NASD: A Cost-Effective, High-Bandwidth Storage Architecture

Garth Gibson, David Nagle, Khalil Amiri, Jeff Butler, Fay Chang, Howard Gobioff, Charles Hardin, Erik Riedel, David Rochberg, Jim Zelenka

Computer Science and Computer Engineering, CMU

Responding to data rate improvements

- disk data rate averaging 40% faster each year
- fastest drive in 1998: 27.5 MB/s internally
- peripheral interconnect at 100 MB/s and rising

Sponsored by DARPA/ITO Quorum/Scalable Systems
and HP, Quantum, Seagate, STK, Symbios, Clariion, Compaq, Wind River, Intel, 3Com

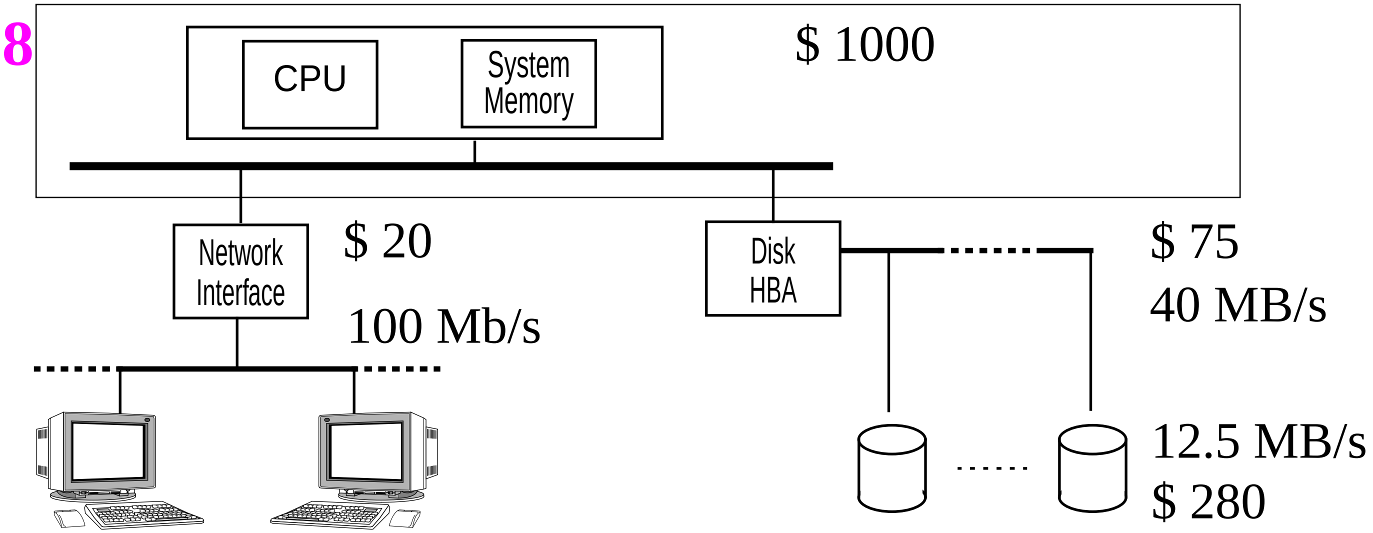
Server-Attached Disks (SAD) don’t deliver bandwidth

Cheap server machine, fast ether, UltraWide SCSI

- one net, one drive with **server overhead cost of > 390**
- AMORTIZE low-cost server with more drives
  5 drive, 5 NICs, 2 HBAs (7 PCI slots?)   **> 89%**

