A Case for Redundant Arrays of Inexpensive Disks (RAID)

David A. Patterson, Garth Gibson, and Randy H. Katz

Computer Science Division
Department of Electrical Engineering and Computer Sciences
571 Evans Hall
University of California
Berkeley, CA 94720
(pattrsn@ginger berkeley edu)

Abstract Increasing performance of CPUs and memories will be squandered if not matched by a similar performance increase in I/O While the capacity of Single Large Expensive Disks (SLED) has grown rapidly, the performance improvement of SLED has been modest Redundant Arrays of Inexpensive Disks (RAID), based on the magnetic disk technology developed for personal computers, offers an attractive alternative to SLED, promising improvements of an order of magnitude in performance, reliability, power consumption, and scalability This paper introduces five levels of RAIDs, giving their relative cost/performance, and compares RAID to an IBM 3380 and a Fujitsu Super Eagle

1 Background: Rising CPU and Memory Performance

The users of computers are currently enjoying unprecedented growth in the speed of computers Gordon Bell said that between 1974 and 1984, single chip computers improved in performance by 40% per year, about twice the rate of minicomputers [Bell 84] In the following year Bill Joy predicted an even faster growth [Joy 85]

$$MIPS = 2^{Year-1984}$$

Mainframe and supercomputer manufacturers, having difficulty keeping pace with the rapid growth predicted by "Joy's Law," cope by offering multiprocessors as their top-of-the-line product.

But a fast CPU does not a fast system make Gene Amdahl related CPU speed to main memory size using this rule [Siewiorek 82]

Each CPU instruction per second requires one byte of main memory,

If computer system costs are not to be dominated by the cost of memory, then Amdahl's constant suggests that memory chip capacity should grow at the same rate Gordon Moore predicted that growth rate over 20 years ago

$$transistors/chip = 2^{Year-1964}$$

As predicted by Moore's Law, RAMs have quadrupled in capacity every two [Moore 75] to three years [Myers 86]

Recently the ratio of megabytes of main memory to MIPS has been defined as *alpha* [Garcia 84], with Amdahl's constant meaning alpha = 1 In part because of the rapid drop of memory prices, main memory sizes have grown faster than CPU speeds and many machines are shipped today with alphas of 3 or higher

To maintain the balance of costs in computer systems, secondary storage must match the advances in other parts of the system A key measure of magnetic disk technology is the growth in the maximum number of bits that can be stored per square inch, or the bits per inch in a track times the number of tracks per inch Called M A D, for maximal areal density, the "First Law in Disk Density" predicts [Frank87]

$$MAD = 10^{(Year-1971)/10}$$

Magnetic disk technology has doubled capacity and halved price every three years, in line with the growth rate of semiconductor memory, and in practice between 1967 and 1979 the disk capacity of the average IBM data processing system more than kept up with its main memory [Stevens81]

Capacity is not the only memory characteristic that must grow rapidly to maintain system balance, since the speed with which instructions and data are delivered to a CPU also determines its ultimate performance The speed of main memory has kept pace for two reasons
(1) the invention of caches, showing that a small buffer can be managed automatically to contain a substantial fraction of memory references,
(2) and the SRAM technology, used to build caches, whose speed has improved at the rate of 40% to 100% per year

In contrast to primary memory technologies, the performance of single large expensive magnetic disks (SLED) has improved at a modest rate These *mechanical* devices are dominated by the seek and the rotation delays from 1971 to 1981, the raw seek time for a high-end IBM disk improved by only a factor of two while the rotation time did not change[Harker81] Greater density means a higher transfer rate when the information is found, and extra heads can reduce the average seek time, but the raw seek time only improved at a rate of 7% per year There is no reason to expect a faster rate in the near future

To maintain balance, computer systems have been using even larger main memories or solid state disks to buffer some of the I/O activity This may be a fine solution for applications whose I/O activity has locality of reference and for which volatility is not an issue, but applications dominated by a high rate of random requests for small pieces of data (such as transaction-processing) or by a low number of requests for massive amounts of data (such as large simulations running on supercomputers) are facing a serious performance limitation

