第八章 输入输出系统

冯 丹
华中科技大学计算机科学与技术学院

第八章 输入输出系统

8.1 I/O系统性能

8.2 I/O总线

8.3 廉价磁盘冗余阵列RAID

8.4 I/O系统的可靠性、可用性和可信性

8.1 I/O系统的性能

1. **输入/输出系统简称I/O系统**

   它包括：

   - I/O设备
   - I/O设备与处理机的连接

2. **I/O系统是计算机系统中的一个重要组成部分**

   - 完成计算机与外界的信息交换
   - 给计算机提供大容量的外部存储器

3. **按照主要完成的工作进行分类：**

   - 存储I/O系统（本章内容）
   - 通信I/O系统

8.1 I/O系统的性能

4. **I/O系统的性能对CPU的性能有很大的影响**，若两者的性能不匹配，I/O系统就有可能成为整个系统的瓶颈。
5. **系统的响应时间**(衡量计算机系统的一个更好的指标)
   - 从用户输入命令开始，到得到结果所花费的时间。
   - 由两部分构成：
     - I/O系统的响应时间
     - CPU的处理时间
6. **多进程技术只能够提高系统吞吐率**，并不能够减少系统响应时间。

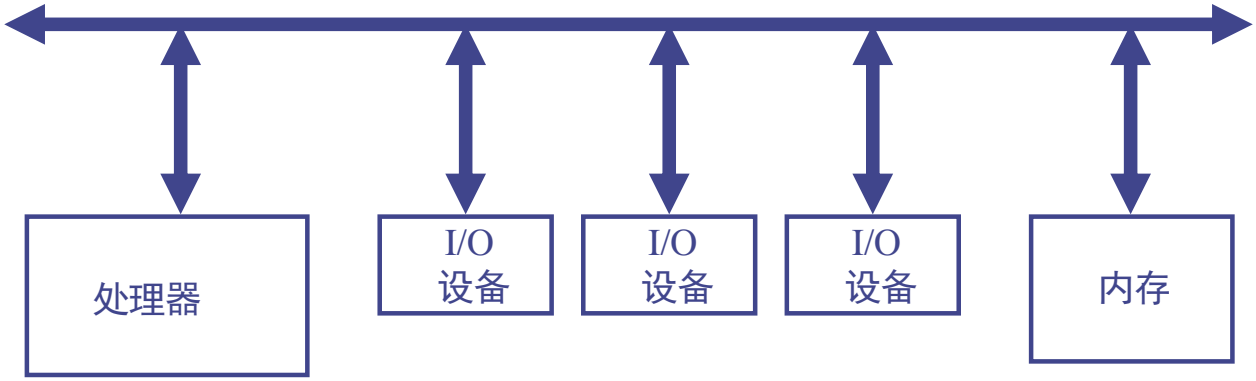
8.1 I/O系统的性能

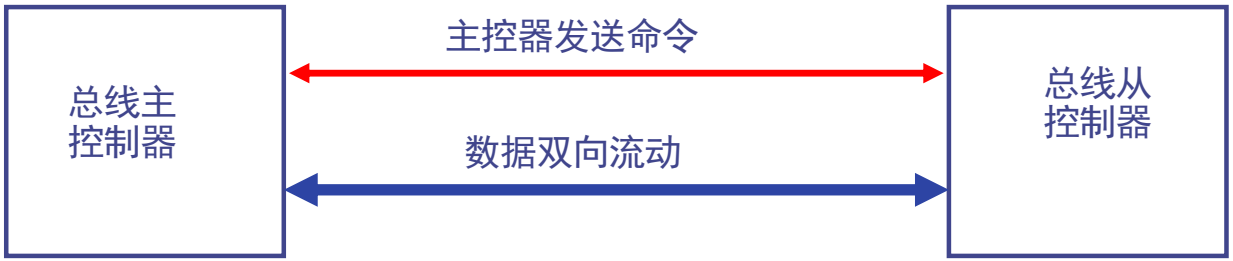
7. **评价I/O系统性能的参数主要有：**
   - 连接特性

     （哪些I/O设备可以和计算机系统相连接）
   - I/O系统的容量

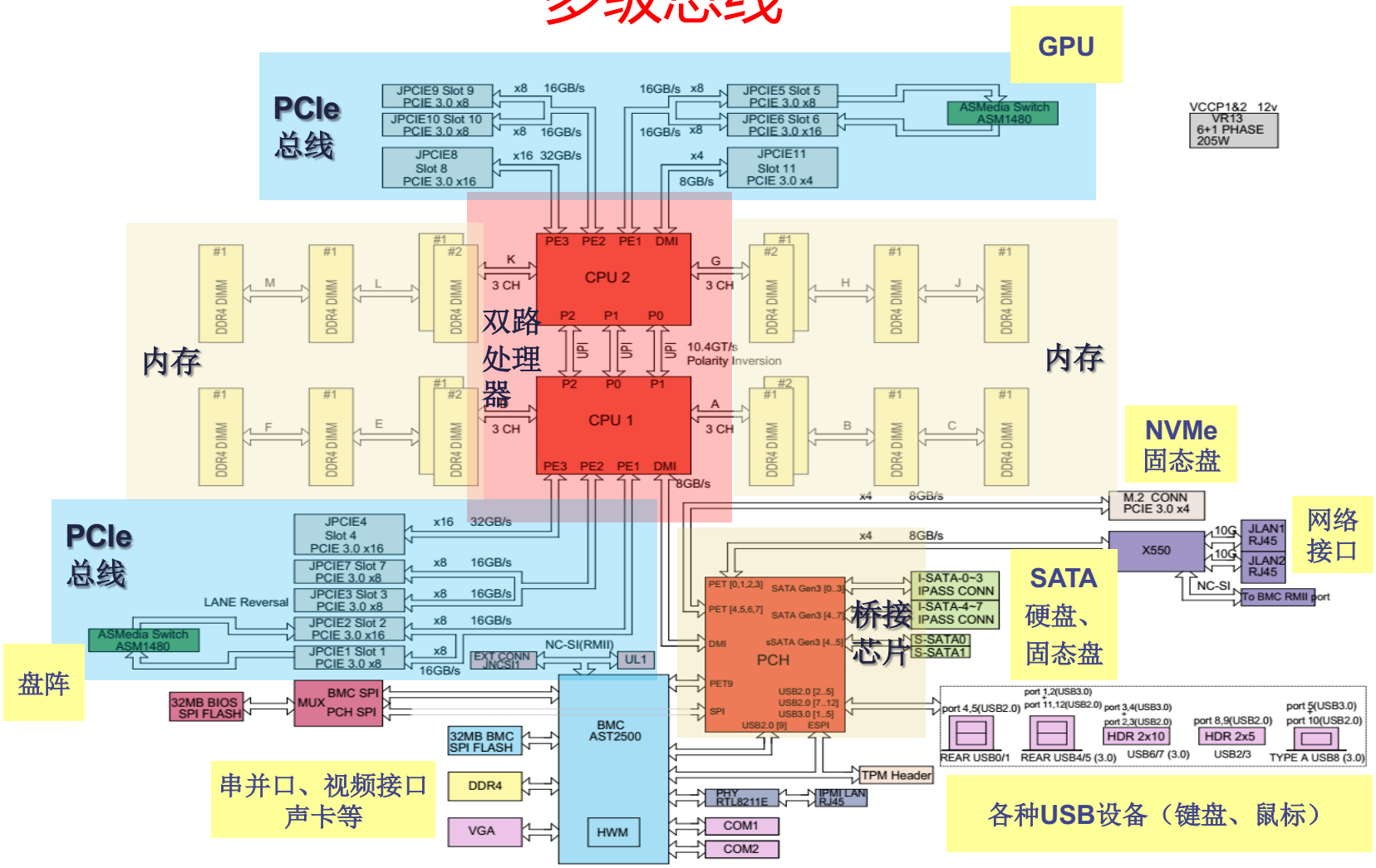
     （I/O系统可以容纳的I/O设备数）
   - 响应时间和吞吐率等
8. **另一种衡量I/O系统性能的方法：考虑I/O操作对CPU执行的打扰(interference)情况。**
   - 即考查由于其他进程的I/O操作，使得某个进程的执行时间增加了多少。

8.2 输入输出总线





多级总线


GPU
PCIe
总线
JPCIE9 Slot 9
PCIE 3.0 x8
JPCIE10 Slot 10
PCIE 3.0 x8
JPCIE8
Slot 8
PCIE 3.0 x16
JPCIE5 Slot 5
PCIE 3.0 x8
JPCIE6 Slot 6
PCIE 3.0 x16
JPCIE11
Slot 11
PCIE 3.0 x4
ASMedia Switch
ASM1480
16GB/s
8GB/s
VCCP1&2 12v
VR13
6+1 PHASE
205W
内存
双路
处理
器
CPU 2
CPU 1
PE3 PE2 PE1 DMI
P2 P1 P0
UPI
10.4GT/s
Polarity Inversion
3 CH
DDR4 DIMM
内存
NVMe
固态盘
M.2 CONN
PCIE 3.0 x4
X550
JLAN1
RJ45
JLAN2
RJ45
10G
NC-SI
To BMC RMII port
网络
接口
PCIe
总线
JPCIE4
Slot 4
PCIE 3.0 x16
JPCIE7 Slot 7
PCIE 3.0 x8
JPCIE3 Slot 3
PCIE 3.0 x8
JPCIE2 Slot 2
PCIE 3.0 x16
JPCIE1 Slot 1
PCIE 3.0 x8
LANE Reversal
ASMedia Switch
ASM1480
EXT CONN
JNCSI1
NC-SI(RMII)
UL1
盘阵
桥接
芯片
PCH
SATA Gen3
sSATA Gen3
I-SATA-0~3
IPASS CONN
I-SATA-4~7
IPASS CONN
S-SATA0
S-SATA1
SATA
硬盘、
固态盘
32MB BIOS
SPI FLASH
MUX
BMC SPI
PCH SPI
32MB BMC
SPI FLASH
BMC
AST2500
DDR4
VGA
HWM
TPM Header
PHY
RTL8211E
IPMI LAN
RJ45
COM1
COM2
串并口、视频接口
声卡等
USB2.0
USB3.0
ESPI
port 4,5(USB2.0)
port 1,2(USB3.0)
port 11,12(USB2.0)
port 3,4(USB3.0)
port 2,3(USB2.0)
HDR 2x10
port 8,9(USB2.0)
HDR 2x5
port 5(USB3.0)
port 10(USB2.0)
REAR USB0/1
REAR USB4/5 (3.0)
USB6/7 (3.0)
USB2/3
TYPE A USB8 (3.0)
各种USB设备（键盘、鼠标）

总线标准

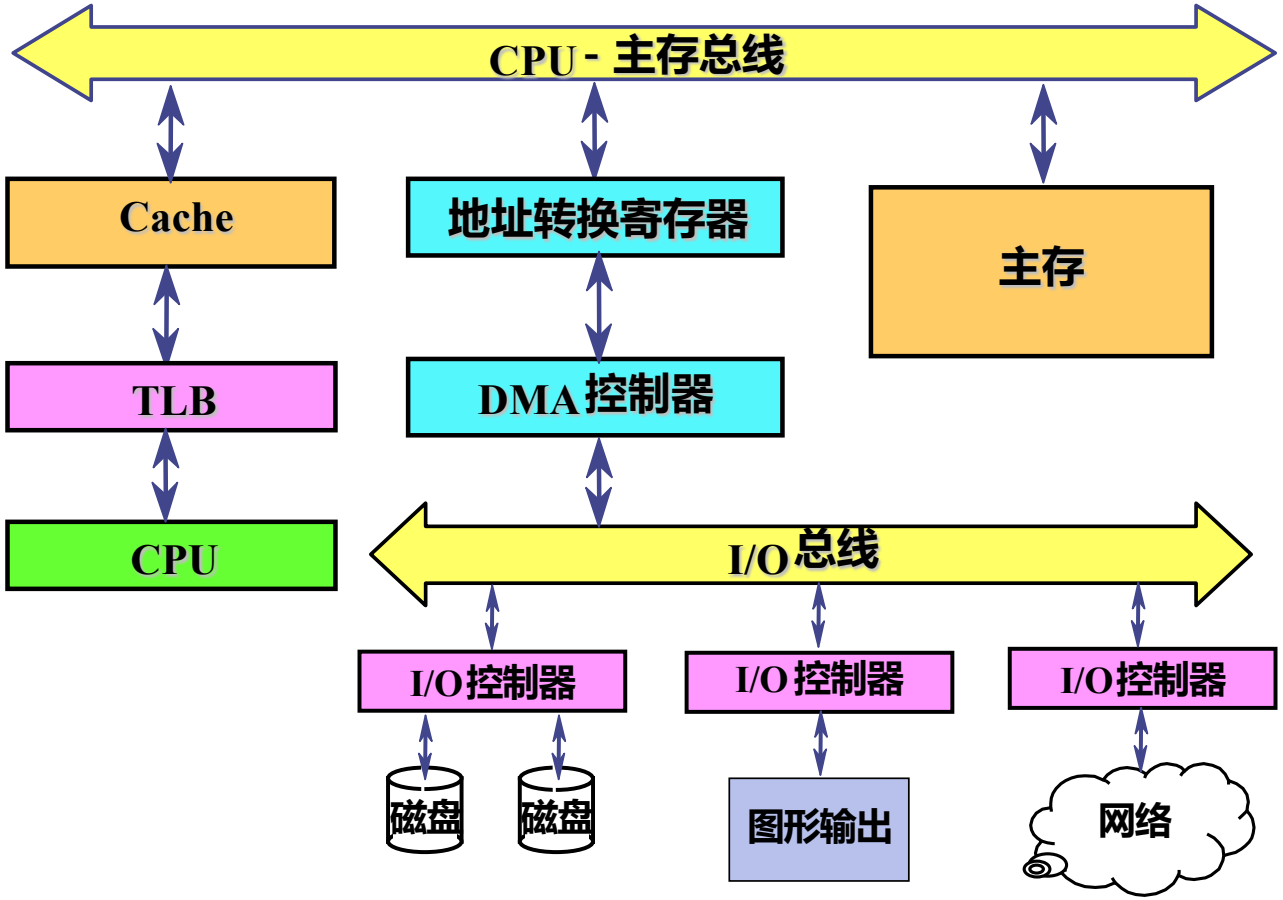
Characteristic	Firewire (1394)	USB 2.0	PCI Express	Serial ATA	Serial Attached SCSI
Intended use	External	External	Internal	Internal	External
Devices per channel	63	127	1	1	4
Basic data width (signals)	4	2	2 per lane	4	4
Theoretical peak bandwidth	50 MB/sec (Firewire 400) or 100 MB/sec (Firewire 800)	0.2 MB/sec (low speed), 1.5 MB/sec (full speed), or 60 MB/sec (high speed)	250 MB/sec per lane (1x); PCIe cards come as 1x, 2x, 4x, 8x, 16x, or 32x	300 MB/sec	300 MB/sec
Hot pluggable	Yes	Yes	Depends on form factor	Yes	Yes
Maximum bus length (copper wire)	4.5 meters	5 meters	0.5 meters	1 meter	8 meters
Standard name	IEEE 1394, 1394b	USB Implementors Forum	PCI-SIG	SATA-IO	T10 committee

