Chapter 1

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

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 Disk (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 RAIDs to an IBM 3380 and a Fujitsu Super Eagle.

1.1 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 this 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 [Moore 86].

Recently this 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 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

Reprint from Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.109-116, June 1988

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.

1.2 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].

1.3 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 1.1 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 1.1 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 Synchronous Interface (SCSI). Such standards have encouraged companies like Adaptec to offer SCSI interfaces as single chips, in turn allowing disk companies to embed mainframe controller functions at low cost. Figure 1.1 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.

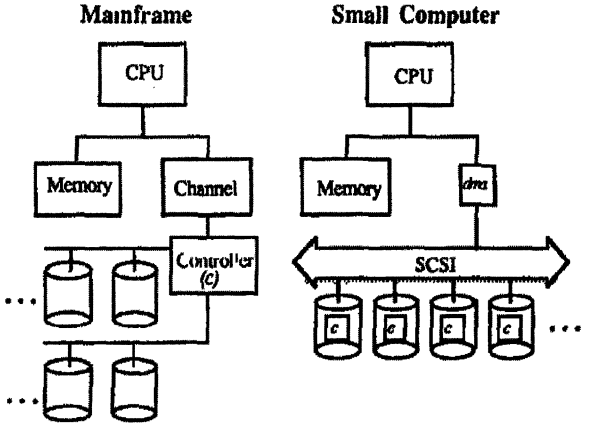


Figure 1.1 Comparison of organizations for typical mainframe and small computer disk interface. Single chip SCSI interfaces such as the Adaptec AIC-6250 allow the small computer to use a single chip to be the DMA interface as well as provide an embedded controller for each disk [Adaptec87] (The price per megabyte in Table 1.1 includes everything in the shaded boxes above).

Such characteristics lead to the proposal of building I/O systems as arrays of inexpensive disks, either interleaved for the large transfers of supercomputers [Kim 86][Livny 87][Salem 86] 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.

1.4 Caveats

We cannot explore all issues associated with such arrays in the space available for this paper, so we concentrate on the price-performance and reliability. Our reasoning is that there are no advantages in price-performance or terrible disadvantages in reliability, then there is no need to explore further. We characterize the 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 transaction-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.

1.5 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 1.1, 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.

1.6 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 the 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 the 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 the human operator replaces all failed disks. Here are some other terms that we use:

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

G = number of data disks in a group (not including the 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 the disk manufacturers make--that the 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 disk-head assembles 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 years old disks?

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

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

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

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

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

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

The average number of second failure 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 m 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. Figure 1.2 shows the ideal operation of large and small disk accesses in a RAID.

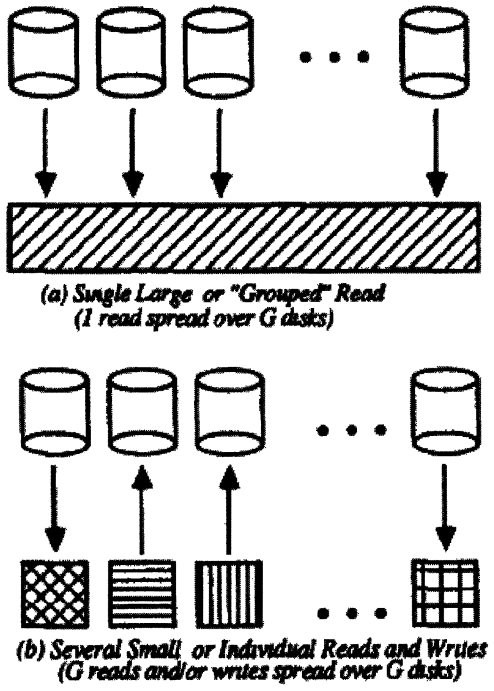


Figure 1.2 Large transfer vs. small transfer in a group of G disks

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 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.

1.7 First Level RAID: Mirrored Disks

Mirrored disks are a traditional approach for improving reliability of magnetic disks. This is the most expensive option since all disks are duplicated (G=l and C=l), 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 let reads occur in parallel. Table 1.2I 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 1.2 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 is 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 the read-modify-writes can 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 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, $1 \le S \le 2$ 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 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.

