ELSEVIER

Journal of Systems Architecture 42 (1996) 37–53

JOURNAL OF
SYSTEMS
ARCHITECTURE

Performance analysis of hybrid disk array architectures to meet I/O requirements [1]

Chan-Ik Park *, Kyung Sook Hwang

Dept. of Computer Science, POSTECH, San 31, HyoJa dong, Pohang, 790-784 South Korea

Received 4 November 1994; revised 6 November 1995; accepted 29 January 1996

Abstract

Disk array is an I/O subsystem incorporating multiple disks to improve the I/O performance. This paper considers general architectural models of hybrid disk array systems which integrate asynchronous and synchronous architectures together. An I/O workload is generally characterized by average request size, average request rate, and mean response time requirement. In this paper, we propose an analytic performance model of hybrid disk array systems for a given I/O workload. With the help of the analytic model, we can determine the three important design parameters of a hybrid disk array system in order to meet the mean I/O response time requirement. These parameters include the minimum number of required disks, the degree of declustering, and the degree of synchronization. Our analytic model is verified by the simulation.

Keywords: Asynchronous disk; Declustering; Disk array; Disk striping; Hybrid architecture; Performance analysis; Response time; Synchronous disk

1. Introduction

With the rapid progress of computer architecture and semiconductor technology, the processing power of a computer system has been improved greatly in recent years. This trend allows the size of application programs to grow, i.e., the amount of data to be processed is becoming huge. In this situation, it is very significant to balance disk I/O speed with CPU processing performance in order to achieve a good system performance.

* Corresponding author. Email: cipark@vision.postech.ac.kr

[1] This research has been supported in part by KOSEF, in part by ADD, and in part by ETRI, 1995.

PII S1383-7621/0165-60764(96)00014-2

However, disk systems have intrinsic limitations on performance improvement due to their mechanical components. Furthermore, much higher computing power can be obtained in a massively parallel system by incorporating multiple processing elements. This has caused a large performance gap between CPU and disk. With such large gap between processing and I/O, system performance will be eventually limited by the I/O bandwidth.

A disk system can be constructed by incorporating multiple disk drives, and accessed by interleaving in order to achieve a level of performance according to the number of disk drives. Disk interleaving (also called disk striping) has been suggested as a means of improving I/O bandwidths [7,11,13]. In an interleaved disk system, a blockwise interleaving of a file is adopted: a data block B may be partitioned into n subblocks $b_1, b_2, \ldots, b_n$. A subblock b_i is assigned to a disk unit $((i-1) \bmod n)+1$, where n is the degree of interleaving. That is, if subblock b_i is assigned to disk j, then b_{i+1} is to disk $j+1$, and so on. A subblock is called a striping unit (SU) and the degree of interleaving is called the striping width (SW). According to disk behaviours and applications, the size of a SU can vary from one bit to multiple tracks, and the SW can vary anywhere from 1 to N, where N is the total number of disks.

Generally, disks are treated independently of each other, i.e., disks are asynchronous. When a disk I/O request arrives at the disk system, then that request is broken up into a number of several subrequests and each subrequest is queued at the corresponding disk. As a result, the seek and rotational delays involved in the transfer of a request are different for each disk. When this type of disks are incorporated in a disk system, the size of an SU is usually equal to a sector or multiple of sectors so that each disk can issue I/O independently. A group of disks is said to be fully synchronized if each disk head is positioned at the same sector as well as the same track. In this case, the size of an SU becomes a bit or a byte so that a group of disks are simultaneously put to serve a request as if it is a single disk.

In a traditional disk system, all the disks are organized as independent units and each file is located only on one disk. The design choice in implementing a traditional disk system includes how fast a single disk can handle a request and how large data a single disk can store. In a multiple disk system adopting disk striping, however, there are several design parameters to be considered to provide an appropriate disk I/O bandwidth for a certain workload: (1) the type of disks to be used, (2) total number of disks to be used, (3) the striping width, and (4) the size of a striping unit. Among these, the issue related to the size of a striping unit has been extensively considered in [1]. We, therefore, do not consider the size of a striping unit in our further analysis.

The response time of an I/O request is usually assumed to consist of four components: queueing delay, seek delay, rotational delay, and data transfer. Weikum et al. [15,16] have proposed an analytic model to determine the striping width for a multiple disk system consisting only of a certain number of asynchronous disks. Their analytic model, however, completely ignores queueing delay. Chen and Towsley [2] have proposed an analytic model to estimate the performance of a multiple disk system. However, their model has considered queueing delay without dealing with synchronization effects among asynchronous disks. Another analytic model to estimate the disk performance has been proposed in [8], where queueing delay is ignored. Moreover, only one disk organization is considered consisting of asynchronous disks. Therefore, these previous models are not general enough to be used to determine the design parameters. Simulation work has been carried out by Reddy and Barnerjee [12] on various configurations obtained from different disk organizations, different number of disks, and different striping width. In this paper, we present an analytic model considering queueing delay as well as synchronization effects among disks in order to help determine these design parameters in multiple disk systems.

The paper is organized as follows. In Section 2, we explain our analysis model for various disk organizations. The analysis model is verified in Section 3 by simulating a multiple disk system. Conclusions are given in Section 4.

2. Our analysis model

In this section, we evaluate the expected response time of an I/O request to determine design parameters for a multiple disk system when an I/O workload is specified by the three components: mean request size R, mean arrival rate of requests λ, and mean response time requirement r_{req}.