Store-and-forward copying **doubles** storage cost

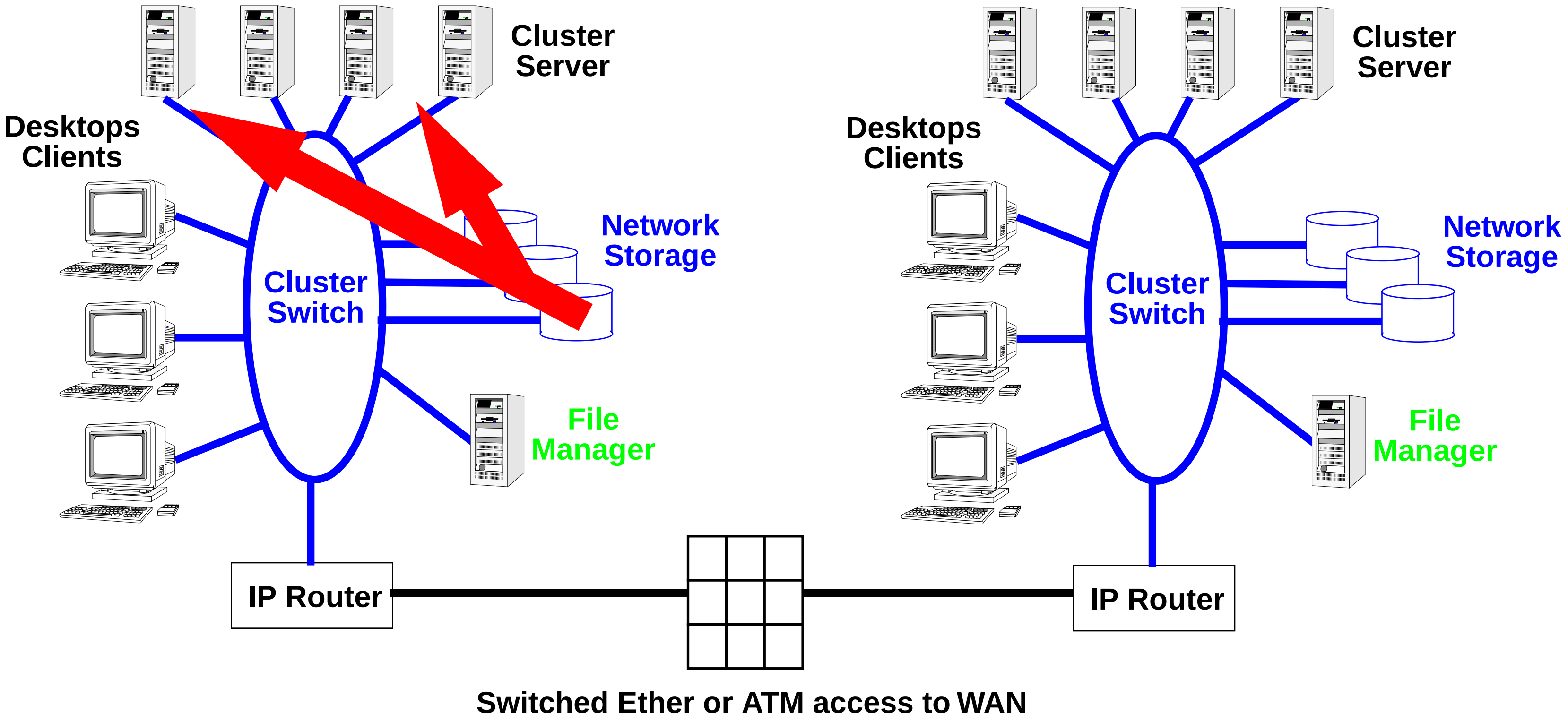
10/98



Take file server off datapath: 3rd party transfer

Direct transfer between client and storage

- exploit scalable switched cluster area networking
- split file service: into primitives (in drive), policy (in manager)



