Scaling File Service Up and Out

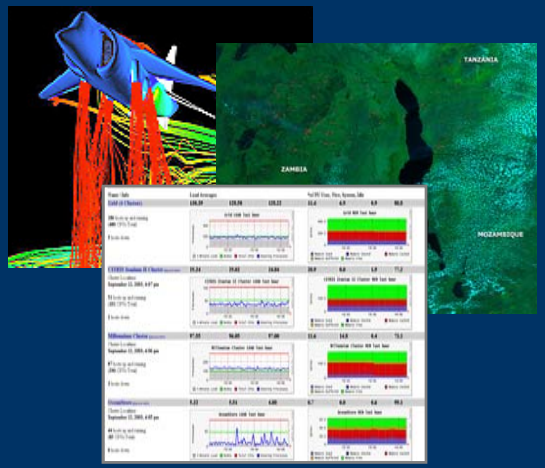
Garth Gibson
garth@panasas.com

Panasas Inc, and
Carnegie Mellon University

3rd USENIX Conf on File and Storage Technologies, March 31, 2004

Quality Drives Growth

High Performance Computing



- 100+ Teraflops
- Throughput Goal: **1 GB/sec per Teraflop = 100 GB/sec**

Commercial Media Applications

- Rendering & Post-Production
- Throughput Goal: **5 min/hr = 1.2 GB/sec texture & polygons**

Consumer Media Applications

- Data rate & capacity growth
- Throughput Goal: **acceptable to mass market**

Los Alamos NATIONAL LABORATORY · San Diego Supercomputer Center · Sandia National Laboratories · NOAA · Berkeley Lab

PIXAR Animation Studios · Walt Disney Feature Animation · Skywalker Ranch · PDI DreamWorks

DIRECTV · TiVo · real · ImageStation

Enabling Growth: Linux Clusters

Monolithic Supercomputers

Linux Clusters

- Specialized, but expensive
- Price/performance: often > **$100M/TFLOPS**

- Powerful, scalable, affordable
- Price/performance: often < **$1M/TFLOPS**

Clusters dominating Top500 Supercomputers:	
1998:	**2**
2002:	**94**
2003:	**208**

Source: Top500.org

panasas

Demand New Scale & Cost Effectiveness

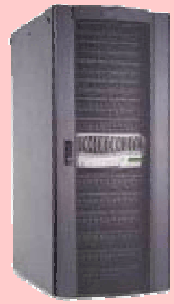
Traditional HPC Computing

Monolithic Computers

BOTTLENECK

A few data paths

Monolithic Storage



Cluster Computing

Linux Compute Cluster

?

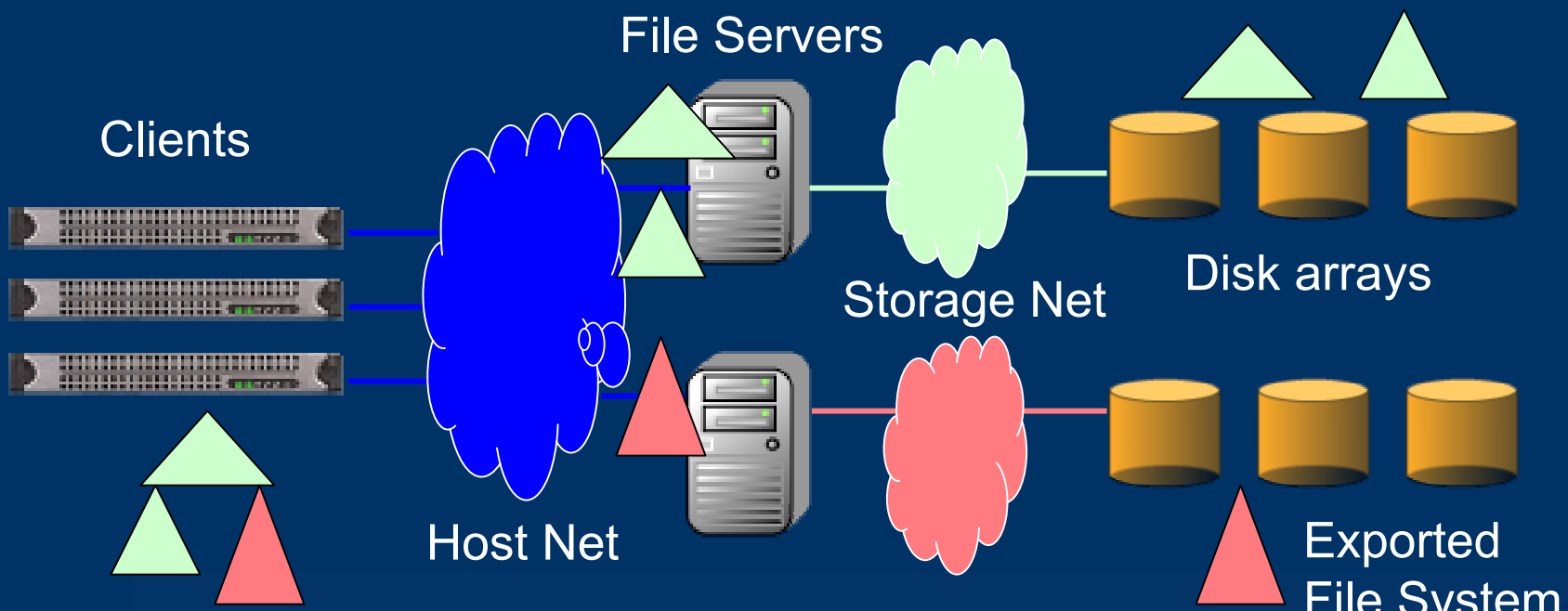
Scale to bigger box?

Scale:
file & total bandwidth
file & total capacity
load & capacity balancing

But lower $ / Gbps

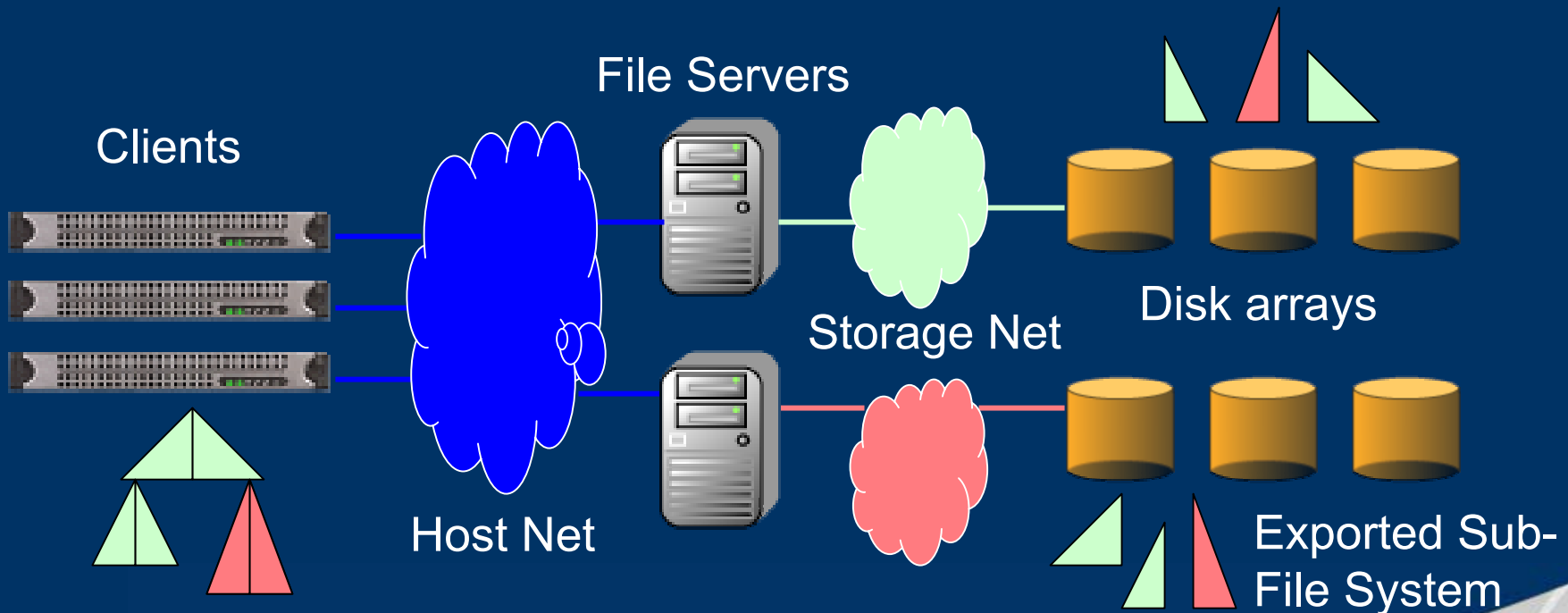
Limits on Scaling Up NAS File Server

- **Bigger file server boxes can't keep up with growth of demand or supply**
  - Cluster client demand scales as client speed X cluster size
  - Storage supply scales as disk speed X disk array size X arrays on SAN
  - But file server data bandwidth scales at only the pace of single client speed



