操作系统 Operation System

磁盘管理

王雷

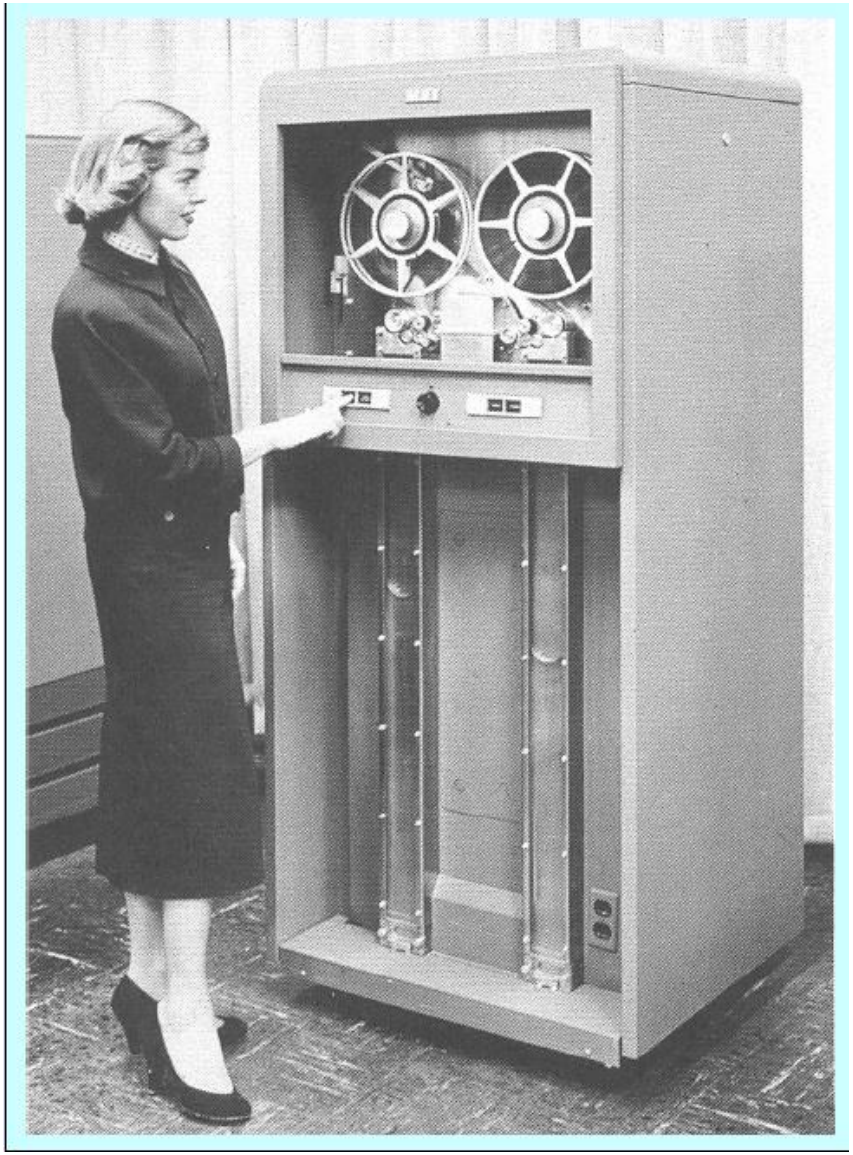
wanglei@buaa.edu.cn

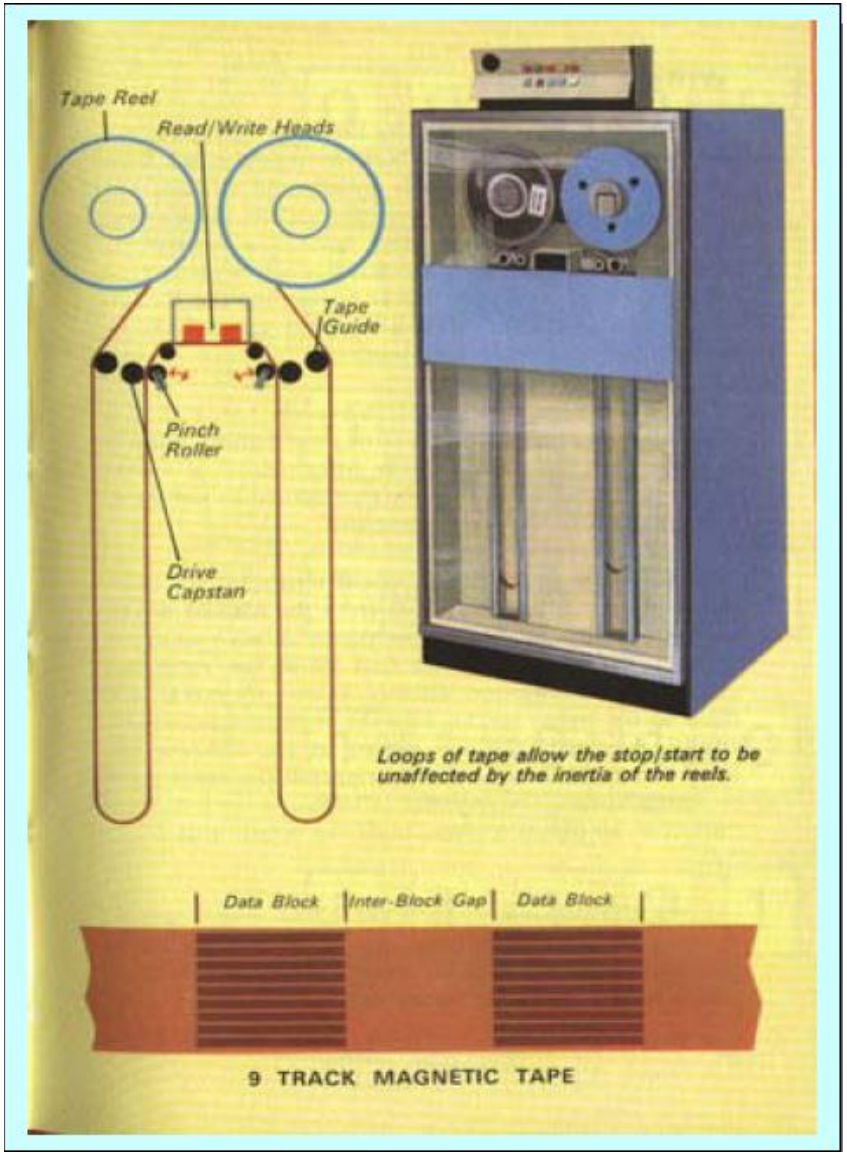
TEL：82316284

内容提要

- 磁盘的历史及工作原理
- 磁盘的组织与调度算法
- 磁盘空间的管理
- RAID
- 提高I/O速度
- 磁盘管理实例

硬盘诞生前的外存——磁带




Tape Reel
Read/Write Heads
Tape Guide
Pinch Roller
Drive Capstan
Loops of tape allow the stop/start to be unaffected by the inertia of the reels.
Data Block
Inter-Block Gap
Data Block
9 TRACK MAGNETIC TAPE

温彻斯特”盘的来历

- “温彻斯特”硬盘——IBM把硬盘称做温彻斯特（Winchester）硬盘，也称温盘。“温彻斯特”这个名字有个小小的来历；IBM3340拥有两个30MB的存储单元，而当时一种很有名的“温彻斯特来复枪”的口径和装药也恰好包含了两个数字“30”；于是这种硬盘的内部代号就被定为“温彻斯特”。

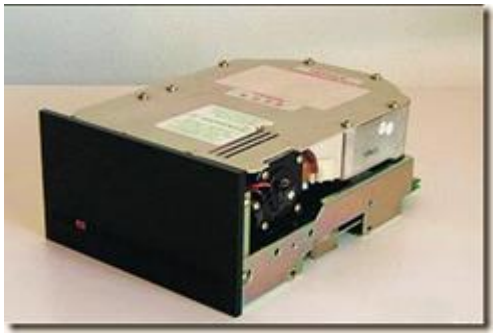
硬盘简史

世界上第一块硬盘出生在1956年，当时IBM公司制造出世界上第一块硬盘350 RAMAC (Random Access Method of Accounting and Control)，其容量为5MB，盘片直径为24英寸，盘片数为50片，磁盘总重量1吨。盘片上有一层磁性物质，被轴带着旋转，有磁头移动着存储数据，实现了随机存取。当时这种硬盘被用于银行，医学等领域。虽然350 RAMAC还不能称为严格意义上的硬盘，但它为计算机发展史掀开了新的一页。

硬盘简史

62~78年
14吋硬盘

78~80年
8吋硬盘



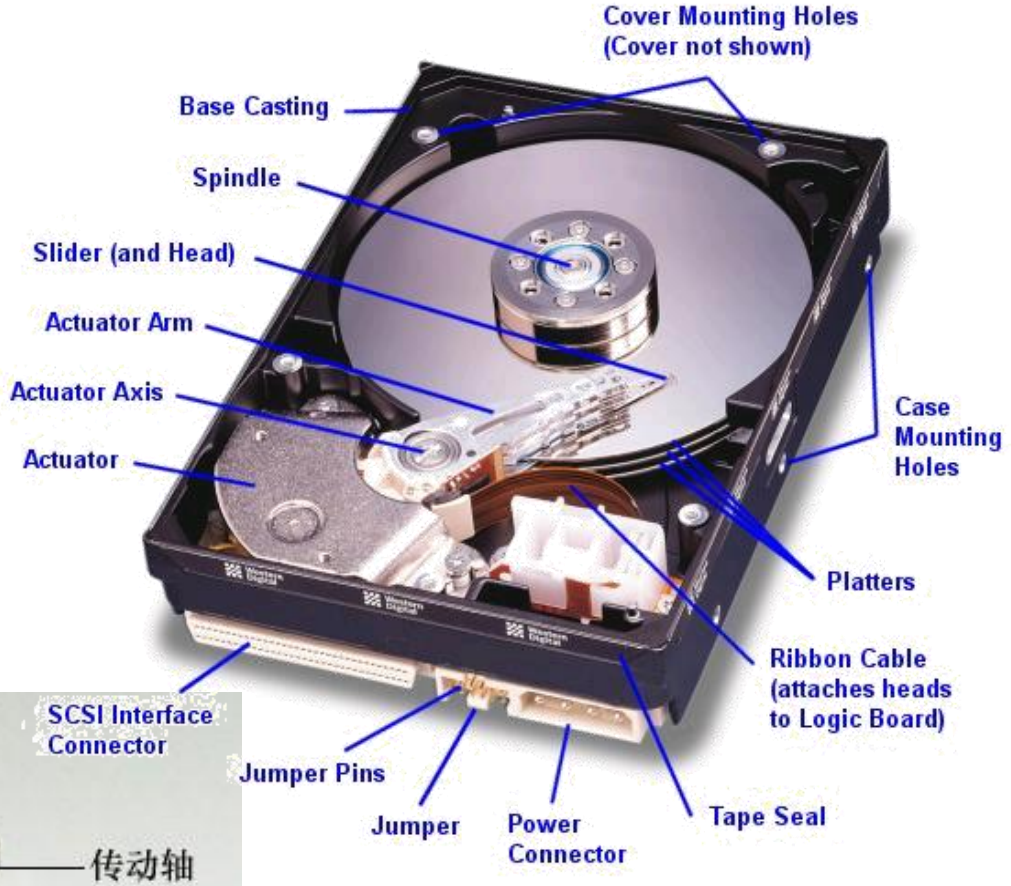
80~84年
5.25吋硬盘

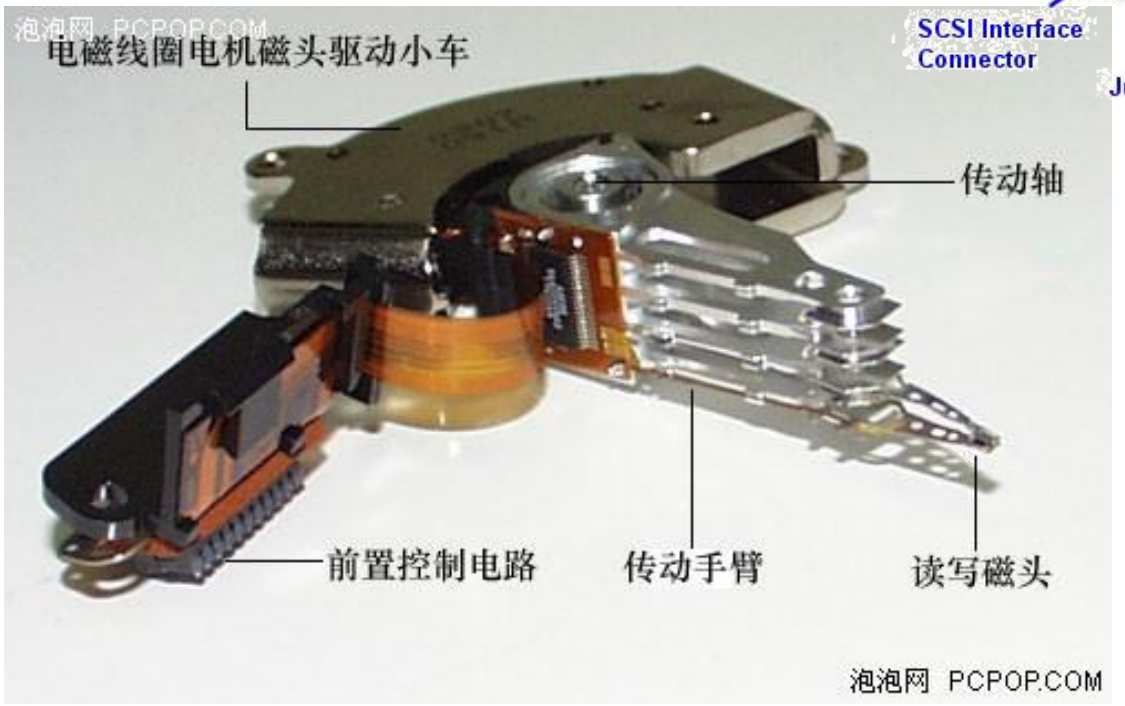
84~89年
3.5吋硬盘

89~92年
2.5吋硬盘

92年~今
1.8吋硬盘

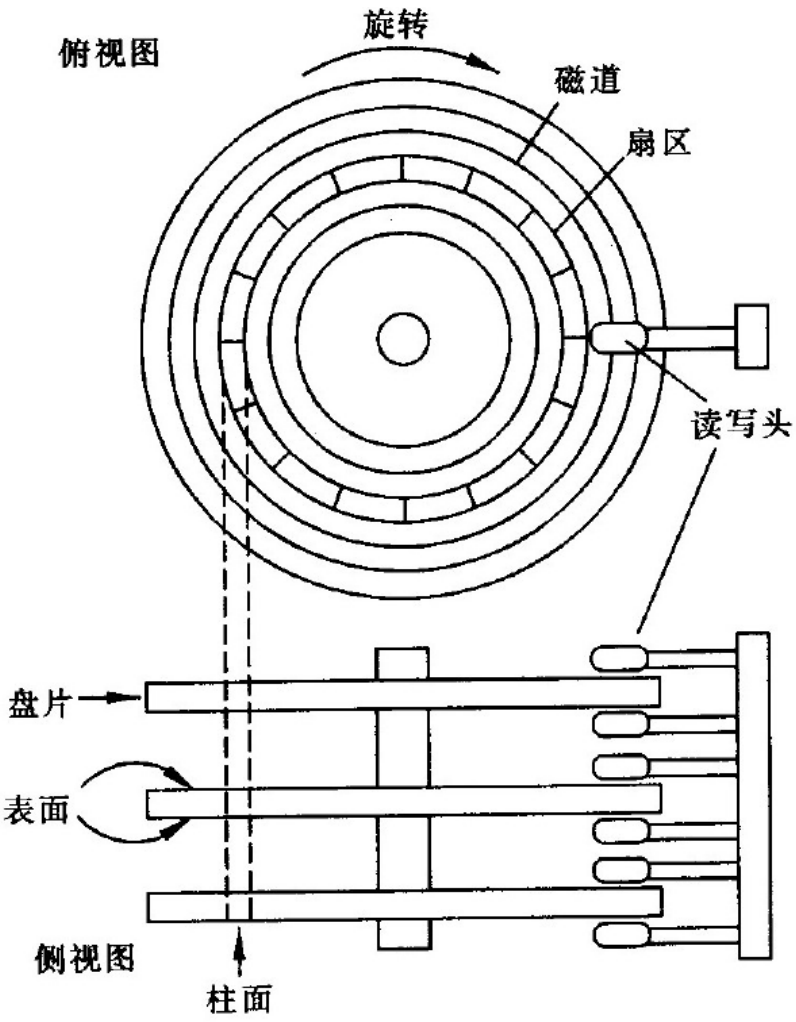
硬盘的机械结构


Cover Mounting Holes
(Cover not shown)
Base Casting
Spindle
Slider (and Head)
Actuator Arm
Actuator Axis
Actuator
Case
Mounting
Holes
Platters
Ribbon Cable
(attaches heads
to Logic Board)
SCSI Interface
Connector
Jumper Pins
Jumper
Power
Connector
Tape Seal


