

## 5.3 存储设备

主要内容  
磁盘管理  
磁盘调度的方法  
磁盘阵列

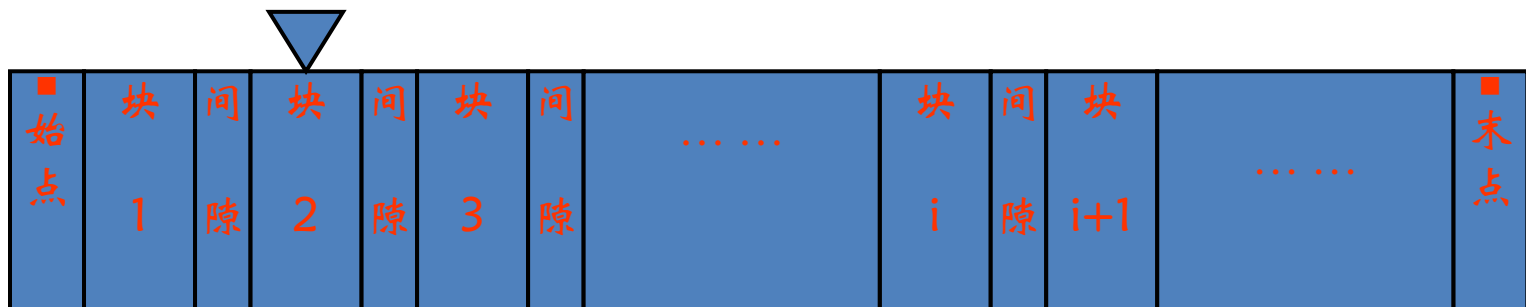
# 主要目录

- **5.3 存储设备**
  - **5.3.1 存储设备的物理结构**
  - **5.3.2 磁盘调度**
  - **5.3.3 磁盘阵列**

## 5.3.1 存储设备的物理结构(1)

- **顺序存取存储设备**是严格依赖信息的物理位置进行定位和读写的存储设备
- 具有存储容量大、稳定可靠、卷可装卸和便于保存等优点

磁头(正走,反走,正读,反读,正写,反写,倒带)