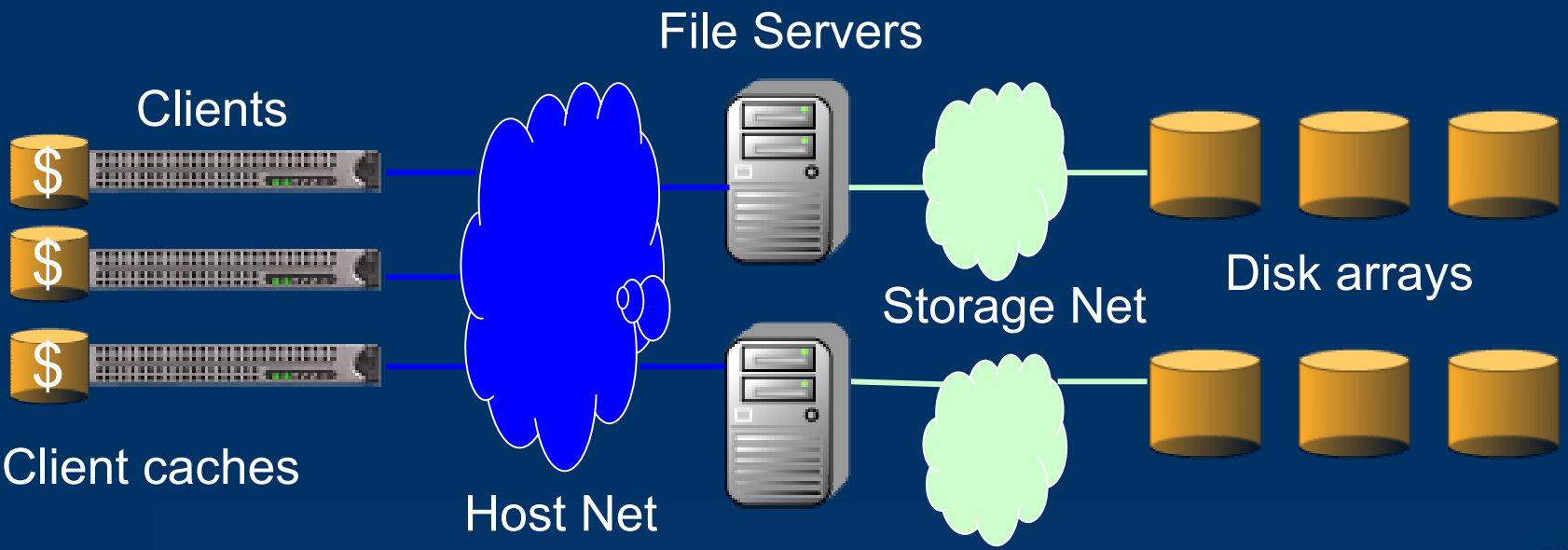
Today's NAS Clustering Practices

- **Customer namespace decomposition over multiple server subsystems**
  - Customer administrator does decomposition
    - Implications capacity & load balance, backup & quota management
  - Expensive administrator given messy, never ending, rarely satisfactory tasks
  - Single files & directory decomposition very visible to users -- more expensive still



Scale Out Phase 1: Exploit Clients

- **Opportunistic offloading to clients, especially data caching**
  - Offload read hit and defer/coalesce write: 80s Sprite/Welch, AFS/Howard
  - Offload RAID computation: 90s Zebra/Hartman
  - Offload locking, open-close cache validation: 00s NFSv4
  - With dramatic growth of cost effective cycles in cluster clients, expect more of this
- **Helps, but too many workloads are too big, too little reuse to be solution**



panasas

Clusters Demand a New Storage Architecture

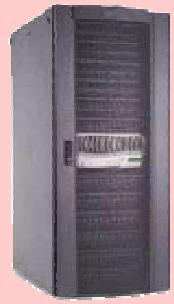
Traditional HPC Computing

Monolithic Computers

BOTTLENECK

A few data paths

Monolithic Storage



Cluster Computing

Linux Compute Cluster

?

More cost-effective boxes

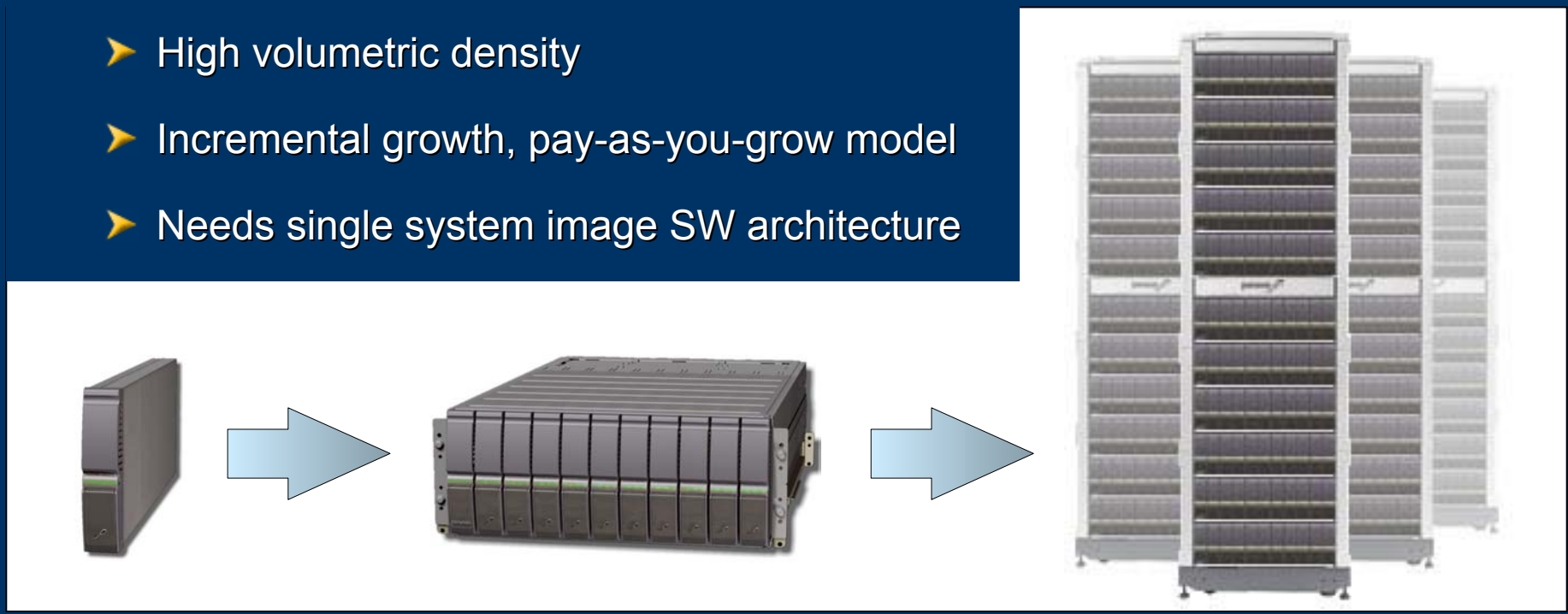
Scale:
file & total bandwidth
file & total capacity
load & capacity balancing

With lower $ / Gbps

Scalable Storage Cluster Architecture

Lesson of compute clusters: Scale out commodity components

- **Bladeserver approach provides**
  - High volumetric density
  - Incremental growth, pay-as-you-grow model
  - Needs single system image SW architecture



