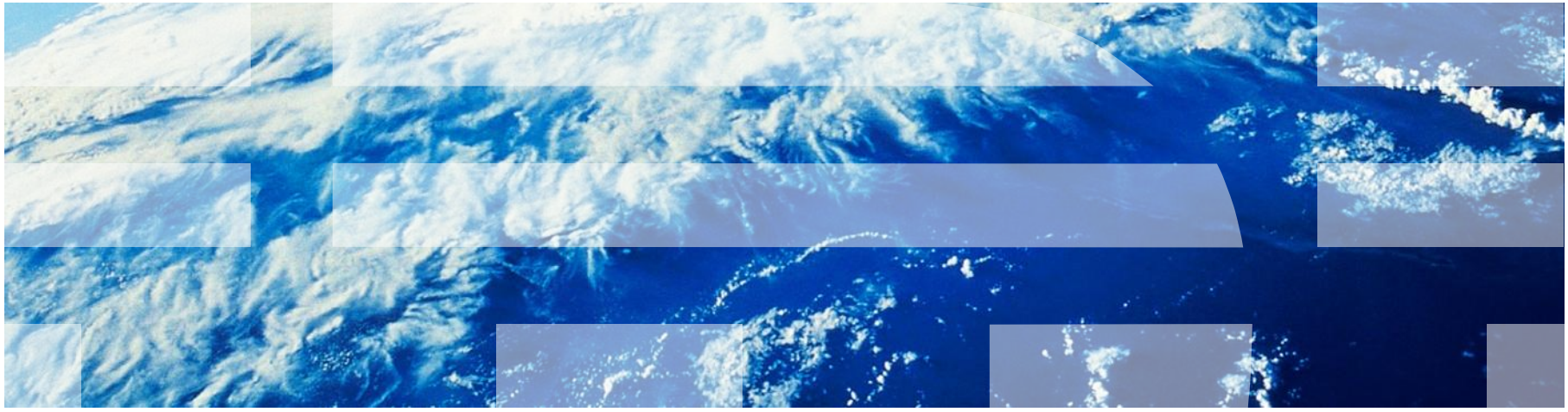
pNFS and GPFS

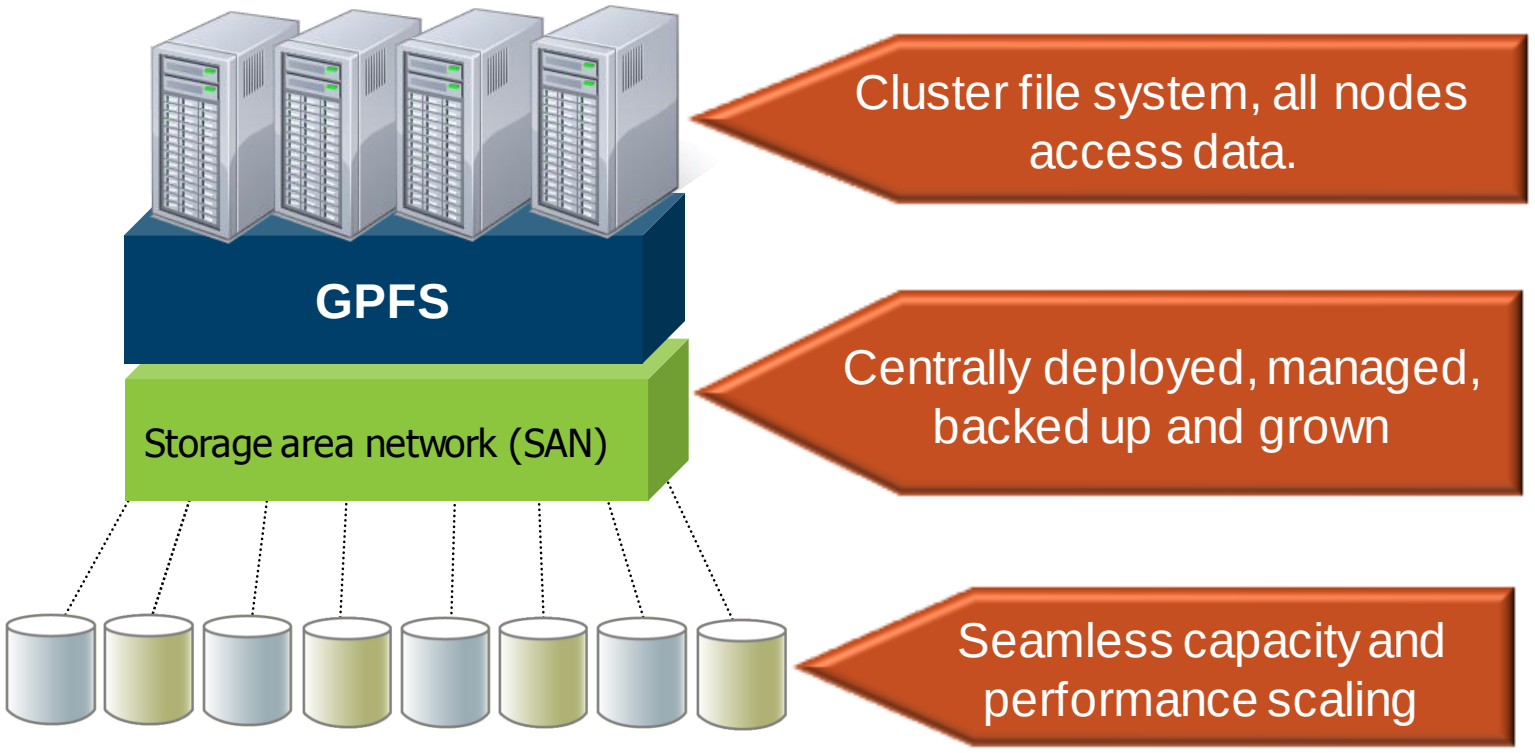
Dean Hildebrand – Storage Systems – IBM Almaden Research Lab



A Bit About Me…

- Research Staff Member at IBM Almaden Research Center for 5 years
  - GPFS, NAS, VMs, HPC
- PhD - University of Michigan, 2007
  - NFS development is a small community
  - At the time, Linux NFS client and server maintainers sat in our research office
  - As NFS has become more prominent over the last 10 years, most of these developers now work for Netapp, EMC, Redhat, etc…
    - Most still based out of Ann Arbor, MI
- Dissertation:
  - "Distributed Access to Parallel File Systems"
  - Goal was to tackle HPC interoperability and software maintenance issues
- pNFS emerged from my dissertation
  - Key part of NFSv4.1
  - Now has large and dedicated development community

What is GPFS?