FIGURE 6.8 Key characteristics of five dominant I/O standards. The intended use column indicates whether it is designed to be used with cables external to the computer or just inside the computer with short cables or wire on printed circuit boards. PCIe can support simultaneous reads and writes, so some publications double the bandwidth per lane assuming a 50/50 split of read versus write bandwidth.

PCI-Express 接口标准

版本	数据传输带宽	单向单通道带宽	双向16通道带宽	原始传输率	发表日期
1.0	2Gb/s	250MB/s	8GB/s	2.5GT/s	2002年7月22日
1.0a	2Gb/s	250MB/s	8GB/s	2.5GT/s	2003年4月15日
1.1	2Gb/s	250MB/s	8GB/s	2.5GT/s	2005年3月28日
2.0	4Gb/s	500MB/s	16GB/s	5.0GT/s	2006年12月20日
2.1	4Gb/s	500MB/s	16GB/s	5.0GT/s	2009年3月4日
3.0	8Gb/s	1GB/s	32GB/s	8.0GT/s	2010年11月10日
4.0	16Gb/s	2GB/s	64GB/s	16.0GT/s	2014年-2015年[3]

虚拟DMA的I/O连接


CPU - 主存总线
Cache
地址转换寄存器
主存
TLB
DMA控制器
CPU
I/O总线
I/O控制器
I/O控制器
I/O控制器
磁盘
磁盘
图形输出
网络

I/O和Cache数据一致性

1. Cache会使一个数据出现两个副本

   一个在Cache中，另一个在主存中。

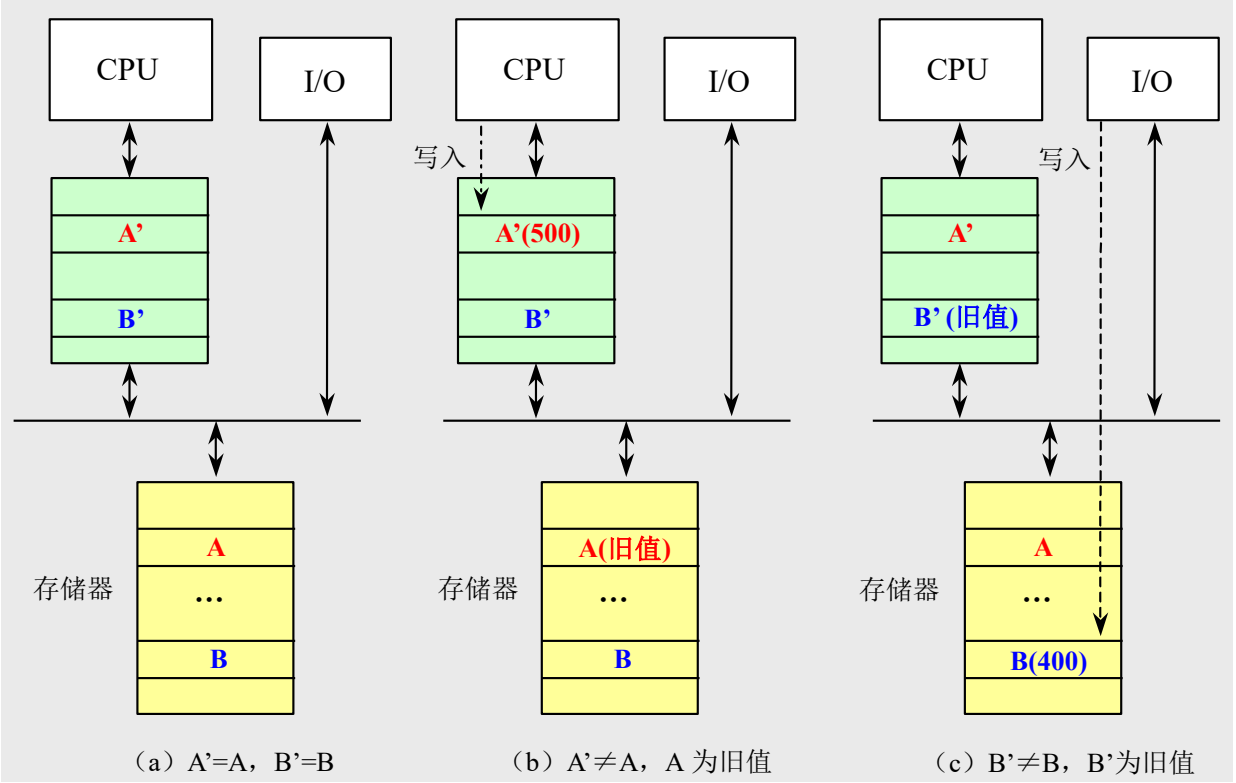
2. I/O设备可以修改存储器中的内容

   - 把I/O连接到存储器上

     会出现以下情况：

     - CPU修改了Cache的内容后，由于存储器的内容跟不上Cache内容的变化，I/O系统进行输出操作时所看到的数据是旧值。（写直达Cache没有这样的问题）
     - I/O系统进行输入操作后，存储器的内容发生了变化，但CPU在Cache中所看到的内容依然是旧值。

举例：假设Cache采用写回法，并且A’是A的副本，B’是B的副本。



解决内容一致性问题的方法

（不管Cache是采用写直达法还是写回法）

- 软件的方法
  - 设法保证I/O缓冲器中的所有各块都不在Cache中。
  - 具体做法有两种
    - 把I/O缓冲器的页面设置为不可进入Cache的，在进行输入操作时，操作系统总是把输入的数据放到该页面上。
    - 在进行输入操作之前，操作系统先把Cache中与I/O缓冲器相关的数据“赶出”Cache，即把相应的数据块设置为“无效”状态。

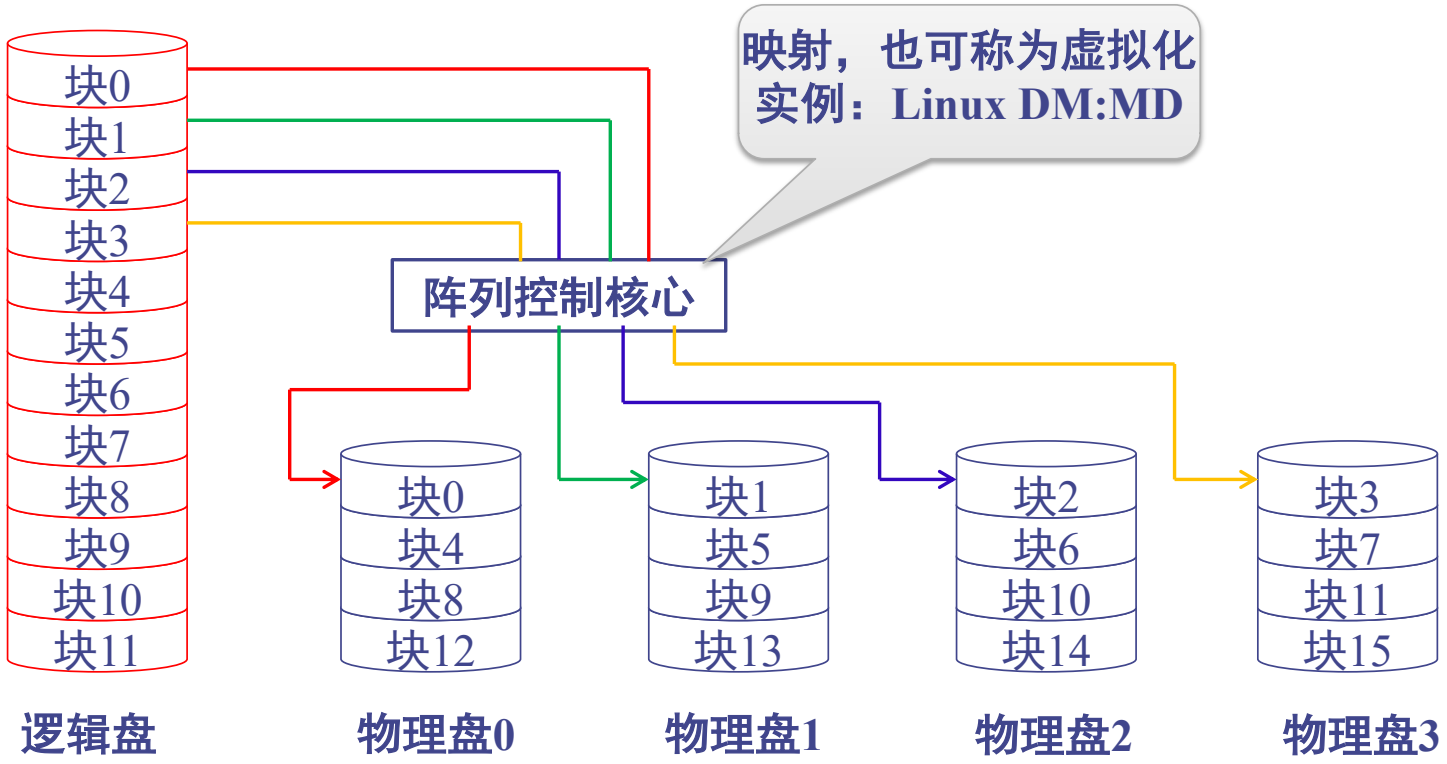
- 硬件的方法
  - 在进行输入操作时，检查相应的I/O地址（I/O缓冲器中的单元）是否在Cache中（即是否有数据副本）。
  - 如果发现I/O地址在Cache中有匹配的项，就把相应的Cache块设置为“无效”。

8.3 磁盘阵列RAID

磁盘阵列（Redundant Array of Inexpensive /Independent Disks）：
使用多个磁盘（包括驱动器）的组合来代替一个大容量的磁盘

- 多个磁盘并行工作；以条带为单位把数据均匀地分布到多个磁盘上
- 条带存放可以使多个数据读/写请求并行处理，从而提高总的I/O性能
  - 多个独立的请求可以由多个盘来并行地处理
    减少了I/O请求的排队等待时间
  - 如果一个请求访问多个块，就可以由多个盘合作来并行处理
    提高了单个请求的数据传输率

RAID逻辑上构成一个大盘



数据交叉存放的粒度

- 细粒度磁盘阵列是在概念上把数据分割成相对较小的单位交叉存放。
  - 优点：所有I/O请求都能够获得很高的数据传输率。
  - 缺点：在任何时间，都只有一个逻辑上的I/O在处理当中，而且所有的磁盘都会因为为每个请求进行定位而浪费时间
- 粗粒度磁盘阵列是把数据以相对较大的单位交叉存放
  - 多个较小规模的请求可以同时得到处理
  - 对于较大规模的请求又能获得较高的传输率

冗余数据的计算方法以及存放方式

- 如何计算冗余信息?
  - 大多都是采用奇偶校验码
  - 也有采用汉明码（Hamming code）或Reed-Solomon码
- 如何把冗余信息分布到磁盘阵列中的各个盘?

