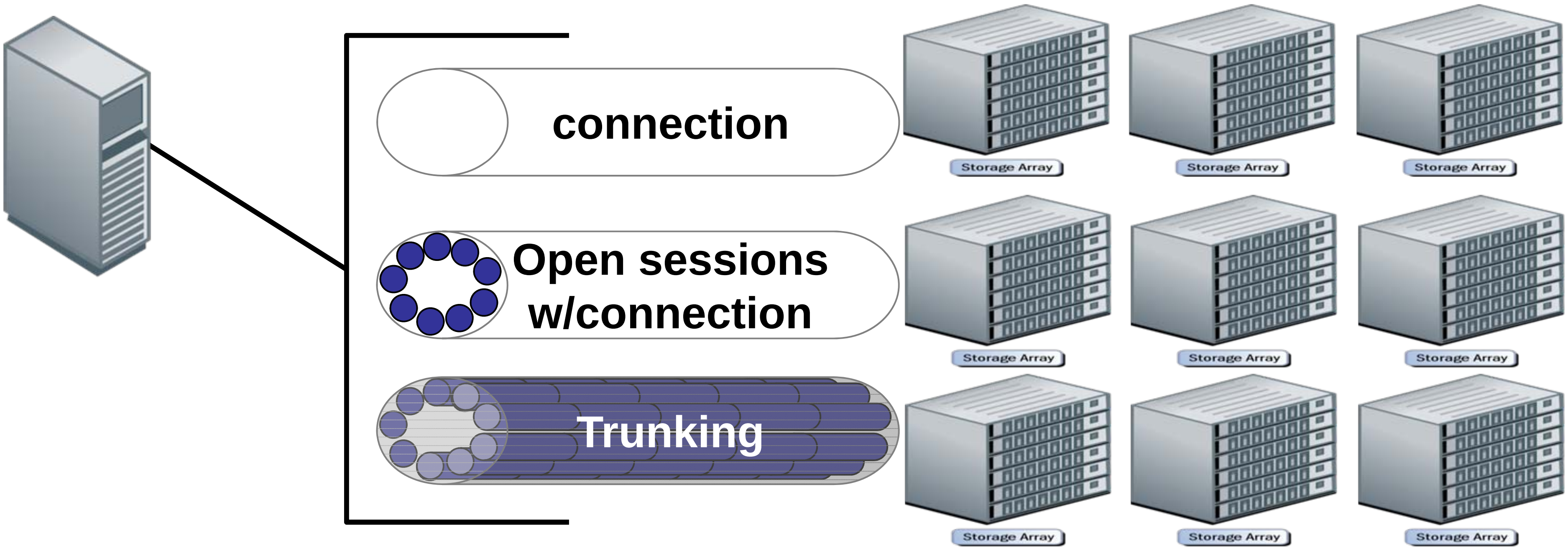
Education SNIA



VMs accessing volume w/layout

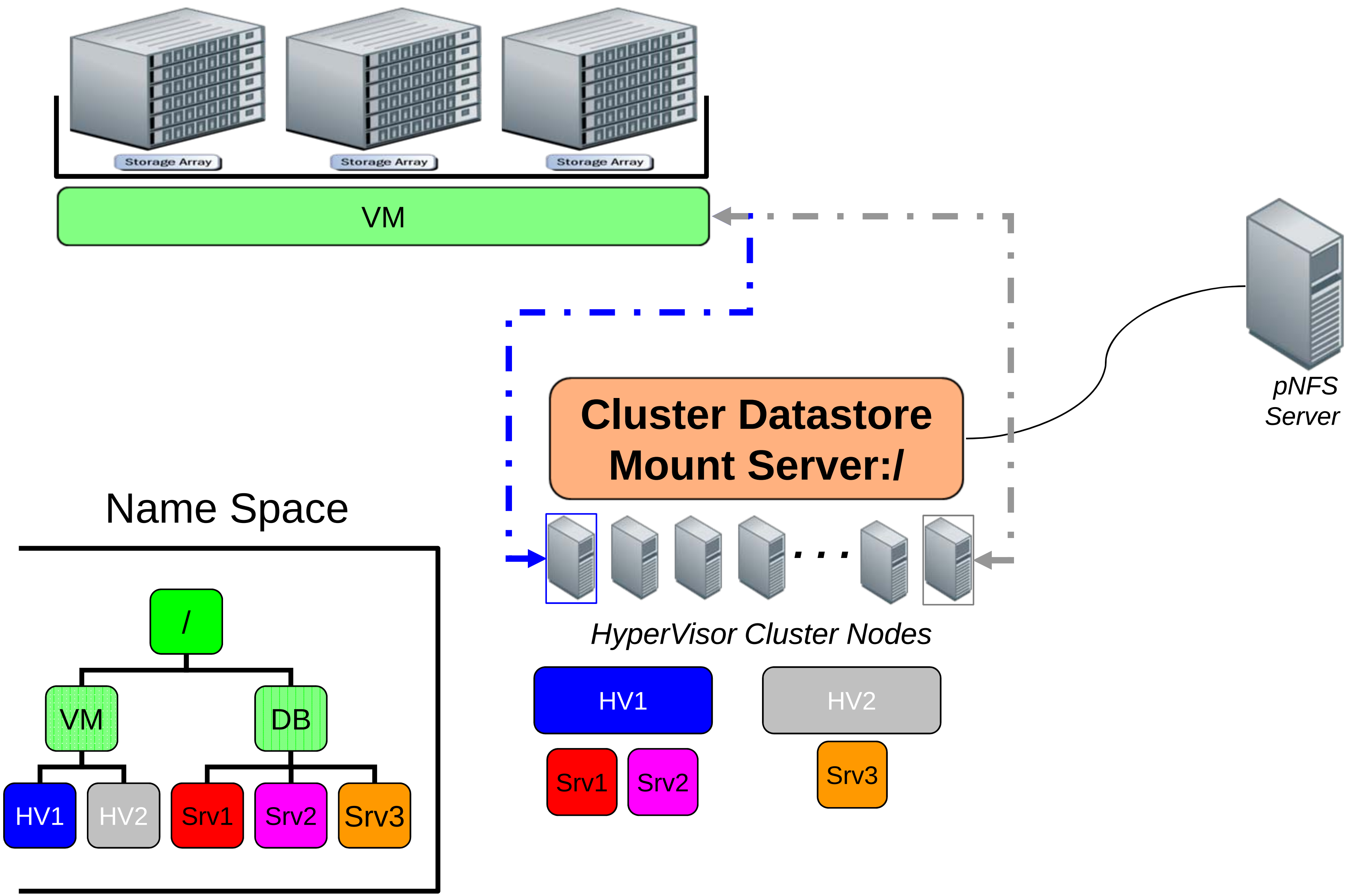


NFSv4.1 Trunking/Sessions



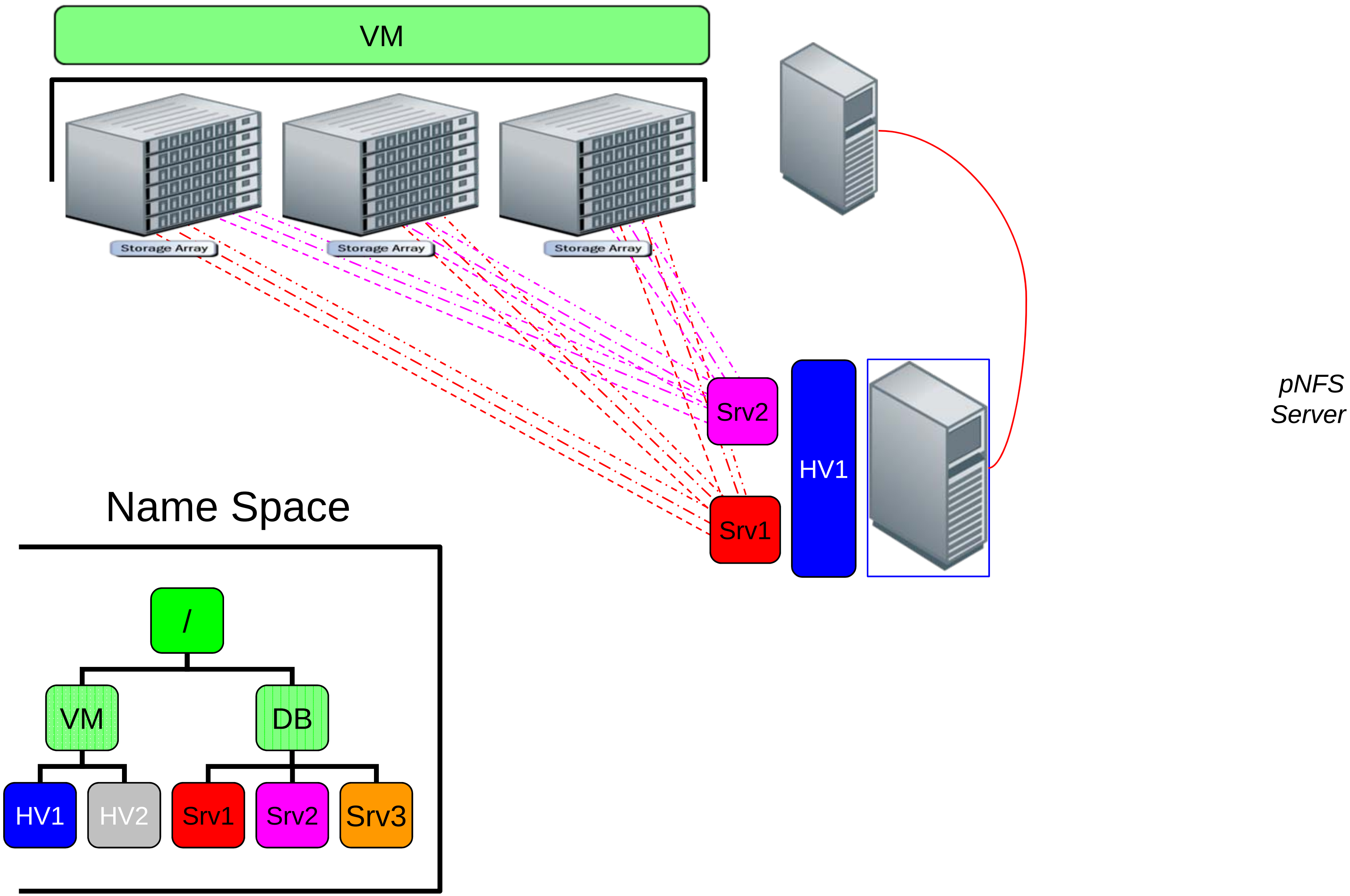
1. A single connection limits data throughput based on protocol
2. Trunking expands throughput and can reduce latency by opening multiple sessions to the same file handle/server resource
   - Host application consumes 10GigE bandwidth