2. The Pending I/O Crisis

What is the impact of improving the performance of some pieces of a problem while leaving others the same? Amdahl's answer is now known as Amdahl's Law [Amdahl67]

$$S = \frac{1}{(1-f) + f/k}$$

where
S = the effective speedup,
f = fraction of work in faster mode, and
k = speedup while in faster mode

Suppose that some current applications spend 10% of their time in I/O Then when computers are 10X faster--according to Bill Joy in just over three years--then Amdahl's Law predicts effective speedup will be only 5X When we have computers 100X faster--via evolution of uniprocessors or by multiprocessors--this application will be less than 10X faster, wasting 90% of the potential speedup

While we can imagine improvements in software file systems via buffering for near term I/O demands, we need innovation to avoid an I/O crisis [Boral 83]

3   A Solution: Arrays of Inexpensive Disks

Rapid improvements in capacity of large disks have not been the only target of disk designers, since personal computers have created a market for inexpensive magnetic disks These lower cost disks have lower performance as well as less capacity Table I below compares the top-of-the-line IBM 3380 model AK4 mainframe disk, Fujitsu M2361A "Super Eagle" minicomputer disk, and the Conner Peripherals CP 3100 personal computer disk

Characteristics	IBM 3380	Fujitsu M2361A	Conners CP3100	3380 v 3100 (>1 means 3100 is better)	2361 v 3100
Disk diameter (inches)	14	10 5	3 5	4	3
Formatted Data Capacity (MB)	7500	600	100	01	2
Price/MB(controller incl)	$18-$10	$20-$17	$10-$7	1-2 5	1 7-3
MTTF Rated (hours)	30,000	20,000	30,000	1	1 5
MTTF in practice (hours)	100,000	?	?	?	?
No Actuators	4	1	1	2	1
Maximum I/O's/second/Actuator	50	40	30	6	8
Typical I/O's/second/Actuator	30	24	20	7	8
Maximum I/O's/second/box	200	40	30	2	8
Typical I/O's/second/box	120	24	20	2	8
Transfer Rate (MB/sec)	3	2 5	1	3	4
Power/box (W)	6,600	640	10	660	64
Volume (cu ft)	24	3 4	03	800	110

Table I *Comparison of IBM 3380 disk model AK4 for mainframe computers, the Fujitsu M2361A "Super Eagle" disk for minicomputers, and the Conners Peripherals CP 3100 disk for personal computers By "Maximum I/O's/second" we mean the maximum number of average seeks and average rotates for a single sector access Cost and reliability information on the 3380 comes from widespread experience [IBM 87] [Gawlick87] and the information on the Fujitsu from the manual [Fujitsu 87], while some numbers on the new CP3100 are based on speculation The price per megabyte is given as a range to allow for different prices for volume discount and different mark-up practices of the vendors (The 8 watt maximum power of the CP3100 was increased to 10 watts to allow for the inefficiency of an external power supply, since the other drives contain their own power supplies)*

One surprising fact is that the number of I/Os per second per actuator in an inexpensive disk is within a factor of two of the large disks In several of the remaining metrics, including price per megabyte, the inexpensive disk is superior or equal to the large disks

The small size and low power are even more impressive since disks such as the CP3100 contain full track buffers *and* most functions of the traditional mainframe controller Small disk manufacturers can provide such functions in high volume disks because of the efforts of standards committees in defining higher level peripheral interfaces, such as the ANSI X3 131-1986 Small Computer System Interface (SCSI) Such standards have encouraged companies like Adeptec to offer SCSI interfaces as single chips, in turn allowing disk companies to *embed* mainframe controller functions at low cost Figure 1 compares the traditional mainframe disk approach and the small computer disk approach The same SCSI interface chip embedded as a controller in every disk can also be used as the direct memory access (DMA) device at the other end of the SCSI bus

