Reliable, Parallel Storage Architecture: RAID & Beyond

Garth Gibson

School of Computer Science & Dept of Electrical and Computer Engineering
Carnegie Mellon University

garth.gibson@cs.cmu.edu
URL: http://www.cs.cmu.edu/Web/Groups/PDL/
Anonymous FTP on ftp.cs.cmu.edu in project/pdl

Patterson, Gibson, Katz, Sigmod, 88.
Gibson, Hellerstein, Asplos III, 89.
Gibson, MIT Press, 92.
Gibson, Patterson, J. Parallel and Distributed Computing, 93.
Drapeau, et al., ISCA, 94.
Chen, Lee, Gibson, Katz, Patterson, ACM Computing Surveys, 94.
Stodolsky, Courtright, Holland, Gibson, ACM Trans on Computer Systems, 94.
Holland, Gibson, Siewiorek, J. Distributed and Parallel Databases, 94.
Patterson, Gibson, Parallel and Distributed Information Systems, 94.
Patterson, et al, CMU-CS-95-134, 95.
Gibson, et al, Compcon, 95.

Tutorial Outline

- RAID basics: striping, RAID levels, controllers
- recent advances in disk technology
- expanding RAID markets
- RAID reliability: high and higher
- RAID performance: fast recovery, small writes
- embedding storage management
- exploiting disk parallelism: deep prefetching
- RAID over the network

Review of RAID Basics (RAID in the 80s)

Motivating need for more storage parallelism

Striping for parallel transfer, load balancing

Rapid review of original RAID levels

Simple performance model of RAID levels 1, 3, 5

Basic controller design

RAID = Redundant Array of Inexpensive (Independent) Disks

Patterson, Gibson, Katz, Sigmod, 88.
P. Massiglia, DEC, RAIDBook, 93.

What I/O Performance Crisis?

Existing huge gap in performance

- access time ratio, disk:dram, is 1000 X sram:dram, onchip:sram

Cache-ineffective applications stress hierarchy

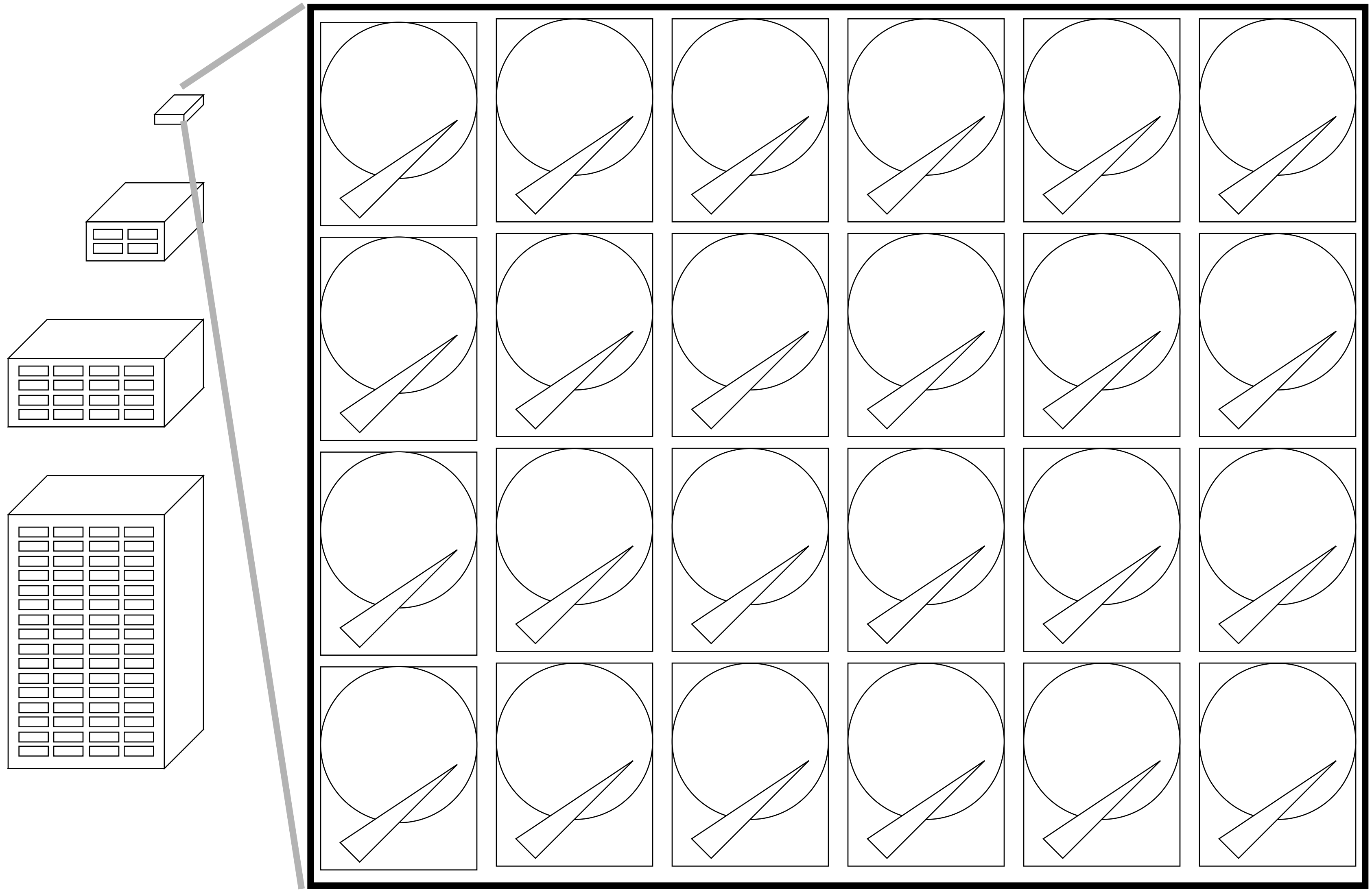
- video, data mining, scientific visualization, digital libraries

Increasing gap in performance

- 40-100+%/year VLSI versus 20-40%/year magnetic disk

Amdahl's law implies diminishing decreases in application response time

Disk Arrays: Parallelism in Storage



More Disk Array Motivations

Volumetric and floorspace density

Exploit best technology trend to smaller disks

Increasing requirement for high reliability

Increasing requirement for high availability

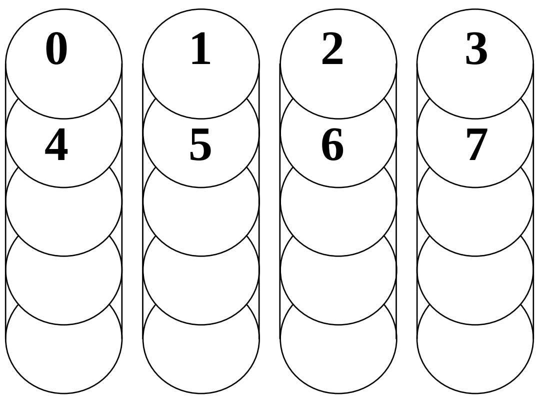
Enabling technology: SCSI storage abstraction

Data Striping for Read Throughput

Parallelism will only be effective if

- load balance high concurrency, small accesses
- parallel transfer low concurrency, large accesses

Striping data provides both



- **uniform load for small independent accesses**
  stripe unit large enough to contain single accesses
- **parallel transfer for large accesses**
  stripe unit small enough to spread access widely

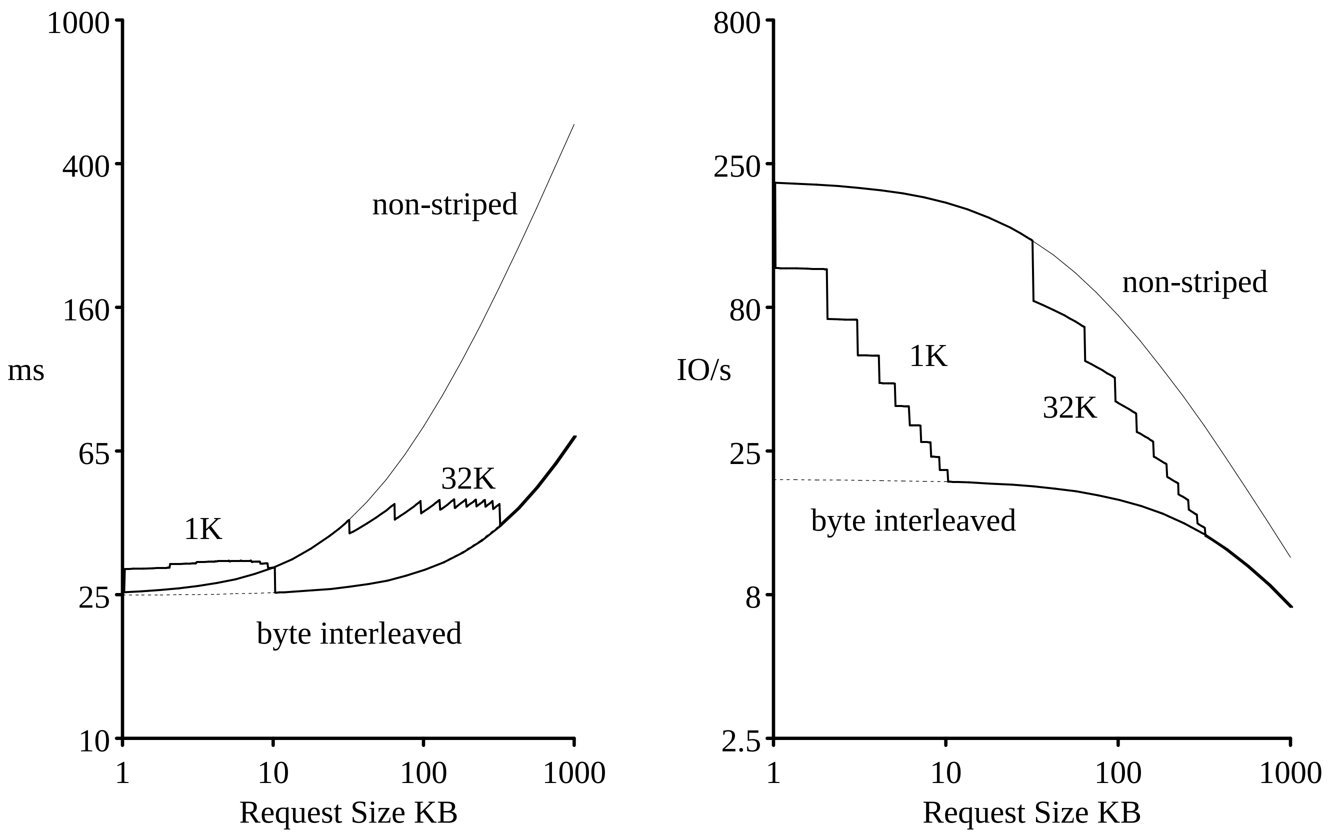
Livny, Khoshafian, Boral, Sigmetrics, 87.

Sensitivity to Stripe Unit Size

Simple model of 10 balanced, sync'd disks

Response Time
at low loads

Throughput
at 50% utilization



Gibson, MIT Press, 92 (derived from J. Gray, VLDB, 90).

Selecting a Good Striping Unit (S.U.)

Balance benefit to penalty:

- **benefit: parallel transfer shortens transfer time**
- **penalty: parallel positioning consumes more disk time**

Stochastic simulation study of maximum throughput yields simple rules of thumb

- **16 disks, sync'd; find max min of normalized throughput**

Given I/O concurrency (conc)

- **S.U. = 1/4 * (positioning time) * (transfer rate) * (conc - 1) + 1**

Given zero workload knowledge

- **S.U. = 2/3 * (positioning time) * (transfer rate)**

Chen, Patterson, ISCA, 90.

Adding Redundancy to Striped Arrays

Meet performance needs with more disks in array

- implies more disks vulnerable to failure

Striping all data implies failure effects most files

- failure recovery involves processing full dump and all increments

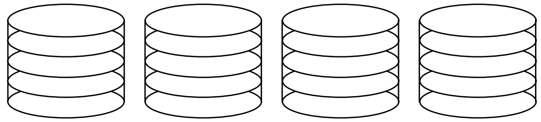
Provide failure protection with redundant code == RAID

Single failure protecting codes

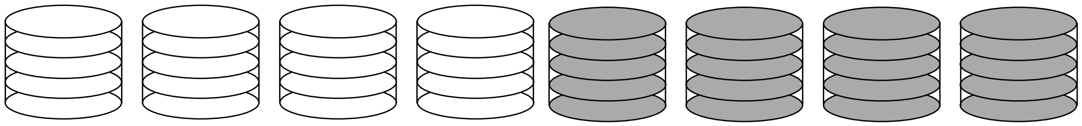
- general single-error-correcting code too powerful
- disk failures are self-identifying - called erasures
- fact: T-error-detecting code is also a T-erasure-correcting code

