pNFS:
Extend NFSv4 for Parallel Storage

Garth Gibson, Benny Halevy, Brent Welch
Panasas Inc., www.panasas.com

NEPS Workshop, Dec 4, 2003, Ann Arbor, MI

December 4, 2003

The big picture, or, Why are we here?

- **Improve (Market)/(support $) for proprietary advanced FS client SW**
  - Customers: no competitive, interoperable market; vendors: client platform $$$$$
- **Rally behind one open industry-standard advanced FS client SW**
  - Customer acceptance up and vendor support costs for client SW down
- **IETF NFS is unrivalled as open industry-standard FS client SW**
  - Is raising (Market)/(support $) worth giving up proprietary feature control?
- **NFSv4 a big step “advanced” relative to v3**
  - Delegations, kerberos, ACLs, named attributes, failover locations
  - Extensibility
- **Are there a few extensions that would make it worth getting started?**
  - Understanding, from NFS IETF mailing list lurking, that other enhancements are being considered, roadmapped, evolved (e.g. richer delegations).
  - Direct client access per file/dir to multiple storage addrs using SBC, OSD & NFS?
- **Shall we standardize advanced FS client SW? In IETF NFS forum?**

“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
- Wire standards led to **standard client SW**: share client support $$$

Delegations for File Address Maps

- **“Recallable delegations allow clients holding a delegation to locally make many decisions normally made by the server”**
- **Propose that when using delegations**
  - A client requesting a delegation asks for out-of-band file address maps
  - Server protects integrity of maps while delegation lasts, and understands file data may change out-of-band
  - Server can re-synch with file contents by recalling the delegations
- **File address map, logically parts of inode & data pointers**
  - For OSD objects, Panasas uses list of device address, object id, capability, striping parameters, RAID parameters
- **Protocol support in addition to delegation consistency & recovery**
  - What storage systems can a client access
  - When file address map is huge, get in pieces
  - For allocating new space during writing, e.g. begin-allocate & end-writing
  - Requesting changes in the map itself (wider striping, replication, etc)

Multiple Data Server Protocols

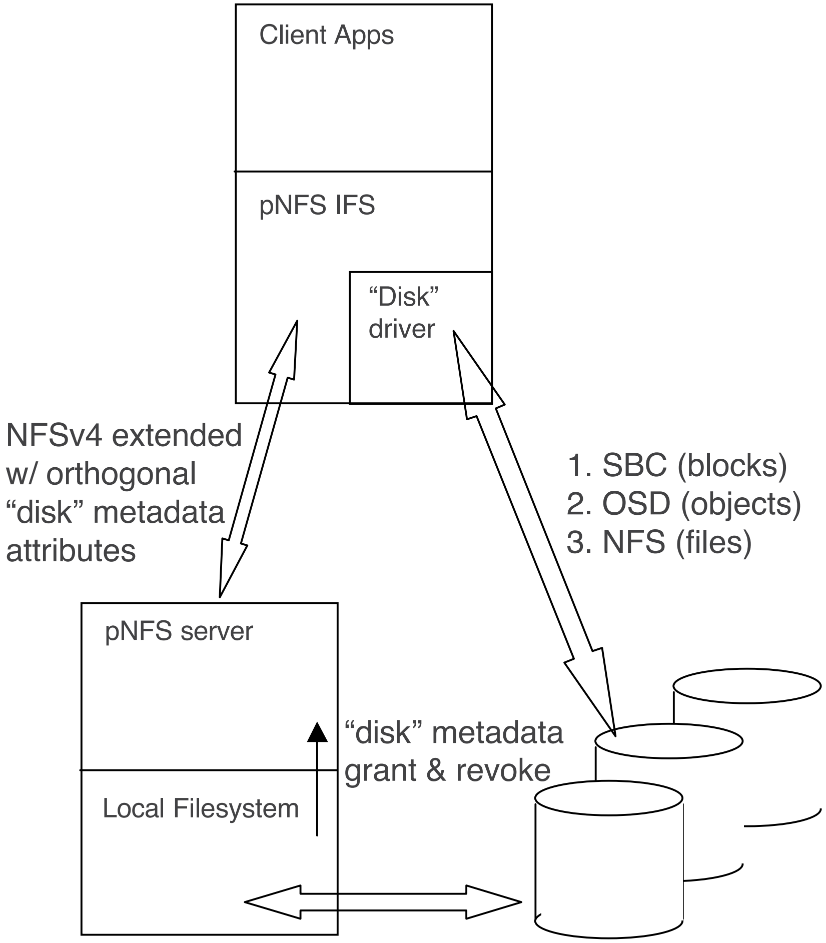
BE INCLUSIVE !!