VM Access using single mount

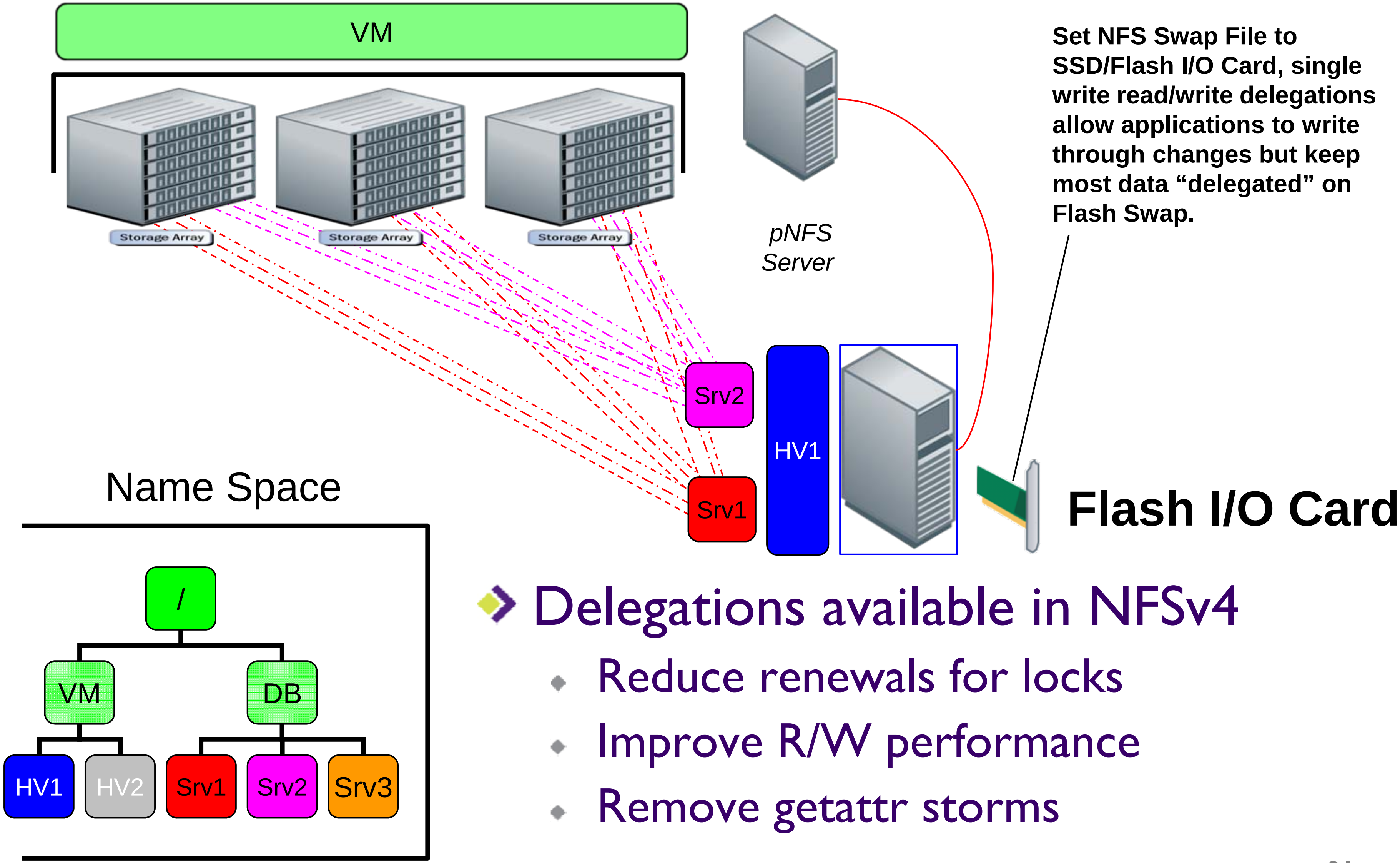


VM access using pNFS + Trunking

Education SNIA



NFSv4.1 Directory/File Delegations

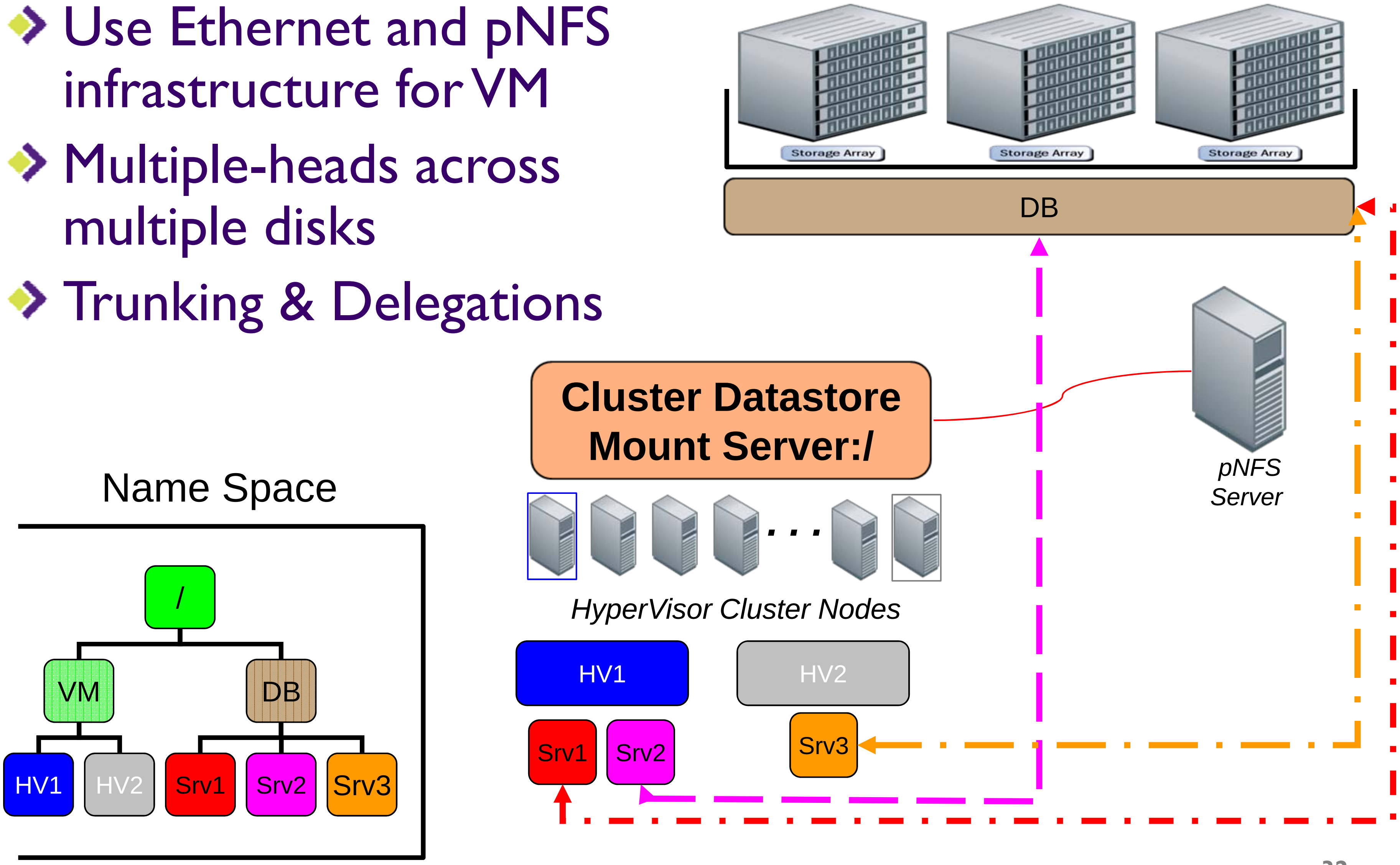


Set NFS Swap File to SSD/Flash I/O Card, single write read/write delegations allow applications to write through changes but keep most data "delegated" on Flash Swap.