Parity is single-disk-failure-correcting

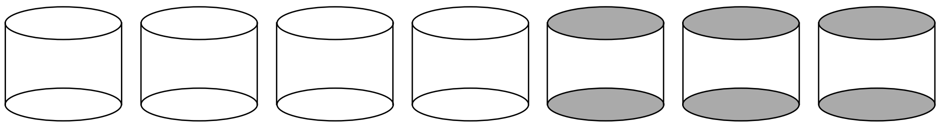
Synopsis of RAID Levels



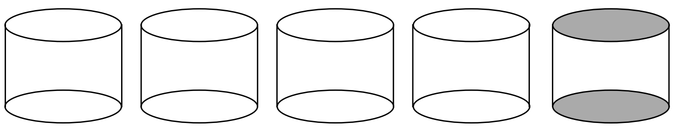
RAID Level 0: Non-redundant



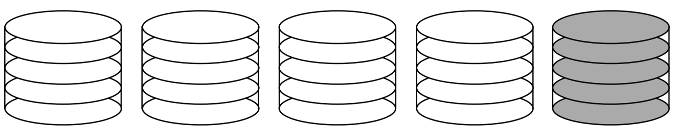
RAID Level 1: Mirroring



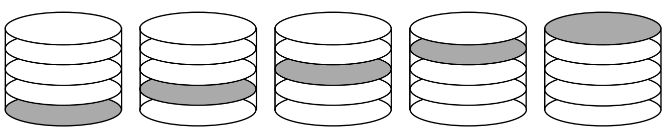
RAID Level 2: Byte-Interleaved, ECC



RAID Level 3: Byte-Interleaved, Parity



RAID Level 4: Block-Interleaved, Parity

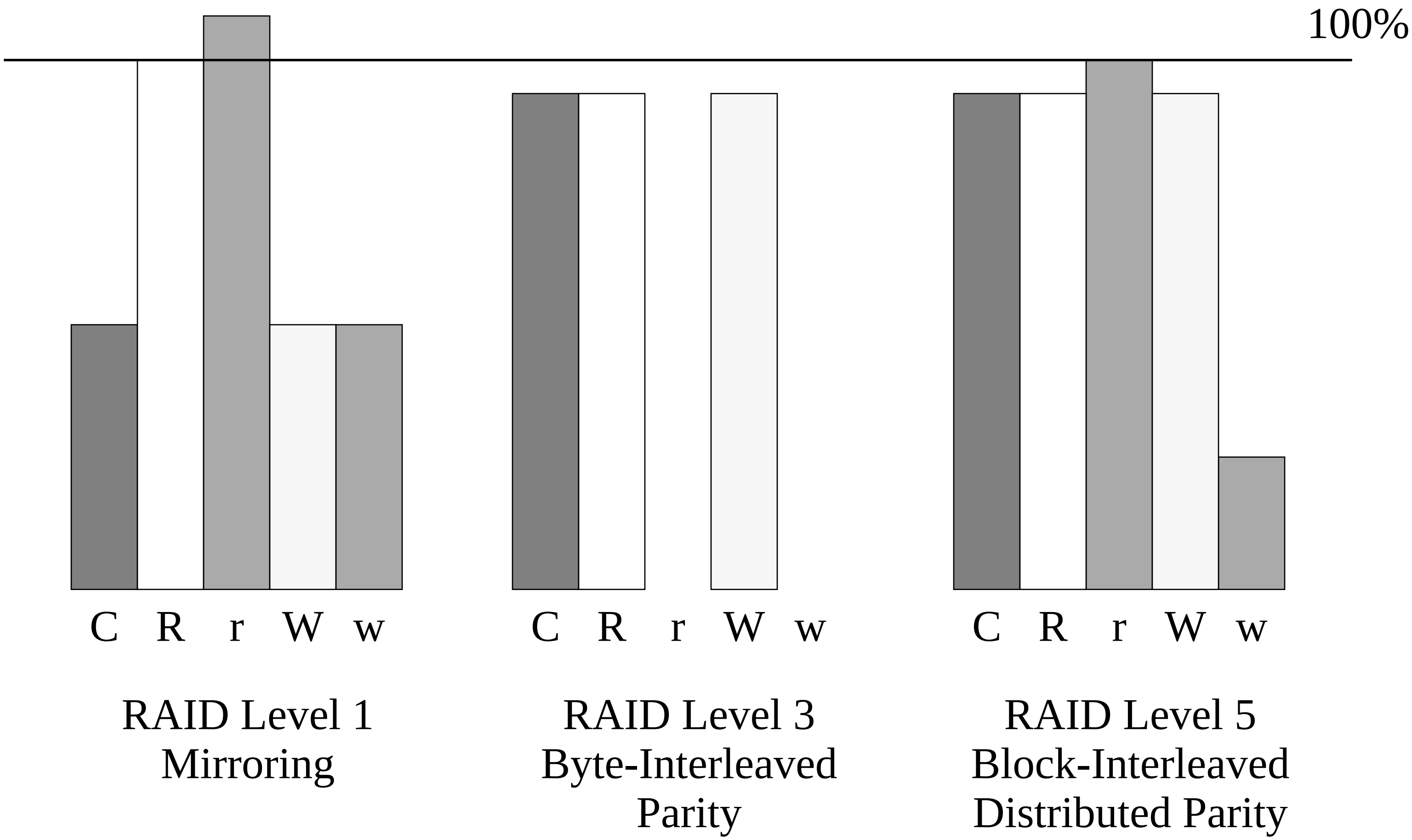


RAID Level 5: Block-Interleaved, Distributed Parity

Chen, Lee, Gibson, Katz, Patterson, ACM Computing Surveys, 94.

Basic Tradeoffs in RAID Levels

Relative to non-redundant, 16 disk (sync'd) array



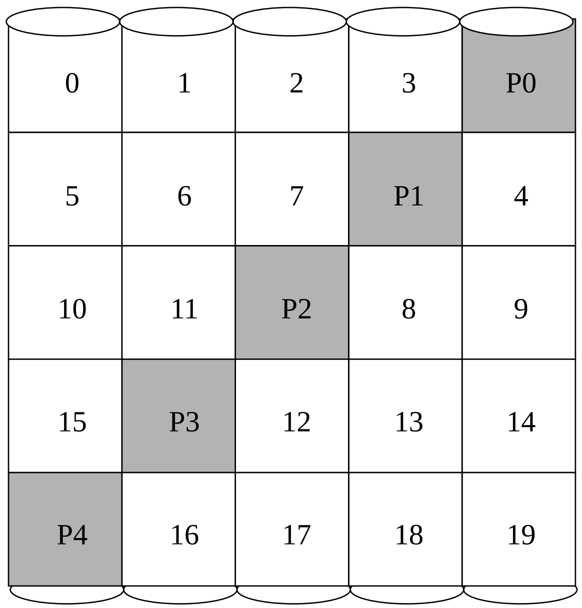
Patterson, Gibson, Katz, Sigmod, 88.
Bitton, Gray, VLDB, 88.

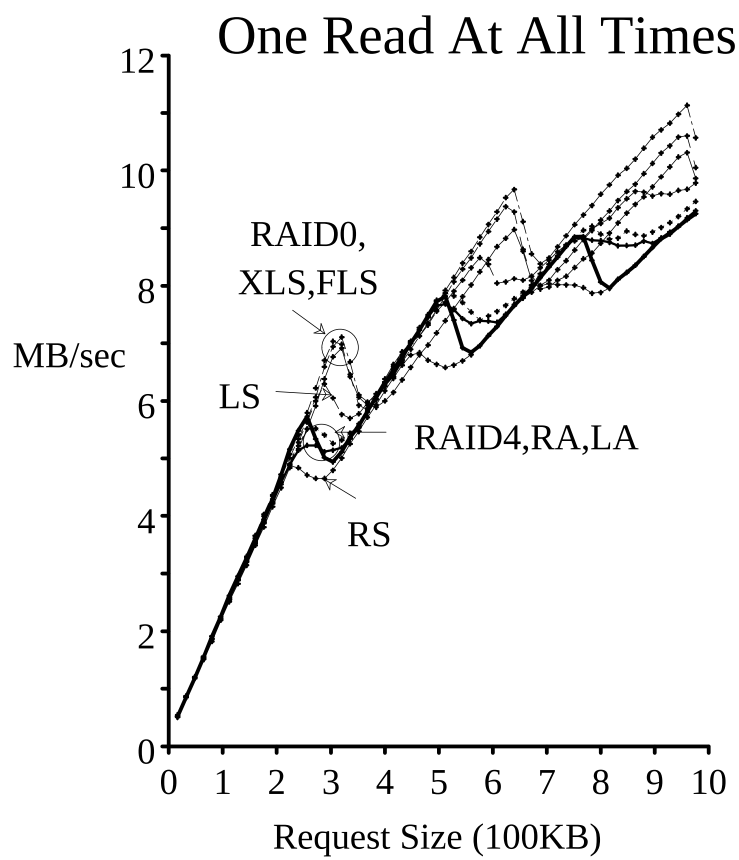
Parity Placement

Throughput generally insensitive

- **stochastic workload on 2 groups of 4+1, 32 KB stripe units**

Left Symmetric





Lee, Katz, Asplos IV, 91.

Stripe Unit (S.U.) Size in RAID

Writes in RAID must update redundancy code

- full stripe overwrites can compute new code in memory
- other accesses must pre-read disk blocks to determine new code
- write work limits concurrency gains with large S.U.

Rerun Chen's ISCA 90 simulations in RAID 5

Given I/O concurrency (conc)

- read S.U. = 1/4 * (positioning time) * (transfer rate) * (conc - 1) + 1
- write S.U. = 1/20 * (positioning time) * (transfer rate) * (conc - 1) + 1

Given zero workload knowledge

- S.U. = 1/2 * (positioning time) * (transfer rate)

Chen, Lee, U. Michigan CSE-TR-181-93.

Modeling RAID 5 Performance

Queueing model for RAID 5 response time

- controller directly manages disk queues
- separate channels for each disk
- first-come-first-serve scheduling policies
- accesses are 1 to N stripe units, N = number of data disks
- detailed model of disk seek and rotate used

Model succeeds for case of special parity handling

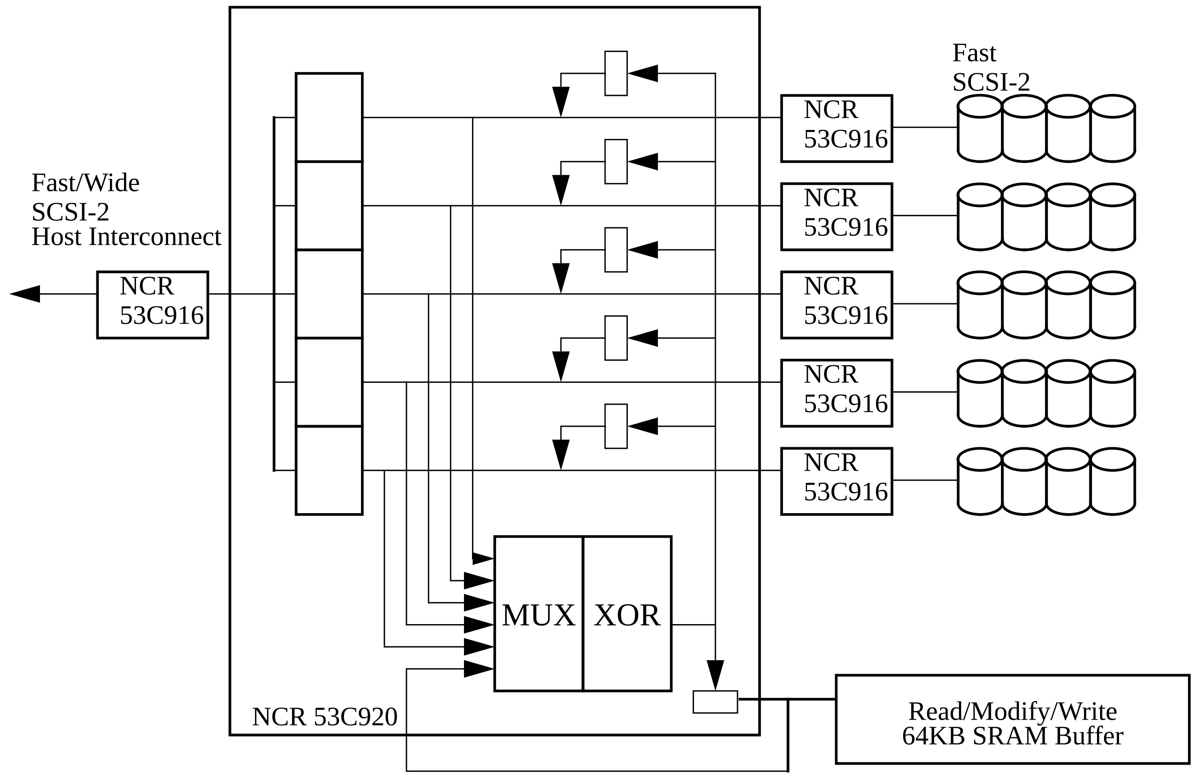
- separate, high-priority queue for parity R-M-W operations
- parity R-M-W not generated until (last) data operation starts
- parity R-M-W does not preempt current data operation
- implies parity update nearly always one rotation, finishing last

S. Chen, D. Towsley, J. Parallel and Distributed Computing, 93.

Example Controller: NCR (Symbios) 6299

Minimal buffering for RAID 5 parity calc. only

- **low cost and complexity controller - quick to market**
- **problem: concurrent disk accesses return out-of-order (TTD/CIOP)**
- **problem: max RAID 5 bandwidth is single drive bandwidth**

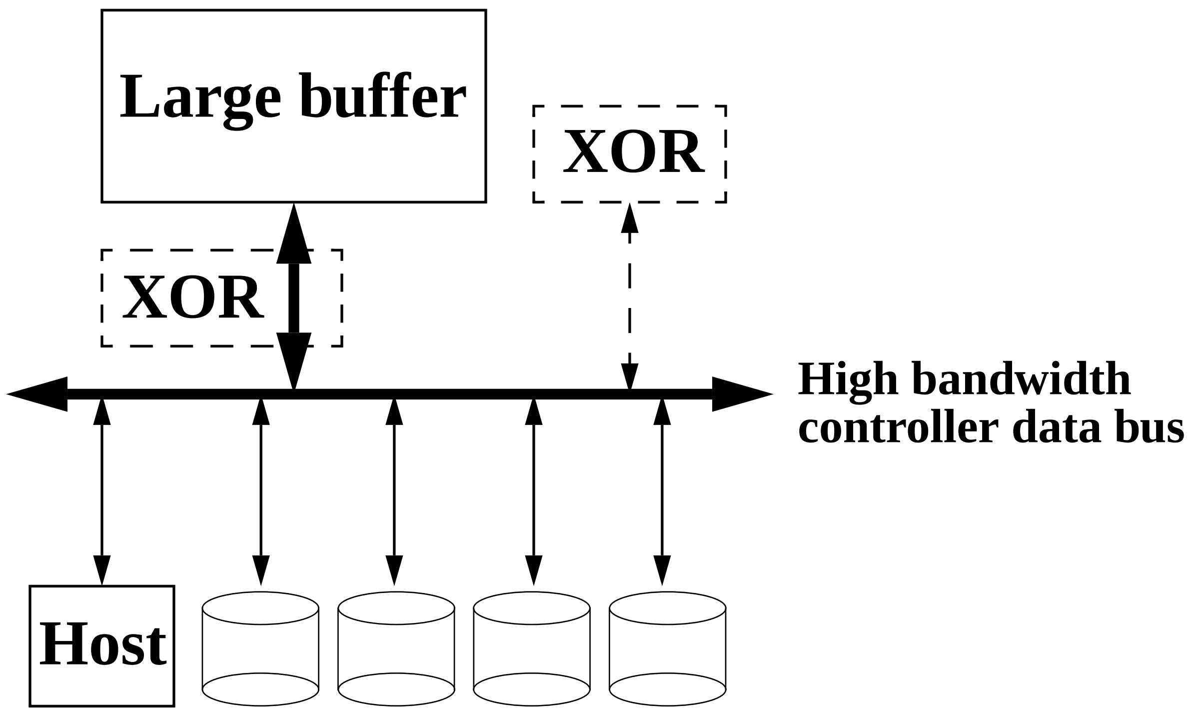


Chen, Lee, Gibson, Katz, Patterson, ACM Computing Surveys, 94.