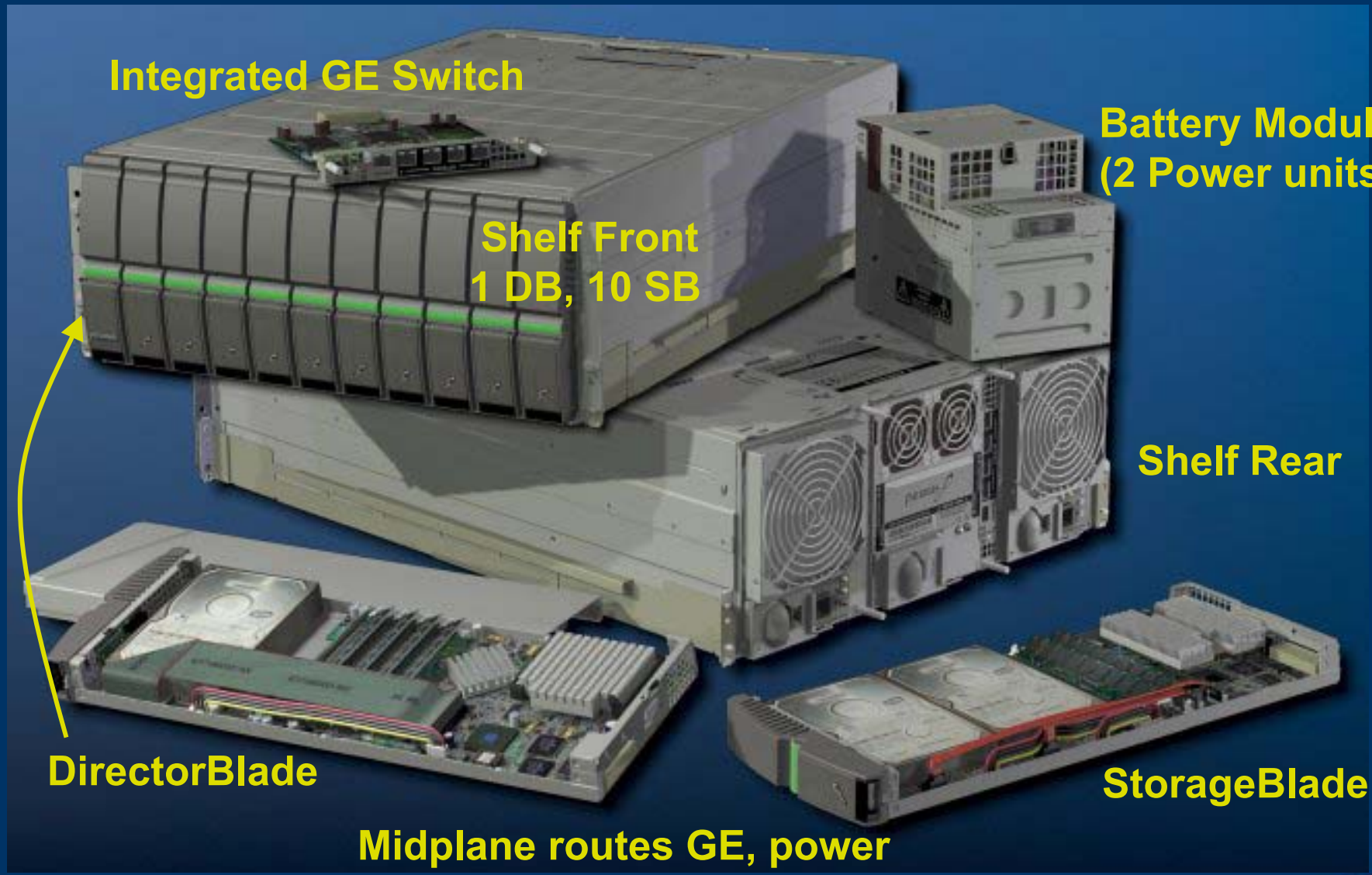
StorageBlade
2 SATA spindles

Shelf of Blades
5 TB, 4 Gbps

Single System Image
55 TB, 44 Gbps per rack

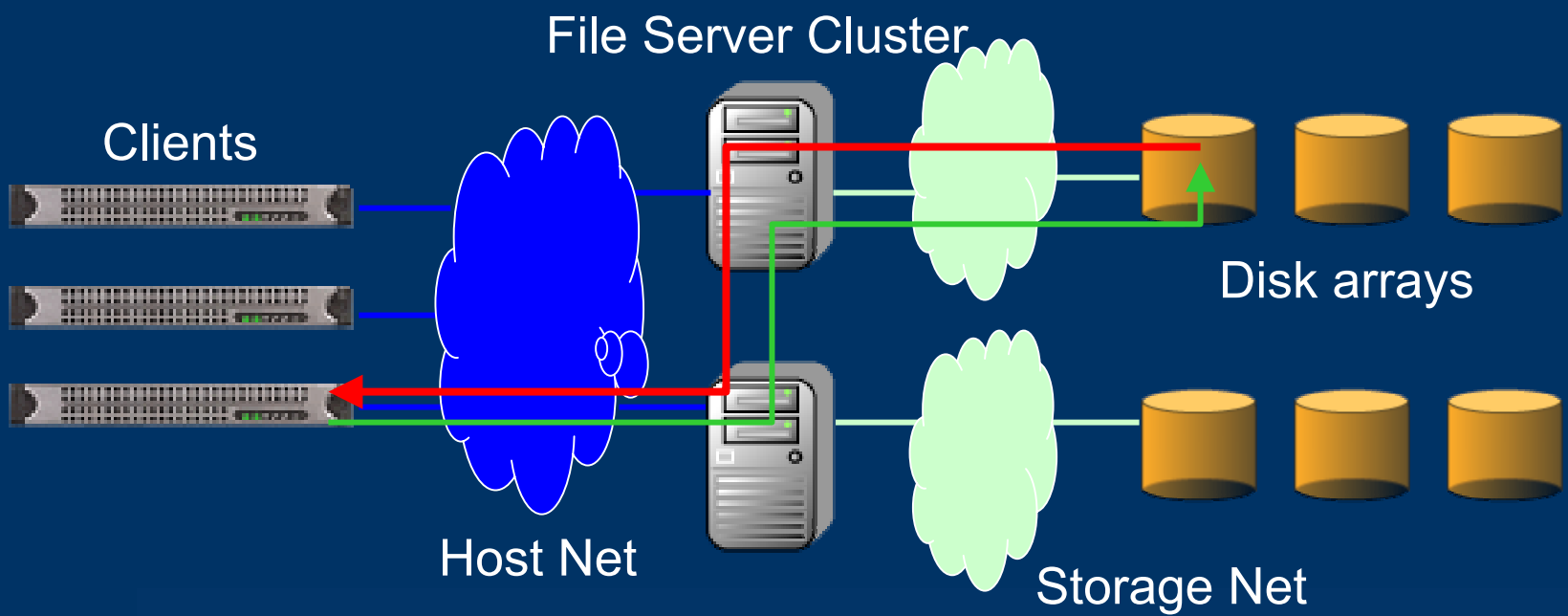
panasas

BladeServer Storage Cluster



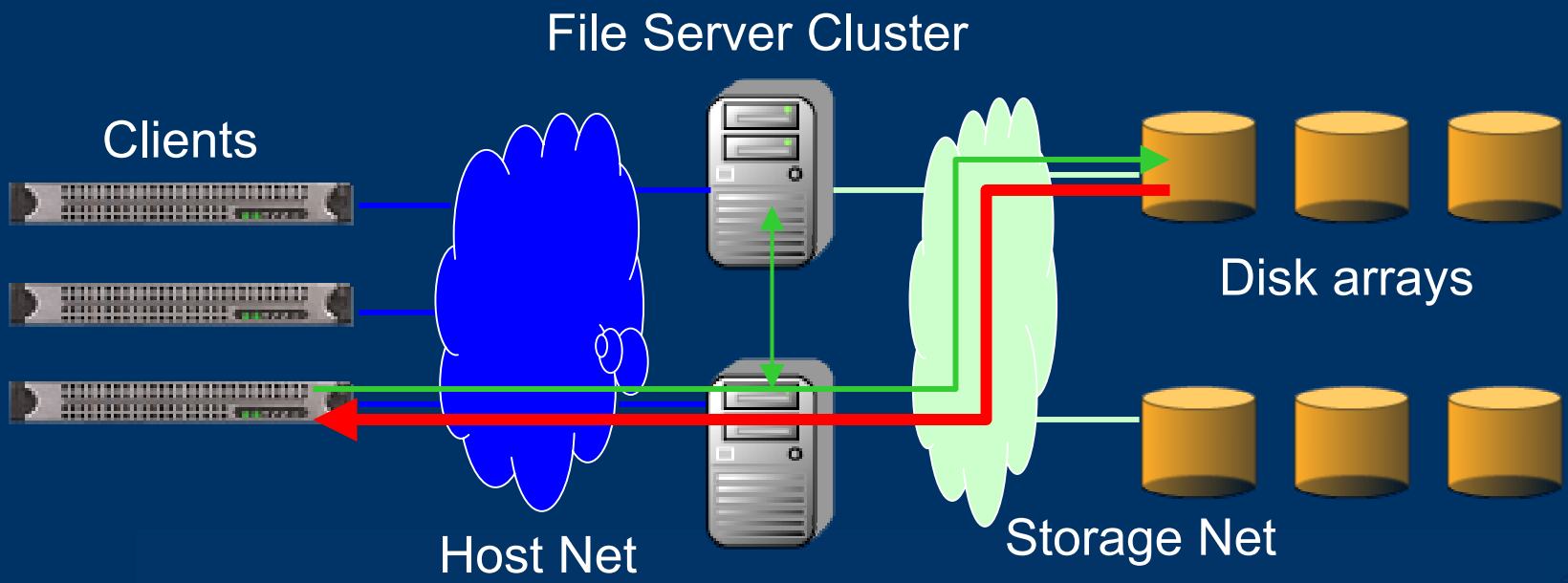
Scale Out Phase 2: Forwarding Servers

- **Bind many file servers into single system image with forwarding**
  - Mount point binding less relevant, allows DNS-style balancing, more manageable
  - Control and data traverse mount point path (in band) passing through two servers
  - Single file and single file system bandwidth limited by backend server & storage
  - Tricord, Spinnaker



