Optimal Striping in RAID Architecture

Hai Jin **Kai Hwang**

University of Southern California, Los Angeles, CA 90089

Email: {hjin, kaihwang}@ceng.usc.edu

Abstract: To access a RAID (redundant arrays of inexpensive disks), the disk stripe size greatly affects the performance of the disk array. In this paper, we present a performance model to analyze the effects of striping with different stripe sizes in a RAID. The model can be applied to optimize the stripe size. Compared with previous approaches, our model is simpler to apply and more accurately to reveal the real performance. Both system designers and users can apply the model to support parallel I/O events.

Keywords: RAID architecture, Parallel I/O, Optimal Design, Striping, Disk Modeling

1. INTRODUCTION

Computer technology advanced at an astonishing rate in the past decade. Processor speed and memory capacity have increased rapidly. However, disk speed has improved at a much slower rate. As a result, many applications are now limited by the speed of their disks rather than by the CPU power. As improvements in processor and memory speed continue surpassing the disk speed, more applications become I/O bound. Redundant Arrays of Independent Disks (RAID) technology [4] is an efficient way to solve the bottleneck problem between CPU and I/O subsystems. Parallel accessing data from the disk array greatly enhance not only the data throughput, but also fault tolerance.

One way to increase the data rate and I/O rate from disk array is by distributing, or striping, the data over multiple disks in the array. Distributed data access across the disk array speeds up I/Os by allowing a single I/O to transfer data in parallel from multiple disks or allowing multiple I/Os to occur in parallel. Data striping is the most common way to distribute data among multiple disks. Hence, striping is the fundamental technology in the disk array.

The main design parameter in data striping is the size of this striping unit. The size of striping unit is the basic metric that has the great influence on the performance of disk array. Smaller striping units cause logical data to be spread over more disks; larger striping units cause logical data to be grouped together on fewer disks. The size of striping unit determines how many disks each logical I/O uses. Proper choosing of the striping unit size can greatly increase the potential disk throughput and maximize the performance of disk array system.

Many studies on the selection of optimal striping unit size were reported. Patterson [4] investigates five ways to introduce redundancy into disk arrays to increase data availability. RAID level 3 has the striping unit of one byte. RAID level 4 and 5 have the striping unit of one block (block-interleaving), where a block remains unspecified. Kim [7] proposes the concept of byte-interleaving (striping unit of one byte). Livny [10] proposes a scheme called declustering where the striping unit is 1 track (26KB in their study) and compare its performance to a scheme with an infinitely large striping unit, called clustering. Reddy [11] evaluates a range of disk striping schemes ranging from byte-interleaving to block-interleaving, with a typical block size of 4KB.

Lee [8] presents a performance model of a two-dimensional disk array system. They find that for a disk cache size of less than 4MB, using a block size of one track can result in a higher throughput. For a large disk cache size, higher throughput can be obtained by using larger block sizes. With the disk array controller board equipped with a 16MB disk cache, using a block size of 2 tracks is a good choice. Lee [9] uses an analytic model of nonredundant disk arrays to derive an equation for the optimal size of data striping.

Chen [2] compares disk arrays with a striping unit of one sector (4KB) and one track (40KB) on Amdahl mainframe. Chen [5] evaluates the performance of disk arrays with various striping units and disk parameters under several workloads. They conclude that the choice of striping unit depends on only two parameters: the number of outstanding requests in the disk system at any given time and the average positioning time × data transfer rate of the disks. Chen [3] conducted study the striping unit for RAID level 5 disk array. Reads in a RAID level 5 are similar to reads/writes in RAID level 0, causing the optimal striping unit for read-intensive workload in RAID level 5 to be identical to that in a RAID level 0. For write-intensive workloads, the optimal striping unit for RAID level 5 is smaller than that for RAID level 0 by a certain factor. They also found that the optimal striping unit for read varies inversely to the number of disks, but for write varies with the number of disks.