泡泡网 PCPOP.COM
电磁线圈电机磁头驱动小车
传动轴
前置控制电路
传动手臂
读写磁头
泡泡网 PCPOP.COM

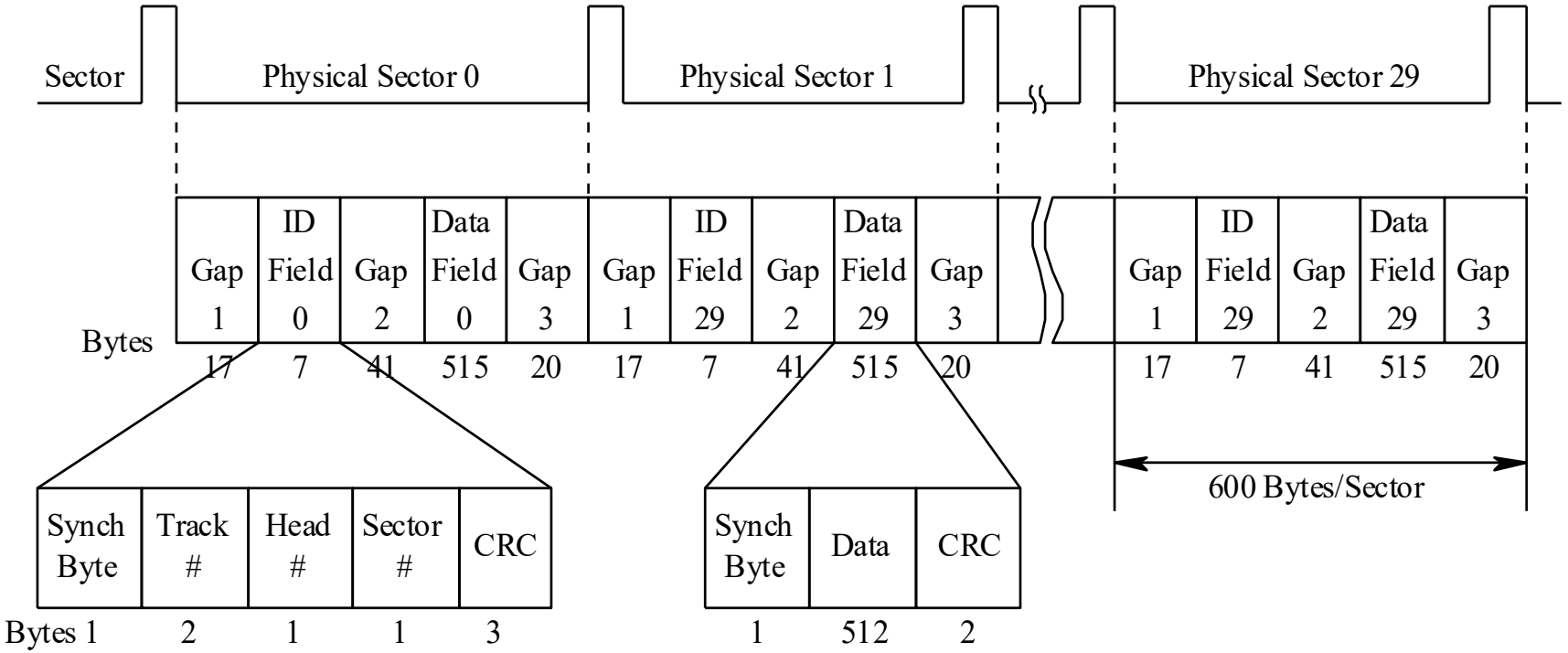
基本概念

- **扇区（sector）**
  - 盘片被分成许多扇形的区域
- **磁道（track）**
  - 盘片上以盘片中心为圆心，不同半径的同心圆。
- **柱面（cylinder）**
  - 硬盘中，不同盘片相同半径的磁道所组成的圆柱。
- 每个磁盘有两个面，每个面都有一个**磁头(head)**。

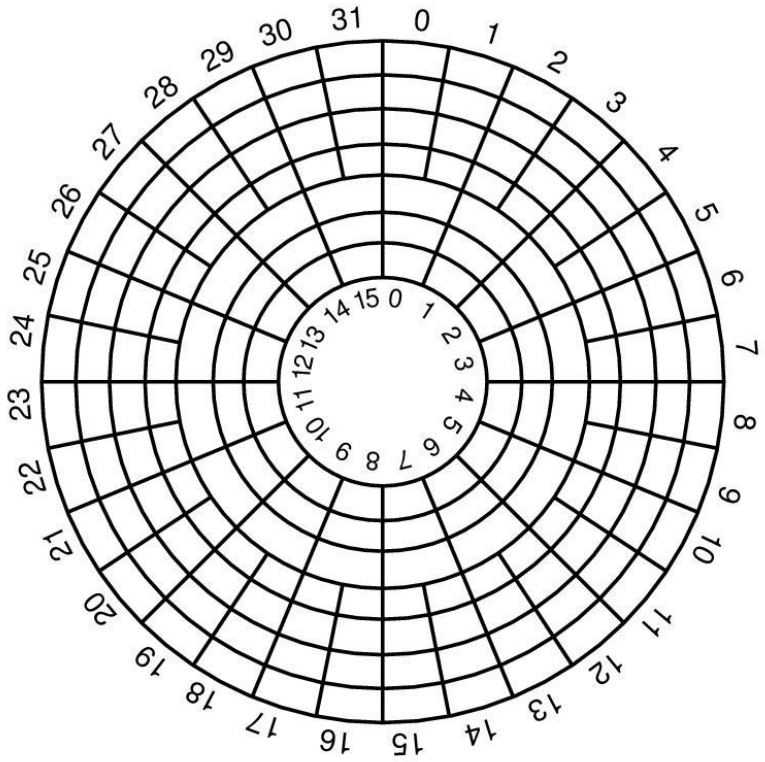


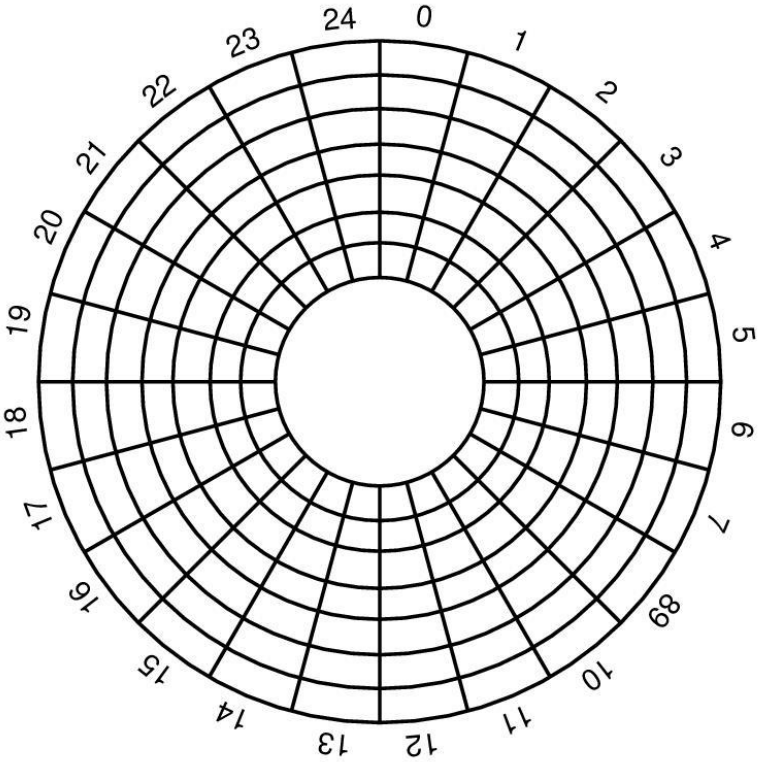
磁盘设备结构示意图

扇区组织及内部结构



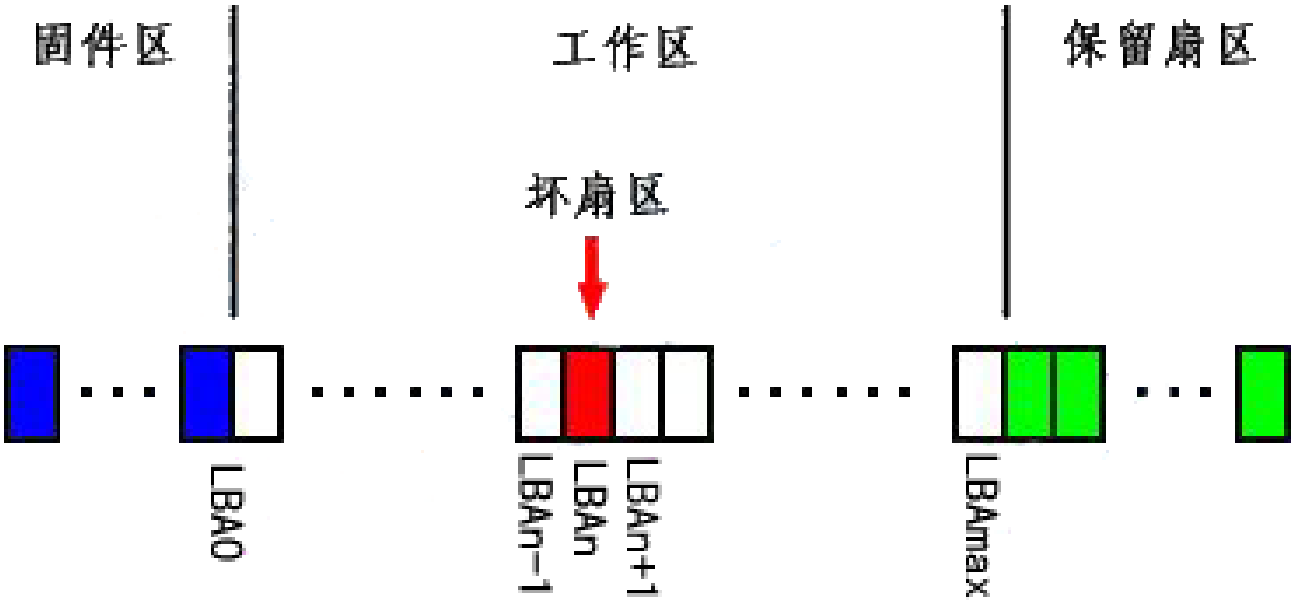
- 对于磁盘，每个磁道的扇区数并不是常量。
- 绝大多数磁盘都有一些缺陷扇区，因此映射必须用磁盘上的其他空闲扇区来替代这些缺陷扇区。


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

磁盘缺陷

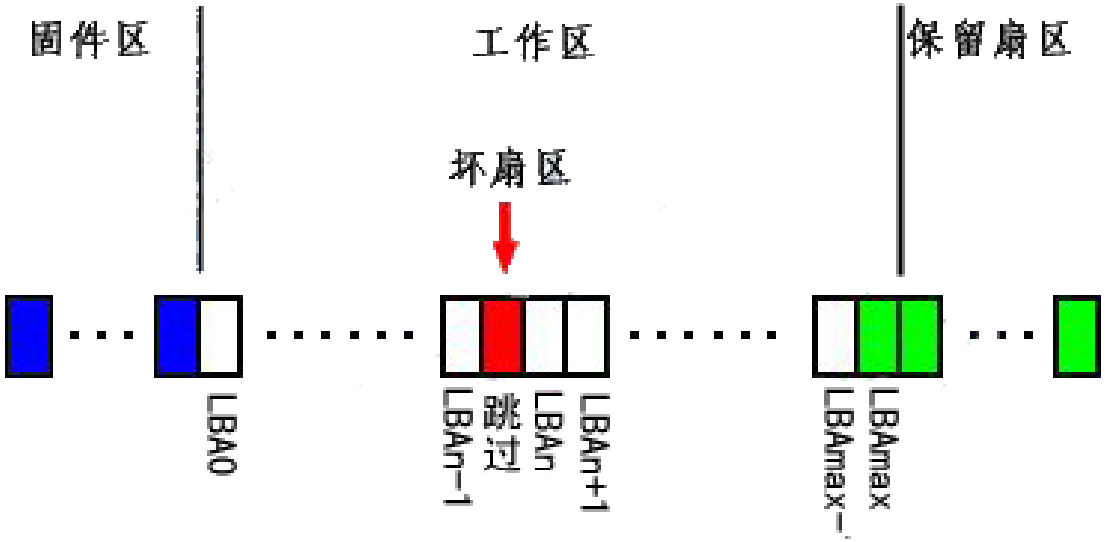
- 硬盘实际扇区数比硬盘标签上标定的大，其中一部份用于存储硬盘的固件（硬盘控制器使用）；
- 一部分是用户存储数据的区域，即工作区，也就是硬盘标定容量的扇区；
- 剩下的就是保留区，超过在固件里定义的硬盘容量的那些扇区就称为保留扇区。



