Scalable Performance of the Panasas Parallel File System

Brent Welch
Director of Software Architecture
Panasas, Inc.

NSC 08

Go Faster. Go Parallel.

Scalable Performance of the Panasas Parallel File System

Brent Welch, Marc Unangst, Zainul Abbasi,
Garth Gibson*, Brian Mueller,
Jason Small, Jim Zelenka, Bin Zhou
Panasas Inc
* Carnegie Mellon and Panasas Inc

USENIX FAST 08 Conference

Go Faster. Go Parallel.

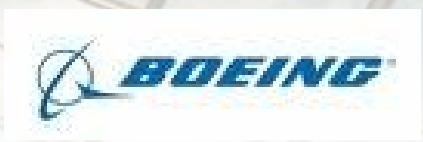
Outline

- Panasas Background, Hardware and Software
- Per-File, Client Driven RAID
- Declustering and Scalable Rebuild
- Metadata management and performance

Panasas Company Overview

Founded	1999 By Prof. Garth Gibson, Co-Inventor of RAID
Technology	**Parallel File System and Parallel Storage Appliance**
Locations	US: HQ in Fremont, CA, USA R&D centers in Pittsburgh & Minneapolis EMEA: UK, DE, FR, IT, ES, BE, Russia APAC: China, Japan, Korea, India, Australia
Customers	ECS October 2003, deployed at 200+ customers Energy Government Manufacturi ISVs: Resellers: Academia Life Sciences F
Primary Investors	

Accelerating Enterprise Parallel Storage Adoption



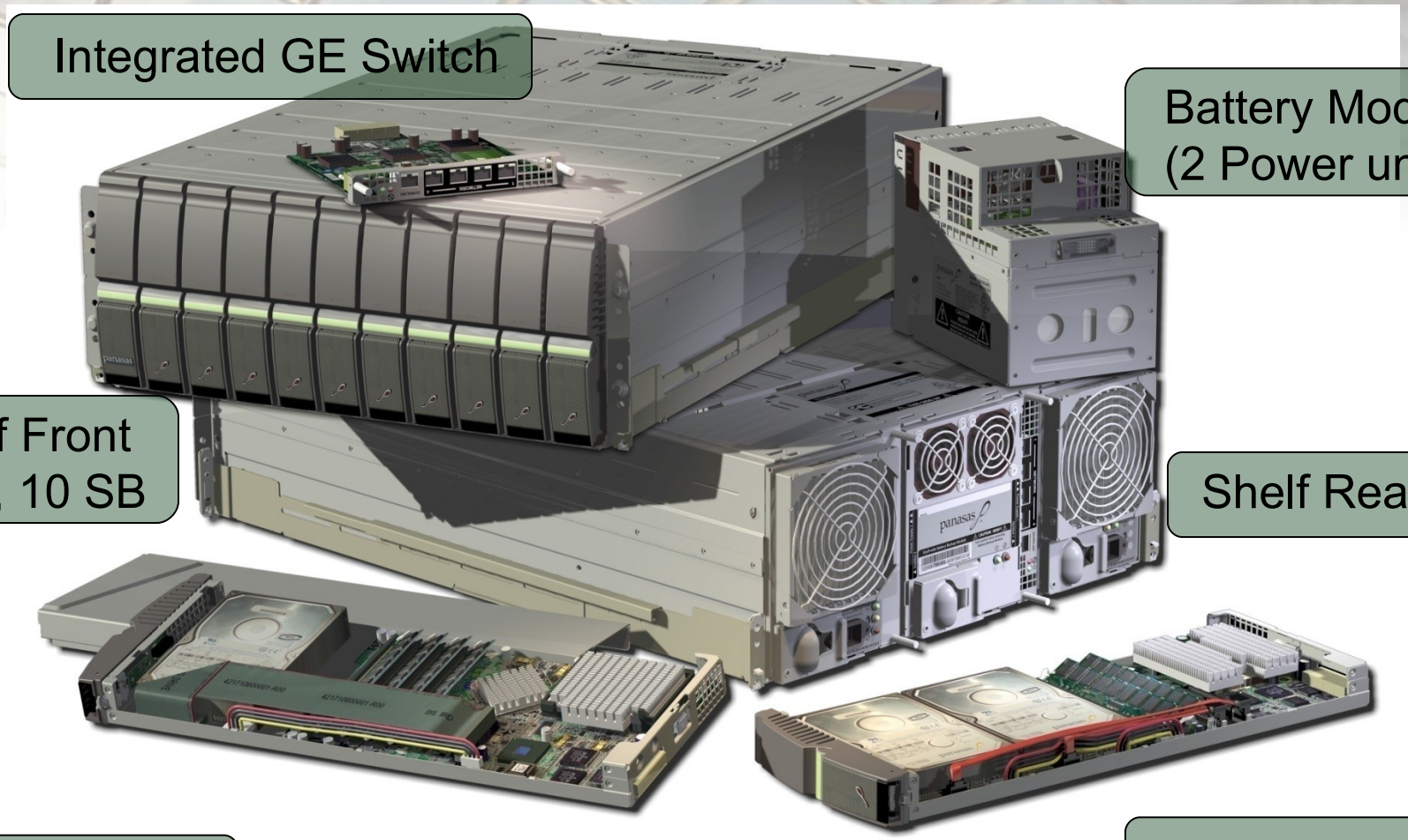
Panasas Architecture

- Cluster technology provides scalable capacity and performance: capacity scales symmetrically with processor, caching, and network bandwidth

Disk
CPU
Memory
Network

- Scalable performance with commodity parts provides excellent price/performance
- Object-based storage provides additional scalability and security advantages over block-based SAN file systems
- Automatic management of storage resources to balance load across the cluster
- Shared file system (POSIX) with the advantages of NAS, with direct-to-storage performance advantages of DAS and SAN

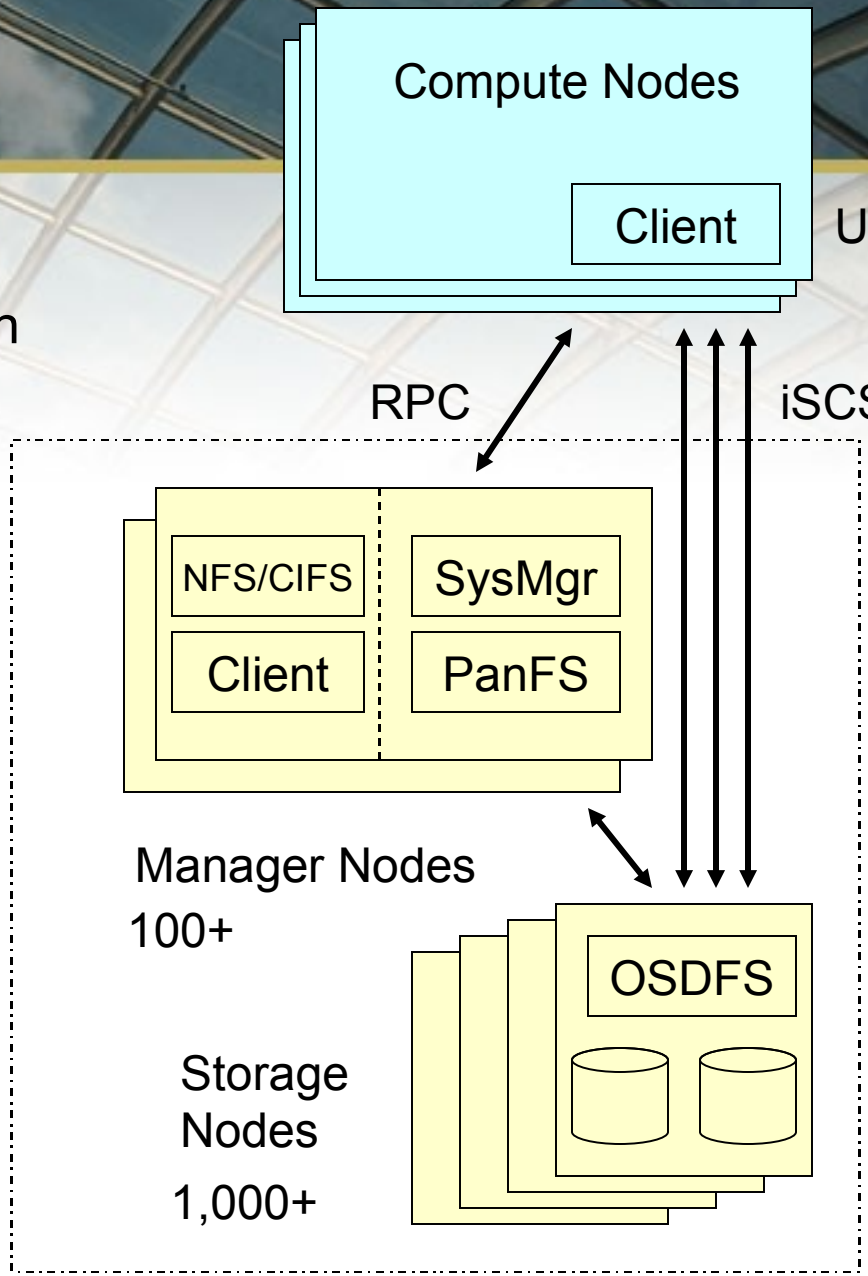
Panasas Blade Hardware



Panasas Product Advantages

- Proven implementation with appliance-like ease of use/deployment
  - Running mission-critical workloads at global F500 companies
- Scalable performance with Object-based RAID
  - No degradation as the storage system scales in size
  - Unmatched RAID rebuild rates – parallel reconstruction
- Unique data integrity features
  - Vertical parity on drives to mitigate media errors and silent corruptions
  - Per-file RAID provides scalable rebuild and per-file fault isolation
  - Network verified parity for end-to-end data verification at the client
- Scalable system size with integrated cluster management
  - Storage clusters scaling to 1000+ storage nodes, 100+ metadata managers
  - Simultaneous access from over 12000 servers

