REDUNDANT DISK ARRAY ARCHITECTURES AND THEIR IMPACT TO DISK SUBSYSTEM THROUGHPUT

Dragutin Vuković

Microlab d.o.o.
Savska cesta 41
41000 Zagreb
HRVATSKA

ABSTRACT: *Various redundant disk array architectures are described. Their applicability in personal computer architectures, for various purposes, is considered. Special attention is given to array architecture impact on disk subsystem's data throughput. Comparative characteristics of disk array architectures, with accent on throughput, are shown at the end of this paper.*

ARHITEKTURE ZALIHOSTNIH DISKOVNIH NIZOVA I NJIHOV UTJECAJ NA PROPUSNOST DISKOVNOG PODSUSTAVA

SAŽETAK: *Opisane su različite arhitekture diskovnih nizova sa zalihošću podataka. Razmatrana je njihova primjenjivost u arhitekturama osobnih računala za različite primjene. Posebna pažnja posvećena je utjecaju arhitekture zalihosnog niza na podatkovnu propusnost diskovnog podsustava. Na kraju je dan usporedni pregled svojstava arhitektura zalihostnih diskovnih nizova s naglaskom na propusnost podsustava.*

1. INTRODUCTION

As widespread use of microprocessor based servers increase the importance of data stored on them, system manufacturers have begun to develop innovative disk subsystem architectures to provide both reliability and data availability, and to achieve access and transfer rates beyond the physical limitations of contemporary disk drives. Eventually, storage subsystems were developed that incorporate multiple disk drives in an architecture that appears to the operating system as a single physical drive.

First papers [1], [2] mentioning the term "Redundant Array of Inexpensive Disks" (RAID) came out of the University of California, Berkeley. The Berkeley papers do not provide a strict definition of the term RAID; they rather imply the definition by giving an example of the architecture. Here we propose the definition which will incorporate the originally described architectures, as well as the manner in which RAID is used in microcomputer based systems today:

A Redundant Array of Inexpensive Disks (RAID) is any disk subsystem architecture that combines two or more standard physical disk drives into a single logical drive in order to achieve data redundancy.

In [1], RAID systems were categorized in terms of "levels". Although in [2] authors abandoned it, the term has been adopted by many industrial sources and has persisted despite the technical inaccuracy. RAID architectures are not true levels of implementation because the higher levels do

Zbornik radova KoREMA 40 (1. svezak), 1995 / KoREMA Proceedings 40 (Part 1) / Bartolić, Juraj; Glavinić, Vlado; Štih, Željko; Szabo, Aleksandar; Borić, Mladen; Mahović, Sanjin; Vukić, Zoran; Mijat, Neven; Novaković, Branko; Essert, Mario; Jakopović, Željko; Kovačić, Zdenko (ur.), ISBN 953-6037-08-4, Zagreb, Hrvatska, 19-21. travnja 1995.

not incorporate all of the features of lower levels. Therefore, we will rather use the term architectures, throughout this paper.

Here we will present the five architectures of RAID systems and discuss their applicability in microcomputer based systems in terms of overhead and seek time. Overhead is defined as the ratio between disk space used by redundant data and total disk space; seek time represents the mean time needed for a disk subsystem to find the place on disk surfaces where data should be written to or read from.

2. RAID Architectures

2.1. RAID 1

RAID 1 architecture, often called "mirrored disks" or "shadowed disks", maintains a duplicate disk with an exact copy of the information for each disk in the subsystem. Every bit is duplicated, so data redundancy is obvious with overhead of 50%.

The impact on performance is more difficult to evaluate. If both drives containing duplicated data are allowed, through optimized driver or controller, to start seeking in the same time, and data are read from the disk that completes the seek first, average access time will be better than for a single drive. Data writes always require writing to two drives, which will incur penalty relative to a single drive, waiting to two drives to complete. In a multitasking system it is possible to take a different approach. Having two exact copies of data, we can satisfy two different requests in the same time by sending one to each drive. If the system is saturated with read requests, twice as many requests can be processed and the seek time will be half that of a single drive. However, this parallel operation will be interrupted every time the write request is received. So this method will heavily depend upon the read/write ratio and the size of blocks transferred.

The primary advantage of RAID 1 is its simplicity. It can be implemented by a dual channel controller or two controllers, with minimal change in device driver and without any changes to the operating system. The most serious disadvantage of RAID 1 is cost. This includes special drivers, custom controllers and disk overhead. The second problem is physical space. RAID 1 requires twice as many disks to achieve the same amount of usable storage space, using physical space that is not abundant in microcomputer systems. They also use twice as much power, fact that is often neglected.