Adding Buffering to RAID Controller

Performance advantages; added complexity

- parallel independent transfers to all disks
- XOR placement: stream through or separate transfer
- buffer management - extra buffers give higher concurrency



Menon, Cortney, ISCA, 93.

Recap RAID Basics

Disk arrays respond to increasing requirements for I/O throughput, reliability, availability

Data striping for parallel transfer, load balancing

Stripe unit size important

- **1/2 to 2/3 of disk positioning time, transfer rate product**

Simple code, parity, protects against disk failure

Berkeley RAID levels

- **Mirroring, RAID 1, is costly, faster small writes**
- **RAID 5, is cheaper, faster large writes; response time model exists**

Controller design hinges on XOR, buffer mgmt

Current Trends in Magnetic Disk Technology

Rapid density increase is major cause of change

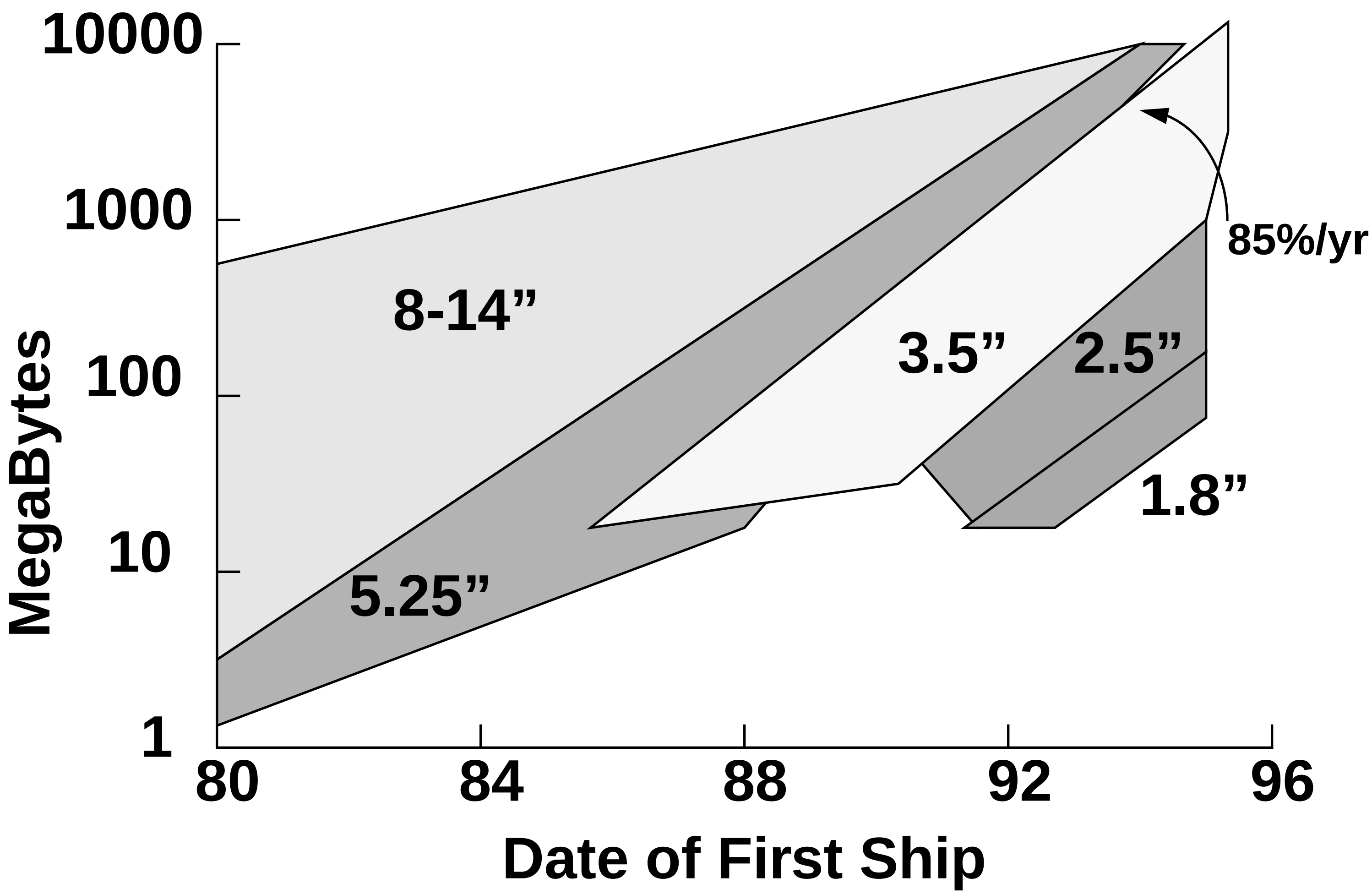
Access time and disk diameter lagging

Embedded intelligence for value-added

Interface technologies up in the air

Competing technologies have niche roles

Magnetic Disk Capacity



G. Milligan, Seagate, RAID’95 Forum, April 95.

Areal Density Driving Magnetic Disk Trends

1989 sea change in trends

- **prior to 1989, density grew at 27% per year**
- **since 1989, density is growing at 60% per year**
  - linear bit density growing at 20+% per year
  - tracks per inch growing at 30+% per year
- **2.5” and 3.5” products with 600+ Mbits/sq.inch in 1995**
- **spurred by wide acceptance of SCSI open standard**
- **magnetic disk densities are greater than optical disk densities**

Much higher densities demonstrated in lab

- **IBM: 1 Gbit/sq.inch in 1989, 3 Gbit/sq.inch in 1995**
- **CMU’s DSSC target is 10 Gbits/sq.inch in 1998**

Magnetic Disk Price Trends

Revenue per unit shipped constant over time

- density growing 60% per year -> price falling 40% per year
- some claim price has been falling by 50% per year since 92

Current price leader 3.5”: .23-.40 $/MB (.31wtd)

- 2.5”: .5-1.1 $/MB; 1.8”: 1.1-2.9 $/MB

Total revenue stream growing slowly

- 1994 brought in $23 billion on 60 million units
- projected yearly revenue increase in billion dollars:
  2.7 in 1995, 5.1 in 1996, 2.3 in 1997, 1.9 in 1998, 1.4 in 1999
- competition for market share may reduce vendors

B. Frank, Augur Visions Inc, J. Molina, Tech Forums, and G. Garrettson, Censtor, RAID’95 Forum, April 95.

Key Magnetic Disk Technologies

Magneto-resistive heads

- replace single inductive (thin film) head with two heads: one inductive write head, one magneto-resistive head
- write-wide, read-narrow; higher signal-noise on read

Ever lower flying heights (< 2 microinches)

- reduced mass sliders, thinner/smoother surfaces
- tolerating frequent “skiing”

Decoding interfering signals (PRML)

- bits too close to ignore interference
- decode based on sequence of samples, neighbor bit values

Magnetic Disk Performance Trends

Access time decreases near 1/year curve

- median access times: 25 ms in 1990, 15 ms in 1995, 10 ms in 1999
- minimum access times: 15 ms in 1990, 12 ms in 1995, 10 ms in 1996

Data rate growing with bits/inch and rpm

- long term growth at 10% year (fixed rpm)
- in last couple of years increased near 75% per year
- limited by decode circuit (analog/digital VLSI)

Disk Diameter Trends

Decreasing diameter dominated 1980s

- **5.25” created desktop market (16+ GB soon)**
- **3.5” created laptop market (4+ GB 1/2 high; 500+ MB 19mm)**
- **2.5” dominating laptop market (200+ MB; IBM 720 MB)**
- **1.8” creating PCMCIA disk market (80+ MB)**

Decreasing diameter trend slowed to a stop

- **1.8” market not 10X 2.5” market**
- **1.3” (HP Kittyhawk) discontinued**
- **vendors continue work on smaller disks to lower access time**

Storage Interfaces

Small Computer System Interface (SCSI)

- dominates market everywhere but cheapest systems

SCSI provides level of indirection

- linear block address space rather than track/head/sector
  allows rapid introduction of non-standard geometries
- internal buffer separates bus from media
  allows rapid introduction of non-standard media data rates
- internal controller command queueing
  allows geometry-specific, real-time disk scheduling
- internal controller and buffer for caching, read-ahead, write-behind

But SCSI does too much handshaking

- short, slow cables, limited bus ports

Trends in Storage Interfaces: Serial Buses

Faster, smaller, longer, more ports

Multiple contenders:

- **Serial Bus (P1394): isochronous desktop peripheral bus (10MB/s)**
- **IBM SSA (X3T10.1): dual ring packetized disk interface (20MB/s)**
- **FibreChannel (X3T9.3): merge peripheral and network interconnect lengths to 10 km, speeds to 100 MB/s, multiple classes of service**

But, parallel SCSI is not dead yet

- **SCSI-2 is 8 or 16 bits parallel, 5 or 10 MHz (5, 10, 20 MB/s)**
- **Fast-20, Fast-40 are 20, 40 MHz (20, 40, 80 MB/s)**

SCSI-3 isolates SCSI from physical layer

What about other storage technologies?

Optical disk?

- limited market; disk density surpassed optical last year
- portable use possible, but contact and flying disks are ahead
- long term role in publishing

Magnetic tape?

- very low cost media in robots
- but low data rates (1-10 MB/s) and slow robot switch (4 min)
- long term role in archival storage

Holographic and other advanced media?

- Tamarack, Austin TX, may deliver 20 GB WORM jukebox
  high data rates possible (100 MB/s?)

Recap Magnetic Disk Technology

Rapid density increase is major cause of change

Access time and disk diameter lagging

Embedded intelligence for value-added

Interface technologies up in the air

Competing technologies have niche roles

Current Trends in RAID Marketplace

Rapid market development

- largest growth in storage subsystems, for LAN environment
- broad range of competition available

RAID standards organization

- RAID Advisory Board focuses on defining/qualifying RAID levels
- RAID support may appear in SCSI-3 specification

RAID Market Trends

Begun 6 years ago, $5.7 billion in 1994

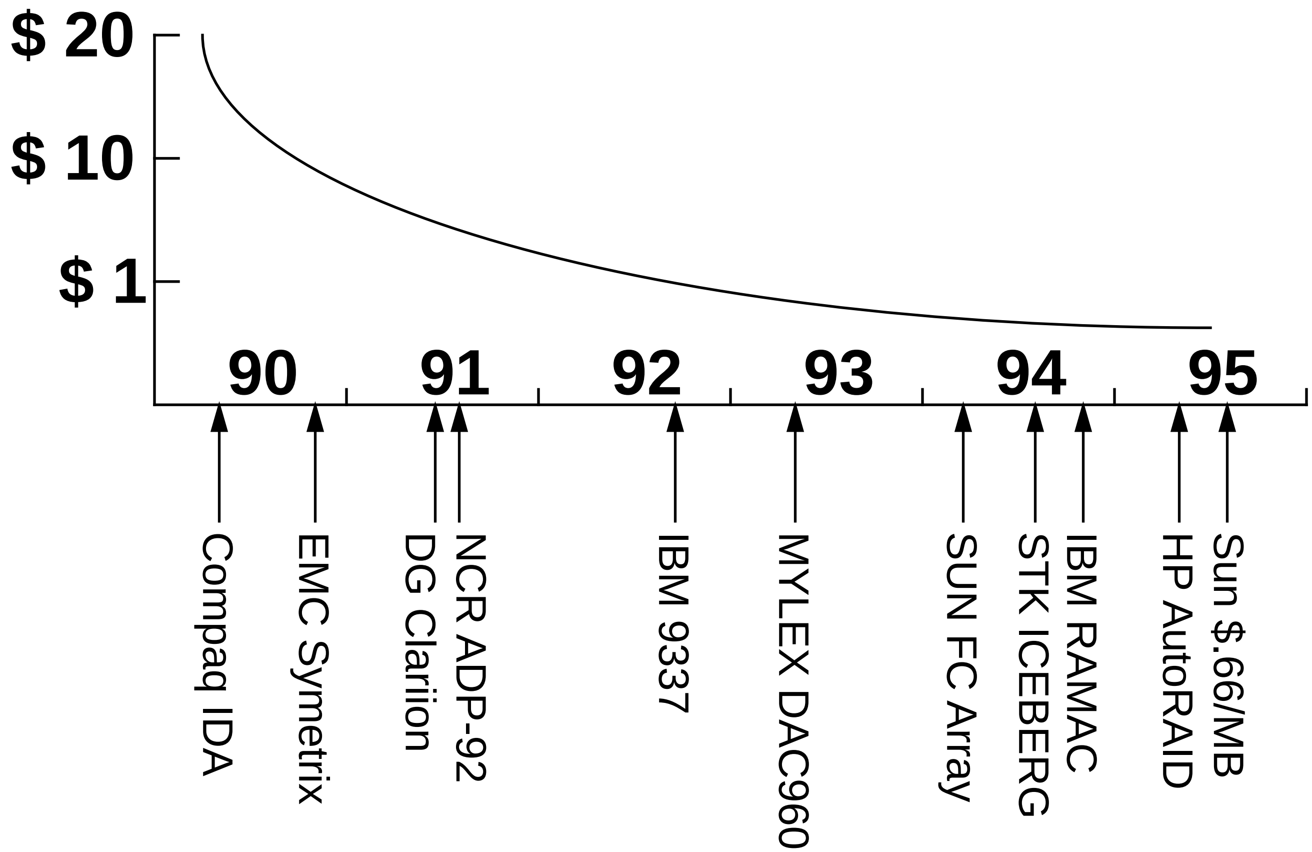
- 23% IBM, 22% EMC, 12% DEC, 10% Compaq, 7% HP, 4% Hitachi, 4% STK, 3% Sun, 3% DG, 2% Tandem, 1% Symbios, 1% Mylex, 1% Auspex, 1% FWB, 1% NEC

Continued growth predicted

- in billion $: 7.8 in 95, 9.7 in 96, 11.6 in 97, 13.2 in 98
- 66% -> 75% into network/minicomputer/multi-user systems
- units shipped 94 thru 97: 400,000 becomes 1,200,000
subsystems: 213,000 x 3.4 -> 730,000
boards: 120,000 x 2.5 -> 293,000
software: 72,000 x 2 -> 140,000

J. Porter, DISK/TREND, 95.

RAID Product History Sample



- **Rule of thumb: RAID MB cost 2x raw disk**

Symbios, RAID'95 Forum, April 95.