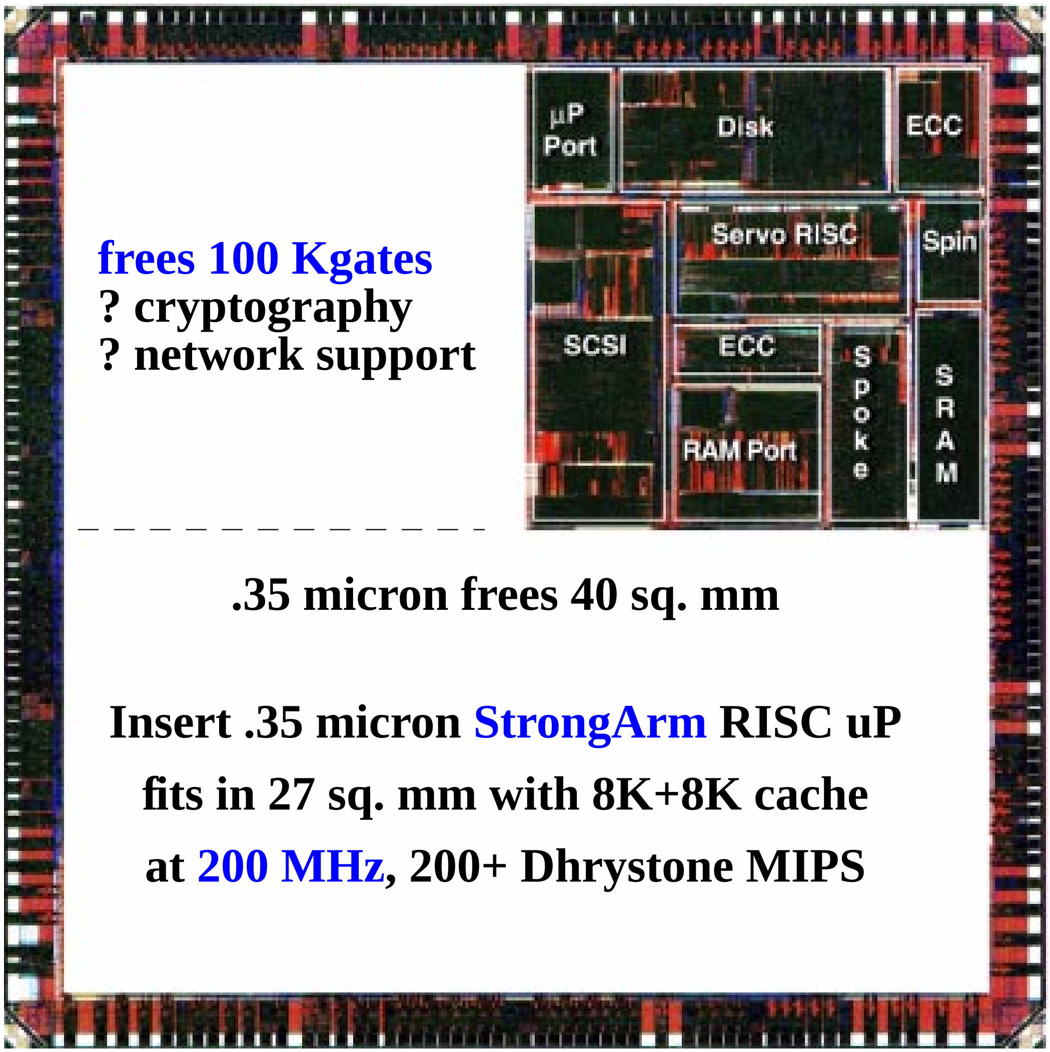
Device cycles are available now

Quantum Trident drive

- **Control: M68020**
- **Datapath ASIC** →
- **.68 micron in 1997**
- **4 indep clock domains, each 40 MHz**
  SCSI processor
  disk R/W channel
  uP control port
  DRAM port
- **~ 110 Kgates + 22Kb**
- **.35 micron next gen. enables integration of control uP onto ASIC**

Also Siemens TriCore

Current .68 micron chip is 74 sq. mm



frees 100 Kgates
? cryptography
? network support

.35 micron frees 40 sq. mm

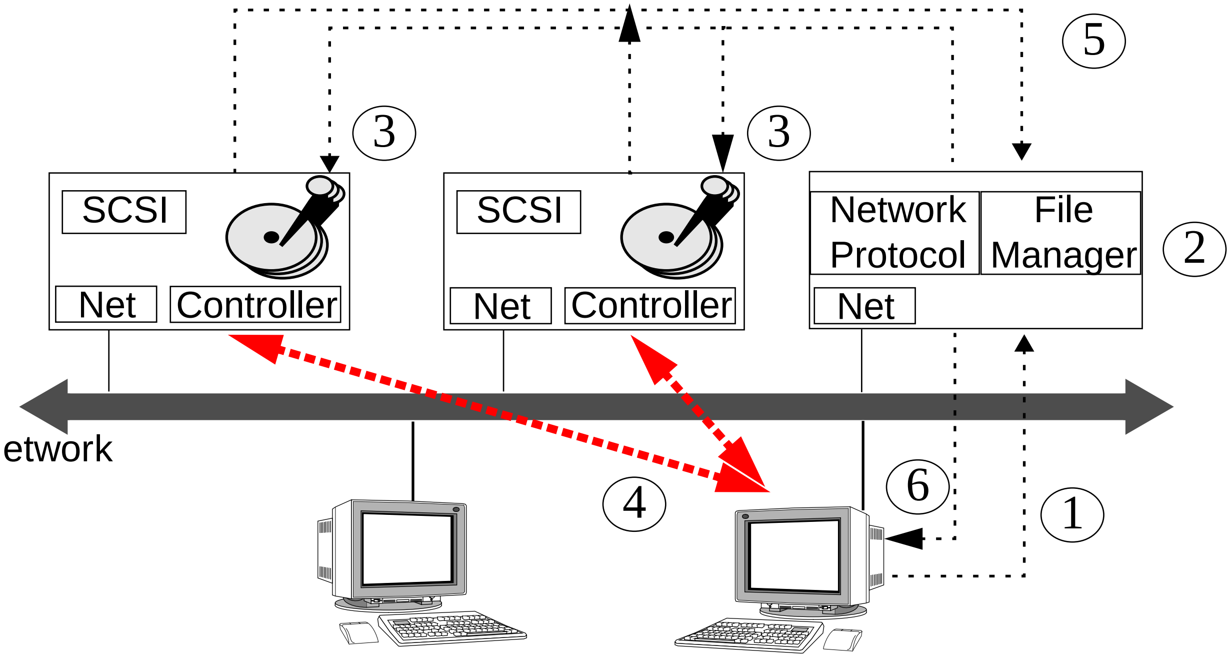
Insert .35 micron StrongArm RISC uP fits in 27 sq. mm with 8K+8K cache at 200 MHz, 200+ Dhrystone MIPS