1.8 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 a 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 the 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 will 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 still 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 1.3 shows the metrics of this Level 2 RAID.

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 must include the slowdown factor S since the access must wait for all the disks to complete.

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 (*vs Single Disk*)	*Full RAID*	*Efficiency Per Disk* *L2*	*L2/L1*	*Efficiency Per Disk* *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 1.3 Characteristics of a Level 2 RAID. The L2/LI 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 I 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/Os 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 the full sectors must be read before new data can be written onto part of each sector.

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.

1.9 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 a failed either through special signals provided in the disk interface or the extra checking information at the end of a sector 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=l) 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 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 1.4 summarizes the third level RAID characteristics and Figure 1.3 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 (vs Single Disk)	*Full RAID*	*Efficiency Per Disk* *L3*	*L3/L2*	*L3/L1*	*Efficiency Per Disk* *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 1.4 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=l: 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.

1.10 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 rotational delays, the observed delays should move towards the worst case, hence the S factor in the equations above.

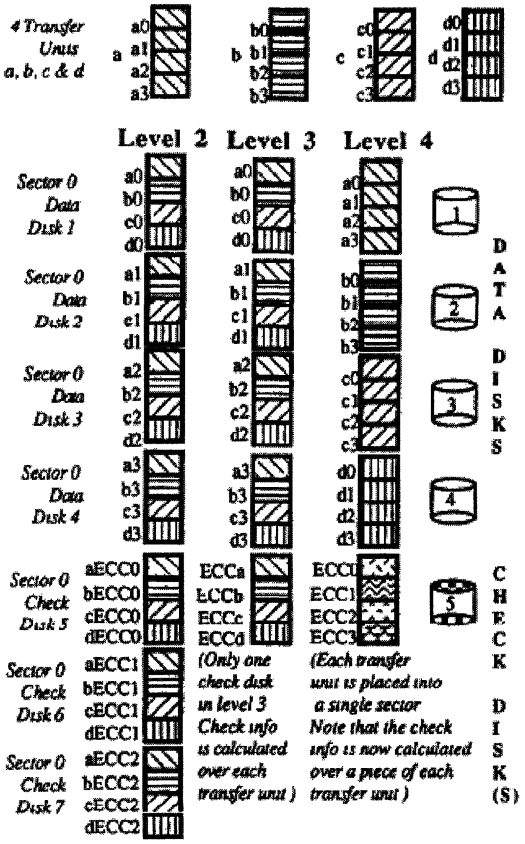


Figure 1.3 Comparison of location of data and check information in sector for RAID level 2, 3, and 4 for G=4. Not shown is the small amount of check information per sector added by the disk controller to detect and correct soft errors within a sector. Remember that we use physical

sector numbers and hardware control to explain these ideas, but RAID can be implemented by software using logical sectors and disks.

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 can 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 on a sector level rather than at the bit level.

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 bits 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 1.5 (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 accesses can proceed in parallel: the old data and old parity can be read at the same time and the new data and new parity can be written at the same time. The performance of small writes *is also divided by G* because the single check disk in a group must be read and written with every small write in that group, thereby limiting the number of writes that can be performed at a time to the number of groups.

The check disk is the bottleneck, and the final level RAID removes this bottleneck.

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 (vs Single Disk)	*Full RAID*	*Efficiency Per Disk* *L4*	*L4/L3*	*L4/L1*	*Efficiency Per Disk* *L4*	*L4/L3*	*L4/L1*
Large Reads	*D/S*	91/S	100%	91%	96/S	100%	96%
Large Writes	*D/S*	91/S	100%	182%	96/S	100%	192%
Large R-M-W	*D/S*	91/S	100%	136%	96/S	100%	146%
Small Reads	*D*	91	1200%	91%	96	3000%	96%
Small Writes	*D/2G*	05	120%	9%	02	120%	4%
Small R-M-W	*D/G*	09	120%	14%	04	120%	6%

Table 1.5 Characteristics of a Level 4 RAID. The L4/L3 column gives the % performance of L4 in terms of L3 and the L4/L1 column gives it in terms of L1 (>100% means U is faster). Small reads improve because they no longer tie up a whole group at a time. Small writes and R-M-Ws improve some because we make the same assumptions as we made in Table 1.2: the slowdown for two related I/Os can be ignored because only two disks are involved.

