
# Object Technology Panel

Garth Gibson
CTO and co-founder
CMU professor, on leave

MSST'03, April 9, 2003

panasas

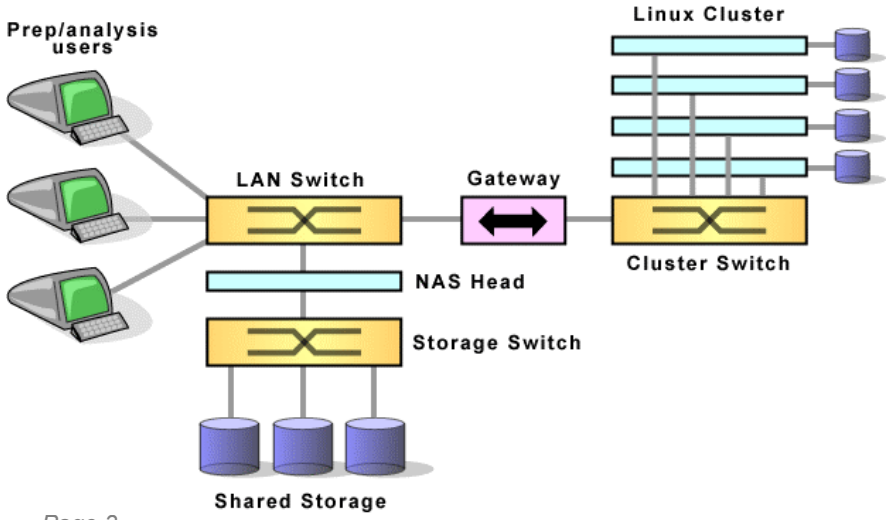
# Storage Objects

- **Science community suffers from bleeding on the edge**
  - Bandwidth, scale, cost-effectiveness requirements often 10+ X commercial
  - 1984 SCSI storage architectures make it hard to meet these requirements
    - Making science expensive, repeated modifications to new commercial generation
  - Need architectural changes that make science solutions easy to support
    - Need out-of-band -- Need extensible -- Can't be useless for non-science
- **Objects are familiar -- basically Inode layer of traditional file system**
  - Directory/naming, authentication/authorization, cache consistency separate
  - File layout, file attributes, RAID, media under the interface
  - Bandwidth and scale from exposing a map of inodes
  - Cost-effectiveness by charting path to commodity implementations

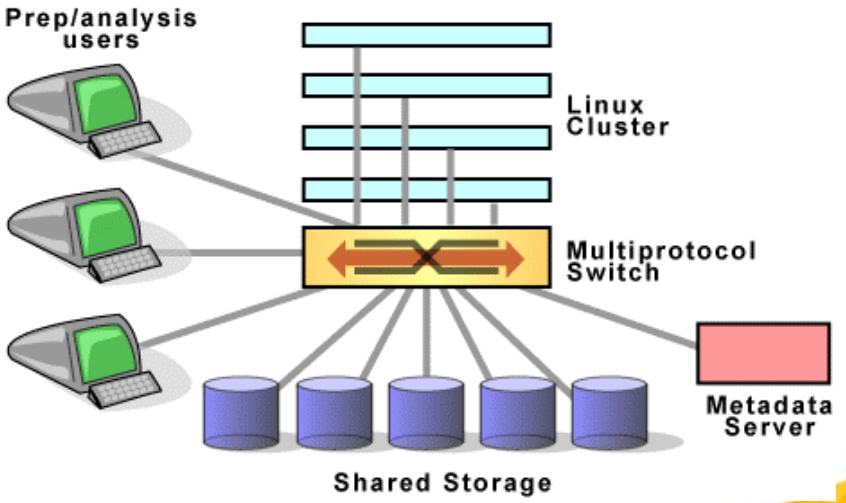
# Scale Architecturally

- **Direct parallel access compute cluster to share-nothing storage cluster**
  - Out-of-band, asynchronous, rarely used metadata service (but not end user, please)
- **Disk per node scales, but turns compute cluster into adhoc MPP fileserver**
  - Need external shared storage repository anyway
  - Scalable cluster networking enables “dial-your-own” storage bandwidth
  - Multiprotocol switching/bridging for specialization of compute and storage network