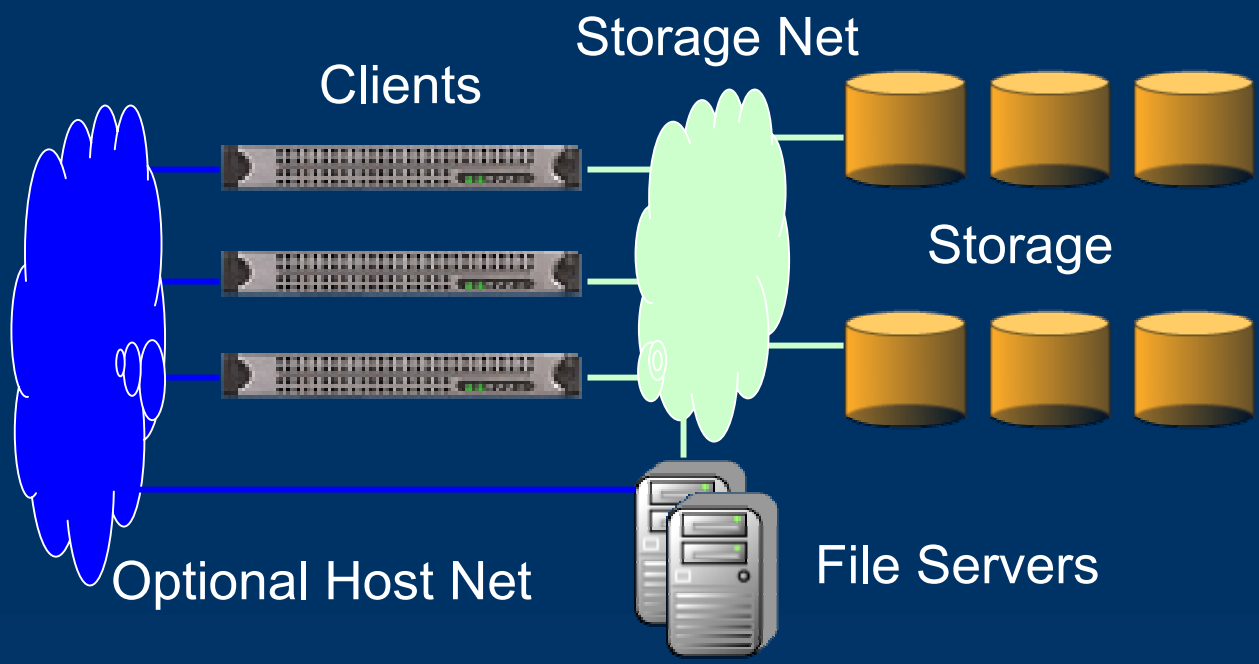
Scale Out Phase 3: Any Server Will Do

- **Single server does all data transfer in single system image**
  - Servers share access to all storage and “hand off” role of accessing storage
  - Control and data traverse mount point path (in band) passing through one server
  - Hand off cheap: difference between 100% & 0% colocated front & back end was about zero SFS97 NFS op/s and < 10% latency increase for Panasas
  - Allows single file & single filesystem capacity & bandwidth to scale with server cluster



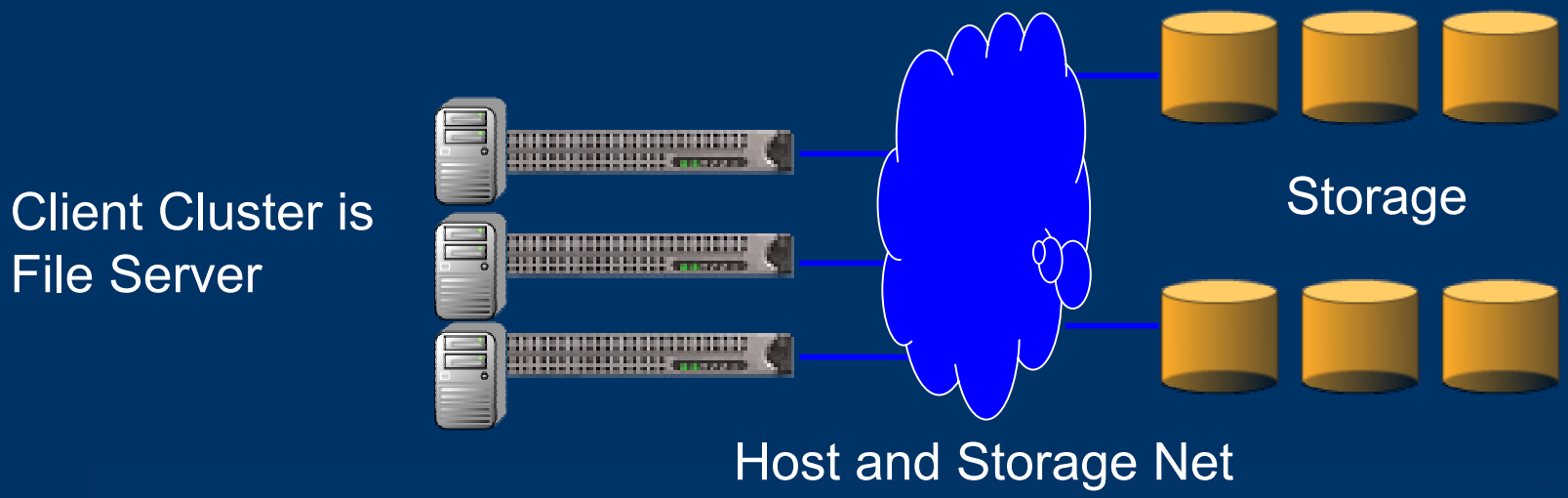
Scale Out Phase 4A: Asymmetric OOB

- **Client sees mount as file server address and many storage addresses**
  - After permission checking, client gets “map” and directly, in parallel, accesses storage
  - Zero file servers in data path: more cost-effective scaling, especially with one net
  - Many proprietary offerings: IBM SAN FS, EMC High Road, SGI CXFS, Panasas, etc
  - Mostly built on block-based SANs which forces servers to trust all data to all clients



Scale Out Phase 4S: Symmetric OOB

- **Server code is multi-threaded so every client is fully capable server**
  - Client's local file server code acquires locks & metadata from other clients as needed
  - Various proprietary offerings: RedHat GFS, IBM GPFS, Sun QFS etc
  - Trust boundary includes all clients even if storage has fine grain enforcement (objects)
  - For legacy clients, typically do hybrid, offering cluster as file server for legacy clients



New *Object Storage* Architecture

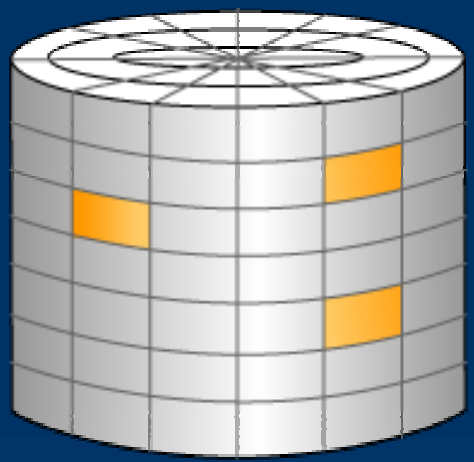
- **An evolutionary improvement to standard SCSI storage interface**
- **Raises level of abstraction: Object is container for "related" data**
  - Storage *understands* how different blocks of a "file" are related
- **Offload most datapath work from server to intelligent storage**

Block Based Disk

Operations:
Read block
Write block

Addressing:
Block range

Allocation:
External

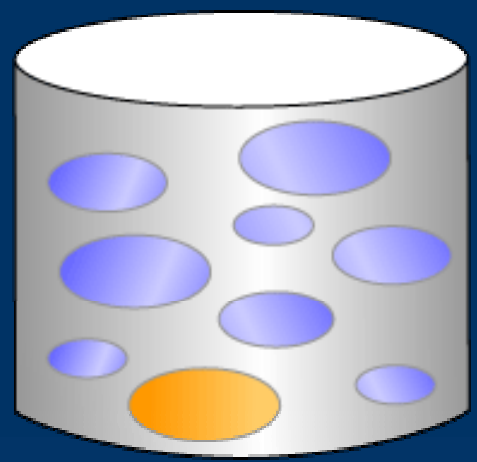


Object Based Disk

Operations:
Create object
Delete object
Read object
Write object

Addressing:
[object, byte range]