  有两种方法：
  - 把冗余信息集中存放在少数的几个盘中
  - 把冗余信息均匀地存放到所有的盘中

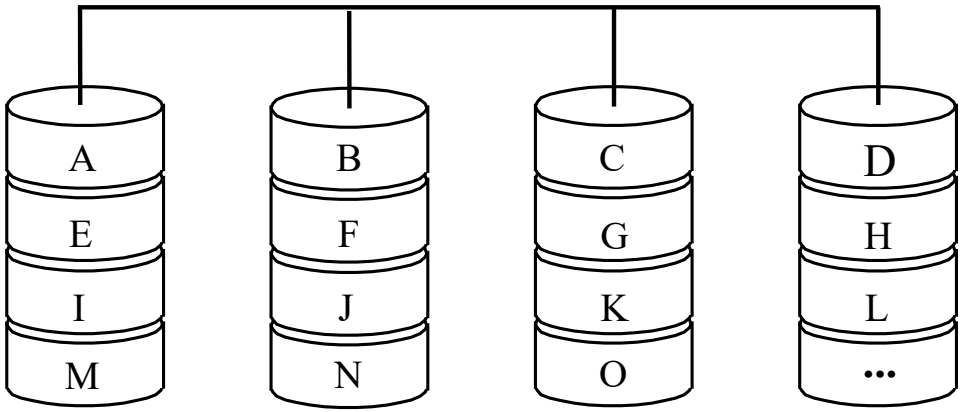
    （能避免出现热点问题）

RAID 级别（Level）

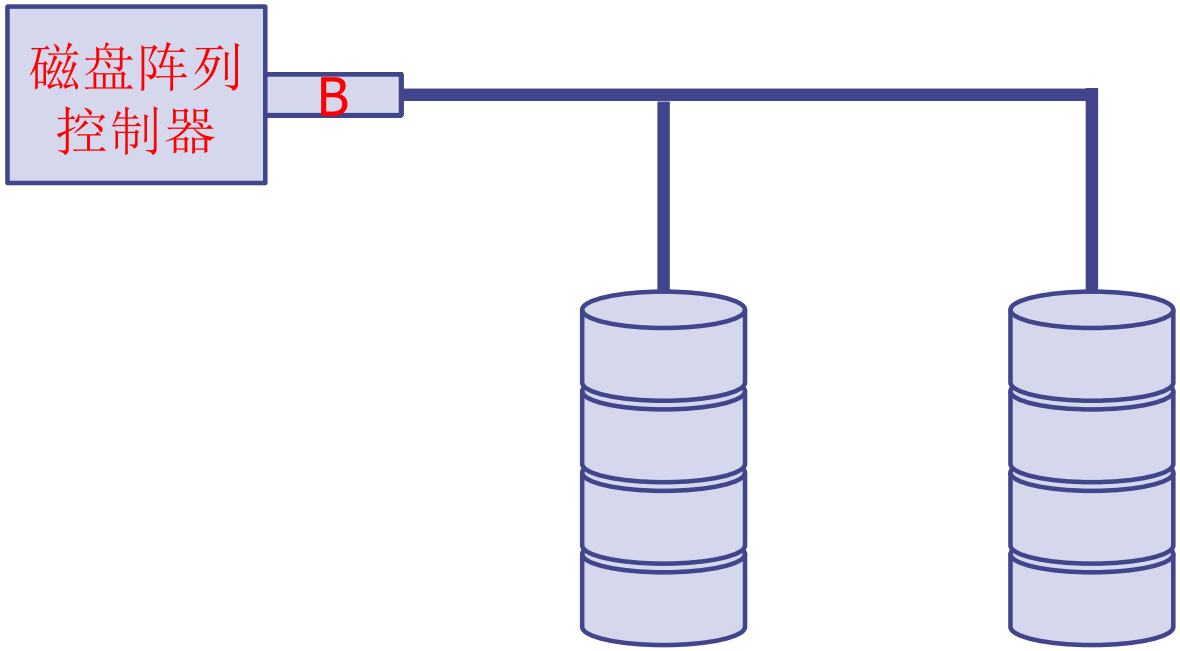
- RAID 0：数据分割，无容错能力
- RAID 1：镜像(双拷贝)
- RAID 2：海明码，不具有商业生命力
- RAID 3：并行，位交叉，单校验盘
- RAID 4：并行，块交叉，单校验盘
- RAID 5：独立，循环校验盘
- RAID 6：容双盘错，块交叉，近年被广泛重视
- RAID 10: RAID1 + RAID 0

1、RAID0

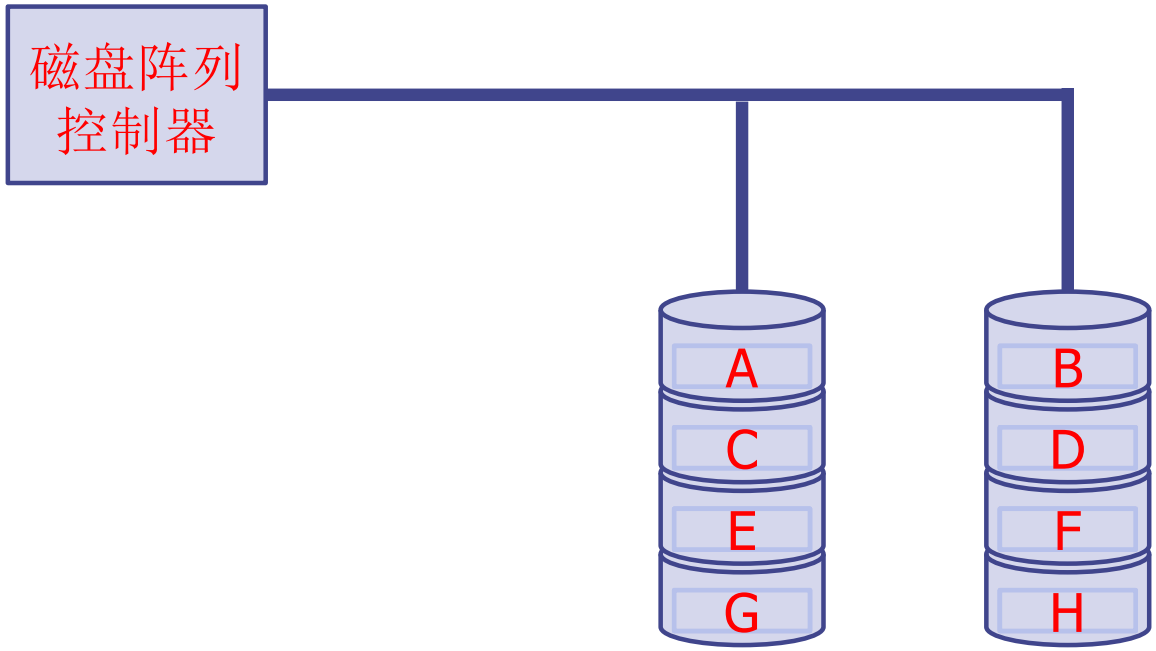
- 非冗余磁盘阵列，无冗余信息
- 把数据切分成条带，以条带为单位交叉地分布存放到多个磁盘中
- 优点：
  - 空间利用率高；读写快；设计简单
- 缺点：
  - 没有容错能力