In this paper, we use mathematical analysis method to determine the size of striping unit. By analyzing the CPU waiting time for each disk in the array, we can easily get that the optimal striping unit size of disk array is one cylinder of one member disk. Compared with the previous approaches to choose the striping unit size, the way presented in this paper is simpler and more accurate, and the fact affecting the striping unit size is easy to be gotten.

The paper is organized as follows: Section 2 gives the basic mathematics model we have developed to analyze the optimal striping unit size. Section 3 describes how to derive the optimal stripe unit size by calculating CPU waiting time of each disk. Section 4 outlines the simulation of disk access time on the target platform. We compare the simulation results with the theoretical optimal striping unit size. We compare our result with the previous approaches in section 5. From the discussion we find our method improved from the previous approaches in many aspects. Conclusions are given at the end.

2. DISK ACCESS-TIME MODEL

We first define the *striping unit* as the maximum amount of logically contiguous data that is stored in a single disk. A large striping unit will tend to keep a file clustered together on a few disks (possible one); a small striping unit tends to spread each file across many disks in the array. Unlike RAID, in this basic model, we do not include any redundant data into our data-striping scheme. Data from each file is simply distributed in round-robin fashion over disks.

Unlike Chen [5] who use the throughput as the main metric to measure the performance of disk system, in this paper, we use the *waiting time of CPU* as the metric of disk system on-line performance. The less of waiting time of CPU, the better on-line performance of disk system. We define the waiting time of CPU, T_{wait}, as the time to wait for the system to read or write amount of data from or to the disk. In stripped disk array system, the increment of on-line performance of disk array is to minimize the waiting time of CPU for each disk in the array.

The disk response time to the I/O requests of CPU is composed of three factors, they are seeking time, rotational latency and data transfer time. *Seeking time* is the time needed for the arm/head to move to the longitudinal position. It depends upon the moving distance of arm/head. *Rotational latency* is the time needed for the disk to rotate to the right angular position. It depends upon the rotation speed of spindle. *Data transfer time* is the time needed for the amount of data to transfer between transfer channel and CPU. It is proportional to the data amount needed to transfer. Thus, the total waiting time of CPU is composed of the head switching time, the head stall time, data transfer time, the seeking time and the rotational latency.

For simplicity, we assume that each disk in the array has the same physical parameters. Suppose the size of striping unit is S. The amount of data exchange between CPU and disk array is B_S. There are N disks in the array. The data transfer rate of each disk is D_r. For each disk, the size of each sector, track and cylinder is C, B_T, and B_y, respectively. The disk rotation time is t_r. The amount of data exchange with disk at one time is B_C. The way to calculate the CPU waiting time is as follow [6]:

$$T_{wait} = p[wT_h + mT_s + mT_r + fT(x) + fT(y)]$$

where p is the times to access disk while exchange amount of data between CPU and disk, it depends on the management of device and the operating system or file system. For simplicity, we choose p as a constant for discussion.

The notations used during analysis in this paper are listed in Table 1. The detail explanation for each variable and the way of calculation will be given in next section.

Table 1 Notations of Variable for Analysis

Meaning of Variable	Notation	Meaning of Variable	Notation
Number of disks in array	N	Data amount accessed	B_S
Striping unit size	S	Buffer size per disk	B_C
Minimum seek time	T_{min}	Sector size per disk	C
Average seek time	T_{avg}	Track size per disk	B_T
Maximum seek time	T_{max}	Cylinder size per disk	B_y
Head switching time	T_h	Total track number of the disk	M
Head stalling time	T_s	Total sector numbers per track	K
Seek time	$T(x)$	Disks per drive	d
Rotation latency	$T(y)$	Disk rotation time per disk	t_r
Data transfer time	T_r	Data transfer rate of disk	D_r

3. OPTIMAL STRIPING UNIT SIZE

In this section, we conduct the optimal striping unit size by calculating the CPU waiting time for each member disk in the array.