Allocation:
Internal

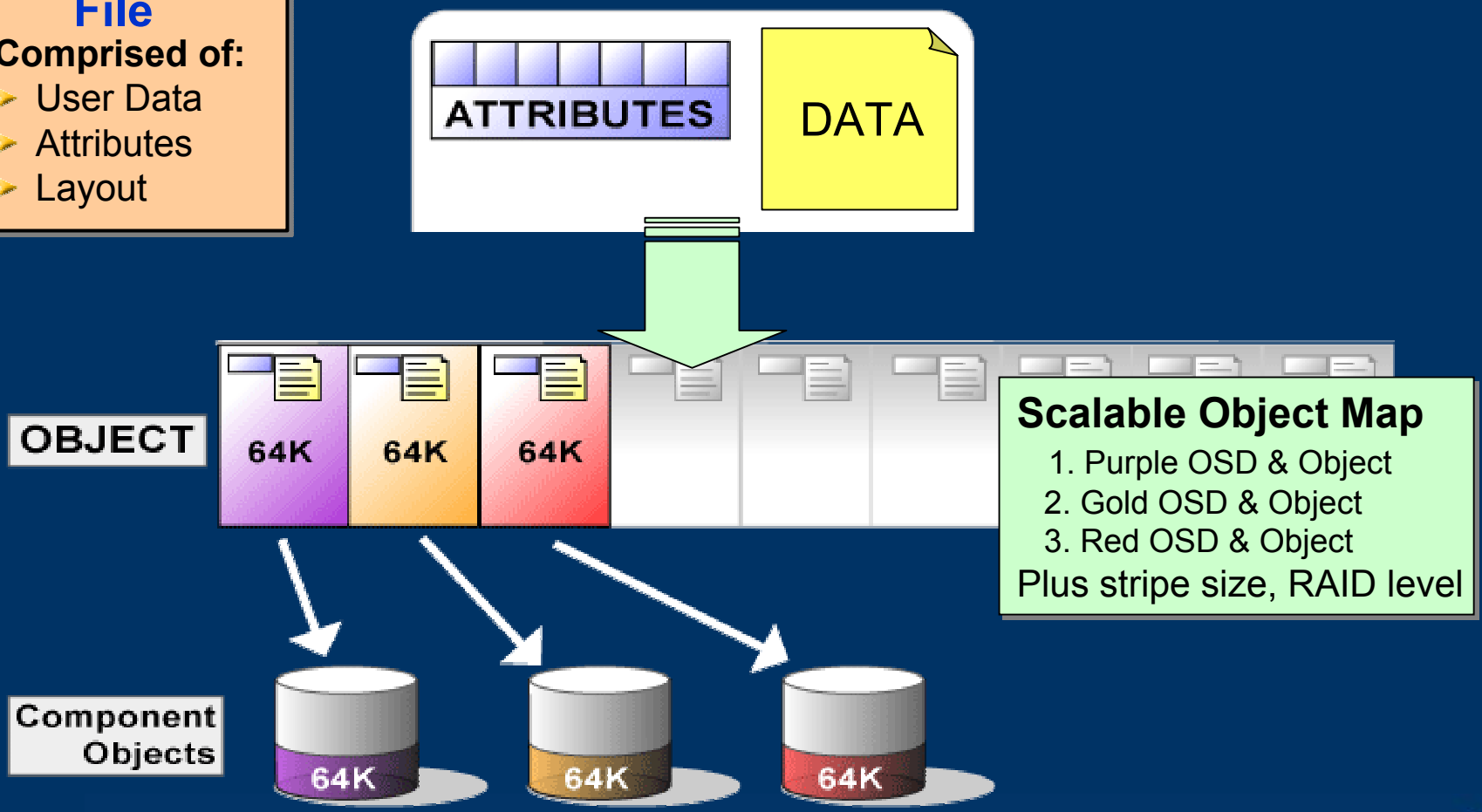


Source: Intel

How Does an Object Scale?

Scale capacity, bandwidth, reliability by striping according to small map

File
Comprised of:
- User Data
- Attributes
- Layout

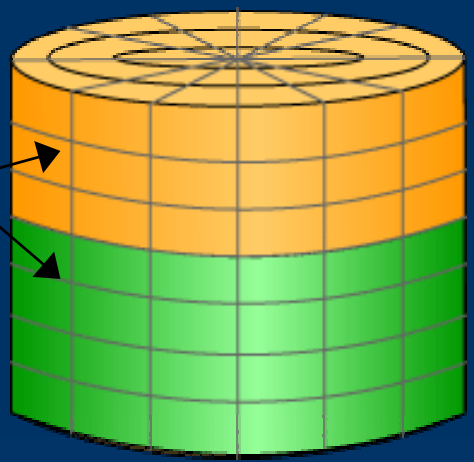


Additional Strengths Of Object Model

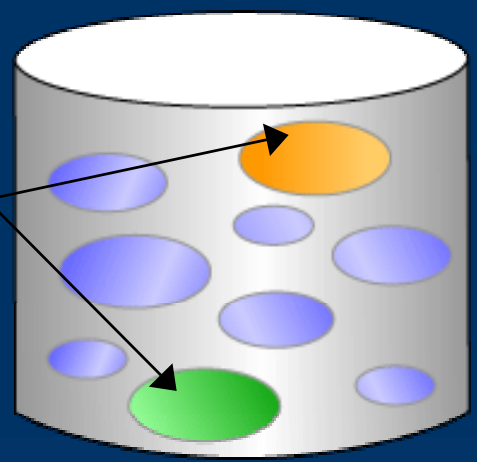
Disk knows relationships among data within object

- **Allows autonomous action within Object Storage Device**
  - Storage evolves to become self-managing, optimized to workload of each file
- **Rich sharing among heterogeneous clients via higher level interface**
- **Finer granularity of security: protect & manage one file at a time**