RAID Advisory Board

To promote use and understanding of RAID

- 55 member organization since formation in 92

Education

- RAIDBook - technology and RAID level definitions
- publishes in Computer Technology Review
- hosts Comdex RAID Technology Center

Standardization

- functional test suite: IO, buffering, queueing, error injection
- performance tests: synthetic: TP, FS, DB, video, backup, scientific
- host interface spec: joint development of SCSI-3 RAID support

SCSI-3 Support for Storage Arrays

Storage Array Conversion Layer (SACL)

- defines, manages, accesses, reconstructs array components
- pushes mapping of data and parity into SCSI software

SACL objects

- p_extents - contiguous range of blocks on one device
- ps_extents - portion of p_extent excluding redundancy
- redundancy groups - group of p_extents sharing protection
- volume sets - group of ps_extents contiguous in user data space
- spare - p_extent, device or component avail to replace failure

G. Penokie, RAID'95 Forum, April 95.

Recap RAID Market Trends

Rapid market development

- largest growth in storage subsystems, for LAN environment
- broad range of competition available

RAID standards organization

- RAID Advisory Board focuses on defining/qualifying RAID levels
- RAID support may appear in SCSI-3 specification

RAID Reliability

Single correcting codes (parity)

- coping with dependent failure modes (controllers, cables, power):
- contrasting mirroring (RAID 1) with parity (RAID 5)
- limitations in ever larger disk arrays

Double correcting redundancy codes (RAID 6)

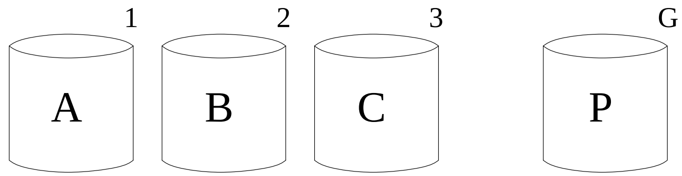
- basics: intersecting codewords allowing recovery choices
- binary, one bit per disk (2D parity)
- non-binary, one symbol per disk (Reed-Solomon)
- binary, multiple bits per disk (IBM EvenOdd)

Increasing focus on electronics, software impact

Basic Disk Data Reliability Model

Exponential disk lifetime and repair time

- **recovery must not fail; repair with on-line hot spare**

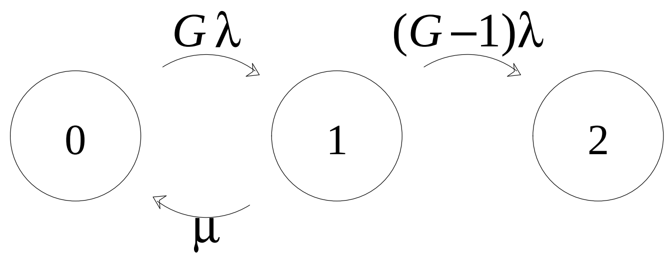


P = XOR(A,B,C)

C = XOR(A,B,P)

disk failure rate: λ = 1/MTTF-disk

disk repair rate: μ = 1/MTTR-disk



MTTF-disk >> MTTR-disk

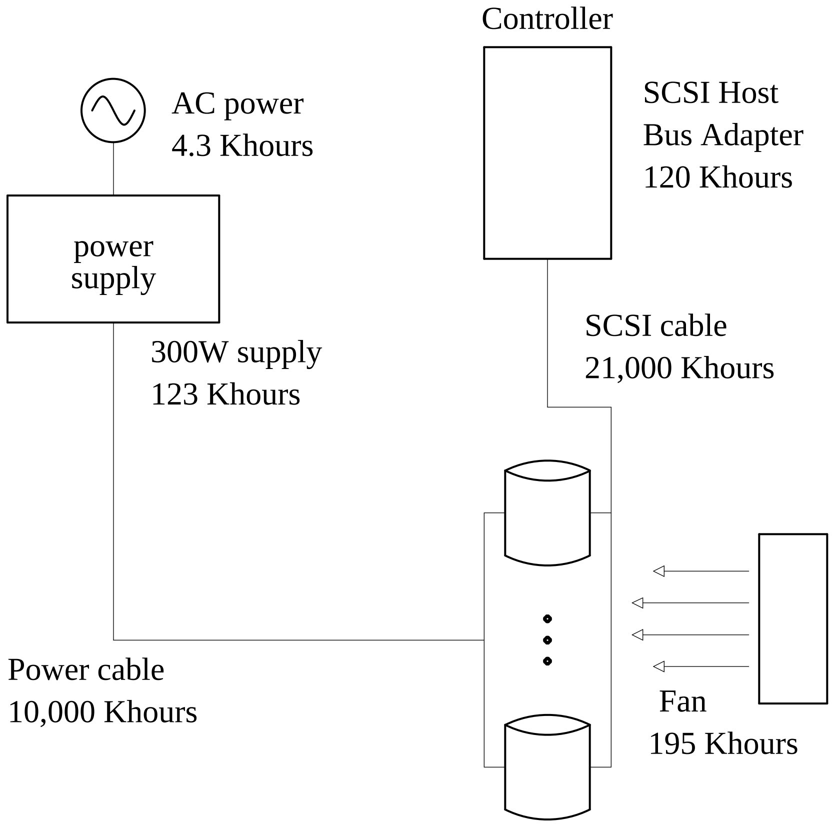
$$\text{MTTDL-RAID} = \frac{\text{MTTF-disk}^{2}}{\text{N G (G-1) MTTR-disk}}$$

Patterson, Gibson, Katz, Sigmod, 88.

Arrays contain Support Hardware

More hardware than just disks in an array

- **must defend against external power quality first**
- **combined effects of non-disk components rival disk failures**

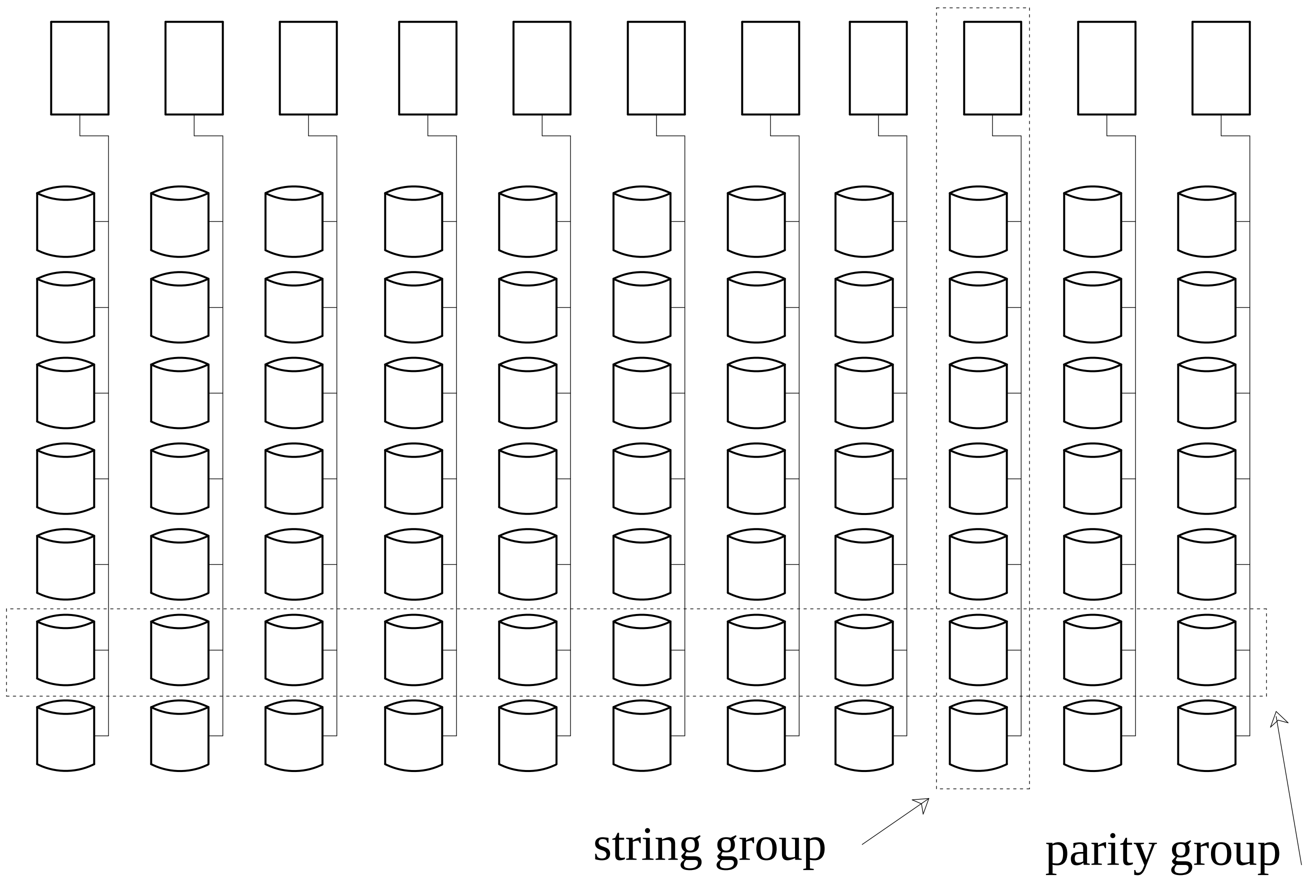


Schulze, Compcon, 89.

Orthogonal Parity Groups

Limit exposure of each parity group to one disk

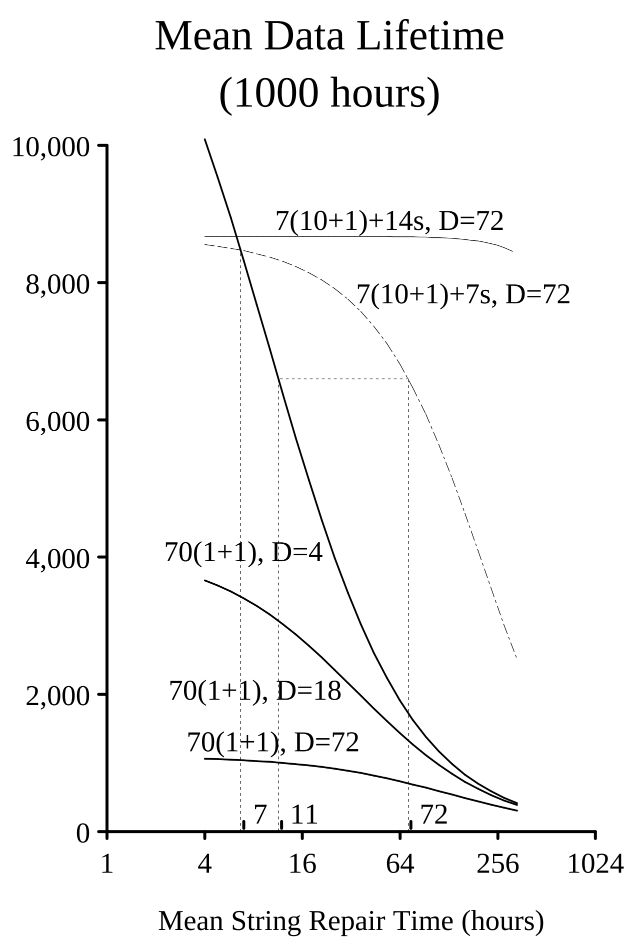
- organize support hardware into independent columns

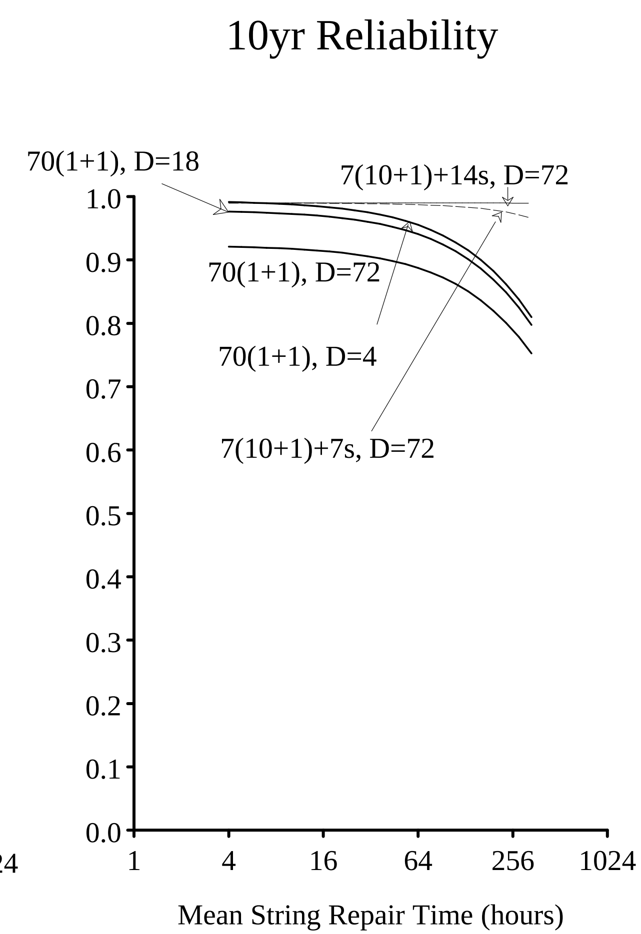


Mirror versus Parity Data Reliability

Simple mirrors less reliable and more costly

1 hour mean recovery to spare; 150 Khour mean disk, string lifetime





Carnegie Mellon

Gibson, Patterson, J. Parallel and Distributed Computing, 93.