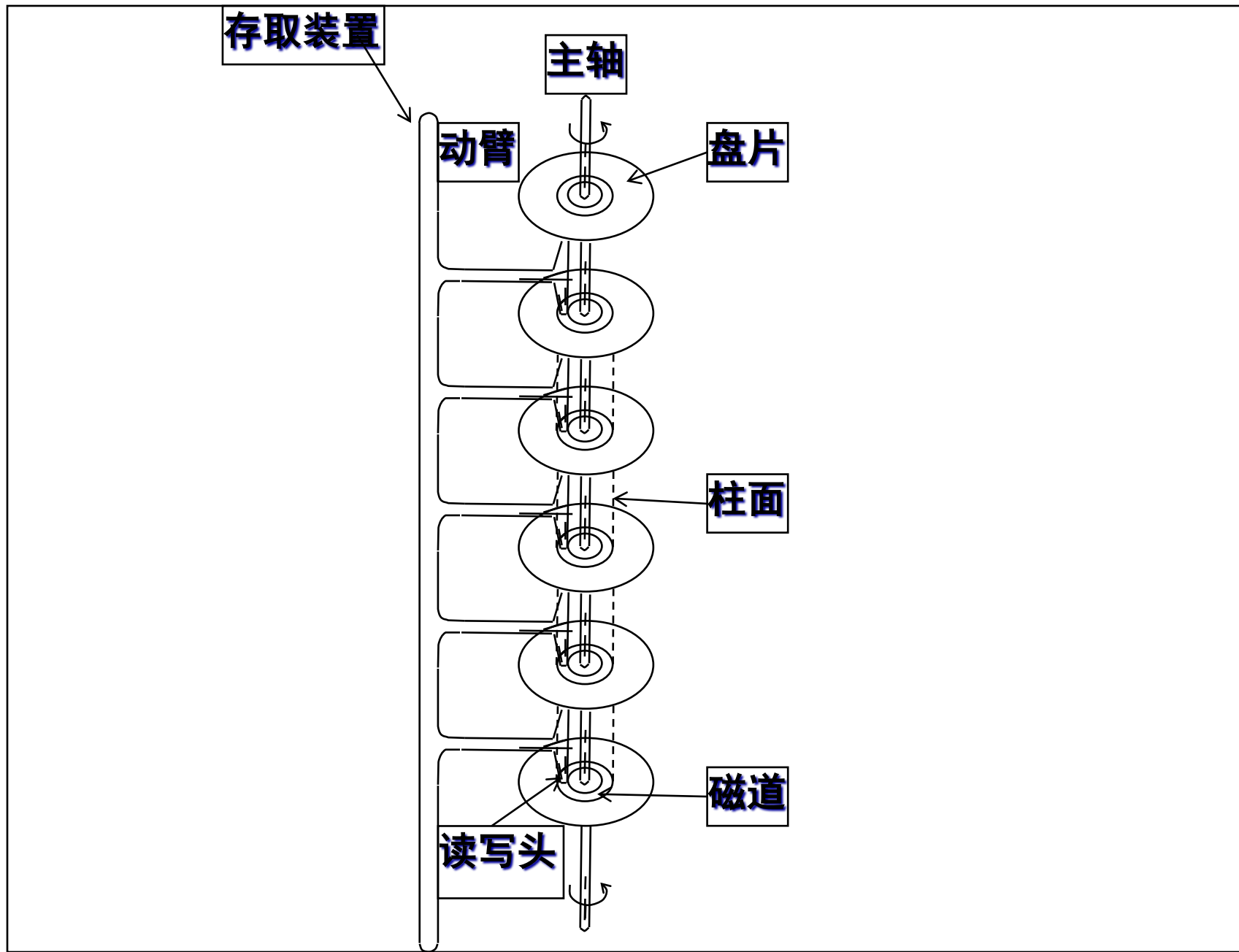


# 存储设备的物理结构(2)

## 直接存取存储设备

- 磁盘是一种直接(随机)存取存储设备。每个物理记录有确定的位置和唯一的地址，存取任何一个物理块所需的时间几乎不依赖于此信息的位置。
- 访问磁盘记录参数：柱面号、磁头号、块号

# 多磁头活动头盘示意图



## 5.3.2 磁盘调度

**磁盘地址：台号+柱面号+磁头号（盘面）+扇区号**

**读写一次磁盘信息所需的时间可分解为：  
查找时间、搜索延迟、传输时间**

**为提高磁盘传输效率，软件应着重考虑减少查找时间和搜索延迟**

# 1 循环排序

- 旋转型存储设备减少延迟时间，设定输入输出请求有现实意义
- 考虑磁道保存4个记录的旋转型设备，假定收到四个I/O请求。

请求次序 记录号

(1) 读记录4

(2) 读记录3

(3) 读记录2

(4) 读记录1

# 循环排序

- 多种I/O请求排序方法

方法1：按照I/O请求次序读记录4、3、2、1，平均用 $1/2$ 周定位，再加上 $1/4$ 周读出记录，总处理时间等于3周，即60毫秒

$$1/2 + 1/4 + 3 * 3/4 = 3 \text{ 周}$$

- 方法2：如果次序为读记录1、2、3、4。总处理时间等于1.5周，即30毫秒

$$1/2 + 1/4 + 3 * 1/4 = 1.5 \text{ 周}$$

- 方法3：如果知道当前读位置是记录3，则采用次序为读记录4、1、2、3。总处理时间等于1周，即20毫秒

$$1/4 + 3 * 1/4 = 1 \text{ 周}$$



## 2 优化分布(1)

- 信息在存储空间中的排列方式会影响存取等待时间。考虑10个逻辑记录A, B....., J被存于旋转型设备上, 每道存放10个记录, 安排如下:

物理块

逻辑纪录

1-10

A-J

- 处理10个记录的总时间

10毫秒(移动到记录A的平均时间)+ 2毫秒(读记录A)+4毫秒(处理记录A)+9×[16毫秒(访问下一记录) +2毫秒(读记录)+4毫秒(处理记录)] = 214毫秒

# 优化分布(2)

按照下面方式对信息优化分布

物理块

逻辑纪录

1	A
2	H
3	E
4	B
5	I
6	F
7	C
8	J
9	G
10	D

# 优化分布(3)

处理10个记录的总时间为

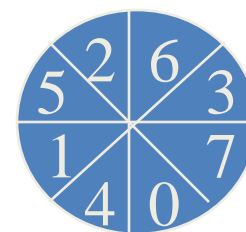
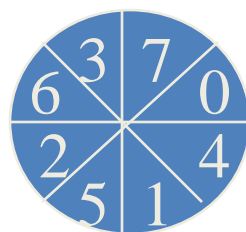
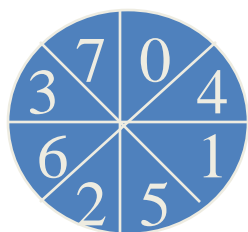
$$\begin{aligned} & 10 \text{ 毫秒 (移动到记录A的平均时间)} + 10 \times [2 \text{ 毫秒 (读记录)} \times 4 \\ & \quad \text{毫秒 (处理记录)}] \\ & = 70 \text{ 毫秒} \end{aligned}$$

### 3 交替地址

- 每个记录重复记录在设备的多个区域，读相同的数据，有几个交替地址，也称为多重副本或折迭。
- 成功与否取决于下列因素：数据记录总是读出使用，不需修改写入；数据记录占用的存储空间总量不太大；数据使用极为频繁。