First approach: Networked SCSI (NetSCSI)

Minimize change in HW, SW, IF: RAID-II, HPSS

- server translates (2) and forwards (3) request (1)
- drive delivers data directly to client (4)
- drive status to server (5), server status to client (6)

Scalable bandwidth through network striping



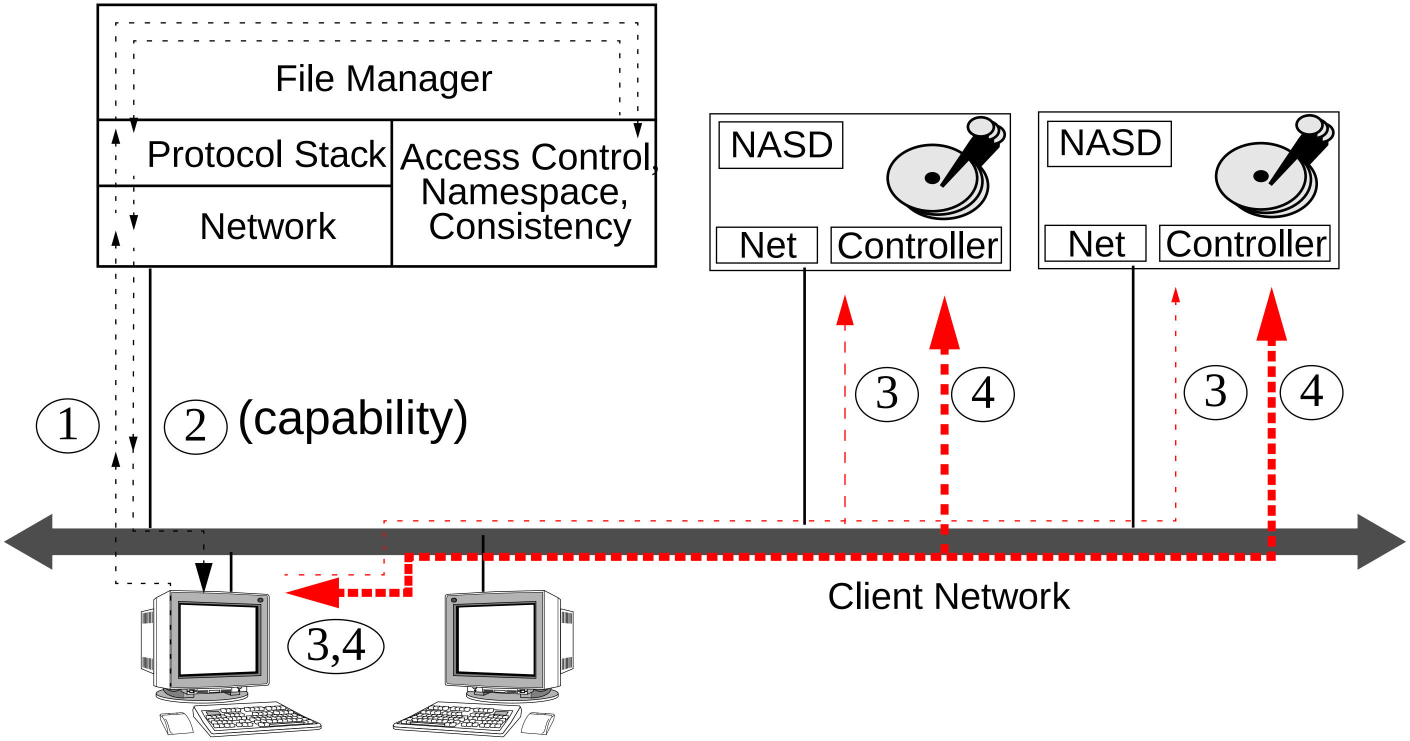
Sigmetrics97

More scalable: NASD enforces cached policy decisions

Avoid file manager unless new policy decision needed

- spread access computation over all drives under manager
- access control once (1,2) for all accesses (3,4) to drive object