**Traditional Scalable Bandwidth Cluster**

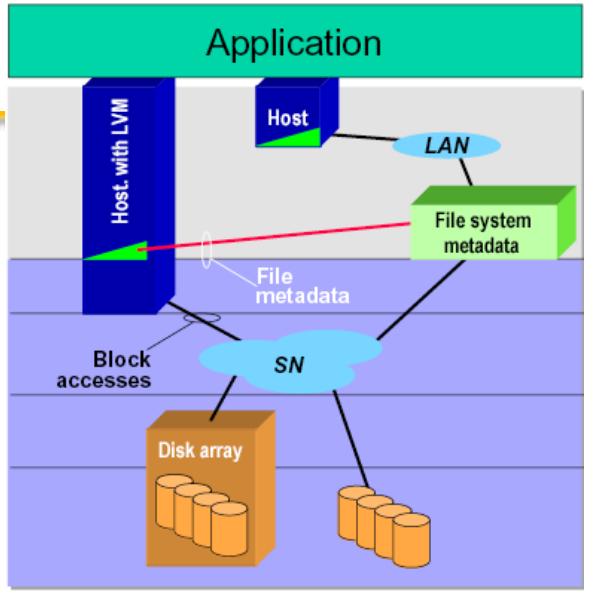



**Out-of-Band Scalable Bandwidth Cluster**

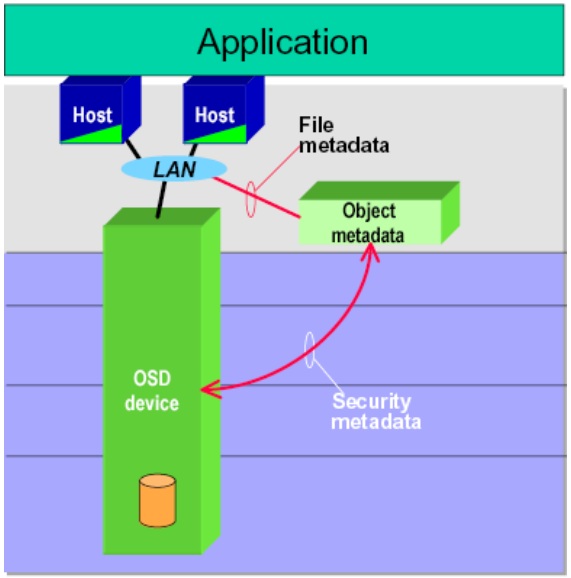


# Controlled Scalable Bandwidth

- **Out-of-band, asymmetric data vs control**
  - Expose client parallelism to device parallelism
    - High bandwidth & high concurrency
    - Offload metadata service & reduced queuing
    - Specialize networking for metadata vs data
  - Block-based with FC or iSCSI/GE
    - Give clients table of pointers in all-storage-address space
  - Object-based with OSD/iSCSI or OST/portals
    - Give a client a list of inodes and limit access to this list
  - Elimination of bottlenecks eliminates convenient serialization, allocation, authorization
    - Metadata service provides these or offloads to either client (block-based) or device (object-based)
      - More done at client or device yields more scalability
    - Firewall in server is also bypassed, so vpn or physical security needed