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



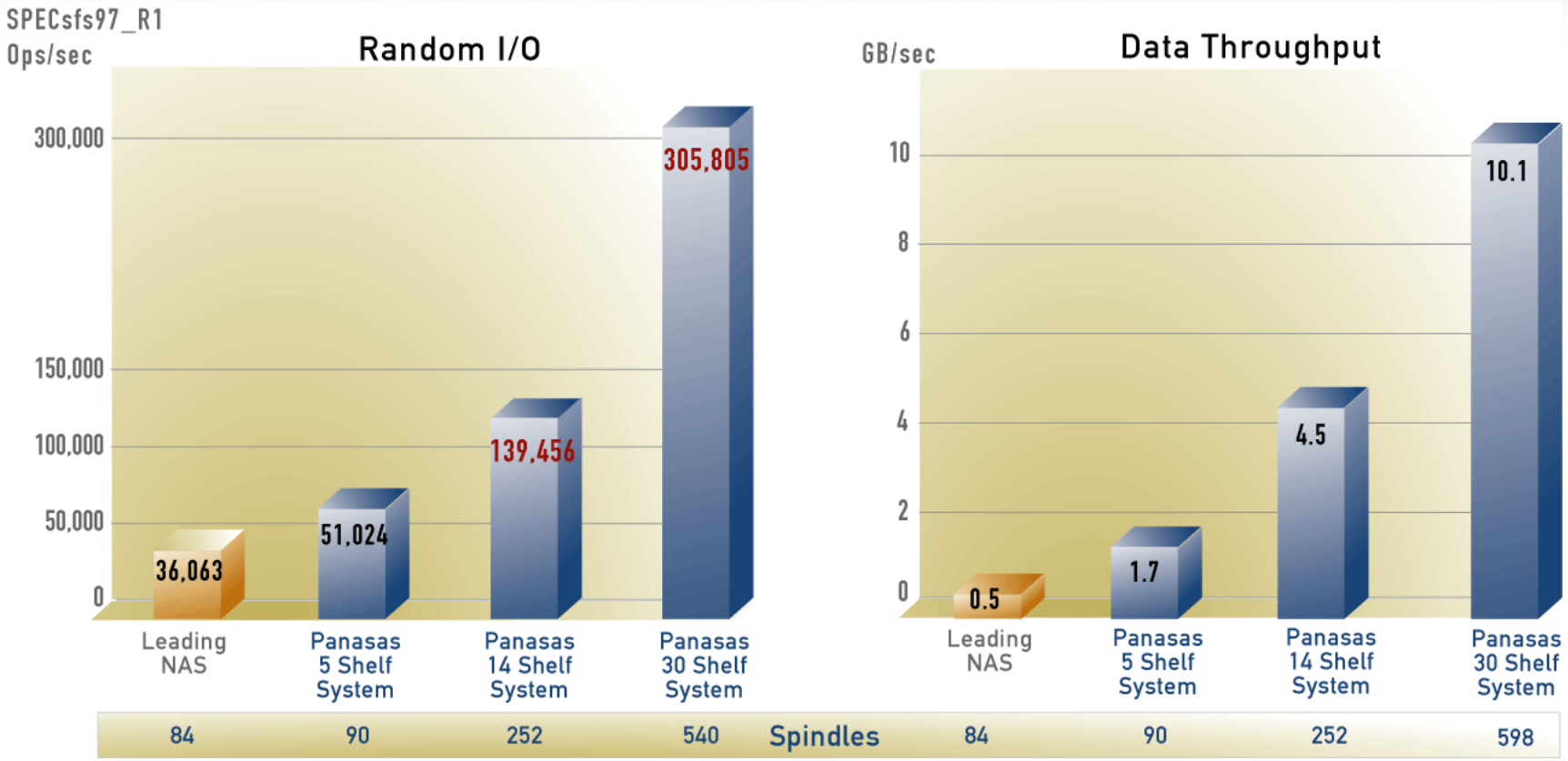
Internal cluster management makes a large collection of blades work as a single system

Proven Panasas Scalability

- Storage Cluster Sizes Today (e.g.)
  - **Boeing**, 50 DirectorBlades, 500 StorageBlades in one system. (plus 25 DirectorBlades and 250 StorageBlades each in two other smaller systems.)
  - **LANL** RoadRunner. 100 DirectorBlades, 1000 StorageBlades in one system today, planning to increase to 144 shelves next year.
  - **Intel** has 5,000 active DF clients against 10-shelf systems, with even more clients mounting DirectorBlades via NFS. They have qualified a 12,000 client version of 2.3, and will deploy “lots” of compute nodes against 3.2 later this year.
  - **BP** uses 200 StorageBlade storage pools as their building block
  - **LLNL**, two realms, each 60 DirectorBlades (NFS) and 160 StorageBlades
  - Most customers run systems in the 100 to 200 blade size range

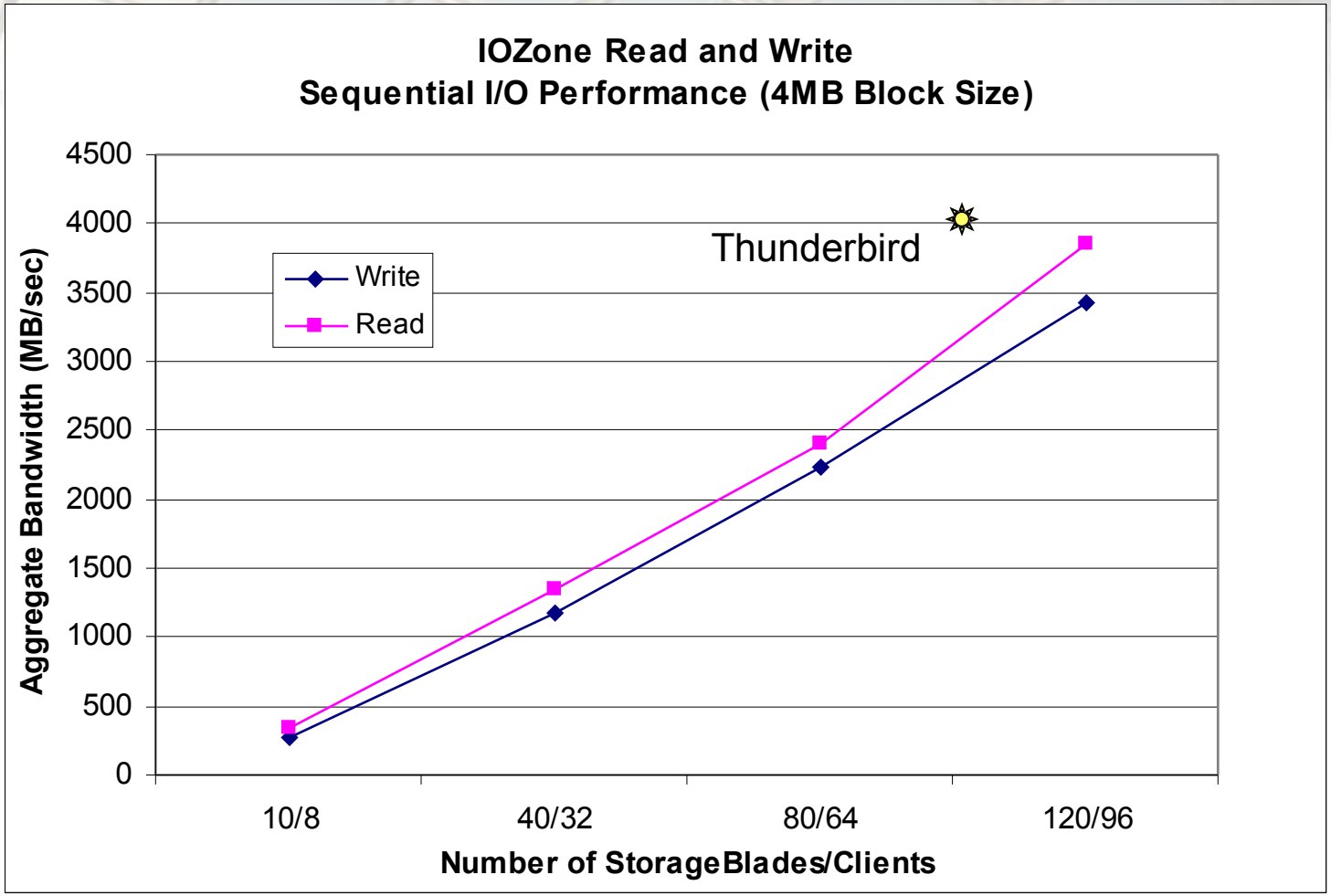
Linear Performance Scaling

- Breakthrough data throughput AND random I/O
  - Performance and scalability for all workloads