2.1. Notations and basic assumptions

I/O requests traffic is assumed to be a Poisson process with mean arrival rate λ. A request is partitioned to a number of subrequests and it is queued at one of the disks selected with equal probability. It is also assumed that one subrequest can be handled by a single seek and a single rotational delay, and the start address of a subrequest is randomly determined. Since, as explained before, the issues on the size of a striping unit are not discussed in this paper, the striping unit is assumed to be one disk block, i.e., one sector.

In addition, the following notations are used in the analysis.

- *BS*: the size of one disk block
- *BT*: the number of blocks per track
- *CD*: the number of cylinders per disk
- *FRT*: the time taken to rotate a single track

2.2. Analysis of request response time: A single disk

The response time of an I/O request consists of two components: queueing delay and disk service time. The disk service time is composed of seek delay, rotational delay, and data transfer time. In this subsection, all components constituting a request response time are analyzed. The random variables S_{d}, R_{d}, T_{d}, D_{d}, and D_{r} are used to denote seek delay, rotational delay, data transfer time, disk service time, and request response time, respectively.

2.2.1. Seek delay

We assume that the start address of each request is randomly determined. Therefore, the probability density function $f_{\mathrm{p}}(x)$ of positioning the head at an arbitrary place becomes $f_{\mathrm{p}}(x) = 1/c$ for $0 \leqslant x \leqslant c$, where $c = CD - 1$. Seek distance can be computed from current and next head positions, each of which has the same probability density function $f_{\mathrm{p}}(x)$. Therefore, we get the probability distribution function $F_{\mathrm{d}}(x)$ for seek distance x:

$$F_{\mathrm{d}}(x) = \mathrm{P}(X \leqslant x) = \frac{2x}{c} - \frac{x^2}{c^2}, \quad 0 \leqslant x \leqslant c.$$

Since the seek time t is modelled by a non-linear function on seek distance x, $a+b\sqrt{x}$ [8], the probability distribution function of seek time $F_{S_d}(t)$ and the probability density function (pdf) $f_{S_d}(t)$ are obtained as follows.

$$F_{S_d}(t)=\frac{2}{c}\left(\frac{t-a}{b}\right)^2-\frac{1}{c^2}\left(\frac{t-a}{b}\right)^4. \tag{1}$$

$$f_{S_d}(t)=\frac{4(t-a)}{cb^2}\left(1-\frac{(t-a)^2}{cb^2}\right),\quad a\leqslant t\leqslant a+b\sqrt{c}. \tag{2}$$

2.2.2. Rotational delay

Since the start addresses of each disk requests are randomly selected, the rotational delay has the uniform distribution on [0, FRT). That is, its pdf $f_{R_d}(x)$ is derived as follows:

$$f_{R_d}(x)=\frac{1}{FRT},\quad 0\leqslant x<FRT. \tag{3}$$

2.2.3. Data transfer time

The time required to serve a request of R blocks is derived as constant:

$$DataTransferTime=R\times\frac{FRT}{BT}.$$

2.2.4. Disk service time

The random variable D_d denoting disk service time is represented by the sum of three random variables S_d, R_d, and T_d. Note that T_d has constant for a given request size of R blocks. Then, by the convolution of $f_{S_d}(x)$ and $f_{R_d}(x)$, the pdf of disk service time D_d, $f_{D_d}(t)$, is specified as Eqs. (4)–(6). (For details, refer to [4].) Here, m represents the data transfer time for a given request size of R blocks, i.e., $m=R\times FRT/BT$, and l represents maximum rotational delay, i.e., $l=FRT$.

(1) $a+m\leqslant t<l+a+m$.

$$\begin{aligned}f_{D_d}(t)&=\int_a^{t-m}f_{S_d}(x)f_{R_d}(t-m-x)\,\mathrm{d}x\\&=\frac{1}{l}\left(\frac{2(t-m-a)^2}{cb^2}-\frac{(t-m-a)^4}{b^4c^2}\right).\end{aligned} \tag{4}$$

(2) $l+a+m\leqslant t<a+b\sqrt{c}+m$.

$$\begin{aligned}f_{D_d}(t)&=\int_{t-m-l}^{t-m}f_{S_d}(x)f_{R_d}(t-m-x)\,\mathrm{d}x\\&=\frac{4(t-m-a)-2l}{cb^2}+\frac{-4(t-m-a)^3+6l(t-m-a)^2-4l^2(t-m-a)+l^3}{b^4c^2}.\end{aligned} \tag{5}$$

(3) $a+b\sqrt{c}+m \leqslant t < a+b\sqrt{c}+l+m$.

$$f_{D_{\mathrm{d}}}(t)=\int_{t-m-l}^{a+b\sqrt{c}} f_X(x) f_Y(t-m-x)\,\mathrm{d}x$$

$$=\frac{1}{l}\left(1+\frac{(t-m-l-a)^4}{b^4c^2}-\frac{2(t-m-a-l)^2}{cb^2}\right). \tag{6}$$

We denote mean disk service time for a request of R blocks as s_{d}, and its deviation as σ_{d}. Since S_{d}, R_{d}, and T_{d} are independent of each other, we get

$$s_{\mathrm{d}}=\mathrm{E}[D_{\mathrm{d}}]=\mathrm{E}[S_{\mathrm{d}}+R_{\mathrm{d}}+T_{\mathrm{d}}]=\mathrm{E}[S_{\mathrm{d}}]+\mathrm{E}[R_{\mathrm{d}}]+\mathrm{E}[T_{\mathrm{d}}], \tag{7}$$

where $\mathrm{E}[S_{\mathrm{d}}]$, $\mathrm{E}[R_{\mathrm{d}}]$, and $\mathrm{E}[T_{\mathrm{d}}]$ can be easily derived as follows:

$$\mathrm{E}[S_{\mathrm{d}}]=\int_a^{a+b\sqrt{c}} x\times\frac{4(x-a)}{cb^2}\left(1-\frac{(x-a)^2}{cb^2}\right)\mathrm{d}x,$$

$$\mathrm{E}[R_{\mathrm{d}}]=\frac{FRT}{2},$$

$$\mathrm{E}[T_{\mathrm{d}}]=R\times\frac{FRT}{BT},$$

Note that only $\mathrm{E}[T_{\mathrm{d}}]$ is dependent of the request size R.