Such characteristics lead to our proposal for building I/O systems as arrays of inexpensive disks, either interleaved for the large transfers of supercomputers [Kim 86][Livny 87][Salem86] or independent for the many small transfers of transaction processing Using the information in Table I, 75 inexpensive disks potentially have 12 times the I/O bandwidth of the IBM 3380 and the same capacity, with lower power consumption and cost

4   Caveats

We cannot explore all issues associated with such arrays in the space available for this paper, so we concentrate on fundamental estimates of

price-performance and reliability Our reasoning is that if there are no advantages in price-performance or terrible disadvantages in reliability, then there is no need to explore further We characterize a transaction-processing workload to evaluate performance of a collection of inexpensive disks, but remember that such a collection is just one hardware component of a complete tranaction-processing system While designing a complete TPS based on these ideas is enticing, we will resist that temptation in this paper Cabling and packaging, certainly an issue in the cost and reliability of an array of many inexpensive disks, is also beyond this paper's scope
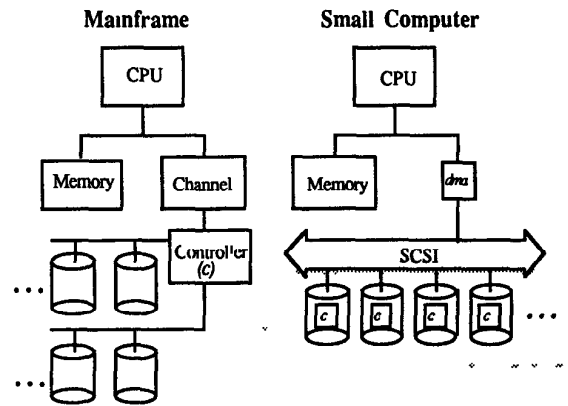

Figure 1 *Comparison of organizations for typical mainframe and small computer disk interfaces Single chip SCSI interfaces such as the Adaptec AIC-6250 allow the small computer to use a single chip to be the DMA interface as well as provide an embedded controller for each disk [Adeptec 87] (The price per megabyte in Table I includes everything in the shaded boxes above)*

5.   And Now The Bad News: Reliability

The unreliability of disks forces computer systems managers to make backup versions of information quite frequently in case of failure What would be the impact on reliability of having a hundredfold increase in disks? Assuming a constant failure rate--that is, an exponentially distributed time to failure--and that failures are independent--both assumptions made by disk manufacturers when calculating the Mean Time To Failure (MTTF)--the reliability of an array of disks is

$$MTTF \ of \ a \ Disk \ Array \ = \ \frac{MTTF \ of \ a \ Single \ Disk}{Number \ of \ Disks \ in \ the \ Array}$$

Using the information in Table I, the MTTF of 100 CP 3100 disks is 30,000/100 = 300 hours, or less than 2 weeks Compared to the 30,000 hour (> 3 years) MTTF of the IBM 3380, this is dismal If we consider scaling the array to 1000 disks, then the MTTF is 30 hours or about one day, requiring an adjective worse than dismal

Without fault tolerance, large arrays of inexpensive disks are too unreliable to be useful

6.   A Better Solution: RAID

To overcome the reliability challenge, we must make use of extra disks containing redundant information to recover the original information when a disk fails Our acronym for these Redundant Arrays of Inexpensive Disks is *RAID* To simplify the explanation of our final proposal and to avoid confusion with previous work, we give a taxonomy of five different organizations of disk arrays, beginning with mirrored disks and progressing through a variety of alternatives with differing performance and reliability We refer to each organization as a RAID *level*

The reader should be forewarned that we describe all levels as if implemented in hardware solely to simplify the presentation, for RAID ideas are applicable to software implementations as well as hardware

Reliability Our basic approach will be to break the arrays into reliability groups, with each group having extra "check" disks containing redundant information When a disk fails we assume that within a short time the failed disk will be replaced and the information will be

reconstructed on to the new disk using the redundant information This time is called the mean time to repair (MTTR) The MTTR can be reduced if the system includes extra disks to act as "hot" standby spares, when a disk fails, a replacement disk is switched in electronically Periodically a human operator replaces all failed disks Here are other terms that we use

