pNFS State of the Union
FAST-11 BoF

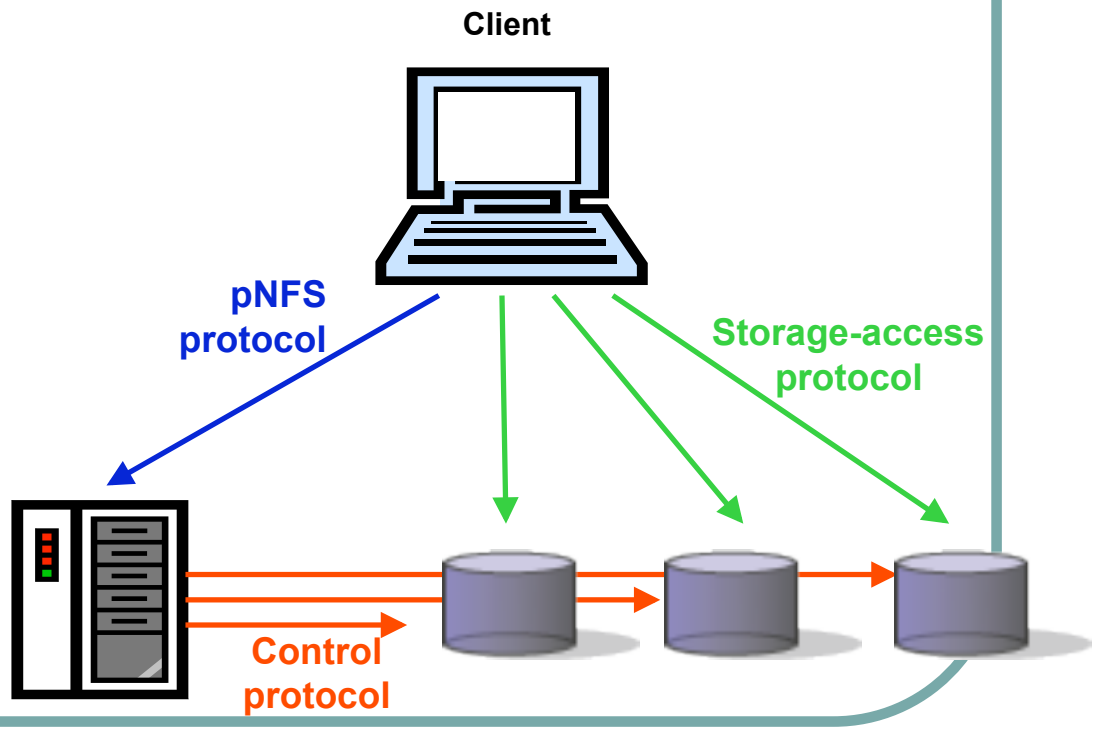
Sorin Faibish- EMC, Peter Honeyman - CITI

Outline

- What is pNFS?
- pNFS Tutorial
- pNFS Timeline
- Standards Status
- Industry Support
- EMC Contributions
- Q&A

What is pNFS?

- pNFS protocol
  - standardized: NFSv4.1
- Storage-access protocol
  - files (NFSv4.1)
  - blocks (FC, iSCSI, FCoE, IB)
  - objects (OSD2)
- Control protocol
  - Outside of the pNFS standard
- Distributes data across storage cluster
- Eliminates or reduces load and capacity balancing
- And yes: can accelerate I/O