- Broaden the market reach

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



Recommended Principles

- **Orthogonal and complimentary to transport improvements (RDMA)**
- **Start with NFSv4 and stay as close as performance allows**
  - Maybe a roadmap of use cases where specialized workloads benefit from more extensive changes -- should collaborate closely with core NFSv4 team
- **At any time all operations can be completed through server**
  - Make all direct actions idempotent; error recovery by retry against server
  - Concurrent sharing can be simply handled through server
  - Legacy support and simple allocation
- **NFS extentions for control & consistency of metadata, not meaning**
  - Separate docs (per storage type) describe wire format of metadata
  - While only describing wire format, achieve “principles of client function”
- **Clients negotiate ability to use and type of direct access (discovery)**

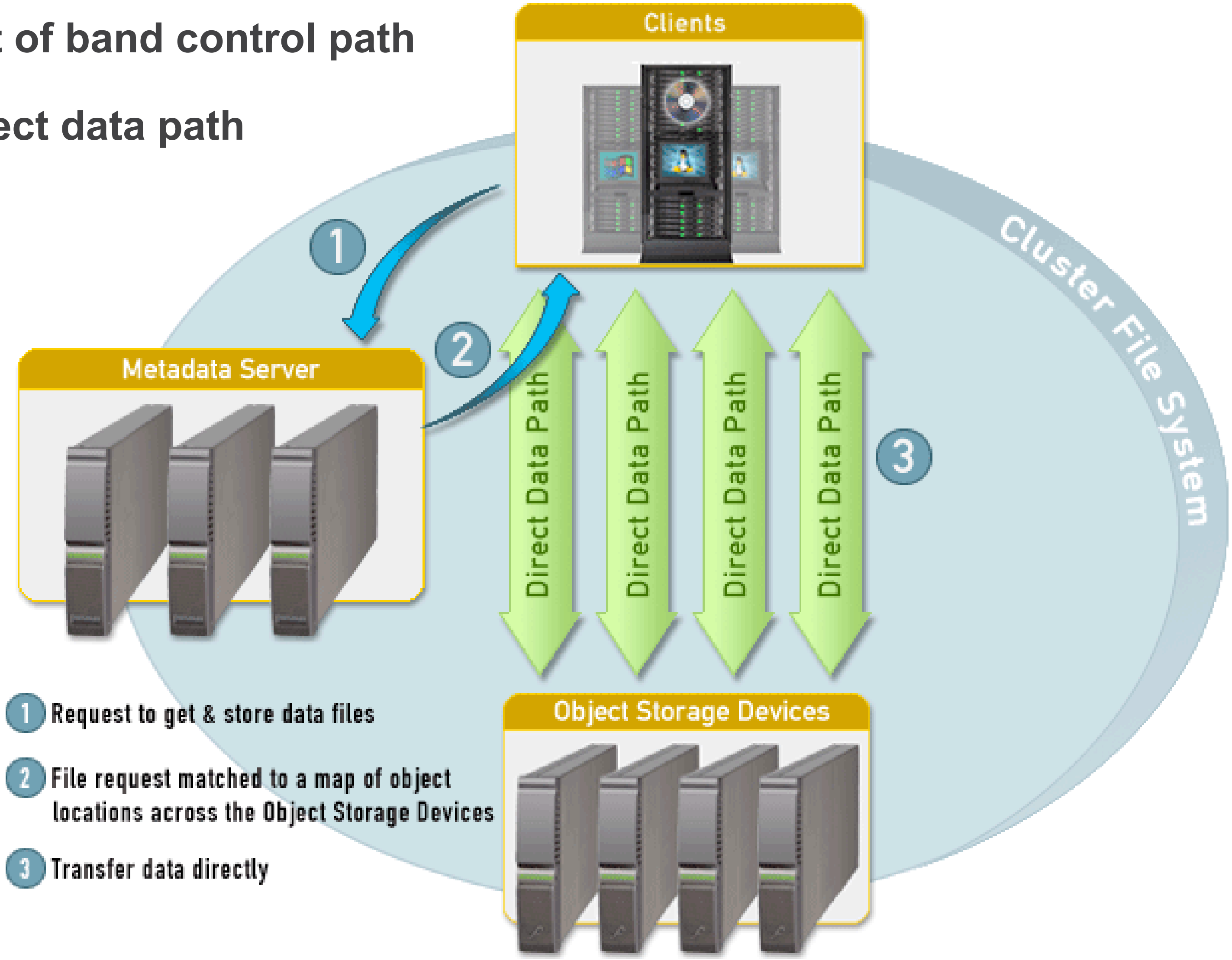
Panasas’ Object Storage:

Redefining Bandwidth for Linux Clusters

December 4, 2003

Object Storage Data Path

- Out of band control path
- Direct data path



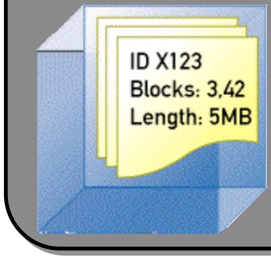
1 Request to get & store data files

2 File request matched to a map of object locations across the Object Storage Devices

3 Transfer data directly

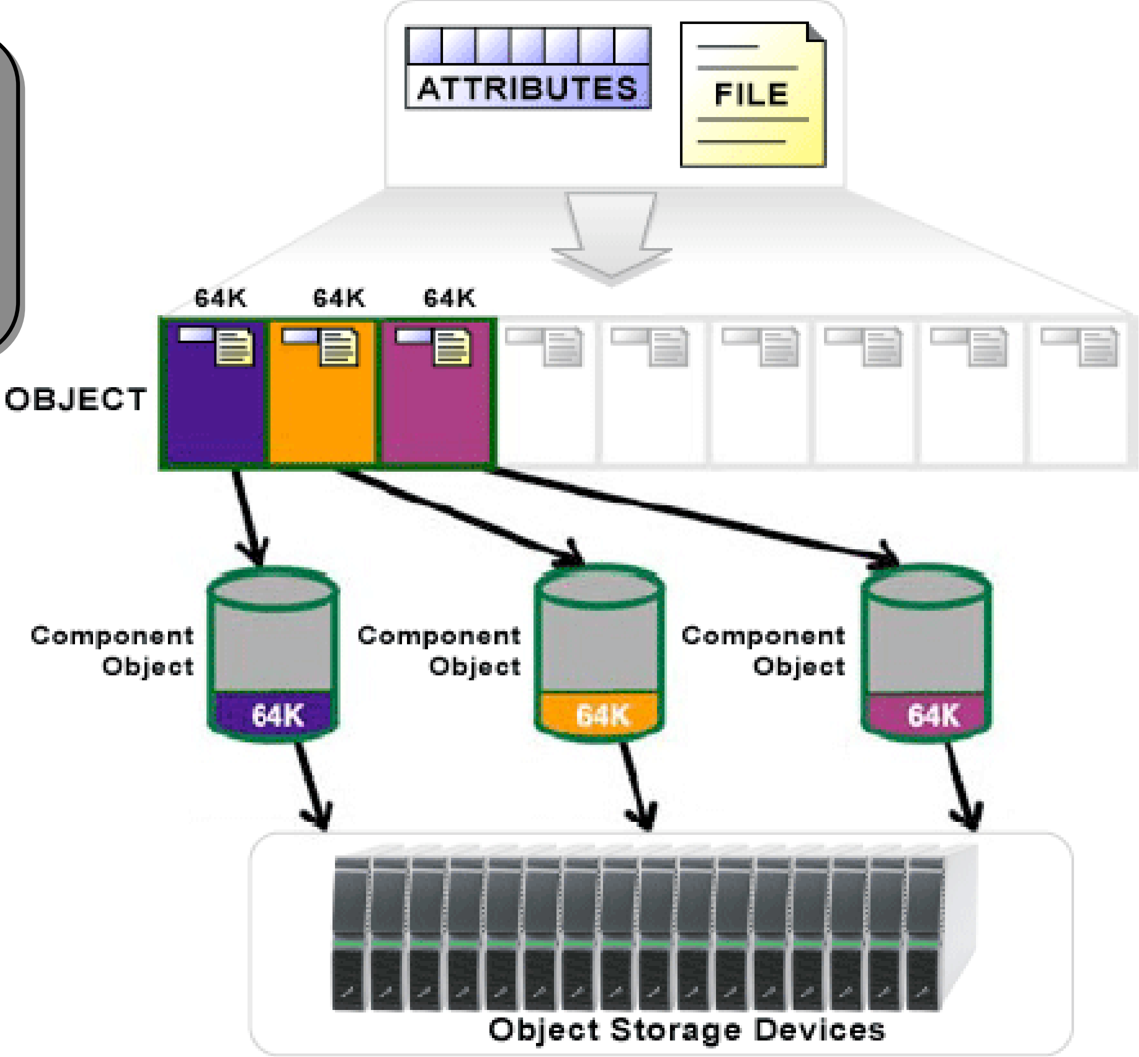
What is an Object?