D = total number of disks with data (not including extra check disks),

G = number of data disks in a *group* (not including extra check disks),

C = number of check disks in a group,

n_G = D/G = number of groups,

As mentioned above we make the same assumptions that disk manufacturers make--that failures are exponential and independent (An earthquake or power surge is a situation where an array of disks might not fail independently) Since these reliability predictions will be very high, we want to emphasize that the reliability is only of the the disk-head assemblies with this failure model, and not the whole software and electronic system In addition, in our view the pace of technology means extremely high MTTF are "overkill"--for, independent of expected lifetime, users will replace obsolete disks After all, how many people are still using 20 year old disks[7]

The general MTTF calculation for single-error repairing RAID is given in two steps First, the group MTTF is

$$MTTF_{Group} = \frac{MTTF_{Disk}}{G+C} * \frac{1}{\text{Probability of another failure in a group before repairing the dead disk}}$$

As more formally derived in the appendix, the probability of a second failure before the first has been repaired is

$$\text{Probability of Another Failure} = \frac{MTTR}{MTTF_{Disk}/(No\ Disks-1)} = \frac{MTTR}{MTTF_{Disk}/(G+C-1)}$$

The intuition behind the formal calculation in the appendix comes from trying to calculate the average number of second disk failures during the repair time for X single disk failures Since we assume that disk failures occur at a uniform rate, this average number of second failures during the repair time for X first failures is

$$\frac{X * MTTR}{MTTF\ of\ remaining\ disks\ in\ the\ group}$$

The average number of second failures for a single disk is then

$$\frac{MTTR}{MTTF_{Disk}/No\ of\ remaining\ disks\ in\ the\ group}$$

The MTTF of the remaining disks is just the MTTF of a single disk divided by the number of good disks in the group, giving the result above

The second step is the reliability of the whole system, which is approximately (since $MTTF_{Group}$ is not quite distributed exponentially)

$$MTTF_{RAID} = \frac{MTTF_{Group}}{n_G}$$

Plugging it all together, we get

$$MTTF_{RAID} = \frac{MTTF_{Disk}}{G+C} * \frac{MTTF_{Disk}}{(G+C-1)*MTTR} * \frac{1}{n_G}$$

$$= \frac{(MTTF_{Disk})^2}{(G+C)*n_G * (G+C-1)*MTTR}$$

$$MTTF_{RAID} = \frac{(MTTF_{Disk})^2}{(D+C*n_G)*(G+C-1)*MTTR}$$

Since the formula is the same for each level, we make the abstract numbers concrete using these parameters as appropriate $D=100$ total data disks, $G=10$ data disks per group, $MTTF_{Disk}$ = 30,000 hours, $MTTR$ = 1 hour, with the check disks per group C determined by the RAID level

Reliability Overhead Cost This is simply the extra check disks, expressed as a percentage of the number of data disks D As we shall see below, the cost varies with RAID level from 100% down to 4%

Useable Storage Capacity Percentage Another way to express this reliability overhead is in terms of the percentage of the total capacity of data disks *and* check disks that can be used to store data Depending on the organization, this varies from a low of 50% to a high of 96%

Performance Since supercomputer applications and transaction-processing systems have different access patterns and rates, we need different metrics to evaluate both For supercomputers we count the number of reads and writes per second for large blocks of data, with large defined as getting at least one sector from each data disk in a group During large transfers all the disks in a group act as a single unit, each reading or writing a portion of the large data block in parallel

A better measure for transaction-processing systems is the number of individual reads or writes per second Since transaction-processing systems (e g , debits/credits) use a read-modify-write sequence of disk accesses, we include that metric as well Ideally during small transfers each disk in a group can act independently, either reading or writing independent information In summary supercomputer applications need a *high data rate* while transaction-processing need a *high I/O rate*