And, its standard deviation, σ_{d}, is obtained as follows.

$$\sigma_{\mathrm{d}}=\sqrt{\mathrm{Var}[S_{\mathrm{d}}+R_{\mathrm{d}}+T_{\mathrm{d}}]}=\sqrt{\mathrm{Var}[S_{\mathrm{d}}]+\mathrm{Var}[R_{\mathrm{d}}]+\mathrm{Var}[T_{\mathrm{d}}]}=\sqrt{\mathrm{Var}[S_{\mathrm{d}}]+\mathrm{Var}[R_{\mathrm{d}}]}, \tag{8}$$

where

$$\mathrm{Var}[S_{\mathrm{d}}]=\int_a^{a+b\sqrt{c}}(x-\mathrm{E}[S_{\mathrm{d}}])^2\times\frac{4(x-a)}{cb^2}\left(1-\frac{(x-a)^2}{cb^2}\right)\mathrm{d}x,$$

$$\mathrm{Var}[R_{\mathrm{d}}]=\int_0^{FRT}(x-\mathrm{E}[R_{\mathrm{d}}])^2\frac{1}{FRT}\,\mathrm{d}x,$$

$$\mathrm{Var}[T_{\mathrm{d}}]=0.$$

2.2.5. Request response time

An M/G/1 queueing model [5,9] can be constructed for a disk with random requests specified by mean request size R_{d} and mean arrival rate λ_{d}. Let s_{r} and σ_{r} denote mean request response time and its deviation.

Then, mean request response time s_r can be derived as follows [5],

$$s_{\mathrm{r}}=s_{\mathrm{d}}+\text{queueing delay}=s_{\mathrm{d}}+\rho\frac{s_{\mathrm{d}}(1+\alpha^2)}{2(1-\rho)} \tag{9}$$

and its variance σ_r^2 as

$$\sigma_r^2 = \sigma_d^2 + \frac{\lambda_d s_d^{**}}{3(1-\rho)} + \frac{(\lambda_d s_d^*)^2}{4(1-\rho)^2}, \tag{10}$$

where

$$\rho = \lambda_d \times s_d,$$

$$\alpha = \frac{\sigma_d}{s_d},$$

$s_d = \mathrm{E}[D_d]$ with the request size R_d,

$s_d^* = \mathrm{E}[D_d^2]$ with the request size R_d,

$s_d^{**} = \mathrm{E}[D_d^3]$ with the request size R_d.

Note that both of s_r and σ_r depends on R_d and λ_d.

2.3. Analysis of request response time: Hybrid organizations

Fig. 1 shows a hybrid organization with two levels. In the lower level, a set of synchronous disks constitutes a synchronous organization called a group. The number of synchronous disks in a group is called the degree of synchronization and denoted as P_s. Each group has the same number of disks in a hybrid organization. Note that each group can be regarded as a single disk from the performance point of view since all disks in a group work synchronously. Assuming each group as a single disk, we can organize the higher level by combining some

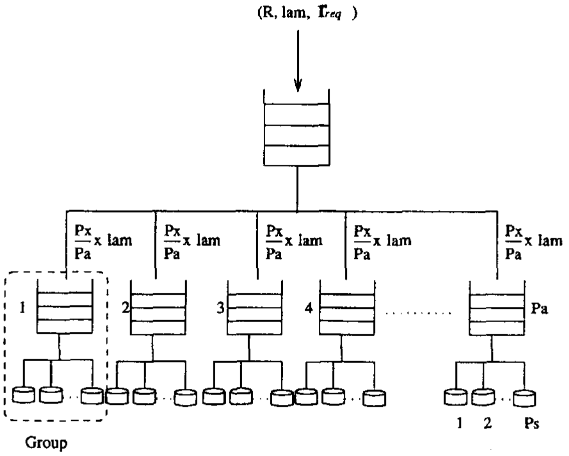

Fig. 1. A hybrid organization: I/O requests are characterized by $(R,\ lam(\lambda),\ r_{req})$. P_a represents the number of groups, P_s represents the number of synchronous disks in a group, and P_x represents the striping width. The request arrival rate in a local queue is given by $(P_s / P_a) \times lam(\lambda)$.

number of groups such that all groups are independently working. The number of groups is denoted as P_a. That is, the higher level constitutes an asynchronous organization. It is assumed to adopt bytewise interleaving in a synchronous organization and blockwise interleaving in an asynchronous organization. Each request arrived at the disk system is broken up into a number of subrequests and it is queued at the local queues of the corresponding groups. The number of subrequests for a given request is called the degree of declustering or the striping width. We denote it as P_x. Therefore, we have P_x groups working concurrently.

Synchronous organizations can easily be modeled from hybrid organizations shown in Fig. 1 by setting $P_a = 1$ and $N = P_s$ where N represents total number of disks. Likewise, asynchronous organizations can also be modeled from hybrid organizations by setting $P_s = 1$ and $N = P_a$. We now analyze hybrid organizations in order to determine design parameters such as P_s, P_a, and P_x. The total number of disks N is then equal to $P_s \times P_a$.

