Network Attached Secure Disks (NASD)

A proposal to NSIC for collaborative storage interface research

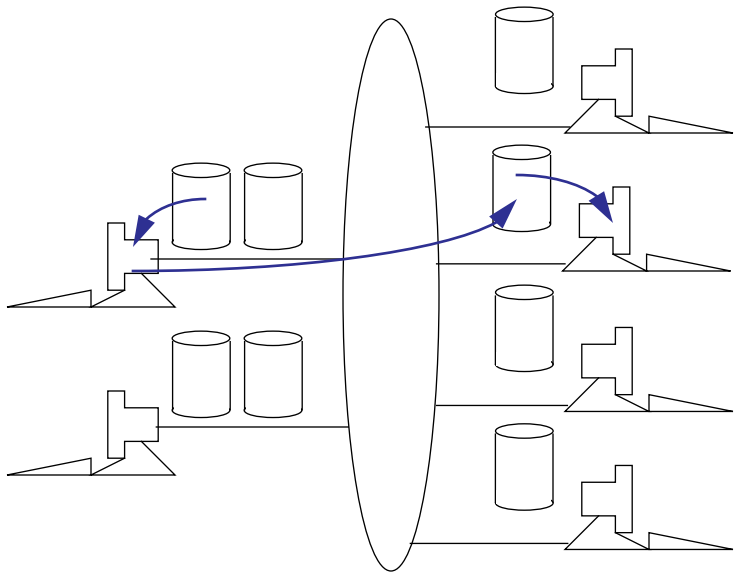
Garth Gibson, CMU's Parallel Data Lab
garth.gibson@cs.cmu.edu
http://www.cs.cmu.edu/Web/Groups/PDL/

- **Storage Architecture Trends**
  - **Excessive Copying in Network Storage Architecture**
  - **Disk and Network Bandwidth Converging**
  - **Increasing Storage-embedded Function**
- **Network-Attached Storage Alternatives**

Trends: Data bytes travel over LANs

Workstation a poor and costly server

- designed around caching for processor-local work
- open physical system bus standard slows advances
- network bandwidth limited to approx single disk bandwidth
- induces extra copying



Trends: Migration to serial drive interface

Drive data rate rising rapidly

- linear bit density up 20% per year
- spindle RPM doubled in 4 years
- new partial response, maximum likelihood encodings

SCSI physical interconnect under stress

- electrically noisy; large connectors; limited addressability

Switching to serial interconnects

- Firewire - desktop system bus
- SSA - 20 MB/s SCSI packet ring
- Fibrechannel - 120MB/s, multi-protocol, packet ring/switch

• Are storage interconnects good LANs?

Trends: Growth in drive-embedded functionality

Fast-moving, profit-tight marketplace

VLSI gives more cycles & bytes at constant price

- controllers in 2000 expected to have 200 MHz processors
- compete with value-added intelligence for performance

Disk scheduling

- use SCSI queue tags, only place for geometry-senstive detail

Readahead/writebehind

- plenty of systems have OSs with poor caching

RAID support

- offload parity update computation

Cost of managing storage per year 7X storage cost

- dynamic adaptation to workload
- selection and migration through redundancy schemes
- support for backup

Dynamic mapping for transparent optimizations

- log-structured for write (and read) bandwidith
- log-structured for large write optimization
- floating parity or parity logging
- cache and media compression

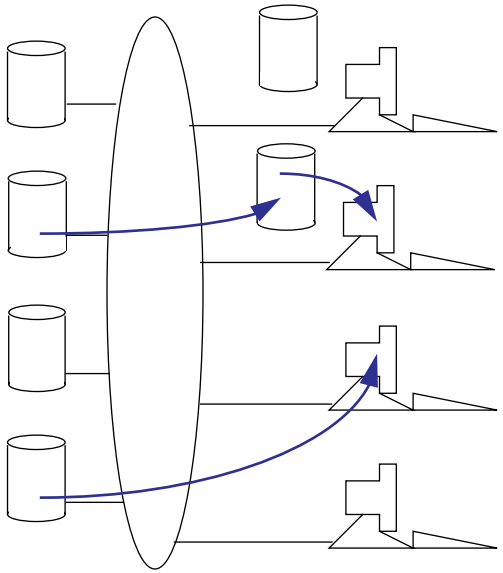
• SCSI storage abstraction too low level ?

Alternative: Network attached storage

Attach storage directly to network

- fewer copies and appropriate bandwidth
- addressability for drive-to-drive transfer

Streamline LAN to storage performance standard



Alternative: Raise storage functional interface

< object, offset, length >

- better readahead/writebehind and caching
- at-storage allocation management (dynamic mapping)
- object specific handling (compression, availability)

Richer functional interface

- hint for access patterns, allocation
- priority levels for interspersing small & large accesses
- admission-controlled isochronous transfers

Alternative: Repartition file system model

Out-source device specific control from file system

- retain namespace, access list interpretation, coherence model
- similar to client/server network file system partition

Extensible, client library to parallelize storage

- network RAID, parallel program support
- dynamic availability, load balancing
- client filesystem code and client machine untrusted

Alternative: Integrate drive into LAN security protocol

Arbitrary packets must not be seen as commands

- private key cryptography based on drive serial number
- tamper-resistant encryption for authentication check
- authentication server enables access to drive by passing drive a UID and session key

May improve system security model

- no logins at drive (SNMP style interface for administration)
- user configurable encryption over net or on media
- encryption over net requirements pay for authentication

CMU PDL/DSSC Plans

Focus on drive embodiment

- far enough out for academic research

Cooperation

- NSIC interface pre-standards working group

Starting model of repartitioned file system

- Scotch Parallel File System

Controlling cost

- thin protocol stack on-drive; off-drive protocol emulation

Drive architecture

- drive system bus for caching, streaming, encrypt, parity, ...

Summary

Network-attached Secure Disks (NASD)

- **“first class network citizens”**
- **direct data transfer to client**
- **eliminate workstation file server**
- **raise drive interface to file system**
- **integrate with authentication-based security**
- **repartition file system between client and drive**
- **support multiple client filesystem personalities**
- **optimizedprotocol processing, emulation for interoperation**
- **cost-constrained, transfer-oriented drive architecture**