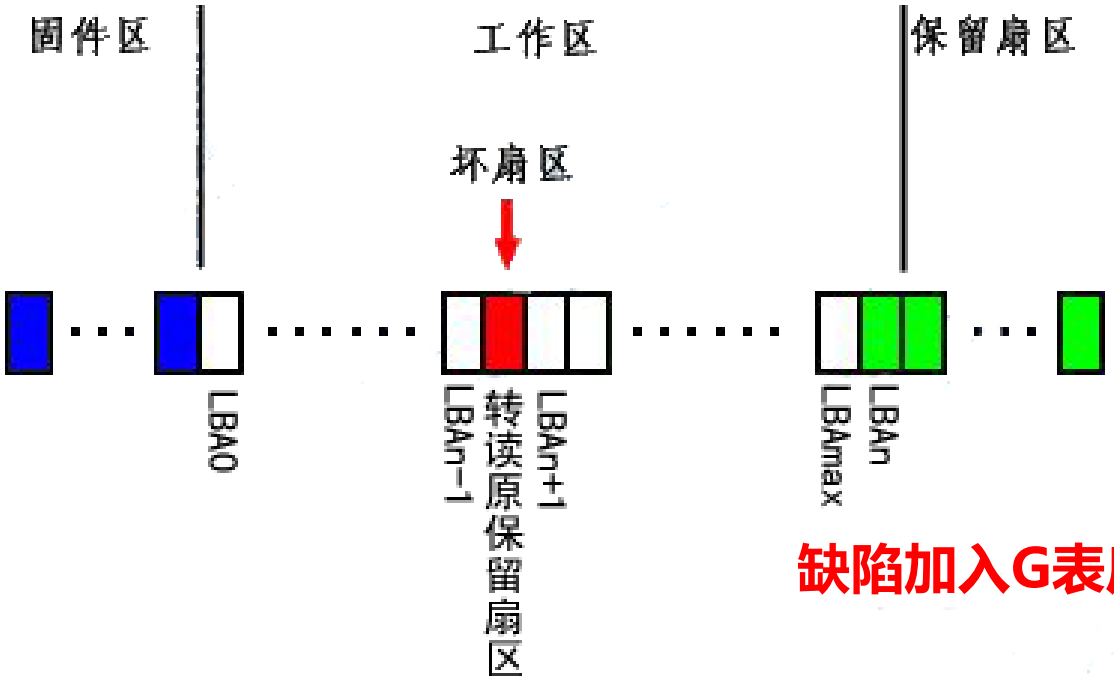
磁盘扇区的一般结构

磁盘缺陷

- P表：又称为永久缺陷列表，用于记录硬盘生产过程中产生的缺陷。



缺陷加入P表后的扇区结构



- G表：G表又称为增长缺陷列表，用于记录硬盘使用过程中由于磁介质性能变弱而引起的缺陷。

缺陷加入G表后的扇区结构

内容提要

- 磁盘的历史及工作原理
- 磁盘的组织与调度算法
- 磁盘空间的管理
- RAID
- 提高I/O速度
- 磁盘管理实例

磁盘的组织

- 主引导扇区(MBR)
- 分区表(DPT)
- 分区引导扇区(DBR)
- 容量及访问时间

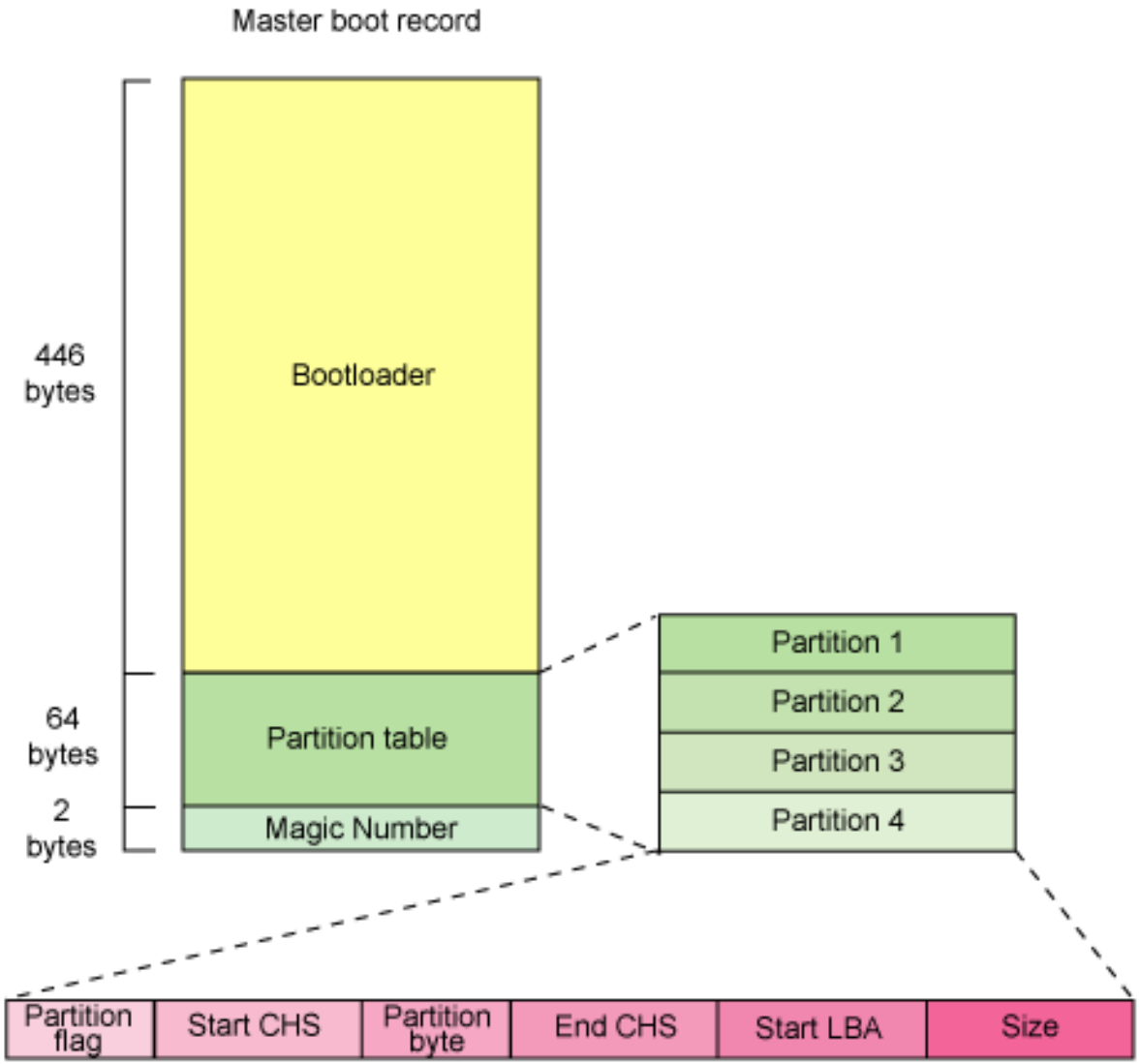
主引导扇区（MBR）

- 硬盘的0柱面、0磁头、1扇区称为主引导扇区（也叫主引导记录MBR），该记录占用512个字节，它用于硬盘启动时将系统控制权转给用户指定的、在分区表中登记了某个操作系统分区。MBR的内容是在硬盘分区时由分区软件（如FDISK）写入该扇区的，MBR不属于任何一个操作系统，不随操作系统的不同而不同，即使不同，MBR也不会夹带操作系统的性质，具有公共引导的特性。

MBR的结构

- MBR(Master Boot Record)主引导记录包含两部分的内容，前446字节为启动代码及数据；
- 之后则是分区表（DPT），分区表由四个分区项组成，每个分区项数据为16字节，记录了启动时需要的分区参数。这64个字节分布在MBR的第447-510字节。
- 后面紧接着两个字节AA和55被称为幻数(Magic Number), BOIS读取MBR的时候总是检查最后是不是有这两个幻数,如果没有就被认为是一个没有被分区的硬盘。

MBR (Master Boot Record)



MBR

Address		Description		Size (bytes)
Hex	Dec			
+0x0000	+0	Bootstrap code area		446
+0x01BE	+446	Partition entry #1	*Partition table* (for primary partitions)	16
+0x01CE	+462	Partition entry #2		16
+0x01DE	+478	Partition entry #3		16
+0x01EE	+494	Partition entry #4		16
+0x01FE	+510	55h	*Boot signature*	2
+0x01FF	+511	AAh		
		Total size: 446 + 4×16 + 2		512

MBR

- 由于MBR的限制 只能有4个主分区，系统必须装在主分区上面。
- 硬盘分区有三种，主磁盘分区、扩展磁盘分区、逻辑分区。
- 一个硬盘主分区至少有1个，最多4个，扩展分区可以没有，最多1个。且主分区+扩展分区总共不能超过4个。逻辑分区可以有若干个。
- 主分区只能有一个是激活的（active），其余为inactive。

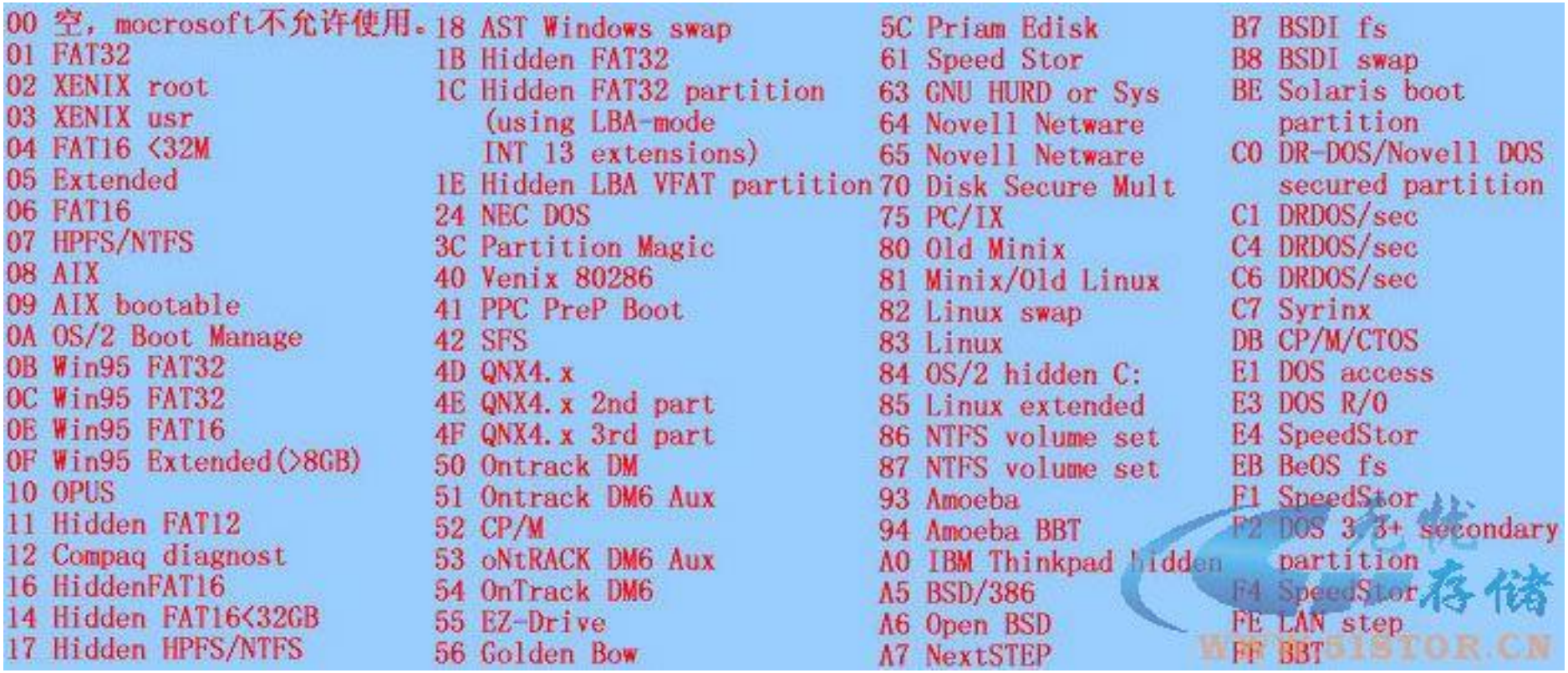
MBR

- 分出主分区后，其余的部分可以分成扩展分区，一般是剩下的部分全部分成扩展分区，也可以不全分，剩下的部分就浪费了。
- 扩展分区不能直接使用，必须分成若干逻辑分区。所有的逻辑分区都是扩展分区的一部分。

硬盘分区表DPT

- 从偏移01BEH开始到偏移01FDH结束的64字节
- 硬盘分区表分为四小部分，每一小部分表示一个分区的信息，占16字节
- 第0个字节是自举标志，其值为80H时表示该分区是当前活动分区
- 第4字节是分区类型

存贮字节位	内容及含义
第1字节	引导标志。若值为80H表示活动分区，若值为00H表示非活动分区。
第2、3、4字节	本分区的起始磁头号、扇区号、柱面号。其中： 磁头号——第2字节； 扇区号——第3字节的低6位； 柱面号——为第3字节高2位+第4字节8位。
第5字节	分区类型符。 00H——表示该分区未用（即没有指定）； 06H——FAT16基本分区； 0BH——FAT32基本分区； 05H——扩展分区； 07H——NTFS分区； 0FH——（LBA模式）扩展分区（83H为Linux分区等）。
第6、7、8字节	本分区的结束磁头号、扇区号、柱面号。其中： 磁头号——第6字节； 扇区号——第7字节的低6位； 柱面号——第7字节的高2位+第8字节。
第9、10、11、12字节	本分区之前已用了的扇区数。
第13、14、15、16字节	本分区的总扇区数。


00 空，mocrosoft不允许使用。
01 FAT32
02 XENIX root
03 XENIX usr
04 FAT16 <32M
05 Extended
06 FAT16
07 HPFS/NTFS
08 AIX
09 AIX bootable
0A OS/2 Boot Manage
0B Win95 FAT32
0C Win95 FAT32
0E Win95 FAT16
0F Win95 Extended(>8GB)
10 OPUS
11 Hidden FAT12
12 Compaq diagnost
16 HiddenFAT16
14 Hidden FAT16<32GB
17 Hidden HPFS/NTFS
18 AST Windows swap
1B Hidden FAT32
1C Hidden FAT32 partition
(using LBA-mode
INT 13 extensions)
1E Hidden LBA VFAT partition
24 NEC DOS
3C Partition Magic
40 Venix 80286
41 PPC PreP Boot
42 SFS
4D QNX4.x
4E QNX4.x 2nd part
4F QNX4.x 3rd part
50 Ontrack DM
51 Ontrack DM6 Aux
52 CP/M
53 oNtRACK DM6 Aux
54 OnTrack DM6
55 EZ-Drive
56 Golden Bow
5C Priam Edisk
61 Speed Stor
63 GNU HURD or Sys
64 Novell Netware
65 Novell Netware
70 Disk Secure Mult
75 PC/IX
80 Old Minix
81 Minix/Old Linux
82 Linux swap
83 Linux
84 OS/2 hidden C:
85 Linux extended
86 NTFS volume set
87 NTFS volume set
93 Amoeba
94 Amoeba BBT
A0 IBM Thinkpad hidden
A5 BSD/386
A6 Open BSD
A7 NextSTEP
B7 BSDI fs
B8 BSDI swap
BE Solaris boot
partition
C0 DR-DOS/Novell DOS
secured partition
C1 DRDOS/sec
C4 DRDOS/sec
C6 DRDOS/sec
C7 Syrinx
DB CP/M/CTOS
E1 DOS access
E3 DOS R/O
E4 SpeedStor
EB BeOS fs
F1 SpeedStor
F2 DOS 3.3+ secondary
partition
F4 SpeedStor
FE LAN step
FF BBT