Multiple Failure Tolerance?

I/O parallelism grows with processing speed

- larger arrays have more disks vulnerable to failure

But disk reliability has been growing at 50%/year

- allows 125%/year increase in processing speed at fixed reliability

No need for more powerful failure tolerance

- will disk reliability continue to rise at this rate?

Increasing needs for highly reliable storage

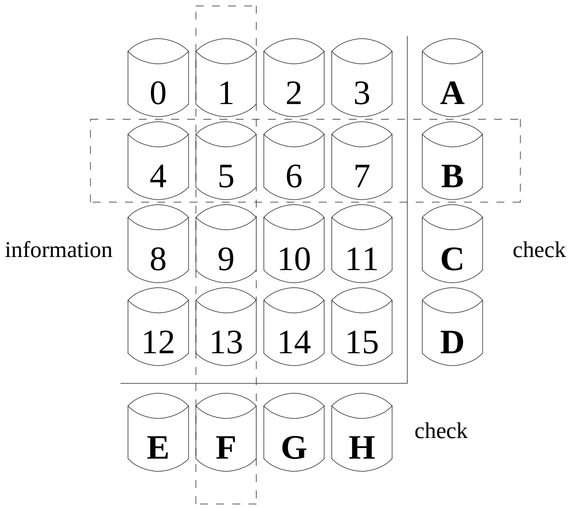
- are customers prepared to pay performance cost?

Simple, Fast Double Correcting Arrays

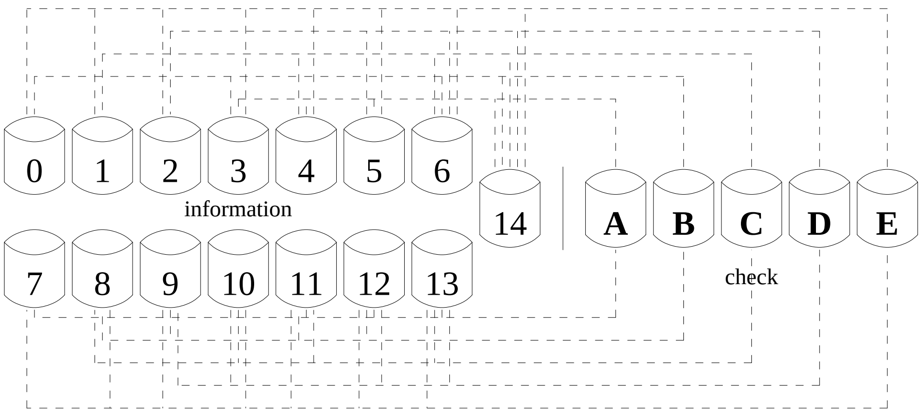
Borrow from earlier approaches (1 bit at a time)

- **orthogonal parity groups**
- **double-error detecting codes from memory systems**

2d- Parity



Extended Hamming



Overheads: check space versus check update time

- **2d-parity has minimal time overhead (3), but space grows as root**
- **Hamming has lower space overhead, but higher avg time cost**

Gibson, Hellerstein, Asplos III, 89.

Non-binary Codes (P+Q in STK Iceberg)

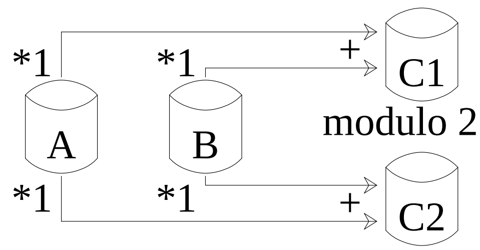
Exploit existing hardware (Reed-Solomon)

Binary (b=1)

C1 = C2 = (A+B) mod 2

A = f(C1,C2) ?

B = g(C1,C2) ?



Unique check disk pair for each data disk

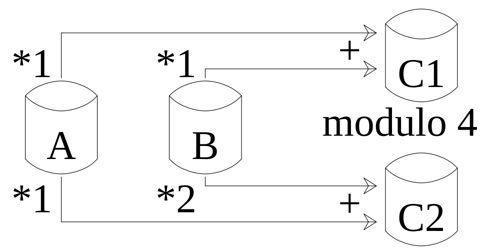
Nonbinary (b=2)

C1 = (A+B) mod 4

C2 = (A+2B) mod 4

A = (2C1-C2) mod 4

B = (C2-C1) mod 4



Multiple data disks can share same check disk pair

Gibson, MIT Press, 92.

IBM EvenOdd

disk 0	disk 1	disk 2	disk 3	disk 4	horiz parity	diag parity
1	0	1	1	0	p0	d0
0	1	1	0	0	p1	d1
1	1	0	0	0	p2	d2
0	1	0	1	1	p3	d3

Careful selection of intersecting codewords

- **horizontal codewords: a bit from each disk at same offset**

  (p0,p1,p2, p3) = (1, 0, 0, 1)

- **diagonal codewords: a bit from each disk from different offsets**

  main diagonal, {disk, offset}=(4,0),(3,1),(2,2),(1,3), has parity 1
  add main diagonal parity to parity of each other diagonal
  (d0, d1, d2, d3, d4) = (1, 1, 1, 1)+(1, 1, 0, 1) = (0, 0, 1, 0)

All computations are XOR (no RS hardware)

Small random write updates 3 disks (minimum)

Blaum, Brady, Bruck, Menon, ISCA, 94.

Trends to Higher Reliability: TQM

Drive reliabilities rising quickly

- 30,000 hr MTTF in 1987 -> 800,000 hr MTTF in 1995
- 50% of failures in electronics

Subsystem reliability rising more slowly

- much larger parts count in array controllers
- much large software component in array controller

Subsystem optimizations introduce new risk

- write-back cache failures significant

Increased emphasis on super-disk failure modes

Recap RAID Reliability

Basic reliability decreases linearly in group size

Dependent failure in support hw: orthogonal arrays

Parity + spares reliability, cost best simple mirrors

Need for multiple failure tolerance unclear

Multiple failure tolerance possible at low cost

But small write penalty substantially increased

Real push is tolerance for electronics/software faults

RAID Performance

On-line failure recovery: high availability

Exploiting spares: faster recovery

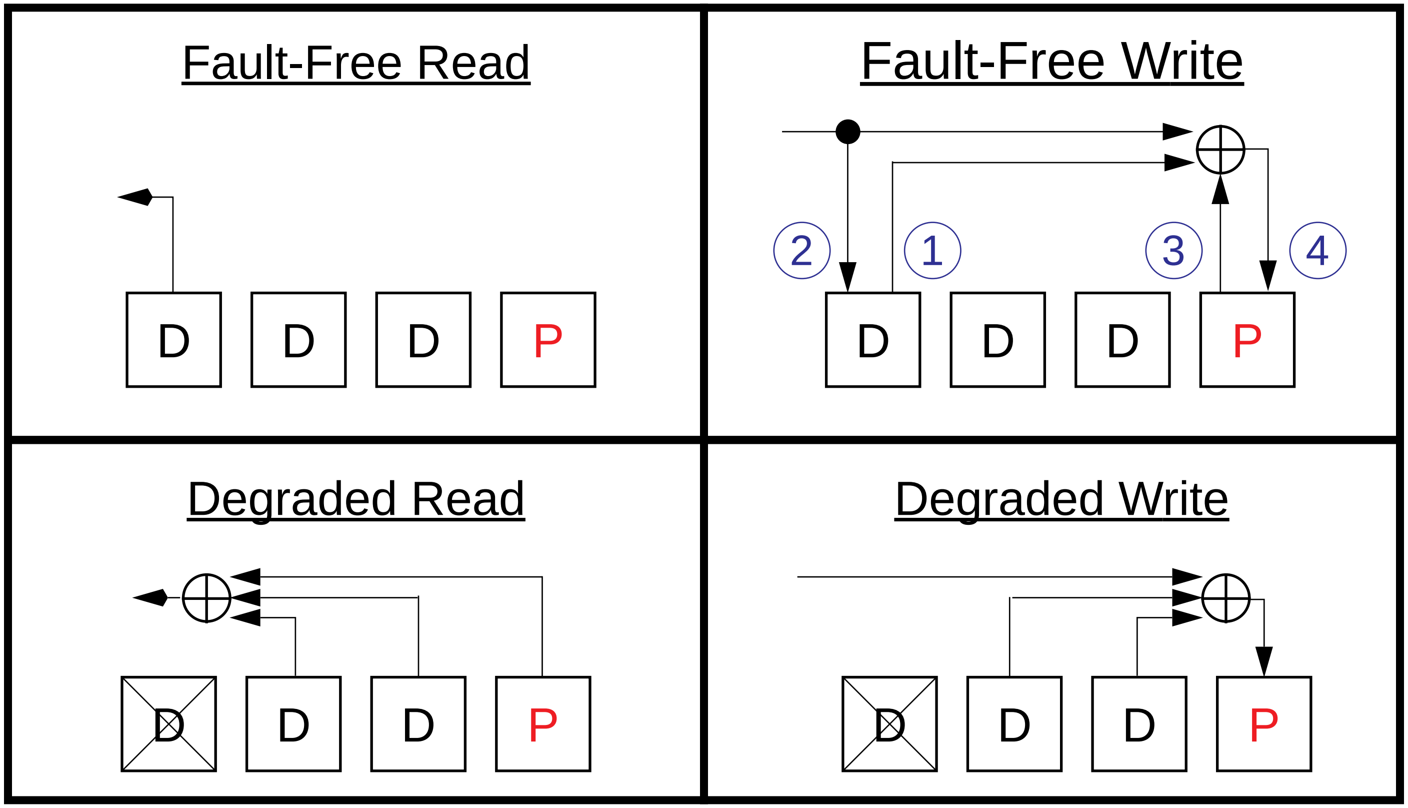
Writeback caching: optimize write-induced work

Parity logging: defer parity update until efficient

Floating data and parity: remap for fast R-M-W

Log-structured: convert small to large writes

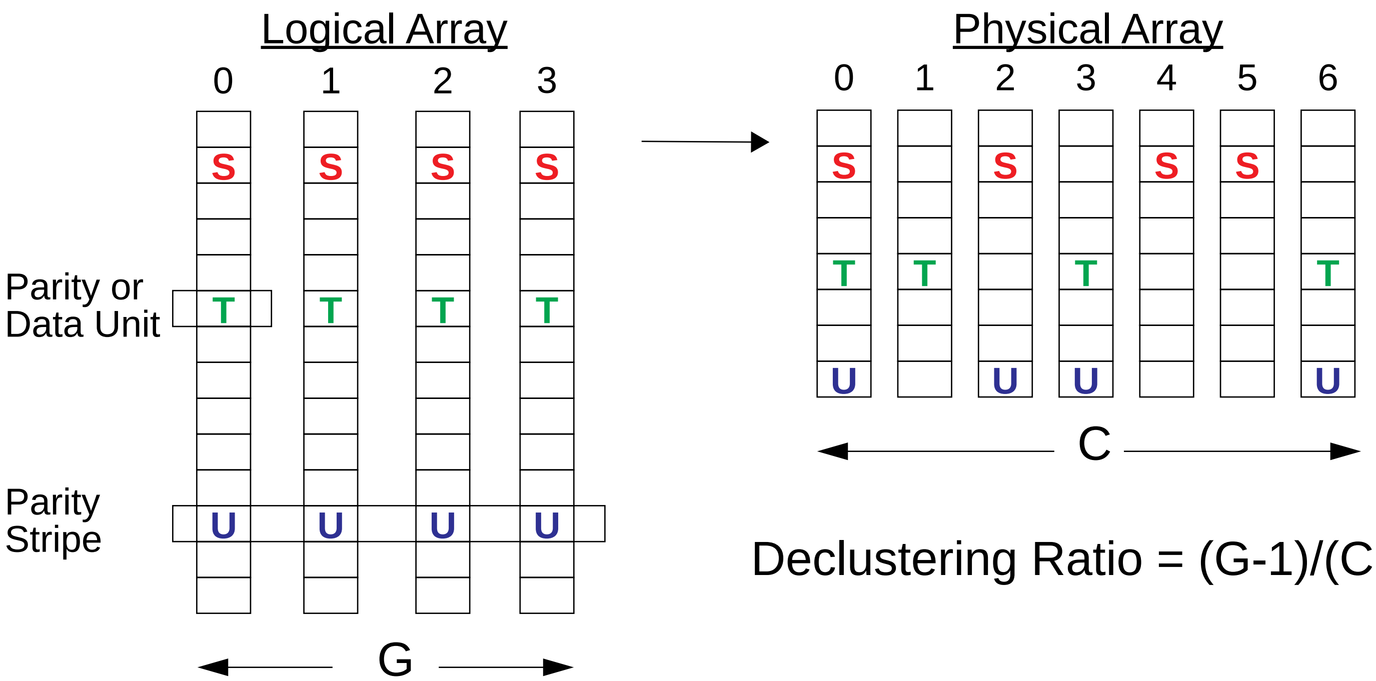
On-Line Failure Recovery Performance



Per-disk load increase in degraded mode ≈ 1 + *r* + 0.25*w*

- **50% throughput wall; long response times; long recovery time**

Reducing Load Increase: Parity Declustering



Declustering Ratio = (G-1)/(C-1)