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

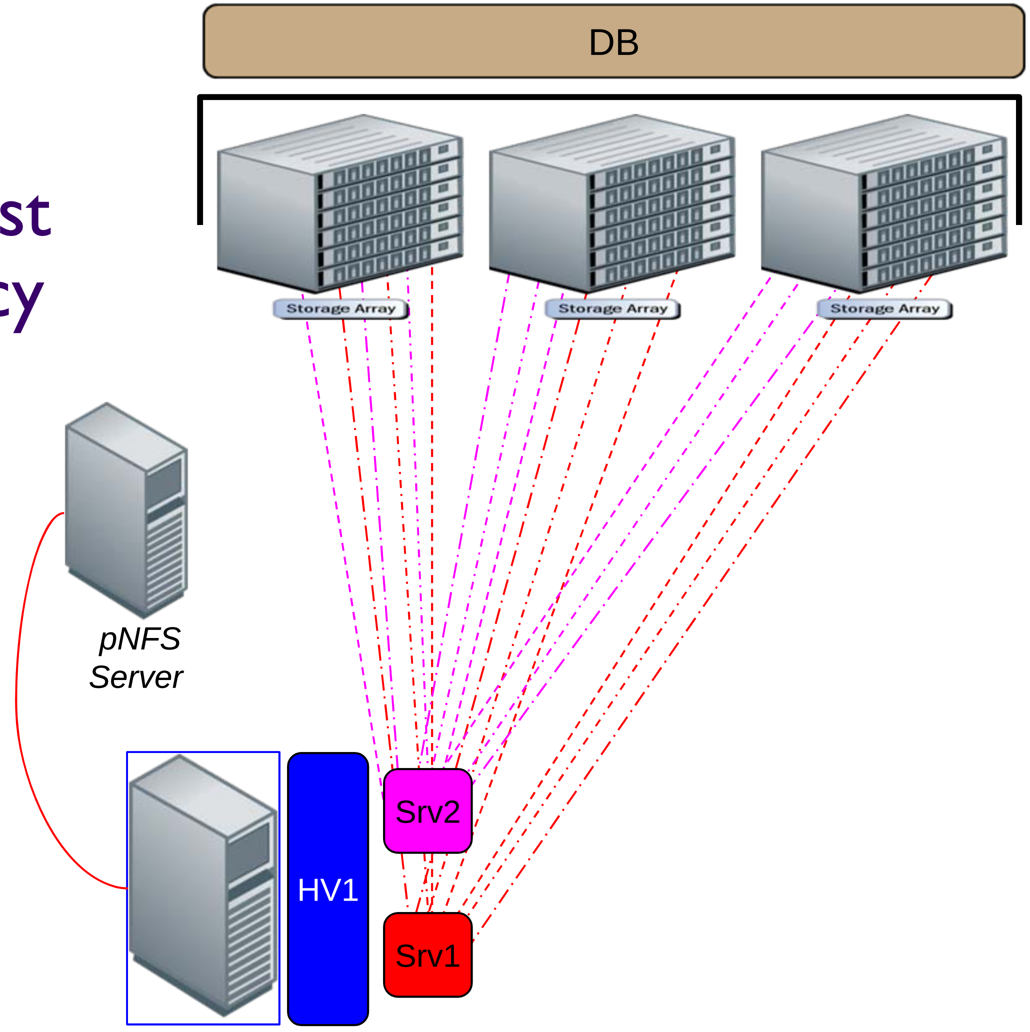
NFSv4.1 – Database enhancements

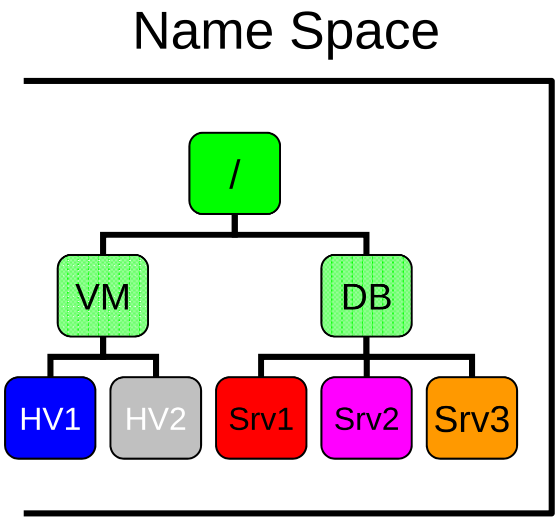
- Use Ethernet and pNFS infrastructure for VM
- Multiple-heads across multiple disks
- Trunking & Delegations