8.46GB问题

- 第2至第4字节是该分区起始地址。其中第2字节为起始磁头号（面号）；第3字节的低6位为起始扇区号，高2位则为起始柱面号的高2位；第4字节为起始柱面号的低8位。
- 柱面号最大值为2^10 = 1024（0~1023）
- 扇区号不会超过2^6 - 1 = 63（0~63）
- 磁头号不会超过2^8 = 256（0~255）
  - 1024 × 256 × 63 = 16,515,072
- 这个扇区之前的所有物理扇区所包含的字节数为：
  - 16,515,072 × 512Bytes ≈ 8.46GB。

INT 13 服务扩展标准

- 硬盘厂商定义了新的INT 13服务扩展标准以扩大可支持的硬盘容量。
  - INT 13服务扩展标准不再使用OS的寄存器传递硬盘的寻址参数，而使用存储在OS内存里的地址包。
  - 地址包里保存的是64位LBA地址，如果硬盘支持 LBA寻址，就把低28位直接传递给ATA接口，如果不支持，操作系统就先把LBA地址转换为CHS地址，再传递给ATA接口。
  - 通过这种方式，能实现在 ATA总线基础上CHS寻址的最大容量是136.9 GB，而LBA寻址的最大容量是137.4GB。
  - 新的硬盘传输规范ATA 133规范又把28位可用的寄存器空间提高到48位，从而支持更大的硬盘。

磁盘地址与块号的转换

- 磁盘地址（CHS）：Cylinder，Head，Sector；
- 块号：又称逻辑块号（LBA），Logic Block Address，现代磁盘驱动器可以看做一个一维的逻辑块的数组，逻辑块是最小的传输单位。

已知块号，则磁盘地址：

柱面号＝[块号/（磁头数×扇区数）]

磁头号＝[（块号mod（磁头数×扇区数））/扇区数]

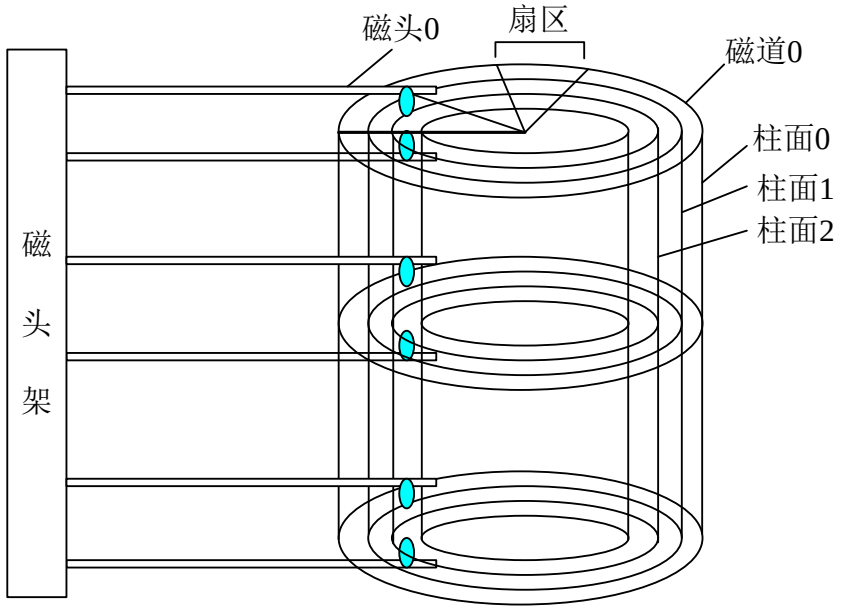
扇区号＝（块号mod（磁头数×扇区数））mod 扇区数

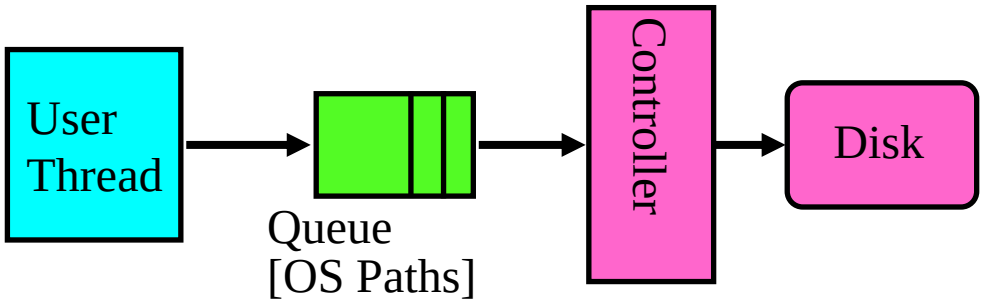
已知磁盘地址：

块号＝柱面号×（磁头数×扇区数）＋磁头号×扇区数＋扇区号

磁盘访问时间

- 寻道时间
- 旋转延迟时间
- 传输时间





Response Time = Queue+Disk Service Time

寻道时间与旋转延迟时间

寻道时间：

- 把磁臂（磁头）从当前位置移动到指定磁道上所经历的时间。该时间是启动磁盘的时间s与磁头移动n条磁道所花费的时间之和。

$$Ts = m \times n + s$$，其中m是一个常数。

旋转延迟时间：

$$Tr = 1/(2r)$$

- 硬盘典型的旋转速度为3600 r/min，每转需时16.7 ms，平均旋转延迟时间*Tr*为8.3 ms。软盘旋转速度为300或600 r/min，平均*Tr*为50~100 ms。

传输时间

- Tt是指把数据从磁盘读出，或向磁盘写入数据所经历的时间，Tt的大小与每次所读/写的字节数b，旋转速度r以及磁道上的字节数N有关

$$T\text{t} = \text{b}/(\text{rN})$$

  - 旋转速度r, 通常为: 3600 RPM to 15000 RPM
  - 因此，硬盘1秒钟可传输的数据为2MB~ 50MB

访问时间

- 访问时间=寻道时间+旋转延迟时间+传输时间，即

$$Ta = Ts + 1/(2r) + b/(rN)$$

- 例如，我们假定寻道时间和旋转延迟时间平均为30 ms，而磁道的传输速率为1 MB/s，如果传输1K字节，此时总的访问时间为31 ms，传输时间所占比例是相当地小。当传输10K字节的数据时，其访问时间也只是40 ms，即当传输的数据量增加10倍时，访问时间只增加了约30%。

磁盘调度算法

- 先来先服务算法（FCFS），
- 最短寻道时间优先算法（SSTF，Shortest Seek Time First），
- 扫描算法（SCAN），
- 循环扫描算法（CSCAN）

先来先服务算法（FCFS）

- 算法思想：

  按访问请求到达的先后次序服务。

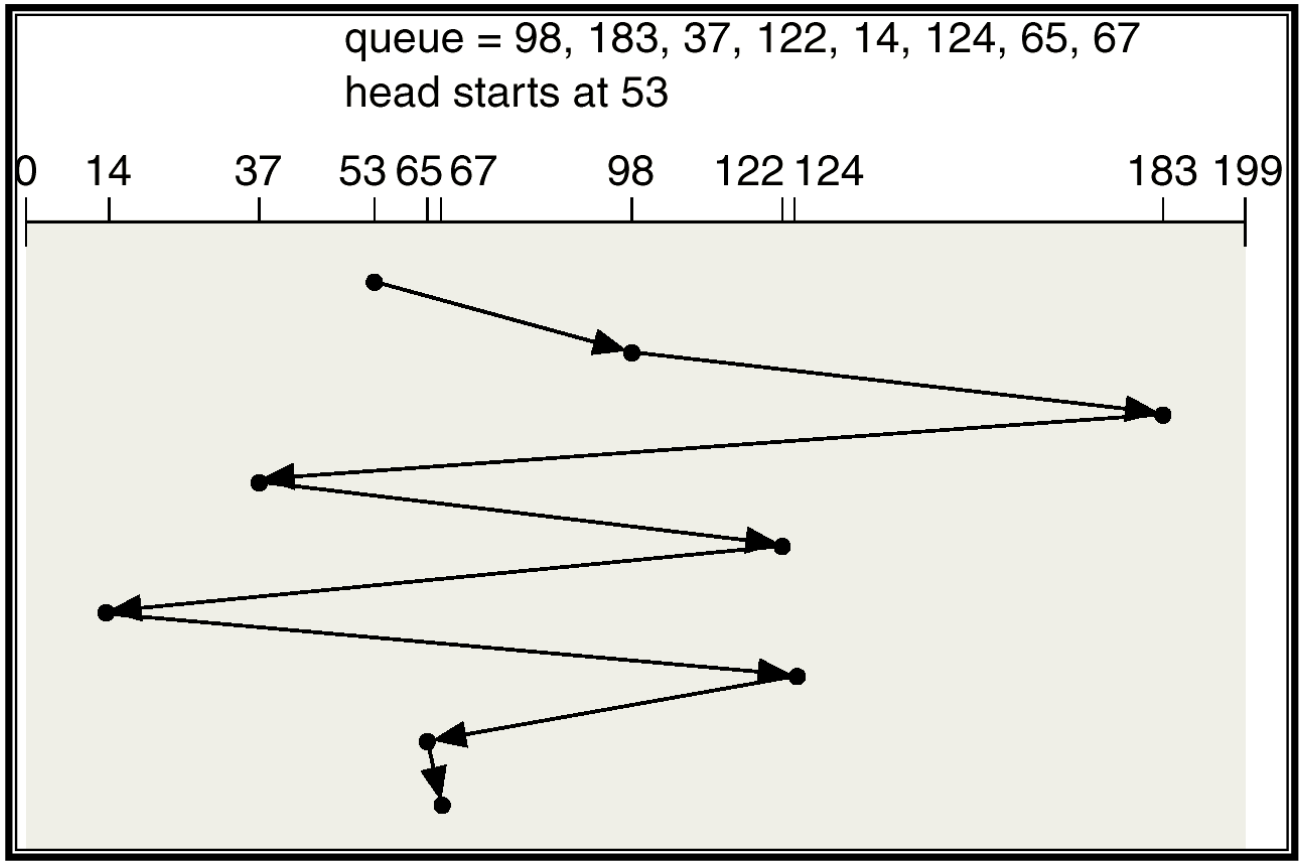
- 优点：

  简单，公平。

- 缺点：

  效率不高，相邻两次请求可能会造成最内到最外的柱面寻道，使磁头反复移动，增加了服务时间，对机械也不利。

先来先服务



- 总行程：640 cylinders

最短寻道时间优先算法（SSTF）

- 算法思想：

  优先选择距当前磁头最近的访问请求进行服务，主要考虑寻道优先。

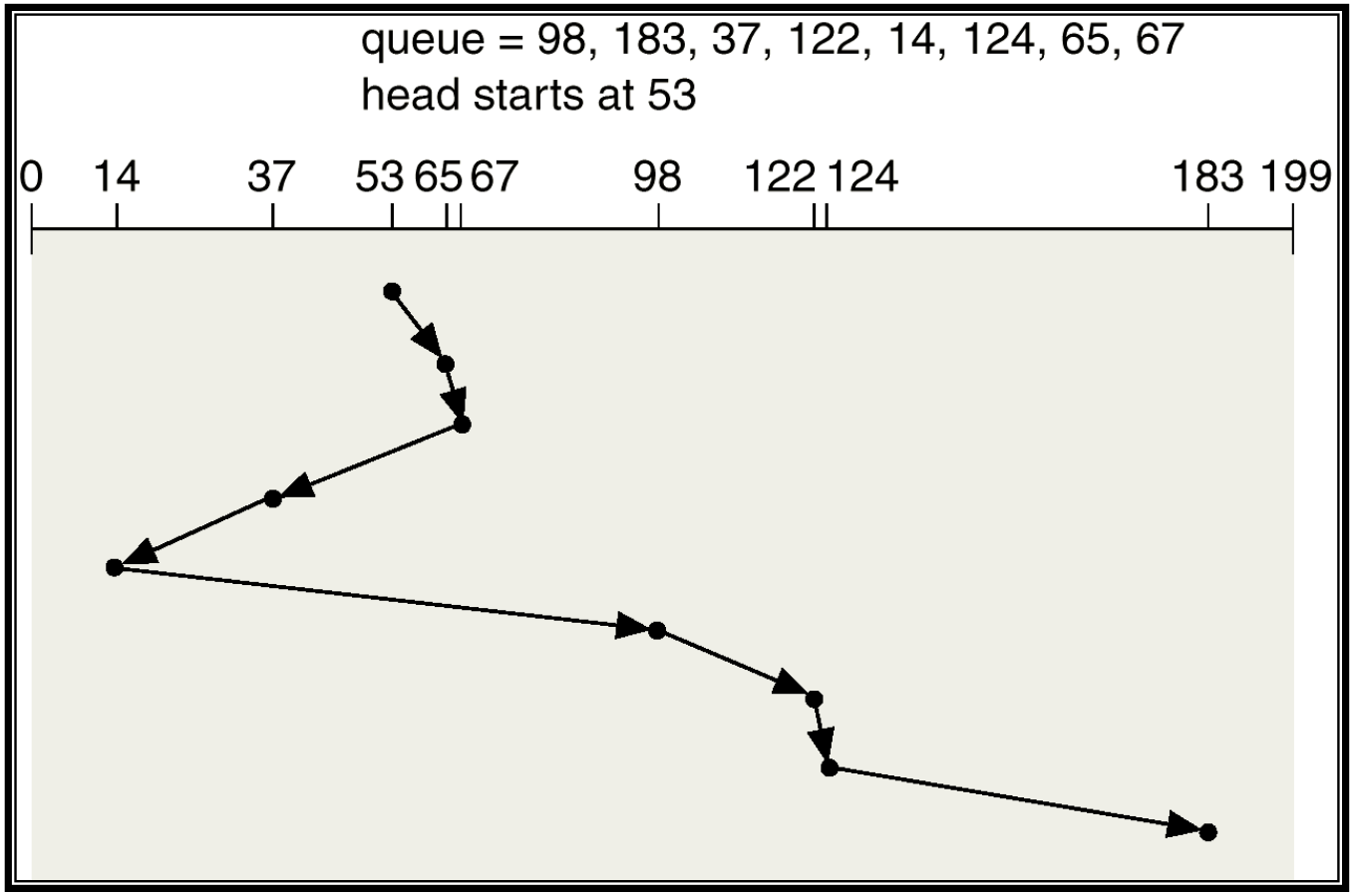
- 优点：

  改善了磁盘平均服务时间。

- 缺点：

  可能产生“饥饿”现象，造成某些访问请求长期等待得不到服务。

最短寻道时间优先



- 总行程：236 cylinders.