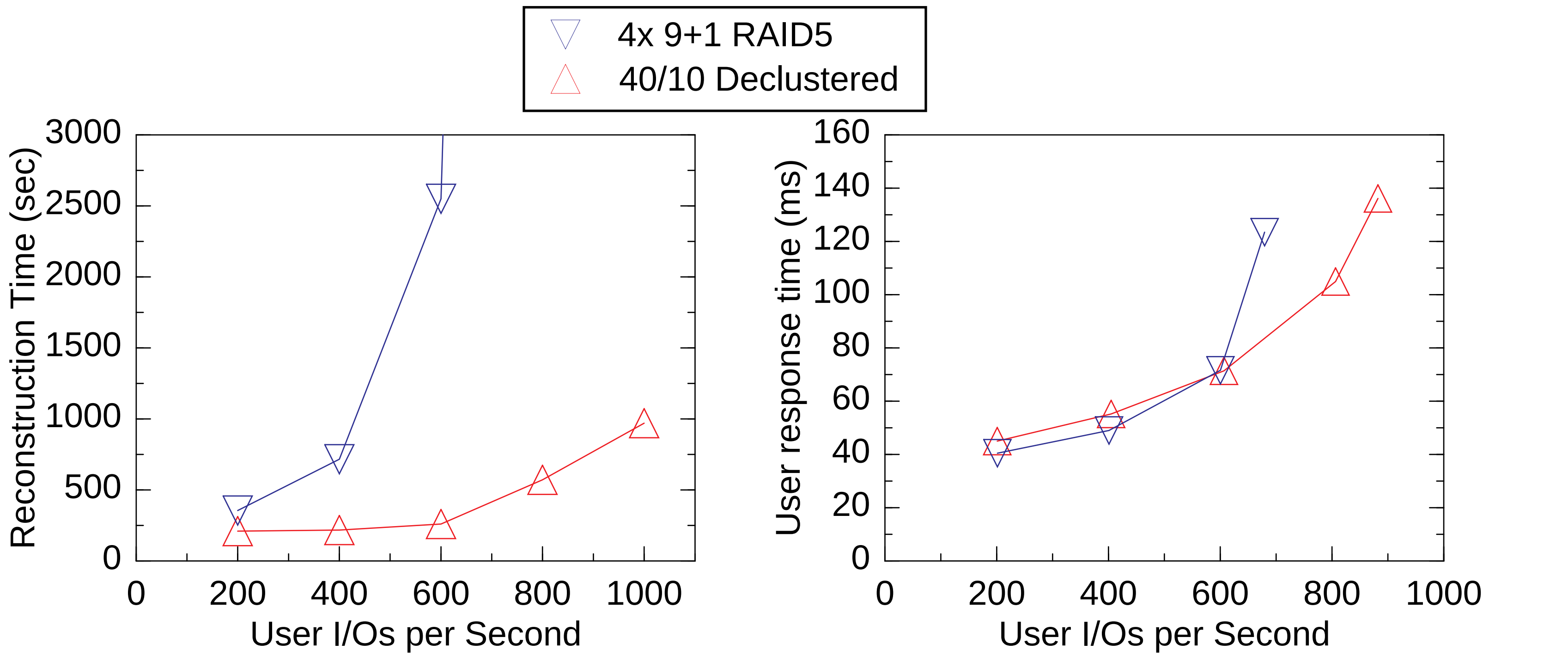
- **Per-disk failure-induced workload increase reduced**
- **Entire array bandwidth available for reconstruction**
- **Allows fault-free utilization > ~50%**
- **Map parity groups using *Balanced Incomplete Block Designs* or *Random Selection of Permutations***

Muntz, Lui, VLDB 1990. Holland, Gibson, ASPLOS V, 92.
Merchant, Yu, FTCS 92.

Comparing to Multiple RAID Level 5 Groups

RAID5: **4 groups of 9+1** $\Rightarrow$ **40 disks, 10% ovhd**
Declustered: **1 group of C=40/G=10** $\Rightarrow$ **40 disks, 10% ovhd**

Performance during reconstruction



Holland, Gibson, Siewiorek, J. Distributed and Parallel Databases, 94.

Exploiting Hot Spares: Distributed Sparing

Distribute spare disk space in same way as parity

- **Use spare actuator(s) to improve fault-free performance**
- **No need to reconstruct spare units**
- ***Alleviate spare-disk bottleneck at low declustering ratio***

RAID Level 5 with Dedicated Sparing

Disk Number

Offset	0	1	2	3	4
0	D	D	D	P	S
1	D	D	P	D	S
2	D	P	D	D	S
3	P	D	D	D	S
4	D	D	D	P	S

RAID Level 5 with Distributed Sparing

Disk Number

Offset	0	1	2	3	4
0	D	D	D	P	S
1	D	P	S	D	D
2	S	D	D	D	P
3	D	D	P	S	D
4	P	S	D	D	D

Menon, Mattson, ISCA, 92.

The Small-Write Throughput Problem

RAID Level 5 random small write cost 2X mirrors

- mirroring writes two copies: two disk accesses
- RAID 5 must toggle parity bits when data bits toggle costs read then write of data, read then write of parity: four disk accesses!

Small writes non-negligible: OLTP, Network FS

Major approaches

- Caching: delay writes in cache, schedule update later
- Logging: delay parity update in log, schedule update later
- Dynamic mapping: rearrange parity mapping as needed

- most require fault-tolerance of in-memory cache or mapping tables

Zero Latency Writes: Writeback Caching

Delay in NVRAM or in redundant controllers

Prioritize reads over delayed writes

Schedule writeback efficiently

- aggregate small writes into larger writes
- exploit idle time on actuator
- greedy Shortest Access Time First scheduling

Achieve deep queue, write costs 6-8 ms

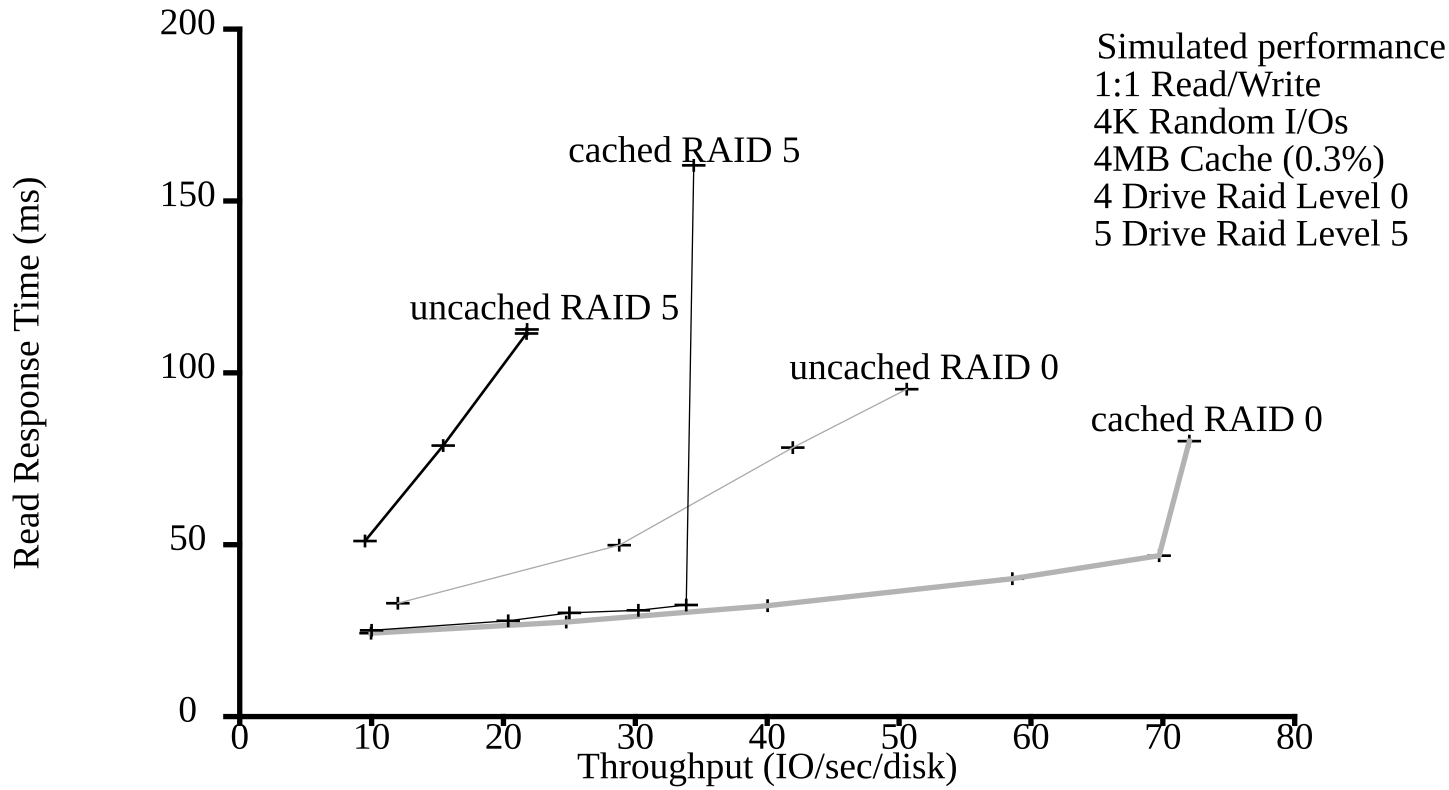
Fast writes -> shorter read behind write queueing

Widely used in most RAID products

Menon, Cortney, ISCA, 93.
Solworth, Orji, Sigmod, 90.

Read Performance given Zero Latency Writes

- **While idle available, read response independent of writeback work**
- **Idle exhausts according writeback work**



Parity Logging Extension to RAID Level 5

Read/Write data at block rates

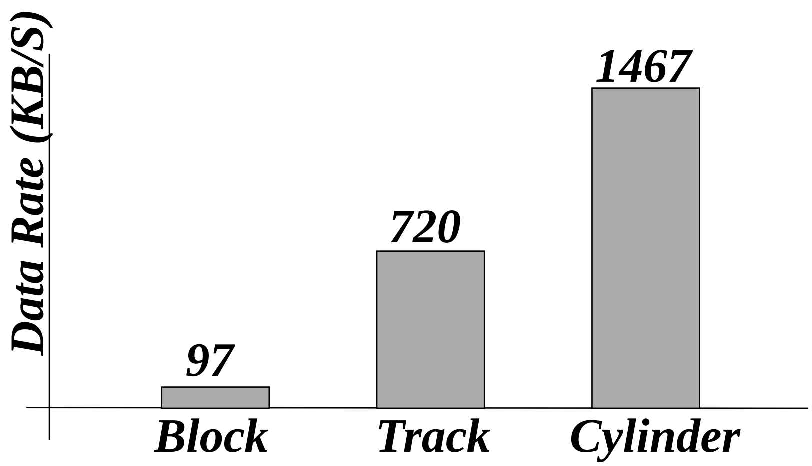
- **collect XOR of old and new data into "parity update log"**

Read/Write log and parity at cylinder rates

- **delay reintegration of parity updates into parity until efficient**

Adds 2 More I/Os but 4 I/Os are > 8 times faster

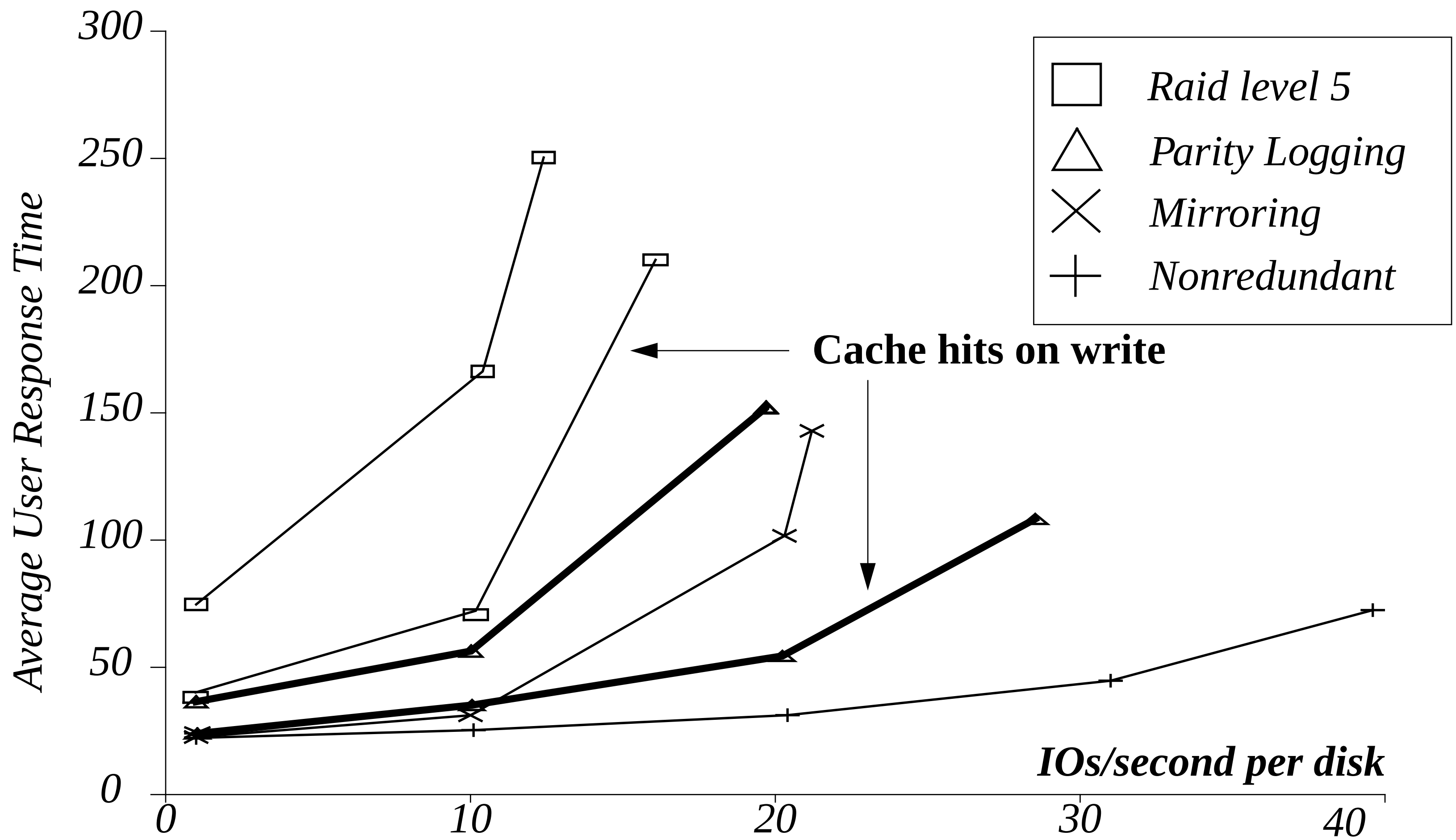
- **disk seconds spent for small writes comparable to mirroring**



Stodolsky, Courtright, Holland, Gibson, ACM Trans on Computer Systems, 94.
Bhide, Dias, IBM TR 17879, 92.

