pNFS BOF
SC08
2008-11-19

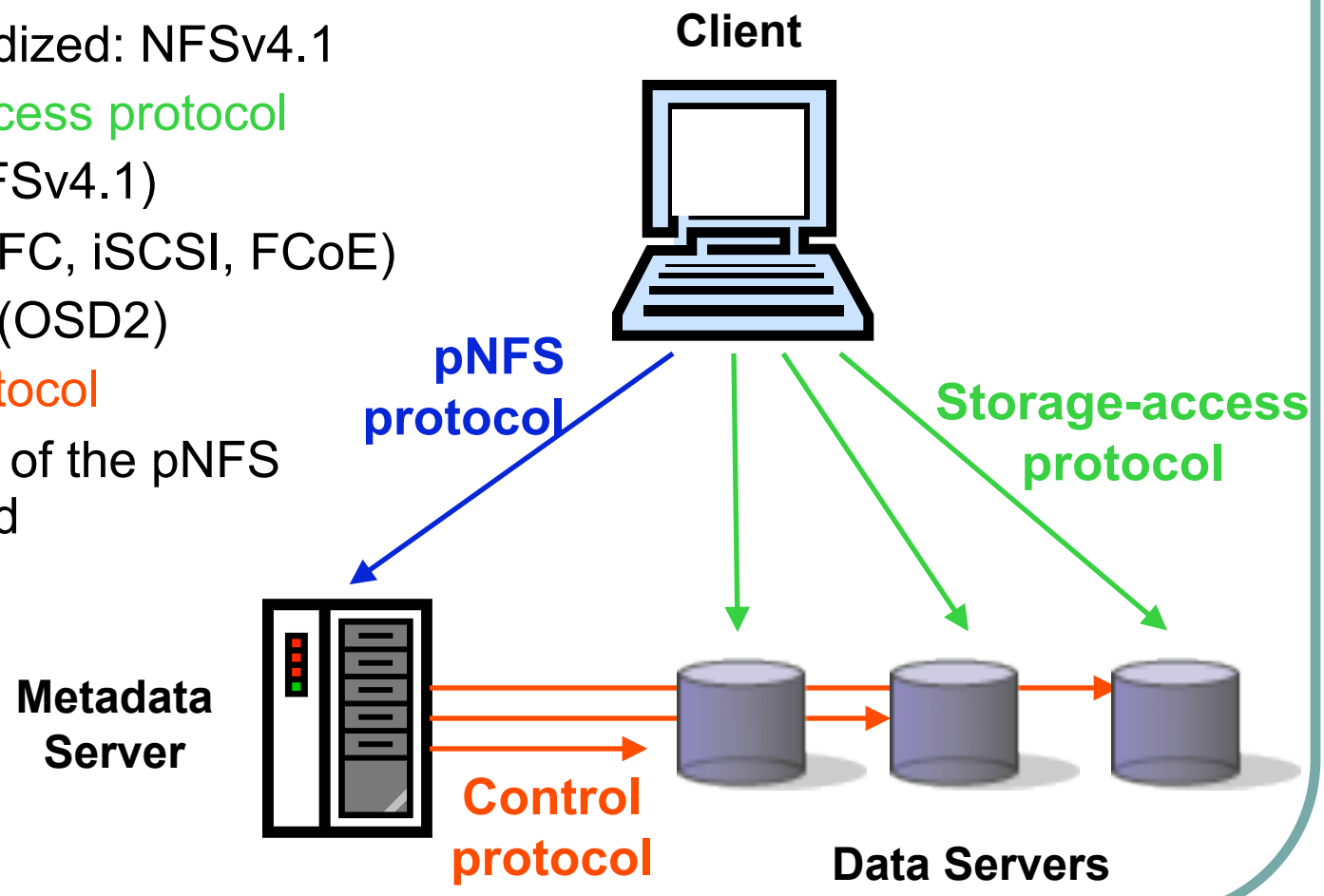
- Spencer Shepler, **StorSpeed**
- Bruce Fields, **CITI (University of Michigan)**
- Sorin Faibish, **EMC**
- Roger Haskin, **IBM**
- Ken Gibson, **LSI**
- Joshua Konkle, **NetApp**
- Brent Welch, **Panasas**
- Bill Baker, **SUN Microsystems**

Outline

- What is pNFS?
- pNFS Timeline
- Standards Status
- Industry Support
- Linux Status
- Vendor Presentations
  - EMC, IBM, LSI, NetApp, Panasas

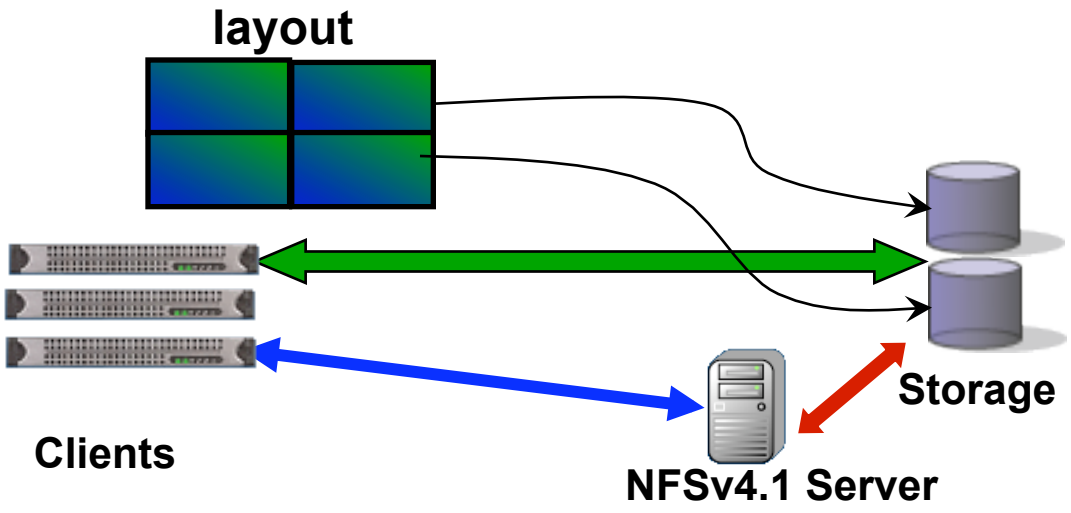
What is pNFS?