Theorem: *For a disk array with same physical parameters for each member disk drive, suppose the disks per drive is d, the sector size is C, the total sector number is K, then the optimal striping unit size for such disk array is:* ***2d* C * K***.

In the previous section, we establish the mathematical analysis model of CPU waiting time for each disk drive in the array as follow:

$$T_{wait} = p[wT_h + mT_s + mT_r + fT(x) + fT(y)] \tag{1}$$

In equation (1), T_h is the head switching time of disk when the data access is more than one track but less than one cylinder. This part of time is too small to count during calculation.

T_s is the time for the head of disk to stall during one disk move, it depends upon the disk physical structure. We can view it as a constant here.

T_r is the data transfer time at one time, that is the time needed for amount of data to exchange between CPU and disk drive at one time, it depends upon D_r and B_C. Here, it is the time when B_C such data to exchange between CPU and disk drive. The way to calculate T_r is as follow:

$$T_r = 8B_c / D_r \tag{2}$$

T(x) is the seeking time for the head of disk to seek x tracks during one disk access. It is the function of random variable x. The time for the head of disk to seek is different during each seek operation. We use the following model to calculate *T(x)* [1][9].

$$T(x) = \begin{cases} 0 & if \;\; x = 0 \\ a\sqrt{x-1} + b(x-1) + c & if \;\; M \geq x > 0 \end{cases} \tag{3}$$

here, x is the seek distance, M is the total track number of the disk drive. Coefficient a, b and c can be gotten using minimum seek time, T_{min} (the time to seek one track, here $x=1$), average seek time, T_{avg}, and maximum seek time, T_{max} (the time for head to seek from most inner track to most outer track, here $x=M$).

In order to get the way to calculate the average seek time, we first calculate the total seek times l as follow:

$$l = \sum_{L=1}^{M} 2(M-L) = M^2$$

Then, we calculate the probability of seek distance L, *P(L)*. *P(L)* is the ratio of the times of seek distance L and the total seek times l. Thus, *P(L)* is as follow:

$$P(L) = \frac{2(M-L)}{M^2}$$

Thus, we can get the way to calculate the average seek time as follow:

$$T_{avg} = \sum_{L=1}^{M} T(L) * P(L) = \sum_{L=1}^{M} \frac{2(M-L)[a\sqrt{L-1} + b(L-1) + c]}{M^2} \qquad (4)$$

Usually, it is very easy to get the three value T_{min}, T_{max}, and T_{avg} (they are the performance characteristics of disk drive, thus be announced by the disk drive manufactures). Using equation (4) and set x to 1 and M respectively in equation (3), we can easily get these three coefficients as follow:

$$a = (-10T_{\min} + 15T_{avg} - 5T_{\max}) \big/ (3\sqrt{M})$$

$$b = (7T_{\min} - 15T_{avg} + 8T_{\max}) / (3M)$$

$$c = T_{\min}$$

T(y) and rotational latency time for the head of disk to rotate y sectors during one disk access. It is the function of random variable y. It has the relationship with the rotation speed of disk drive and the angles needed to rotate in order to wait for the needed sector after indexed. The calculation of *T(y)* is as follow:

$$T(y) = (t_r / B_T / C) y \qquad (5)$$

In Equation (1), the most important thing is to determine all the coefficients. These coefficients are expressed below:

$$w = \lfloor B_s / N / B_T \rfloor \qquad (6)$$

$$m = \lfloor B_s / N / B_c \rfloor \qquad (7)$$

$$f = \begin{cases} \lfloor B_s / N / S \rfloor & (if\ s < B_y) \\ \lceil B_s / N / B_y \rceil & (if\ s \geq B_y) \end{cases} \qquad (8)$$

From the discussion above can we see that striping unit size S is only has the relationship with coefficients f, and the most time-consuming factor for data exchange between CPU and disk array is the track seeking time and sector positioning time. Thus, in order to increase the on-line performance of striped disk array, or to reduce the waiting time of CPU, the best way is to reduce the coefficients of f. From Equation (8), we can find that when $S<B_y$, f has inverse relation with striping unit size. That is to say, the larger the size of S, the smaller value of f. But when S ³B_y, f has the constant value. Thus, the optimal size of striping unit is the size of one cylinder. Suppose each disk drive

composes of *d* disks, the number of sectors within per track is *K*, the relationship between cylinder size, track size and sector size is as follow:

$B_T = C * K$

$B_y = 2\mathrm{d} * B_T$

The optimal size of striping unit *S* thus determined by:

$$S = 2\mathrm{d} * C * K \tag{9}$$

■

4. SIMULATION PLATFORM AND EXPERIMENTAL RESULTS

In this section, we use one kind of disk called *test disk* to compose disk array. The disk array is composed of 5 test disks. That is $N=5$. For simplicity, we assume that each disk has different path to the disk array controller, thus, we ignore the influence of bus delay due to the bus contention. From the simulation of the relationship of waiting time of CPU and the striping unit size, we can choose the proper striping unit size to minimize the waiting time of CPU. The parameters of test disk are shown in Table 2.

Table 2 Disk parameters in a typical RAID

Parameters of disk drive	Value
minimum seek time T_{min} (ms)	2
average seek time T_{avg} (ms)	5
maximum seek time T_{max} (ms)	10
disks per drive *d*	2
tracks per disk *M*	1000
sectors per track *K*	50
data transfer rate D_r (MB/s)	10
rotation time per circle t_r(ms)	16.6
buffer size per disk B_C (KB)	128
stable time of disk T_s(μs)	10

As each disk drive composes two disks, then $B_y=4B_T$. Buffer size of each disk drive is the amount of data exchange with disk at one time, that is B_c. According to equation (9), we can easily get the cylinder size of test disk is 100KB. We perform the experiments for the different workload to verify the theoretical optimal stripe unit size. For the different workloads, we test the CPU waiting time in different stripe unit size, shown in Figure 1. From the simulation result, we can see that the high workload can distribute data over all the disks in the array, the optimal striping unit size is the size of cylinder of disk (100KB in this simulation). For the light workload, the striping unit size is proportional to B_C, determined by the I/O request and the scale of disk array.