# 减少延迟时间的方法

一般常将盘面扇区交替编号；磁盘选中不同盘面错开命名。



# 调度策略

- 减少寻道时间
- 减少延迟时间

## 5.3.2 搜查定位(1)

- 移臂调度有若干策略

(1)“电梯调度”算法

(2)“最短查找时间优先”算法

(3)“扫描”算法

(4)“分步扫描”算法

(5)“单向扫描”算法

## “最短查找时间优先”算法

本算法考虑了各个请求之间的区别，总是先执行查找时间最短的那个磁盘请求，从而，较“先来先服务”算法有较好的寻道性能。



## “扫描”算法

磁盘臂每次沿一个方向移动，扫过所有柱面，遇到最近的I/O请求便进行处理，直到最后一个柱面后，再向相反方向移动回来。

## “分步扫描”算法

将I/O请求分成组，每组不超过N个请求，每次选一个组进行扫描，处理完一组后再选下一组。

# 磁盘调度

模拟磁盘调度CSCAN算法

按磁道移动方向

☒ 按磁道增加方向移动

☐ 按磁道减小方向移动

输入数据：

12  
3  
3  
12  
34  
4  
5  
6  
4  
12  
23  
4  
54  
56

加载数据

模拟结果

输入当前磁道号：

3

磁道序列

3  
3  
4  
4  
5  
6  
12  
12  
12  
23  
34  
54  
56

移动距离

0  
0  
1  
0  
0  
1  
1  
6  
0  
0  
11  
11  
20  
2

平均寻道长度：

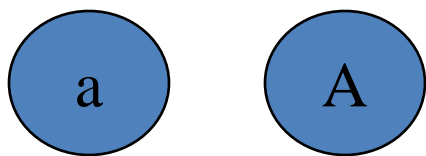
6.8125

运行模拟过程

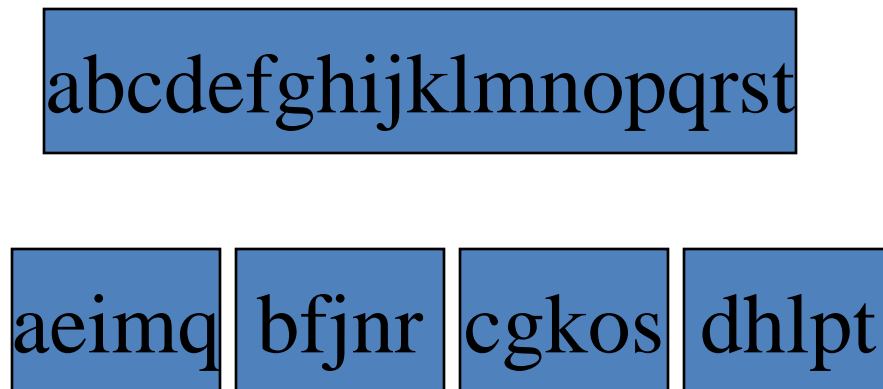
### 5.3.3 盘阵

**通过冗余提高可靠性：**如建立镜像盘。

**通过并行性提高性能：**如将原来在一个物理盘连续的数据分条分布到多盘。



A为a的备份



将数据分布到多个盘中

# 磁盘容错

- 计算机硬盘又是一位娇嫩的“千金”，与其它电脑部件相比又十分“脆弱”，单颗硬盘容量有限
- **RAID容错技术**是美国加州大学伯克莱分校的Patterson教授与1988年首先提出了**RAID(Redundant Array of Inexpensive Disk)**技术，并说明了RAID0至RAID5六种不同级别的定义

# 最初研制RAID的目的

- 希望将廉价的数块小容量的硬盘组合成一块逻辑大硬盘，以代替昂贵的单颗大容量硬盘，从而降低数据存储的费用
- 采用冗余数据的方式，使得单个硬盘故障时不会影响到正常的数据访问，从而开发出一种数据保护技术
- 把速度提高到单个硬盘驱动器的数倍磁盘阵列把多个磁盘驱动器连接在一起协同工作，大大提升了速度