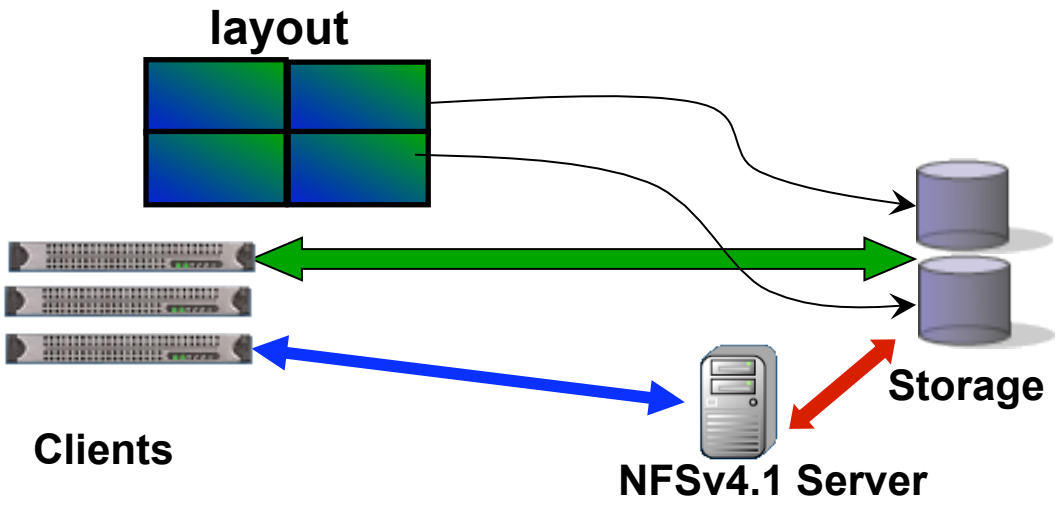


pNFS Value Proposition

- Distributes data across storage cluster
- Eliminates or reduces load and capacity balancing
- And yes: can accelerate I/O