Scalable BW thru striping, off-load manager



NASD Interface Design: Storage Objects

Per-file metadata in drive to avoid manager

- Not at client: don’t rest integrity on trusted client
- Not in capability: too large, hard to optimize in drive

Layout is best (actually) done below SCSI

- real-time support possible; accurate geometry
- transparent performance optimization (ie. AutoRAID)

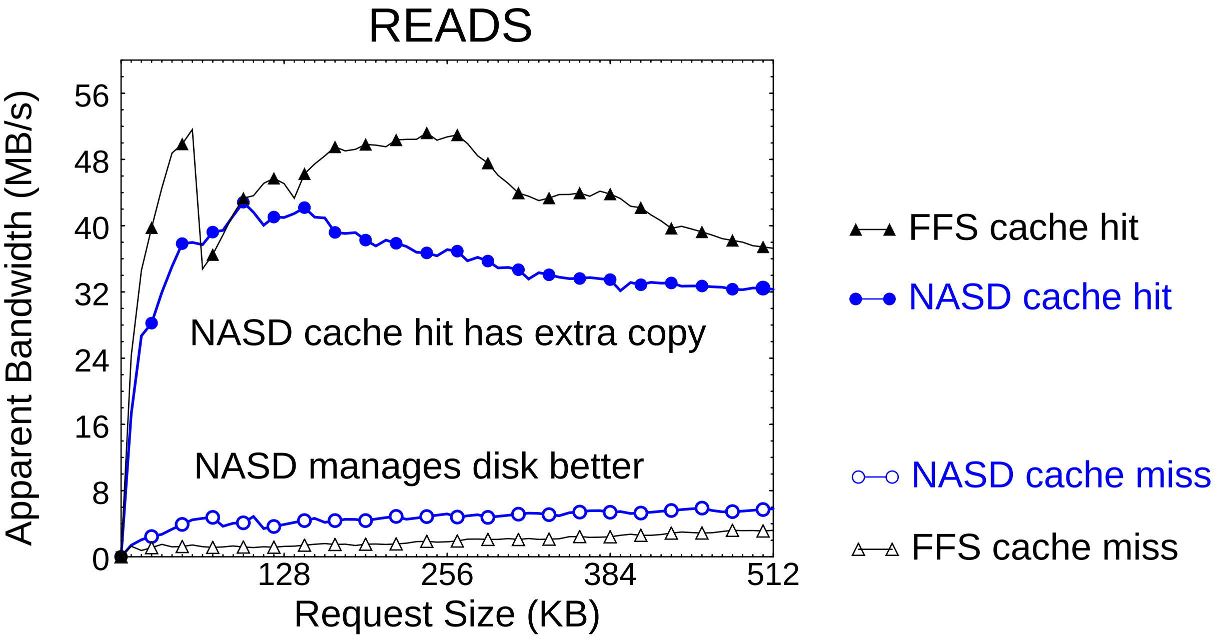
A NASD is an Object-Based Disk

NASD object store prototype

Prototype as psuedo-device in DU3.2, 16K loc, DCE

Performance comparable to FFS for file access

- 133 MHz Alpha, striped dual ST52160s
- replace NASD RPC interface with local system call



NASD cache hit has extra copy

NASD manages disk better

NASD computation is affordable

Prototype measured: 40 Kinstr/request + 3 instr/byte

- scale to 200 MHz: plenty fast enough for cache misses
- too slow during cache hits (need 0.3ms 1B; 2.2ms 64KB)
- but instrumentation shows most code in RPC/protocol stack

Commodity drives not built like workstations

- ASIC state machines for data: communications; copying
- of course, Alpha (21064) is not a microcontroller

Operation	Total Instructions (K) / % Communications						Operation time (msec) (200 MHz, CPI = 2.2)		
Request Size (B)	1		8 K		64 K		1	8 K	64 K
read - cold cache	46	70	67	79	247	90	0.51	0.74	2.7
read - warm cache	38	92	57	94	224	97	0.42	0.63	2.5

Adapting filesystems to NASD drives

Reorganize decomposition of function (aka port)

Primitives become drive responsibility

- data transfer, synchronous/automatic metadata updates

Policy remains manager responsibility

- namespace definition/navigation
- access control policy
- client cache management
- multi-access atomicity

Managers retain control through capabilities

- exploiting attributes for naming and revocation

Mapping filesystem to NASD objects

Simple model

- each file and directory bound to separate NASD object
- file attributes inherit object attributes (times, logical size)

Multiple objects per file?

- internal structure: database pages, mpeg group-of-pictures
- NASD striping, redundancy

Multiple files/directories per object?