Security At Volume Level



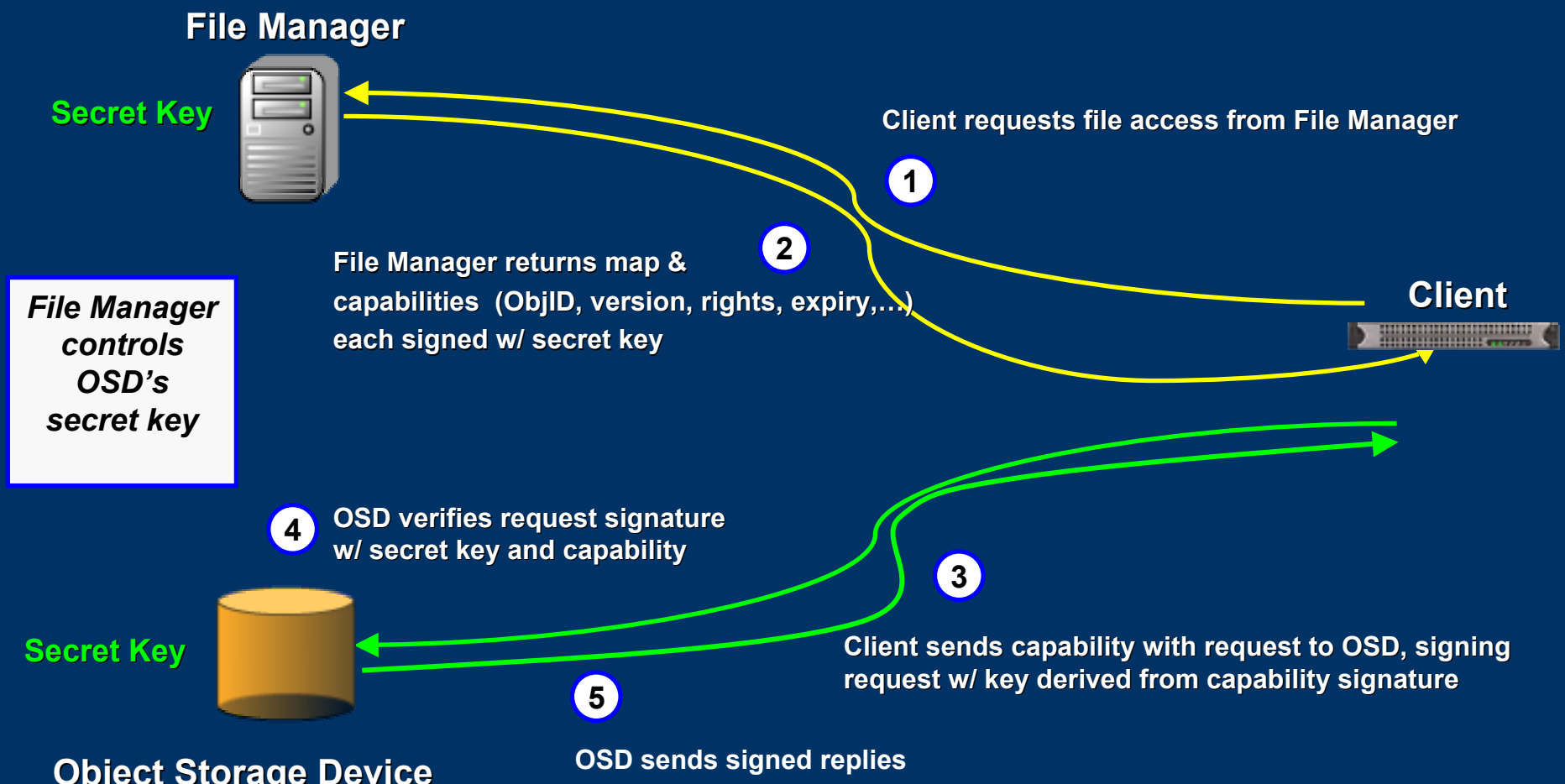
Security At Object Level



Source: Intel

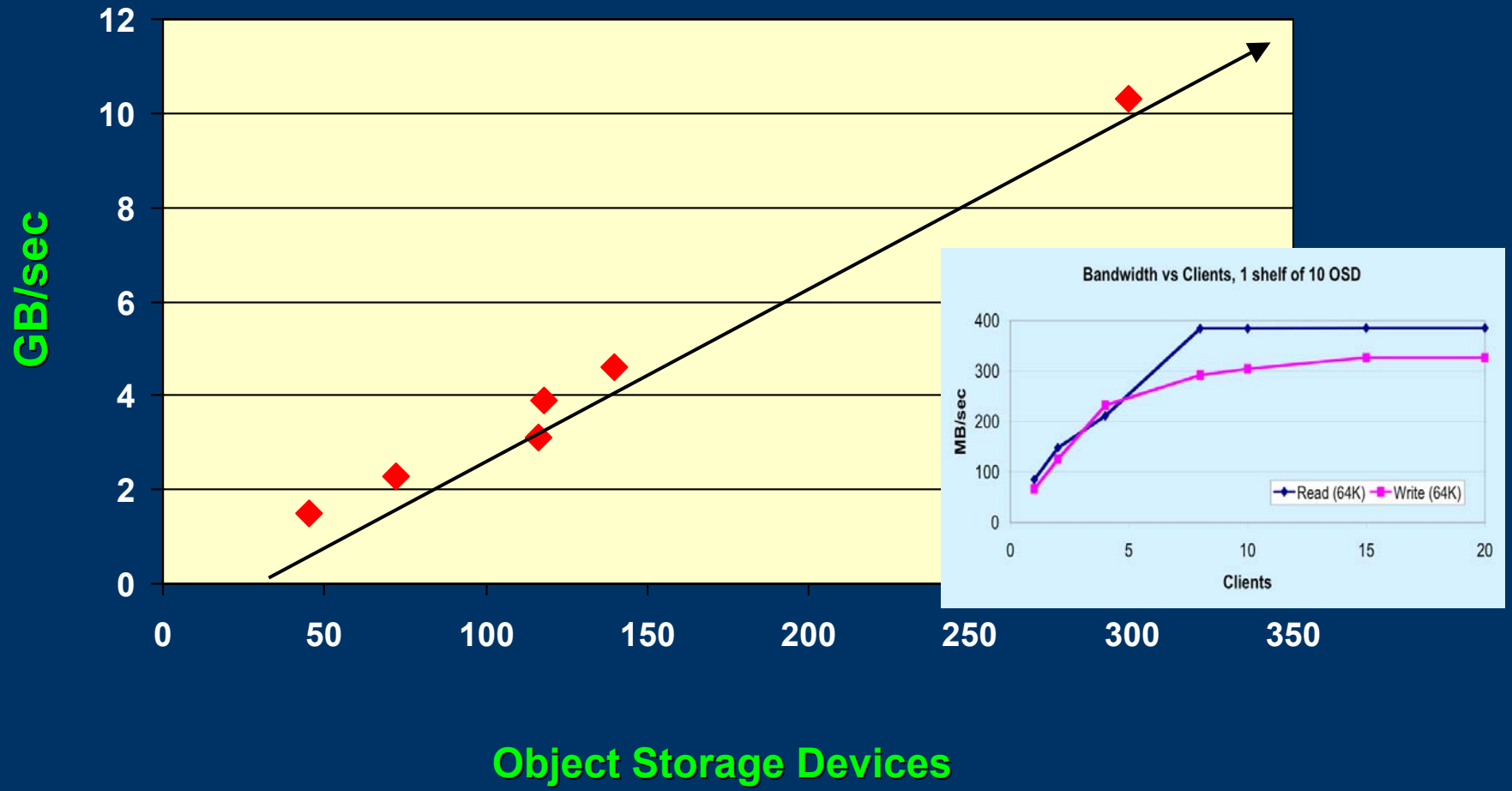
Object Storage Access Security

Digitally sign every request with evidence of access rights



Object Storage Bandwidth

Scalable Bandwidth demonstrated with GE switching



Object Storage Systems

Expect wide variety of Object Storage Devices

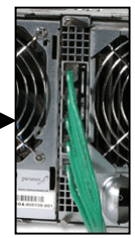
- Disk array subsystem
- Ie. LLNL with Lustre