- pNFS protocol
  - standardized: NFSv4.1
- Storage-access protocol
  - files (NFSv4.1)
  - blocks (FC, iSCSI, FCoE)
  - objects (OSD2)
- Control protocol
  - Outside of the pNFS standard



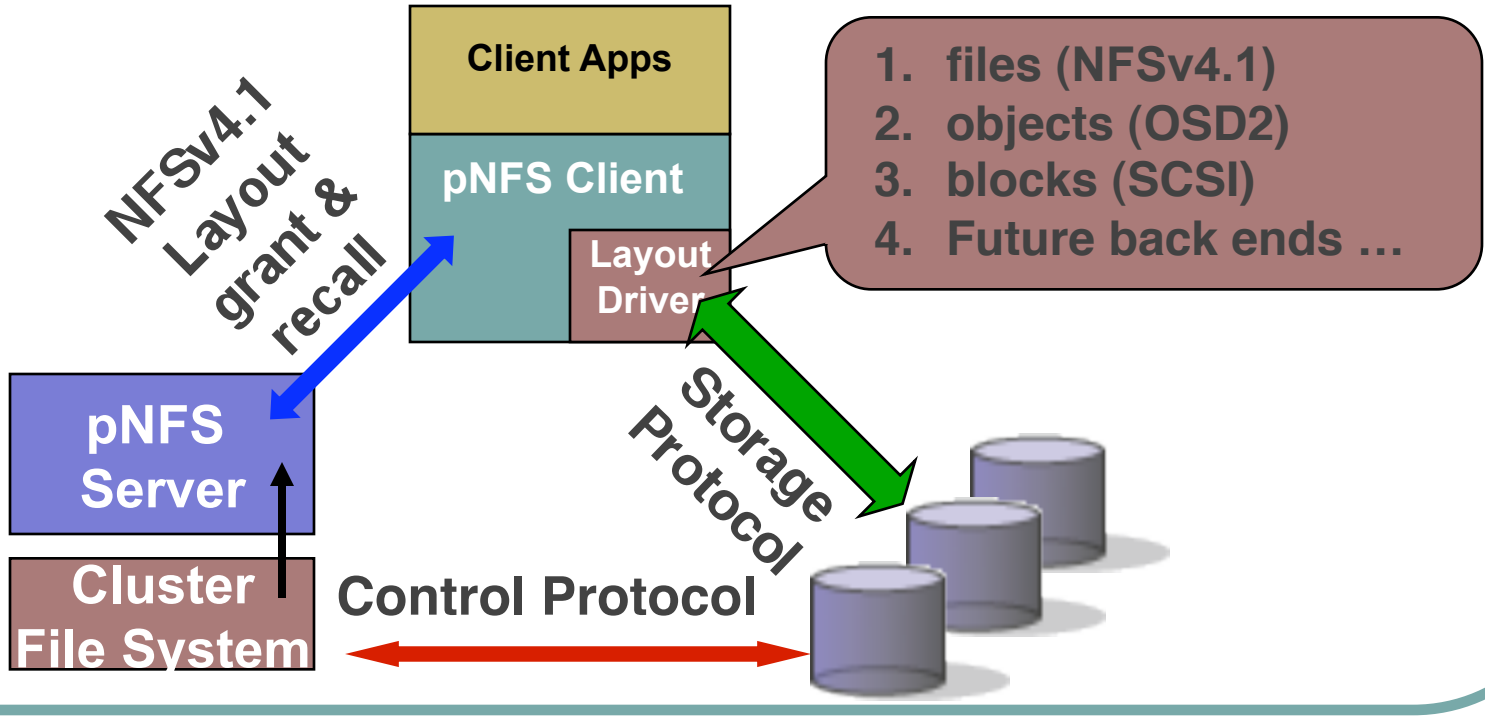
pNFS Layouts

- Client gets a *layout* from the NFSv4.1 server
- The layout maps the file onto storage devices and addresses
- The client uses the layout to perform direct I/O to storage
- At any time the server can recall the layout
- Client commits changes and returns the layout when it's done
- pNFS is optional, the client can always use regular NFSv4.1 I/O



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



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, Connectathons
  - 26 iterations of NFSv4.1/pNFS spec
- 2008 – NFSv4.1/pNFS reaches IETF Last Call

pNFS Standards Status

- NFSv4.1/pNFS are being standardized at IETF
  - NFSv4 working group (WG)
- In the end game:
  - WG last call (DONE)
  - Area Director review (DONE)
  - IETF last call (November, 2008)
  - IANA review (TBD)
  - IESG approval for publication (Expected December, 2008)
  - RFC publication (Expected early 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)

Industry Contributors to pNFS Standard

- BlueArc
- CITI
- CMU
- EMC
- IBM
- LSI
- NetApp
- Ohio SuperComputer Ctr
- Panasas
- Seagate
- StorSpeed
- Sun Microsystems

Industry Support - Implementations

- Clients
  - Linux
  - Sun (Solaris)
- Servers
  - Desy
  - EMC
  - IBM
  - Linux
  - NetApp
  - Panasas
  - Sun (Solaris)