1.11 Fifth Level RAID: No Single Check Disk

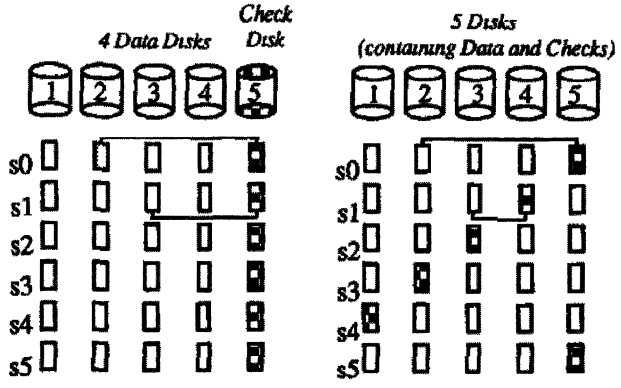
While level 4 RAID achieved parallelism for reads, writes are still limited to one per group since every write to a group must read and write the check disk. The final level RAID distributes the data and check information across all the disks--including the check disks. Figure 4 compares the location of check information in the sectors of disks for levels 4 and 5 RAIDs.

The performance impact of this small change is large since RAID level 5 can support multiple individual writes per group. For example, suppose in Figure 1.4 above we want to write sector 0 of disk 2 and sector 1 of disk 3. As shown on the left Figure 1.4, in RAID level 4 these writes must be sequential since both sector 0 and sector 1 of disk 5 must be written. However, as shown on the right, in RAID level 5 the writes can proceed in parallel since a write to sector 0 of disk 2 still involves a write to disk 5 but a write to sector 1 of disk 3 involves a write to disk 4.

These changes bring RAID level 5 near the best of both worlds: small read-modify-writes now perform close to the speed per disk of a level 1 RAID while keeping the large transfer performance per disk and high useful storage capacity percentage of the RAID levels 3 and 4. Spreading

the data across all disks even improves the performance of small reads, since there is one more disk per group that contains data. Table 1.6 summarizes the characteristics of this RAID.

Keeping in mind the caveats given earlier, a Level 5 RAID appears very attractive if you want to do just supercomputer applications, or just transaction processing when storage capacity is limited, or if you want to do both supercomputer applications and transaction processing.



(a) Check information for Level 4 RAID for G=4 and C=1 The sectors are shown below the disks (The checked areas indicate the check information) Writes to s0 of disk 2 and s1 of disk 3 imply writes to s0 and s1 of disk 5 The check disk (5) becomes the write bottleneck.

(b) Check information for Level 5 RAID for G=4 and C=1 The sectors are shown below the disks, with the check information and data spread evenly through all the disks Writes to s0 of disk 2 and s1 of disk 3 still imply 2 writes, but they can be split across 2 disks to s0 of disk 5 and to s1 of disk 4

Figure 1.4 Location of check information per sector for Level 4 RAID vs. Level 5 RAID

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 (vs Single Disk)	*Full RAID*	*Efficiency Per Disk* *L5*	*L5/L4* *L5/L1*	*Efficiency Per Disk* *L5*	*L5/L4* *L5/L1*
Large Reads	*D/S*	91/S	100% 91%	96/S	100% 96%
Large Writes	*D/S*	91/S	100% 182%	96/S	100% 192%
Large R-M-W	*D/S*	91/S	100% 136%	96/S	100% 144%
Small Reads	(1+C/G)D	1 00	110% 100%	1 00	104% 100%
Small Writes	(1+C/G)D/4	25	550% 50%	25	1300% 50%
Small R-M-W	(1+C/G)D/2	50	550% 75%	50	1300% 75%

Table 1.6 Characteristics of a Level 5 RAID. The L5/L4 column gives the % performance of L5 in terms of L4 and the L5/L1 column gives it in terms of L1 (~100% means L5 is faster). Because reads can be spread over all disks, including what were check disks in level 4, all small I/Os improve by a factor of 1 + C/G. Small writes and R-M-Ws improve because they are no longer constrained by group size, getting the full disk bandwidth for the 4 I/O's associated with these accesses. We again make the same assumptions as we made in Tables II and V: the slowdown for two related I/Os can be ignored because only two disks are involved.