- probable contiguity, prefetching; shared metadata overhead
- capabilities can be restricted to object region

NFS, AFS simple model; Cheops PFS multiple per file

Cheops: striping storage middleware for bandwidth

Asynchronous storage management oversight

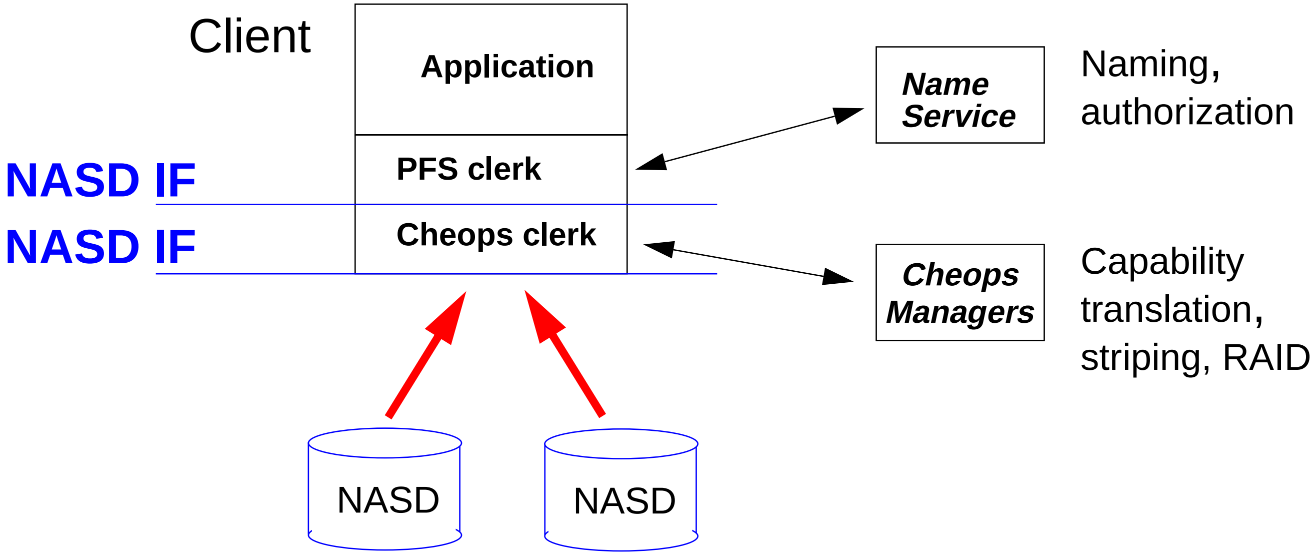
- first access installs capabilities/maps for aggregate object

Client asking for service pays for it (synchronizer)

- striping, RAID, incremental growth, consistent caches

Compatible with user-level network access

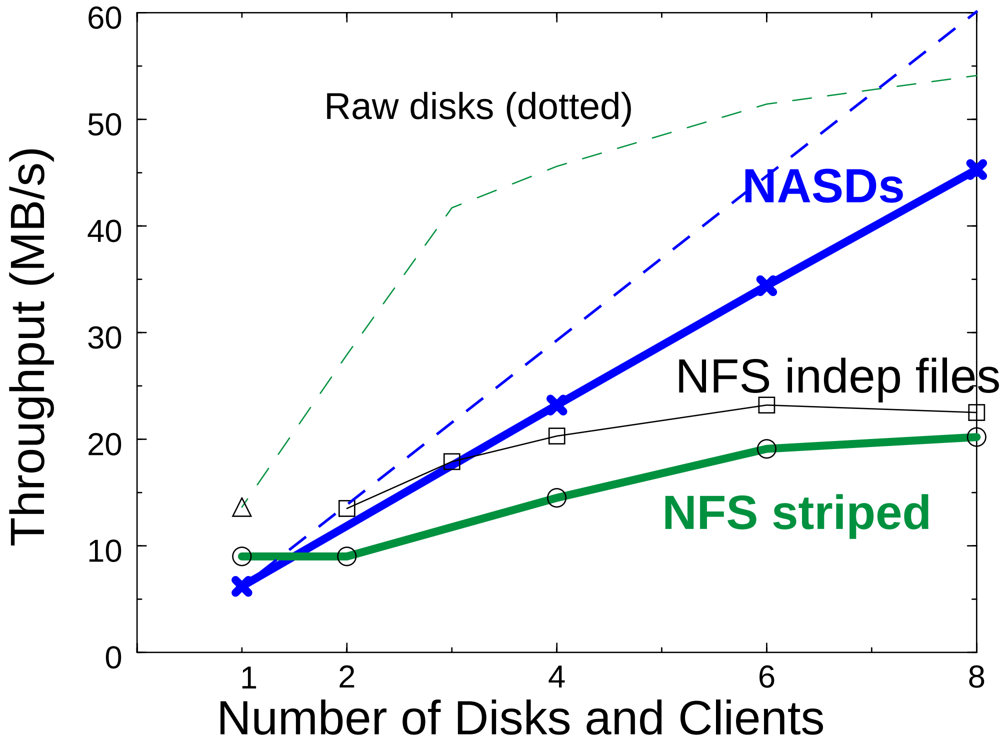
- NIC protocol processing leaves client to run application