Several other implementations have been tested at Bake-a-thons and Connectathons

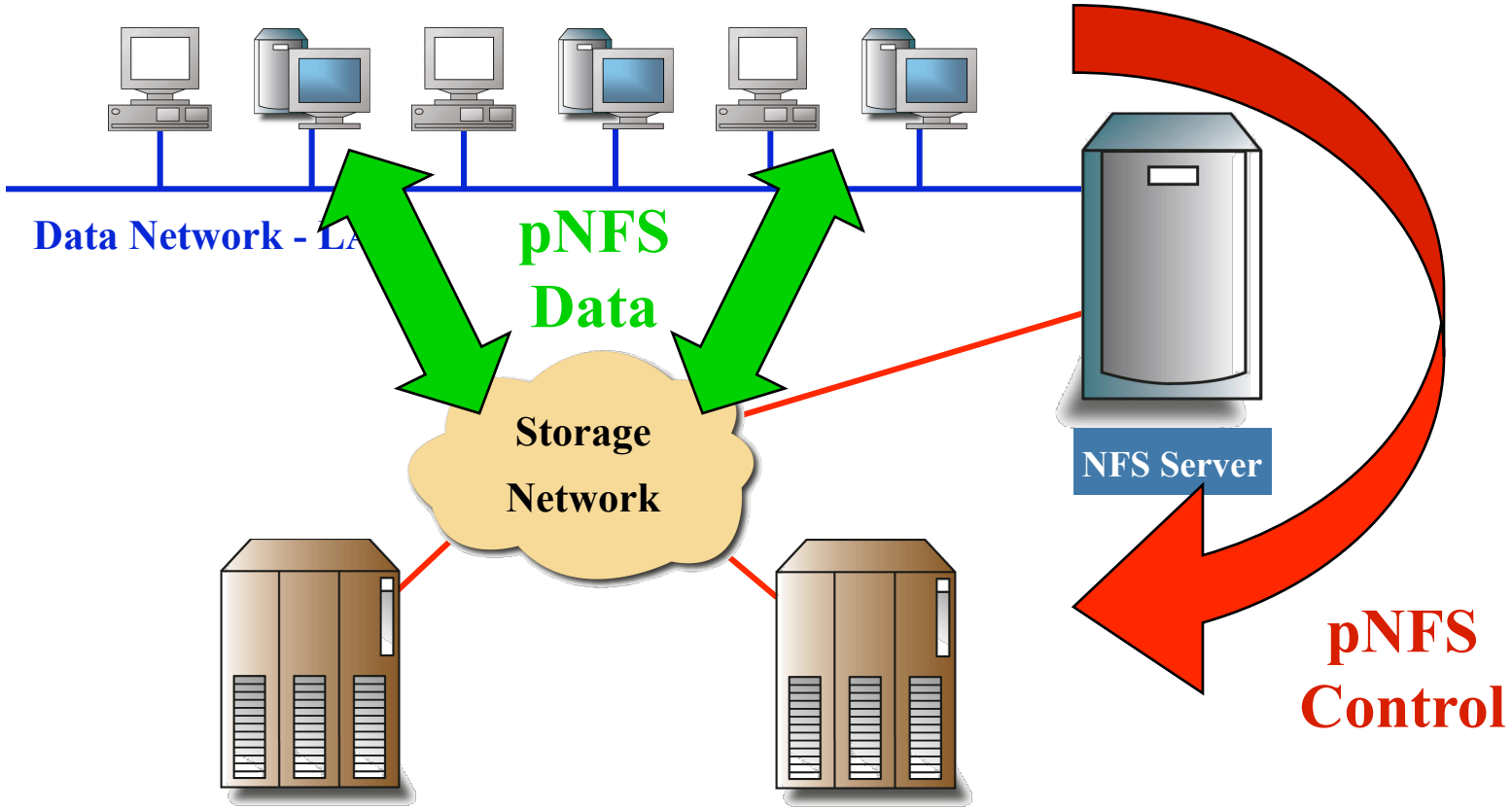
Linux Status

- Client
  - Consists of generic pNFS client and “plug ins” for “layout drivers”
  - Supports files, blocks, objects
  - Contributors: CITI, EMC, NetApp, Panasas
- Server
  - Supports files, blocks, objects
  - Contributors: CITI, EMC, IBM, NetApp, Panasas
- Finalizing patches for kernel.org – NFSv4.1 sessions
- Predicted timeline:
  - Basic NFSv4.1 features 1H2009
  - NFSv4.1 pNFS and layout drivers by 2H2009
  - Linux distributions shipping supported pNFS in 2010.

EMC and pNFS
SC08

Sorin Faibish – EMC DE
David L. Black – EMC DE
Per Brashers – EMC MPFS Architect

Parallel NFS - pNFS



- NFS file naming, management, and administration
- Parallel high bandwidth file access (via Storage Network)
- Block Layout leverages existent SAN infrastructures

pNFS Block Layout – The beginning

- The ancestors of pNFS Block Layout are NAS accelerators - 1998:
  - EMC-MPFS, Quantum-StoreNext and Mercury-Sanergy
- EMC donate the FMP (MPFS) protocol and IP
  - Open source version of FMP client (iRoad) - 2003
  - IETF pNFS Block Layout = modified open storage FMP protocol - 2004
- EMC support pNFS Block Layout in Linux kernel by join work with CITI: Peter Honeyman, Fred Isaman, Bruce Fields
  - Current pNFS block layout open source client and NFSv4.1 demonstrated at bake-a-thons
  - Ongoing funding the project, in 4th year = strong EMC commitment
  - Customers can experience the value of pNFS using the EMC FMP open source driver, or by installing current shipping MPFS product.

pNFS Block Layout – Now

- pNFS will supports any SAN storage (LSI, EMC, other SAN)
  - Working with other SAN vendors to promote pNFS Block Layout
- EMC plans to support NFSv4.1 and pNFS server only after RFC approval and pNFS clients in Linux kernel
  - Prototype demonstrated at latest Bake-a-thon
  - Demo on Laptop with VM and real clients
- EMC is working with all the pNFS developers to accelerate adoption by HPC
  - The goal is to combine all flavors of pNFS servers accessed by each Linux client in one single infrastructure
  - Working with Linux Distributions and Linux kernel developers
- What value brings pNFS block layout
  - Leverage existent SAN storage and connectivity
  - Allow access to SAN storage by NFSv4 network clients
  - Virtualizes multi-vendor storage arrays into a single unified view

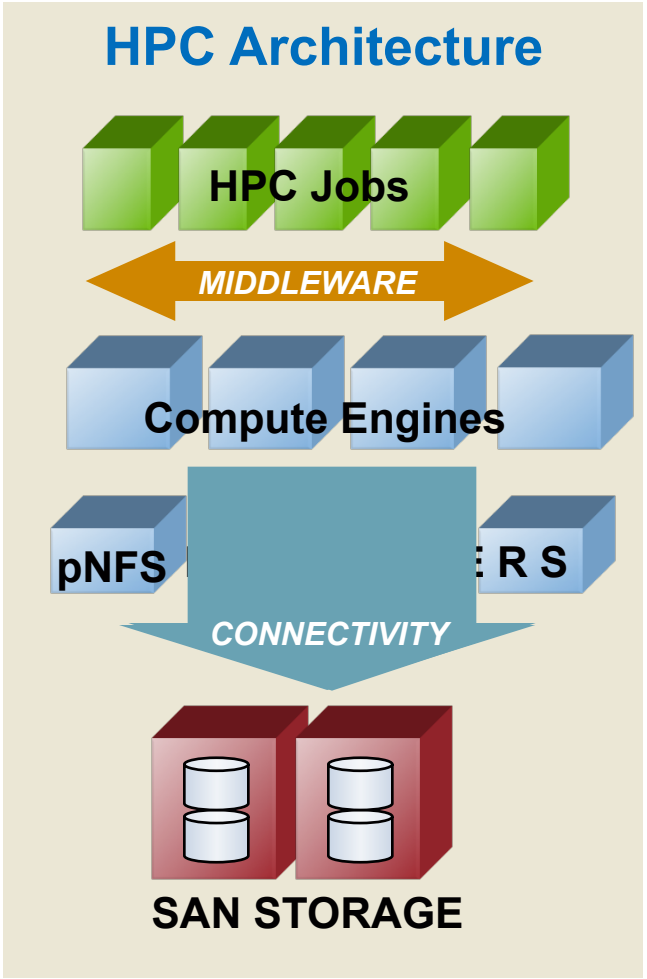
pNFS Block Layout deliver high I/O speeds to HPC

EMC² where information lives

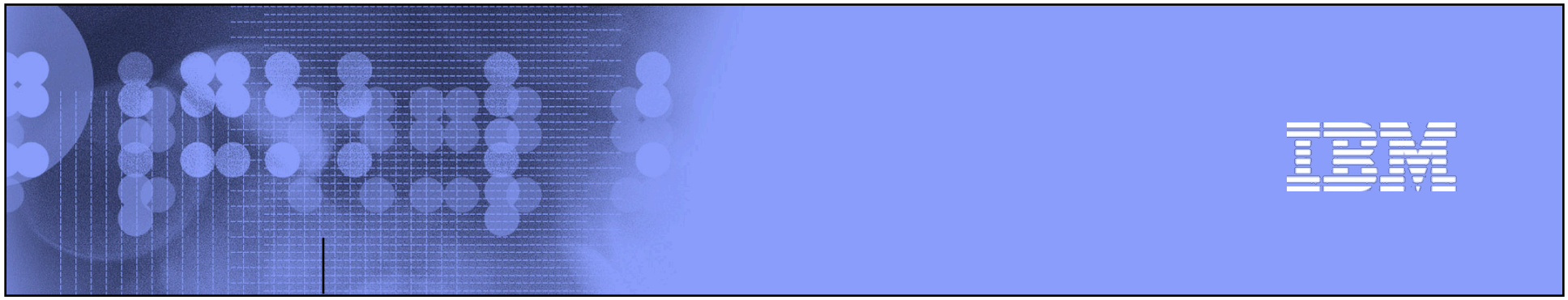
pNFS addresses storage access issues

- Remove servers layer between CE and shared storage
- Separates MD traffic from Data Traffic
- Asymmetric storage architectures increase scalability
- Leverages SSD to increase I/O speed
  - Automatic tiering
- Improves utilization to any SAN infrastructure:
  - FCoE, Infiniband, FC, iSCSI
- Enable access to PB's of storage at GB's/ sec speeds
  - Demonstrated by existent MPFS deployments
- Combine multiple MD servers in a unified storage system
- MD server is any Celerra NAS server supporting:
  - NFSv3, CIFS, MPFS and pNFS
  - Tiered services for increased scalability