扫描算法（SCAN）电梯调度

- 算法思想：

  当有访问请求时，磁头按一个方向移动，在移动过程中对遇到的访问请求进行服务，然后判断该方向上是否还有访问请求，如果有则继续扫描；否则改变移动方向，并为经过的访问请求服务，如此反复。

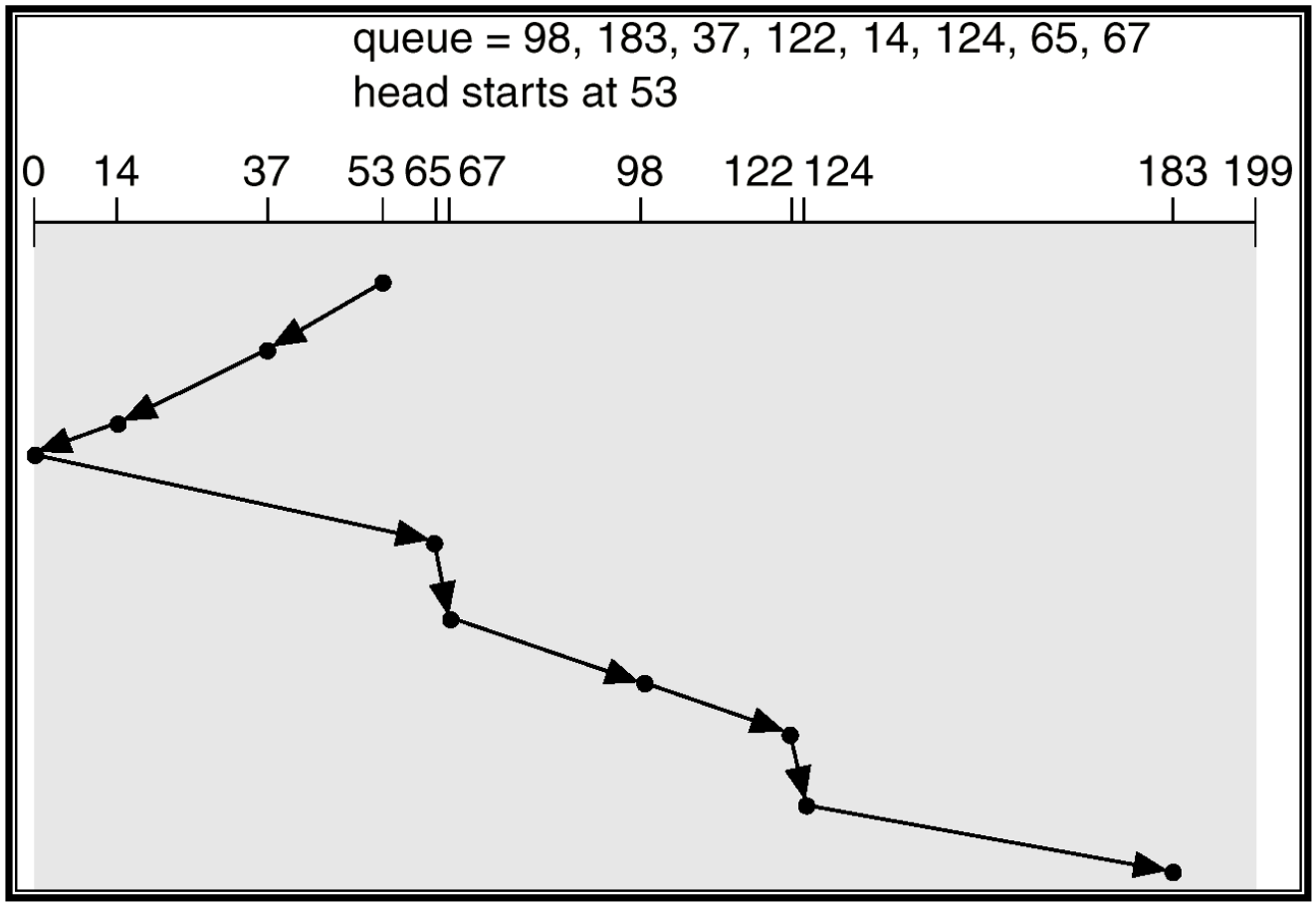
- 优点：

  克服了最短寻道优先的缺点，既考虑了距离，同时又考虑了方向

- 缺点：

  但由于是摆动式的扫描方法，两侧磁道被访问的频率仍低于中间磁道。

扫描算法

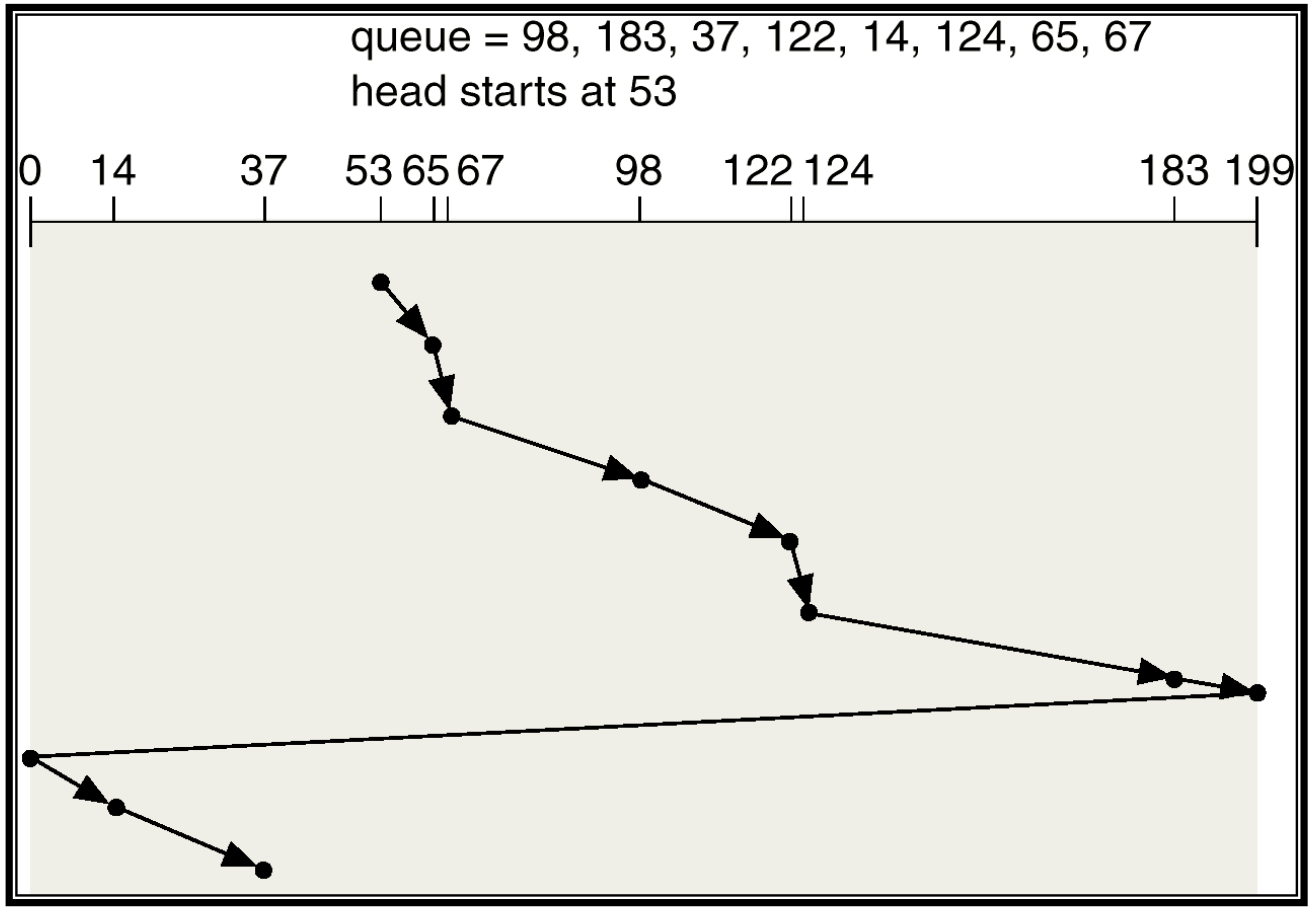


- 总行程：208 cylinders.

循环扫描算法（CSCAN）

- 算法思想：
  - 按照所要访问的柱面位置的次序去选择访问者。
  - 移动臂到达最后一个柱面后，立即带动读写磁头快速返回到0号柱面。
  - 返回时不为任何的等待访问者服务。
  - 返回后可再次进行扫描。
- 由于SCAN算法偏向于处理那些接近最里或最外的磁道的访问请求，所以使用改进型的C-SCAN算法可避免这个问题。

循环扫描算法



- 总行程：382 cylinders.

算法比较

	优点	缺点
FCFS算法	公平、简单	平均寻道距离大，仅应用在磁盘I/O较少的场合
SSTF算法	性能比“先来先服务”好	不能保证平均寻道时间最短，可能出现“饥饿”现象
SCAN算法	寻道性能较好，可避免“饥饿”现象	不利于远离磁头一端的访问请求
C-SCAN算法	消除了对两端磁道请求的不公平	--

LOOK和C-LOOK

- 采用SCAN算法和C-SCAN算法时磁头总是严格地遵循从盘面的一端到另一端，显然，在实际使用时还可以改进，即磁头移动只需要到达最远端的一个请求即可返回，不需要到达磁盘端点。
- 这种形式的SCAN算法和C-SCAN算法称为LOOK和C-LOOK调度。这是因为它们在朝一个给定方向移动前会查看是否有请求。注意，若无特别说明，也可以默认SCAN算法和C-SCAN算法为LOOK和C-LOOK调度。

- N-Step-SCAN
  - 将磁盘请求队列分成若干个长度为N的子队列，解决磁头粘黏问题
  - N很大～SCAN；N＝1～FCFS
- FSCANS
  - 分为两个区域（当前请求和新请求）

内容提要

- 磁盘的历史及工作原理
- 磁盘的组织与调度算法
- 磁盘空间的管理
- RAID
- 提高I/O速度
- 磁盘管理实例

磁盘空间的管理

- 位图
- 空闲表法
- 空闲链表法
- 成组链接法

位图

- 用一串二进制位反映磁盘空间中分配使用情况，每个物理块对应一位，分配的物理块为0，否则为1。
- 申请物理块时，可以在位示图中查找1的位，返回对应的物理块号；归还时，将对应位转置为1。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0
2	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1
3	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
4																
⋮																
16																