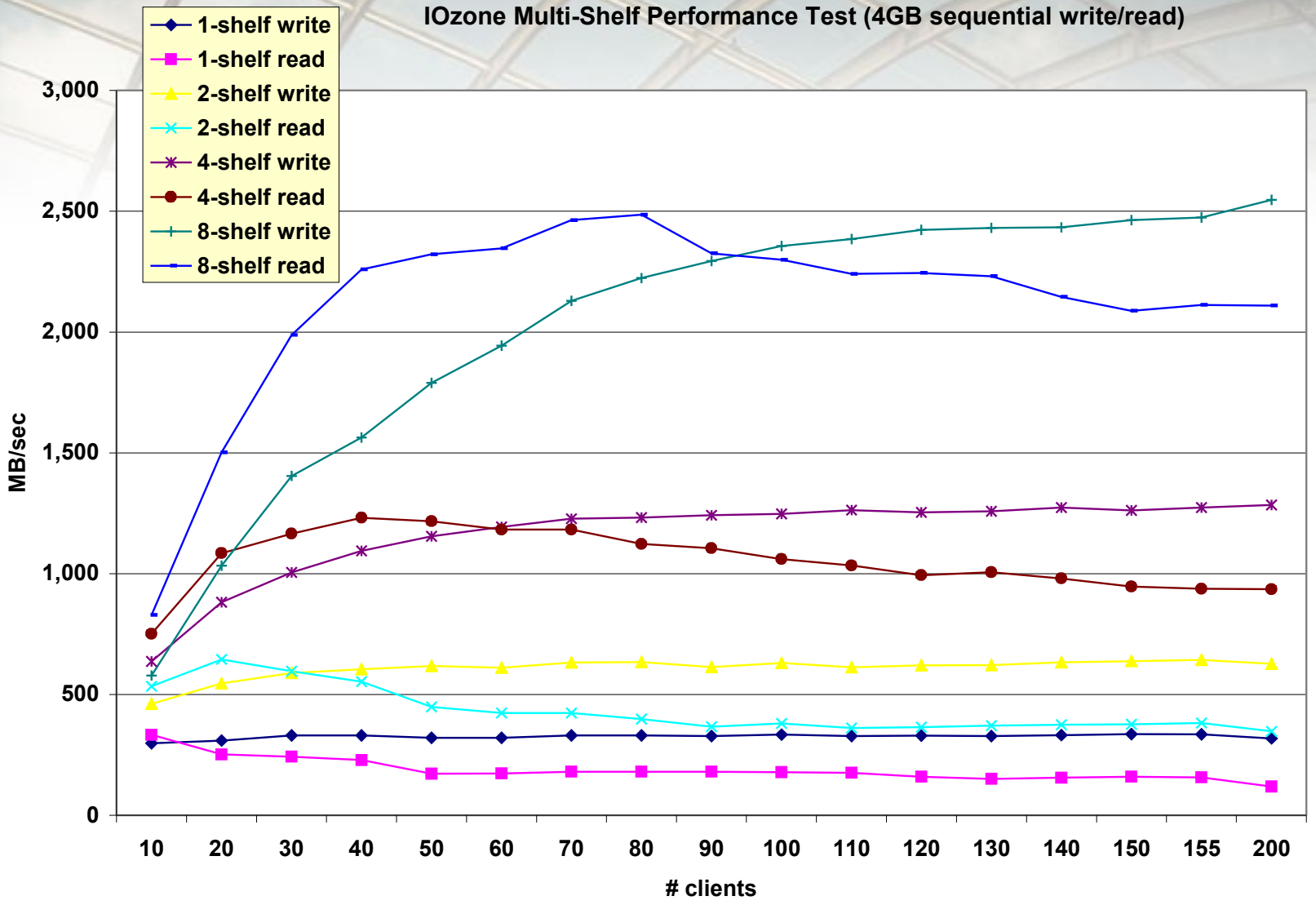


Scaling the system

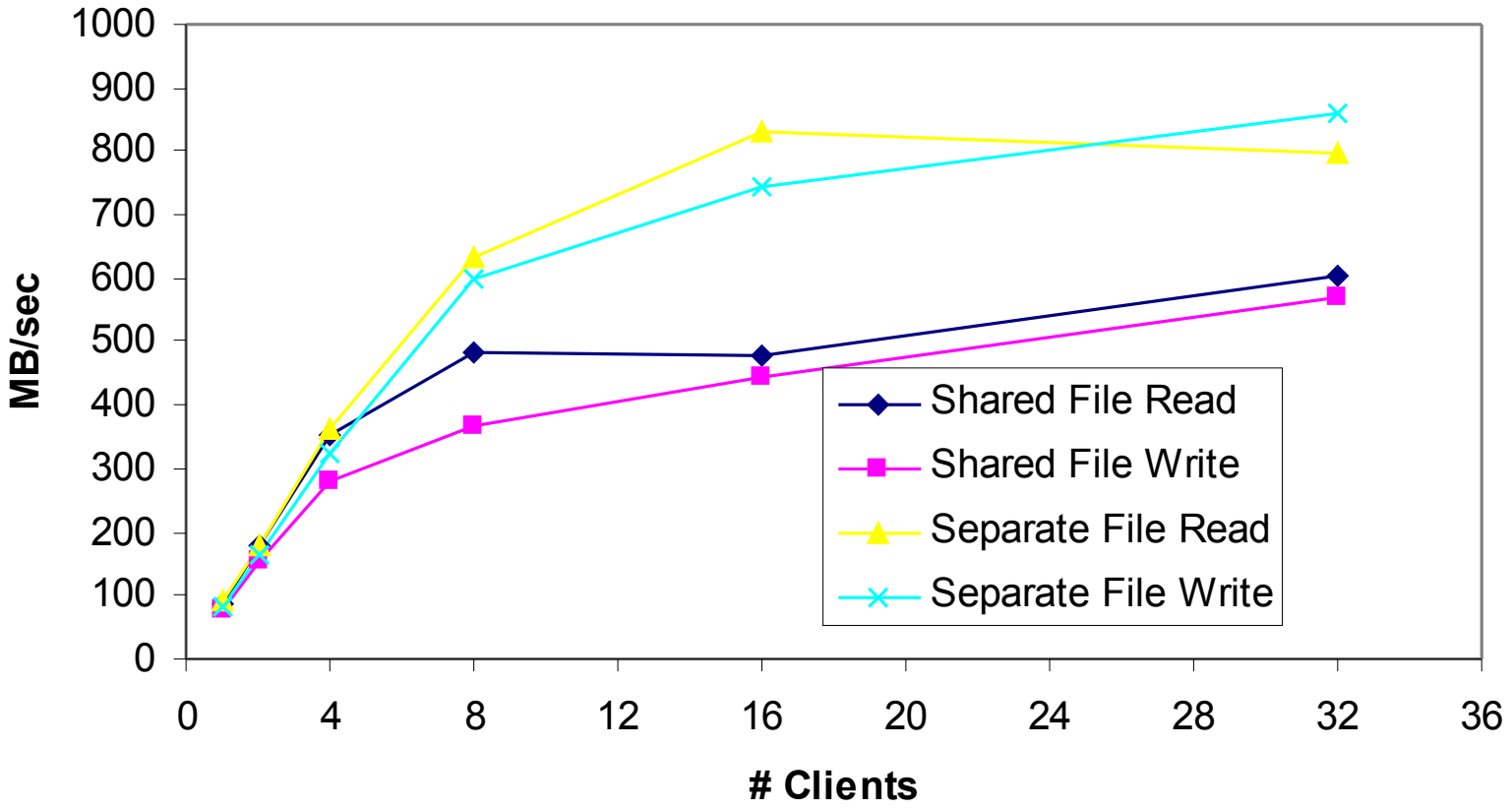
- Scale the system and clients at the same time (N-to-N IOzone)