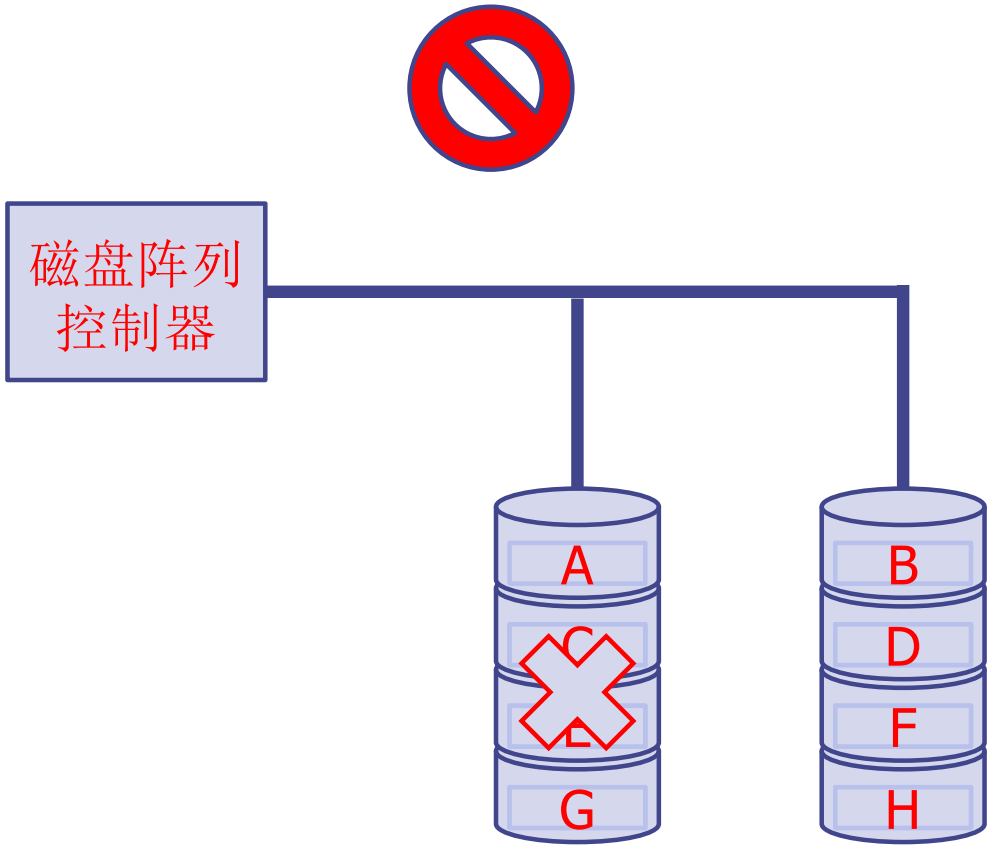
RAID 0 写



RAID 0 读

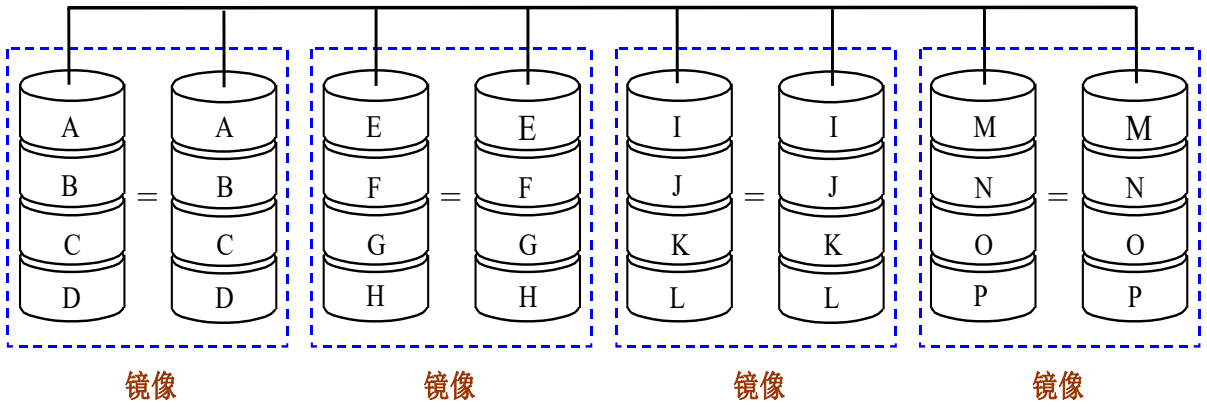


RAID 0 无法恢复

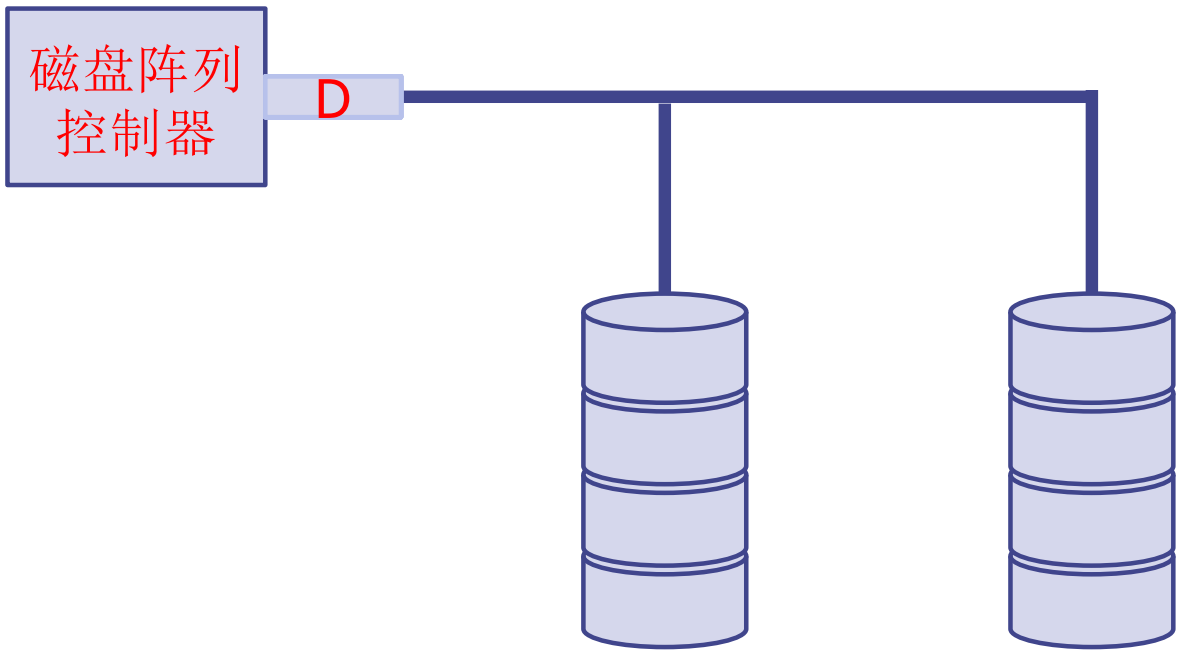


2、RAID1

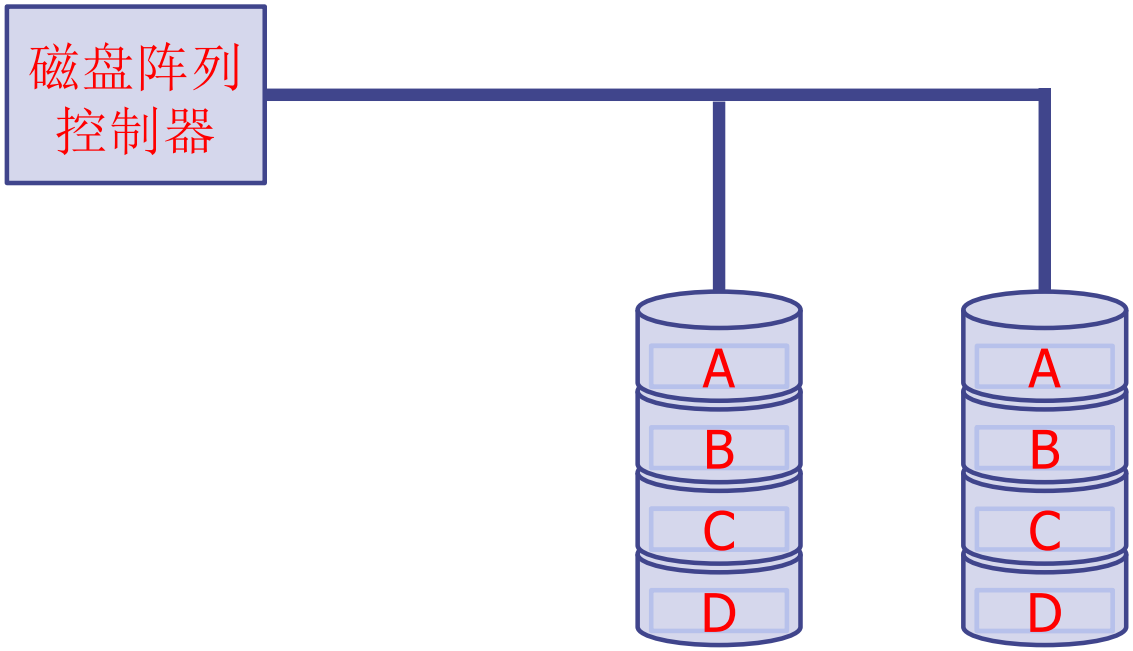
- **镜像磁盘**，对所有的磁盘数据提供一份冗余的备份
  - 每当把数据写入磁盘时，将该数据也写入其镜像盘。在系统中所有的数据都有两份
- **RAID1特点**
  - 能实现**快速**的读取操作;对于写入操作，镜像的两个磁盘都要写入。但可并行进行，而且不需要计算校验信息，所以其速度比级别更高的RAID都快。
  - **可靠性**很高，数据的恢复很简单。但实现成本最高



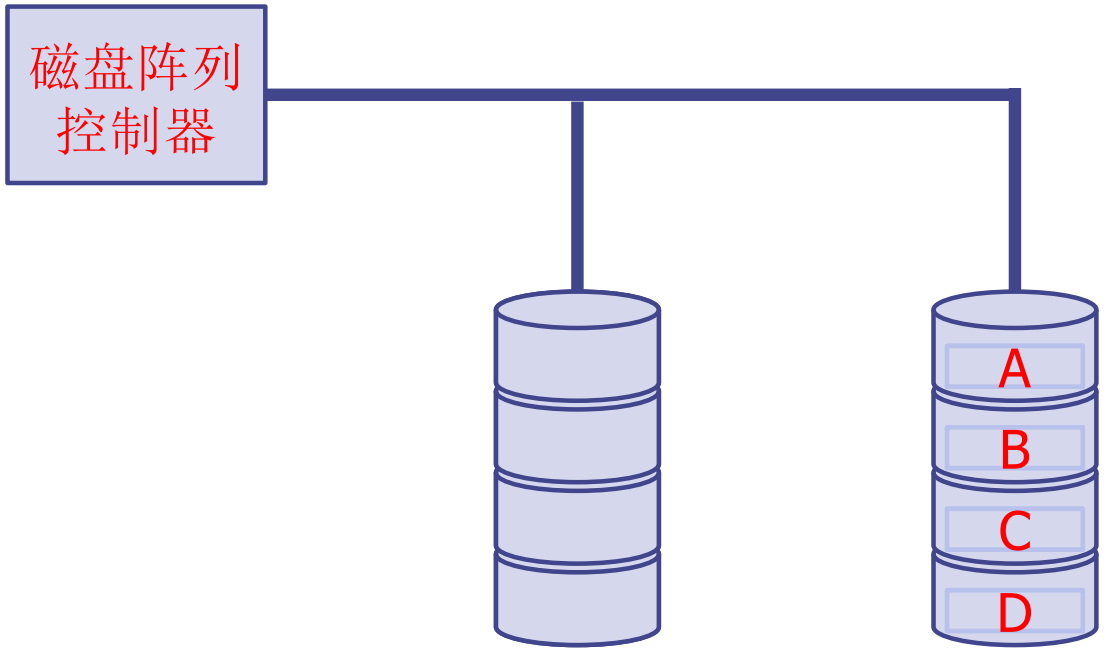
RAID 1 写



RAID 1 读



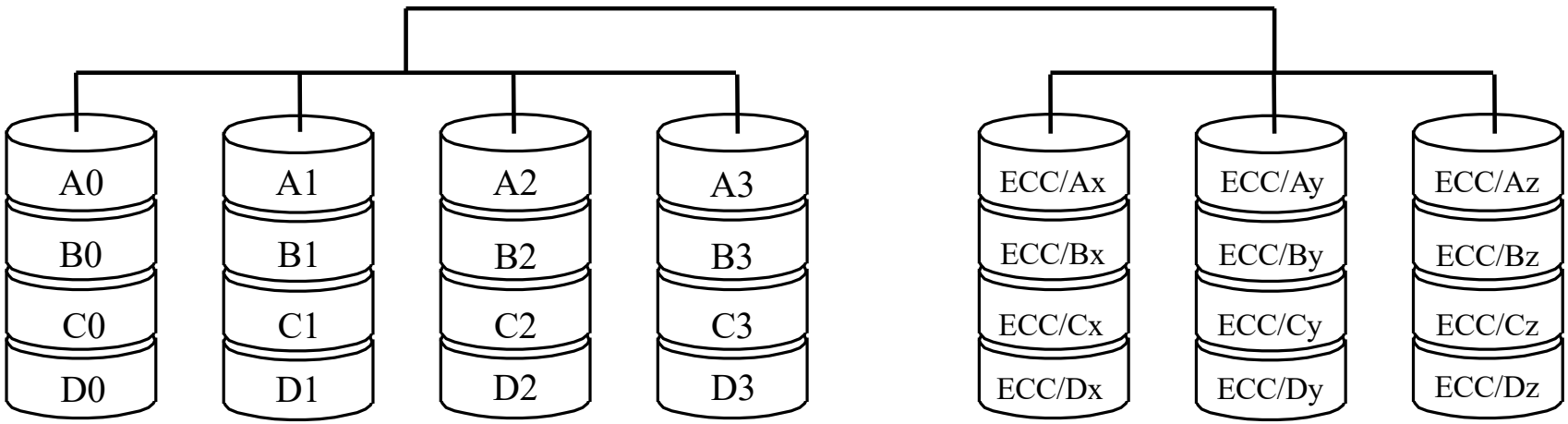
RAID 1 恢复



3、RAID 2

存储器式的磁盘阵列（按汉明纠错码的思路构建）

- 含4个数据盘的RAID2



字 A＝A0～A3　字 B＝B0～B3
字 C＝C0～C3　字 D＝D0～D3

字 A 的 ECC＝Ax～Az　字 B 的 ECC＝Bx～Bz
字 C 的 ECC＝Cx～Cz　字 D 的 ECC＝Dx～Dz

7/4 汉明纠错码

$h0 = b0 \oplus b1 \oplus b3$

$h1 = b0 \oplus b2 \oplus b3$

$h2 = b1 \oplus b2 \oplus b3$

H0(<1101>) = 0

H1(<1101>) = 1

H2(<1101>) = 0

Hamming(<1101>) = <b3, b2, b1, h2, b0, h1, h0> = <1100110>

如果有1位反转了, 例如 <1110110>

Hamming(<1111>) = <h2, h1, h0> = <111> 比较到 <010>, <101> 是错误的。

错误发生在bit 5。

4、RAID 3

- 交叉奇偶校验磁盘阵列
  - 写数据时：为每行数据形成奇偶校验位并写入校验盘
  - 读出数据时：如果控制器发现某个磁盘出故障，就可以根据故障盘以外的所有其他盘中的正确信息恢复故障盘中的数据。（通过异或运算实现）
- 细粒度的磁盘阵列，即采用的条带宽度较小。

  （可以是1个字节或1位）
  - 多个磁盘的并行访问
  - 不能同时进行多个I/O请求的处理 。
- 只需要一个校验盘，校验空间开销比较小。

位奇偶校验

校验Parity ： 计算位为1的个数,如果是偶数，则为0；否则为1

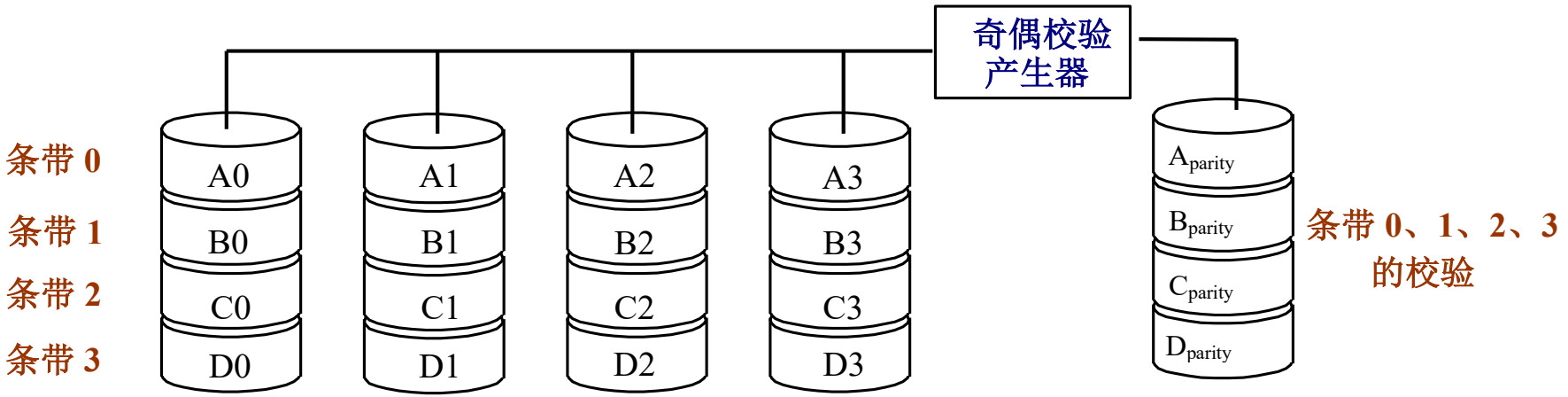
Parity(<b3, b2, b1, b0 >) = P0 = b0 ⊕ b1 ⊕ b2 ⊕ b3