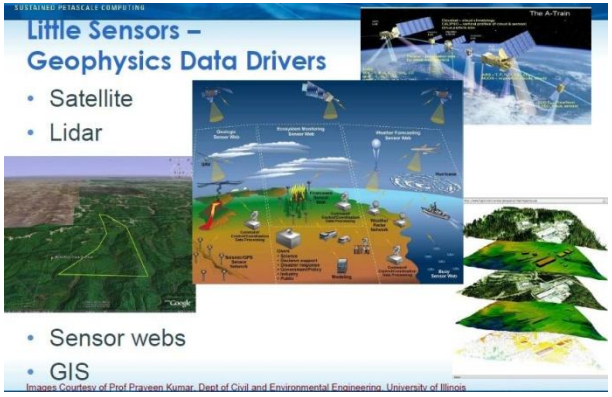
All features are included. All software features: snapshots, replication and multisite connectivity are included in the GPFS license. With no license keys except for client and server to add on, you get all of the features up front.

GPFS Pioneered Big Data Management

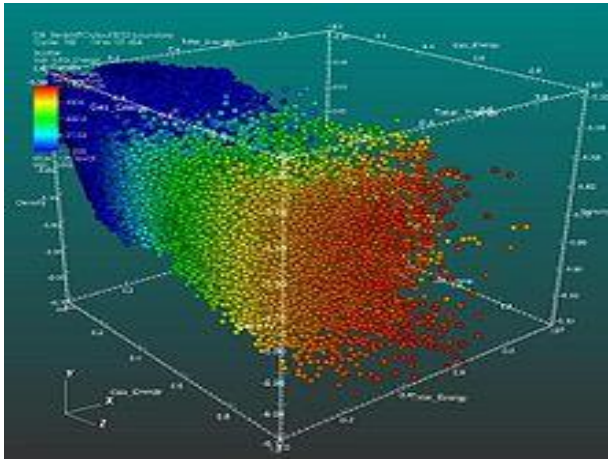
Extreme Scalability	Proven Reliability	Performance
File system 2^{63} files per file system Maximum file system size: 2^{99} bytes Maximum file size 2^{63}-1 bytes Production 5.4 PB file system (1 year ago) **Number of nodes** 1 to 8192	No Special Nodes Add/remove on the fly Nodes Storage Rolling Upgrades Administer from any node	High Performance Metadata Striped Data Equal access to data Integrated Tiered storage

Business Concerns of Unprecedented File System Data Growth

- New and larger digital content types continuously evolving
  - File sizes are growing to TBs
  - Moving from still images to HD video
  - Millions of files to Gazillions of objects
  - More kinds of files and objects
  - Volumes are reaching PBs per day
- Greater collaboration requirements
  - Creations points are many and everywhere
  - Collection points were few but now many and spread globally
- New requirements to meet mission deployment / time to delivery
  - Analysis needs information instantly
  - Access expected anywhere, anytime
  - Analysts no longer only in Washington, are deployed everywhere in the world
- Mandates for longer retention periods and requirements for Archive & Compliance



IEEE Massive File Storage presentation, author: Bill Kramer, NCSA: http://storageconference.org/2010/Presentations/MSST/1.Kramer.pdf:



Source: Wikibon March 2011

Overview

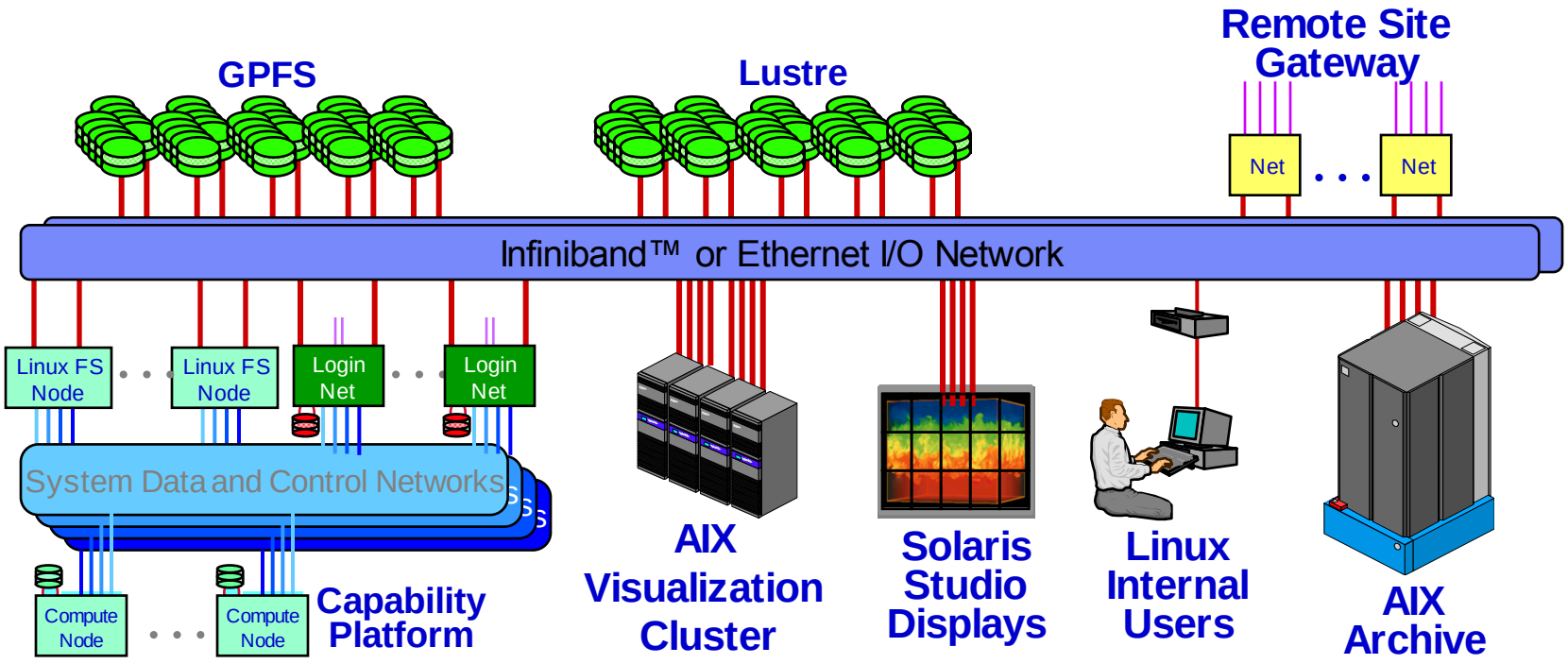
- pNFS/NFSv4.1
- pNFS and GPFS
- pNFS and GPFS-SNC
- pNFS and Panache
- NFSv4.2

Why does IBM Research (and other companies) care about pNFS?