Object



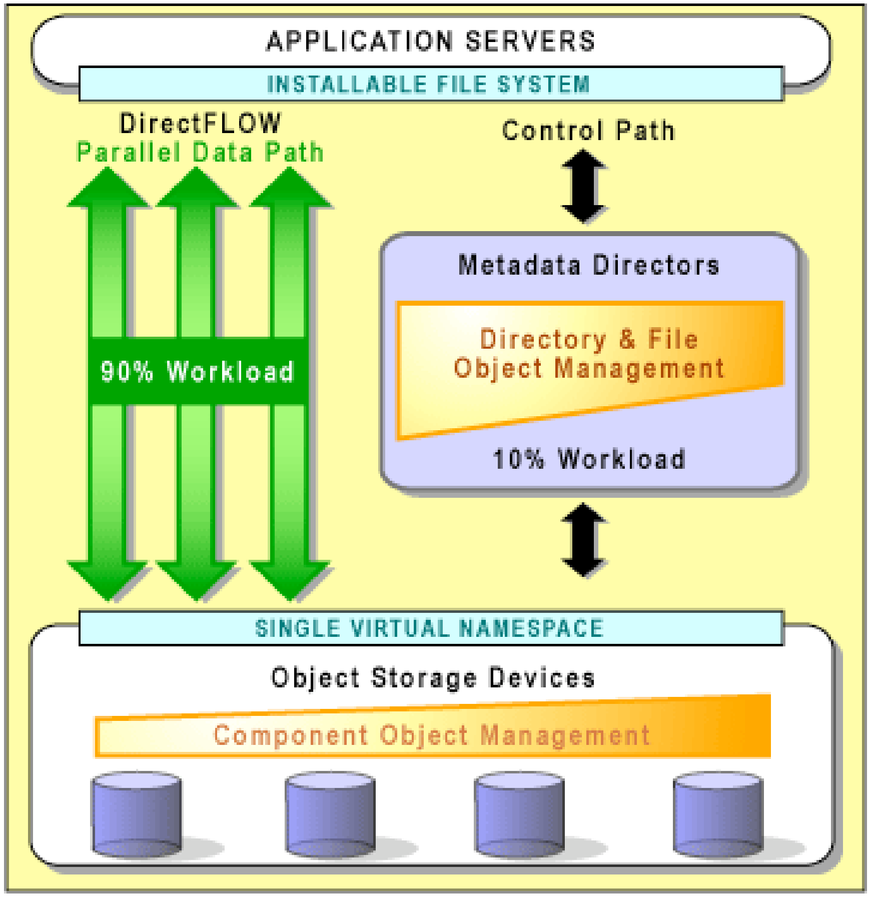
Comprised of:

- User Data
- Attributes
- Layout



Object Storage System Architecture

- **Moves low-level storage functions into the storage device itself**



Key Object Storage Features

Intelligent space management in storage layer

- Media geometry aware placement
- Late binding allocation
- Data aware prefetching, caching & recovery