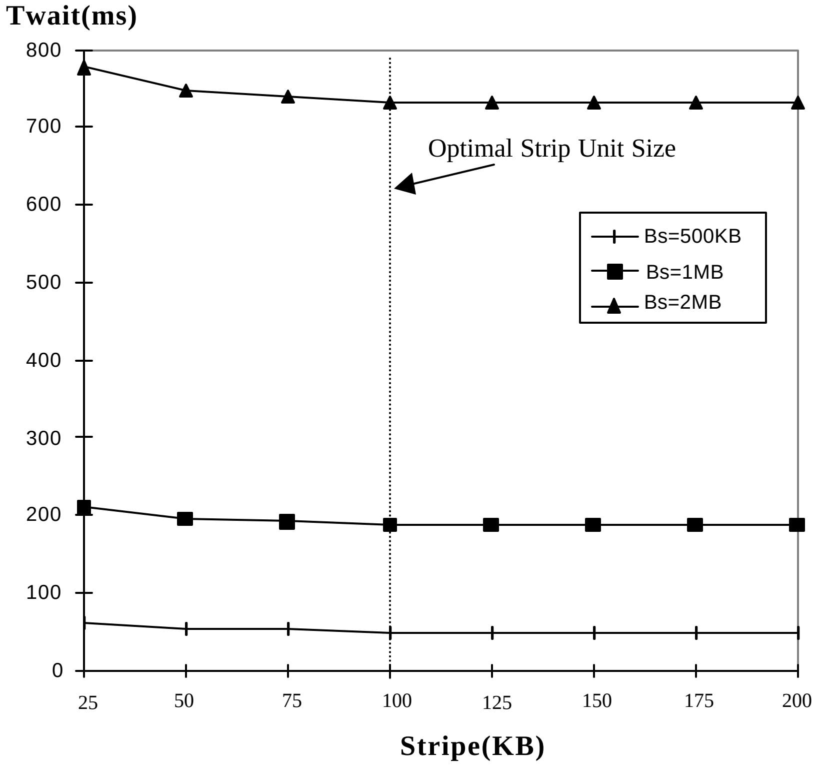

Figure 1 CPU Waiting time vs. Striping unit size over different workload

5. COMPARISONS AND CONCLUSIONS

From the discussion in section 3 and the simulation result in section 4, we can see that both theoretical analysis and simulation result can get the similar conclusion of the choosing of striping unit size. Generally, it is better to choose the striping unit size as the size of one cylinder without knowing the system burden or workload. The other conclusion is that using drive with more disks in single drive may increase the performance of disk array. This is because the more disks per disk drive, the larger the size of one cylinder in disk drive, thus the larger the striping unit size. But on the other hand, the large striping unit size tends to keep a file clustered together on a few disks, hence reduce the parallelism (or concurrency) of I/O request. Thus, how to determine the number of disks in one disk drive is still an open problem.

Table 3 compares the methods presented in this paper with the previous research result in selecting the optimal stripe unit size for disk array. From the comparison we can see that the method presented in this paper is simpler and more precise. All the parameters in calculating the optimal stripe unit size are easy to determine.

Table 3 Comparison of three optimal stripe size selection methods

Source of methods	**Optimal stripe size**	**Comment**
Our method	2d* C * K	d is disks per drive, C is sector size, K is total sector number
Chen [5]	*Z × average positioning time × data transfer rate*	Z is constant, it is sensitive to the number of disks in the array
Lee [9]	$\sqrt{\frac{PX(L-1)Z}{N}}$	P is the average disk positioning time, X the average disk transfer rate, L the concurrency, Z the request size, N the array size in disks

REFERENCES

[1] V. Catania, A. Puliafito, S. Riccobene and L. Vita, "Design and Performance Analysis of a Disk Array System", *IEEE Transactions on Computers*, Vol.44, No.10, Oct. 1995, pp.1236-1247

[2] P. M. Chen, "An Evaluation of Redundant Arrays of Disks Using an Amdahl 5890", *Proceedings of the 1990 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, May 1990

[3] P. M. Chen and E. K. Lee, "Striping in a RAID Level 5 Disk Array", *Proceedings of 1995 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pp. 136-145, May 1995.

[4] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz and D. A. Patterson. "RAID: high-performance, reliable secondary storage". *ACM Computing Surveys*, Vol. 26, No.2, pp.145-185, June 1994

[5] P. M. Chen and D. A. Patterson, “Maximize Performance in a Striped Disk Array”, *Computer Architecture News*, Vol.18, No.2, June 1990, pp.322-331

[6] H. Jin, H. Yang and J. Zhang, “On-line Performance Evaluation of RAID 5 using CPU Utilization”, *Proceedings of Signal and Data Processing of Small Targets 1998*, SPIE, Vol. 3373, 14-16, April, 1998, Orlando, Florida, USA, pp. 498-509

[7] M. Y. Kim, “Asynchronous Disk Interleaving: Approximating Access Delays”, *IEEE Transactions on Computers*, vol.C-40, No.7, July 1991, pp.801-810

[8] C.- S. Lee and T. – M. Parng, “Performance Modeling and Evaluation of a Two Dimensional Disk Array System”, *Journal of Parallel and Distributed Computing*, Vol.38, 1996, pp.16-27

[9] E. K. Lee and R. H. Katz, “An Analytic Performance Model of Disk Arrays”, *Proceedings of 1993 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, May 1993, pp.98-109

[10] M. Livny, S. Khoshafian and H. Boral, “Multi-Disk Management Algorithms”, *Proceedings of the 1987 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pp.69-77

[11] A. L. N. Reddy and P. Banerjee, “Performance Evaluation of Multiple-Disk I/O Systems”, *IEEE Transactions on Computers*, December 1989