空闲表

- 将所有空闲块记录在一个表中，即空闲表；
- 主要记录两项内容：起始块号，块数

序号	第一空闲盘块号	空闲盘块数
1	2	4
2	9	3
3	15	5
4	—	—

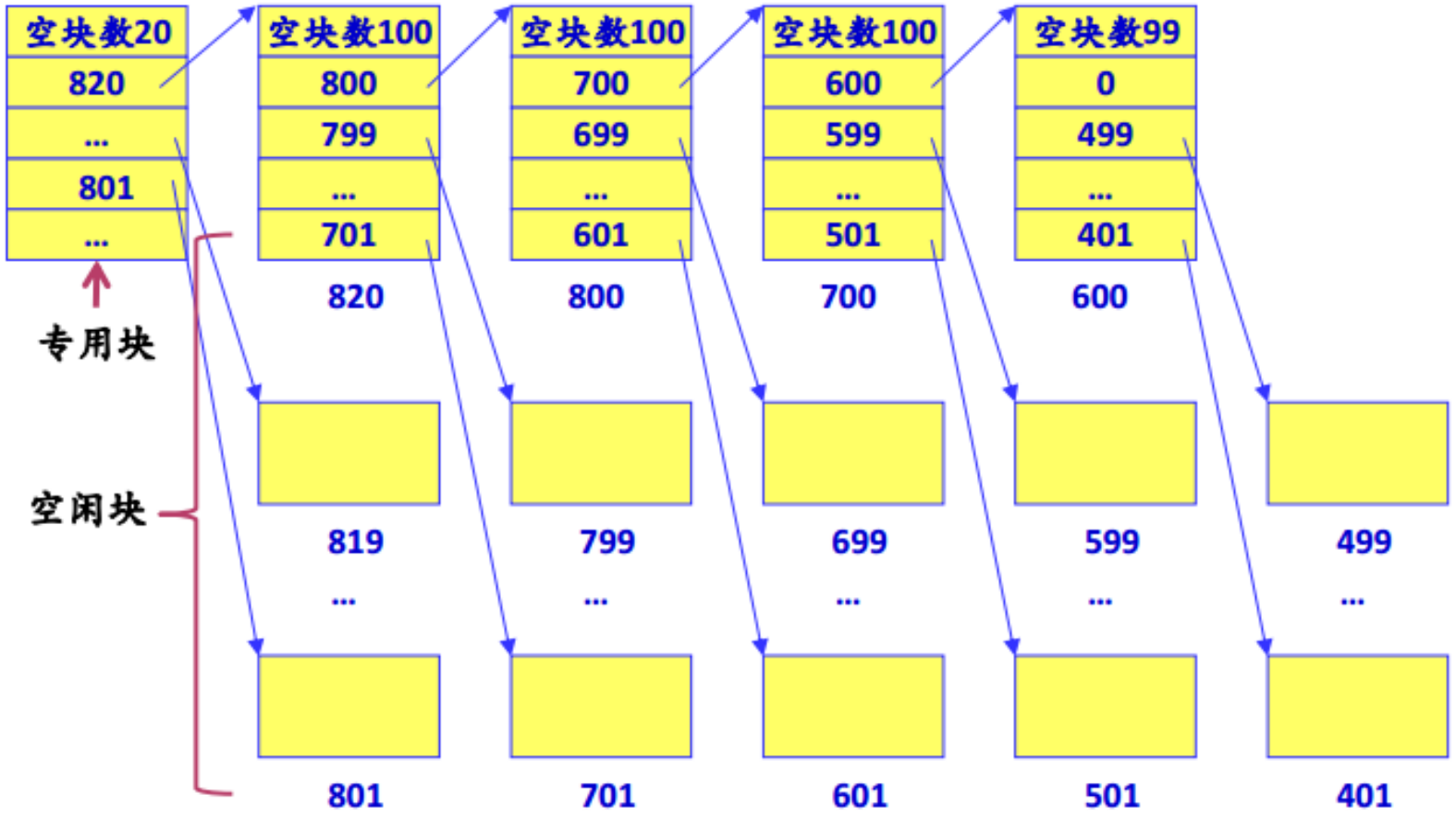
成组链接法

- 空闲块链表：把所有空闲块链成一个表，链会很长。
- 成组链接法：

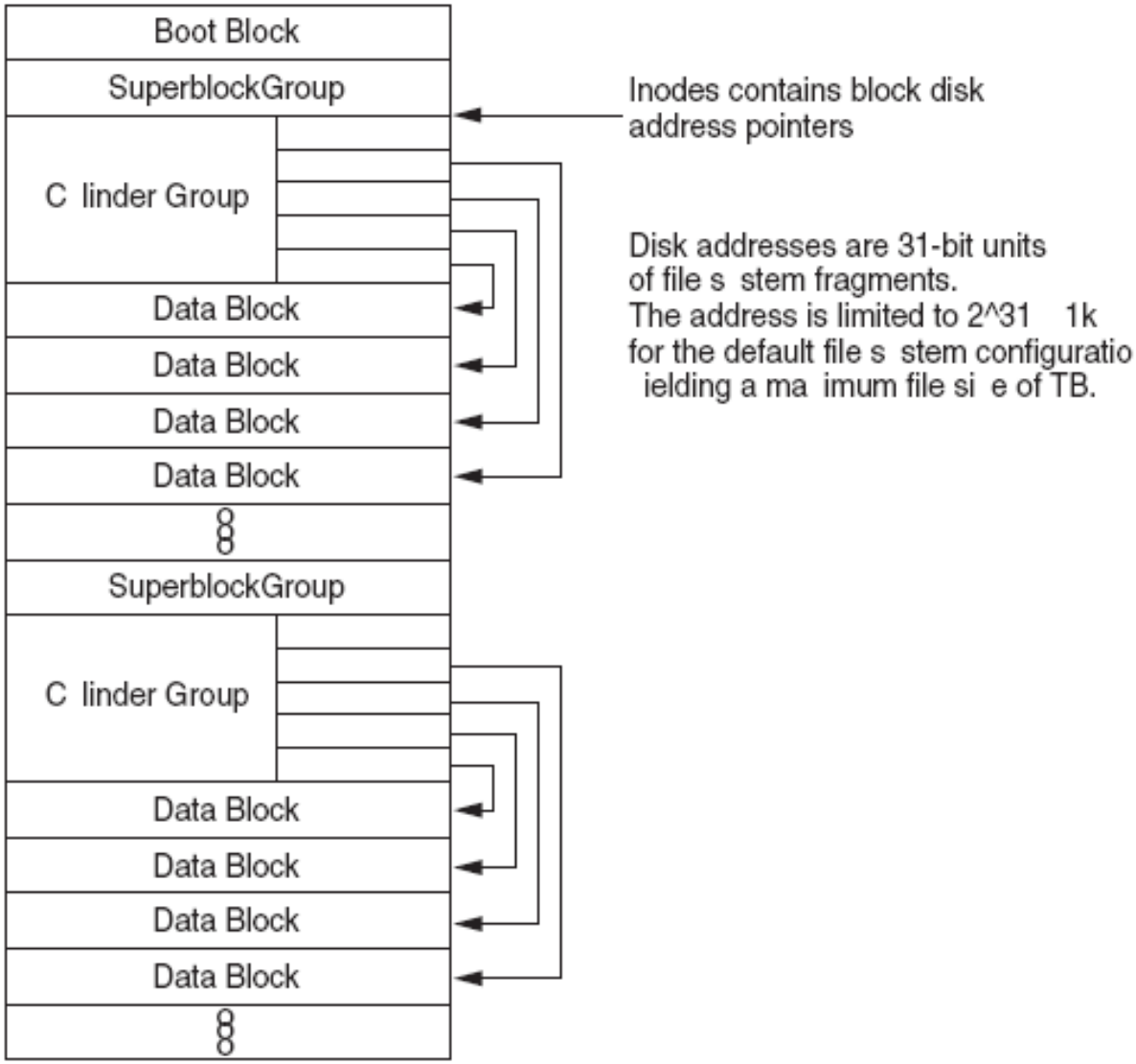
  把空白物理块分成组，在通过指针把组与组之间链接起来，这种管理空白块的方法称为成组链接法。
- 成组链接法的优点：

  1. 空白块号登记不占用额外空间；
  2. 节省时间；
  3. 采用后进先出的栈结构思想 。

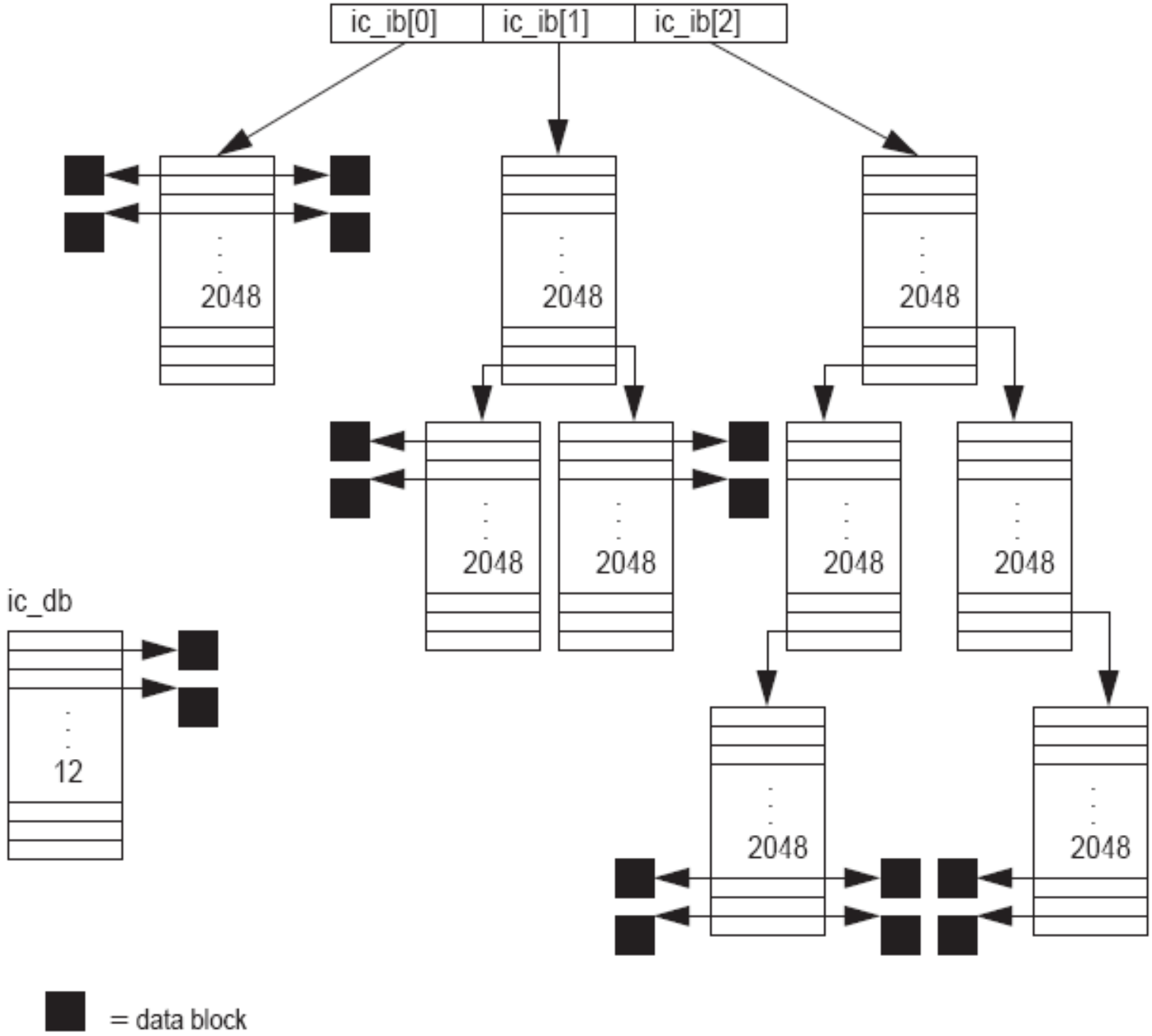
成组链接法


空块数20
820
...
801
...
专用块
空闲块
空块数100
800
799
...
701
820
819
...
801
空块数100
700
699
...
601
800
799
...
701
空块数100
600
599
...
501
700
699
...
601
空块数99
0
499
...
401
600
599
...
501
499
...
401

UFS


Boot Block
SuperblockGroup
C linder Group
Data Block
Data Block
Data Block
Data Block
SuperblockGroup
C linder Group
Data Block
Data Block
Data Block
Data Block
Inodes contains block disk address pointers
Disk addresses are 31-bit units of file s stem fragments.
The address is limited to 2^31 1k for the default file s stem configuratio ielding a ma imum file si e of TB.


fs superblock
struct cg + bitmaps
inodes
data blocks ...


ic_ib[0]
ic_ib[1]
ic_ib[2]
2048
2048
2048
2048
2048
2048
2048
2048
ic_db
12
= data block

- 8K*（（12+2048）+（2048*2048）+（2048*2048*2048））=64Tbytes
- 2^31*1K=2TB

内容提要

- 磁盘的历史及工作原理
- 磁盘的组织与调度算法
- 磁盘空间的管理
- RAID
- 提高I/O速度
- 磁盘管理实例

小专题：Flash Disk

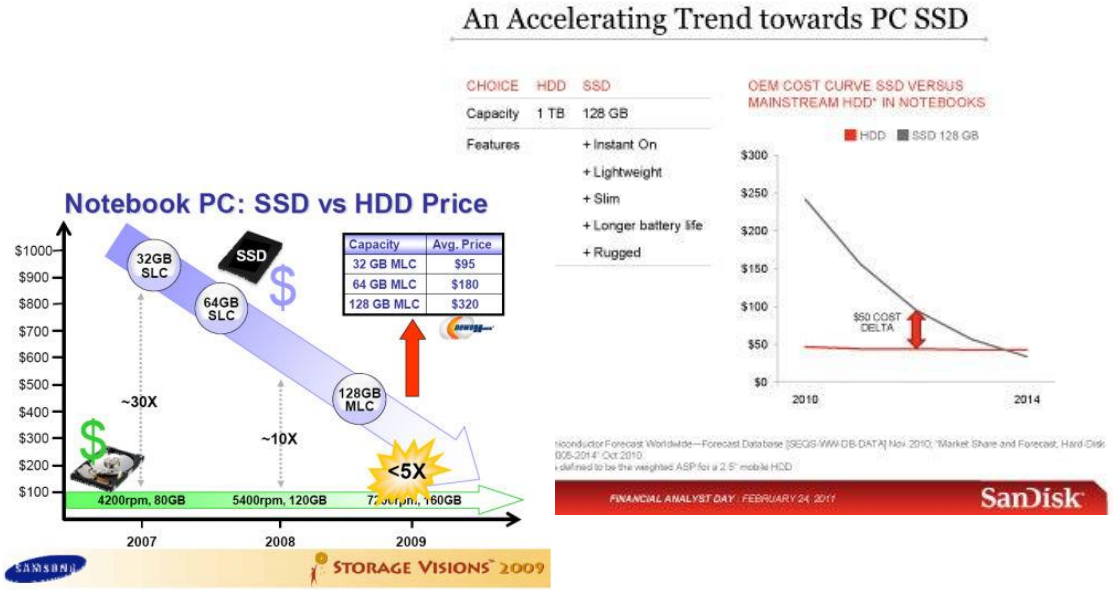
随着半导体技术的发展，半导体盘的应用越来越广泛。

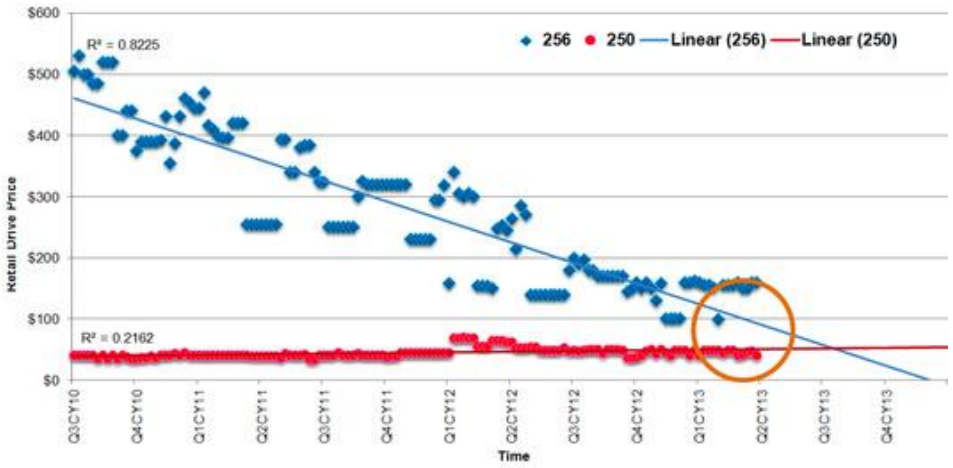
	SM	CF	MMC	SD	Memory Stick
发布时间	1995 年	1994 年	1997 年	1999 年	1997 年
发布厂商	东芝	SanDisk	SanDisk 西门子	东芝、松下 SanDisk	索尼
物理尺寸(mm)	45*37*0.76	43*36*3.3	32*24*1.4	32*24*2.1	50*21.5*0.28
集成控制器	否	是	是	是	
兼容性	差	好	好	好	差
工作电压	3.3V, 5V	3.3V, 5V	2.7V-3.6V	2.7V－3.6V	2.7V-3.6V
公开标准	是	是	是	是	否
适用领域	MP3, DC	MM	MP3, DC, DV, MP	MP3, DC, DV, E-Book	仅索尼公司的产品
支持厂商	东芝、三星、 索尼、夏普等	3COM, IBM、HP、 佳能、松下等	SanDisk、西门 子等	三星等	索尼
其它		操作系统都支 持	有 MMC 和 SPI 两种模式	数据安全性高 支持物理写保护	索尼公司使用，无 其它公司支持

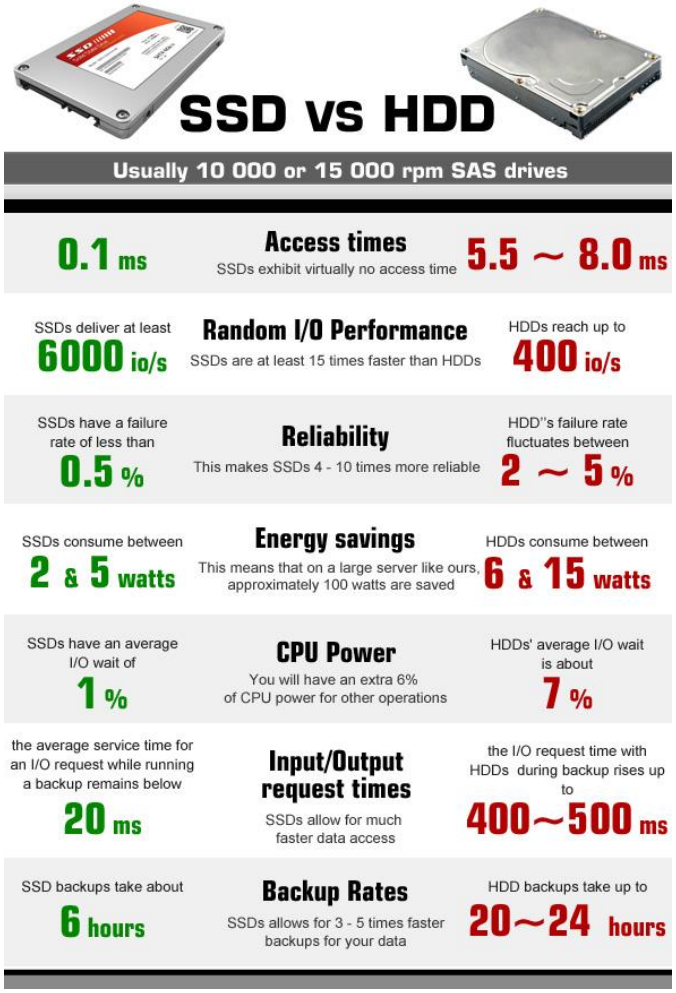
Flash Disk

- Flash盘的两种技术：
  - NOR技术是由Intel公司1988年首先推出
  - NAND技术是由东芝公司1989年发明
- 在数据传输速度：
  - NOR无论是在随机读取还是连续传输速度上都比NAND快，但相对而言，在连续大数据传输速度上，二者差异较小。
- 产品成本：
  - NOR Flash的平均每MB成本是NAND Flash成本的三到四倍。