Demonstration: scalable bandwidth for applications

NASD PFS delivers aggregate of raw disks’ bandwidth

- **Parallel association rule discovery on 300 MB of sales records**
- **NASD middleware fetches 4 x 512KB blocks in parallel**
- **NFS server delivers 20% disk BW (60% net BW) @ 8 pairs**



- **133Mhz NASDs**
  6 MB/s drive’s max
- **233Mhz clients**
- **MPI + SIO LLAPI**
- **switched OC3 ATM**
- **500 Mhz NFS server**
  14 MB/s drive’s max
  dual OC3 links

What to do with device cycles left over?

Large database systems - lots of disks, lots of power

System	Processing (MHz)		Data Rate (MB/s)	
	CPU	Disks	I/O Bus	Disks
Compaq TPC-C	4 x 200=**800**	*113* x 75=**8,475**	**133**	**1,130**
Microsoft Terraserver	4 x 400=**1,600**	*320* x 75=**24,000**	**532**	**3,200**
Digital 500 TPC-C	1 x 500=**500**	*61* x 75=**4,575**	**266**	**610**
Digital 4100 TPC-D	4 x 466=**1,864**	*82* x 75=**6,150**	**532**	**820**

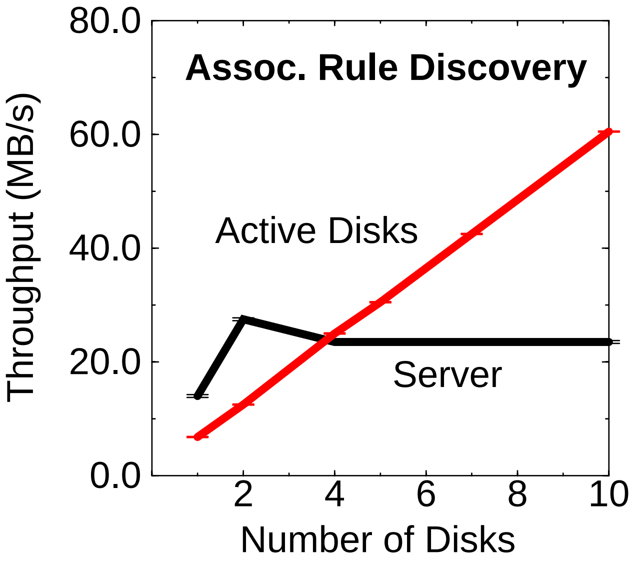
- **assume disk offers equivalent of 75 host MHz**
- **assume disk sustained data rate of 10 MB/s**

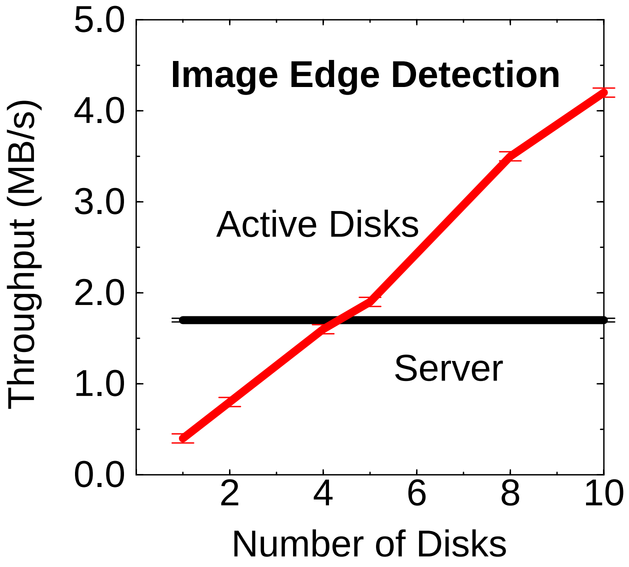
More cycles and MB/s in disks than in host

Simple throughput model for scan apps

Offload parallelized filter/scan operators

- speedups of 2-3X on 10 disks for 4 mining/image apps
- object model makes programming in drive simple