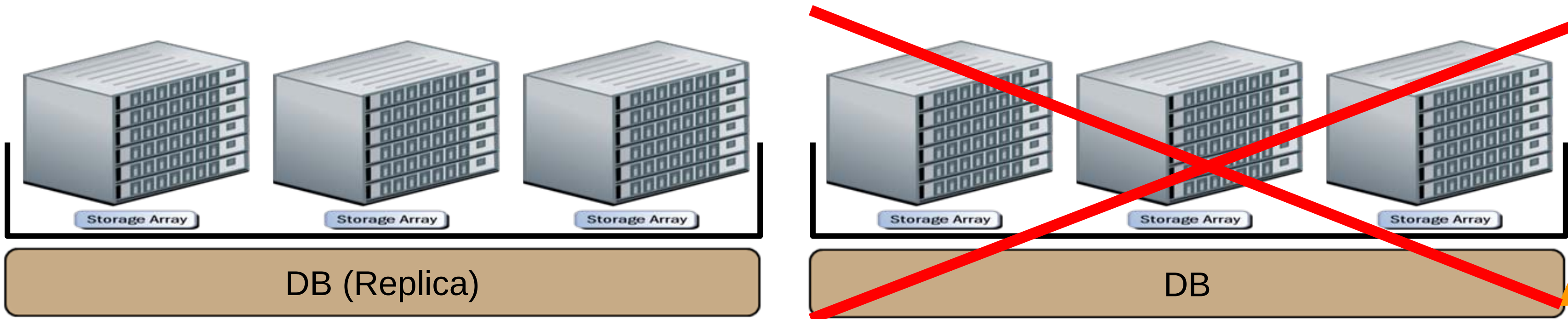
DB access using pNFS + Trunking

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

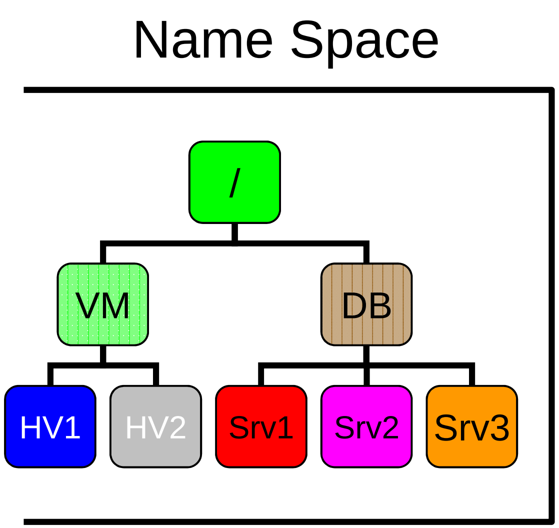


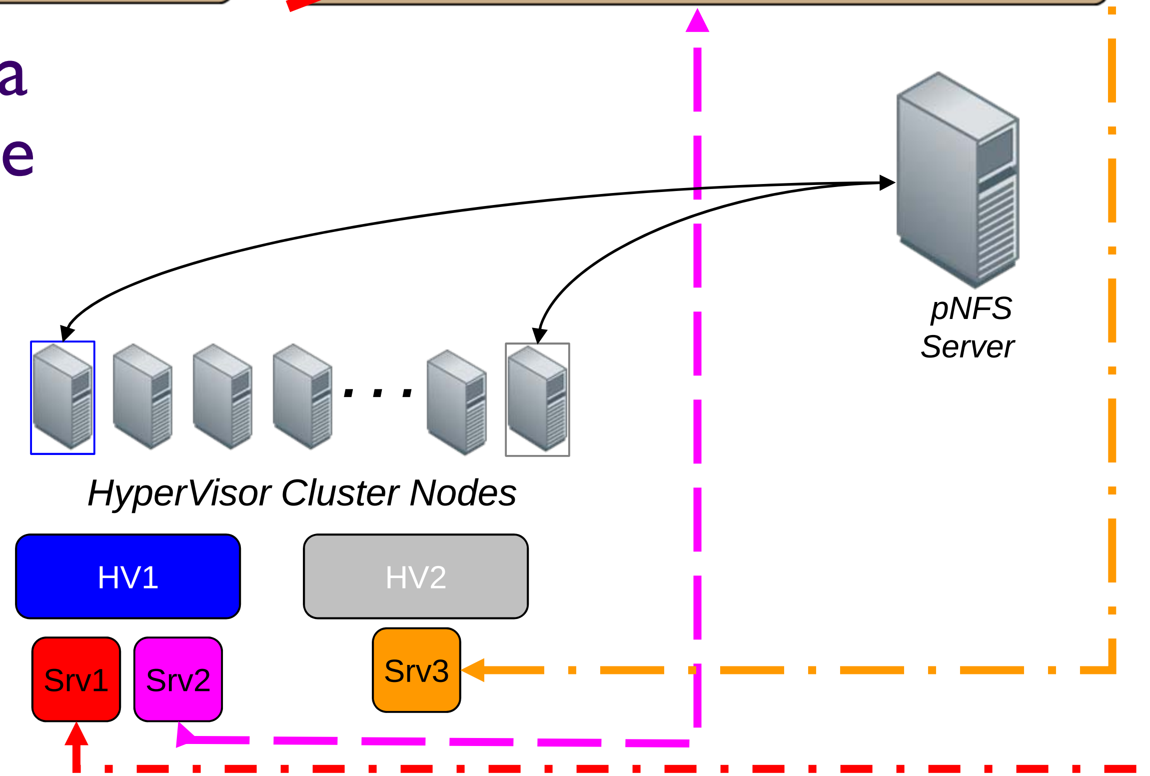


NFSv4.1 – Layout Callbacks

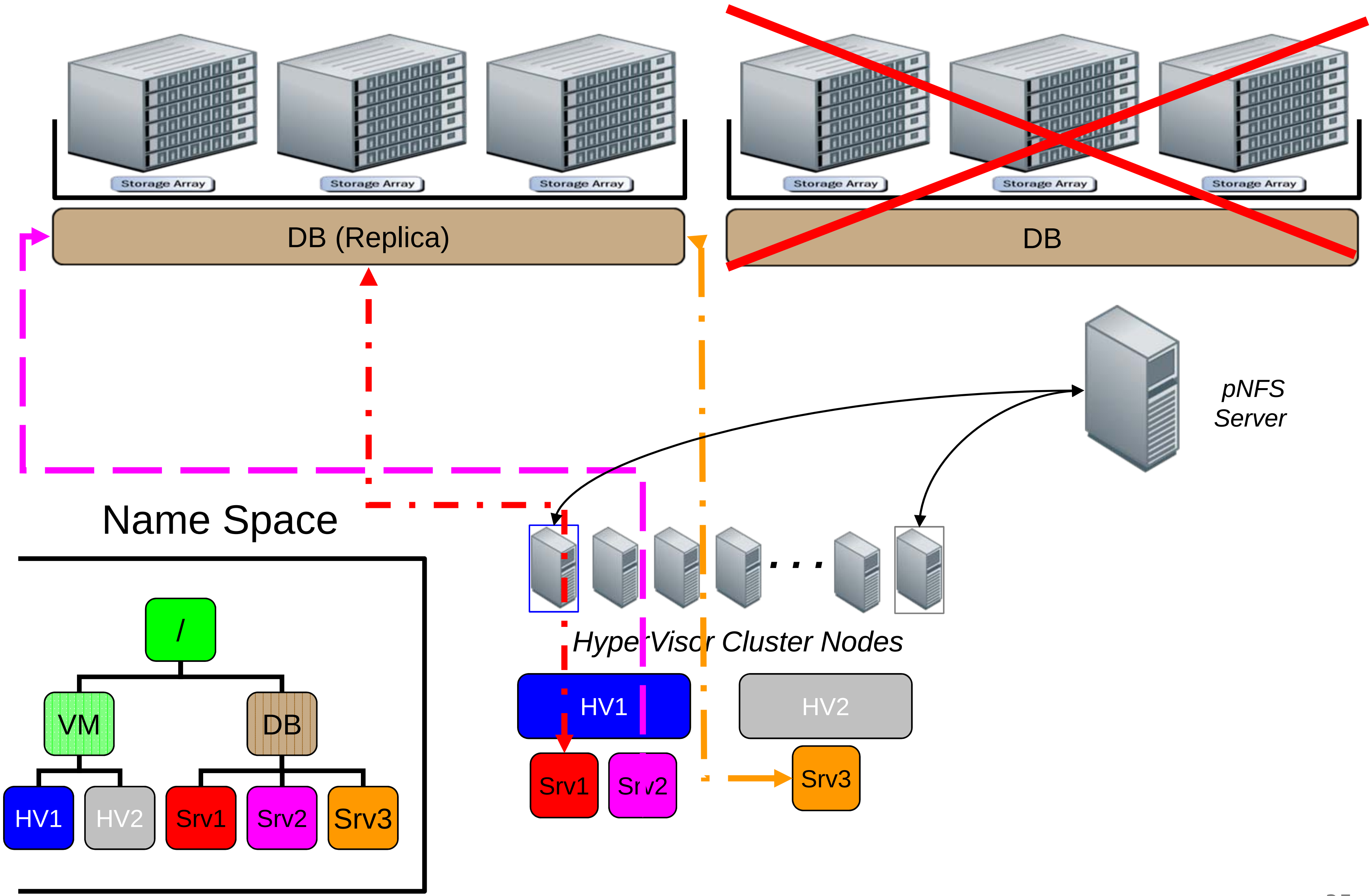


- Non-disruptive data moves using storage control protocols

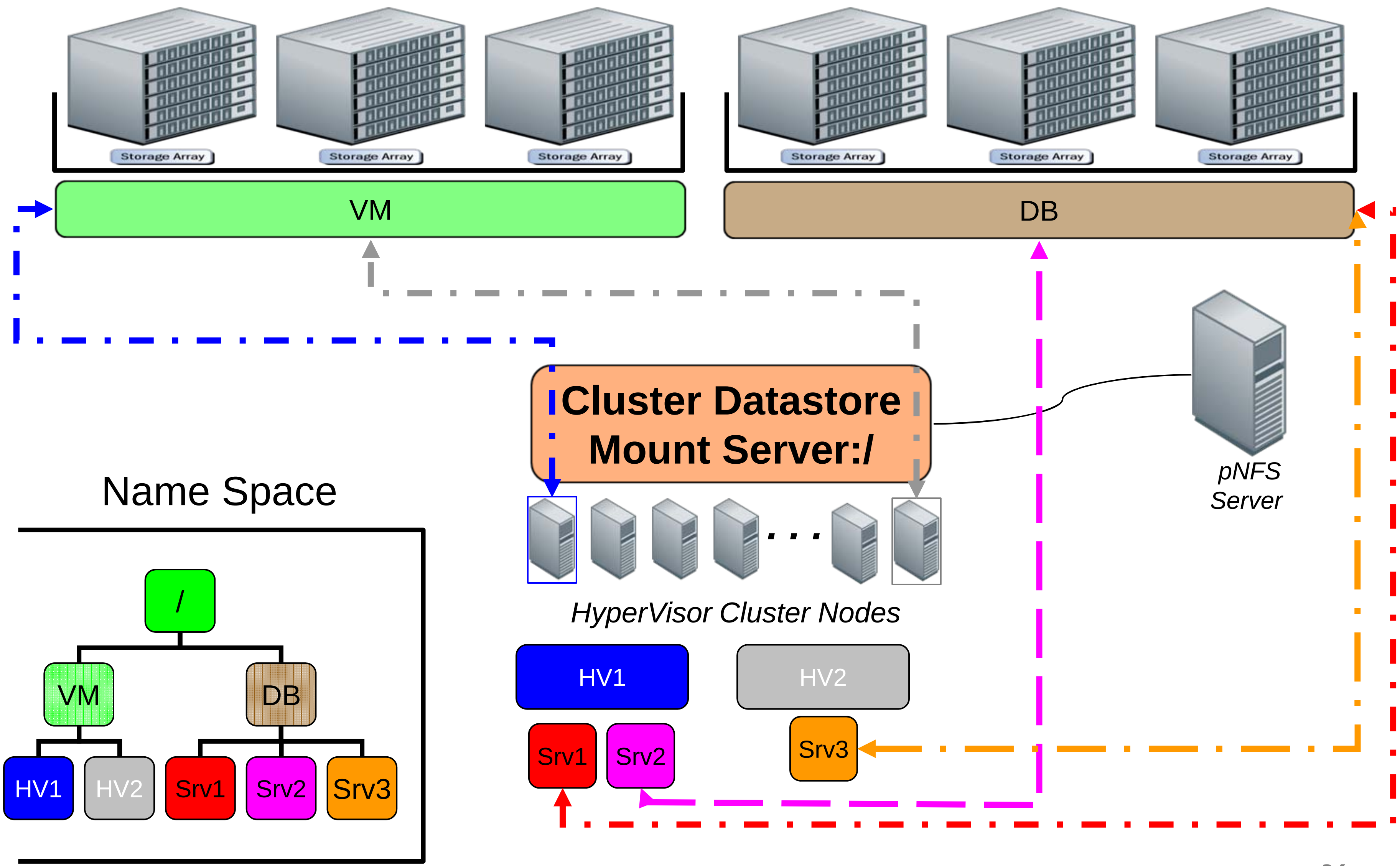




NFSv4.1 – Layout Callbacks



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 today
  - Eases transition to pNFS

Q&A / Feedback

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

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

- SNIA Education Committee