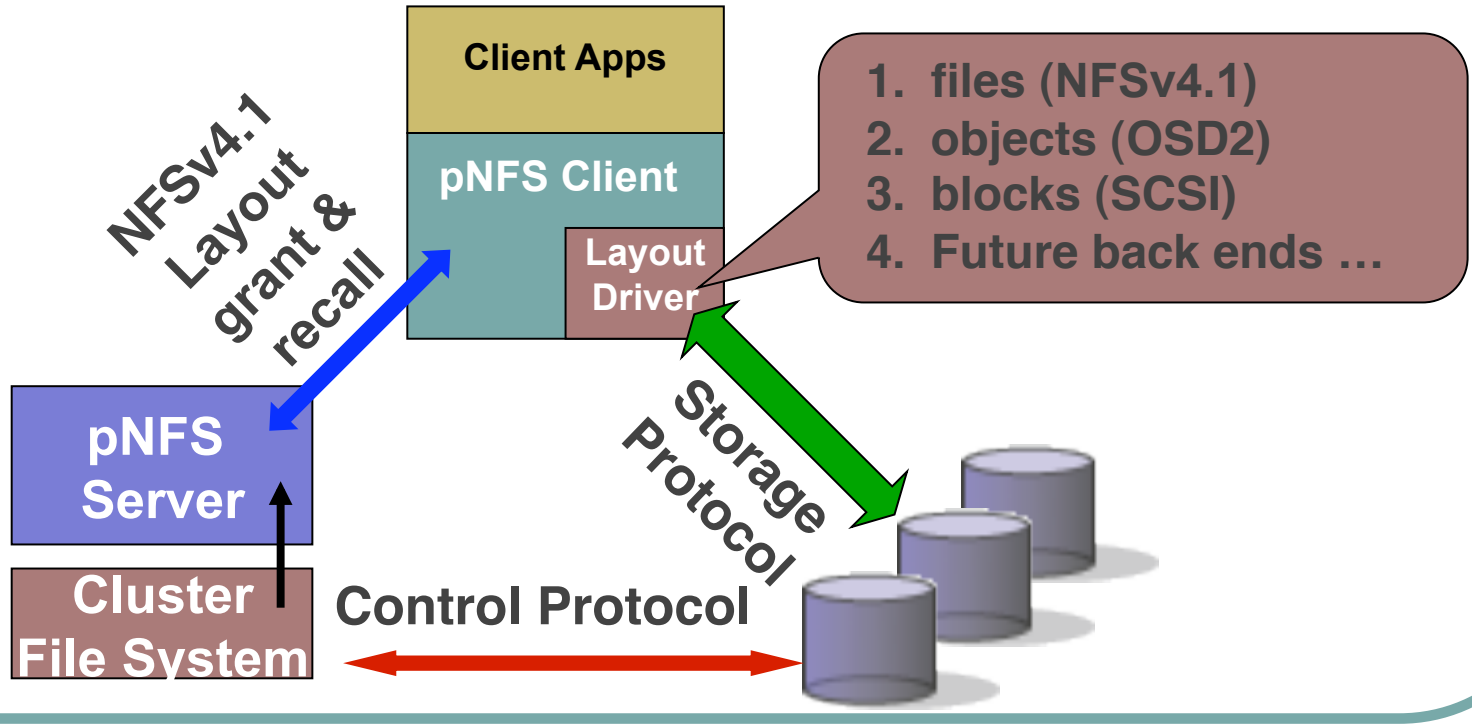
pNFS Layouts

- Client gets a *layout* from the NFSv4.1 server
  - Layout is a map: file offset ➔ storage device + address
  - Layout can be recalled by MDS
- The client uses layout to perform I/O directly to storage
- Client commits changes, optionally returns layout
- pNFS is optional: MDS must be prepared for client I/O requests



Linux pNFS Client

- Transparent to applications
- Common client for different storage back ends
- Fewer support issues for storage vendors
- Normalizes access to clustered file systems



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
- Performance via Delegations
  - File Delegations allow client workloads for single writer and multiple reader
  - Client's can perform all reads/writes in local client cache
  - Delegations are leased and must be renewed
  - Delegations reduce lease lock renewal traffic

NFSv4.1 – OpenSource Status

- Two OpenSource Implementations
  - OpenSolaris and Linux (file, osd and block)
- OpenSolaris Client and Server
  - Support only file-based layout
  - S
    upport 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)
  - Cli
    e
    nt consists of generic pNFS client and “plug ins” for “layout drivers
- Windows NFSv4.1 Client from CITI - NEW

NFSv4.1 – OpenSource (EMC view)

- Linux pNFS block layout Client and Server
  - Support generic layout – CITI Univ. of Michigan
  - Support block layout client (SCSI) - F-15 CITI
  - Block layout s e rver (LSI + EMC) first release available in F-16 (EMC CITI nursing)
  - Maintenance of block layout – CITI via Linux kernel Bugzilla
  - Performance monitoring and patching – CITI via Bugzilla
- Support to Linux Distributions
  - Support block layout – Elab work with RedHat to qual pNFS
  - EMC Elab will qualify Fedora 15

Timeline

- 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-a-thons (Last in EMC), Connectathons
  - 26 iterations of NFSv4.1/pNFS spec
- 2009 – RFC submitted (680 pages)
- 2010 – RFC published
- 2011 – Fedora 14 includes pNFS server/client gits and rpms (did you try it yet?)