- Solves huge problem of GPFS with NAS hot spots
- Fills Grid, Cloud, and HPC need
  - Standard and open method of accessing GPFS
- Increases number of potential GPFS users
- Increase GPFS install base
  - Single client access to multiple parallel file systems
- GPFS WAN Caching for the Cloud (Panache)
  - High-performance clustered caching solution built on GPFS
  - Use pNFS as a standard way caching data from remote file systems

pNFS Original Motivation: National Laboratories

ASC Platform, Data Storage, and File System Architecture†

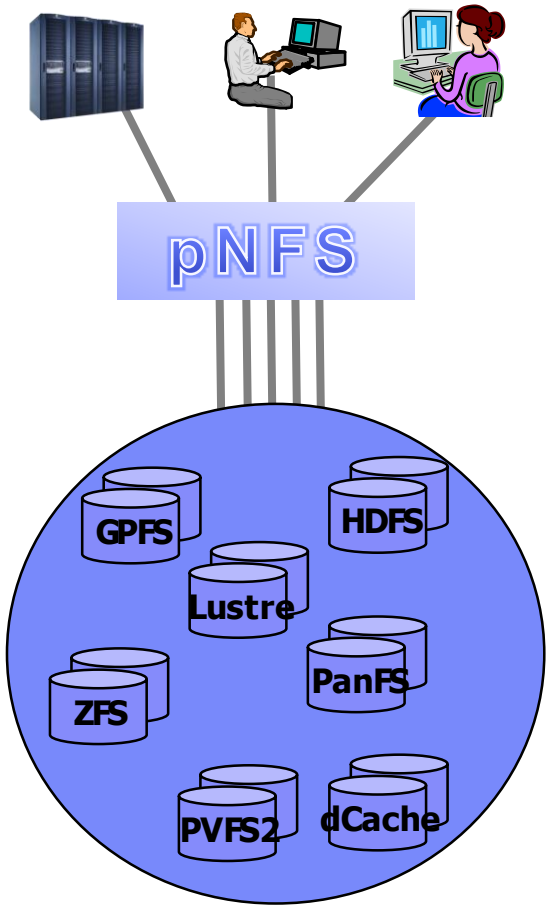


NAS Scalability Limitations

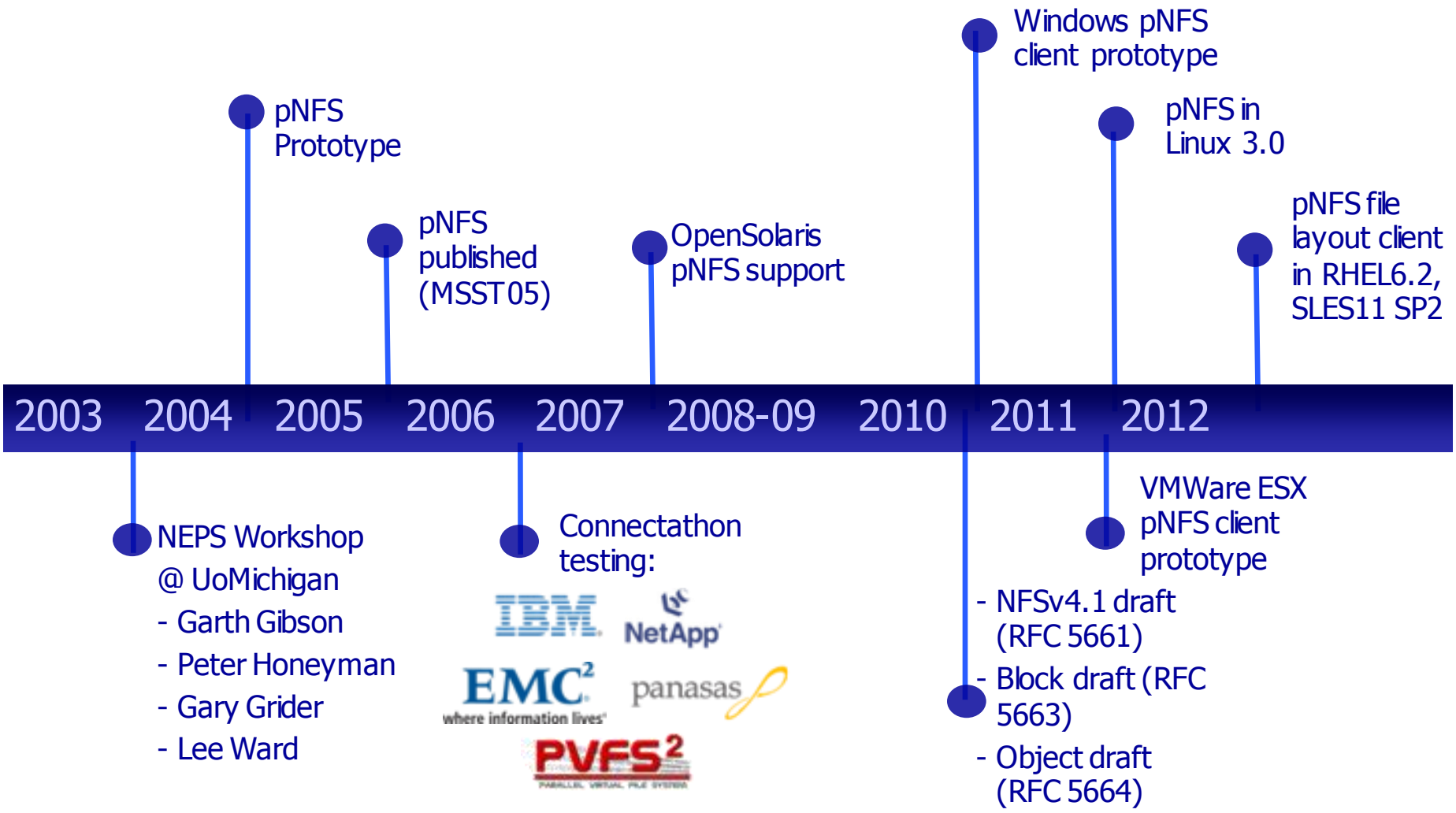
- Current file access protocols are client-server only (1:1)
  - NFSv3, NFSv4, NFSv4.1
  - CIFS/SMB
  - iSCSI
  - FCoE
- Current client-server protocols create data hotspots
  - I/O requests not balanced across entire cluster
- Cannot access distributed/striped data *directly*
  - No understanding of data layout across storage servers
  - All access through a single access point

What is pNFS?