For both the large and small transfer calculations we assume the minimum user request is a sector, that a sector is small relative to a track, and that there is enough work to keep every device busy Thus sector size affects both disk storage efficiency and transfer size Figure 2 shows the ideal operation of large and small disk accesses in a RAID
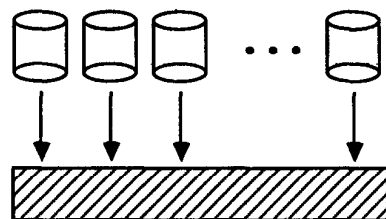
(a) Single Large or "Grouped" Read
(1 read spread over G disks)
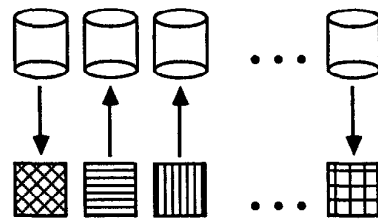
(b) Several Small or Individual Reads and Writes
(G reads and/or writes spread over G disks)

Figure 2. *Large transfer vs small transfers in a group of G disks*

The six performance metrics are then the number of reads, writes, and read-modify-writes per second for both large (grouped) or small (individual) transfers Rather than give absolute numbers for each metric, we calculate efficiency the number of events per second for a RAID relative to the corresponding events per second for a single disk (This is Boral's I/O bandwidth per gigabyte [Boral 83] scaled to gigabytes per disk) In this paper we are after fundamental differences so we use simple, deterministic throughput measures for our performance metric rather than latency

Effective Performance Per Disk The cost of disks can be a large portion of the cost of a database system, so the I/O performance per disk--factoring in the overhead of the check disks--suggests the cost/performance of a system This is the bottom line for a RAID

111

7. First Level RAID: Mirrored Disks

Mirrored disks are a traditional approach for improving reliability of magnetic disks. This is the most expensive option we consider since all disks are duplicated ($G=1$ and $C=1$), and every write to a data disk is also a write to a check disk. Tandem doubles the number of controllers for fault tolerance, allowing an optimized version of mirrored disks that lets reads occur in parallel. Table II shows the metrics for a Level 1 RAID assuming this optimization.

MTTF	Exceeds Useful Product Lifetime (4,500,000 hrs or > 500 years)	
Total Number of Disks	2D	
Overhead Cost	100%	
Useable Storage Capacity	50%	
Events/Sec vs Single Disk	Full RAID	Efficiency Per Disk
Large (or Grouped) Reads	2D/S	1 00/S
Large (or Grouped) Writes	D/S	50/S
Large (or Grouped) R-M-W	4D/3S	67/S
Small (or Individual) Reads	2D	1 00
Small (or Individual) Writes	D	50
Small (or Individual) R-M-W	4D/3	67

Table II. *Characteristics of Level 1 RAID Here we assume that writes are not slowed by waiting for the second write to complete because the slowdown for writing 2 disks is minor compared to the slowdown S for writing a whole group of 10 to 25 disks Unlike a "pure" mirrored scheme with extra disks that are invisible to the software, we assume an optimized scheme with twice as many controllers allowing parallel reads to all disks, giving full disk bandwidth for large reads and allowing the reads of read-modify-writes to occur in parallel*

When individual accesses are distributed across multiple disks, average queueing, seek, and rotate delays may differ from the single disk case. Although bandwidth may be unchanged, it is distributed more evenly, reducing variance in queueing delay and, if the disk load is not too high, also reducing the expected queueing delay through parallelism [Livny 87]. When many arms seek to the same track then rotate to the described sector, the average seek and rotate time will be larger than the average for a single disk, tending toward the worst case times. This affect should not generally more than double the average access time to a single sector while still getting many sectors in parallel. In the special case of mirrored disks with sufficient controllers, the choice between arms that can read any data sector will reduce the time for the average read seek by up to 45% [Bitton 88].