Storage must be Networked

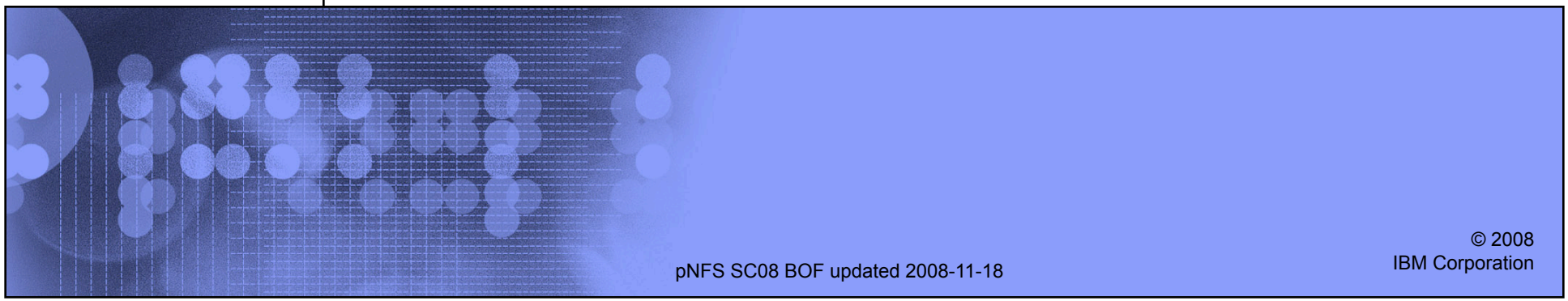



EMC²
where information lives®



GPFS and pNFS

Roger Haskin
Senior Manager, File Systems
IBM Almaden Research Center



GPFS and pNFS

Why are we interested in pNFS?

To augment GPFS, not by any means to replace it!

- Parallel import/export of data into/out of GPFS
- Parallel access to GPFS from unsupported platforms
- Makes GPFS native file system features available to open clients
  - GPFS ILM (storage pools and data migration policies)
  - HPSS, TSM, and other HSM solutions built on GPFS
- To enable GPFS-based pNFS servers

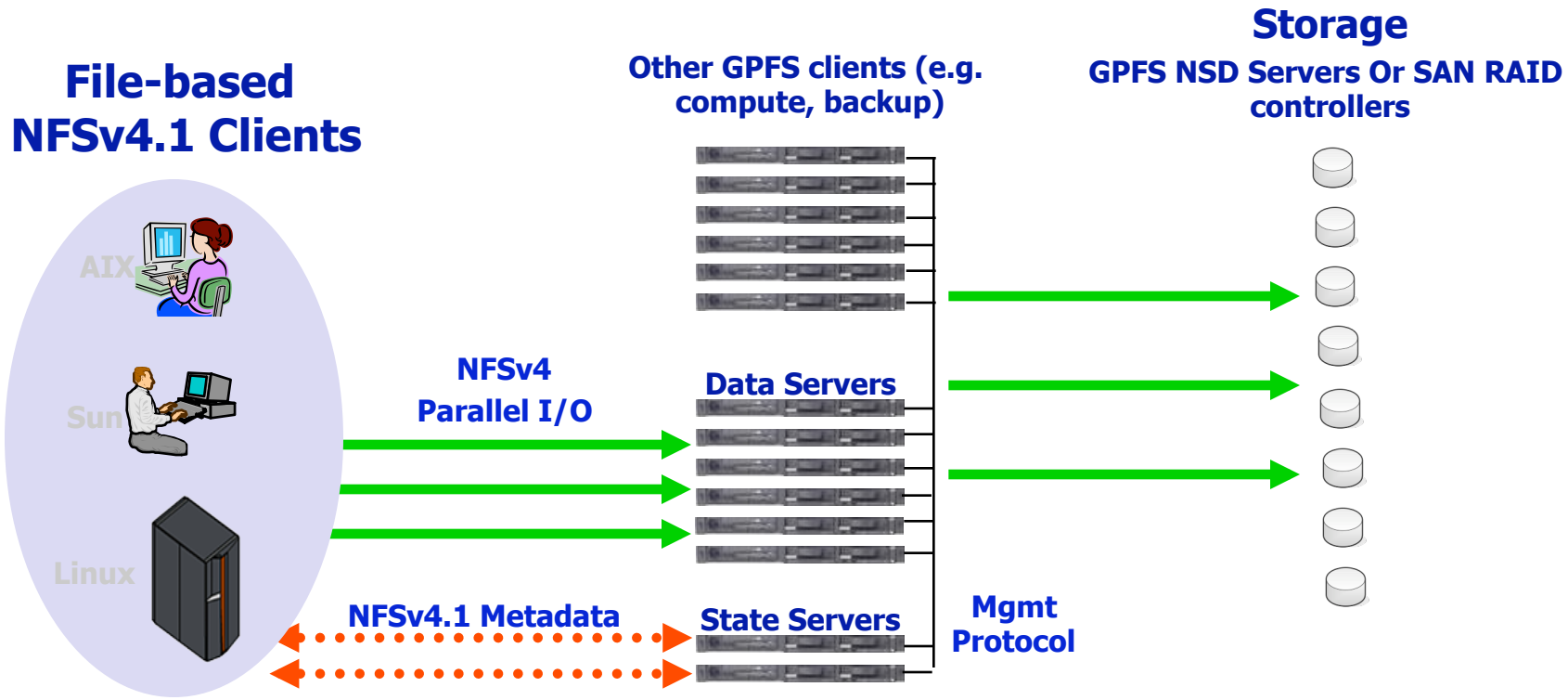
What are we doing?

Linux pNFS server on GPFS

- Participating in IETF standardization efforts
- Funding Linux pNFS work at University of Michigan (CITI)
- Defining open interface API's between pNFS server and generic cluster file system
  - Fully open-source reference implementation on Red Hat GFS2
- Contributing to the implementation of Linux pNFS
  - Client and server common code, file layout driver
  - Basic I/O path (< 1 month of effort)
  - Now supports most pNFS operations
  - CITI now doing performance testing

The goal: A High-quality Linux pNFS server on GPFS

pNFS with GPFS



- **Fully-symmetric GPFS architecture - scalable data *and metadata***
  - pNFS client can mount and retrieve layout from any GPFS node
  - metadata requests can be load balanced across cluster
- **pNFS server and native GPFS clients can share the same file system**
  - Backup, deduplication, and other management functions don't need to be done over NFS
  - pNFS server can be integrated into the compute cluster