- Parallel NFS
- Scalable access to YOUR data
  - Direct and parallel data access
  - Scale with underlying file system
  - Improve individual and aggregate client performance
  - If NFS can access it, then so can pNFS (but better)
- Standard file access (part of OS)
  - Open client, no client licensing issues
- Key feature in NFSv4.1 (RFC 5661)
  - File layout part of NFSv4.1 draft
  - Object and Block variants in separate Internet Drafts
  - Linux default is now NFSv4
- Security and Access Control
  - Control path uses NFSv4.1 security
  - Data path uses parallel file system security
    - RPCSEC_GSS
    - OSD capabilities

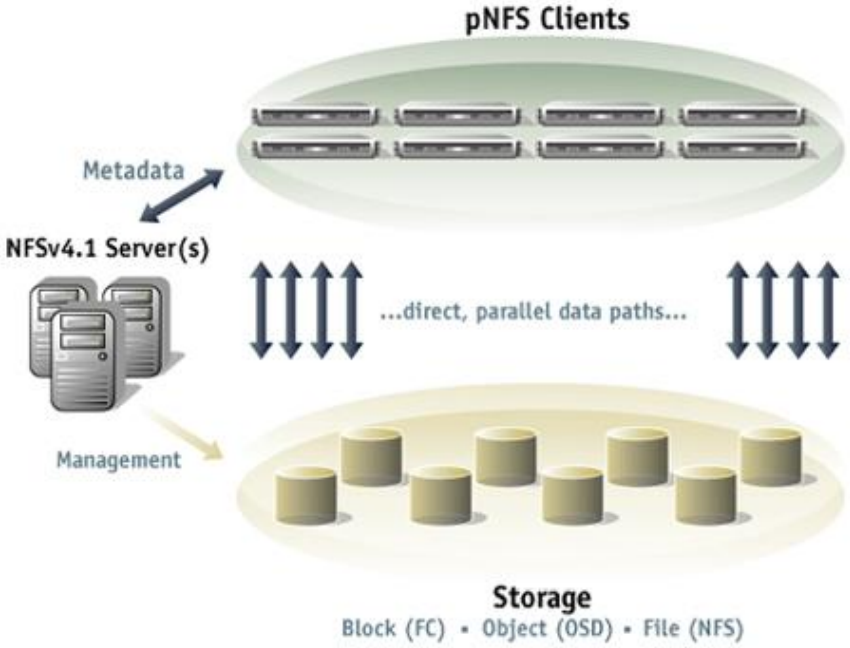


The Long Road of pNFS



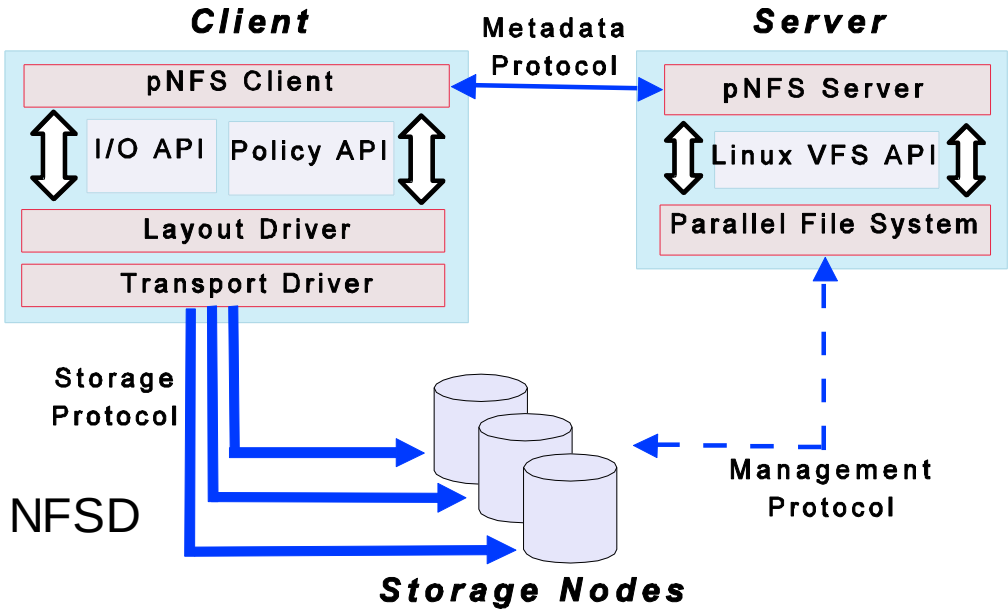
pNFS Basics

- Architecture defines 3 components:
  - clients, data servers, and state servers
- Support for file, object, and block access
  - File layout part of NFSv4.1 draft
  - Object and Block variants in separate IDs
- pNFS operations in IETF NFSv4.1 protocol
  - GETDEVICELIST, GETDEVICEINFO
    - Retrieve data server device information
  - LAYOUTGET
    - Retrieve mapping of byte ranges to data servers
  - LAYOUTRETURN
    - Inform server layout is no longer required for byte range
  - LAYOUTCOMMIT
    - Update metadata to make data visible to other clients
  - CB_LAYOUTRECALL
    - Callback from state server to recall layout from client due to change in mapping, etc
  - CB_NOTIFY_DEVICEID
    - Callback from state server to inform client of device changes/failures



Linux pNFS Architecture

- I/O Interface:
  - Access method
    - Direct
    - Page cache
    - O_DIRECT
- Policy Interface:
  - Blocksize
  - NFS writeback cache
  - Page cache
  - Write/Read threshold
  - Layoutget timing
- Kernel NFS server requires extended NFSD Interface
  - Layout retrieval and update
  - StateID distribution
- Linux kernel NFS server has many many issues
  - Can only add API's for existing kernel file systems
  - Must contact user-space for many things
  - Unstable, poor performance, lack of support, …