并行交叉存取

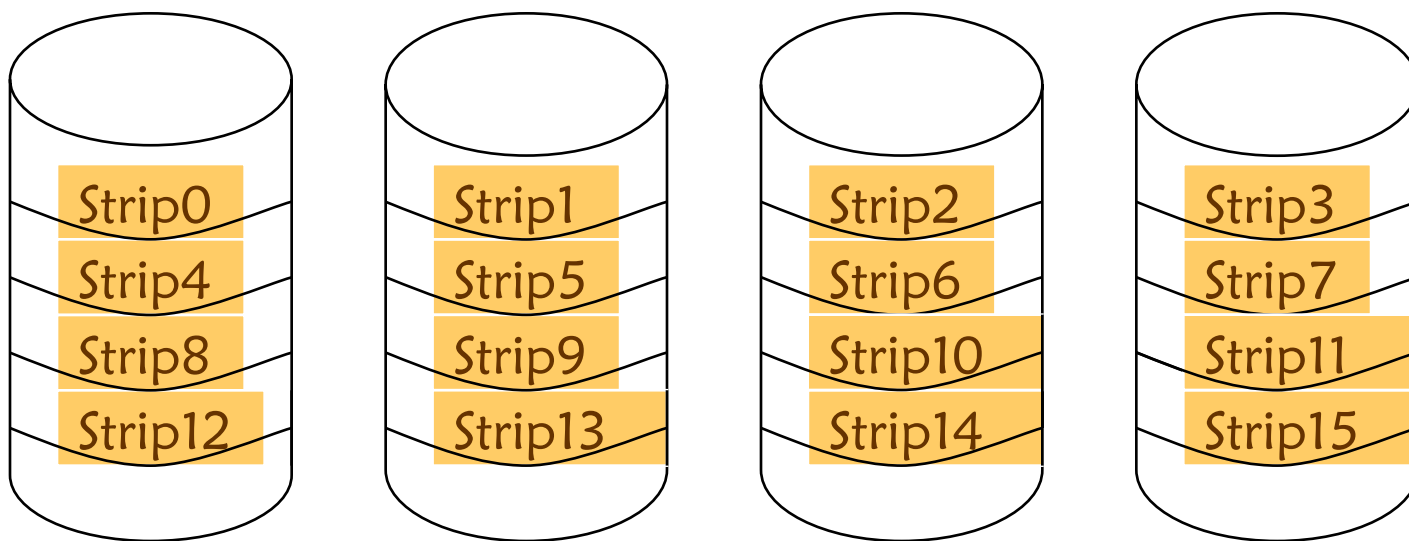
# 独立磁盘冗余阵列

## RAID level 0(1)

- 数据划成条块被分布存储在横跨阵列中的所有磁盘上
- 逻辑上连续的数据条块，在物理上可被依次存储在横向相邻的磁盘驱动器上
- 通过阵列管理软件进行逻辑地址空间到物理地址空间的映射
- 优点和缺点

# 独立磁盘冗余阵列

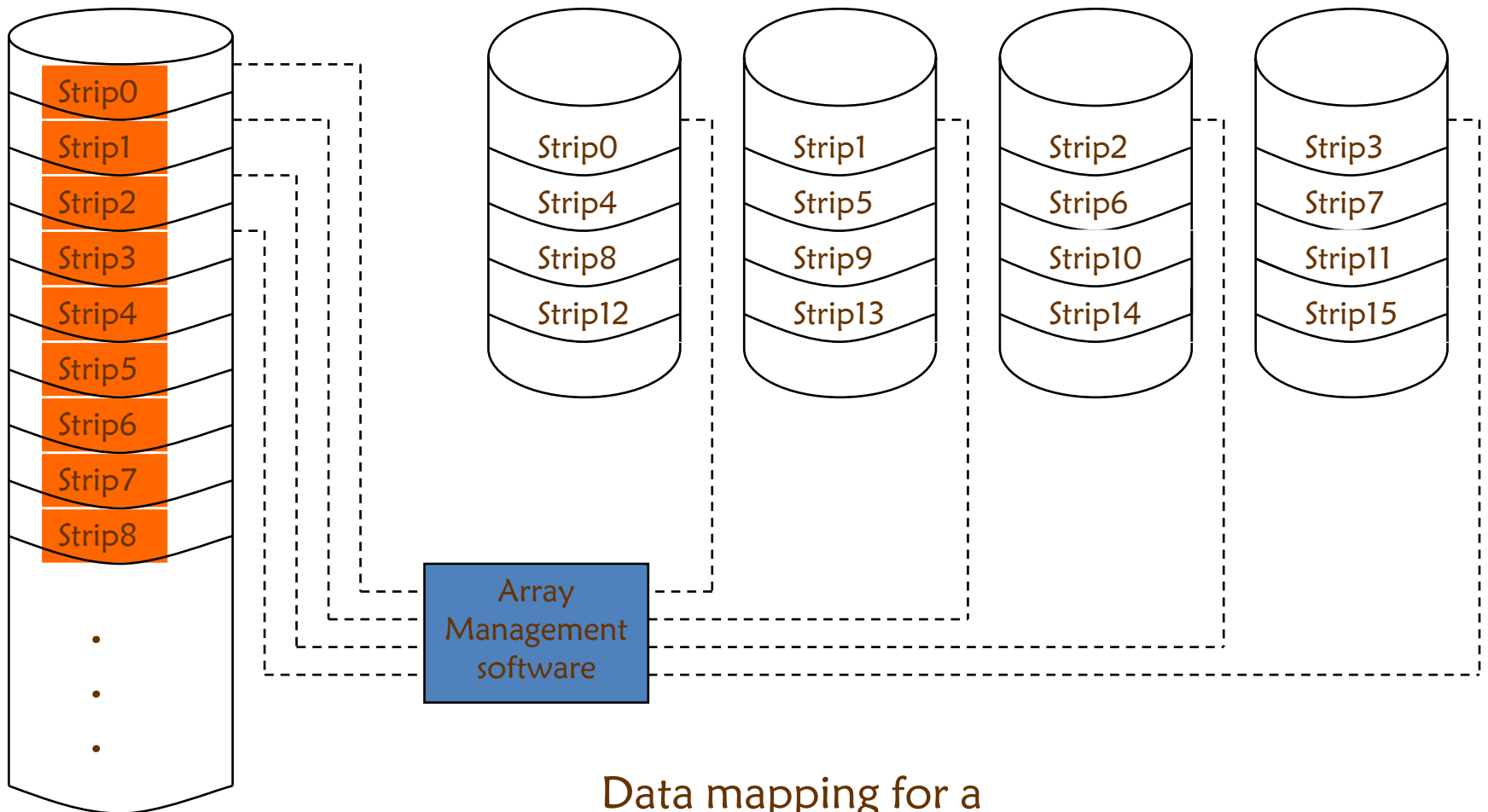
RAID Level 0(1)