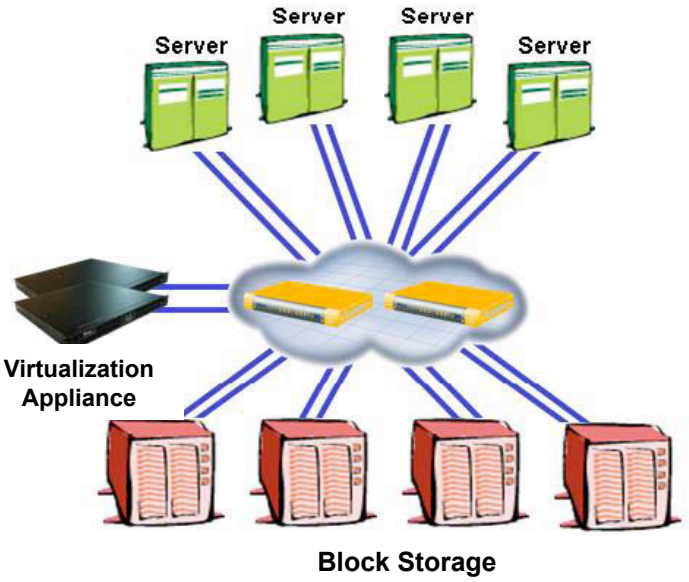
LSI and Block pNFS

Ken Gibson
Engenio Storage Group

Ken.Gibson@lsi.com

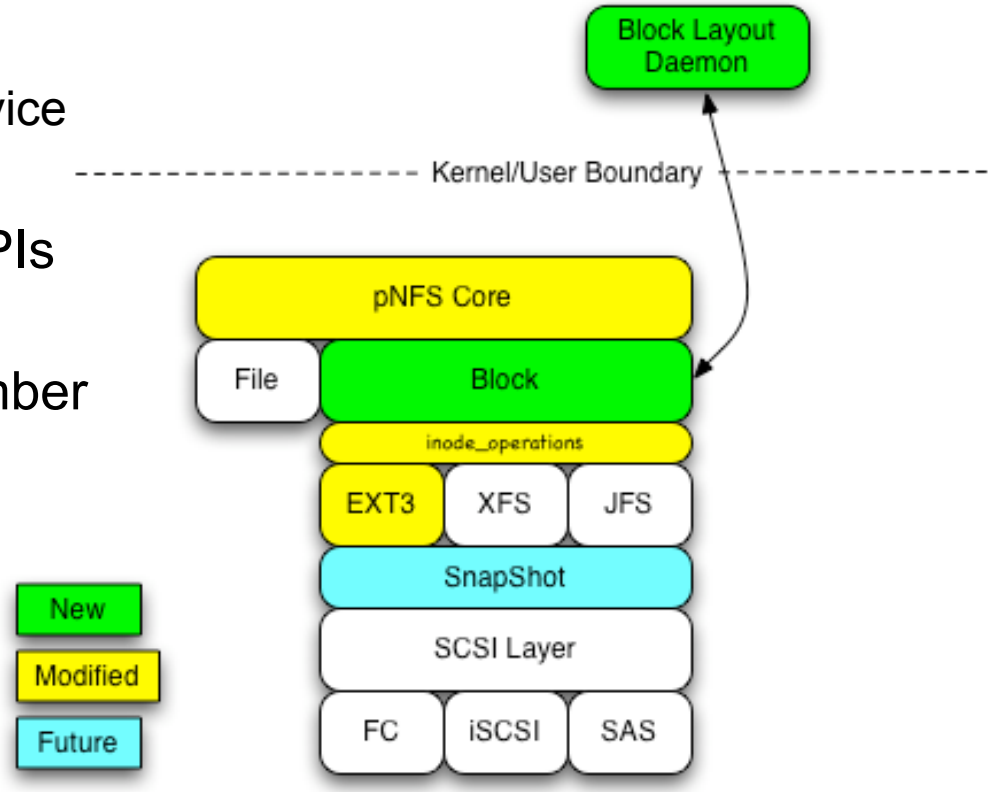
Why Block pNFS?

- Lots of networked block storage in the world
- There will always be a block layer
- Common need to aggregate and virtualize block storage
- LSI and others provide non-standard block virtualization today
- Benefits of standards
- Need for Block, Object and File storage to co-exist in real-world datacenters



LSI pNFS Block Layout Prototype

- Added XDR routines for GETDEVINFO and LAYOUTGET
- New daemon used to gather information for GETDEVINFO
  - Breaks apart LVM volumes
  - Finds partition offsets
  - Locates EFI signature on each device
  - Investigating kernel APIs
- Investigating proposed kernel APIs
- Testing against UM client
- Tested at Bake-a-thon in September



Next Steps

- Validate layout driver and pursue integration in kernel
- Understand failure handling
  - Failed nodes
  - Fencing…
- Explore enhanced data services
  - Snapshots
  - Replication
- Understand co-existence with File and Object MetaData servers