To begin with, we develop an analysis model for a synchronous organization. And then, hybrid organizations are analyzed. Assume that an I/O workload is characterized by mean request size R, mean arrival rate λ, and mean response time requirement r_{req}.

2.3.1. Mean response time of a synchronous organization

Let R_g and λ_g denote subrequest size coming to a group and subrequest arrival rate seen at the group. In Fig. 1, each request arrived at a disk system is broken up into P_x subrequests (the degree of declustering) and it is queued at the local queues of the corresponding groups (or disks). Since the destination local queue of each subrequest is randomly selected, i.e., each local queue has an equal probability of being selected by a subrequest, then the subrequest arrival rate at a local queue becomes $(P_x/P_a) \times \lambda$, and the size of a subrequest is $\lceil R/P_x \rceil$ where R represents the size of a request. Therefore, $\lambda_g = (P_x/P_a) \times \lambda$ and $R_g = \lceil R/P_x \rceil$.

Since a set of P_s synchronous disks in a group can be regarded as a single virtual disk, mean response time s_g and its deviation σ_g of a group are simply derived from s_r and σ_r in Eqs. (9) and (10) by setting $R_d = R_g/P_s$ and $\lambda_d = \lambda_g$.

2.3.2. Mean response time of a hybrid organization

As mentioned before, each request arrived at a disk system is broken up into P_x subrequests and it is queued at the local queues of the corresponding groups (or disks). Since P_x denotes the number of subrequests coming from a request, there are P_x groups involved in the request handling.

If we denote D_{g_i} as a random variable representing subrequest response time by group g_i, then the response time of a request of R blocks, D_{array}, is derived as the maximum of all D_{g_i} for $1 \leqslant i \leqslant P_x$.

$$D_{array} = \max\left(D_{g_1}, D_{g_2}, \ldots, D_{g_{P_x}}\right) \tag{11}$$

In order to make further derivation tracktable, we assume that each D_{g_i} for $1 \leqslant i \leqslant P_x$ has identical and independent distribution (i.i.d.). Thus, the problem is to derive the expected value of the maximum of a set of i.i.d. random variables. In the case that D_{g_i} has simple distributions such as the exponential and the uniform distributions, one can easily obtain closed form expressions for the expected maximum [8,10]. In [8], D_{g_i} is assumed to have normal distribution so that an approximate form had been derived. Unfortunately, D_{g_i} does not have a simple distribution such as exponential, uniform, or normal. Therefore, it is very hard to obtain a closed form expression for the expected maximum $\mathrm{E}[D_{\mathrm{array}}]$. In [10], it is shown that an approximated form can be derived even when the distribution of D_{g_i} is not clearly defined.

The expected maximum $\mathrm{E}[D_{\mathrm{array}}]$ is approximated by the characteristic maximum of the random variable D_{g_i}. The characteristic maximum of D_{g_i} is defined by a quantity x_{cm} satisfying the following equation,

$$x_{\mathrm{cm}} = \inf\left\{ x \,|\, 1 - F_{D_{g_i}}(x) \leqslant \frac{1}{P_{\mathrm{x}}} \right\}, \tag{12}$$

where $F_{D_{g_i}}(x)$ is the distribution function of D_{g_i} [3,8,10]. For continuous functions, the characteristic maximum is obtained from the equation

$$F_{D_{g_i}}(x_{\mathrm{cm}}) = 1 - \frac{1}{P_{\mathrm{x}}}. \tag{13}$$

However, since the distribution function $F_{D_{g_i}}$ is not clearly defined in our analysis model, we have to approximate $F_{D_{g_i}}$ by the method given in [10].

We will show how their approximation method applies to our analysis. To begin with, x_{cm} satisfying Eq. (13) is assumed to have the following form since mean response time of a group s_{g} can be used as an estimate of $\mathrm{E}[D_{g_i}]$:

$$x_{\mathrm{cm}} = s_{\mathrm{g}} + s_{\Delta}. \tag{14}$$

The problem is to derive s_{Δ} since we already have s_{g}. According to the approximation method of [10], $F_{D_{g_i}}$ can be approximated by

$$F_{D_{g_i}}(x) \simeq \frac{s_{\Delta}^2}{2\sigma_{\mathrm{g}}^2}. \tag{15}$$

Combining Eqs. (13) and (15), we get $s_{\Delta} = \sigma_{\mathrm{g}}\sqrt{2(1 - 1/P_x)}$. Finally, we can derive $\mathrm{E}[D_{\mathrm{array}}]$ which is approximated by x_{cm} in Eq. (14):

$$\mathrm{E}\left[D_{\mathrm{array}}\right] \simeq s_{\mathrm{g}} + s_{\Delta} = s_{\mathrm{g}} + \sigma_{\mathrm{g}}\sqrt{2\left(1 - \frac{1}{P_{\mathrm{x}}}\right)}. \tag{16}$$

Since the approximation in Eq. (16) has large errors when P_{x} is small [8,10], we have devised the following approximation in the case of $1 \leqslant P_{\mathrm{x}} \leqslant 10$:

$$\mathrm{E}\left[D_{\mathrm{array}}\right] \simeq s_{\mathrm{g}} + s_{\Delta} = s_{\mathrm{g}} + \sigma_{\mathrm{g}}\sqrt{1.8\left(\frac{P_{\mathrm{x}} - 1}{10 - 1}\right)}. \tag{17}$$

Note that two approximations (Eqs. (16) and (17)) have the same value at both end points of $P_{\mathrm{x}} = 1$ and $P_{\mathrm{x}} = 10$.