Ganesha: User Space NFS Server

- Ganesha History
  - Developed by Philippe Deniel (CEA)
  - Originally designed for local use with Lustre (not for critical data)
- Ganesha Features
  - Efficient caching device for data and metadata
  - Scalable, High Performance
  - Per-namespace (file system) modules
    - Abstraction that allows each file system to perform its own optimizations
    - Also allows for proxy support and other non-traditional back-ends
- Running in User Space makes many things easier.
  - Security Managers (like Kerberos) reside in User Space
    - Can be accessed directly via GSSAPI (no need for “rpc_pipefs”)
  - ID mappers (NIS, LDAP) reside in User Space
    - Accessed directly (the daemon is linked with libnfsidmap)
  - Less constraints for memory allocation than in kernel space
    - Managing huge pieces of memory is easy
  - Developing in User Space is soooooooooooo much easier
    - Hopefully increased community support

Ganesha FSAL: File System Abstraction Layer

- FSAL provides a namespace independent API
  - Translation layer between NFS server and underlying file system
  - Allows file system to customize how files and directories are managed
  - Allows for file system value add
- Focus on supporting NFSv4, NFSv4.1
  - But allows easy implementation of NFSv2/NFSv3
- Handle based API (lookup, readdir, …)
- Implements namespaces' specific authentication mechanisms.
- Many FSAL modules exist
  - GPFS, hpss, proxy, lustre, xfs, zfs, ceph

Ganesha Open Source Collaborations and Status

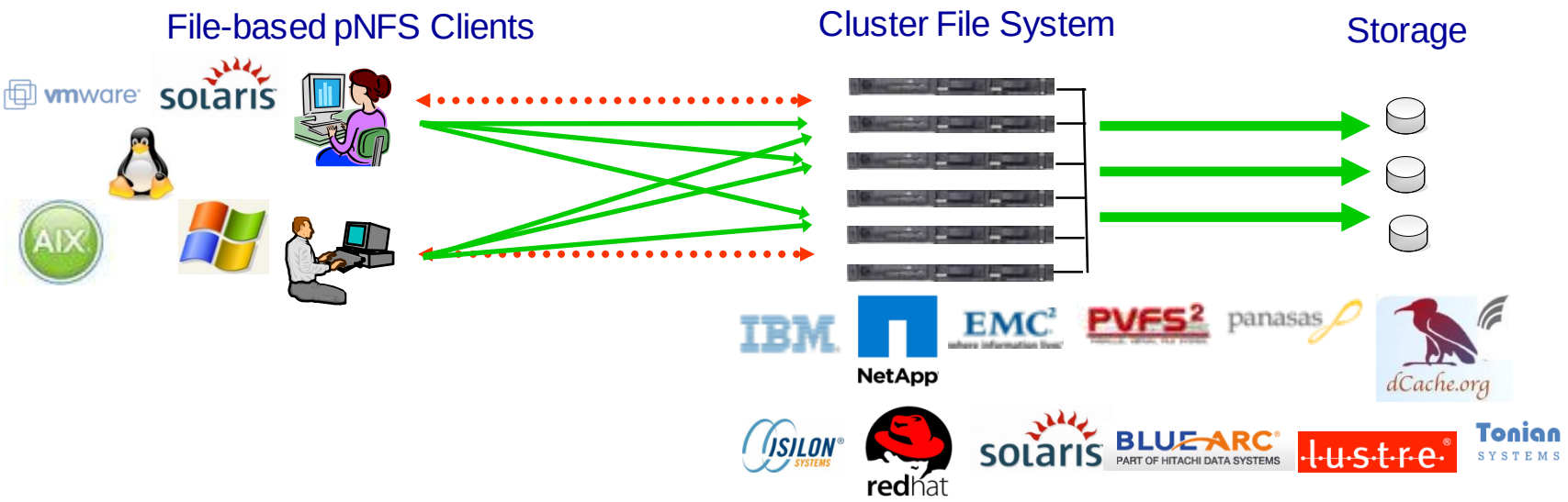
- Since early 2010, several companies started to collaborate on NFS-Ganesha
  - IBM the first company to come in early 2010
  - Research, LTC, SoNAS, XIV, ProtecTIER
  - LinuxBox (Ann Arbor) joined later in 4Q2010
  - Panasas joined the community in 2011
  - DDN 2012
- OpenSource development and collaboration is a great boost for the project.

DESY Laboratory - pNFS with dCache

- Provide high-performance disk-based access to archive data
  - Data archived on tape
- Java-based file servers
  - Written and maintained by DESY
- ATLAS Hammercloud test*
  - 8248 jobs in total
  - Cancelled after 4 days
  - No trouble during test

*Dmitri Ozerov and Yves Kemp, “LHC Data Analysis Using NFSv4.1 (pNFS)”, CHEP’10

Industry Support



pNFS Client	• File layout available in RHEL 6.2 and SLES 11 SP2 • ESX (block) prototype in development • Windows client (file) in development • OpenSolaris (defunct?)
pNFS Server	• Linux Ganesha (user-level) • Linux kernel (very little support, API changes required) • OpenSolaris with ZFS (defunct?) • Many proprietary implementations

Overview

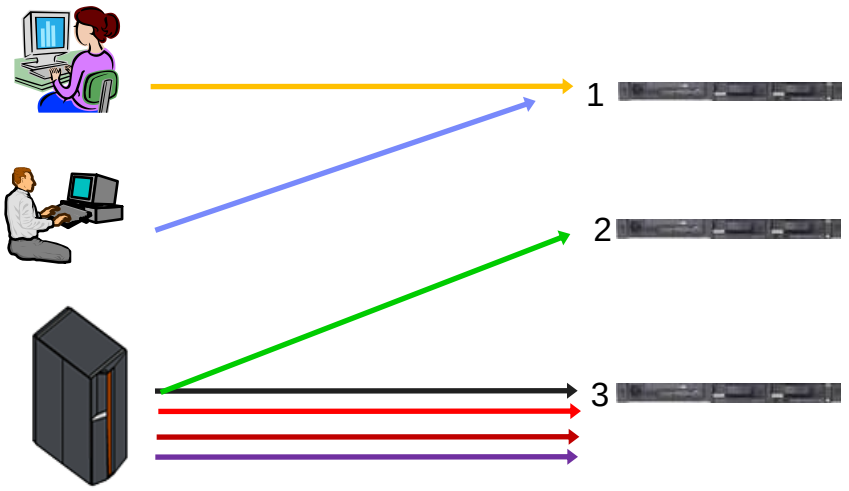
- pNFS/NFSv4.1
- pNFS and GPFS
- pNFS and GPFS-SNC
- pNFS and Panache
- NFSv4.2