1.12 Discussion

Before concluding the paper, we wish to note a few more interesting points about RAIDs. The first is that while the schemes for disk stripping and parity support were presented as if they were done by hardware, there is no necessity to do so. We just give the method, and the decision between hardware and software solutions is strictly one of cost and benefit. For example, in cases where disk buffering is effective, there is no extra disks reads for level 5 small writes since the old data and old parity would be in main memory, so software would give the best performance as well as the least cost.

In this paper we have assumed the transfer unit is a multiple of the sector. As the size of the smallest transfer unit grows larger than one sector per drive--such as a full track with an I/O protocol that supports data returned out-of-order--then the performance of RAIDs improves significantly because of the full track buffer in every disk. For example, if every disk begins transferring to its buffer as soon as it reaches the next sector, then S may reduce to less than 1 since there would be no rotational delay. With transfer units the size of a track, it is not even clear if synchronizing the disks in a group improves RAID performance.

This paper makes two separable points: the advantages of building I/O systems from personal computer disks and the advantages of five different disk array organizations, independent of disks used in those array. The later point starts with the traditional mirrored disks to achieve acceptable reliability, with each succeeding level improving:

- the data rate, characterized by a small number of requests per second for massive amounts of sequential information (supercomputer applications);
- the I/O rate, characterized by a large number of read-modify-writes to a small amount of random information (transaction-processing);
- or the useable storage capacity;

or possibly all three.

Figure 1.5 shows the performance improvements per disk for each level RAID. The highest performance per disk comes from either Level 1 or Level 5. In transaction-processing situations using no more than 50% of storage

capacity, then the choice is mirrored disks (Level 1). However, if the situation calls for using more than 50% of storage capacity, or for supercomputer applications, or for combined supercomputer applications and transaction processing, then Level 5 looks best. Both the strength and weakness of Level 1 is that it duplicates data rather than calculating check information, for the duplicated data improves read performance but lowers capacity and write performance, while check data is useful only on a failure.

Inspired by the space-time product of paging studies [Denning 78], we propose a single figure of ment called the space-speed product, the useable storage fraction times the efficiency per event. Using this metric, Level 5 has an advantage over Level 1 of 17 for reads and 33 for writes for G=10.

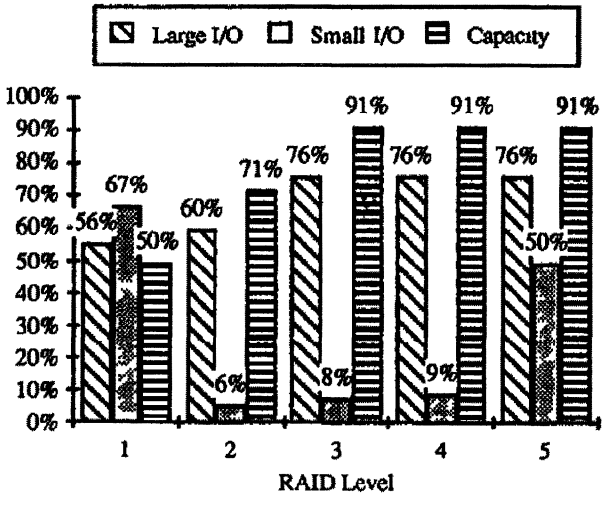


Figure 1.5 Plot of Large (Grouped) and Small (Individual) Read-Modify-Writes per second per disk and usable storage capacity for all five levels of RAID (D=100, G=10). We assume a single S factor uniform for all levels with S=13 where it is needed.