# 独立磁盘冗余阵列

## RAID level 0(2)



Data mapping for a  
RAID Level0 Array

# 独立磁盘冗余阵列(6)

## RAID level 1(1)

- 双份数据，每个盘都有一个包含相同数据的镜像盘。

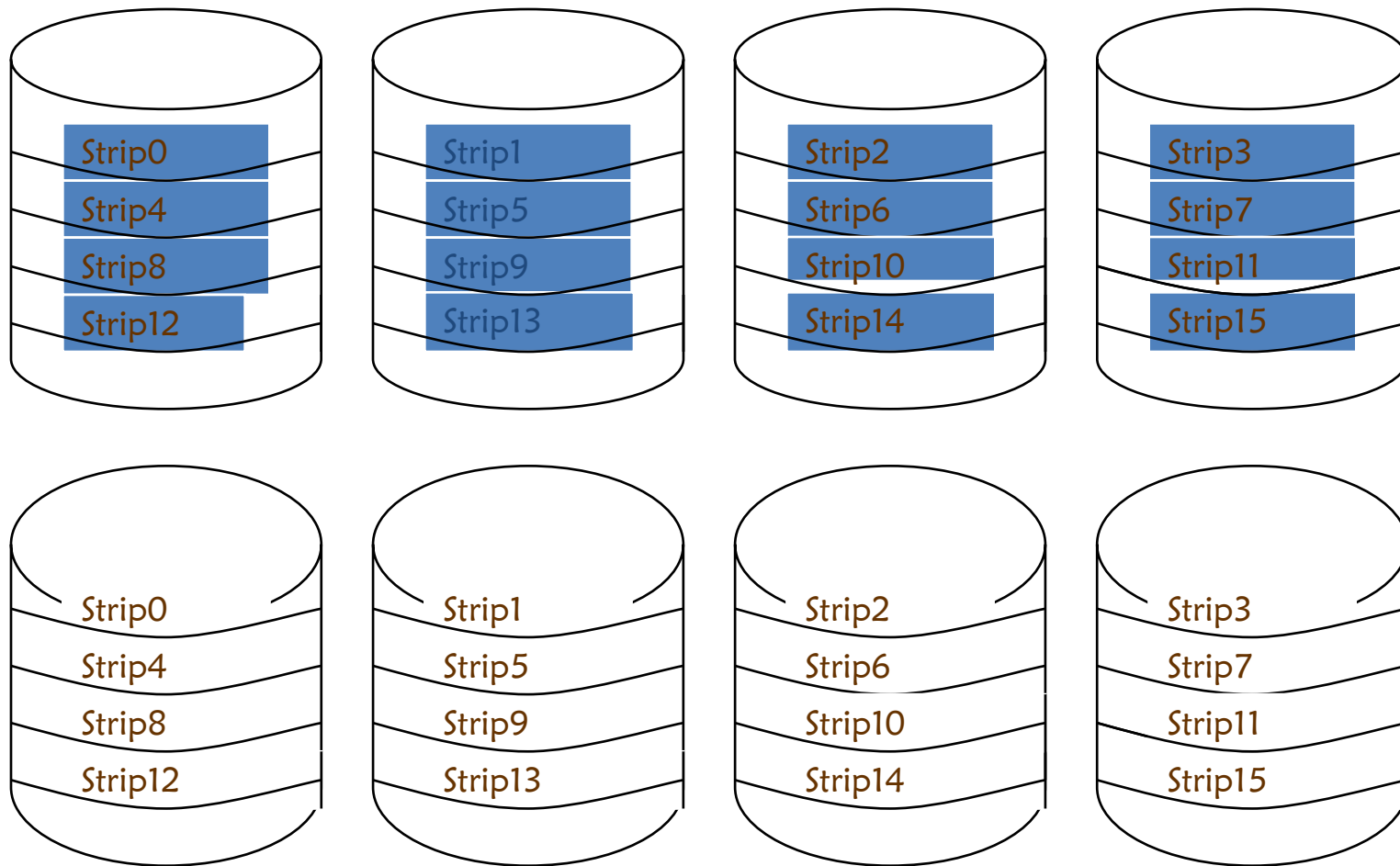
①读请求能通过包含相同请求数据中的任何一个磁盘提供服务，其中的一个所化查找和搜索时间最少；

②写操作时，要求改写对应的两个数据子块，可采用并行操作，写操作的性能由并行操作中较慢的一个决定；

③一个驱动器出现故障，数据可以从镜像盘获得。

# 独立磁盘冗余阵列(5)

## RAID level 1



RAID Level 1 (Mirrored)