Encapsulation of data and attributes

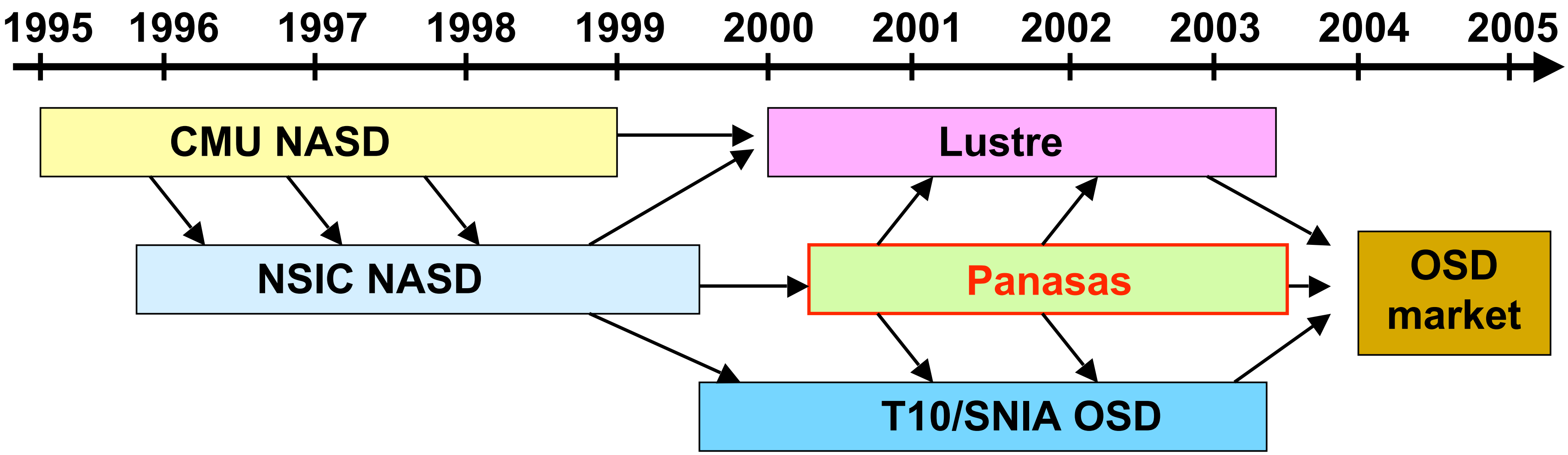
- Native object interface, good programming model
- Storage interpreted attributes for per file properties

Key Object Storage Advantages

- Robust, shared access by many clients
- Scalable performance via an offloaded data path
- Strong fine-grained end-to-end security

panasas

Standardization Timeline



- **SNIA TWG is nearing completion of proposed OSD standard**
  - Great participation by leading storage industry vendors
  - SNIA OSD V1 draft sent for review/ratification to ANSI T10 OSD committee
  - Next steps for OSD standards is under development
    - Roadmap includes SMIS & Information Life Cycle management support

hp invent IBM intel. EMC² where information lives Seagate We turn on ideas VERITAS

Object Storage Systems

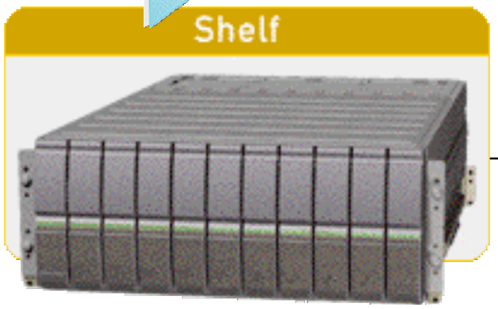
Wide variety of Object Storage Devices

- Disk array subsystem
- Used with Lustre

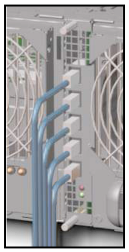
- Smart disk holding objects
- Panasas StorageBlade uses Serial ATA disks for up to 500 GB