Using $\mathrm{E}[D_{\text{array}}]$ derived above, the design parameters such as P_s, P_a, and P_x have to be determined such that the resulting disk system satisfies the following condition:

$$\mathrm{E}[D_{\text{array}}]=\begin{cases} s_g+\sigma_g\sqrt{1.8\left(\dfrac{P_x-1}{10-1}\right)}\leqslant r_{\text{req}}, & \text{when } 1\leqslant P_x\leqslant 10,\\ s_g+\sigma_g\sqrt{2\left(1-\dfrac{1}{P_x}\right)}\leqslant r_{\text{req}}, & \text{when } 11\leqslant P_x. \end{cases} \tag{18}$$

Remember that r_{req} denotes mean response time requirement of a request characterizing an I/O workload.

Now the problem is to determine three parameters (P_s, P_a, P_x) from a single equation specified as Eq. (18). Thus, a solution may not exist or more than one solution may exist. In the case of multiple solutions, first we choose a solution with minimum number of disks $(P_s \times P_a)$. And then we select a solution with minimum response time.

3. Verification by simulation

We now verify our analysis model by simulating multiple disk systems. Our goal is to determine design parameters for a given I/O workload. This section compares design parameters obtained from the analysis model with those from simulation.

3.1. Simulation environment

3.1.1. Disk parameters

Each disk is assumed to have the characteristics shown in Table 1 similar to Seagate ST1480 disk [14]. And the seek time is modeled by a non-linear function of seek distance x: $1.7+0.8\sqrt{x}$ ms for $0\leqslant x\leqslant 724$.

3.1.2. Workload generation

The request size has a normal distribution with mean R and its traffic is considered Poissonian with mean λ. Total number of generated requests is 5,100 and the first 100 requests are used to stabilize simulation experiments. Final results are obtained by the average of experiments executed ten times. The confidence interval of experimental results is 90% and its width is less then 10% of the mean. The workload parameters used in simulation are shown in Table 2.

Table 1
Disk parameters

Block size (BS)	512 Byte
Blocks per track (BT)	60
Full rotation time (FRT)	13.6 ms
Cylinders per disk (CD)	725

Table 2
Workload parameters

R blocks	200, 2000
λ (req/sec)	1, 15, 30, 60
r_{req} (msec)	50, 100, 250, 500, 1000, 2000, 3000

3.1.3. Handling requests

A request arrived at the global queue is partitioned into a number of subrequests and it is queued at the local queues of the corresponding groups. Note that the number of subrequests for a request is P_x since P_x represents the striping width. We denote the selected groups as $\{g_1, \ldots, g_{P_x}\}$, which are determined as follows.

$$g_1 = \text{Random}[1, \ldots, P_a], \qquad g_i = (g_1 + i - 1) \bmod P_a \quad \text{for } 2 \leqslant i \leqslant P_x.$$

And we denote the size of the subrequest to g_i as R_{g_i}. Then, if the request size is R blocks, R_{g_i} is determined as follows.

$$R_{g_1} = base + 1, \ \ldots, \ R_{g_{add}} = base + 1, \ R_{g_{add+1}} = base, \ \ldots, \ R_{g_{P_x}} = base$$

where $base = \lfloor R/P_x \rfloor$ and $add = R \bmod P_x$. Note that each disk in a group has to transfer $\lceil R_{g_i}/P_s \rceil$.

Each queue is serviced in FIFO. The seek position of each request is determined randomly between $[1, 2, \ldots, CD]$, and the sector position is also determined randomly between $[1, 2, \ldots, BT]$, where CD and BT represent the number of cylinders per disk and the number of blocks per track, respectively.

3.2. Experimental results

We can see from the experiments that the number of disks increases as the request size R and the arrival rate λ increase. It is expected from our analysis model as well.

The approximations used in Eqs. (14) and (15) may have much errors if the characteristic maximum value is far off from the base value s_g. However, since few errors can be found in comparison between [8] and our simulation, our analysis model can be good enough to be used to determine design parameters. In the worst case, we have 13.69% errors in the number of disks in the case of asynchronous organizations with the input characteristics of $r_{\text{req}} = 50$ ms, $\lambda = 60$ req/sec, and $R = 2000$ blocks. The number of disks obtained from our analysis model is smaller than that from simulation. However, the actual response time obtained from simulation is within 4.66% of r_{req} even when the number of disks is set to the value obtained from the analysis.

Table 3
Design parameters determined from analysis and simulation in a hybrid organization: $R = 2000$ blocks and $\lambda = 60$ req/sec. Note that the values in parenthesis are obtained from simulation

r_{req} (ms)	50	100	250	500	1000	2000	3000
N	140	70	48	39	36(35)	32	32(30)
P_a	4	5	48	13	9(7)	16	16(15)
P_s	35	14	1	3	4(5)	2	2
P_x	1	1	1	1	1	1	1

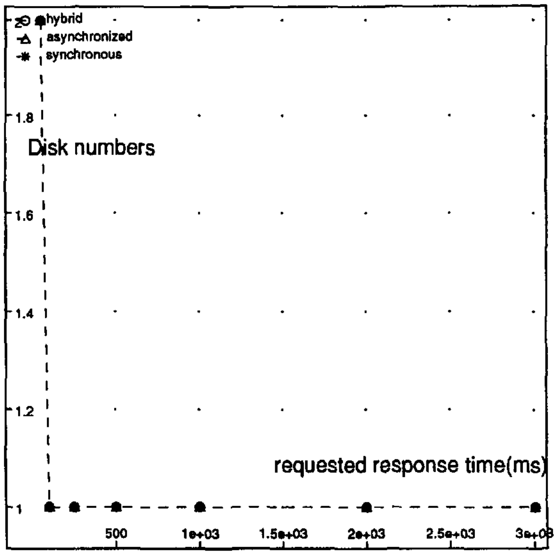

Fig. 2. Comparisons of the number of disks in the three organizations: λ = 1 and R = 200 blocks.