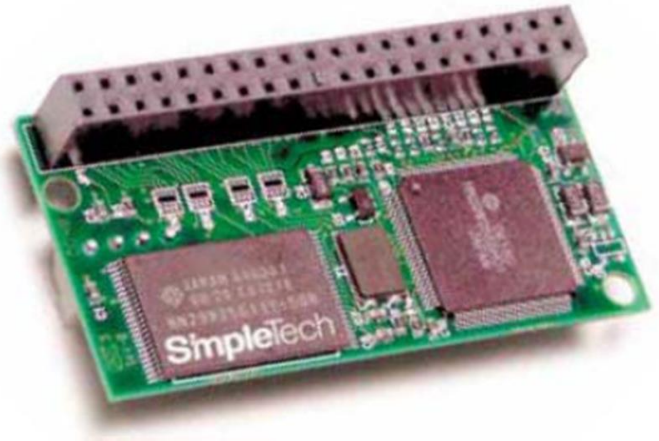
性价比

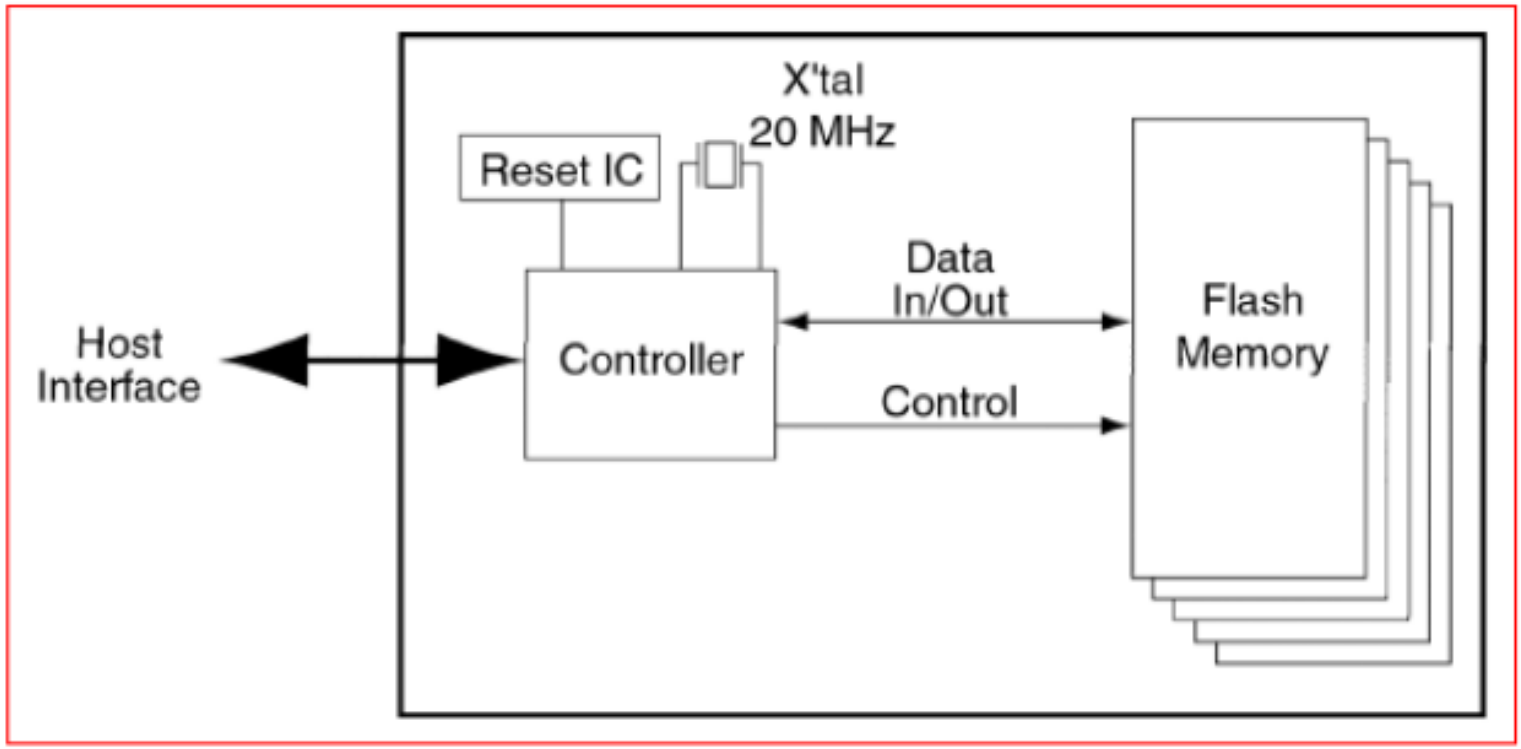

An Accelerating Trend towards PC SSD
CHOICE HDD SSD
Capacity 1 TB 128 GB
Features + Instant On
+ Lightweight
+ Slim
+ Longer battery life
+ Rugged
OEM COST CURVE SSD VERSUS MAINSTREAM HDD* IN NOTEBOOKS
HDD SSD 128 GB
$300
$250
$200
$150
$100
$50
$0
$50 COST DELTA
2010
2014
Notebook PC: SSD vs HDD Price
$1000
$900
$800
$700
$600
$500
$400
$300
$200
$100
32GB SLC
SSD
64GB SLC
128GB MLC
Capacity Avg. Price
32 GB MLC $95
64 GB MLC $180
128 GB MLC $320
~30X
~10X
<5X
4200rpm, 80GB
5400rpm, 120GB
2007
2008
2009
FINANCIAL ANALYST DAY · FEBRUARY 24, 2011
SanDisk
STORAGE VISIONS 2009


$600
$500
$400
$300
$200
$100
$0
R² = 0.8225
R² = 0.2162
256
250
Linear (256)
Linear (250)
Retail Drive Price
Q3CY10
Q4CY10
Q1CY11
Q2CY11
Q3CY11
Q4CY11
Q1CY12
Q2CY12
Q3CY12
Q4CY12
Q1CY13
Q2CY13
Q3CY13
Q4CY13
Time


SSD vs HDD
Usually 10 000 or 15 000 rpm SAS drives
0.1 ms
Access times
SSDs exhibit virtually no access time
5.5 ~ 8.0 ms
SSDs deliver at least
6000 io/s
Random I/O Performance
SSDs are at least 15 times faster than HDDs
HDDs reach up to
400 io/s
SSDs have a failure rate of less than
0.5 %
Reliability
This makes SSDs 4 - 10 times more reliable
HDD''s failure rate fluctuates between
2 ~ 5 %
SSDs consume between
2 & 5 watts
Energy savings
This means that on a large server like ours, approximately 100 watts are saved
HDDs consume between
6 & 15 watts
SSDs have an average I/O wait of
1 %
CPU Power
You will have an extra 6% of CPU power for other operations
HDDs' average I/O wait is about
7 %
the average service time for an I/O request while running a backup remains below
20 ms
Input/Output request times
SSDs allow for much faster data access
the I/O request time with HDDs during backup rises up to
400~500 ms
SSD backups take about
6 hours
Backup Rates
SSDs allows for 3 - 5 times faster backups for your data
HDD backups take up to
20~24 hours

Flash Disk


SimpleTech


X'tal
20 MHz
Reset IC
Data
In/Out
Host
Interface
Controller
Control
Flash
Memory

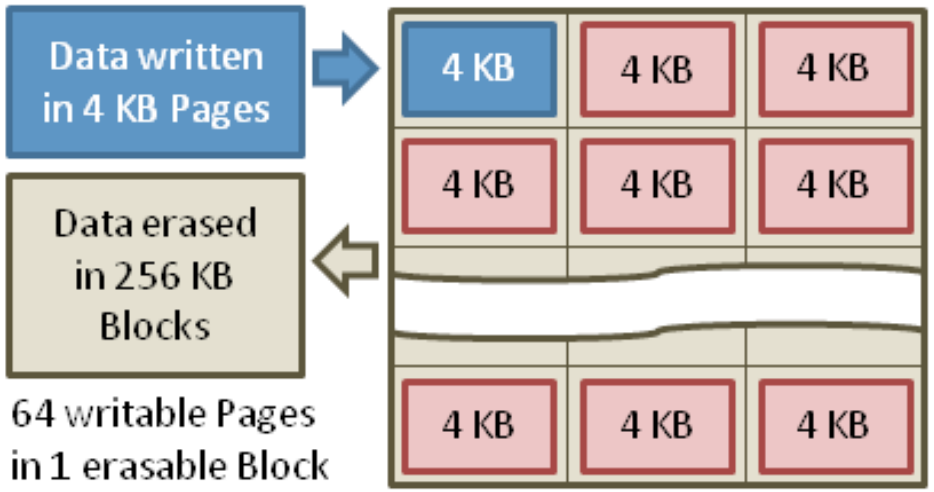
NAND / NOR

- NAND的地址分为三部分:块号、块内页号、页内字节号
  - 一次数据访问，NAND一般是经过三次寻址，先后确定块号、块内页号和页内字节号，至少占用三个时钟周期，因此随机字节读写速度慢。
- NOR的地址线足够多，且存储单元是并列排布，可以实现一次性的直接寻址。另外，由于NOR的数据线宽度很大，即使容量增大，它的数据寻址时间基本上是一个常量，而NAND则比较困难。
- 在数据传输速度上，NOR无论是在随机读取还是连续传输速度上都比NAND快，但相对而言，在连续大数据传输速度上，二者差异较小。
- 从产品成本来看，NOR Flash的平均每MB成本是NAND Flash成本的三到四倍。

Flash Disk

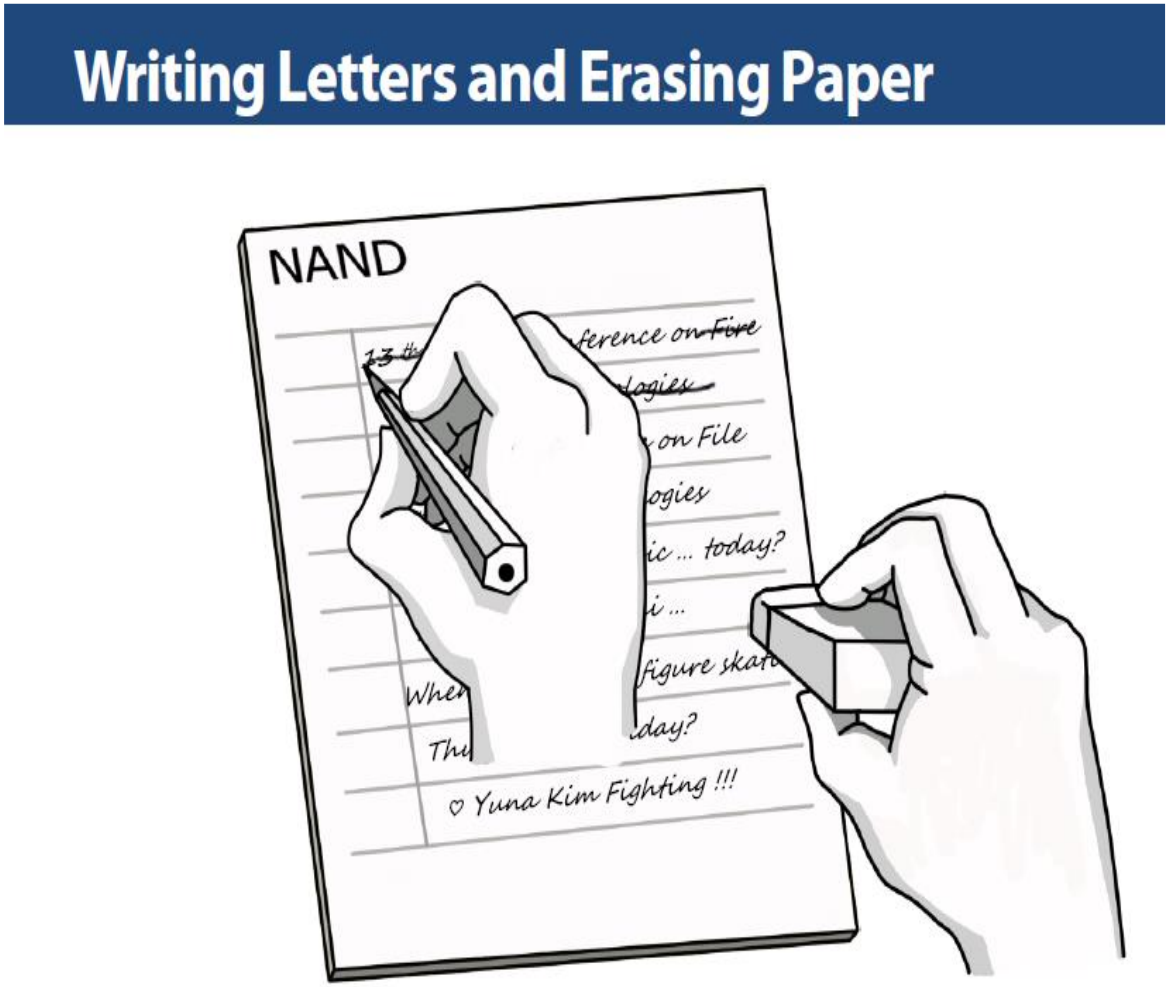
- Flash Disk的特征是低功耗、大容量、数据访问速度高。
  - 没有高速的机械部件，可以在剧烈振动的环境中工作;零下45摄氏度到零上85摄氏度
- Flash Disk执行一次数据读写的周期比硬盘短。
  - 即使是较慢的NAND型Flash Memory作为存储载体的Flash Disk执行一次数据访问的周期大致为15-30μs，对比硬盘的5ms。
- 问题：容量、可靠性。