pNFS Standards Status

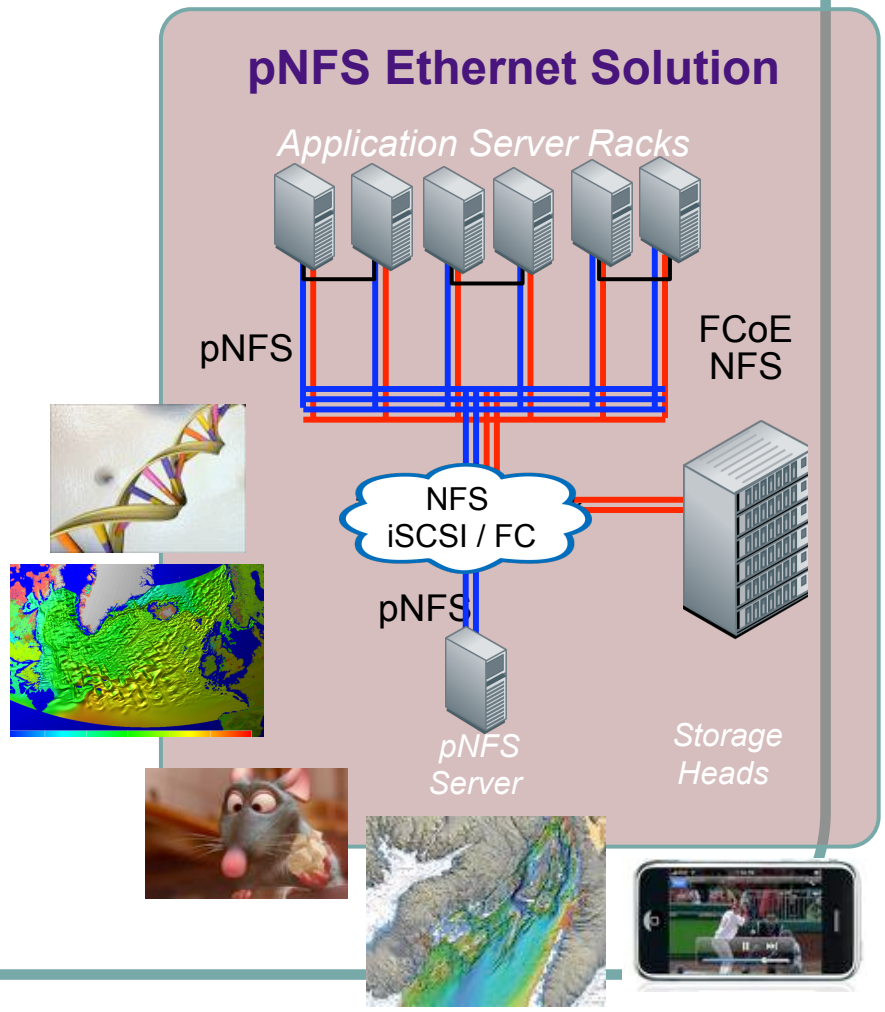
- NFSv4.1/pNFS were standardized at IETF
  - NFSv4 working group (WG)
- All done including RFC 5661,3,4:
  - WG last call (DONE)
  - Area Director review (DONE)
  - IETF last call (DONE)
  - IESG approval for publication (DONE)
  - IANA review (DONE)
  - RFC publication (2010)
- Consists of several documents:
  - RFC 5661 - NFSv4.1/pNFS/file layout
  - RFC 5662 - NFSv4.1 protocol description for IDL (rpcgen) compiler
  - RFC 5663 - blocks layout
  - RFC 5664 - objects layout
  - RFC 5665 - netid specification for transport protocol independence (IPv4, IPv6, RDMA)

pNFS Implementation Timeline

- NFSv4.1/pNFS Linux server/client all layouts – Fedora 15+
- NFSv4.1/pNFS Linux server/client file layout – RHEL 6.1
- NFSv4.1/pNFS ESX client file and block layouts – Demo in current ESX release: EMC + NetApp
- Celerra NFSv4.1/pNFS block server support – Released (Q3 ‘10)