NFS and GPFS: Hot Spots

NFSv3/4/4.1 Clients
(non pNFS)

Data Servers

Per-Client Bandwidth
(1 GigE Link)



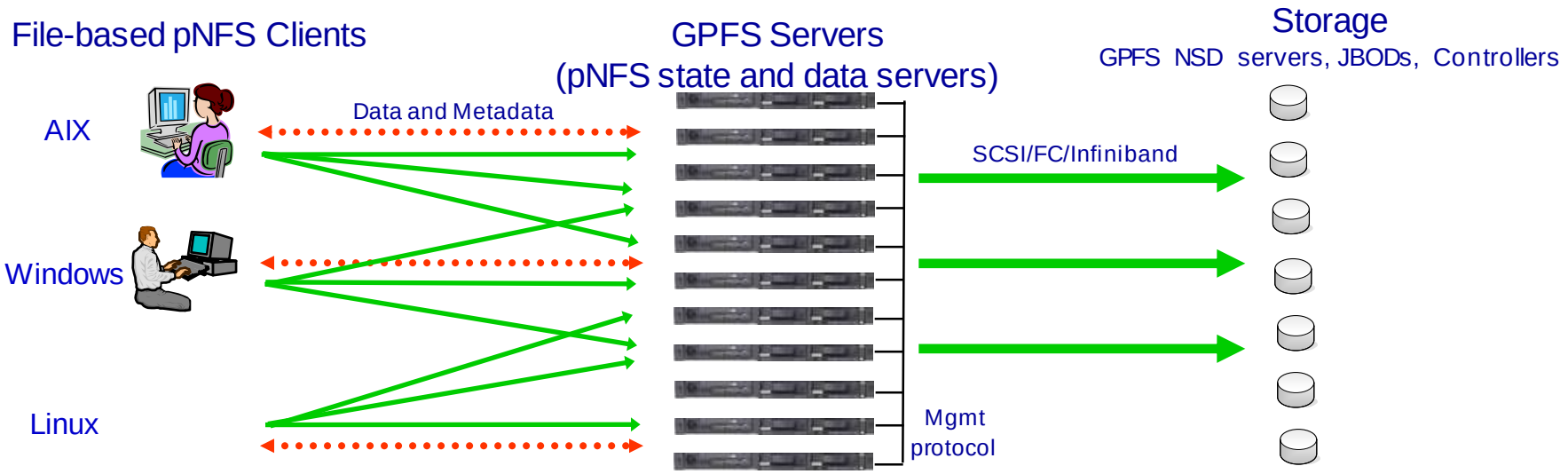
50 MB/s

100 MB/s

25 MB/s

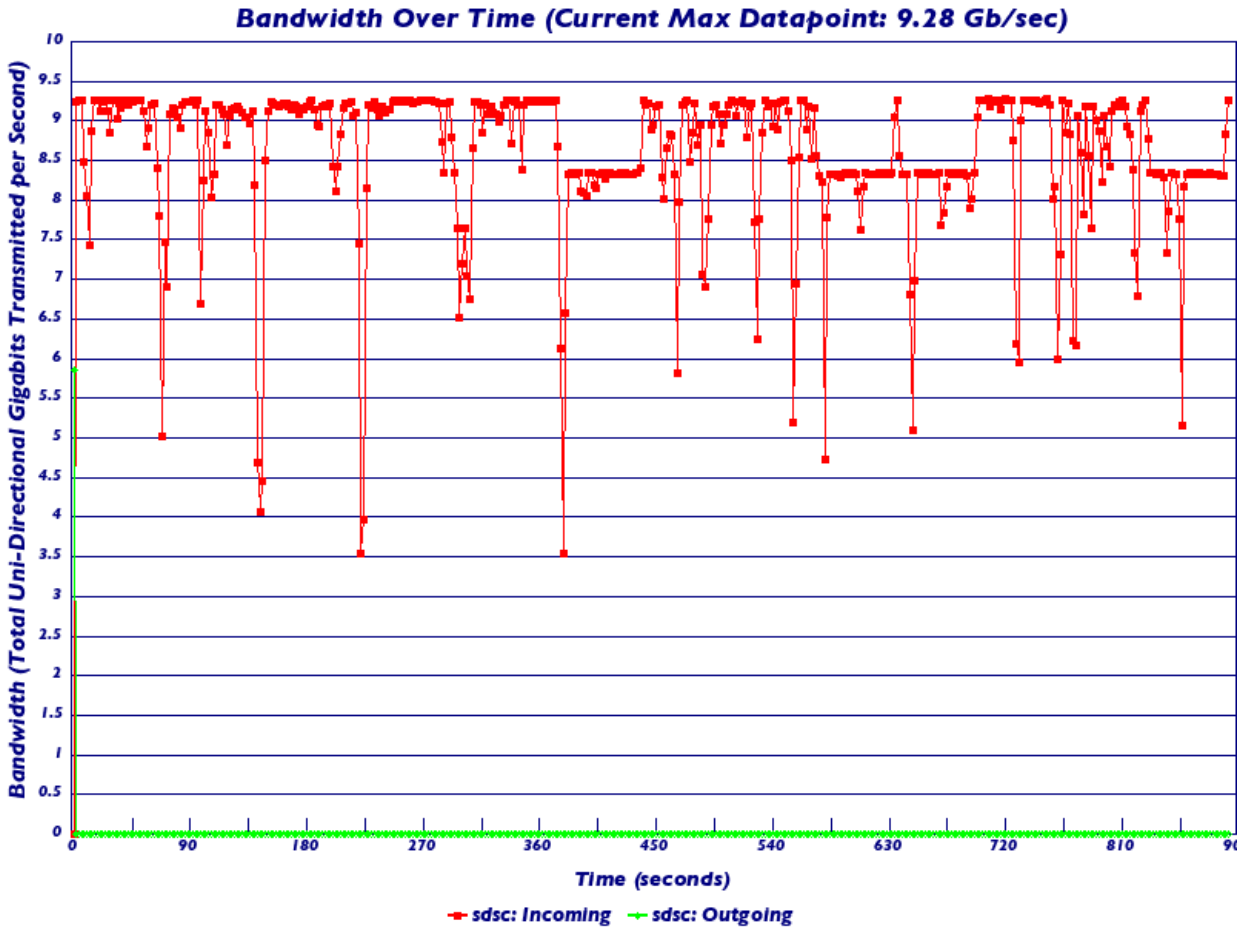
- Gross imbalance between clients
- Clients periodically unmount/mount to new servers when briefly inactive causing future imbalances
- DNS RR does not solve problem
  - Clients can end up accessing different server for lockd

pNFS and GPFS



- Fully-symmetric GPFS architecture - scalable data and metadata
  - pNFS client can mount and retrieve layout from any GPFS node
  - Metadata requests load balanced across cluster
  - Direct data access from any GPFS server
- 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