Let us return to the first point, the advantages of building I/O system from personal computer disks. Compared to traditional Single Large Expensive Disks (SLED), Redundant Arrays of Inexpensive Disks (RAID) offer significant advantages for the same cost. Table 1.7 compares a level 5 RAID using 100 inexpensive data disks with a group size of 10 to the IBM 3380. As you can see, a level 5 RAID offers a factor of roughly 10 improvement in performance, reliability, and power consumption (and hence air conditioning costs) and a factor of 3 reduction in size over this SLED. Table 1.7 also compares a level 5 RAID using 10 inexpensive data disks with a group size of 10 to a Fujitsu M2361A "Super Eagle". In this comparison RAID offers roughly a factor of 5 improvement in performance, power consumption, and size with more than two orders of magnitude improvement in (calculated) reliability.

Characteristics	*RAID 5L (100 10) (CP3100)*	*SLED (IBM 3380)*	*RAID v SLED (>1 better for RAID)*	*RAID 5L (10,10) (CP3100)*	*SLED (Fujitsu M2361)*	*RAID v SLED (>1 better for RAID)*
Formatted Data Capacity (MB)	10,000	7,500	1 33	1,000	600	1 67
Price/MB (controller incl)	$11-$8	$18-$10	2 2- 9	$11-$8	$20-$17	2 5-1 5
Rated MTTF (hours)	820,000	30,000	27 3	8,200,000	20,000	410
MTTF in practice (hours)	?	100,000	?	?	?	?
No Actuators	110	4	22 5	11	1	11
Max I/O's/Actuator	30	50	6	30	40	8
Max Grouped RMW/box	1250	100	12 5	125	20	6 2
Max Individual RMW/box	825	100	8 2	83	20	4 2
Typ I/O's/Actuator	20	30	7	20	24	8
Typ Grouped RMW/box	833	60	13 9	83	12	6 9
Typ Individual RMW/box	550	60	9 2	55	12	4 6
Volume/Box (cubic feet)	10	24	2 4	1	3 4	3 4
Power/box (W)	1100	6,600	6 0	110	640	5 8
Min Expansion Size (MB)	100-1000	7,500	7 5-75	100-1000	600	0 6-6

Table 1.7 Comparison of IBM 3380 disk model AK4 to Level 5 RAID using 100 Conners & Associates CP 3100s disks and a group size of 10 and a comparison of the Fujitsu M2361A "Super Eagle" to a level 5 RAID using 10 inexpensive data disks with a group size of 10. Numbers greater than 1 in the comparison columns favor the RAID.

RAID offers the further advantage of modular growth over SLEDs. Rather than being limited to 7,500 MB per increase for $100,000 as in the case of this model of IBM disk, RAIDs can grow at either the group size (1000 MB for $11,000) or, if partial groups are allowed, at the disk size (100 MB for $1,100). The flip side of the coin is that RAID also makes sense in systems considerably smaller than a SLED. Small incremental costs also makes hot standby spares practical to further reduce MTTR and thereby increase the MTTF of a large system. For example, a 1000 disk level 5 RAID with a group size of 10 and a few standby spares could have a calculated MTTF of 45 years.

A final comment concerns the prospect of designing a complete transaction processing system from either a Level 1 or level 5 RAID. The drastically lower power per megabyte of inexpensive disks allows systems designers to consider battery backup for the whole disk array--the power needed for 110 PC disks is less than two Fujitsu Super Eagles. Another approach would be to use a few such disks to save the contents of battery backed-up main memory in the event of an extended power failure. The smaller capacity of these disks also ties up less of the database during reconstruction, leading to higher availability. (Note that Level 5 ties up all the disks in a group in event of failure while Level 1 only needs the single mirrored disk during reconstruction, giving Level 1 the edge in availability).

1.13 Conclusion

RAIDs offer a cost effective option to meet the challenge of exponential growth in the processor and memory speeds.

We believe the size reduction of personal computer disks is a key to the success of disk arrays, just as Gordon Bell argues that the size reduction of microprocessors is a key to the success m multiprocessors [Bell 85]. In both cases the smaller size simplifies the interconnection of the many components as well as packaging and cabling. While large arrays of mainframe processors (or SLEDs) are possible, it is certainly easier to construct an array from the same number of microprocessors (or PC drives). Just as Bell coined the term "multi" to distinguish a multiprocessor made from microprocessors, we use the term "RAID" to identify a disk array made from personal computer disks.