Parity(<b3, b2, b1, b0, p0>) = 0 所有位都是正确的

Parity(0110) = 0, Parity(01100) = 0

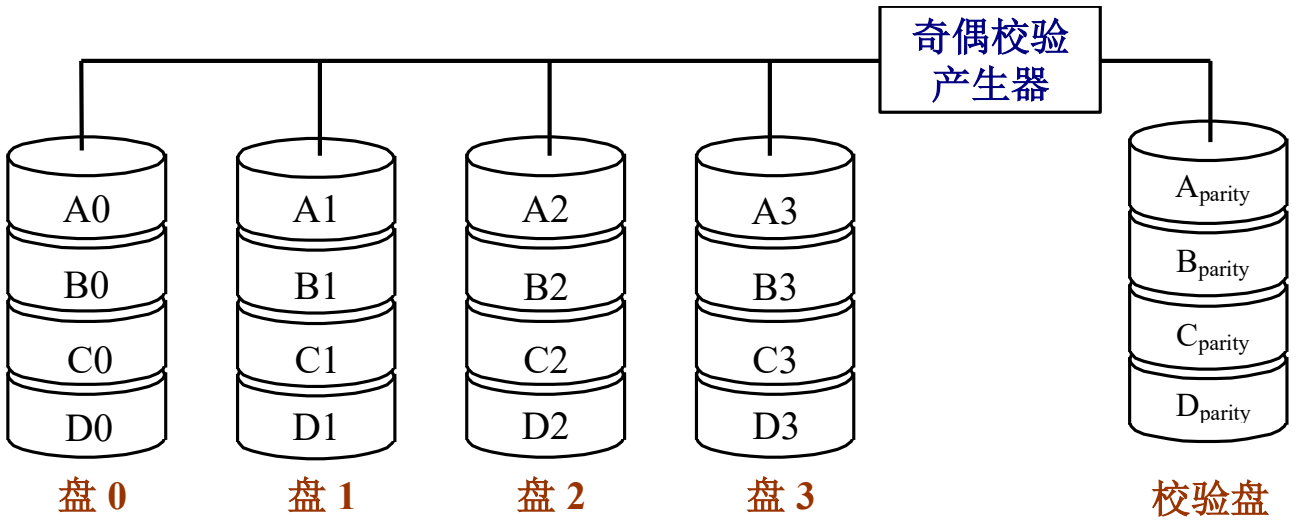
Parity(11100) = 1 => ERROR!

奇偶校验能够检查到1位错误，但是不确定哪一位错误



5、RAID 4

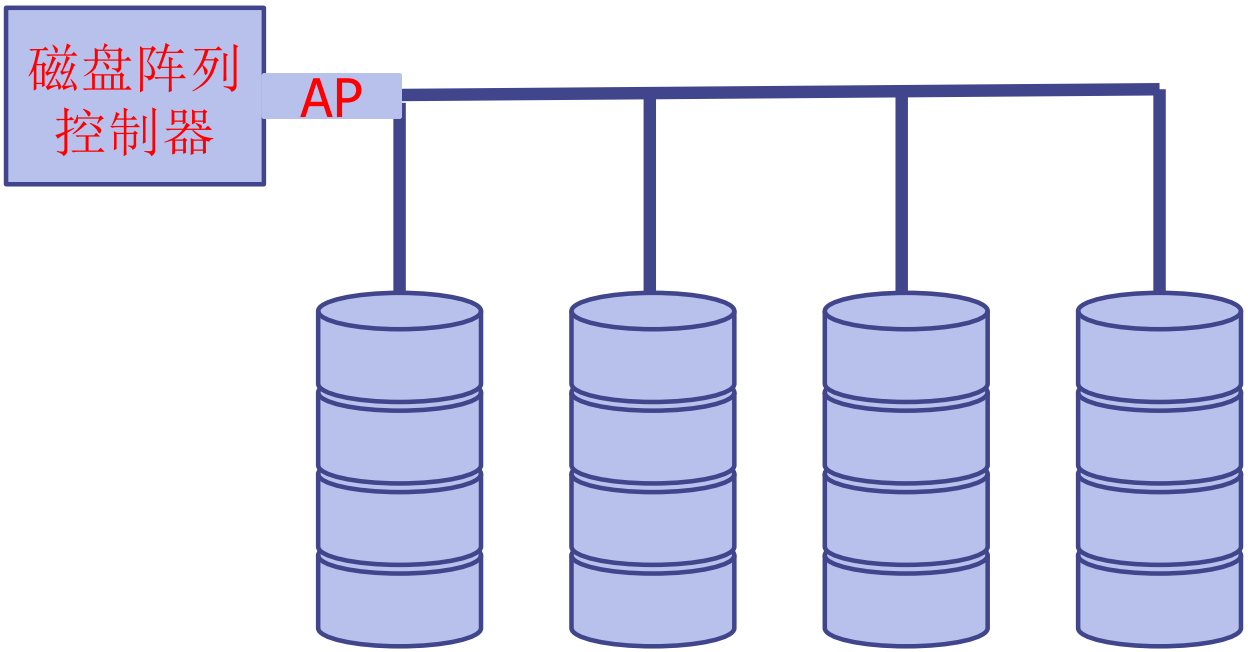
- 块交叉奇偶校验磁盘阵列
- 采用比较大的条带，以块为单位进行交叉存放和计算奇偶校验。
  - 实现目标：能同时处理多个小规模访问请求



RAID 4 读写特点

- 读取操作
  - 每次只需访问数据所在的磁盘。
  - 仅在该磁盘出现故障时，才会去读校验盘，并进行数据的重建。
- 写入操作
  - 写一块数据需要2次磁盘读和2次磁盘写操作。