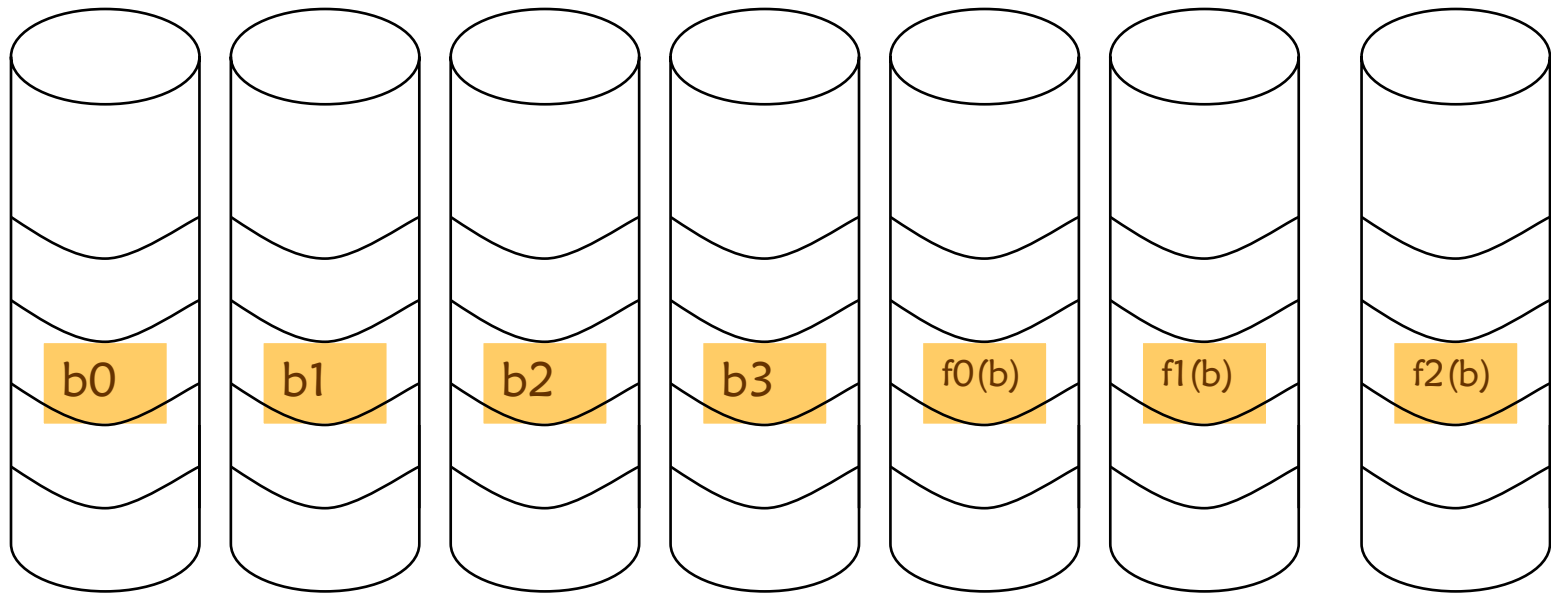
# 独立磁盘冗余阵列(8)

## RAID level 2 (1)

- 采用并行存取技术，驱动器的移动臂同步工作，每个磁盘的磁头都在相同位置。纠错码按照横跨的每个数据盘的相应位计算，并存储在多只奇偶校验盘的相应位的位置。
- 奇偶校验磁盘的数量与数据盘的多少成比例。
- 执行单个读操作时的操作：
- 执行单个写请求时的操作：

# 独立磁盘冗余阵列(6)

## RAID level 2



RAID Level 2 (Redundancy through Hamming Code)

# 独立磁盘冗余阵列(10)

## RAID level 3(1)

- RAID3仅使用一只冗余盘，出现故障时，使用奇偶校验盘的信息校验，数据可用剩下的磁盘的信息重新构造，若X0到X3存放数据，X4为奇偶盘，对于第i位的奇偶校验位可如下计算：

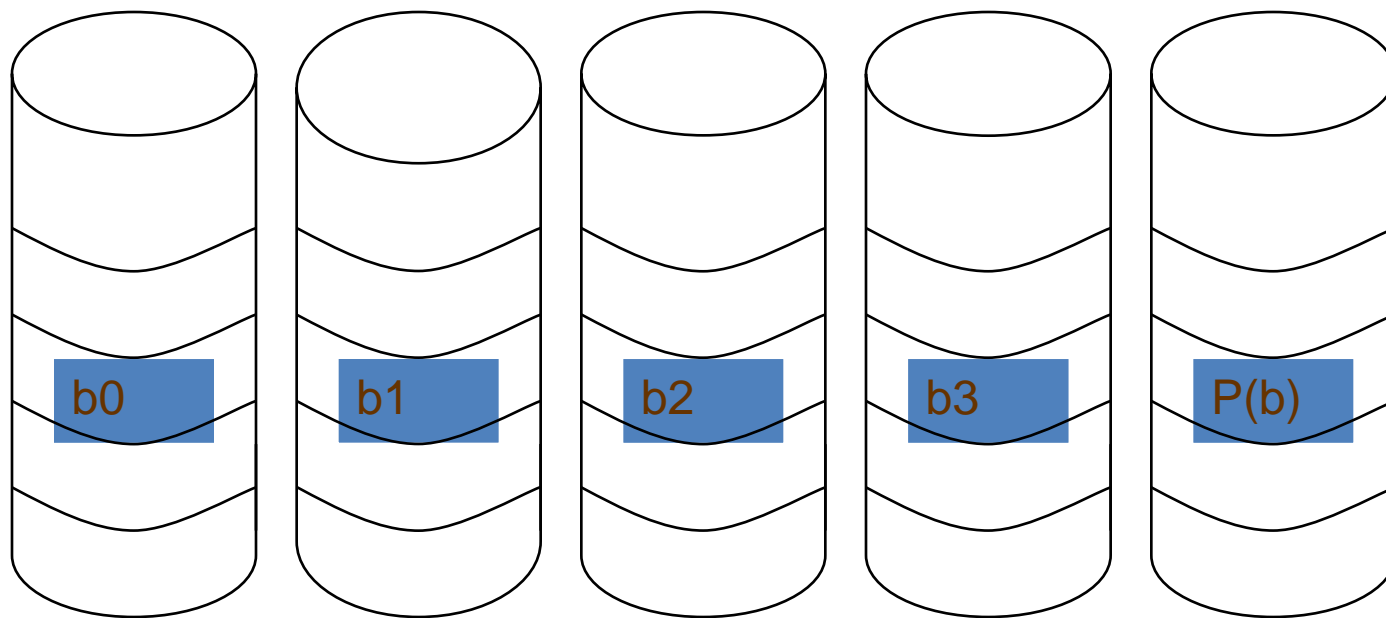
$$X4(i) = X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i)$$