SC07 pNFS-GPFS Read Performance from Disk



Reno<->San Diego
3 clients and 3 data servers

pNFS is important for GPFS (for unique reasons)

- Competitive offerings NEED pNFS
  - pNFS binds together stand-alone filers into single namespace
- GPFS symmetric architecture can *survive* without pNFS
  - Use "traditional" load balancers like RR DNS
  - Data is striped across nodes anyway
- Use pNFS to unleash the full power GPFS symmetric architecture
  - Data load balancing can be made very flexible not just on layout but on performance
    - Possible to change layouts on the fly as the data can be accessed from any node.

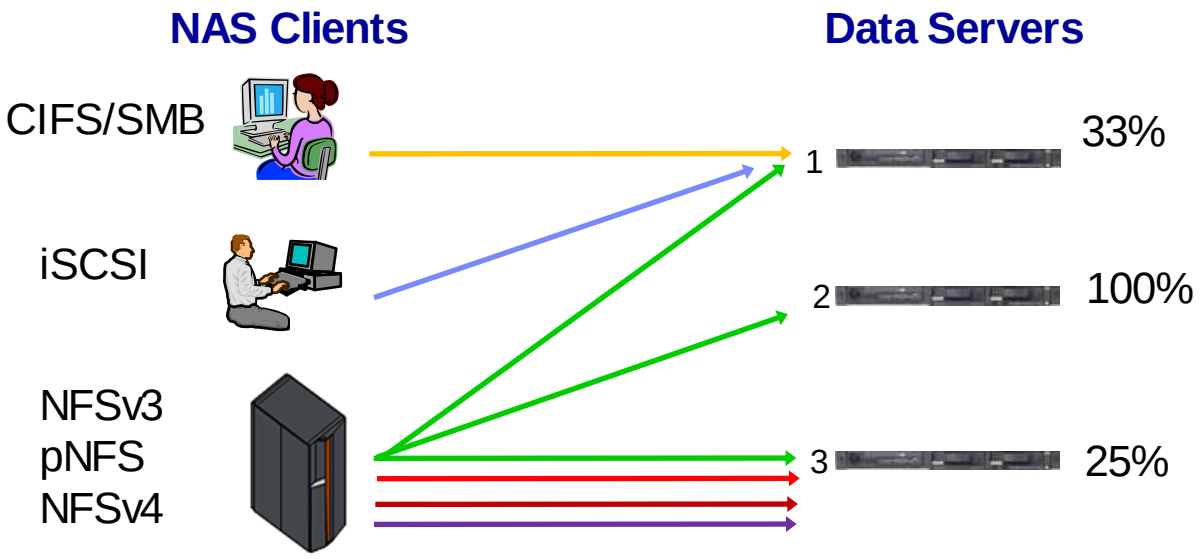
pNFS Layouts for GPFS

- Layout will determine effective load balancing
- Simple layout options
  - Round-robin
  - Random
  - Pseudo-random
  - Read-only, caching data on multiple nodes
- Layout scope
  - Whole file (default)
  - Partial file (byte-range)
    - Linux pNFS client does not support partial file
    - Belief that it is not required (true?)

GPFS and pNFS Limitations

- pNFS not designed to load-balance metadata requests
  - If client-server mapping imbalanced, server will be overloaded
    - GPFS distributed metadata architecture doesn't help
  - Ganesha will allow GPFS to redirect clients to less loaded servers via FS_LOCATIONS
- Heterogeneous protocol access to GPFS cluster
  - Mix of serial and parallel protocols reduce performance

Heterogeneous Protocol Interference

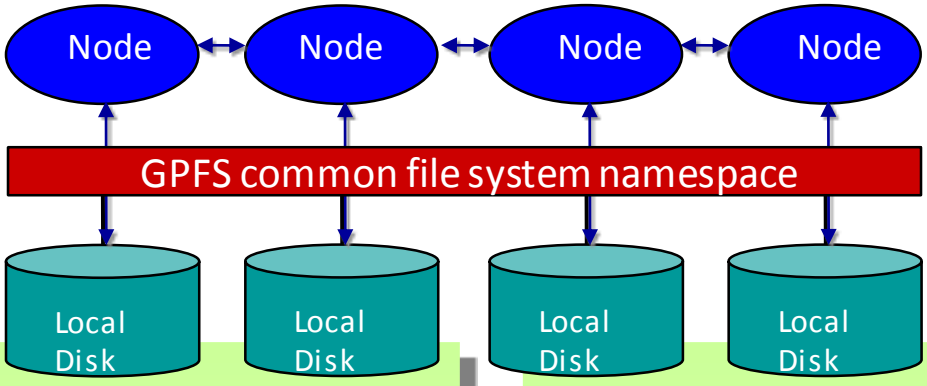


- For round-robin data access, serial clients reduce pNFS performance
- Parallel performance reduced by slowest server
- Options:
  - Separate NFS and pNFS servers
    - GPFS allows specifying which servers to be used for pNFS data servers
  - Complex (partial file) layouts to avoid potential hot spots
    - Currently not supported in Linux

Overview

- pNFS/NFSv4.1
- pNFS and GPFS
- pNFS and GPFS-SNC
- pNFS and Panache
- NFSv4.2

GPFS Shared Nothing Cluster (GPFS-SNC)



GPFS Architecture:

Cluster: thousands of nodes, fast reliable communication, common admin domain.
Shared disk: data and metadata on disk accessible from any node,
Parallel: data *and metadata* flow to/from all nodes from/to all disks in parallel; files striped across all disks.