Mike Eisler,	J. Bruce Fields
Brian "Beepy" Pawloski	Joe White
Howard Goldstein	Brent Welch
David Black	Ken Gibson
Omer Asad	Sachin Chheda
Jason Blosil	Piyush Shivam
Mark Carlson	Sorin Faibash
Rob Peglar	Andy Adamson
Dave Hitz	Pranoop Ersani
Ricardo Labiaga	Dave Noveck

SNIA Ethernet Storage Forum

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

NFSv4.1 – Status and Overview

Education SNIA

- 2004 – CMU, NetApp and Panasas draft pNFS problem and requirement statements
- 2005 – CITI, EMC, NetApp and Panasas draft pNFS extensions to NFS
- 2005 – NetApp and Sun demonstrate pNFS at Connectathon
- 2005 – pNFS added to NFSv4.1 draft
- 2006 - 2008 – specification baked
  - Bake/Connect a thons; 29 iterations of NFSv4.1/pNFS spec
- 2008 – NFSv4.1/pNFS reaches IETF Approval (December)

pNFS Standards Status

- NFSv4.1/pNFS were standardized at IETF
  - NFSv4 working group (WG)
- All done except for RFCs:
  - WG last call (DONE)
  - Area Director review (DONE)
  - IETF last call (DONE)
  - IESG approval for publication (DONE)
  - IANA review (TBD)
  - RFC publication (Expected 2009)
- Will consist of several documents:
  - NFSv4.1/pNFS/file layout
  - NFSv4.1 protocol description for IDL (rpcgen) compiler
  - blocks layout
  - objects layout
  - netid specification for transport protocol independence (IPv4, IPv6, RDMA)