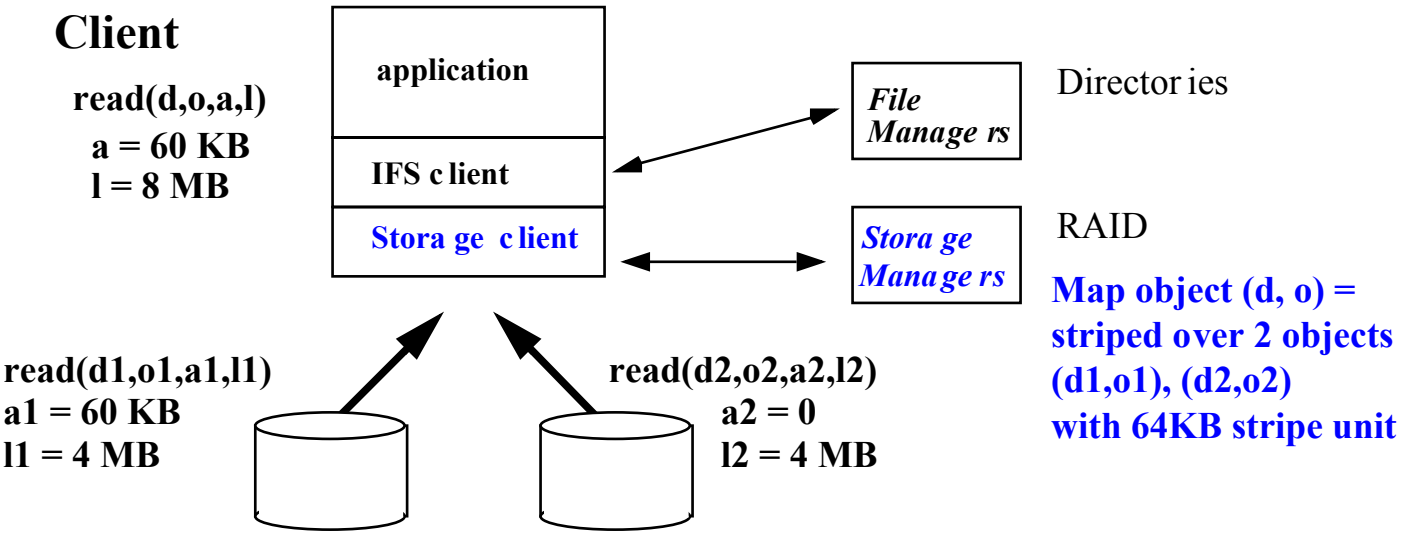



SNIA Shared Storage Model 2001



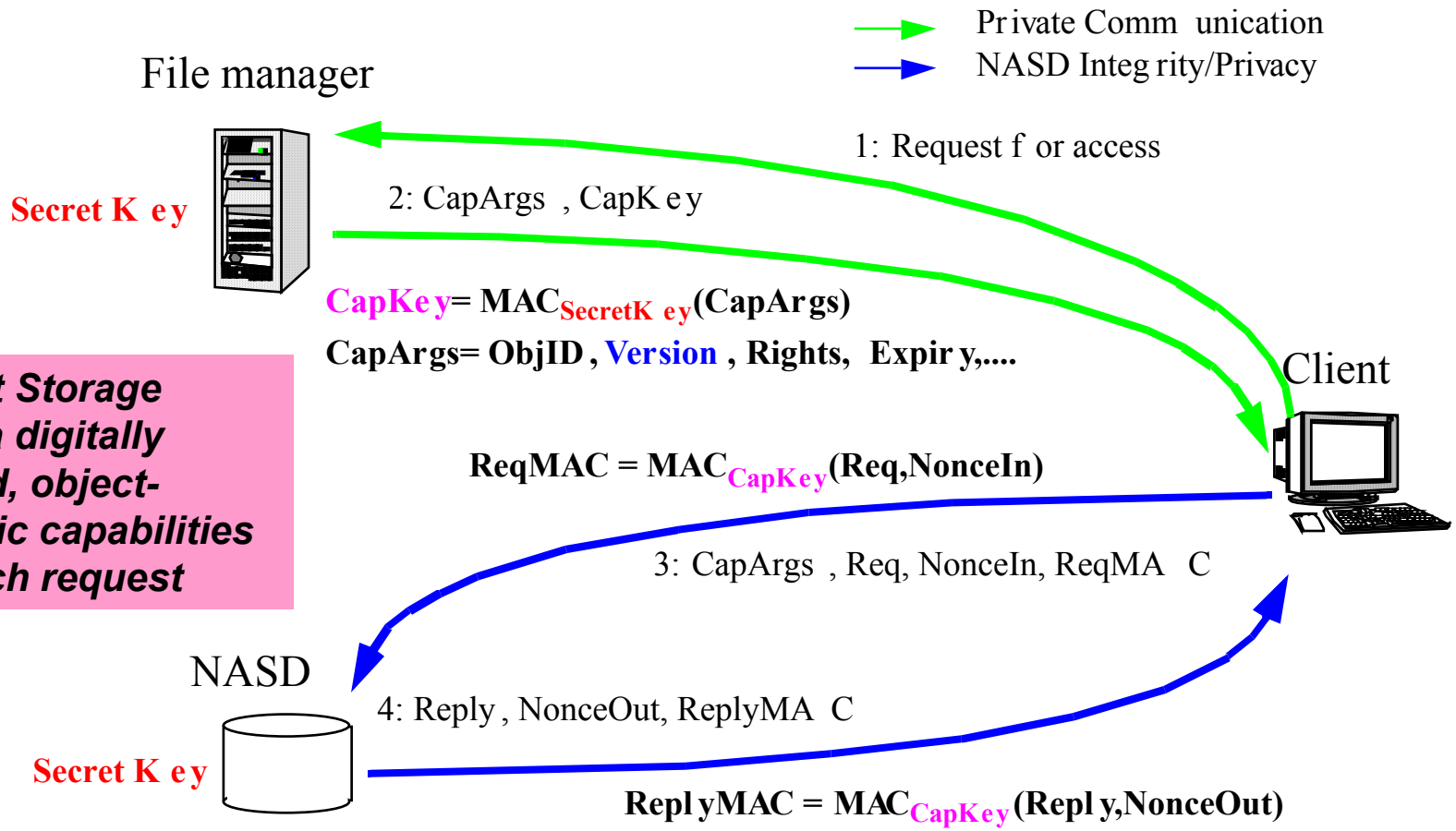
# Compact Virtualization: Maps

- **Striping/RAID representation should be dynamic, file-specific**
  - Update key management attributes (timestamps, size) inband
  - Escrow capacity, defer allocation to object for fast path decision efficiency
  - Cache coherent, on-the-fly remapping for balancing and incremental growth
  - Embed representation in object attributes, extensible for QoStorage specification



# Fine Grain Access Enforcement

- **State of art is VPN of all out-of-band clients, all sharable data and metadata**
  - Accident prone & vulnerable to subverted client; analogy to single-address space computing




***Object Storage uses a digitally signed, object-specific capabilities on each request***

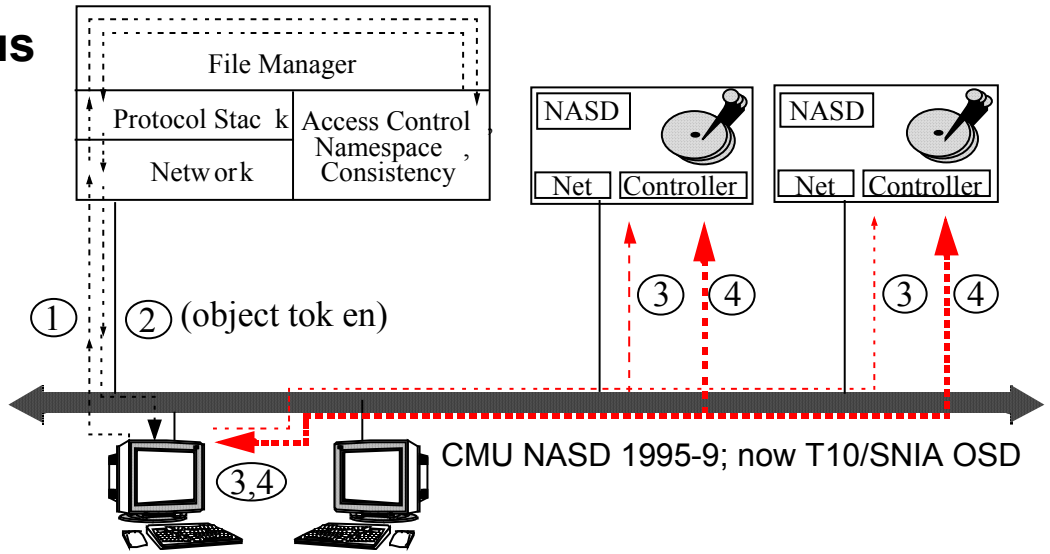