RAID 4 写



RAID 4 写

Block	Data
A1	11110000
A2	11001100
A3	10101010
AP	

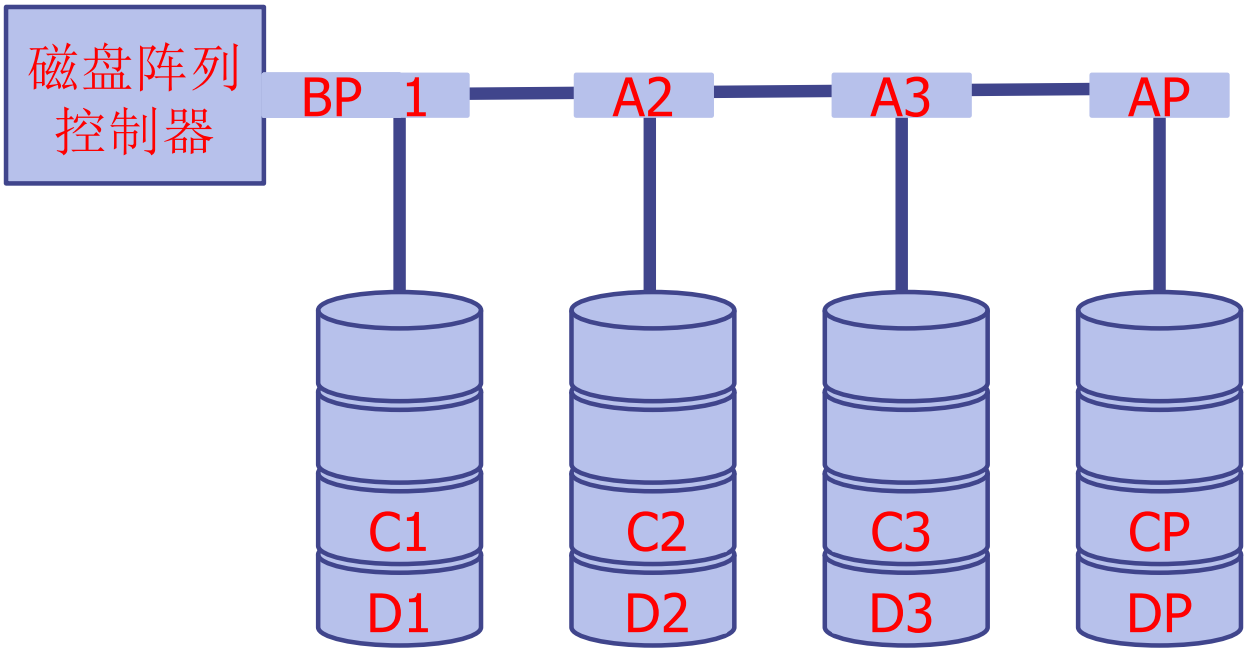
RAID 4写

Block	Data
A1	11110000
A2	11001100
A3	10101010
AP	1

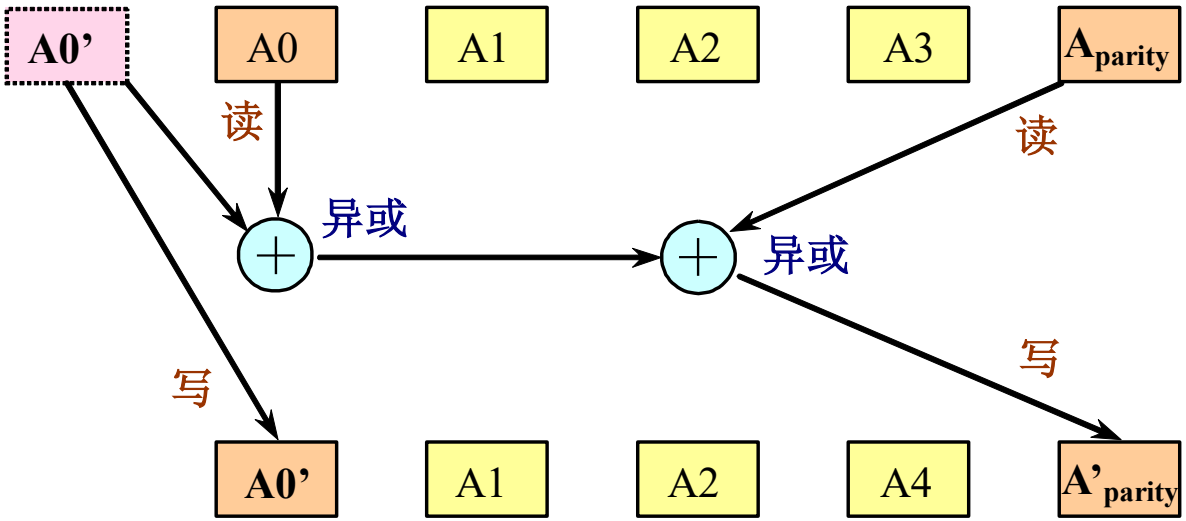
RAID 4写

Block	Data
A1	11110000
A2	11001100
A3	10101010
AP	10010110

RAID 4 写



RAID 4 更新过程



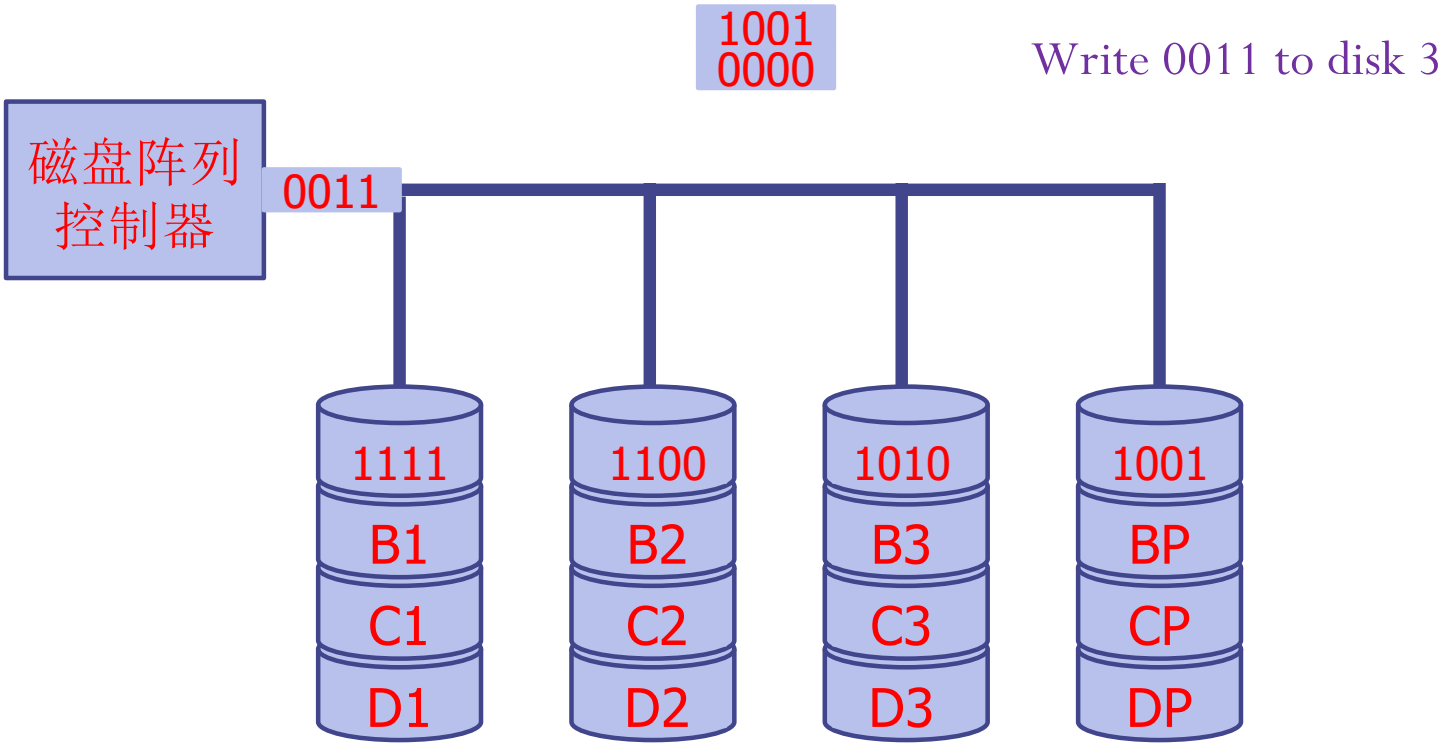
(1) 读数据$A0_{旧值}$和校验码$P_{旧值}$

(2) 计算新校验码

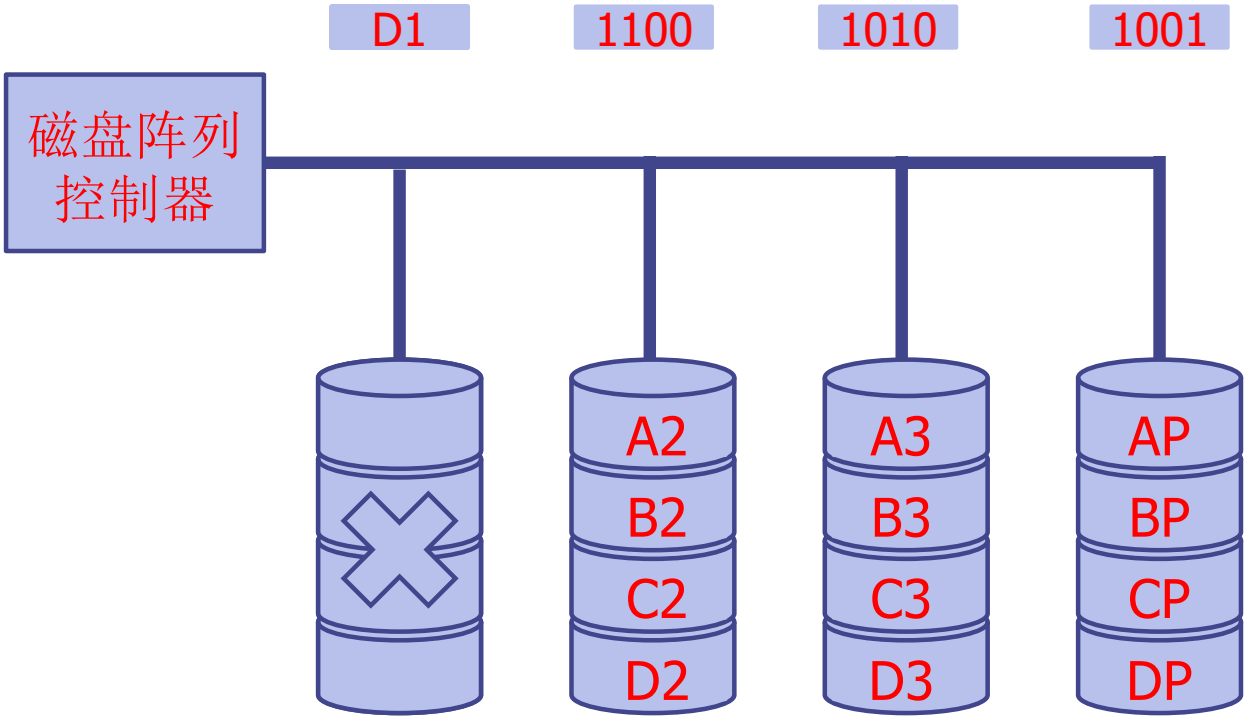
$$P_{新值} = A0_{旧值} \oplus P_{旧值} \oplus A0'_{新值}$$

(3) 写数据$A0'_{新值}$ 和校验码$P_{新值}$

RAID 4 更新过程

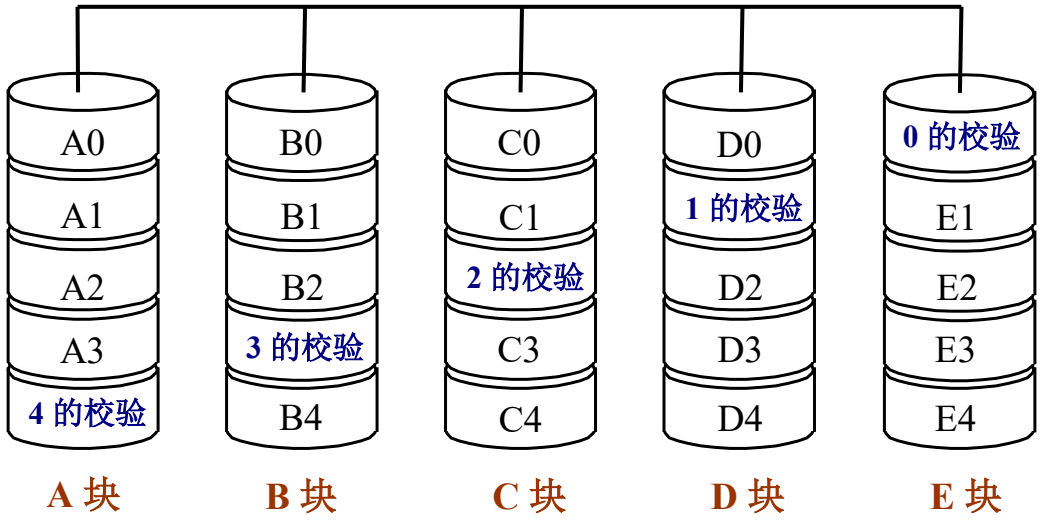

1001
0000
Write 0011 to disk 3
磁盘阵列
控制器
0011
1111
B1
C1
D1
1100
B2
C2
D2
1010
B3
C3
D3
1001
BP
CP
DP

RAID 4 恢复

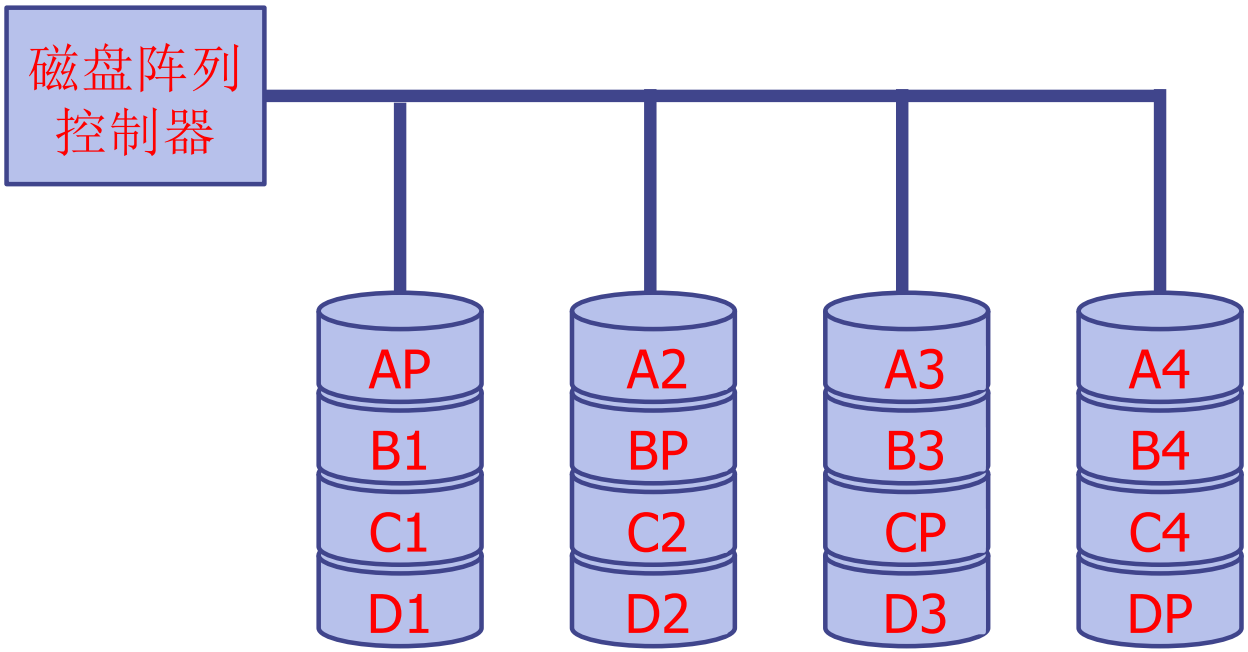


6、RAID 5

- 块交叉分布奇偶校验磁盘阵列
- 数据以块交叉的方式存于各盘，无专用冗余盘，奇偶校验信息均匀分布在所有磁盘上
- 相对于RAID4，就是校验块分散在所有盘上

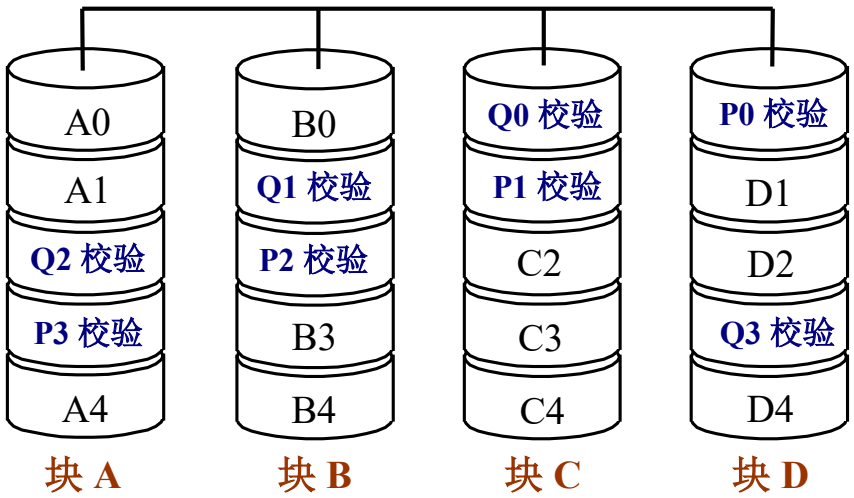


RAID 5



7、RAID 6

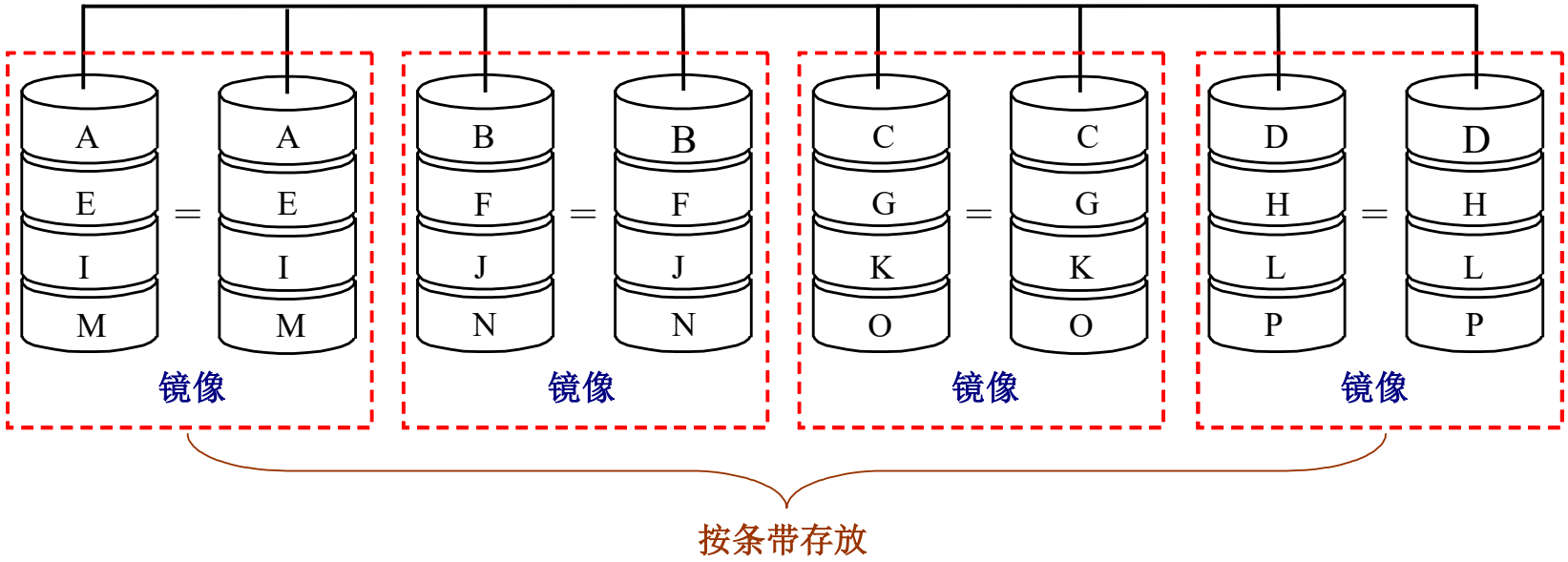
- P＋Q双校验磁盘阵列
- 特点
  - 校验空间开销是RAID5的两倍
  - 容忍两个磁盘出错