Scaling Clients

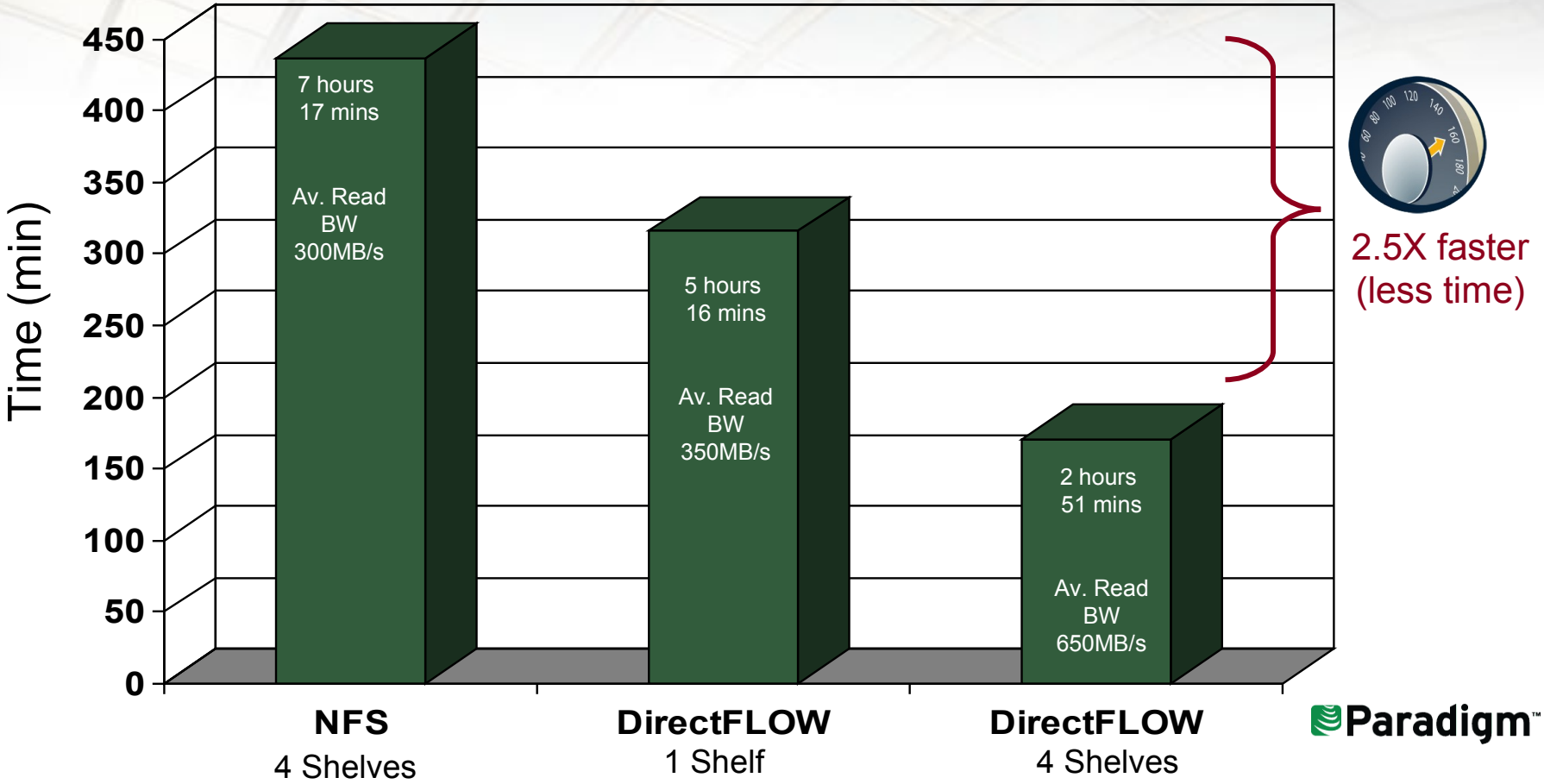


IOR Segmented IO



Panasas Parallel Storage Outperforms Clustered NFS

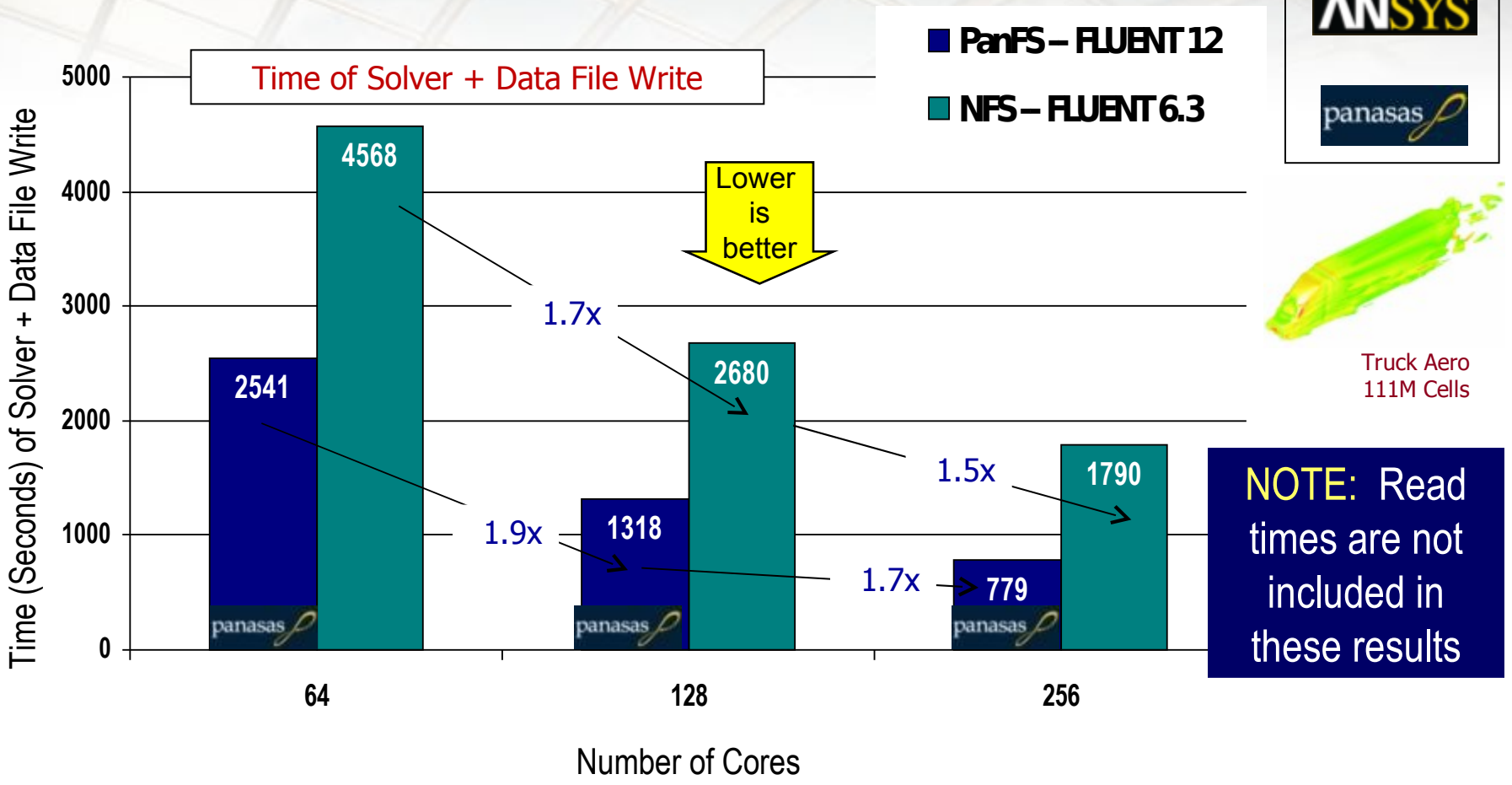
Paradigm GeoDepth Prestack Migration



Source: Paradigm & Panasas, February 2007

Scalability of Solver + Data File Write

FLUENT Comparison of PanFS vs. NFS on University of Cambridge Cluster



Truck Aero
111M Cells

NOTE: Read times are not included in these results

Details of the FLUENT 111M Cell Model

Unsteady external aero for 111 MM cell truck; 5 time steps with 100 iterations, and a single .dat file write

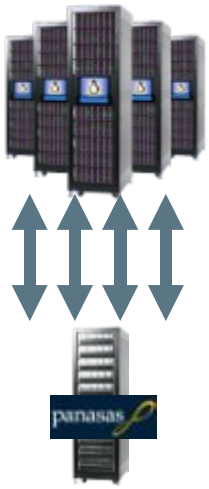
Number of cells		111,091,452
Solver		PBNS, DES, Unsteady
Iterations		5 time steps, 100 total iters - data save after last iteration
Output size:		
FLUENT v6.3	(serial I/O; size of .dat file)	14,808 MB
FLUENT v12	(serial I/O; size of .dat file)	16,145 MB
FLUENT v12	(parallel I/O; size of .pdat file)	19, 683 MB

UNIVERSITY OF CAMBRIDGE

ANSYS

panasas

DARWIN 585 nodes; 2340 cores



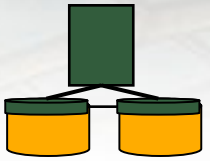
Panasas: 4 Shelves, 20 TB

Univ of Cambridge DARWIN Cluster	
Location: University of Cambridge	http://www.hpc.cam.ac.uk
Vendor: Dell ; 585 nodes; 2340 cores; 8 GB per node; 4.6 TB total mem	
CPU: Intel Woodcrest DC, 3.0 GHz / 4MB L2 cache	
Interconnect: InfiniPath QLE7140 SDR HCAs; Silverstorm 9080 and 9240 switches	
File System: Panasas PanFS, 4 shelves, 20 TB capacity	
Operating System: Scientific Linux CERN SLC release 4.6	

Automatic per-file RAID

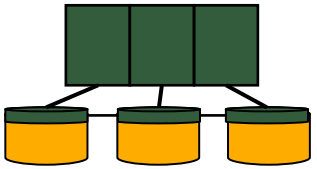
- System assigns RAID level based on file size
  - <= 64 KB RAID 1 for efficient space allocation
  - > 64 KB RAID 5 for optimum system performance
  - > 1 GB two-level RAID-5 for scalable performance
  - RAID-1 and RAID-10 for optimized small writes
- Automatic transition from RAID 1 to 5 without re-striping
- Programmatic control for application-specific layout optimizations
  - Create with layout hint
  - Inherit layout from parent directory

Small File



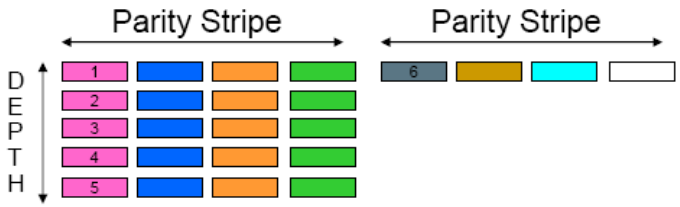
RAID 1 Mirroring

Large File



RAID 5 Striping

Very Large File



2-level RAID 5 Striping

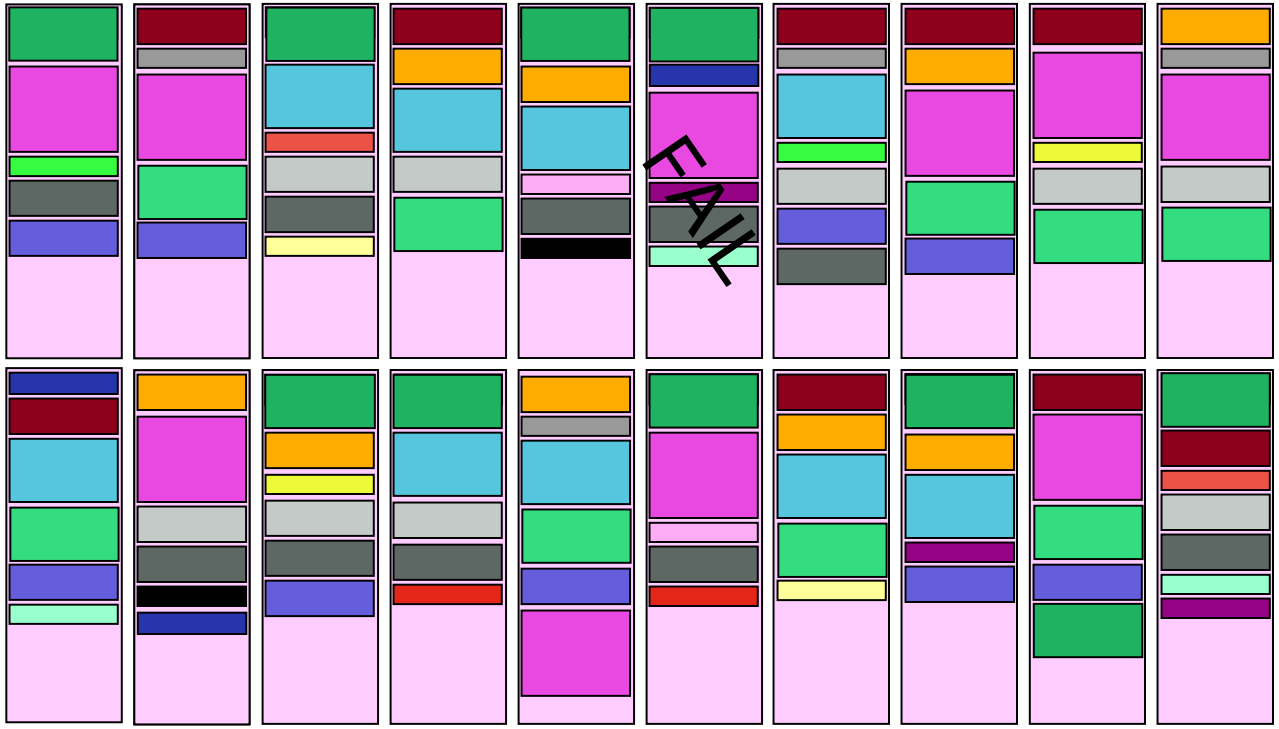
Clients are responsible for writing data and its parity