假定驱动器X1出故障，如果把X4(i)，X1(i)加到上面等式两边，得到

$$X1(i) = X4(i) \oplus X3(i) \oplus X2(i) \oplus X0(i)$$

# 独立磁盘冗余阵列(7)

## RAID level 3



RAID Level 3 (Bit interleaved  
Parity)

# 独立磁盘冗余阵列(12)

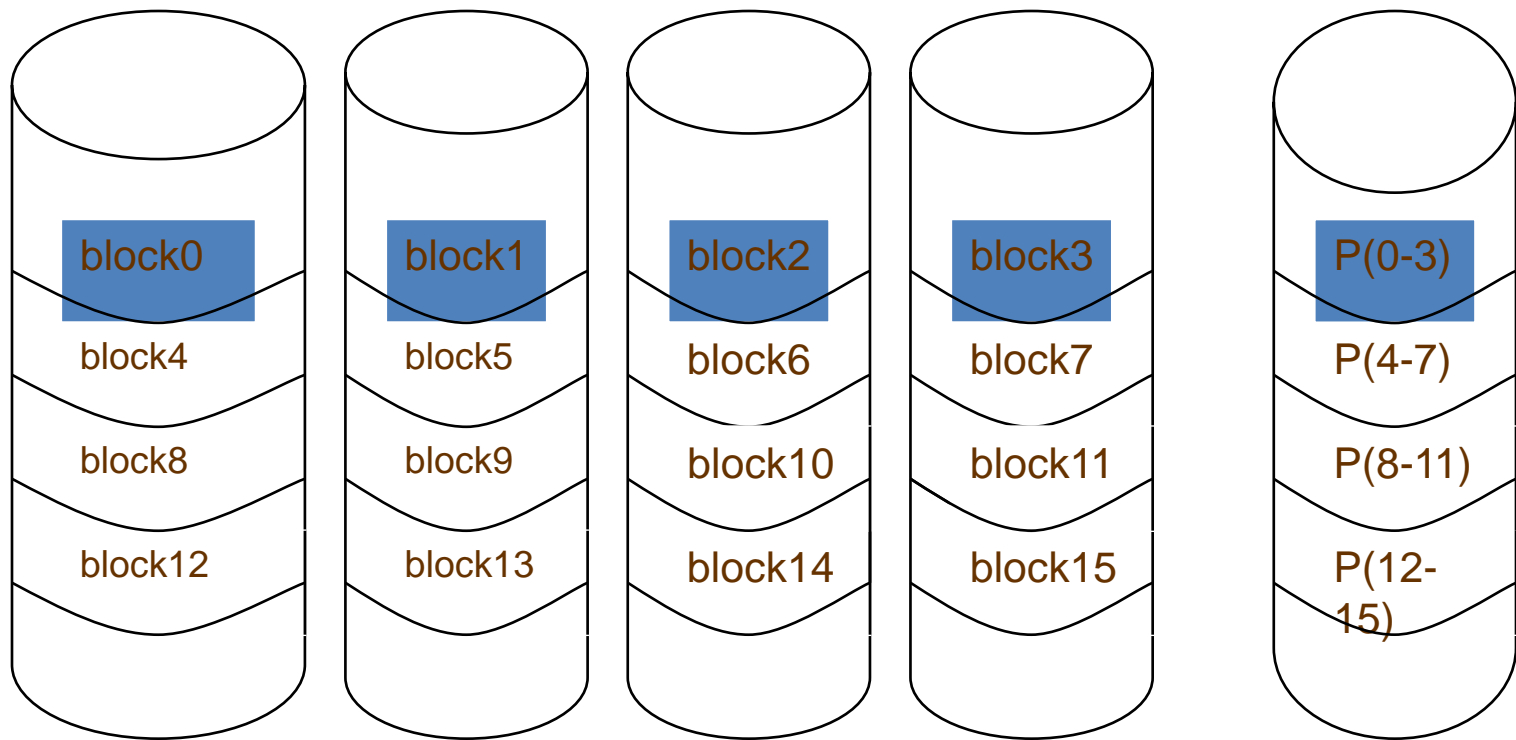
## RAID level 4(1)