8、RAID10与RAID01

RAID10又称为RAID1+0

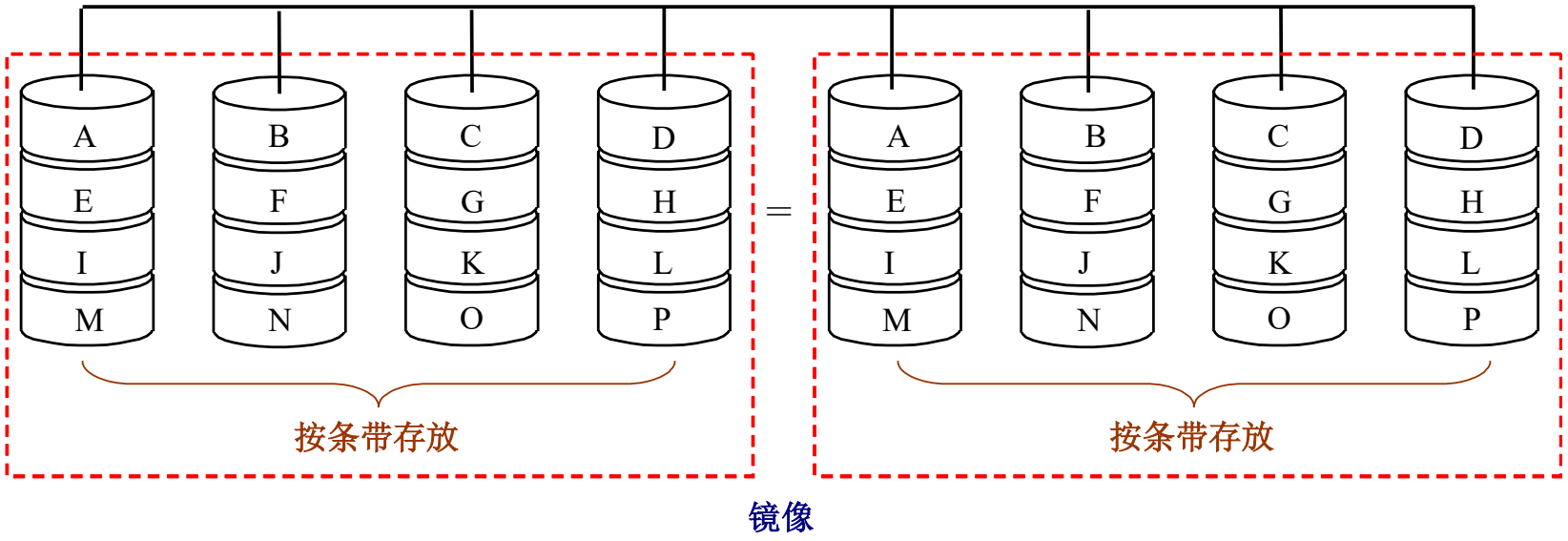
先进行镜像（RAID1），再进行条带存放（RAID0）



RAID10与RAID01

RAID01又称为RAID0+1

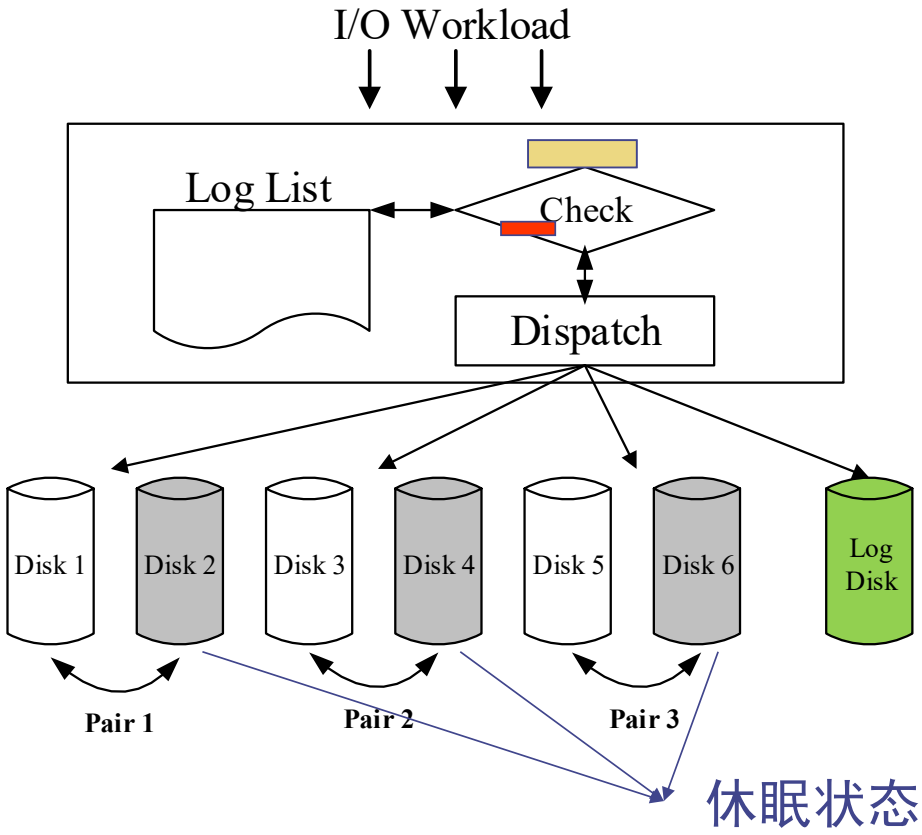
先进行条带存放（RAID0），再进行镜像（RAID1）



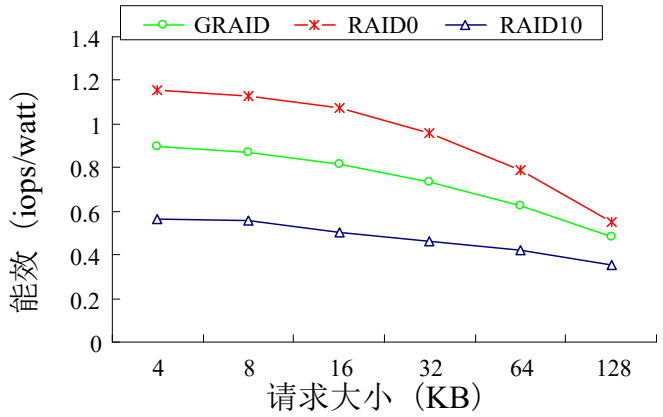
降低RAID10能耗的数据布局

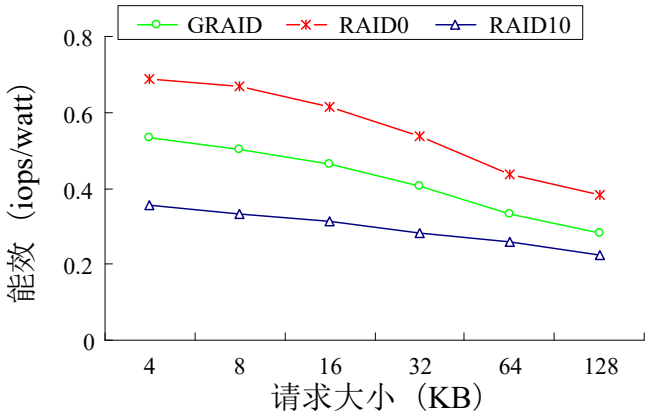
- RAID0采用镜像双冗余结构造成其能耗大
- 能耗固然重要，但数据的价值更为重要，如何在能耗与可靠性及性能之间找到平衡？

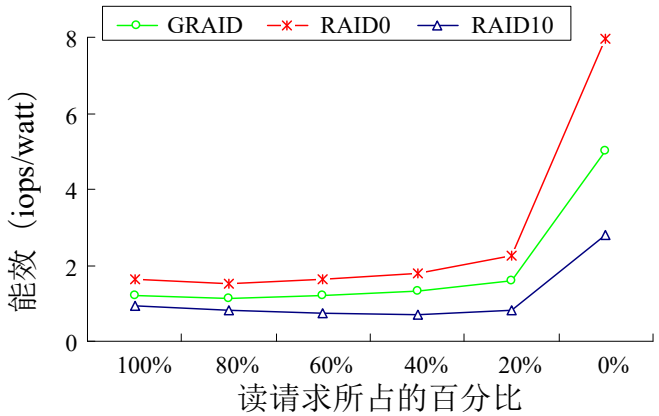
降低RAID10能耗的数据布局



降低RAID10能耗的数据布局



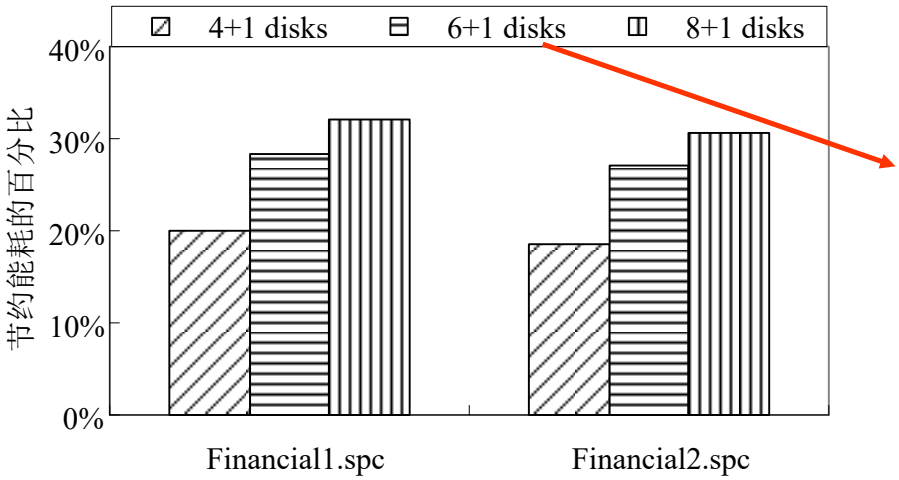


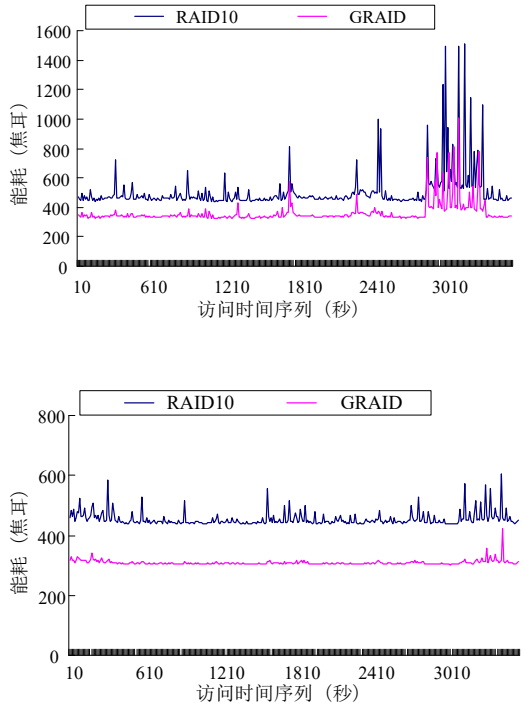


GRAID 能效高于RAID10；

低于RAID0(数据只写一半数目的硬盘)