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



Virtualization Use Case #1: Storage VMotion

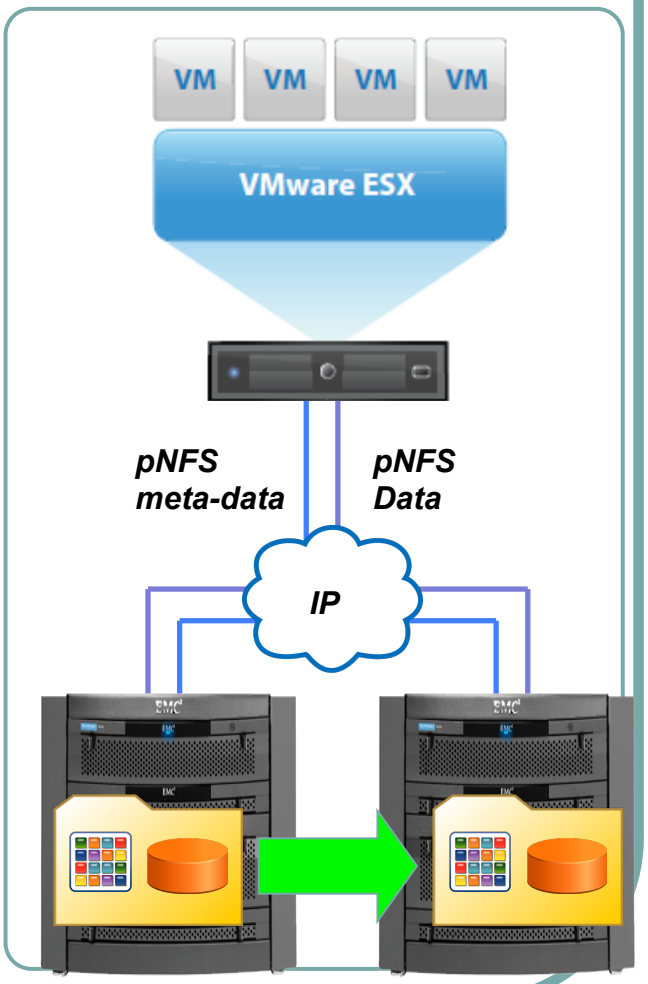
Scenario: Hundreds of virtual machines. Virtual machine disk files stored on shared NAS storage. Upgrade / maintenance on storage array, requires virtual machine disk files to be migrated from one storage array to another with no downtime or service disruption.

Application: Use VMware Storage VMotion to migrate the virtual machine disk files.

Challenge: Hundreds of virual machines that are continuously being accessed. Migrations are taking longer due to copying of many large virtual machine disk files over NAS and number of iterations.

pNFS Benefits:

- ✓ *Multiple ESX servers hosting hundreds of VMs can copy in parallel*
- ✓ *Large virtual machine disk files can be copied faster, resulting in fewer iterations and faster turnaround*
- ✓ *Simplicity of NAS storage with increased performance and scalability*



pNFS enabled NAS storage

Virtualization Use Case #2: Clone Virtual Machines

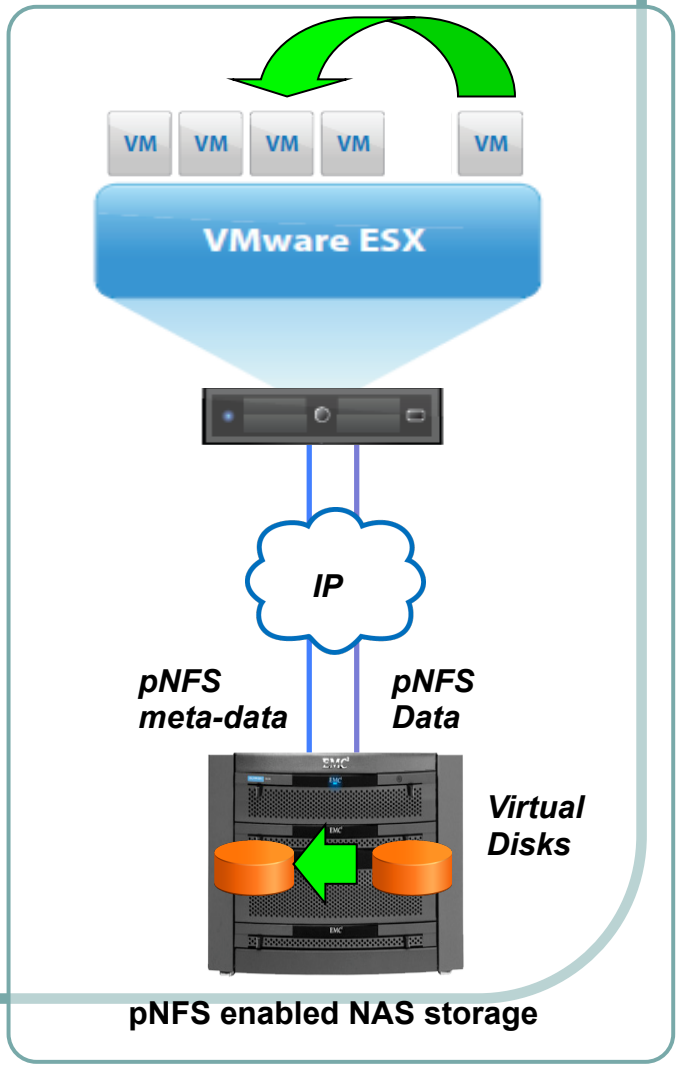
Scenario: Virtual machines added by cloning a master image file. Virtual machine disk files stored on shared NAS storage.