LSI
TM

NetApp™

Go further, faster™

NetApp and pNFS
SC08

Joshua Konkle
Mike Eisler

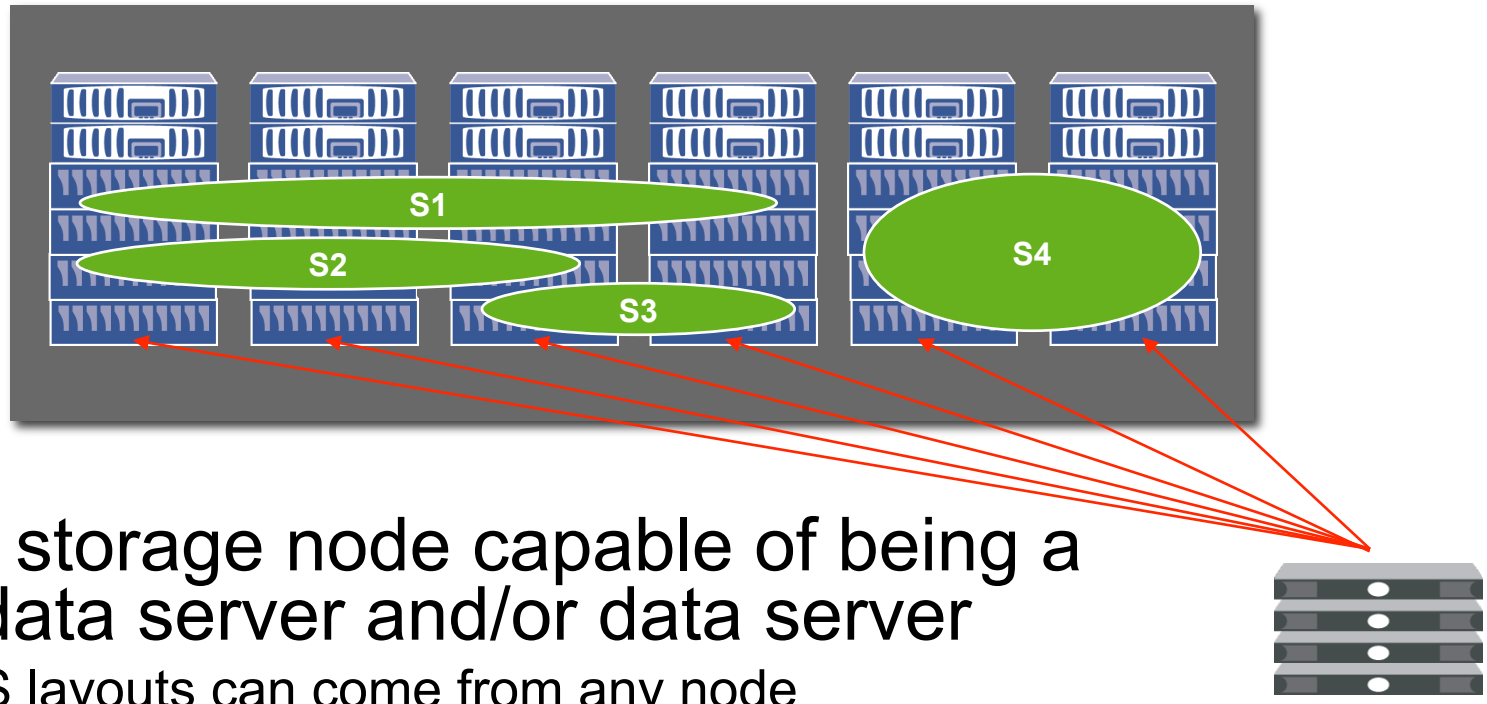
NetApp – Commitment to pNFS

- Data ONTAP GX / Striped WAFL
  - Experience that influenced pNFS specification
- Co-operation with partners and competitors
  - Many NetApp engineers dedicated to standards
    - co-chair, two co-editors, several co-authors
  - Co-developing Linux pNFS client and server with NFS community
  - Co-sponsored Connectathon 2008
    - Brought Linux client and server and Data ONTAP server to Connectathons and Bake-a-Thons

NetApp – Current Status/Adoption

- pNFS server prototype for Data ONTAP
- Leverages existing Data ONTAP GX
  - Storage clustering
  - Striped WAFL
- Striped WAFL addresses pNFS problem statement
  - Data Protection
    - Snapshots, Mirroring, Backup and Recovery
  - Multiprotocol Data Sharing
    - NFSv3, CIFS, pNFS (NFSv4)
- File layouts
  - No need deploy new fabrics
  - It's just NFS over TCP/IP over Ethernet

Data ONTAP, Striped WAFL and pNFS



- Every storage node capable of being a metadata server and/or data server
  - pNFS layouts can come from any node
- Striped WAFL volumes span any/all nodes
  - As a single file system
  - Provides multi-GByte/sec throughput
  - Scales to thousands of TB capacity
  - Online expansion across add-on nodes
  - Management simplicity preserved

NetApp - Summary

- Investing in pNFS eco-system with our partners and competitors
  - standards
  - open source
- NetApp supports scale-out caching today
  - SSD announced; PAM for improved read I/O
- Support pNFS file layout in Data ONTAP prototype
- Unified Storage Architecture product
  - Enterprise NAS & SAN with HPC requirements

NetApp™

Go further, faster™

Accelerating Industry-wide Adoption of Parallel Storage Solutions

panasas®

"The Leader in Parallel Storage"

www.panasas.com

Impetus for a Parallel I/O Standard

- Parallel storage vendors have existing, incompatible parallel products
  - Panasas PanFS
  - IBM GPFS
  - EMC MPFSi (High Road)
  - IBRIX Fusion
- What about open source?
  - Red Hat GFS
  - PVFS
  - Lustre
  - Same compatibility problem combined with robustness concerns

Standards drive adoption, unlock markets and lower costs

Panasas and pNFS

- Co-Led the kick-off workshop in November 2003 that drew representatives from all leading vendors of cluster file systems
  - Thank you Peter Honeyman/CITI for hosting and all their subsequent support for pNFS
- Co-Published initial internet drafts on pNFS
  - Thank you to the nfsv4 working group for being so receptive
- Contributed to Linux open source for iSCSI/OSD
  - Experienced in Linux open source culture for code adoption
- Leading/Coordinating Linux development for pNFS
  - Ushering patches upstream is a full time job
- Panasas storage cluster is pNFS compatible today

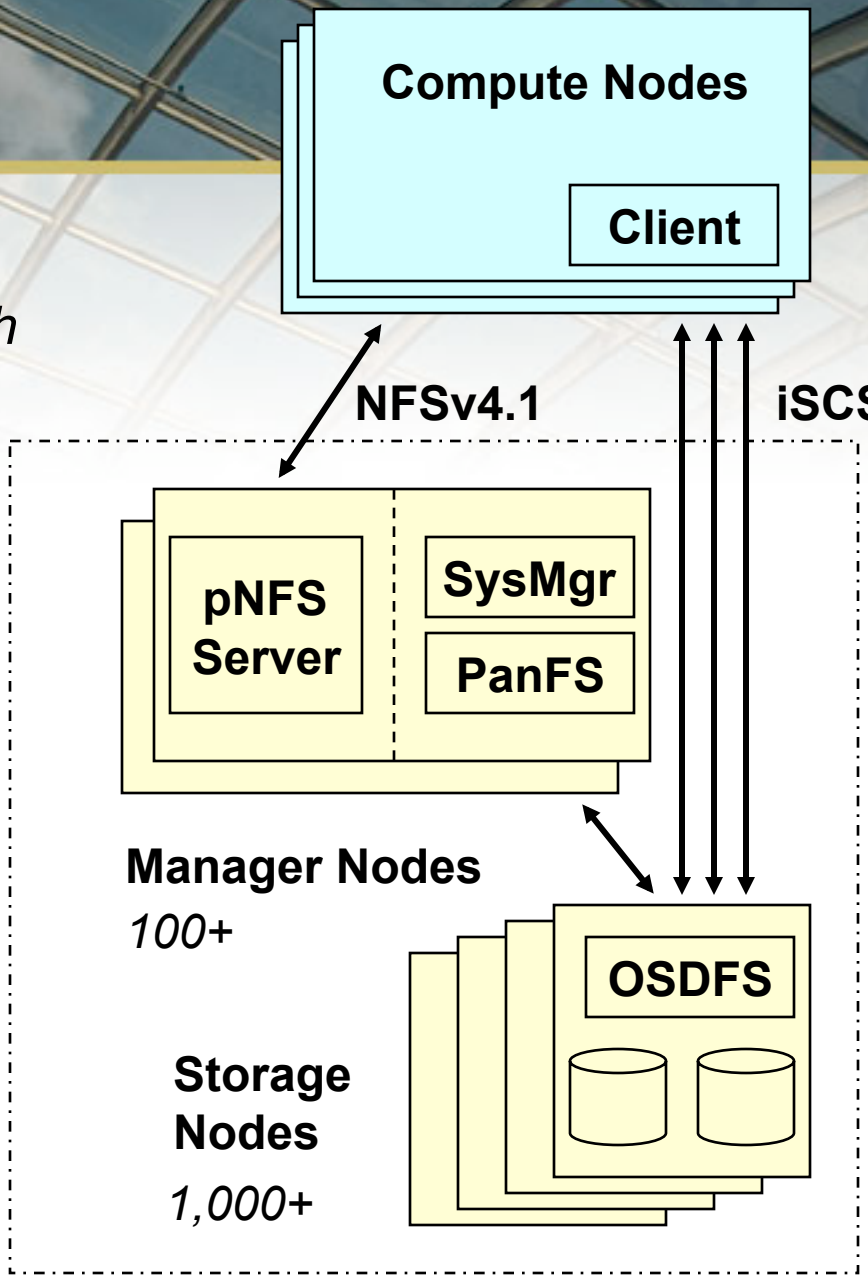
Motivation for Object pNFS

- An Object is like an inode: data + extensible attributes
- Objects have a fine-grained security policy mechanism
  - Metadata servers determine security policy (i.e., file access control decisions)
  - OSD enforce those security policies, all using a strong protocol
  - Support for fencing objects, and fencing clients
  - Supports efficient server-side protocols to set up and enforce access control
- OSD is the latest standard SCSI command set
  - OSDv1 ratified in January 2005, OSDv2 thru letter ballot, being ratified “soon”
  - Designed to be appropriate for implementation on a storage controller
- OSD is the “ideal” building block for clustered storage