Logging Small Random Write Performance

Simulated results comparable to mirroring



Stodolsky, Courtright, Holland, Gibson, ACM Trans on Computer Systems, 94.

Dynamic Mapping: Floating Data and Parity

Reduce cost of one or both read-modify-write in small random overwrite

Dynamically remap "overwrite" to closest free space after read-modify

- 1.6 blocks to next free block if 7% space hidden from user
- read-modify-write response time is 10-20% longer than read-only
- disconnects contiguous data written at different times
- fault-tolerant mapping tables can be too large if data floats

Floating parity only with high cache hit ratio

- small random write complete in little more than 2 access times

Menon, Roche, Kasson, J. Parallel and Distributed Computing, 93.

Dynamic Mapping: Log-Structured

New write data is accumulated and written into an empty stripe

- large write optimization computes parity in memory

Background garbage collection of old data

Log-structured file system

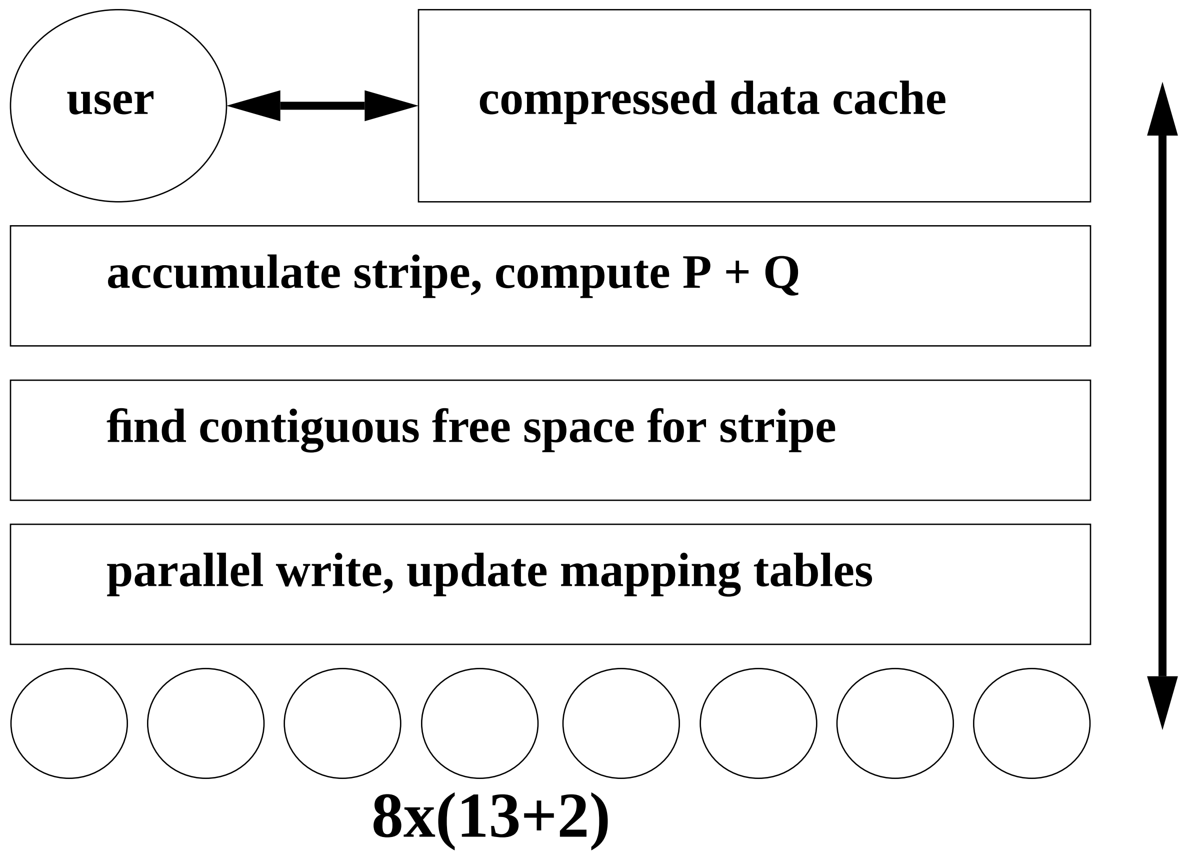
- focus on physical contiguity for bandwidth
- garbage collection involves copying

Virtual parity striping

- focus on minimal parity update cost
- garbage collection involves parity recomputation

Rosenblum, Ousterhout, Symp of Operating Systems Principles, 91.
Mogi, Kitsuregawa, Parallel and Distributed Info Systems, 94.

StorageTek Iceberg



Double-correct, log-structured, compressed data

Saunders, Storage Tech, RAID'95 Forum, April 95.

Recap RAID Performance

On-line recovery drastic reduction in performance

- read workload doubles on surviving disks in RAID 5
- declustering parity allows tunable degradation, cost
- declustering avoids bottlenecks in multi-group RAID 5
- on-line spares effective for very fast recovery

Reducing cost of small writes in parity-based arrays

- write caching for immediate ack, write scheduling
- logging parity changes for later efficient processing
- floating allocation for fast R-M-W
- dynamic remapping (log-structured) for large write optimization

Trends in Transparency

Intelligent storage management

Rapid development of architectures

Aggressive prefetching to increase I/O parallelism

Trends in Storage Organization Transparency

Increasing microprocessor power in controller

- controllers in 2000 expected to have 200 MHz processors

Cost of managing storage per year 7X storage cost

Strong push to embed management in storage

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

STK Iceberg, HP AutoRAID leading the push

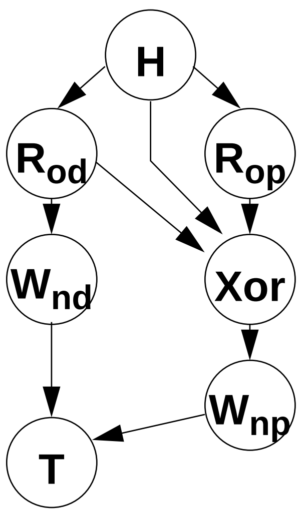
- dynamic migration of storage representation

Rapid Prototyping and Evaluation for RAID

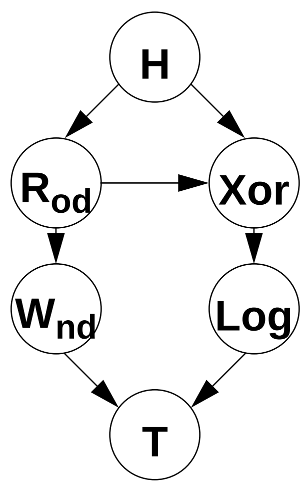
RAIDFrame: separate policy from mechanism

- **Express RAID functions as Directed Acyclic Graph**
- **Execute DAGs on engine unaware of RAID architecture**
- **Distributable, portable "RAID N reference model"**

RAID Level 5 Small Write DAG



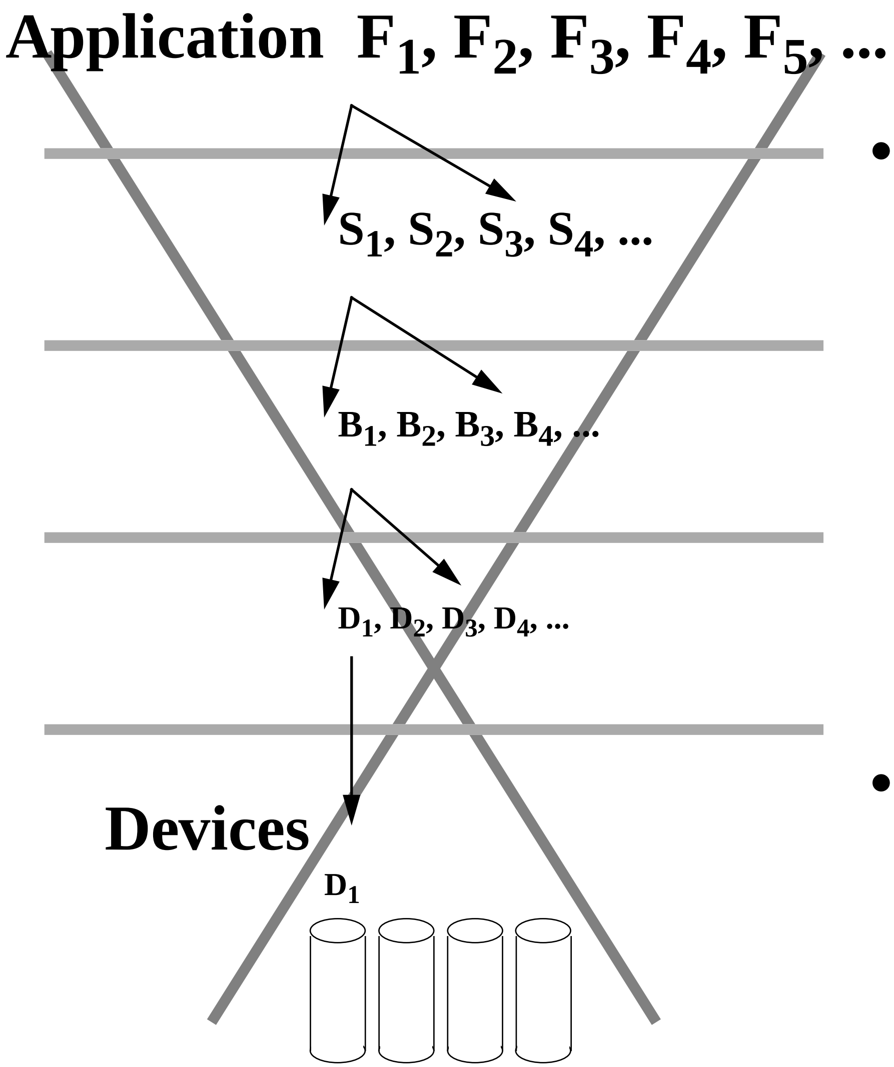
Parity Logging Small Write DAG



- **Roll-back and roll-forward error handling**
- **Automatic construction of uncommon DAGs**
- **Optimization of DAGs**

Gibson, et al, Compcon, 95.

Overcoming Disclosure Bottleneck: Prefetching



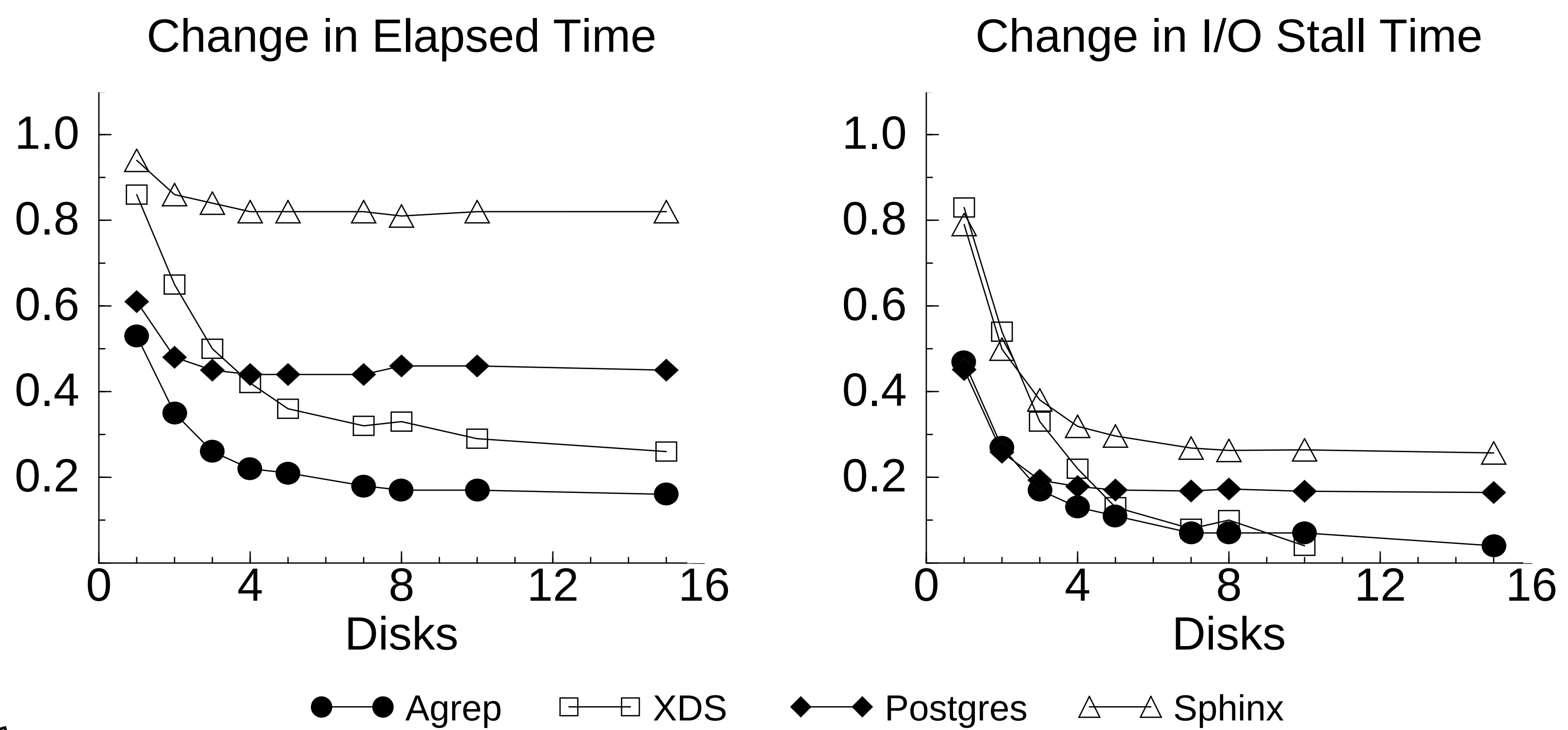
- **Expose concurrency**
  - **overlap I/O and computation**
  - **overlap I/O and think time**
  - **overlap I/O and I/O !!!!**
  - **I/O optimization**
    - seek scheduling
    - batch processing
- **Cache management**
  - **balance buffers between prefetch and demand**

Patterson, Gibson, Parallel and Distributed Information Systems, 94.