Declustered RAID

- Files are striped across component objects on different StorageBlades
- Component objects include file data and file parity for reconstruction
- File attributes are replicated with two component objects
- Declustered, randomized placement distributes RAID workload

2-shelf BladeSet

Mirrored or 9-OSD Parity Stripes



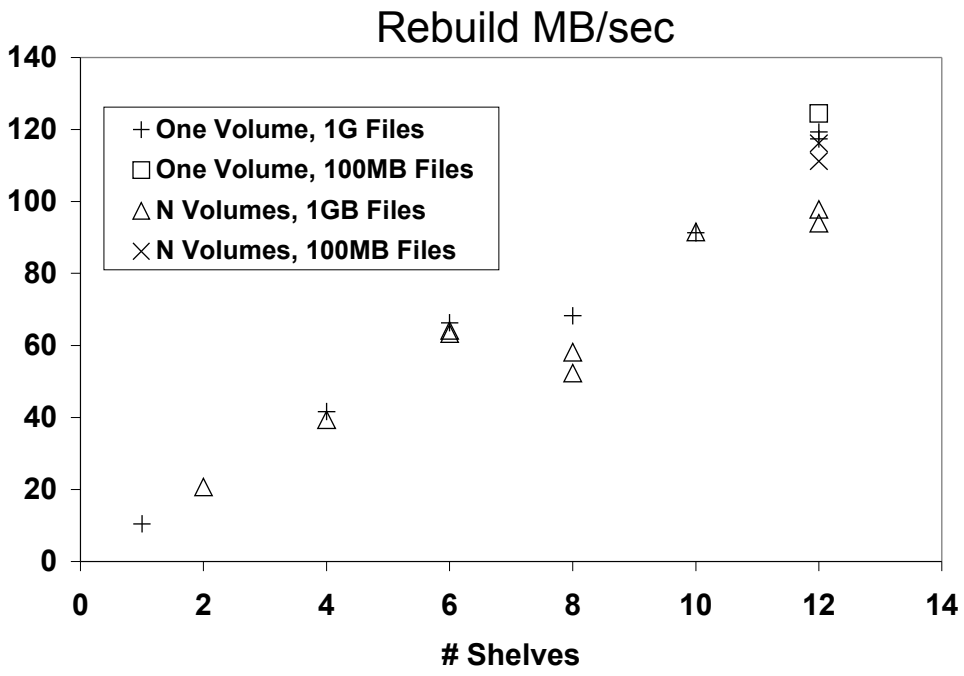
Read about half of each surviving OSD
Write a little to each OSD

Scales linearly

Scalable RAID Rebuild

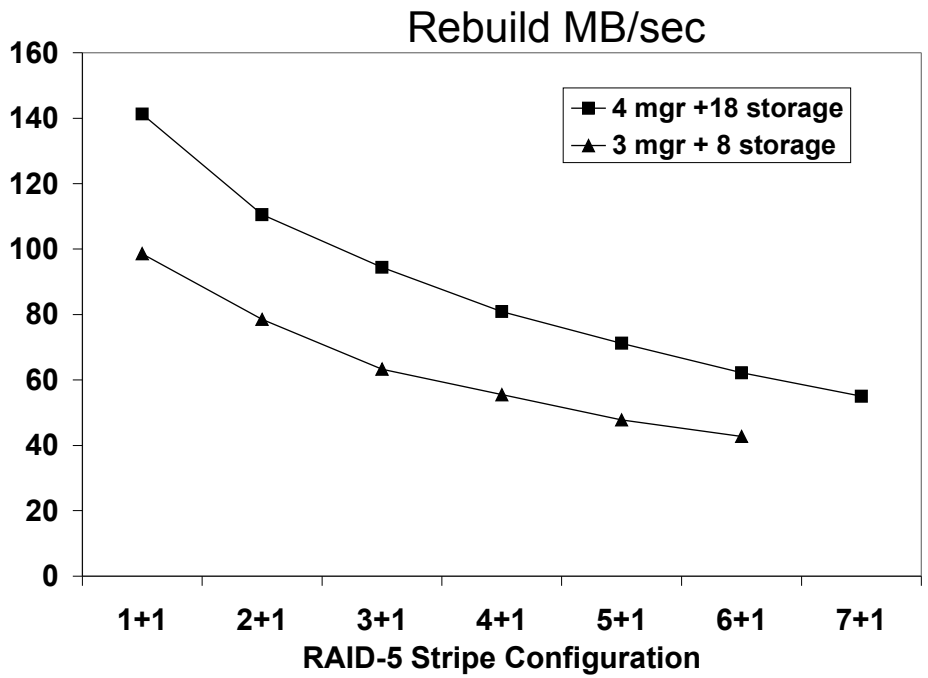
- Rebuild bandwidth is the rate at which data is regenerated (writes)
  - Overall system throughput is N times higher because of the necessary reads
  - Use multiple “RAID engines” (DirectorBlades) to rebuild files in parallel
  - Declustering spreads disk I/O over more disk arms (StorageBlades)

- Shorter repair time in larger storage pools
  - Customers report 30 minute rebuilds for 800GB in 40+ shelf blade set
- Variability at 12 shelves due to uneven utilization of DirectorBlade modules
  - Larger numbers of smaller files was better
- Reduced rebuild at 8 and 10 shelves because of wider parity stripe



RAID Rebuild vs Stripe Width

- Panasas system automatically selects stripe width up to 11 wide
  - 8 to 11 wide is best for bandwidth performance
  - System packs an even number of stripes into Bladeset, leaving at least one spare
- Narrower stripes rebuild faster
  - Less data to read to reconstruct writes
- More DirectorBlades helps
  - 1, 2, or 3 per shelf
  - 50+ in a single system



Scalable rebuild is mandatory

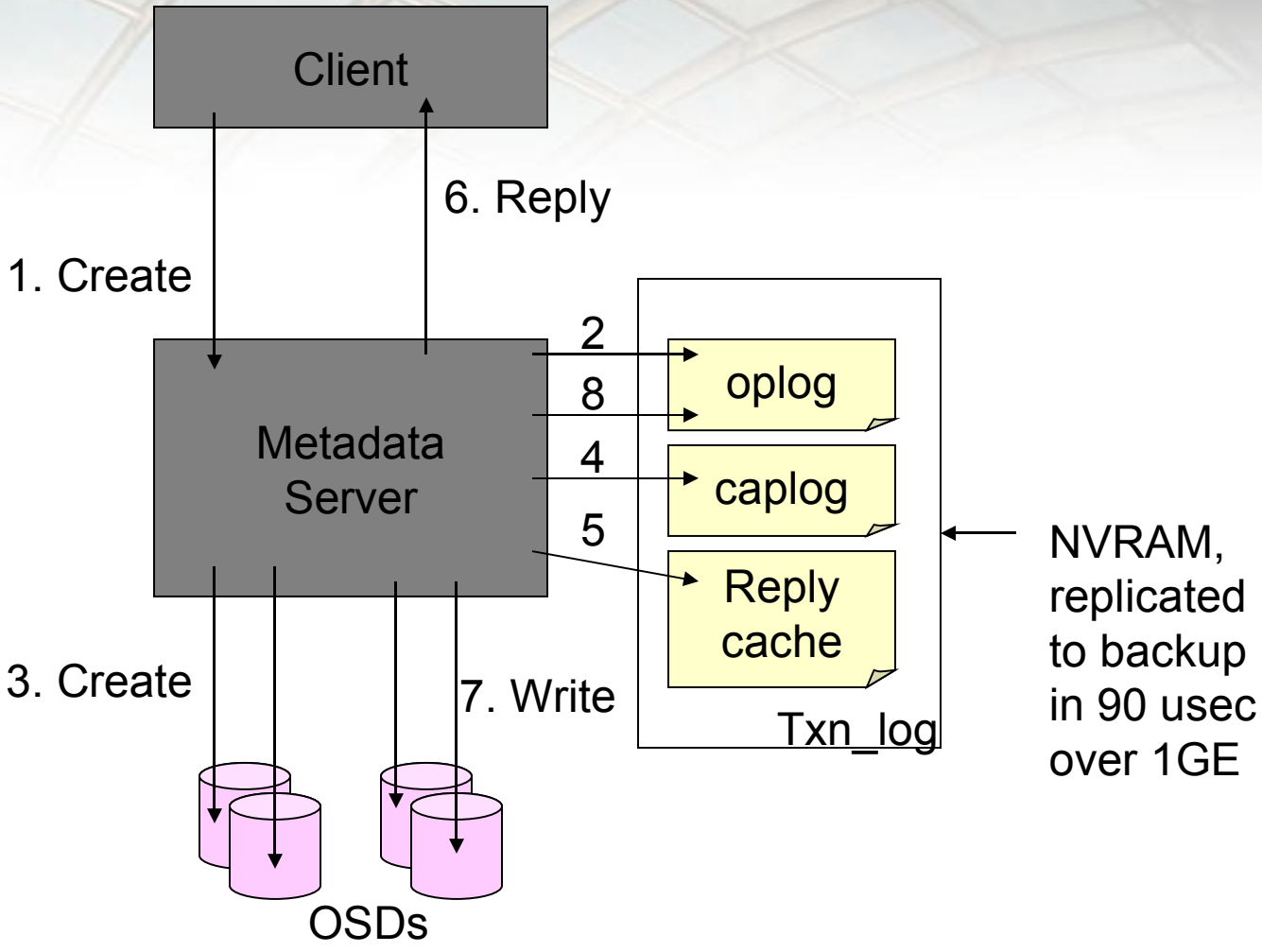
- Having more drives increases risk, just like having more light bulbs increases the odds one will be burnt out at any given time
- Larger storage pools must mitigate their risk by decreasing repair times
- The math says
  - if (e.g.) 100 drives are in 10 RAID sets of 10 drives each and
  - each RAID set has a rebuild time of N hours
  - The risk is the same if you have a single RAID set of 100 drives
  - and the rebuild time is N/10
- Block-based RAID scales the wrong direction for this to work
  - Bigger RAID sets repair more slowly because more data must be read
- Only declustering provides scalable rebuild rates