- OSD research at Seagate
- Highly integrated, single disk

- Orchestrates system activity
- Balances objects across Object Storage Devices



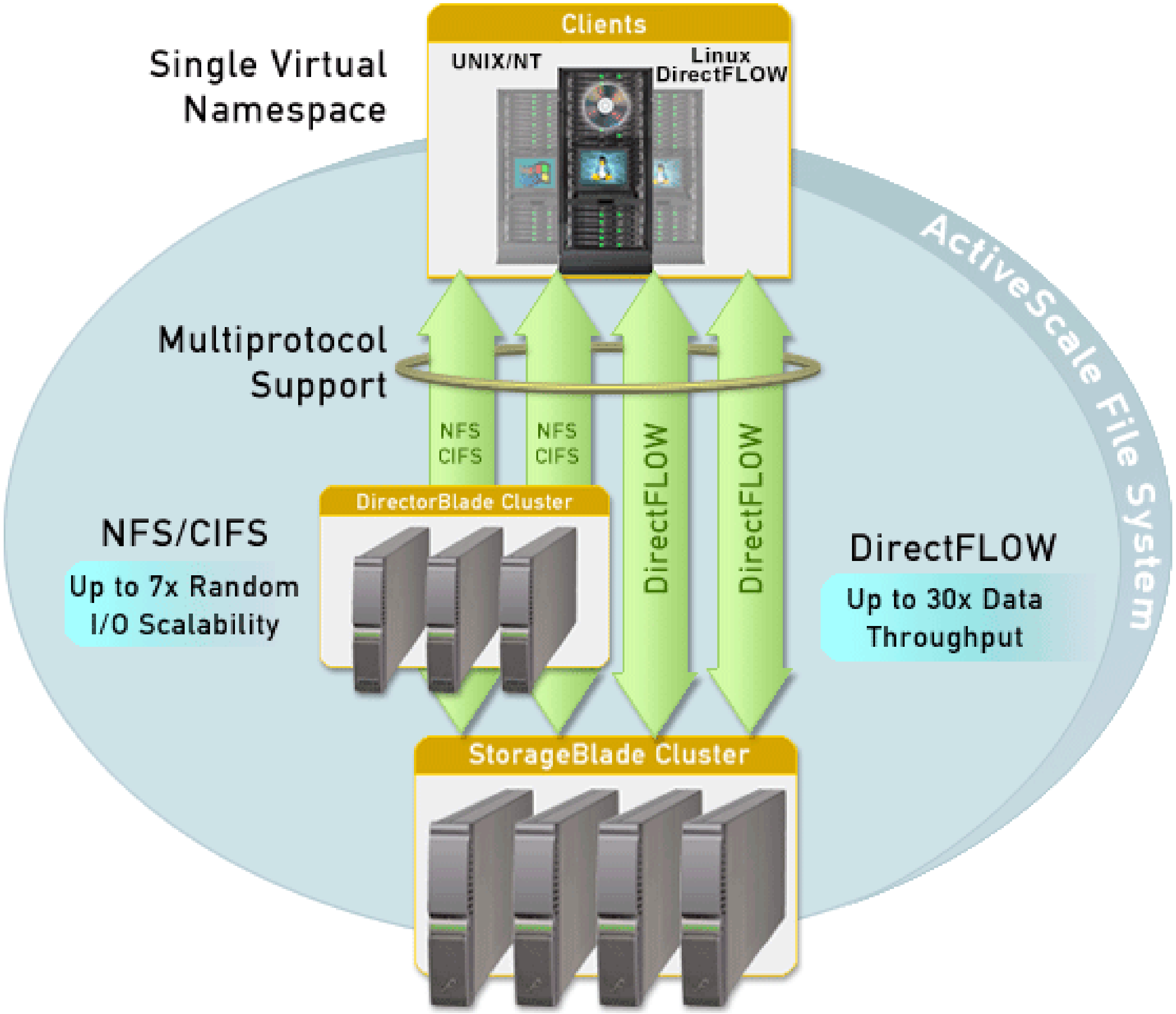
- Stores up to 5 TBs per shelf
- Battery-backed redundant power