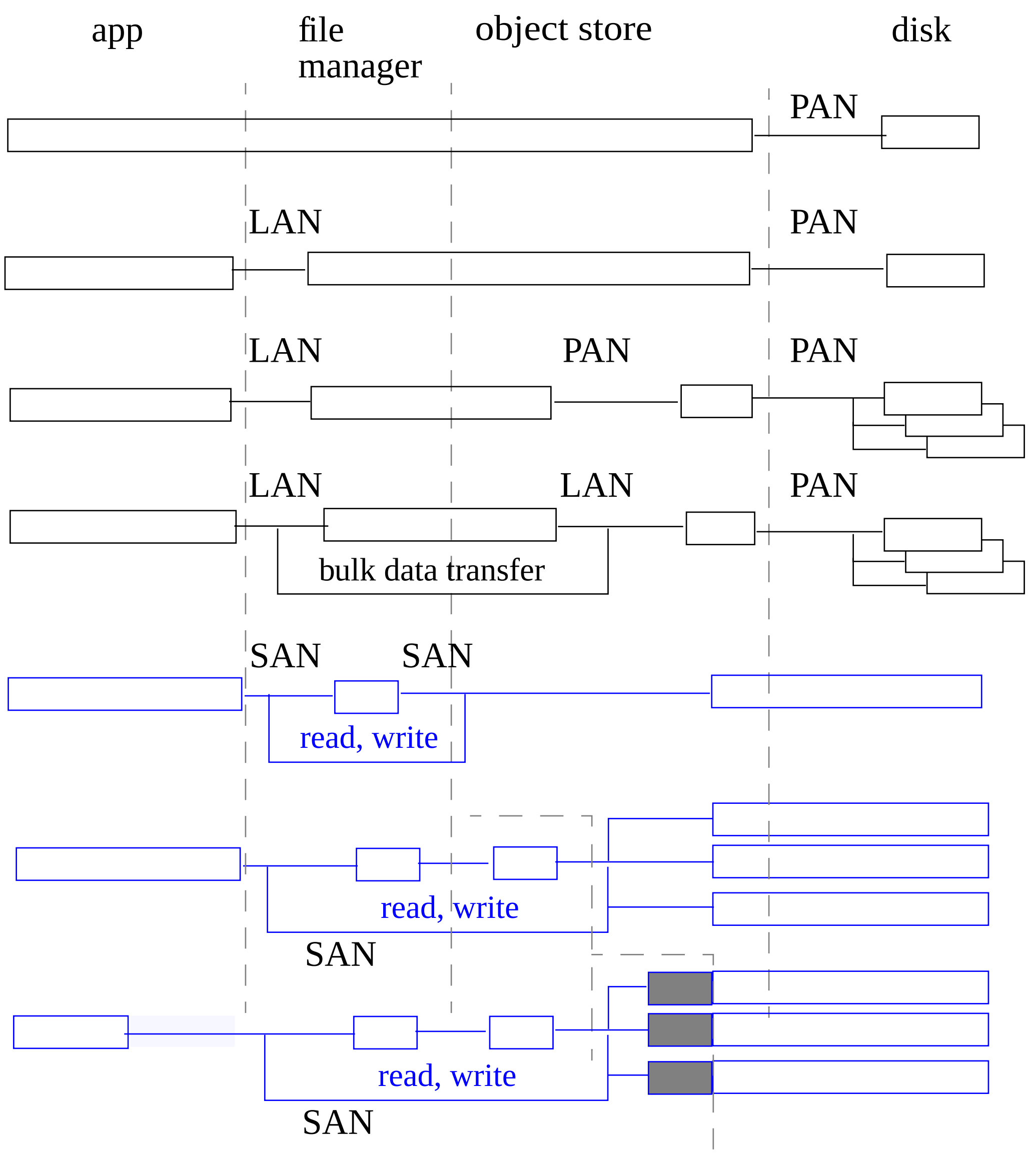




VLDB98

- crossover: host/disk-cpu-speed ratio ~ 4 (2 generations)

Storage interface evolution taxonomy



computers in data/control path	
2	Local Filesystem
3	Distributed FS (DFS)
4	DFS + RAID
3/4	DFS + HPSS bypass
2/2+	DFS w/ NASD
2/2++	DFS w/ NASD, Cheops
2/2++	DFS w/ NASD, Cheops and Active Disks

NASD: A cost-effective, high-bandwidth storage architecture

Cost-effective storage bandwidth starts in the drive

NASD is

- Direct transfer between client & storage device
- Asynchronous policy managememt
- (Cryptographic) capabilities
- Object-based management in drive, across drives

Cost-effective, efficient networking is critical

Storage architecture changes need standards

- www.nsic.org/nasd and www.snia.org