$$MTTF_{RAID} = \frac{(MTTF_{Disk})^2}{(D+C*n_G)*(G+C-1)*MTTR}$$

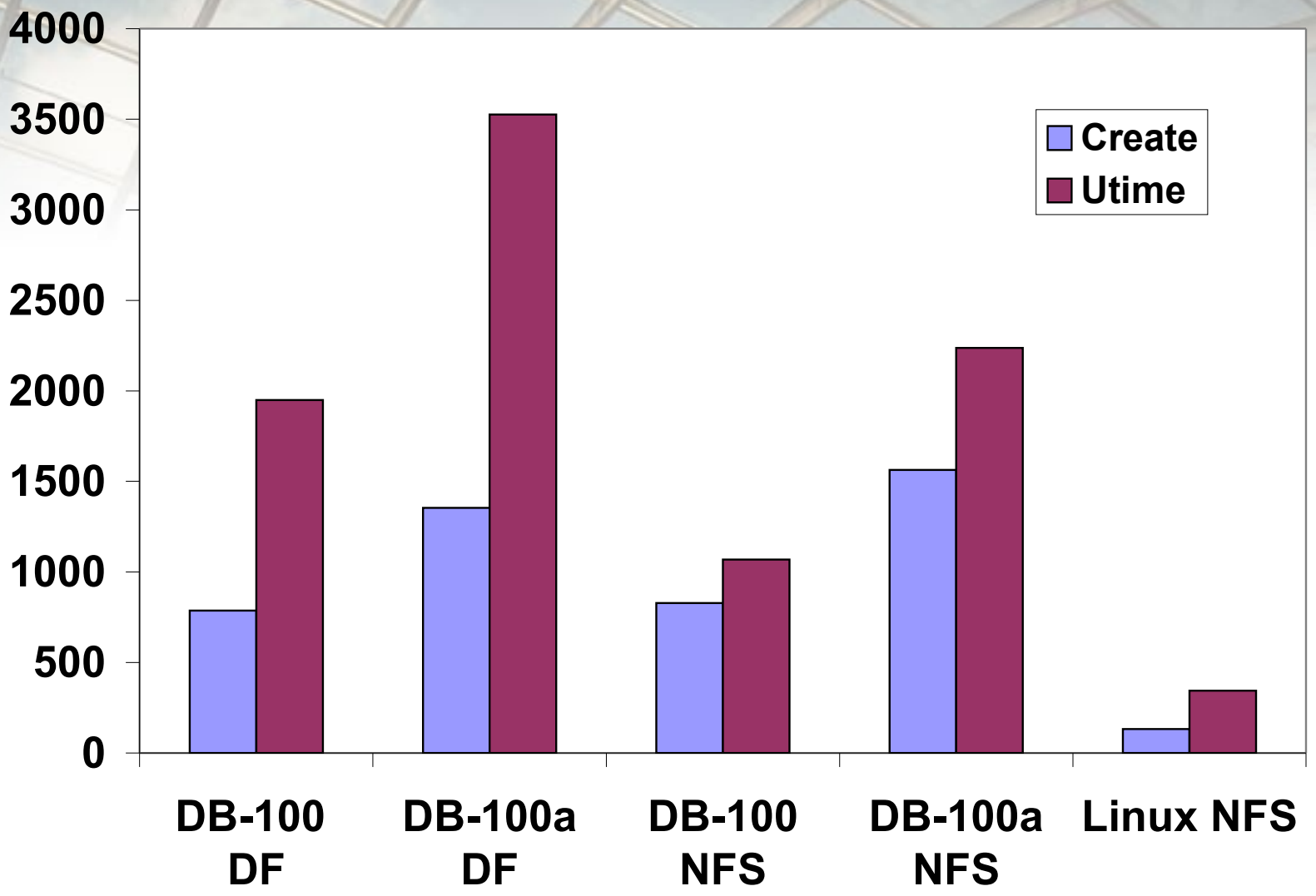
Total number of drives — Drives per RAID set — Repair time

Creating a File in 2 milliseconds

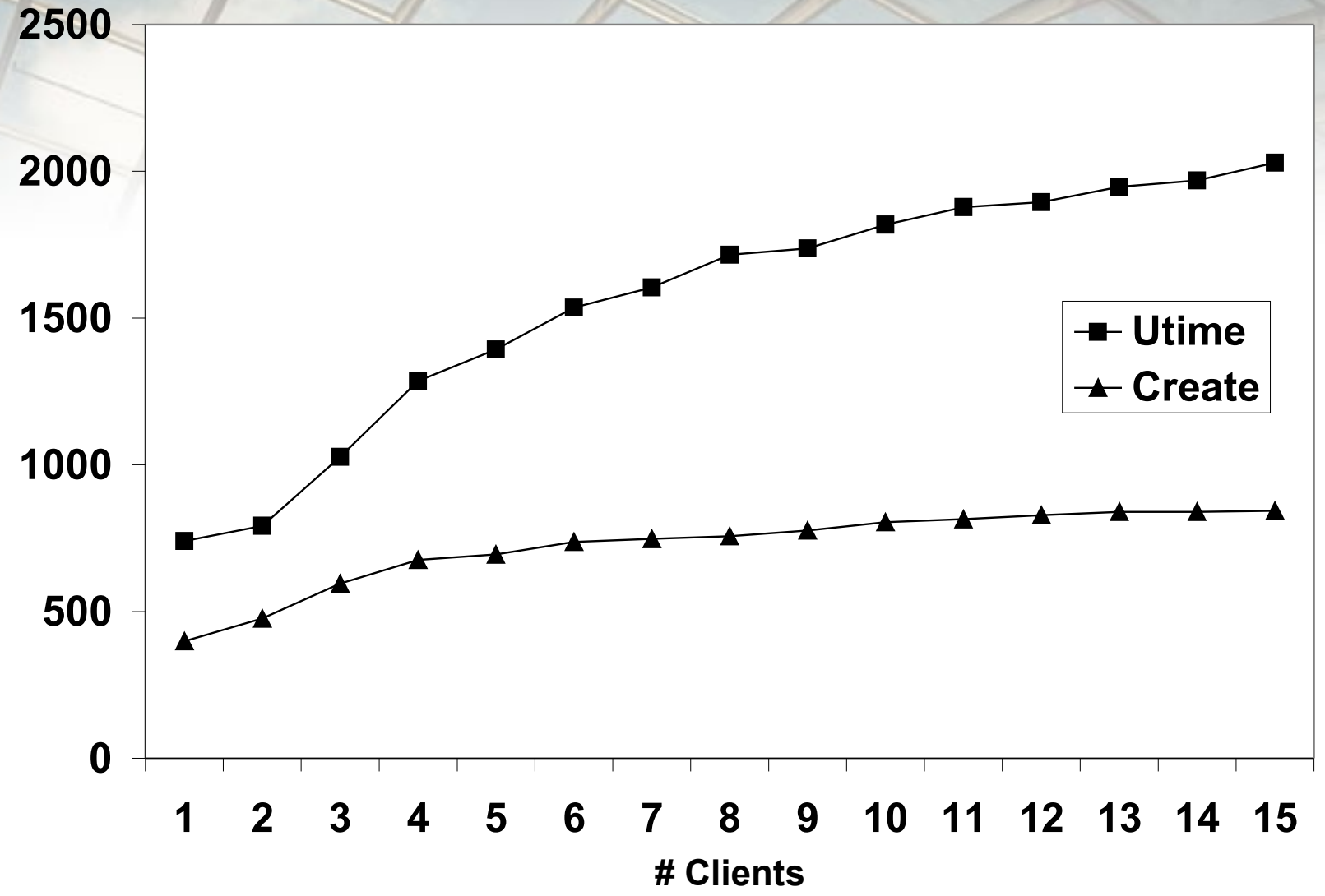
Directories are objects with lists of name to object ID and location hint mappings