Application: Use VMware Clones to clone the virtual machine disk files.

Challenge: Quickly create hundreds of clones to add new virtual machines.

pNFS Benefits:

- ✓ *Large virtual machine disk files can be read and cloned faster*
- ✓ *Simplicity of NAS storage with increased performance and scalability*



Questions – Client Technology

- When will an OTS pNFS client be available?
  - Fedora 14 include all 3 layouts. RHEL 6.2 file. (¿6.3 block+object?)
  - Linux (include disclaimer)
    - 2.6.37 (stable): layout ops
    - 2.6.38 (mainline): layout management
    - 2.6.39: FILE READ
    - 2.6.40: FILE WRITE
    - 2.6.41 and beyond: OSD, block, tuning, features
  - CITI Windows client available for experimental use. Active development.
- How do current cluster file systems work with pNFS client and server?
  - Used as data servers but only the asymmetric ones; Lustre and PVFS have experimental patches. (GPFS?)
- How is client access to data servers coordinated and controlled?
  - Client uses layouts sent/managed by the MDS
- Audience Questions Encouraged…

Questions – Files, Blocks and Objects

- How many layout types can there be?
  - Unlimited. Three IETF standards, others evaluated experimentally.
- How does the pNFS block layout work?
  - Talk MD to MDS and data to SCSI LUNs
- What's an object layout, how do objects compare to blocks/files?
  - Objects are an extension of NASD protocol
- Can my application control how its data is striped?
  - Currently app can only hint but 4.2 will include protocol attributes
- Can File, Object, and Block co-exist in the same storage network?
  - Yes. Even access same FS. Even same file.
- Can a client use volumes accessed via each layout concurrently?
  - Yes, if the server implementation supports that.
- Audience Questions Encouraged…

Questions – Data Management

- How does pNFS make managing a lot of systems easier?
  - Clients can ensure consistency of MD with the MDS
- Will pNFS allow non-disruptive (NDU) upgrades?
  - No more disruptive than Linux/Windows upgrades …
- Can I retain data management practices I use today, e.g, Snapshots and Volume replication?
  - Value added by vendors, Linux BTRFS
- I'm deploying a Unified Ethernet Fabric; how do I secure data access – files, blocks, objects?
  - Specific to each layout type: GSS & ACLs for file; Zoning for iSCSI block, capabilities for objects.
- Audience questions encouraged…

Questions – What else

- When can we expect to see real pNFS performance, not vendor claims based on older technology?
  - Much performance and scalability analysis at CITI, PDL, others
- How will NFSv4.1 and pNFS be received, compared to NFSv4.0?
  - NFSv4.1 and pNFS extensions were added after listening to users
- What would you like to see in NFSv4.2?
  - Sparse files? Device Access Control? Storage Preferences? Metadata Striping?
- What additional functionality do you want to see added to NFSv4.x:
  - FedFS
  - Server side copy
- Audience questions encouraged…

Audience questions

- Audience questions required…