To allow for these factors but to retain our fundamental emphasis we apply a slowdown factor, S, when there are more than two disks in a group. In general, $1 \leq S \leq 2$ whenever groups of disk work in parallel. With synchronous disks the spindles of all disks in the group are synchronous so that the corresponding sectors of a group of disks pass under the heads simultaneously,[Kurzweil 88] so for synchronous disks there is no slowdown and $S = 1$. Since a Level 1 RAID has only one data disk in its group, we assume that the large transfer requires the same number of disks acting in concert as found in groups of the higher level RAIDs 10 to 25 disks.

Duplicating all disks can mean doubling the cost of the database system or using only 50% of the disk storage capacity. Such largess inspires the next levels of RAID.

8 Second Level RAID: Hamming Code for ECC

The history of main memory organizations suggests a way to reduce the cost of reliability. With the introduction of 4K and 16K DRAMs, computer designers discovered that these new devices were subject to losing information due to alpha particles. Since there were many single bit DRAMs in a system and since they were usually accessed in groups of 16 to 64 chips at a time, system designers added redundant chips to correct single errors and to detect double errors in each group. This increased the number of memory chips by 12% to 38%--depending on the size of the group--but it significantly improved reliability.

As long as all the data bits in a group are read or written together, there is no impact on performance. However, reads of less than the group size require reading the whole group to be sure the information is correct, and writes to a portion of the group mean three steps

1) a read step to get all the rest of the data,
2) a modify step to merge the new and old information,
3) a write step to write the full group, including check information

Since we have scores of disks in a RAID and since some accesses are to groups of disks, we can mimic the DRAM solution by bit-interleaving the data across the disks of a group and then add enough check disks to detect and correct a single error. A single parity disk can detect a single error, but to correct an error we need enough check disks to identify the disk with the error. For a group size of 10 data disks (G) we need 4 check disks (C) in total, and if $G = 25$ then $C = 5$ [Hamming50]. To keep down the cost of redundancy, we assume the group size will vary from 10 to 25.

Since our individual data transfer unit is just a sector, bit-interleaved disks mean that a large transfer for this RAID must be at least G sectors. Like DRAMs, reads to a smaller amount implies reading a full sector from each of the bit-interleaved disks in a group, and writes of a single unit involve the read-modify-write cycle to all the disks. Table III shows the metrics of this Level 2 RAID

MTTF		Exceeds Useful Lifetime			
		G=10 (494,500 hrs or >50 years)		G=25 (103,500 hrs or 12 years)	
Total Number of Disks		1 40D		1.20D	
Overhead Cost		40%		20%	
Useable Storage Capacity		71%		83%	
Events/Sec	Full RAID	Efficiency Per Disk		Efficiency Per Disk	
(vs Single Disk)		L2	L2/L1	L2	L2/L1
Large Reads	D/S	71/S	71%	86/S	86%
Large Writes	D/S	71/S	143%	86/S	172%
Large R-M-W	D/S	71/S	107%	86/S	129%
Small Reads	D/SG	07/S	6%	03/S	3%
Small Writes	D/2SG	04/S	6%	02/S	3%
Small R-M-W	D/SG	07/S	9%	03/S	4%

Table III *Characteristics of a Level 2 RAID The L2/L1 column gives the % performance of level 2 in terms of level 1 (>100% means L2 is faster) As long as the transfer unit is large enough to spread over all the data disks of a group, the large I/Os get the full bandwidth of each disk, divided by S to allow all disks in a group to complete Level 1 large reads are faster because data is duplicated and so the redundancy disks can also do independent accesses Small I/Os still require accessing all the disks in a group, so only D/G small I/O can happen at a time, again divided by S to allow a group of disks to finish Small Level 2 writes are like small R-M-W because full sectors must be read before new data can be written onto part of each sector*

For large writes, the level 2 system has the same performance as level 1 even though it uses fewer check disks, and so on a per disk basis it outperforms level 1. For small data transfers the performance is dismal either for the whole system or per disk, all the disks of a group must be accessed for a small transfer, limiting the maximum number of simultaneous accesses to D/G. We also include the slowdown factor S since the access must wait for all the disks to complete