- And, of course, Panasas storage clusters use OSD

panasas

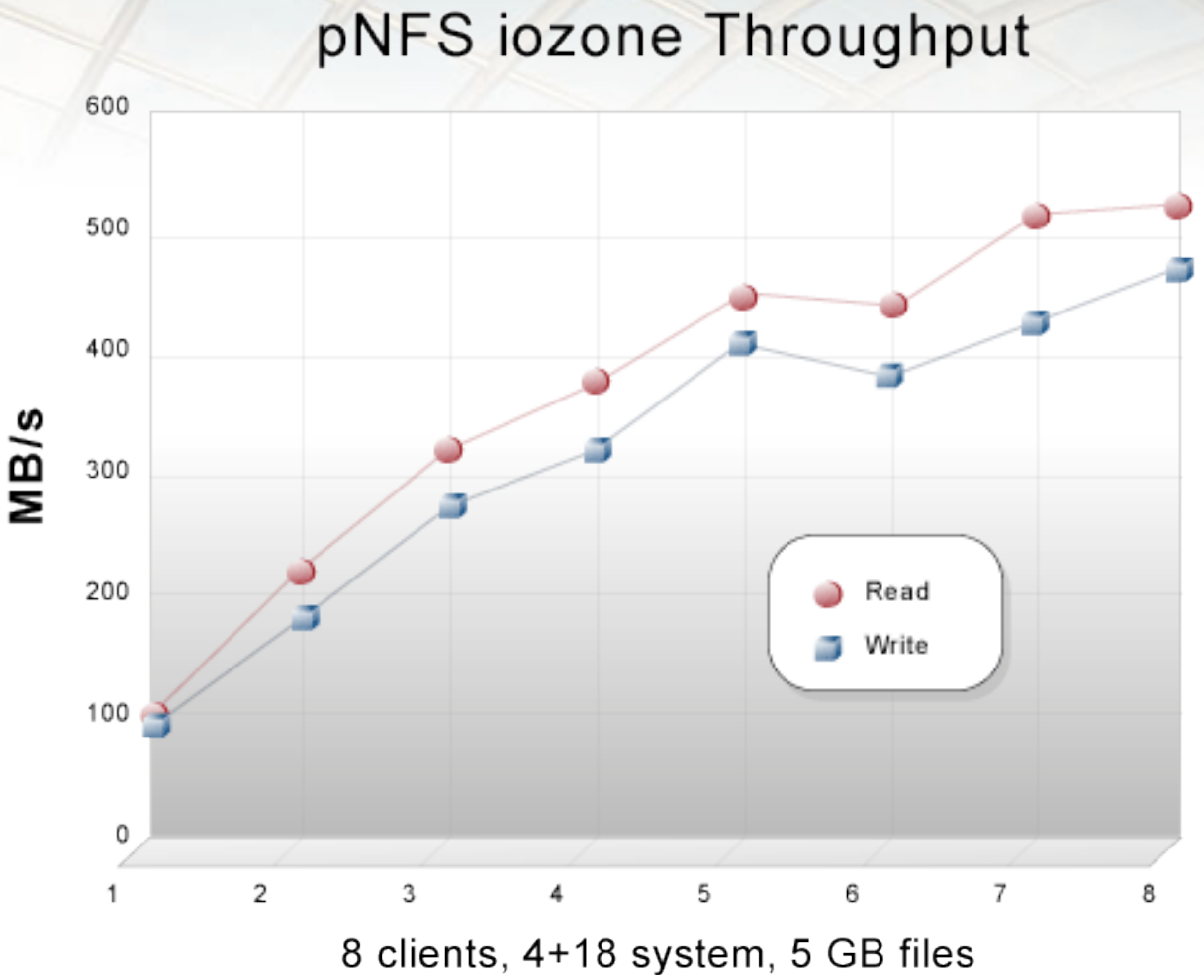
Out of Band architecture with direct, parallel paths from clients to storage nodes



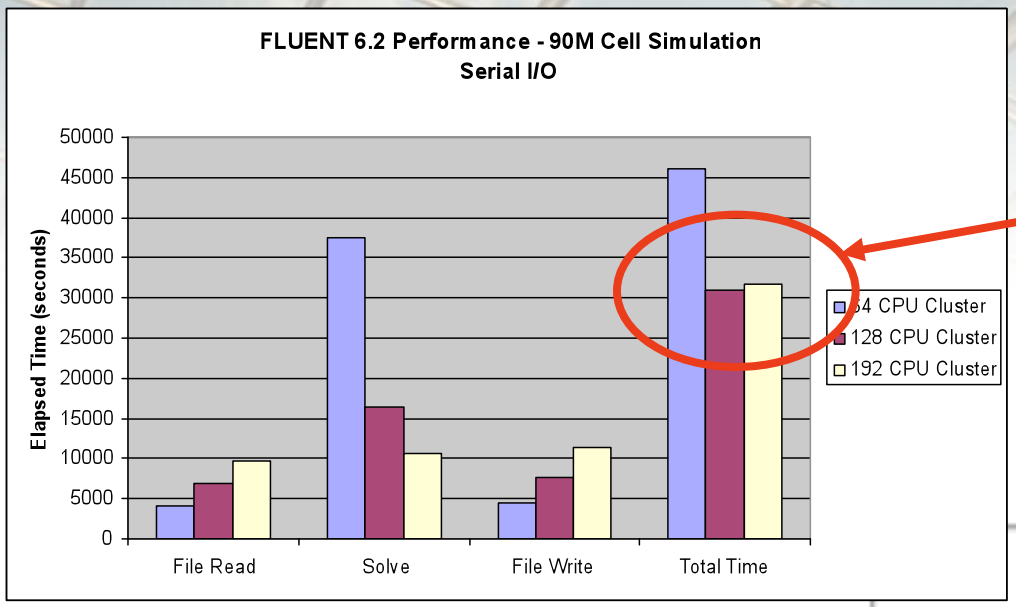
pNFS server is layered on top of the PanFS parallel file system without copying data thru gateways

Internal cluster management makes a large collection of blades work as a single system