Examples: Informed Prefetching Filesystem

Annotated applications: text search, 3D visualization, database join, speech recognition

Digital UNIX (OSF/1 v2), 5 Fast-SCSI, 15 HP2247



Patterson, et al, CMU-CS-95-134, 95.

Recap Storage Transparency

Embedding intelligent storage management

- exploit embedded processor power to simplify configuration
- dynamic migration or representations within storage system

Tools for rapid RAID architecture development

- definition, evaluation, optimization tools

Beyond embarrassingly parallel I/O applications

- aggressive prefetching needed to reduce latency for many tasks

Trends in Network RAID

Avoid file server workstation's memory

- **Fastpath data from disk to network**

Handle controller failure as another disk failure

Large arrays needing more than 1 controller

Support client file access faster than 1 controller

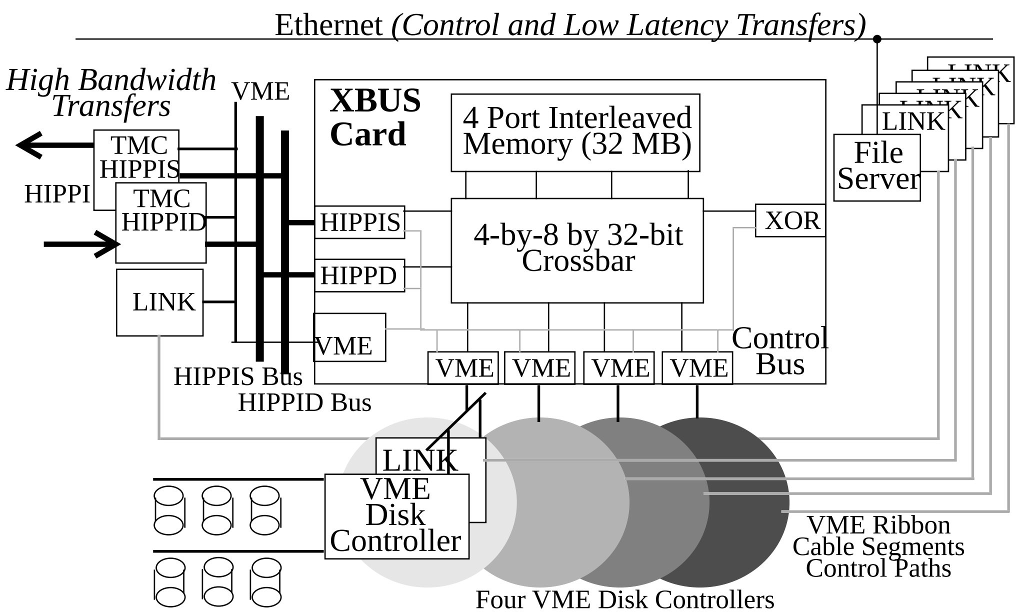
- **Stripe data on controllers & switched networks**

Fastpath Disk and Network

File server processor does not look at most bytes

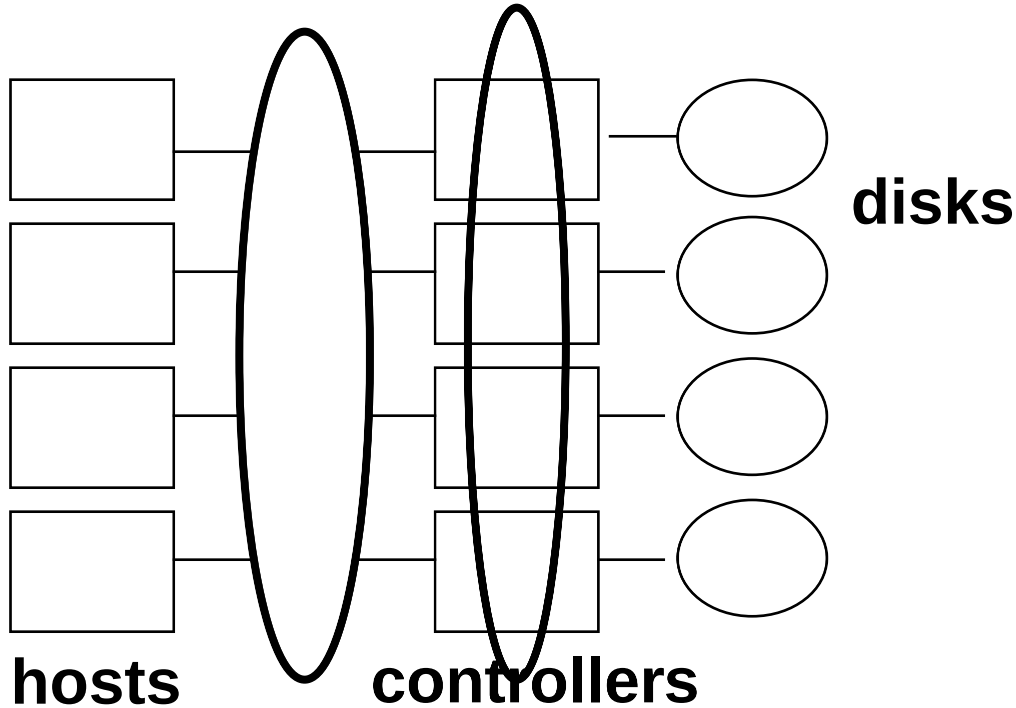
- **attach network and disks to RAID controller with fast bus**
- **scale file network bandwidth and file server CPU separately**
- **separate high bandwidth and low latency traffic on own net**

Berkeley RAID-II prototype: 15-21 MB/s thru LFS



Drapeau, et al., ISCA, 94.

Embedded Controller Parallelism



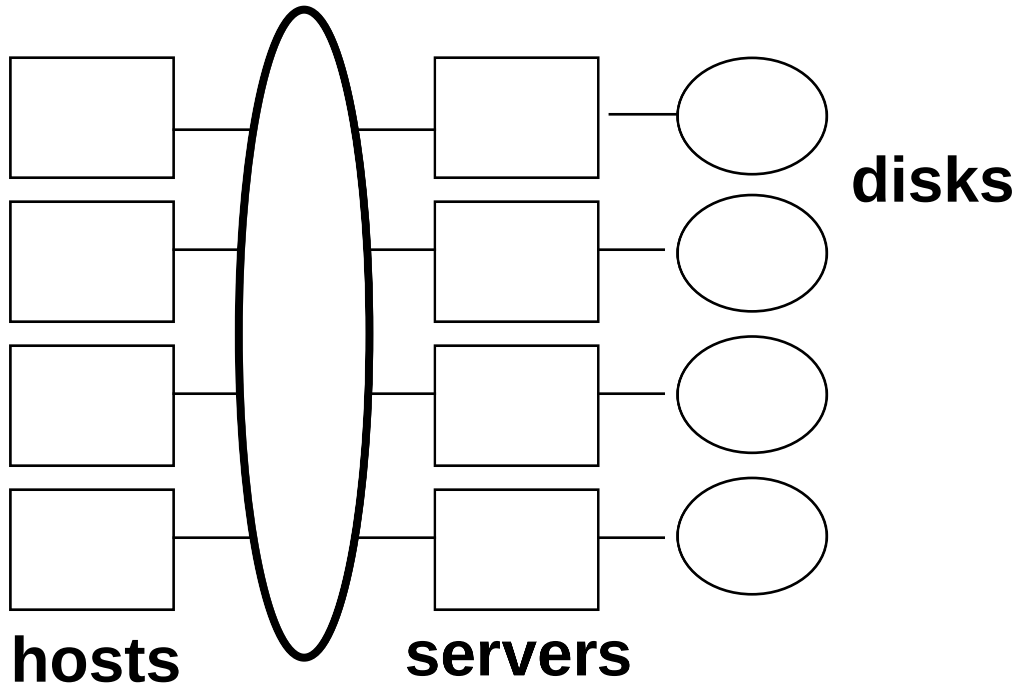
HP TickerTaip - private controller network

- scale controller compute power; tolerate controller failures

Borg Technologies promises similar product

Cao, Lim, Venkataraman, Wilkes, ISCA, 93.
D. Stallmo, Borg Tech, RAID'95 Forum, April 95.

Exposed Controller Parallelism

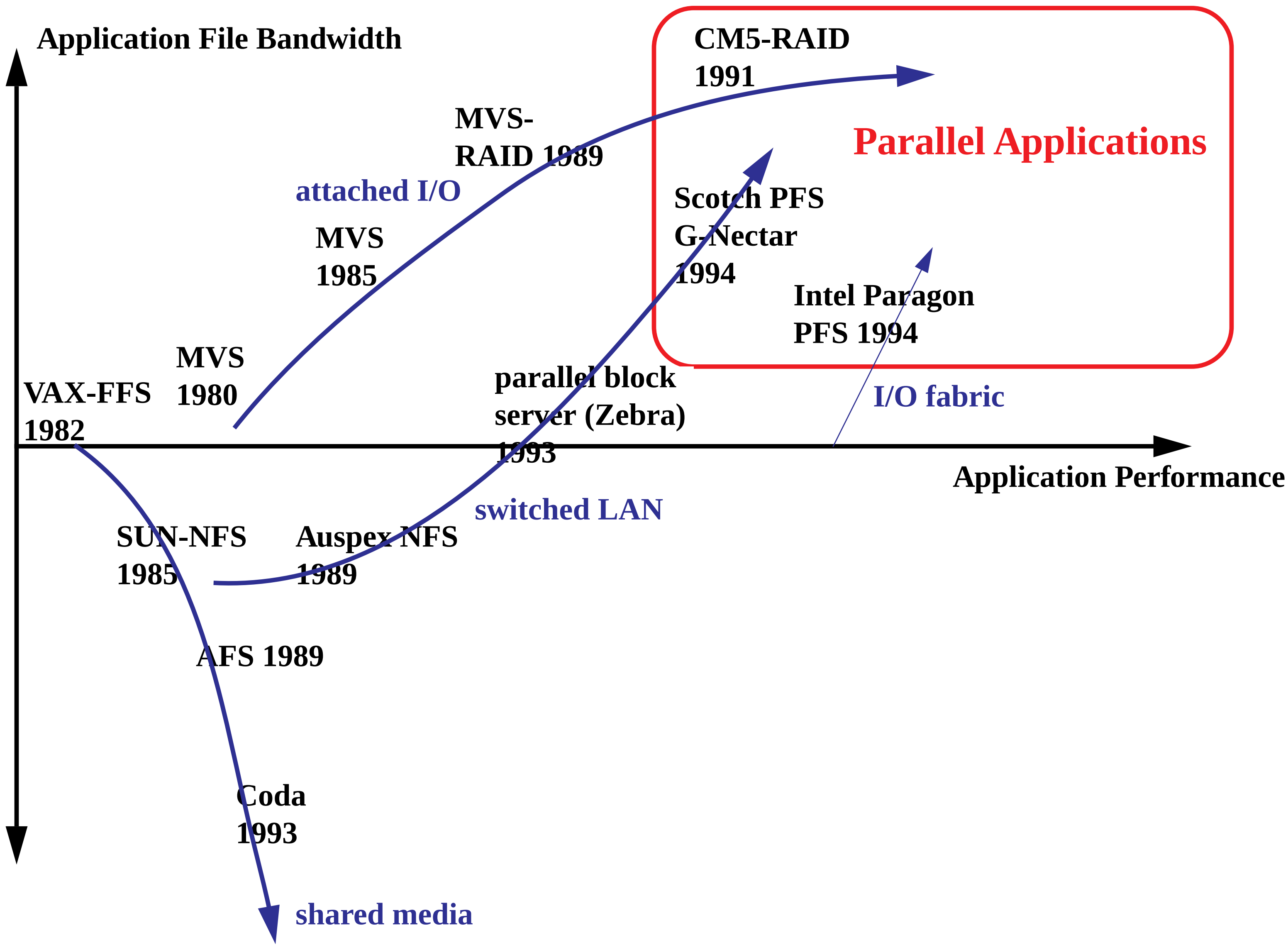


Use host network for controller/server traffic

- **Zebra - log-structured file system over network**
- **Swift - distributed server RAID**
- **Network-attached disk - Conner (ATM), Seagate (FibreChannel)**

Hartman, Ousterhout, Symposium on Operating Systems Principles, 93.
Cabrera, Long, Computing Systems, 91.

Opportunity for Parallelism Increasing



Parallel File Systems for Parallel Programs

- **Concurrent write sharing**
  - **Globally shared file pointers**
- **Performance hungry applications**
  - **High bandwidth to large files**
  - **Application-specific access methods**
  - **Application control over basic PFS parameters**
- **Limited instances of any specific environment**
  - **Emphasis on scalability and portability**
  - **How much integration with network?**

Corbett, Feitelson, Proc Scalable High-Performance Computing Conf, 94.
Gibson, et.al., Compcon, 95.

Recap RAID over the Network

Attach storage "closer" to network

- avoid workstation memory system

Stripe data over multiple controllers/servers

- private controller network in the box
- use host LAN for redundancy and controller communication

Support explicit concurrent write sharing

- parallel file systems for parallel programs
- application assistance in data layout, prefetching, checkpointing

Tutorial Summary

- Basic RAID: Levels 1, 3, 5 useful
- Magnetic disk technology stronger than ever
- RAID market well beyond basic RAID
- Increasingly sophisticated function in subsystem
- How much transparency is too much?
- Striping/RAID over network emerging