Thus level 2 RAID is desirable for supercomputers but inappropriate for transaction processing systems, with increasing group size increasing the disparity in performance per disk for the two applications. In recognition of this fact, Thinking Machines Incorporated announced a Level 2 RAID this year for its Connection Machine supercomputer called the "Data Vault," with $G = 32$ and $C = 8$, including one hot standby spare [Hillis 87]

Before improving small data transfers, we concentrate once more on lowering the cost

9 Third Level RAID: Single Check Disk Per Group

Most check disks in the level 2 RAID are used to determine which disk failed, for only one redundant parity disk is needed to detect an error. These extra disks are truly "redundant" since most disk controllers can already detect if a disk failed either through special signals provided in the disk interface or the extra checking information at the end of a sector used to detect and correct soft errors. So information on the failed disk can be reconstructed by calculating the parity of the remaining good disks and then comparing bit-by-bit to the parity calculated for the original full

group When these two parities agree, the failed bit was a 0, otherwise it was a 1 If the check disk is the failure, just read all the data disks and store the group parity in the replacement disk

Reducing the check disks to one per group (C=1) reduces the overhead cost to between 4% and 10% for the group sizes considered here The performance for the third level RAID system is the same as the Level 2 RAID, but the effective performance per disk increases since it needs fewer check disks This reduction in total disks also increases reliability, but since it is still larger than the useful lifetime of disks, this is a minor point One advantage of a level 2 system over level 3 is that the extra check information associated with each sector to correct soft errors is not needed, increasing the capacity per disk by perhaps 10% Level 2 also allows all soft errors to be corrected "on the fly" without having to reread a sector Table IV summarizes the third level RAID characteristics and Figure 3 compares the sector layout and check disks for levels 2 and 3

MTTF	Exceeds Useful Lifetime	
	G=10 (820,000 hrs or >90 years)	G=25 (346,000 hrs or 40 years)
Total Number of Disks	1 10D	1 04D
Overhead Cost	10%	4%
Useable Storage Capacity	91%	96%

Events/Sec	Full RAID	Efficiency Per Disk			Efficiency Per Disk		
(vs Single Disk)		L3	L3/L2	L3/L1	L3	L3/L2	L3/L1
Large Reads	D/S	91/S	127%	91%	96/S	112%	96%
Large Writes	D/S	91/S	127%	182%	96/S	112%	192%
Large R-M-W	D/S	91/S	127%	136%	96/S	112%	142%
Small Reads	D/SG	09/S	127%	8%	04/S	112%	3%
Small Writes	D/2SG	05/S	127%	8%	02/S	112%	3%
Small R-M-W	D/SG	09/S	127%	11%	04/S	112%	5%

Table IV *Characteristics of a Level 3 RAID The L3/L2 column gives the % performance of L3 in terms of L2 and the L3/L1 column gives it in terms of L1 (>100% means L3 is faster) The performance for the full systems is the same in RAID levels 2 and 3, but since there are fewer check disks the performance per disk improves*

Park and Balasubramanian proposed a third level RAID system without suggesting a particular application [Park86] Our calculations suggest it is a much better match to supercomputer applications than to transaction processing systems This year two disk manufacturers have announced level 3 RAIDs for such applications using synchronized 5 25 inch disks with G=4 and C=1 one from Maxtor and one from Micropolis [Maginnis 87]

This third level has brought the reliability overhead cost to its lowest level, so in the last two levels we improve performance of small accesses without changing cost or reliability

10. Fourth Level RAID Independent Reads/Writes

Spreading a transfer across all disks within the group has the following advantage
- Large or grouped transfer time is reduced because transfer bandwidth of the entire array can be exploited.