We can see from Table 3 that the number of groups P_a decreases and the number of synchronous disks in a group P_s increases as r_{req} decreases. And the striping width is found to be one both in simulation and analysis. This shows that a significant synchronization overhead exists among groups even in asynchronous organizations.

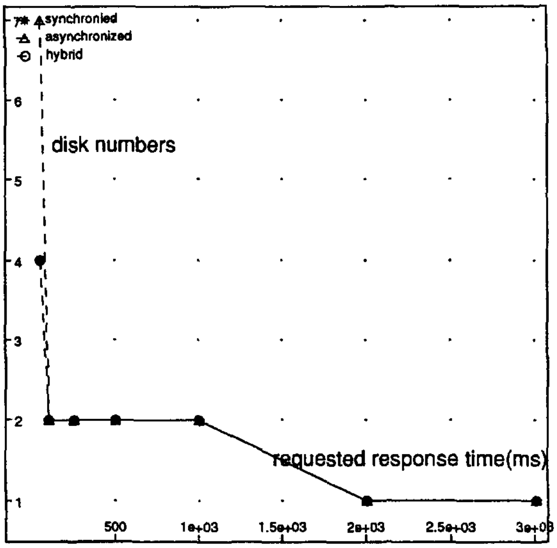

Fig. 3. Comparisons of the number of disks in the three organizations: λ = 15 and R = 200 blocks.

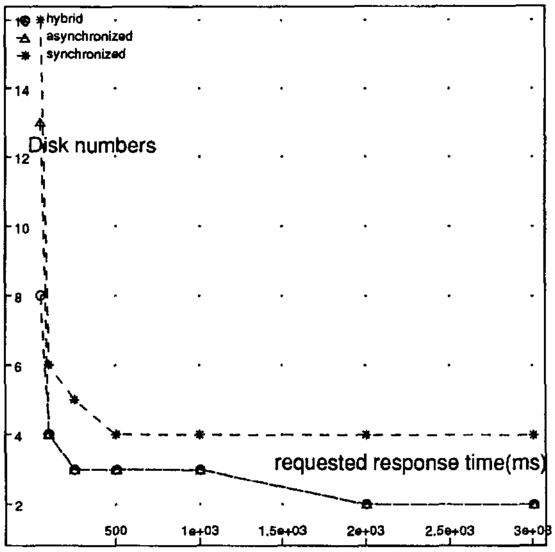

Fig. 4. Comparisons of the number of disks in the three organizations: $\lambda = 30$ and $R = 200$ blocks.

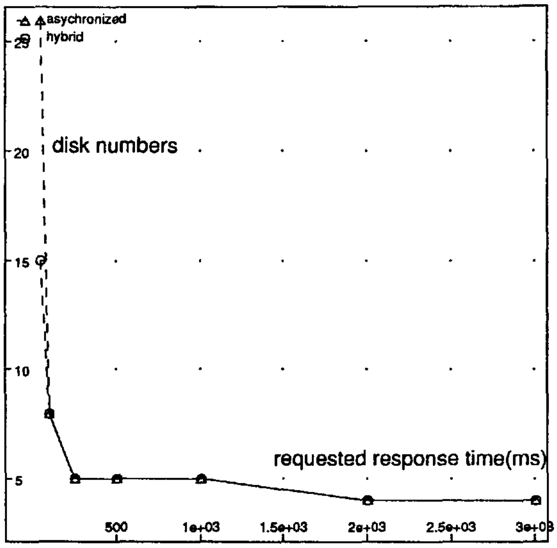

Fig. 5. Comparisons of the number of disks in the three organizations: $\lambda = 60$ and $R = 200$ blocks.

We now compare three organizations from the viewpoint of total number of disks. Figs. 2–9 show the minimum number of disks (vertical axis) that are needed to achieve a specified average response time (horizontal axis) for a given R, λ, and r_{req}.

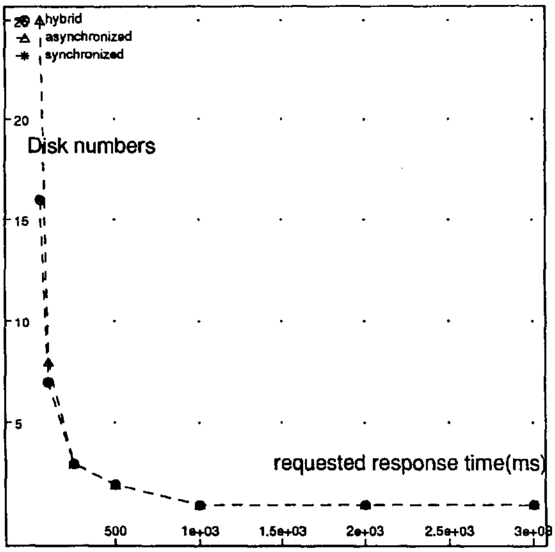

Fig. 6. Comparisons of the number of disks in the three organizations: $\lambda = 1$ and $R = 2000$ blocks.