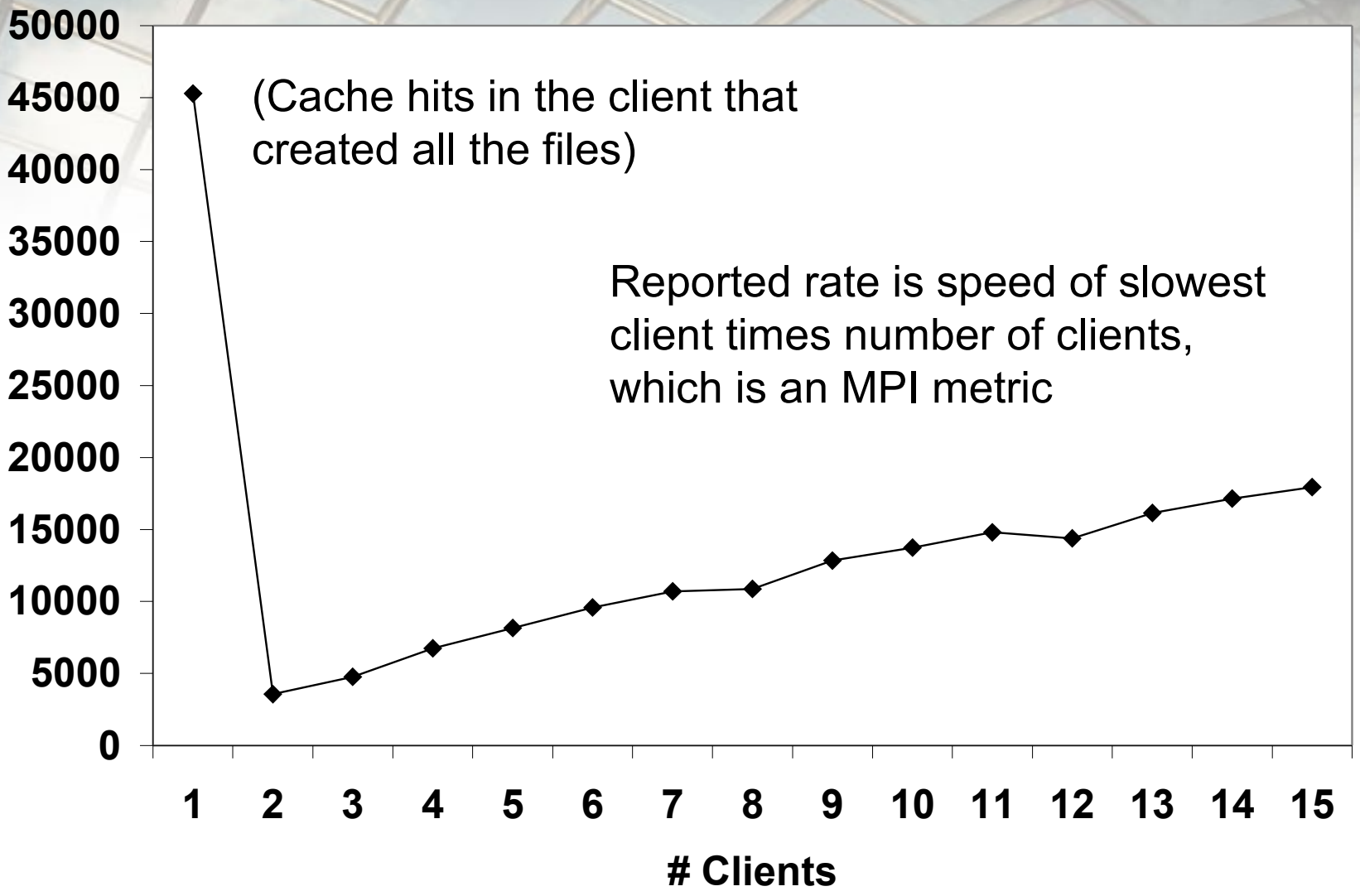
Metarate operations/sec



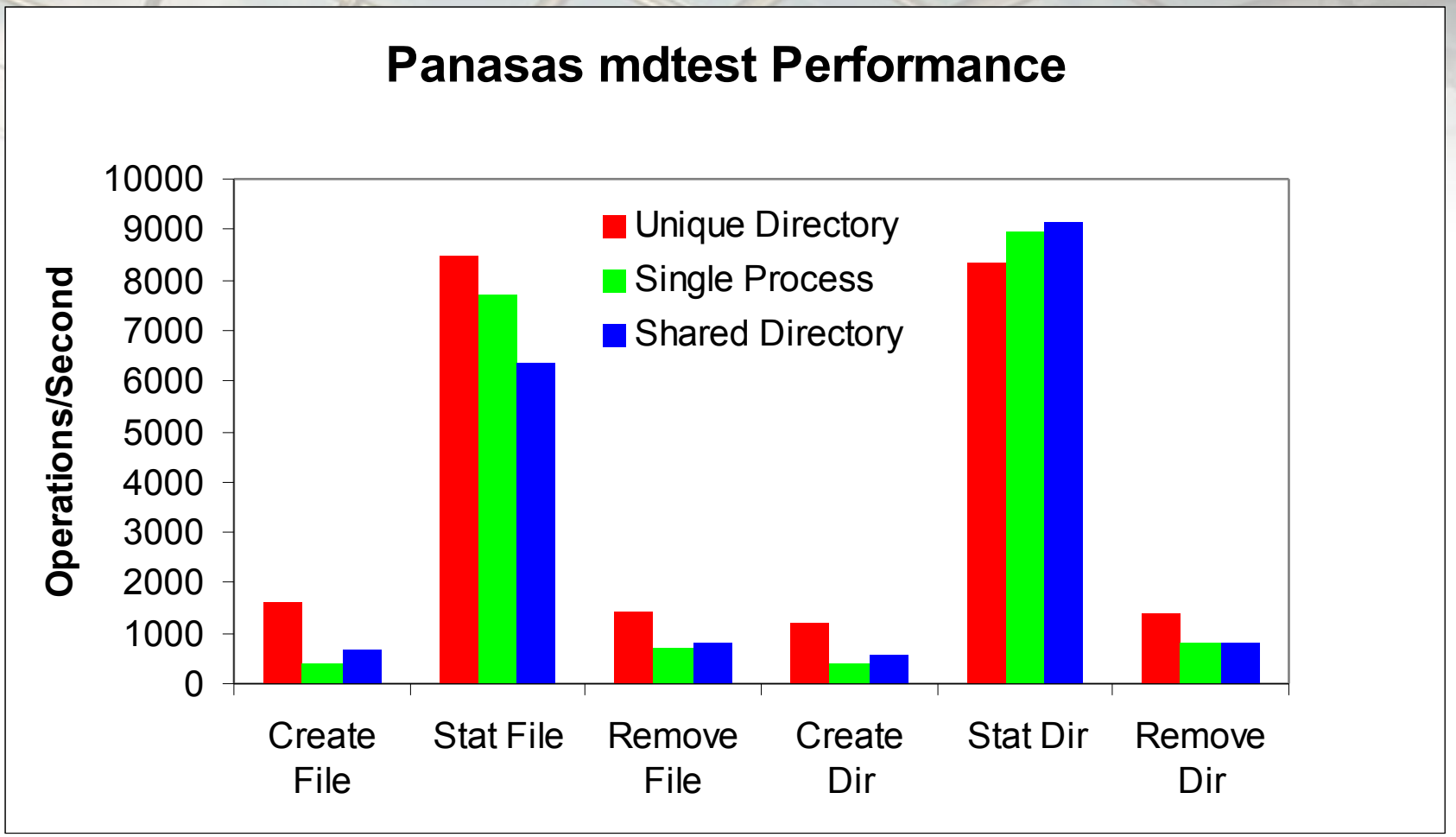
Metarate operations/sec



Metarate operations/sec



MPI Coordinated File Operations



mpirun -n 64 mdtest -d $dir -n 100 -i 3 -N 1 -v -u

Summary

- Per-file, object-based RAID gives scalable on-line performance
  - Offloads the metadata server
  - Parallel block allocation among the storage nodes
- Declustered parity group placement yields linear increase in rebuild rates with the size of the storage pool
  - May become the only way to effectively handle large capacity drives
- Metadata is stored as attributes on objects
  - File create is complex, but made fast with efficient journal implementation
  - Coarse-grained metadata workload distribution is a simple way to scale

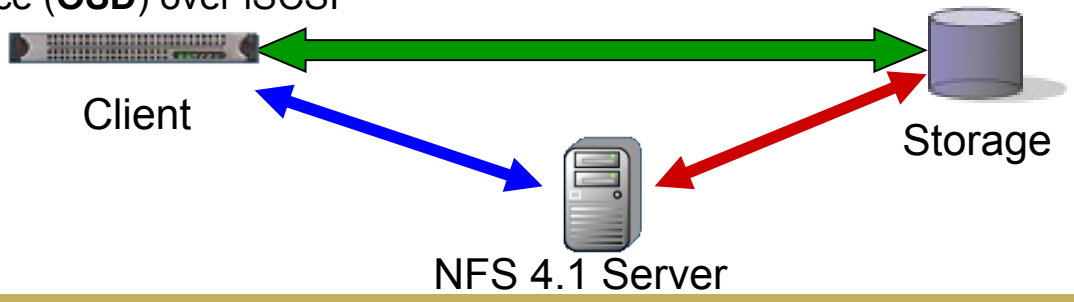
Technology Review

- Turn-key deployment and automatic resource configuration
- Scalable Object RAID
- Very fast RAID rebuild
- Vertical Parity to trap silent corruptions
- Network parity for end-to-end data verification
- Distributed system platform with quorum-based fault tolerance
- Coarse grain metadata clustering
- Metadata fail over
- Automatic capacity load leveling

- Storage Clusters scaling to ~1000 nodes today
- Compute clusters scaling to 12,000 nodes today
- Blade-based hardware with 1Gb/sec building block
  - Bigger building block going forward

The pNFS Standard

- The **pNFS** standard defines the NFSv4.1 protocol extensions between the **server** and **client**
- The **I/O** protocol between the **client** and **storage** is specified elsewhere, for example:
  - SCSI **Block** Commands (**SBC**) over Fibre Channel (**FC**)
  - SCSI **Object**-based Storage Device (**OSD**) over iSCSI
  - Network **File** System (**NFS**)
- The **control** protocol between the **server** and **storage** devices is also specified elsewhere, for example:
  - SCSI **Object**-based Storage Device (**OSD**) over iSCSI