- "Smart" disk for objects
- 2 SATA disks – 240/500 GB

- Prototype Seagate OSD
- Highly integrated, single disk

- Orchestrates system activity
- Balances objects across OSDs

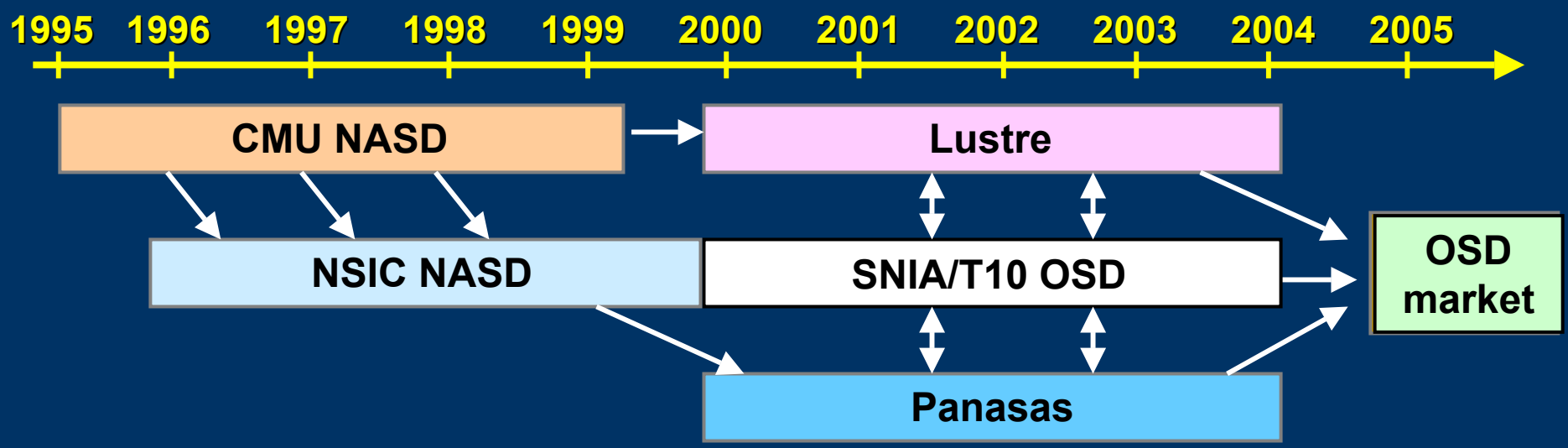
- **Stores up to 5 TBs per shelf**



16-Port GE Switch Blade

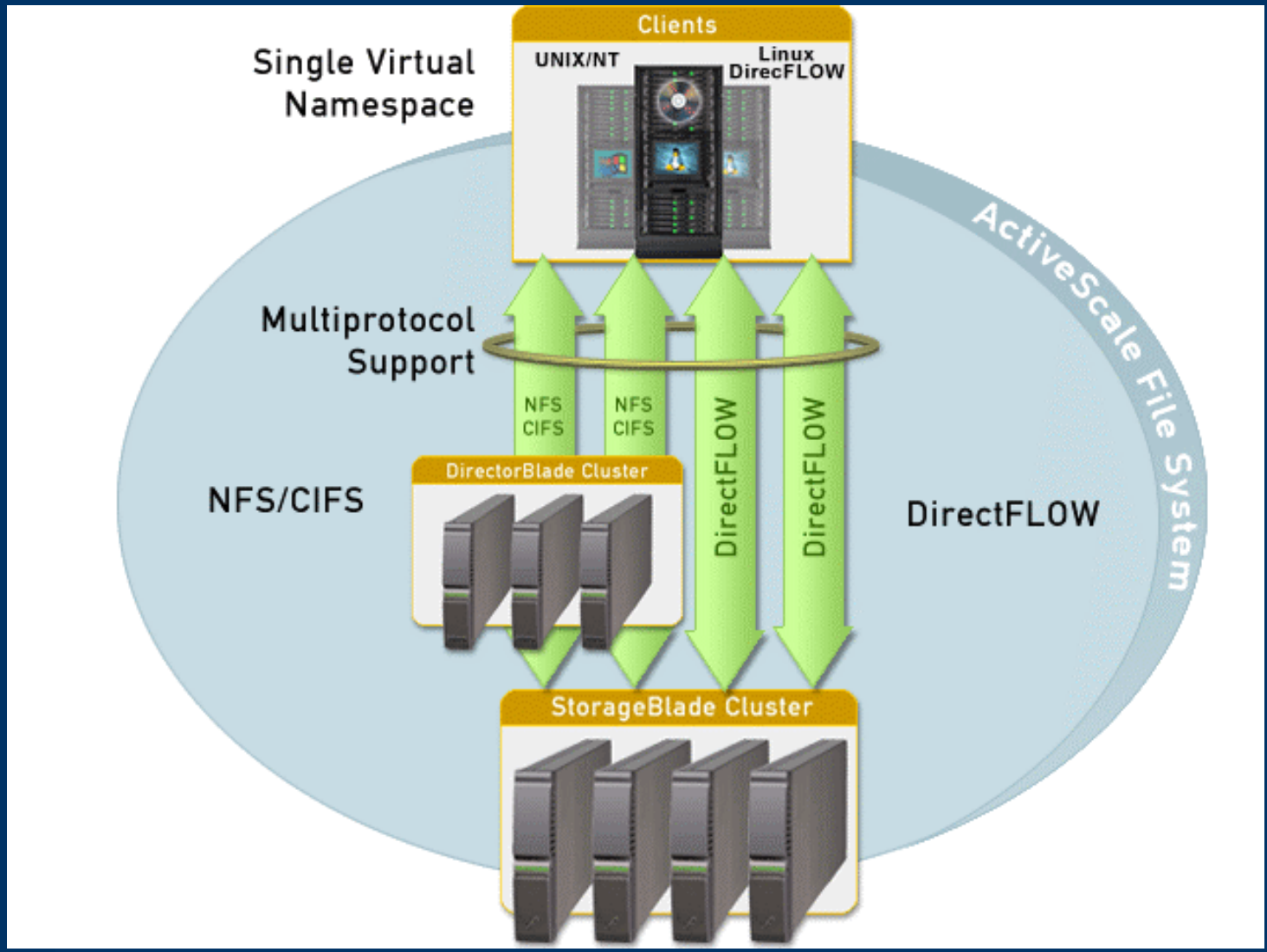
- 4 Gbps per shelf to cluster

Standardization Timeline



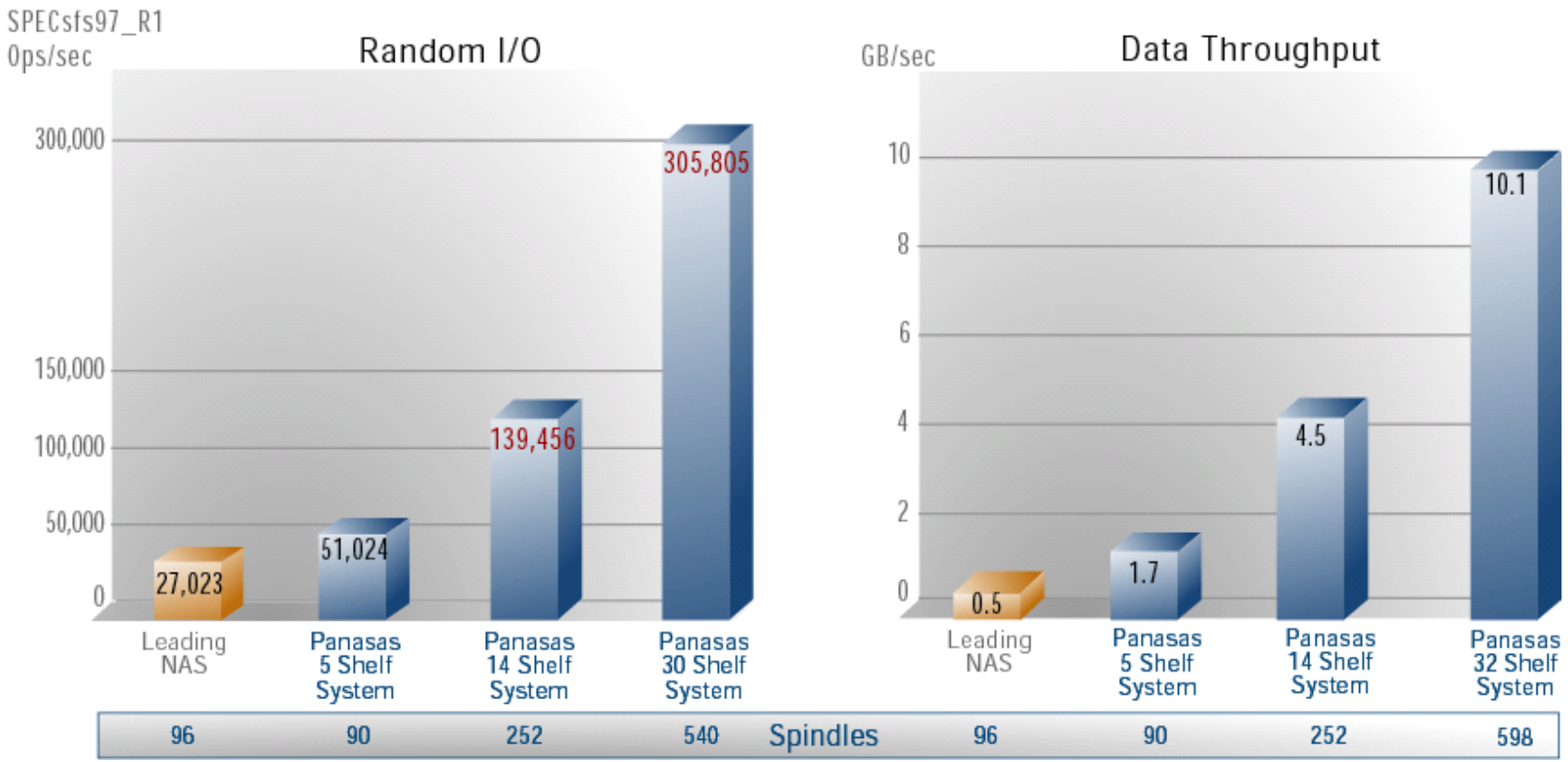
- **SNIA TWG has issued OSD protocol to T10 standards body**
  - Co-chaired by IBM and Intel, protocol is a general framework (transport independent)
  - Sub-committee leadership includes IBM, Seagate, Panasas, HP, Veritas, ENDL
  - Product commitment from HP/Lustre & Panasas; research projects at IBM, Seagate
  - T10 letter ballot (closed Mar 24 04) passed; integrating comments for full public review
  - www.snia.org/tech_activities/workgroups/osd & www.t10.org/ftp/t10/drafts/osd/osd-r09.pdf

Asymmetric Out-of-band & Clustered NAS



Performance & Scalability for all Workloads

Objects: breakthrough data throughput AND random I/O

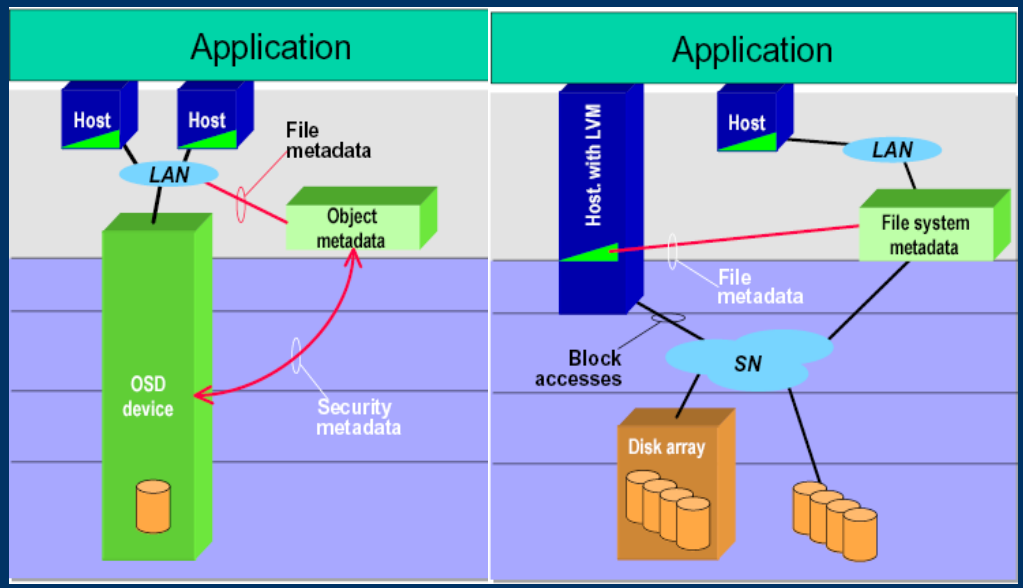


Source: SPEC.org & Panasas

Why I Like Objects

- **Objects are flexible, autonomous**
  - Variable length data with layout metadata encapsulated
    - Share one access control decision
    - Amortize object metadata
    - Defer allocation, smarter caching
  - With extensible attributes
    - E.g. size, timestamps, ACLs, +
    - Some updated inline by device
    - Extensible programming model

SNIA Shared Storage Model, 2001

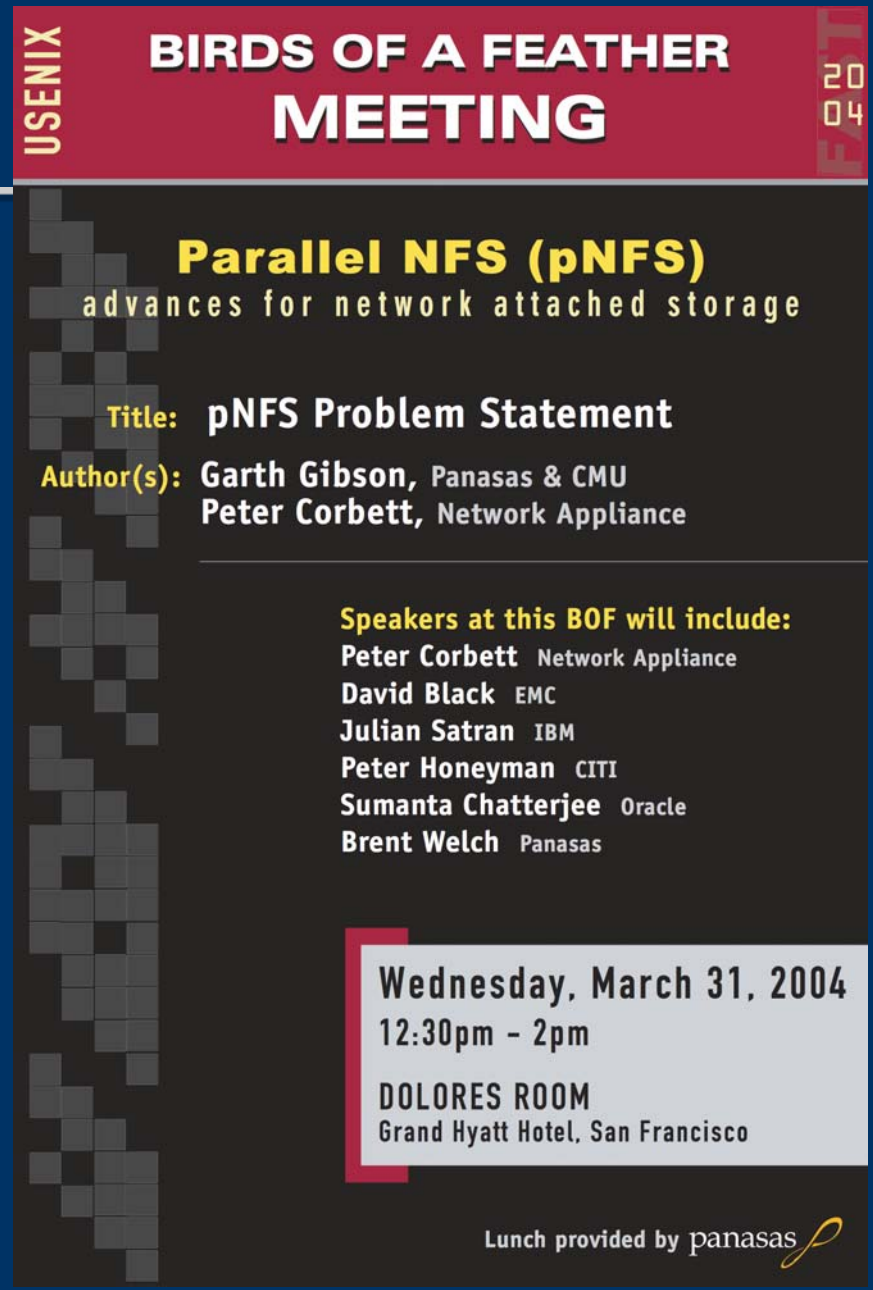


- Metadata server decisions are signed & cached at clients, enforced at device
  - Rights and object map small relative to block allocation map makes for big metadata caches
  - Clients can be untrusted b/c limited exposure of bugs & attacks (only authorized object data)
  - Cache decisions (& maps) replaced transparently -- dynamic remapping -- virtualization