降低RAID10能耗的数据布局

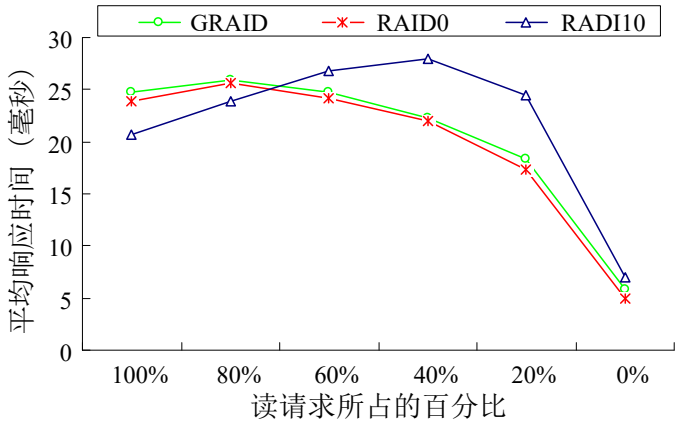


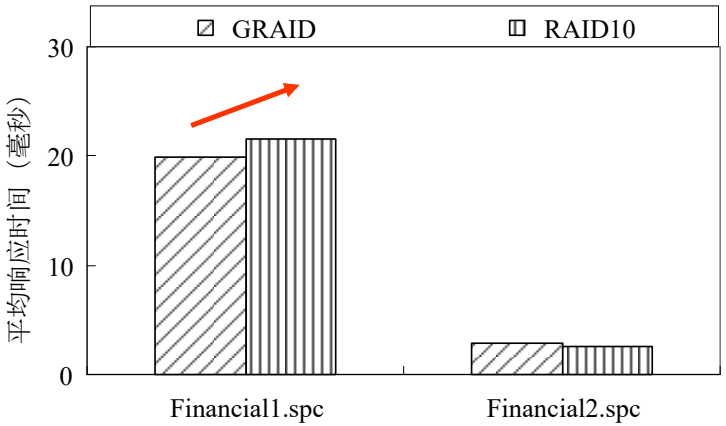


不同负载下不同GRAID 能耗降低20%~35%

GRAID 能耗低于RAID10

降低RAID10能耗的数据布局



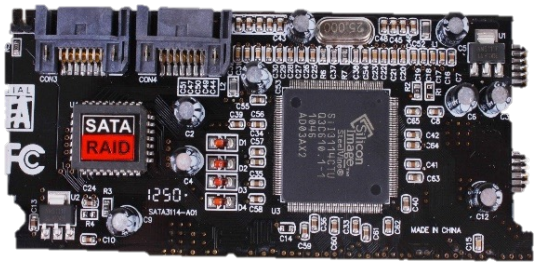


GRAID性能与RAID0相当，当读请求占比低时/写密集型应用，GRAID性能高于RAID10

RAID的实现方式

实现盘阵列的方式主要有三种：

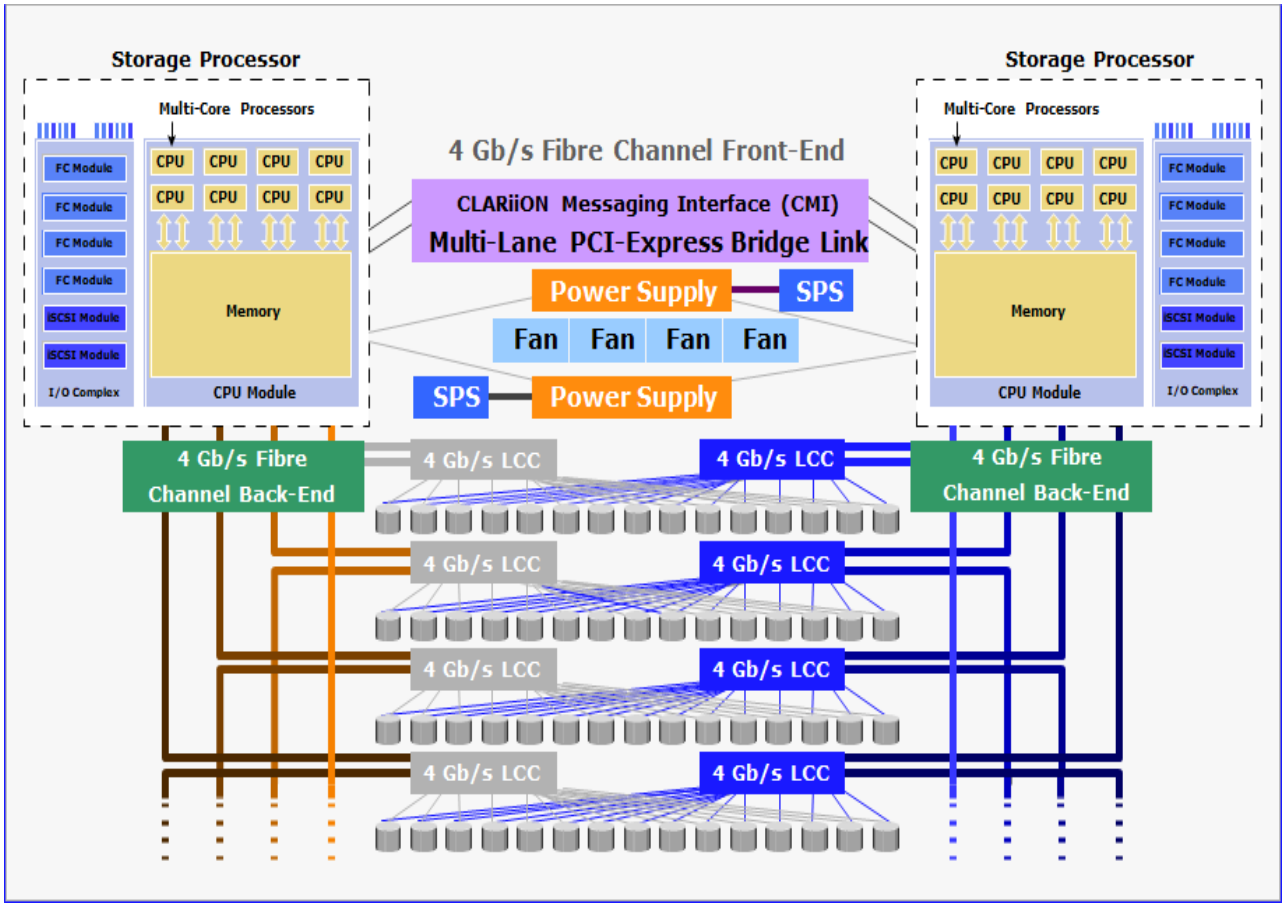
- **软件方式：**阵列管理软件由主机来实现（Linux MD）
  - 优点：成本低
  - 缺点：占用主机时间。
- **阵列卡方式：**把RAID管理软件固化在I/O控制卡上，从而可不占用主机时间，一般用于服务器和PC机



- **专用磁盘阵列：**一种基于通用接口总线的开放式平台，能够被多个服务器同时存取

大型企业级RAID

EMC CLARiiON CX-4 Architecture



8.4 I/O系统的可靠性、可用性和可信性

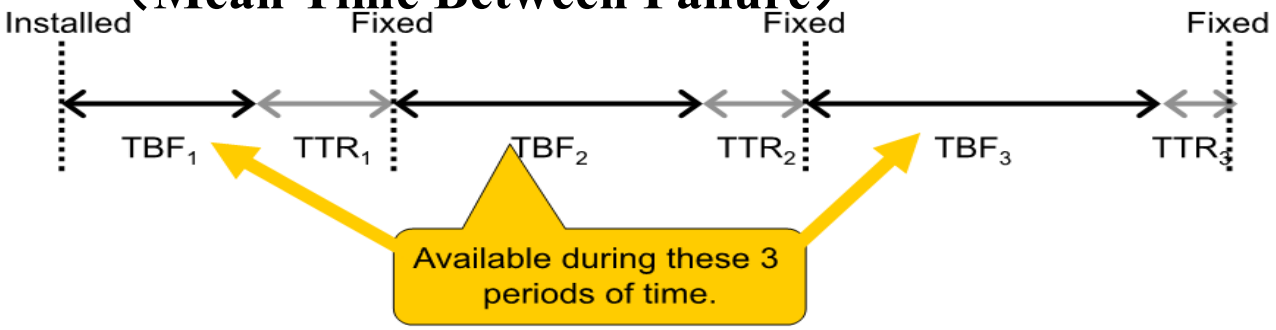
1. 外设可靠性能的参数有：
   - 可靠性（Reliability）
   - 可用性（Availability）
   - 可信性（Dependability）
2. 系统的可靠性：系统从某个初始参考点开始一直连续提供服务的能力。
   - 用平均无故障时间MTTF来衡量。
   - 系统中断服务的时间用平均修复时间MTTR来衡量
   - MTTF的倒数就是系统的失效率。
   - 如果系统中每个模块的生存期服从指数分布，则系统整体的失效率是各部件的失效率之和。

8.4 I/O系统的可靠性、可用性和可信性

3. **系统的可用性：**系统正常工作的时间在连续两次正常服务间隔时间中所占的比率。

$$\text{可用性} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}$$

- MTTF+MTTR：平均失效间隔时间MTBF（Mean Time Between Failure）



4. **系统的可信性：**服务的质量。即在多大程度上可以合理地认为服务是可靠的。（不可以度量）

8.4 I/O系统的可靠性、可用性和可信性

例8.1 假设磁盘子系统的组成部件和它们的MTTF如下：

（1）磁盘子系统由10个磁盘构成，每个磁盘的MTTF为1000000小时；

（2）1个SCSI控制器，其MTTF为500000小时；

（3）1个不间断电源，其MTTF为200000小时；

（4）1个风扇，其MTTF为200000小时；

（5）1根SCSI连线，其MTTF为1000000小时。

假定每个部件的生存期服从指数分布，同时假定各部件的故障是相互独立的，求整个系统的MTTF。

8.4 I/O系统的可靠性、可用性和可信性

解 整个系统的失效率为：

$$\text{系统失效率}=10\times\frac{1}{1000000}+\frac{1}{500000}+\frac{1}{200000}+\frac{1}{200000}+\frac{1}{1000000}=\frac{23}{1000000}$$

系统的MTTF为系统失效率的倒数，即：

$$\text{MTTF}=\frac{1000000}{23}=43500\text{小时}$$

即将近5年。

谢 谢！

冯 丹
Email:dfeng@hust.edu.cn

华中科技大学
HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

可靠性模型

可靠度：R (t)，不可靠度：F (t)，
关系式：R (t) + F (t) = 1。

1. 串联系统

- **系统由n个部件串联而成，其中任何一个部件失效就引起系统的失效，**
- **故串联系统寿命等于其中最先失效部件的寿命。**

若每个单元的可靠度分别为$R_1, R_2 ..., R_n$,且每个单元相互独立，则系统可靠度为：

$$\mathrm{R_s} = \prod_{i=1}^{n} R_i$$

可靠性模型

2. 并联系统

- 系统由n个部件并联而成，当着n个部件都失效时才引起系统失效。
- 故并联系统寿命等于其中最后失效部件的寿命。

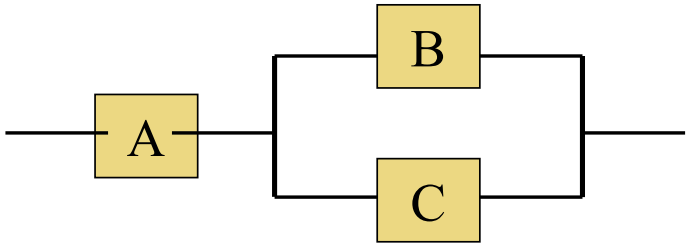
若每个单元的可靠度分别为$R_1, R_2 ..., R_n$,且每个单元相互独立，则系统可靠度为：

$$\mathrm{R_s} = 1 - \prod_{i=1}^{n} (1 - R_i)$$

可靠性模型

3.举例

设一个设备由A、B、C 3个相互独立的部件组成，其可靠性框图见下图，这3个部件的可靠度分别为R_A=0.9 R_B=0.8 R_C=0.6，现不考虑其他因素，求系统的可靠度R。



R_{BC}=1-（1-R_B）（1-R_C）

=1-0.2*0.4=0.92

R=$R_A R_{BC}$=0.9*0.92=0.828

可靠性模型

可靠度：R (t)，不可靠度：F (t)，关系式：R (t) + F (t) = 1。

在各单元相同情况下，用n代表串联倍数，m代表并联倍数。

①串并联系统：先串联、后并联，可靠度

$$R(t) = 1-[1-R_i^n(t)]^m$$

②并串联系统：先并联、后串联，可靠度

$$R(t) = [1-(1-R_i(t))^m]^n$$

公式用途：RAID 0+1是串并联系统，RAID 1+0是并串联系统

Let's look at the impact on the CPU of ***reading a disk page directly into the cache***. Make the following assumptions:

① .Each page is 16 KB, and the cache-block size is 64 bytes.

② .The addresses corresponding to the new page are not in the cache.

③ .The CPU will not access any of the data in the new page.

④ .95% of the blocks that were displaced from the cache will be read in again, and each will cause a miss.

⑤ .The cache uses write back, and 50% of the blocks are dirty on average.

⑥ .The I/O system buffers a full cache block before writing to the cache (this is called a *speed-matching* buffer, matching transfer bandwidth of the I/O system and memory).

⑦ .The accesses and misses are spread uniformly to all cache blocks.

⑧ .There is no other interference between the CPU and I/O for the cache slots.

⑨ .There are 15,000 misses every 1 million clock cycles when there is *no* I/O.

⑩ .The miss penalty is 30 clock cycles, plus 30 more cycles to write the block if it was dirty.

Assuming one page is brought in every 1 million clock cycles, what is the impact on performance?

ANSWER

1. Each page fills 16,384/64 or 256 blocks.(1)
2. I/O transfers do not cause cache misses on their own because entire cache blocks are transferred. However, they do displace blocks already in the cache. (2)
3. If half of the displaced blocks are dirty, it takes 128 x 30 clock cycles to write them back to memory. (5)(10)
4. There are also misses from 95% of the blocks displaced in the cache because they are referenced later, adding another 95% x 256 (4), or 244 misses. 244 x30 cycles (10)
5. Since this data was placed into the cache from the I/O system, all these blocks are dirty and will need to be written back when replaced. Thus, the total is on average 128 x 30 + 244 x 60 more clock cycles than the original 1,000,000+ 7500 x 30 + 7500 x 60.
6. This turns into a 1% decrease in performance:

$$\frac{128\times 30+244\times 60}{1000000+7500\times 30+7500\times 60}=0.011$$