Extensions for Shared Nothing Architectures:

Locality
Write Affinity
Metablocks
Pipelined replication
Distributed recovery

pNFS and GPFS-SNC

- pNFS layouts can expose GPFS data layout across nodes and their local disks
- pNFS can direct I/O directly to GPFS node containing data
- pNFS is ideal for highly parallel data ingest and dispersal.

Overview

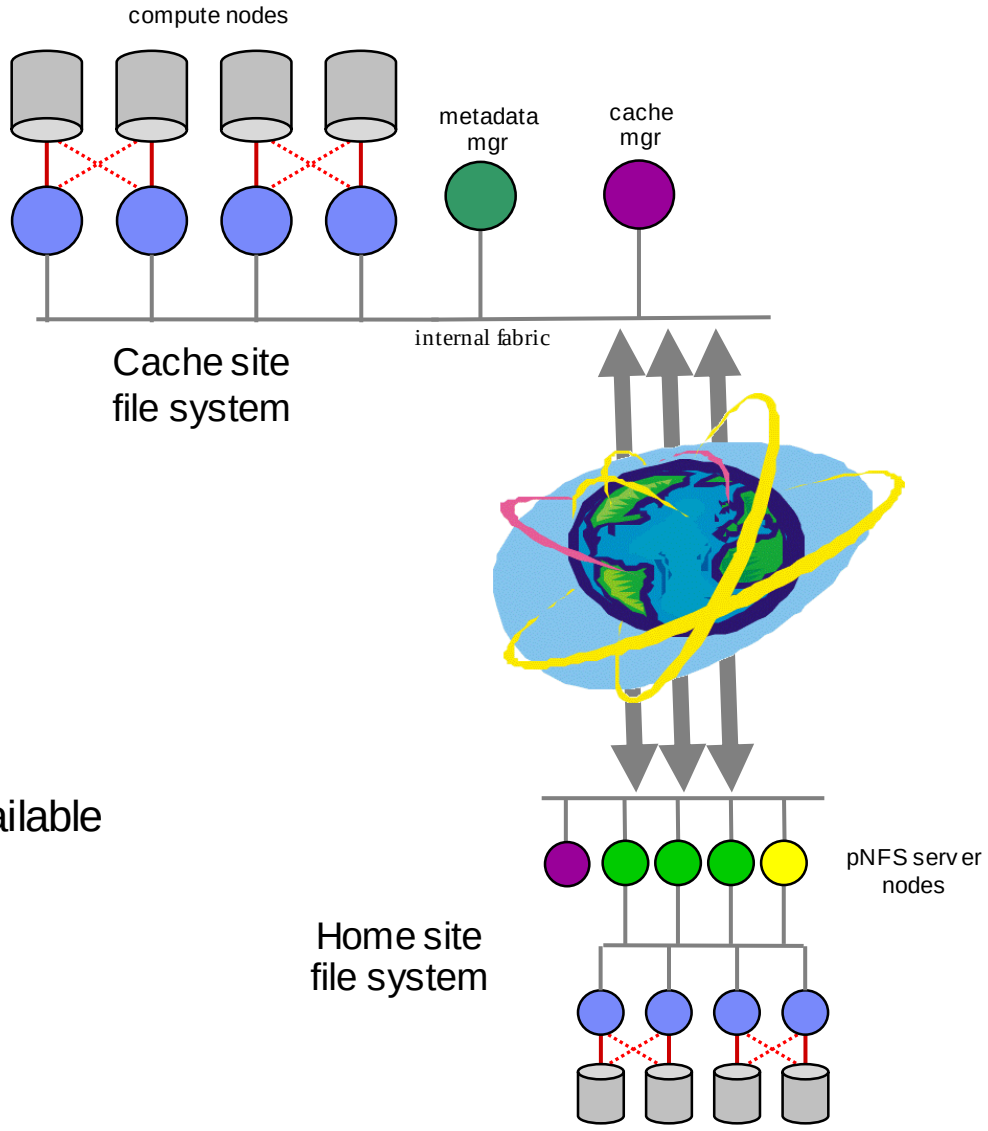
- pNFS/NFSv4.1
- pNFS and GPFS
- pNFS and GPFS-SNC
- pNFS and Panache
- NFSv4.2

Panache – Scalable Data Sharing Within the Cloud

- What is Panache?
  - Policy-driven engine that improves storage efficiency by automatically:
    - Distributing files, images, and application updates to multiple locations
    - Identifying files for backup or replication to a DR location
    - Moving desired files to the right tier of storage including tape in a TSM hierarchy
    - Deleting expired or unwanted files
  - High-performance
    - Scan billions of files in minutes with scale out nodes
- What client value does Panache deliver?
  - Enables ubiquitous access to files from across the globe
  - Reduces networks costs and improves application performance
    - Data becomes local to the user
  - Improves data protection by identifying candidates for backup or DR
  - Lowers storage cost by moving files transparently to most appropriate tier of storage
  - Controls storage growth by transparently moving older files to tape and deleting unwanted or expired files
  - Enhances administrator productivity by automating file management

Panache – Scalable Data Sharing Within the Cloud

- Functions
  - Global WAN Caching
    - Remove latency effects
    - Disconnected operation
  - Global common Namespace
  - Disaster Recovery
  - File virtualization
  - Legacy NAS migration
- Features
  - Parallel access at each level
  - On demand read from remote site
  - Local write performance
  - Full data access when network unavailable
- SONAS/V7K Active Cloud Engine
- GPFS Active File Management



Overview

- pNFS/NFSv4.1
- pNFS and GPFS
- pNFS and GPFS-SNC
- pNFS and Panache
- NFSv4.2

NFSv4.2 - Why you will continue to care about pNFS (and NFS)

- Very virtual machine friendly
- New Features
  - Server side copy and clones
  - Sparse file support
  - Client directed optimizations for server caching and prefetching
  - Space reservations (Thin Provisioning)
  - Hole punching
- Other NFS advancements
  - Federated file systems
  - RDMA
  - Linux WAN performance

More Information

- NFSv4.1 RFC
  - http://www.ietf.org/rfc/rfc5661.txt
- NFSv4.2 Draft Proposal
  - http://tools.ietf.org/html/draft-ietf-nfsv4-minorversion2-13
- Linux information
  - http://www.citi.umich.edu/projects/asci/pnfs/linux
- IBM
  - http://www.almaden.ibm.com/storagesystems/projects/pnfs
- Panasas
  - www.pnfs.com
- Ganesha
  - http://nfs-ganesha.sourceforge.net
  - Git access available under the terms of the LGPLv3 license