In the case of small request size $R = 200$, all organizations behave similarly when the arrival rate is very low, i.e., $\lambda = 1$. If we increase the arrival rate to 15 ($\lambda = 15$), synchronous and hybrid organizations require smaller number of disks than asynchronous organizations when $r_{\text{req}} = 50$ ms. If we relax response time

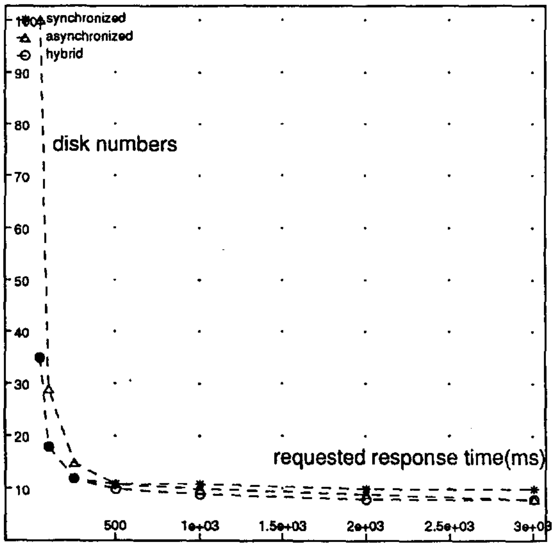

Fig. 7. Comparisons of the number of disks in the three organizations: $\lambda = 15$ and $R = 2000$ blocks.

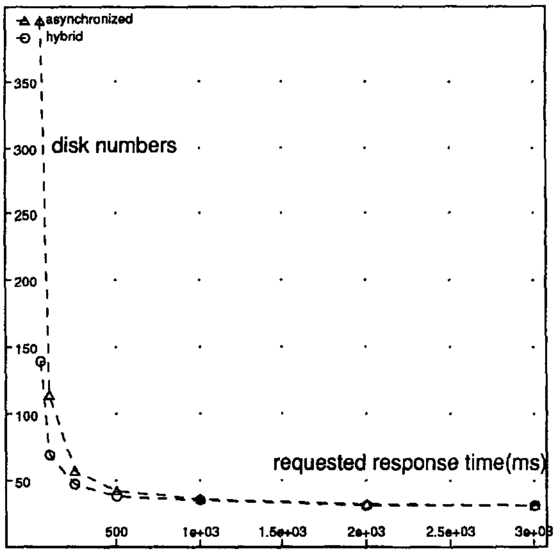

Fig. 9. Comparisons of the number of disks in the three organizations: $\lambda = 60$ and $R = 2000$ blocks.

requirement by increasing $r_{\mathrm{req}} \geqslant 100$ ms, no difference will be found among organizations. If the arrival rate is increased further to 30 ($\lambda = 30$), synchronous organizations need a larger number of disks than asynchronous and hybrid organizations for any value of r_{req}. Since the subrequest arrival rate to a disk (λ_d) in synchronous

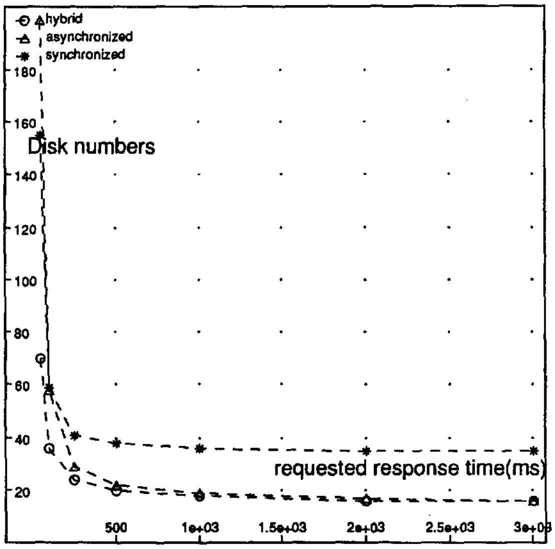

Fig. 8. Comparisons of the number of disks in the three organizations: $\lambda = 30$ and $R = 2000$ blocks.

Table 4
Critical arrival rate λ_0 in synchronous organizations

r_{req} (ms)	50	100	250	500	1000	2000	3000
λ_0 (req/sec)	36.66	44.01	47.76	48.92	49.48	49.76	49.85

organizations are equal to the original arrival rate λ, synchronous organizations can not meet any value r_{req} when the arrival rate is increased to 60, i.e., $\lambda = 60$.

In the case of large request size $R = 2000$, all organizations require the same number of disks for all values of r_{req} (except $r_{\mathrm{req}} = 50, 100$) when $\lambda = 1$. If the arrival rate is increased to 15 ($\lambda = 15$), synchronous organizations are better than asynchronous ones for $50 \leqslant r_{\mathrm{req}} \leqslant 500$ whereas asynchronous ones are better than synchronous ones for $r_{\mathrm{req}} \geqslant 500$. If the arrival rate is further increased to 30 ($\lambda = 30$), synchronous organizations are better than asynchronous ones only for $r_{\mathrm{req}} = 50$ while asynchronous ones are much better for other values of r_{req}. Synchronous organizations cannot meet any value of r_{req} when $\lambda = 60$, as expected. In all, we can see that hybrid organizations always require smaller number of total disks than synchronous and asynchronous organizations.

Note that synchronous organizations cannot meet r_{req} if the arrival rate λ gets larger than a certain value which we call a critical arrival rate denoted as λ_0 (refer to Table 4). In other words, the arrival rate at each disk remains the same as λ in synchronous organizations so that queueing delay at each disk increases rapidly as λ increases. We now derive a critical arrival rate λ_0.