2.2. RAID 2

RAID 2 architecture takes advantage of the Hamming codes [3] to reduce disk overhead. The first drive contains the first bit in each data group, the second disk contains the second bit, and so forth. If each data group has eight bits there should be eight disk drives for data bits, and three disks more for error correcting code (ECC) bits. In microcomputer environment the overhead will range from 27% (11 drives) to 50% (4 drives).

For a read and write operations, all disks must seek (two times for a write), so there will be a significant slowdown relative to single drive. However, once the seek has completed, data transfer rate will be very high, since all disks will transmit data simultaneously.

ECC bits of Hamming code serve for two purposes: they are used to detect an error, and also to identify the faulty bit. In the microprocessor environment disk electronics implements internal error checking and reporting, so we will know which disk is faulty. The Hamming codes are too robust for our need and we pay penalty for storing redundant error isolation data. Thus RAID 2 will prove as unacceptable architecture for microcomputer systems and we will not consider it further.

2.3. RAID 3

RAID 3 architecture assumes that each disk in array can detect and report errors, which is true for disks used in contemporary microcomputer environments. RAID architecture need only maintain the redundant data for error correction. We will have two or more data disks and only one ECC disk. The first byte is on the first disk, the second byte is on the second disk, etc. With *n* data disks, *n+1*st byte is again on the first disk. Each logical sector of the ECC disk contains the bit-wise XOR of the corresponding sector from each data disk.

Data reads require that all of the data disks seek before reading. Write transactions require a read transaction, computing new ECC, a seek by all drives and a write to all drives including ECC drive. Data transfer rates will be high as in RAID 2, but this will generate two disadvantages. Every data drive is involved in every read or write, so RAID 3 can process only one transaction at a time. Logical sector size of RAID 3 storage equals to the sum of physical sector sizes of all data disks, getting larger every time new disk is added to the array. This results in having to read large amount of data to access small records, as well as having trouble accommodating disk buffering schemes in some operating systems. Often it means that only RAID 3 architectures with 2 or 4 data disks can be successfully integrated in microcomputer environment.

2.4. RAID 4

To decline the disadvantages of RAID 3 of having large and inconsistent transfer block sizes and inability to perform simultaneous transactions, RAID 4 eliminates interleaving transfer blocks across all disks. Rather, entire first transfer block is placed on the first data disk, second transfer block on the second drive, and so forth. There is still only one dedicated ECC drive.

Reading data involves only a single data drive and seek time is identical to a single drive architecture. Also, multiple simultaneous requests could be issued to different disks, depending on how data are segregated into distinct subsets for inclusion on different drives.

Write transaction requires reads and writes of the data drive involved and the ECC drive. ECC drive has to be read also, because it contains ECC information for other data blocks. Therefore write operations will have slightly longer seek times relative to single drive. More important is that the ECC drive is involved in every write operation, so parallelism in write operations is not possible as in read operations.

The primary advantage of RAID 4 is the ability to process multiple simultaneous reads, which make it very efficient for transaction or multitasking systems with high read/write ratio.

2.5. RAID 5

Inability to satisfy more than one write request at a time, in both RAID 3 and 4, stems from the use of dedicated ECC disk. RAID 5 tries to eliminate this problem by distributing ECC blocks, so that each disk in array contains a combination of data and ECC blocks. Transfer blocks are entirely placed on single disks as in RAID 4.

Seek times are the same as in RAID 4, but multiple simultaneous writes are possible. In saturated disk request situation, there could be half as many write transactions as the number of disks.

3. SEEK TIMES

In the above descriptions it was mentioned several times that seek times for an array are either lower or higher than for a single drive, because we either could take the best or had to wait for the worst of them.

When data has to be read, two cases exist. If there are multiple copies, we can issue seeks to all of the drives involved and read data only from the first drive to complete the seek. The second case is when data are spread across several drives, so that all of the drives involved must complete their seek before the actual transfer can take place.

Redundancy always involves writing on at least two disks, so multiple drives must seek before the data can be written. In many cases the data and existing ECC will have to be read before writing.

Let us assume that RAID subsystem contains *n* identical disks which is true for the vast majority of them. Also it could be proven that the seek time of single disk varies according to normal distribution $N(\mu,\sigma)$ with μ being the mean seek time for a single disk, and σ being standard deviation of the single drive seek time. Let us also define the $\mu_{i,n}$ to be mean seek time for first *i* out of *n* drives, and $\sigma_{i,n}$ to be the corresponding standard deviation.