Prototype pNFS approaching today's DirectFLOW Performance

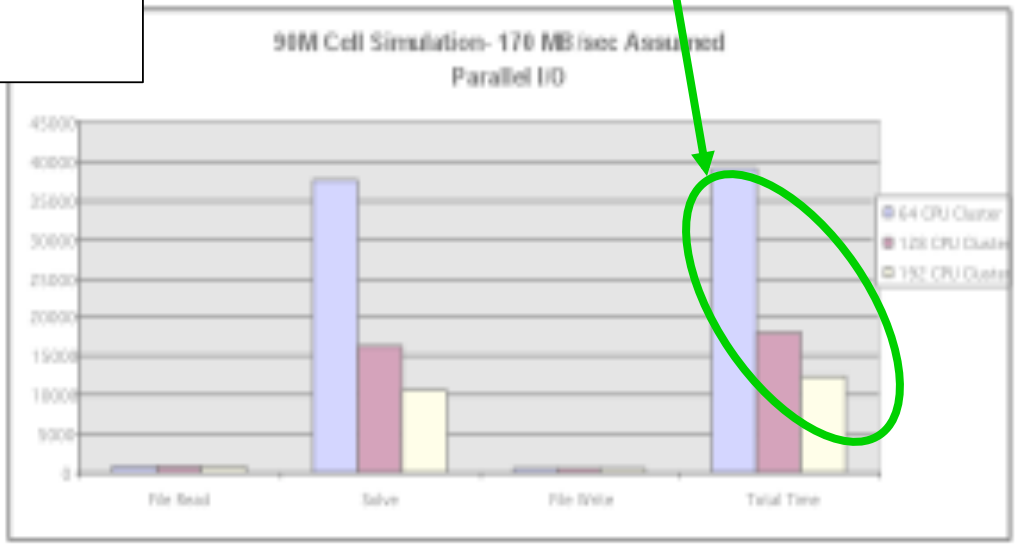


The Advantage of Parallel Storage over NFS:
FLUENT CFD Analysis

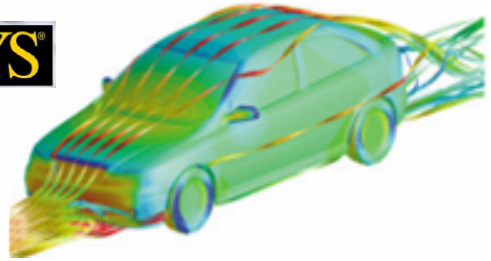


Serial I/O: Increased I/O activity outweighs solver performance improvement

Parallel I/O: Performance scaling maintained



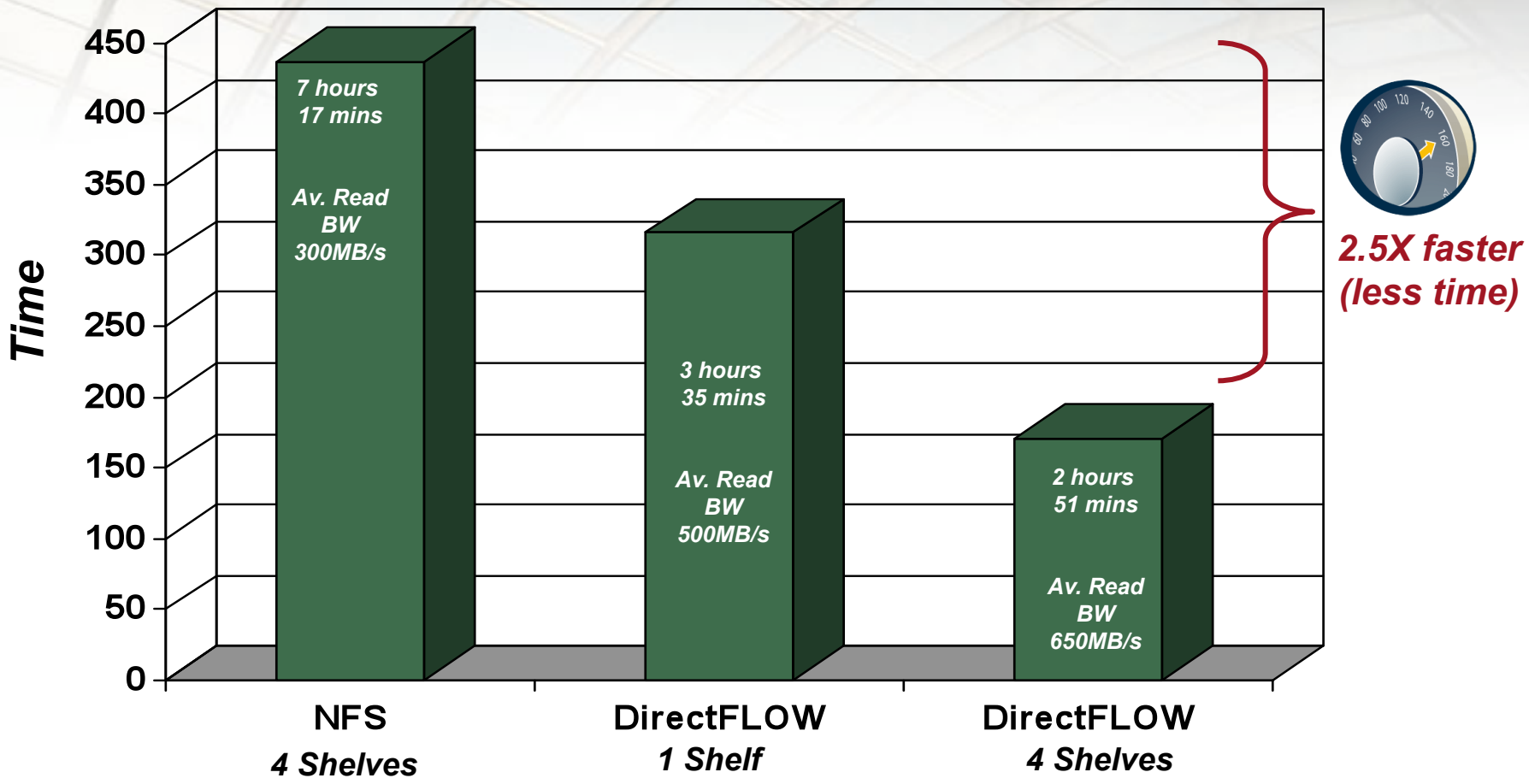
ANSYS



FLUENT

Source: Fluent / ANSYS, November 2006

The Advantage of Parallel Storage over Clustered NFS: *Paradigm GeoDepth Seismic Benchmark*



Source: Paradigm & Panasas, February 2007

pNFS

Bill Baker

Senior Staff Engineer
Sun Microsystems

Open Source Development

<innovate on> opensolaris™

- Developing both pNFS client and server
- Design and development taking place in open community
  - > http://opensolaris.org/os/project/nfsv41/
  - > Binaries as well as source code with design documentation
  - > Source code reviews on *nfsv41-discuss@opensolaris.org*
  - > Live updates – new source and binaries visible within a day
- Early prototype available with instructions

Key Features

- File-based implementation in v1.0
  - Client uses the file interface for I/O with the data servers
- Management via Simple Policy Engine (SPE)
  - Administrative interface on the server to specify policies
  - Examples
    - 2-way striping for files from user A
    - Assign files from user/group C to storage device D
  - Similar interface for specifying policy “hints” on client

Key Features (contd.)

- pNFS over RDMA (on Infiniband)
  - RDMA critical for HPC applications
  - Targeted for initial delivery
  - NFS over RDMA for v3 & v4 available now in opensolaris

Summary and Call to Action

- pNFS is the first open standard for parallel I/O across the network
- pNFS has wide industry support
  - commercial implementations and open source
- Start using NFSv4.0 today
  - Eases transition to pNFS

Urge your O/S (including Linux) distributor and storage vendor to include pNFS