- RAID4和RAID5使用独立存取技术，在一个独立存取的磁盘阵列中，每个驱动器都可以独立地工作，所以，独立的I/O请求可以被并行地得到满足。因此独立存取阵列适合于有频繁I/O请求的应用。



# 独立磁盘冗余阵列(8)

## RAID level 4



RAID Level 4 (Block level Parity)

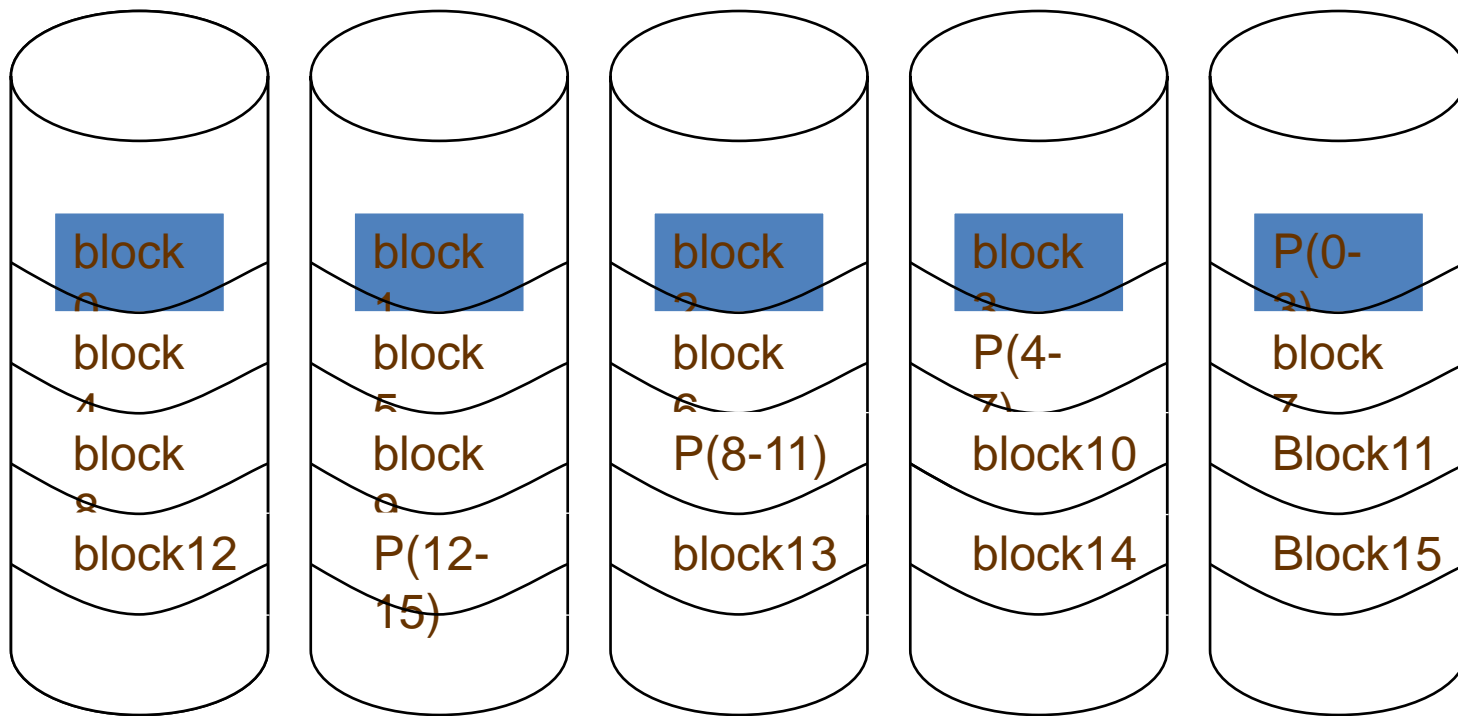
# 独立磁盘冗余阵列(17)

## RAID level 5(1)

- RAID5的奇偶校验码是分布横跨轮转存放在所有的磁盘上，设有n个磁盘的阵列，则开头的n个奇偶校验码螺旋式地位于n个磁盘上，能避免RAID4发生的奇偶校验盘瓶颈口问题。

# 独立磁盘冗余阵列(9)

## RAID level 5



RAID Level 5 (Block level Distributed parity)

# 独立磁盘冗余阵列(19)

## RAID level 6和RAID level 7

- 增强型RAID。RAID6中设置了专用快速的异步校验磁盘，具有独立的数据访问通路，比低级RAID性能更好，但价格昂贵。
- RAID7对RAID6作了改进，该阵列中的所有磁盘都有较高传输速率，性能优异，但价格也很高。