To answer how much does redundant data improve seek we will assume that seek commands are issued to all *n* drives and the data are read from the first drive to complete the seek. In this case we are interested in value of $\mu_{1,n}$ and $\sigma_{1,n}$ which are shown in figure 1 for number of drives of 2 to 8 and supposed μ=10ms and σ=2ms.

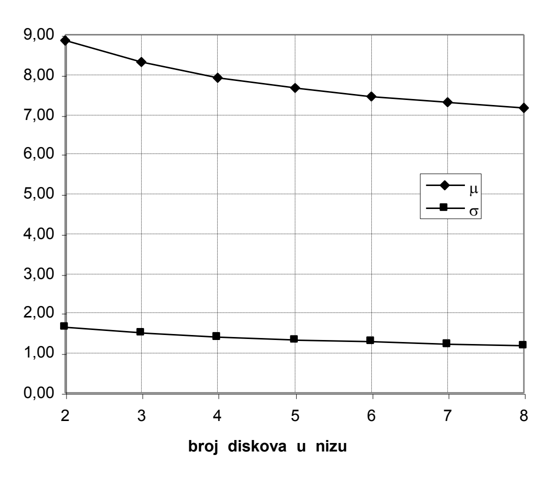

Figure 1. Values of $\mu_{1,n}$ and $\sigma_{1,n}$

How much does multiple drive seeks slow down the system? To answer this, we will assume that the seek commands are issued to all ***n*** drives, and we have to wait until all of them complete the seek. In this case we are interested in $\mu_{n,n}$ and $\sigma_{n,n}$. Their values, shown in figure 2, are computed under the dame assumptions as in previous case.

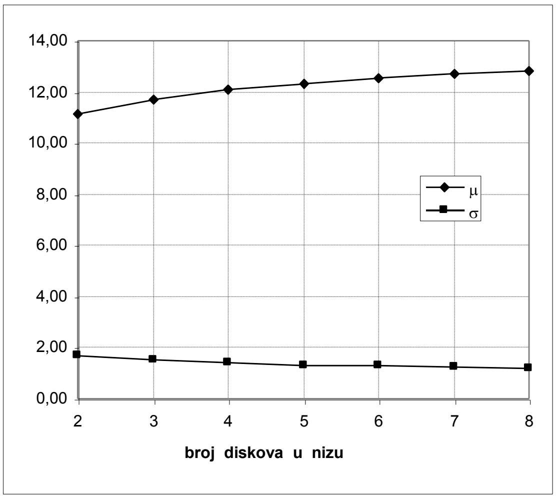

Figure 2.: Values of $\mu_{n,n}$ and $\sigma_{n,n}$

Values from the tables could be computed starting with probability density functions for seek time of the single drive, but the formulas would be too complex to integrate. Therefore, the values were determined using stochastic modeling techniques.

Table 1 summarizes characteristics of various RAID architectures. All values shown are normalized to corresponding value of the single drive. Characteristics shown include: overhead, number of simultaneous reads, number of simultaneous writes, ability to read and write simultaneously, average seek time (t_s) for single or saturated read and writes, and virtual transfer rate (Tr) for single and saturated reads and writes.

architecture characteristics	RAID architecture			
	1	3	4	5
total number of drives	n (even)	n n>2	n n>2	n n>2
overhead	1/2	1/n	1/n	1/n
# simult. R	n/2	1	n-1	n-1
# simult. W	n/2	1	1	n/2
simult. R/W?	if n>2	no	no	yes
t_s (single R)	<1	>>1	1	1
t_s (single W)	1	>>1	>1	>1
t_s (saturated R)	2/n	>>1	1/(n-1)	1/(n-1)
t_s (saturated W)	2/n	>>1	>1	2/n
Tr (single R)	1	n-1	1	1
Tr (single W)	1	n-1	1	1
Tr (saturated R)	n/2	n-1	n-1	n-1
Tr (saturated W)	n/2	n-1	1	n/2

Table 1. RAID architecture characteristics

4. CONCLUSION

Analysis and discussion presented here should not be thought off as aimed to discriminate or point out the generally best of RAID architectures. Applicability of any architecture should be considered on per case basis. It is our hope that this paper could be of use in this process.

LITERATURE

[1] D.A.Patterson, G.Gibson, R.H.Katz, A Case of Redundant Arrary of Inexpensive Disks (RAID), (undated, about 1987)

[2] D.A.Patterson, G.Gibson, R.H.Katz, Introduction to Redundant Arrarys of Inexpensive Disks (RAID), IEEE 1989

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