Key pNFS Participants

- Panasas (Objects)
- Network Appliance (Files over NFSv4)
- IBM (Files, based on GPFS)
- EMC (Blocks, HighRoad MPFSi)
- Sun (Files over NFSv4)
- U of Michigan/CITI (Files over PVFS2)

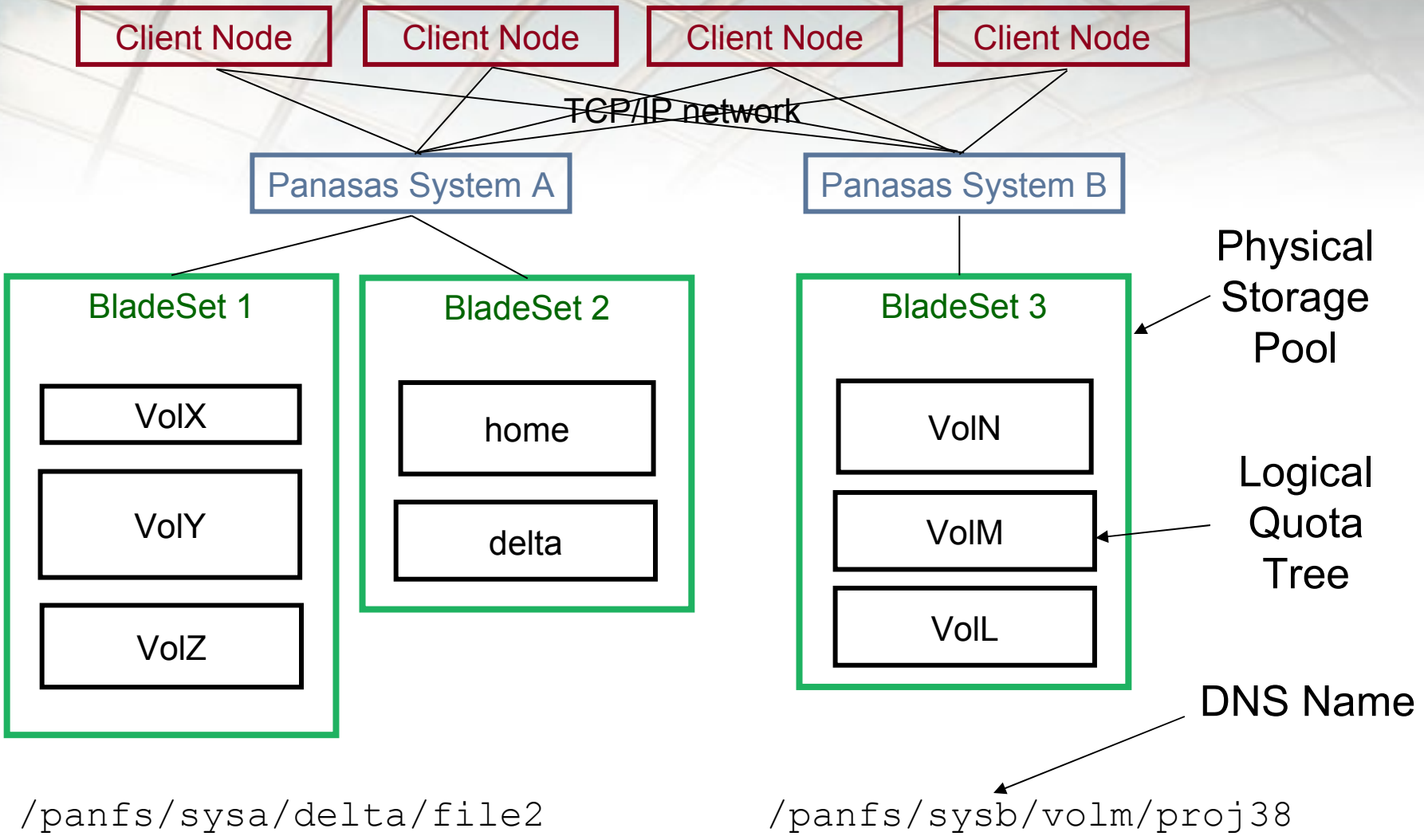
pNFS Status

- pNFS is part of the IETF NFSv4 minor version 1 standard draft
  - Working group is passing draft up to IETF area directors, expect RFC later in ’08
- Prototype interoperability continues
  - San Jose Connect-a-thon March ’06, February ’07, May ‘08
  - Ann Arbor NFS Bake-a-thon September ’06, October ’07
  - Dallas pNFS inter-op, June ’07, Austin February ’08, (Sept ’08)
- Availability
  - TBD – gated behind NFSv4 adoption and working implementations of pNFS
  - Patch sets to be submitted to Linux NFS maintainer starting “soon”
  - Vendor announcements in 2008
  - Early adoptors in 2009
  - Production ready in 2010

Questions?

Thank you for your time!

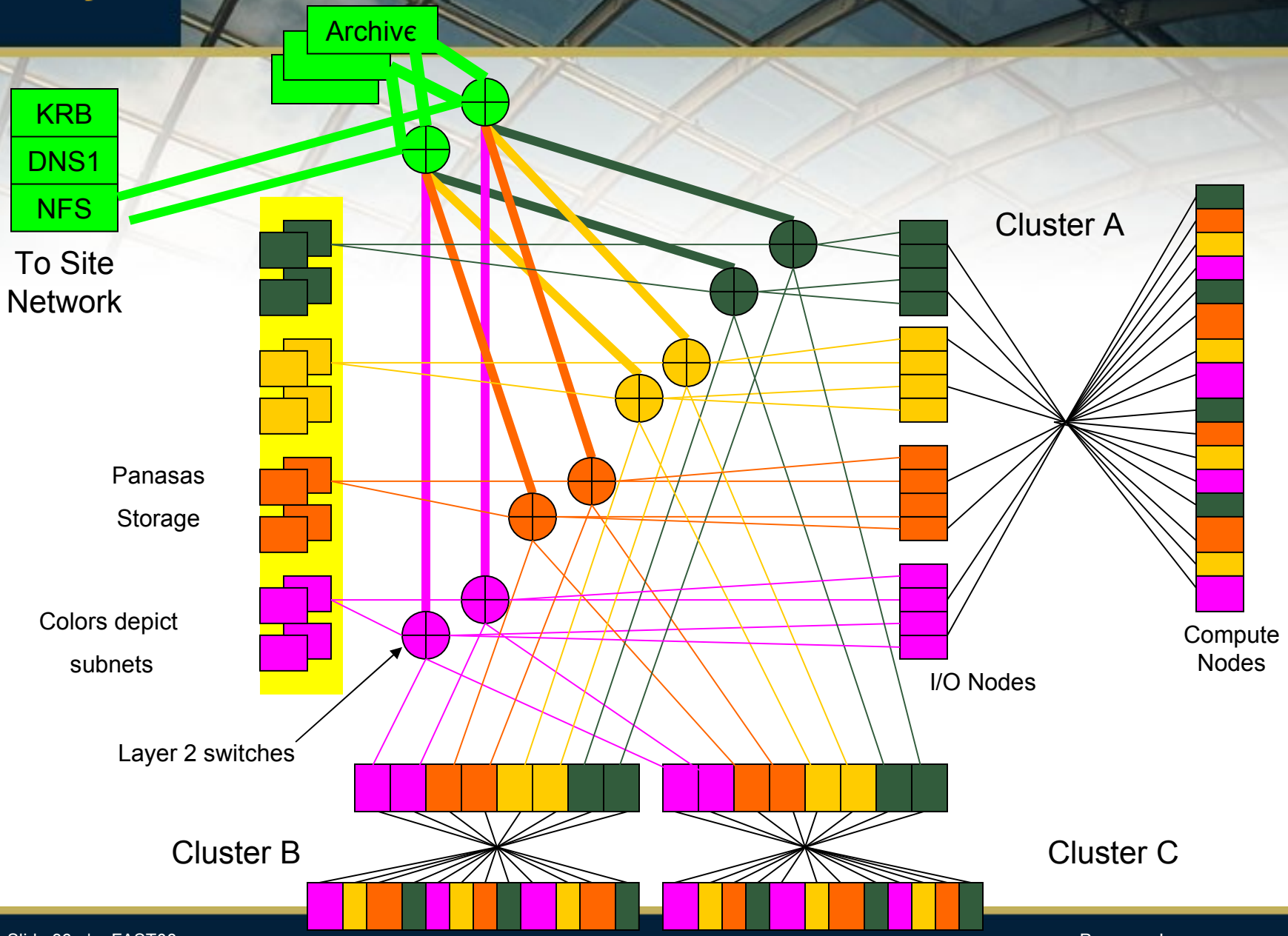
Panasas Global Storage Model



IB and other network fabrics

- Panasas is a TCP/IP, GE-based storage product
  - Universal deployment, Universal routability
  - Commodity price curve
- Panasas customers use IB, Myrinet, Quadrics, …
  - Cluster interconnect *du jour* for performance, not necessarily cost
- IO routers connect cluster fabric to GE backbone
  - Analogous to an “IO node”, but just does TCP/IP routing (no storage)
  - Robust connectivity through IP multipath routing
  - Scalable throughput at approx 650 MB/sec IO router (PCI-e class)
    - Working on a 1GB/sec IO router
- IB-GE switching platforms
  - Cisco/Voltare switch provides wire-speed bridging

Multi-Cluster sharing: scalable BW with fail over



New and Unique: Network Parity

- Extends parity capability across the data path to the client or server node
- Enables *End-to-End* data integrity validation
  - Protects from errors introduced by disks, firmware, server hardware, server software, network components and transmission
  - Client either receives valid data or an error notification