With advantages in cost-performance, reliability, power consumption, and modular growth, we expect RAIDs to replace SLEDs in future I/O systems. There are, however, several open issues that may bare on the practicality of RAIDs:

- *What is the impact of a RAID on latency?*
- *What is the impact on MTTF calculations of non-exponential failure assumptions for individual disks?*
- *What will be the real lifetime of a RAID vs. calculated MTTF using the independent failure model?*
- *How would synchronized disks affect level 4 and 5 RAID performance?*
- *How does "slowdown" S actually behave? [Livny 87]*
- *How do defective sectors affect RAID?*
- *How do you schedule I/O to level 5 RAIDs to maximize write parallelism?*
- *Is there locality of reference of disk accesses in transaction processing?*
- *Can information be automatically redistributed over 100 to 1000 disks to reduce contention?*
- *Will disk controller design limit RAID performance?*
- *How should 100 to 1000 disks be constructed and physically connected to the processor?*
- *What is the impact of cabling on cost, performance, and reliability?*
- *Where should a RAID be connected to a CPU so as not to limit performance? Memory bus? I/O bus? Cache?*
- *Can a file system allow differ striping policies for different files?*
- *What is the role of solid state disks and WORMs in a RAID?*
- *What is the impact on RAID of "parallel access" disks (access to every surface under the read/write head in parallel)?*

Acknowledgements

We wish to acknowledge the following people who participated in the discussions from which these ideas emerged: Michael Stonebraker, John Ousterhout, Doug Johnson, Ken Lutz, Anapum Bhide, Gaetano Boriello, Mark Hill, David Wood, and students in SPATS seminar offered at U. C. Berkeley in Fall 1987. We also wish to thank the following people who gave comments useful in the preparation of this paper: Anapum Bhide, Pete Chen, Ron David, Dave Ditzel, Fred Douglis, Dieter Gawlick, Jim Gray, Mark Hill, Doug Johnson, Joan Pendleton, Martin Schulze, and Herve Touati. This work was supported by the National Science Foundation under grant #MIP-8715235.

Appendix: Reliability Calculation

Using probability theory we can calculate the $MTTF_{Group}$. We first assume independent and exponential failure rates. Our model uses a biased coin with the probability of heads being the probability that a second failure will occur within the MTTR of a first failure. Since disk failures are exponential.

$$\Pr obability\ (at\ least\ one\ of\ the\ remaining\ disks\ failing\ in\ MTTR)$$
$$= [1 - (e^{-MTTR\ /\ MTTF_{Disk}})(G + C - 1)]$$

In all practical cases,

$$MTTR << \frac{MTTF_{Disk}}{G+C}$$

and since $(1 - e^x)$ is approximately X for $0 < X << 1$:

$$\Pr obability(at\ least\ one\ of\ the\ remaining\ disks\ failing\ in\ MTTR)$$
$$= MTTR * (G + C - 1) / MTTF_{Disk}$$

Then that on a disk failure we flip this coin

heads => a system crash, because a second failure occurs before the first was repaired;

tails => recover from error and continue.

Then

$$MTTF_{Group} = Expected[Time\ between\ Failures]$$
$$* Expected[no\ of\ flips\ until\ first\ heads]$$
$$= \frac{Expected[Time\ between\ Failures]}{\Pr obability(heads)}$$
$$= \frac{MTTF_{Disk}}{(G+C) * (MTTR * (G+C-1) / MTTF_{Disk})}$$

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

Group failure is not precisely exponential in our model, but we have validated this simplifying assumption for

practical cases of MTTR << MTTF/(G+C). This makes the MTTF of the whole system just $MTTF_{Group}$ divided by the number of groups, "G.

1.14 References

[Bell 84] C. G. Bell, "The Mini and Micro Industries," *IEEE Computer*, Vol. 17, No. 10 (October 1984), pp. 14-30.

[Joy 85] B. Joy, presentation at ISSCC '85 panel session, Feb. 1985.