The expected response time of a synchronous organization with P_s synchronous disks can be derived from Eq. (18) as

$$s_{\mathrm{d}} + \rho \frac{s_{\mathrm{d}}(1+\alpha^2)}{2(1-\rho)}, \tag{19}$$

where $\rho = \lambda_{\mathrm{d}} \times s_{\mathrm{d}}$, $\alpha = \sigma_{\mathrm{d}}/s_{\mathrm{d}}$, and $s_{\mathrm{d}} = \mathrm{E}[D_{\mathrm{d}}]$ with the size of $R_{\mathrm{d}} = R/P_{\mathrm{s}}$. Note that $\mathrm{E}[D_{\mathrm{d}}]$ can be obtained from Eq. (7). Then, λ_0 can be found by setting $R = 0$ in Eq. (19) where the best case is obtained. Therefore, we get

$$\mathrm{E}[D_{\mathrm{d}}]\,|_{R=0} + \frac{\rho \mathrm{E}[D_{\mathrm{d}}]\,|_{R=0}(1+\alpha^2)}{2(1-\rho)} \leqslant r_{\mathrm{req}}. \tag{20}$$

Rewriting Eq. (20),

$$\lambda_0 = \frac{-2\left(r_{\mathrm{req}} - \mathrm{E}[D_{\mathrm{d}}]\,|_{R=0}\right)}{-\sigma_D^2 - 2r_{\mathrm{req}}\mathrm{E}[D_{\mathrm{d}}]\,|_{R=0} + \mathrm{E}[D_{\mathrm{d}}]\,|_{R=0}^2}. \tag{21}$$

Table 4 shows λ_0 for given r_{req}.

4. Conclusions

We have studied the performance implications of various alternative organizations available for a multiple disk system. For our analysis model, we have used the characteristic maximum approximation [10] for asynchronous organizations, and M/G/1 queueing model for synchronous organizations. Our analysis model can be used to determine important design parameters of multiple disk systems given that the I/O workload is characterized by mean request size, mean arrival rate, and mean response time requirement. Design parameters include the number of synchronous disks constituting a group, the number of asynchronous groups/disks, and the striping width. Simulation results show that our analysis model can be a good approximation of the actual performance of a disk system, and is useful to determine design parameters.

References

[1] P.M. Chen and D.A. Patterson, Maximizing performance in a striped disk array, in: *Proc. SIGARCH 17th Ann. Internat. Symp. on Computer Architecture*, 1990.

[2] S. Chen and D. Towsley, A queueing analysis of RAJD architectures, Dept. of Computer and Information Science, University of Massachusetts, 1992.

[3] A. Gravey, A simple construction of an upper bound for the mean of the maximum, *J. Appl. Probab.* 22 (1985) 844–851.

[4] K.-S. Hwang and C.-I. Park, Performance analysis of disk array architectures for supporting I/O characteristics, TR-SS-94-01, Dept. of Computer Science and Engineering, POSTECH, 1994.

[5] R. Jain, *The Art of Computer Systems Performance Analysis* (Wiley, New York, 1991).

[6] E.K. Lee and R.H. Katz, An analytic performance model of disk arrays and its application, in: *Proc. ACM SIGMETRICS*, Santa Clara, 1993.

[7] M.Y. Kim, Synchronized disk interleaving, *IEEE Trans. Comput.* 35 (11) (1986) 978–988.

[8] M.Y. Kim and A.N. Tantawi, Asynchronous disk interleaving: Approximating access delays, *IEEE Trans. Comput.* 40 (7) (1991) 801–810.

[9] L. Kleinrock, *Queueing Systems*, Vol. 1 (Wiley/Interscience, New York, 1975).

[10] C.P. Kruskal and A. Weiss, Allocating independent subtasks on parallel processors, *IEEE Trans. Software Engrg.* 11 (10) (1985).

[11] M. Livny, S. Khoshafian and H. Boral, Multi-disk management algorithms, *Proc. ACM SIGMETRICS* (May 1987) 69–77.

[12] A.L. N. Reddy and P. Banerjee, An evaluation of multiple-disk I/O systems, IEEE Trans. Comput. 38 (12) (1989) 1680–1691.

[13] K. Salem and H. Garcia-Molina, Disk striping, in: *Proc. Internat. Conf. on Data Engineering* (1986) 336–342.

[14] *ST1480 Family: ST1400N, ST1401N, ST1480N Product Manual* Vol. 1, (Segate Technology, Inc., 1992).

[15] G. Weikum, P. Zabback and P. Scheuermann, Dynamic file allocation in disk arrays, in: *Proc: ACM SIGMOD Conf.*, Denver, 1991.

[16] G. Weikum and P. Zabback, Tuning of striping units in disk-array-based file systems, in: *Proc. 2nd IEEE Internat. Workshop on Research Issues in Data Engineering*, 1991.

Chan-Ik Park is an associate professor in the Department of Computer and Engineering at the Pohang University of Science and Technology, Pohang, Korea. He has received a BS from the Seoul National University in 1979, and an MS and PhD from the Korea Advanced Institute of Science and Technology in 1985 and 1988. After graduation, he joined the Department of Computer Science and Engineering, Pohang University of Science and Technology. He visited IBM Thomas J. Watson Research Center for one year starting from February 1991. His research interests include operating systems, parallel processing, and distributed real-time systems.

Kyung-Sook Hwang has received a BS and MS from Pohang University of Science and Technology in 1991 and 1994. She has been working for POSDATA as a system engineer since 1991 in the area of client-server system and executive information management. Her research interests include operating system, distributed client-server system, and database system.