pNFS

- **December 2003 U. Michigan workshop**
  - Whitepapers at www.citi.umich.edu/NEPS
  - U. Michigan CITI, EMC, IBM, Johns Hopkins, Network Appliance, Panasas, Spinnaker Networks, Veritas, ZForce
- **NFSv4 extensions enabling clients to access data in parallel storage devices of multiple standard flavors**
  - Parallel NFS request routing / file virtualization
  - Extend block or object "layout" information to clients, enabling parallel direct access
- **Problem Statement Internet-Draft published**

  draft-gibson-pnfs-problem-statement-00.txt



panasas

Out-of-band Value Proposition

- Out-of-band means client uses more than one storage address for a given file, directory or closely linked set of files

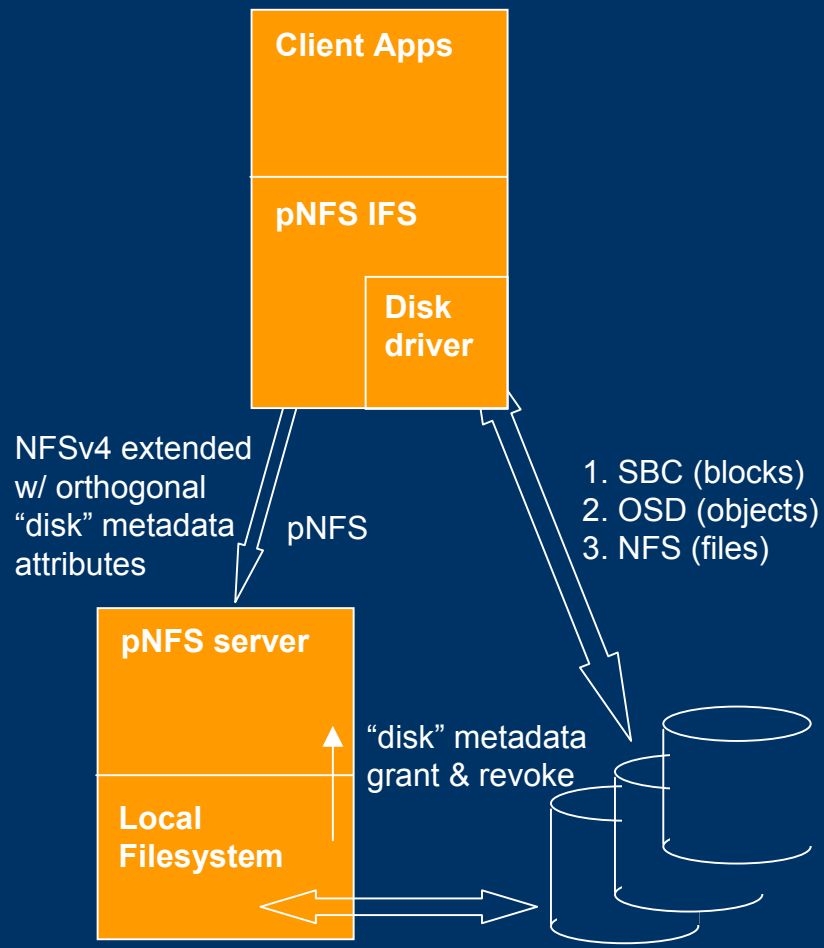
- **Scalable capacity:** file/dir uses space on all storage: can get big
- **Capacity balancing:** file/dir uses space on all storage: evenly
- **Load balancing:** dynamic access to file/dir over all storage: evenly
- **Scalable bandwidth:** dynamic access to file/dir over all storage: big
- **Lower latency under load:** no bottleneck developing deep queues
- **Cost-effectiveness at scale:** use streamlined storage servers
- If wire standards lead to **standard client SW:** share client support $$$

NFS Extensions Approach

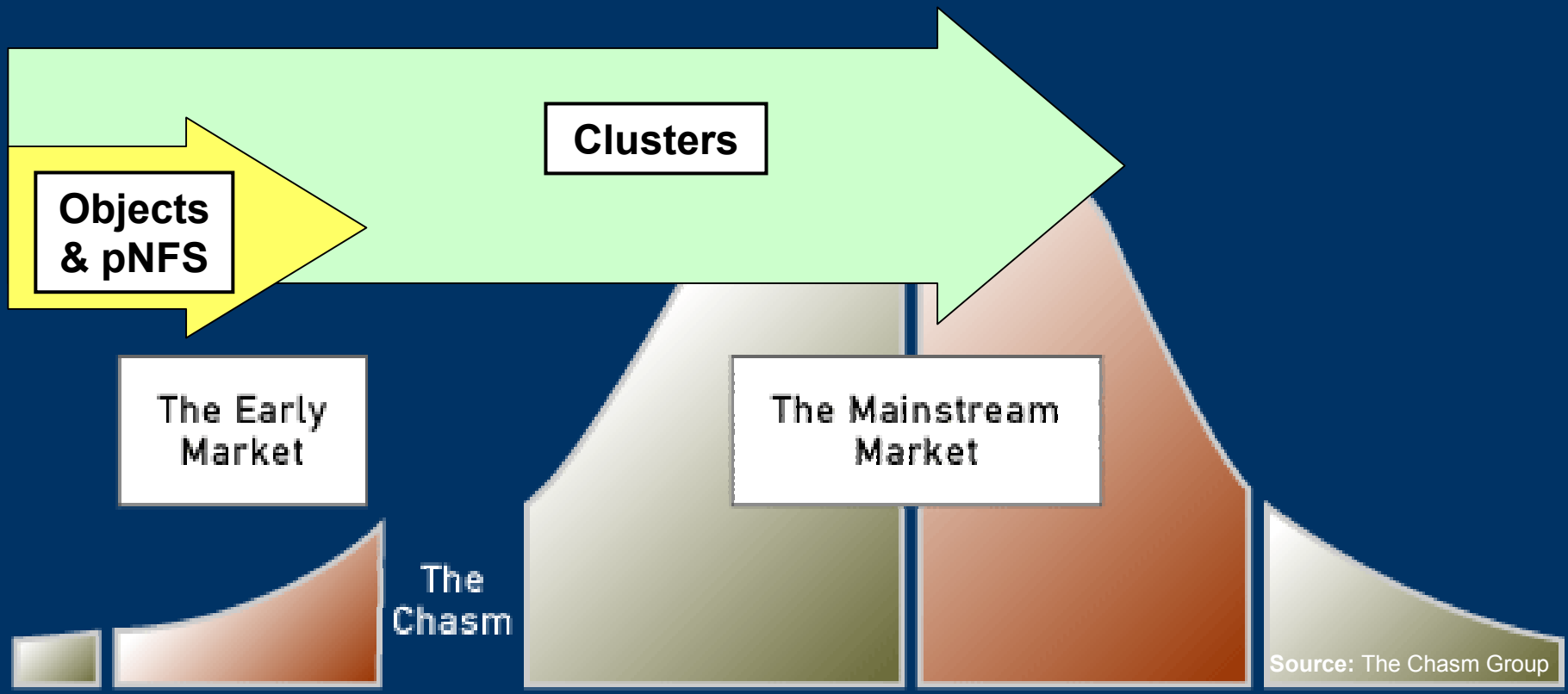
- **Limited (Market)/(support $) for proprietary advanced FS client SW**
  - Customers: interoperating competition; vendors: too many changing client platforms
- **Rally behind one open industry-standard advanced FS client SW**
- **IETF NFS is unrivalled as open industry-standard FS client SW**
  - Raising (Market)/(support $) is worth giving up proprietary feature control
- **V4: "Recallable delegations allow clients holding a delegation to locally make many decisions normally made by the server"**
- **Propose a new flavor of NFSv4 delegations**
  - A client requesting a delegation asks for out-of-band file address maps
  - Server protects map integrity while delegation lasts, knowing file data may change
  - Server can re-synch with file contents by recalling the delegations

Multiple Data Server Protocols

- **Inclusiveness favors success**
- **Three (or more) flavors of out-of-band metadata attributes:**
  - BLOCKS:
    SBC/FCP/FC or SBC/iSCSI…
    for files built on blocks
  - OBJECTS:
    OSD/iSCSI/TCP/IP/GE for files built on objects
  - FILES:
    NFS/ONCRPC/TCP/IP/GE for files built on subfiles
- **Inode-level encapsulation in server and client code**



Crossing the Chasm



Cluster storage for scalable Linux clusters

Garth Gibson
ggibson@panasas.com
www.panasas.com

3rd USENIX Conf on File and Storage Technologies, March 31, 2004