16-Port GE Switch Blade

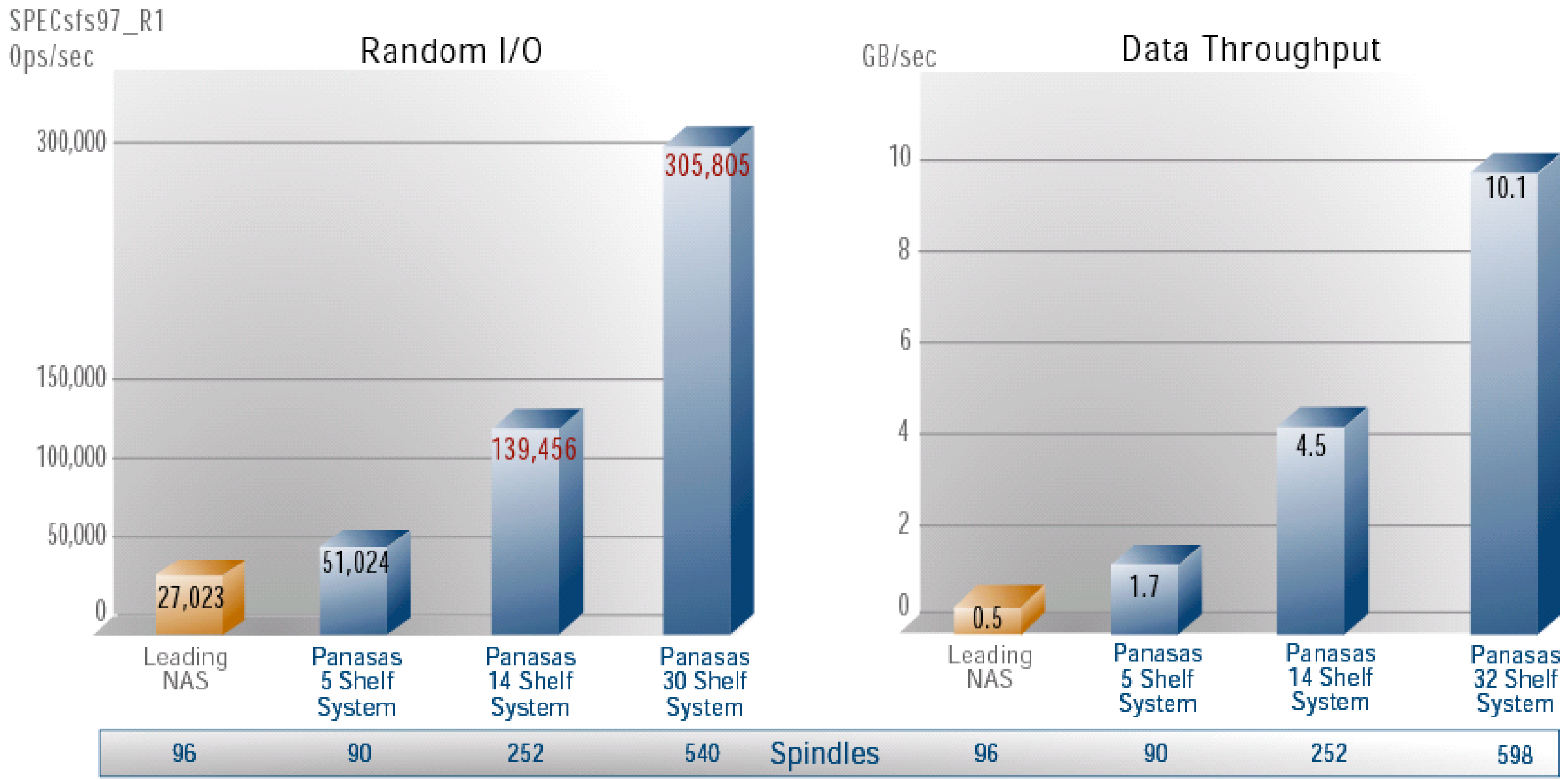
- 4 Gb/sec per shelf to Linux cluster

Full Function Storage Cluster



Objects: Performance & Scalability

Breakthrough Data Throughput AND Random I/O



Object Storage:

Redefining Bandwidth for Linux Clusters

December 4, 2003