[Siewiorek 82] D. P. Siewiorek, C. G. Bell, and A. Newell, *Computer Structures: Principles and Examples*, p. 46.

[Moore 75] G E Moore, "Progress in Digital Integrated Electronics," *Proc. IEEE Digital Integrated Electronic Device Meeting*, (1975), p. 11

[Myers 86] G. J. Myers, A. Y. C. Yu, and D. L. House, "Microprocessor Technology Trends," *Proc. IEEE*, Vol. 74, no 12, (December 1986), pp. 1605-1622

[Garcia 84] H. Garcia Molina, R. Cullingford, P. Honeyman, R.. Lipton, "The Case for Massive Memory," Technical Report 326, Dept of EE and CS. Princeton Univ. May 1984

[Myers 86] W. Myers, "The Competitiveness of the United States Disk Industry," *IEEE Computer*, Vol. 19, No. 11 (January 1986), pp. 85-90.

[Frank 87] P. D. Frank, "Advances in Head Technology," presentation at *Challenges in Disk Technology Short Course*, Institute for Information Storage Technology, Santa Clara University, Santa Clara, California, December 15-17.1987

[Stevens 81] L. D. Stevens, "The Evolution of Magnetic Storage," *IBM Journal of Research and Development*, Vol. 25, No. 5, September 1981, pp. 663-675.

[Harker 81] J. M. Harker et al., "A Quarter Century of Disk File Innovation," *ibid,* pp. 677-689.

[Amdahl 67] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," *Proceedings AFIPS 1967 Spring Joint Computer Conference* Vol. 30 (Atlantic City, New Jersey April 1967), pp. 483-485.

[Boral 83] H. Boral and D. J. DeWitt, "Database Machines: An Ideas Whose Time Has Passed? A Critique of the Future of Database Machines," *Proc. International Conf. on Database Machines*, Edited by H.O. Leilich and M. Misskoff, Springer-Verlag, Berlin, 1983

[IBM 87] "IBM 3380 Direct Access Storage Introduction," IBM GC 26-4491-0, September 1987.

[Gawlick 87] D. Gawlick, private communication, Nov. 1987.

[Fujitsu 87] "M2361A Mini-Disk Drive Engineering Specifications," (revised) Feb. 1987, B03P-4825-0001A.

[Adaptec 87] AIC-6250, *IC Product Guide*, Adaptec, stock # DB0003-00 rev. B, 1987, p. 46

[Livny87] Livny, M., S. Khoshafian, H. Boral. "Multi-disk management algorithms." *Proc. of ACM SIGMETRICS*, May 1987

[Kim 86] M. Y. Kim. "Synchronized disk interleaving," *IEEE Trans. on Computers*, vol. C-35, no. 11, Nov 1986

[Salem 86] K. Salem and Garcia-Molina, H., "Disk Striping," *IEEE 1986 Int. Conf. on Data Engineering*, 1986

[Bitton 88] D. Bitton and J. Gray, "Disk Shadowing," *in press*, 1988

[Kurzweil88] F. Kurzweil, "Small Disk Arrays - The Emerging Approach to High Performance," presentation at Spring COMPCON 88, March 1, 1988, San Francisco, CA

[Hamming 50] R. W. Hamming, "Error Detecting and Correcting Codes," *The Bell System Technical Journal*, Vol. XXVI, No. 2 (April 1950), pp. 147-160.

[Hillis 87] D. Hillis, private communication, October, 1987.

[Park86] A. Park and K. Balasubramanian, "Providing Fault Tolerance in Parallel Secondary Storage Systems," Department of Computer Science, Princeton University, CS-TR-057-86, November 7, 1986.

[Maginnis 87] N. B. Maginnis, "Store More, Spend Less: Mid-range Options Abound9" *Computerworld*, November 16, 1987, p. 71.

[Denning 78] P. J. Denning and D. F. Slutz, "Generalized Working Sets for Segment Reference Strings," *CACM*, vol. 21, no. 9. (Sept. 1978) pp. 750-759

[Bell 85] Bell, C. G, "Multis: a new class of multiprocessor computers," *Science*, vol. 228 (April 26.1985) 462-467