But it has the following disadvantages as well
- Reading/writing to a disk in a group requires reading/writing to all the disks in a group, levels 2 and 3 RAIDs can perform only one I/O at a time per group
- If the disks are not synchronized, you do not see average seek and rotational delays, the observed delays should move towards the worst case, hence the S factor in the equations above

This fourth level RAID improves performance of small transfers through parallelism--the ability to do more than one I/O per group at a time We no longer spread the individual transfer information across several disks, but keep each individual unit in a single disk

The virtue of bit-interleaving is the easy calculation of the Hamming code needed to detect or correct errors in level 2 But recall that in the third level RAID we rely on the disk controller to detect errors within a single disk sector Hence, if we store an individual transfer unit in a single sector, we can detect errors on an individual read without accessing any other disk Figure 3 shows the different ways the information is stored in a sector for

RAID levels 2, 3, and 4 By storing a whole transfer unit in a sector, reads can be independent and operate at the maximum rate of a disk yet still detect errors Thus the primary change between level 3 and 4 is that we interleave data between disks at the sector level rather than at the bit level
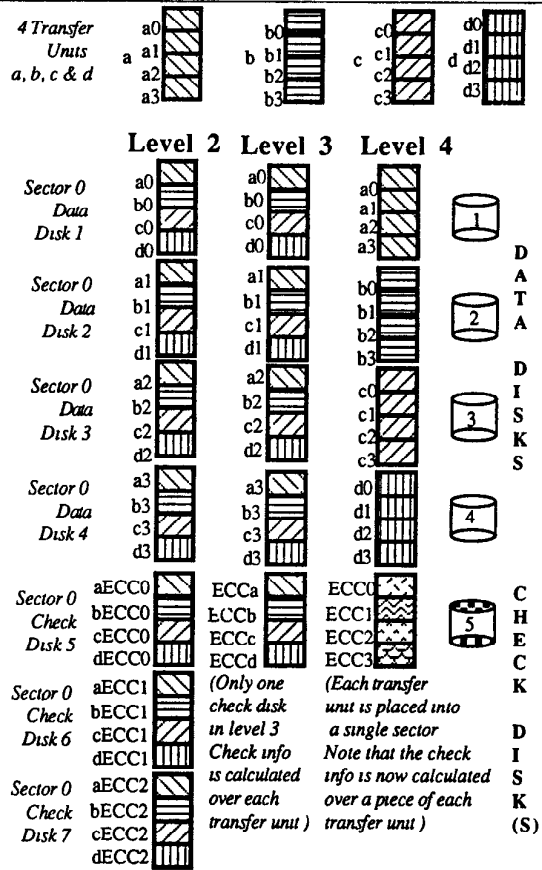

Figure 3 *Comparison of location of data and check information in sectors for RAID levels 2, 3, and 4 for G=4 Not shown is the small amount of check information per sector added by the disk controller to detect and correct soft errors within a sector Remember that we use physical sector numbers and hardware control to explain these ideas but RAID can be implemented by software using logical sectors and disks*

At first thought you might expect that an individual write to a single sector still involves all the disks in a group since (1) the check disk must be rewritten with the new parity data, and (2) the rest of the data disks must be read to be able to calculate the new parity data Recall that each parity bit is just a single exclusive OR of all the corresponding data bit in a group In level 4 RAID, unlike level 3, the parity calculation is much simpler since, if we know the old data value and the old parity value as well as the new data value, we can calculate the new parity information as follows

new parity = (old data xor new data) xor old parity

In level 4 a small write then uses 2 disks to perform 4 accesses--2 reads and 2 writes--while a small read involves only one read on one disk Table V summarizes the fourth level RAID characteristics Note that all small accesses improve--dramatically for the reads--but the small read-modify-write is still so slow relative to a level 1 RAID that its applicability to transaction processing is doubtful Recently Salem and Garcia-Molina proposed a Level 4 system [Salem 86]

Before proceeding to the next level we need to explain the performance of small writes in Table V (and hence small read-modify-writes since they entail the same operations in this RAID) The formula for the small writes divides D by 2 instead of 4 because 2

113

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.