# Scaling Metadata Service

- **Command processing of most operations in storage could offload 90% of small file/productivity workload from servers**
  - Key inband attribute updates: size, timestamps etc

| NFS Operation | Count in top 2% by work (K) | File Server (SAD) | | DMA (NetSCSI) | | Object (NASD) | |
|---|---|---|---|---|---|---|---|
| | | Cycles (B) | % of SAD | Cycles (B) | % of SAD | Cycles (B) | % of SAD |
| Attr Read | 792.7 | 26.4 | 11.8 | 26.4 | 11.8 | 0.0 | 0.0 |
| Attr Write | 10.0 | 0.6 | 0.3 | 0.6 | 0.3 | 0.6 | 0.3 |
| Data Read | 803.2 | 70.4 | 31.6 | 26.8 | 12.0 | 0.0 | 0.0 |
| Data Write | 228.4 | 43.2 | 19.4 | 7.6 | 3.4 | 0.0 | 0.0 |
| Dir Read | 1577.2 | 79.1 | 35.5 | 79.1 | 35.5 | 0.0 | 0.0 |
| Dir RW | 28.7 | 2.3 | 1.0 | 2.3 | 1.0 | 2.3 | 1.0 |
| Delete Write | 7.0 | 0.9 | 0.4 | 0.9 | 0.4 | 0.9 | 0.4 |
| Open | 95.2 | 0.0 | 0.0 | 0.0 | 0.0 | 12.2 | 5.5 |
| Total | 3542.4 | 223.1 | 100 | 143.9 | 64.5 | 16.1 | 7.2 |

# Why I like OSD Objects

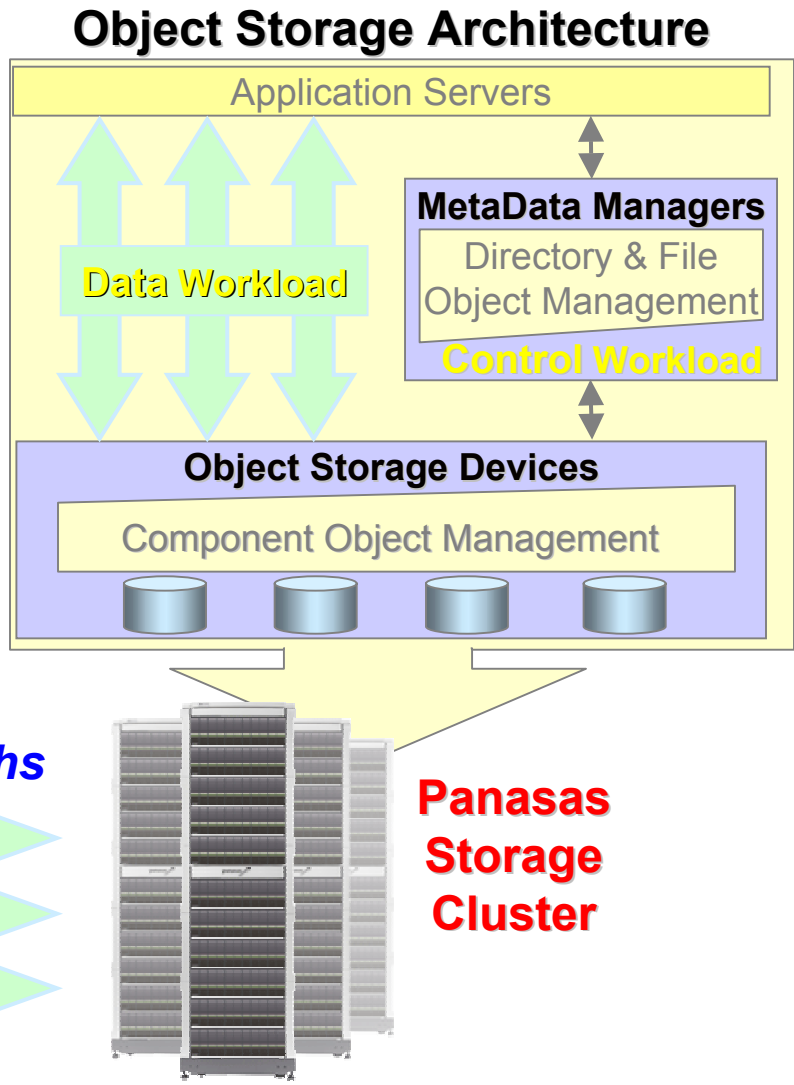
- **Objects are flexible, autonomous**
  - Variable length data with layout metadata encapsulated at device
  - With extensible attributes
    - E.g. size, timestamps, ACLs, +
    - Some updated inline by device
    - Big enough to amortize object metadata
    - Small enough to share one access control decision




  - Metadata server decisions are signed and cached at clients, enforced at device
    - Rights and object map small relative to block allocation map
    - Clients can be untrusted (bugs & attacks expose only authorized object data)
    - Cache decisions (& maps) replaced transparently -- dynamic remapping -- virtualization
  - OSD is command set that works with SCSI architecture model (SAM)
    - Encourages cost-effective implementation by storage device vendors

# Panasas Network Storage

- ***Object-based Intelligent Storage***
  - Seamless global namespace
  - 10X scalable performance !
  - Dynamic clustering
  - Appliance-like management
  - Low-cost commodity building blocks




**Linux Compute Cluster**

***Parallel Data Paths***





**Panasas Storage Cluster**