writes 10x reads,
erasure 10x writes



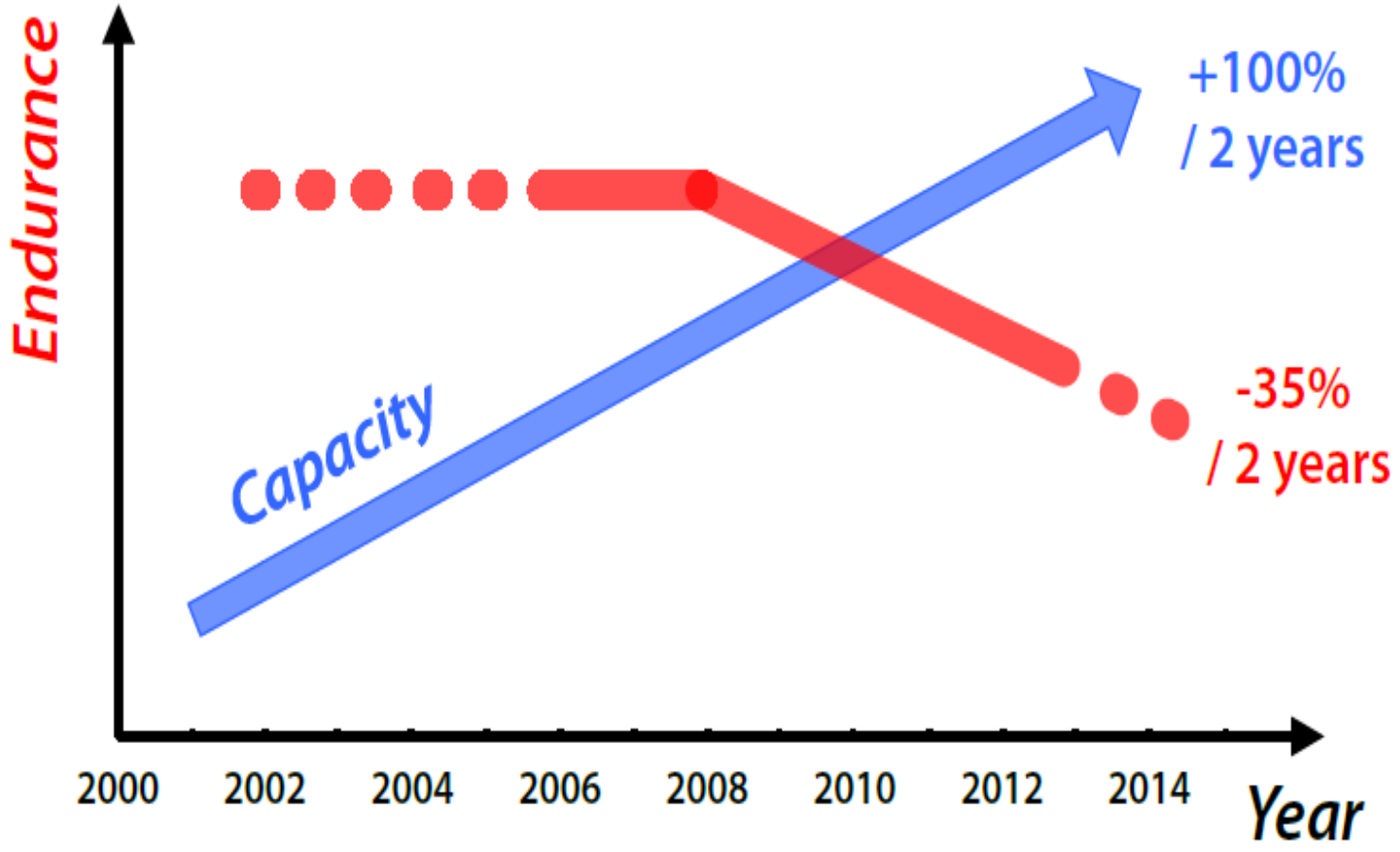
Typical NAND Flash Pages and Blocks

Flash擦除寿命问题



Jaeyong Jeong *et.al.* Lifetime Improvement of NAND Flash-based Storage Systems Using Dynamic Program and Erase Scaling, FAST 2014

Trend of NAND Device Technologies

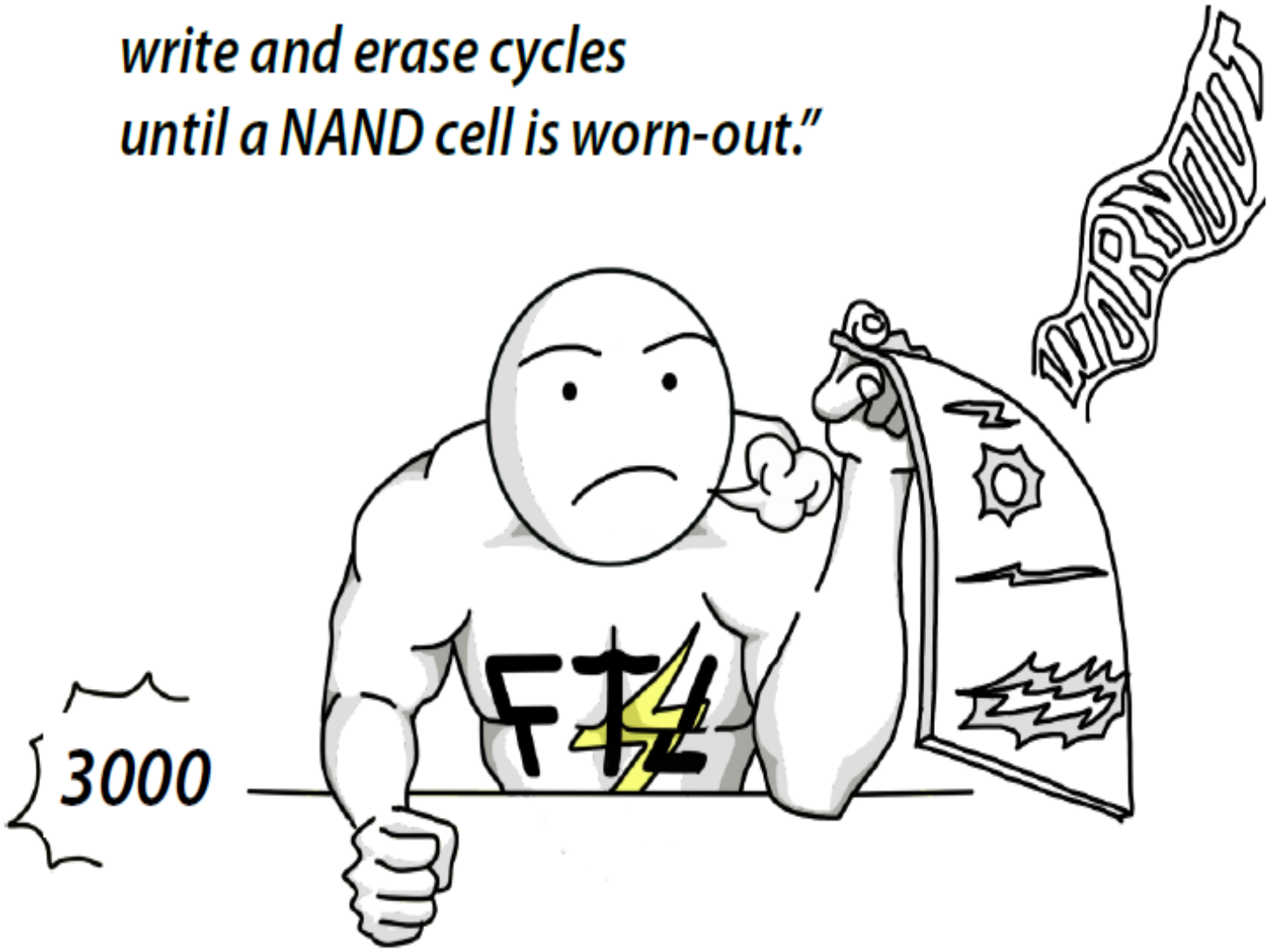


But, NAND endurance is drastically decreased last 6 years.

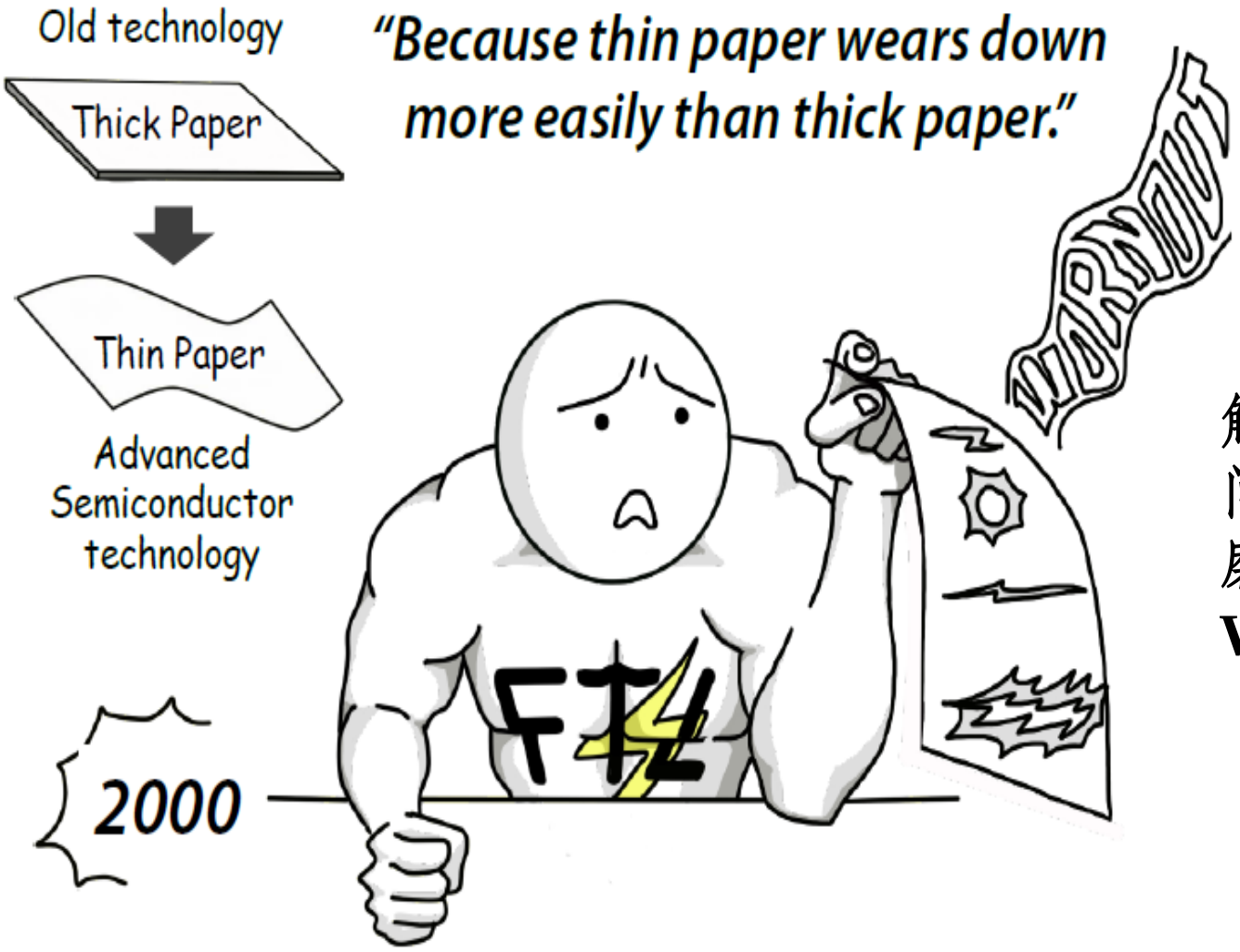
北

What is the NAND Endurance ?

"The maximum number of write and erase cycles until a NAND cell is worn-out."



Why is the NAND Endurance Decreased ?



解决擦除寿命问题的办法：
磨损均衡化
Wear Leveling

思考

- 磁盘调度算法对于Flash Disk还适用么？为什么？

RAID

廉价冗余磁盘阵列(Redundant Arrays of Inexpensive Disks, RAID)

- 1988 年美国加州大学伯克利分校的 D. A. Patterson 教授等首次在论文中提出了 RAID 概念 。
- 磁盘厂商将RAID称为独立硬盘冗余阵列（RAID, Redundant Array of Independent Disks）。
- 其基本思想就是把多个相对便宜的硬盘组合起来，成为一个硬盘阵列组，使性能达到甚至超过一个价格昂贵、容量巨大的硬盘。根据选择的版本不同，RAID比单颗硬盘有以下一个或多个方面的好处：增强数据集成度，增强容错功能，增加处理量或容量。另外，磁盘阵列对于电脑来说，看起来就像一个单独的硬盘或逻辑存储单元。

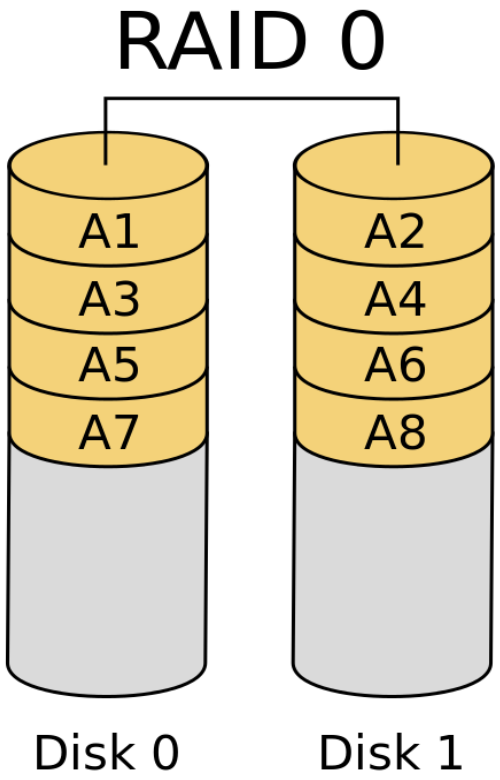
- 廉价冗余磁盘阵列(Redundant Arrays of Inexpensive Disks, RAID)
  - 具有N个磁盘的磁盘组中某些磁盘比特定磁盘具有较高的出错率。假设单个磁盘的平均损坏时间为100000小时，具有100个磁盘的阵列中某些磁盘的平均损坏时间为100000/100小时，大约是41.66天
  - 0.9*0.9*0.9*0.9*0.9=?

RAID

RAID 这种设计思想很快被业界接纳， RAID 技术作为高性能、高可靠的存储技术，已经得到了非常广泛的应用。 RAID 主要利用数据条带、镜像和数据校验技术来获取高性能、可靠性、容错能力和扩展性，根据运用或组合运用这三种技术的策略和架构，可以把 RAID 分为不同的等级，以满足不同数据应用的需求。

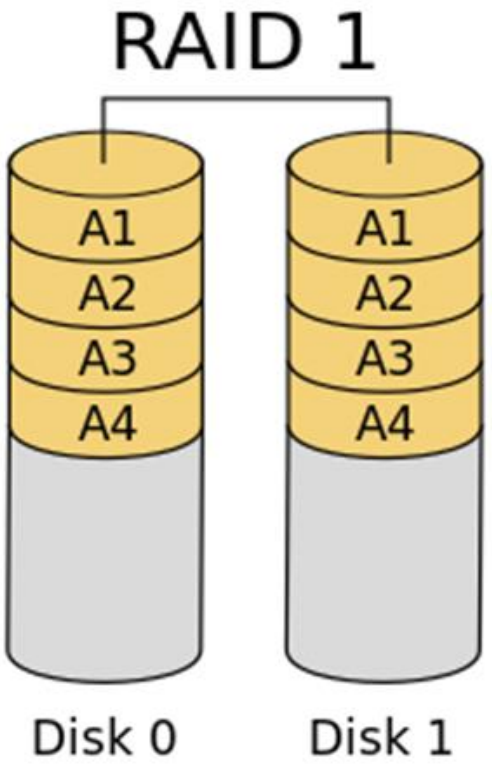
RAID0

- 条带化（Stripe）存储。理论上说，有N个磁盘组成的RAID0是单个磁盘读写速度的N倍。RAID 0连续以位或字节为单位分割数据，并行读/写于多个磁盘上，因此具有很高的数据传输率，但它没有数据冗余，因此并不能算是真正的RAID结构。



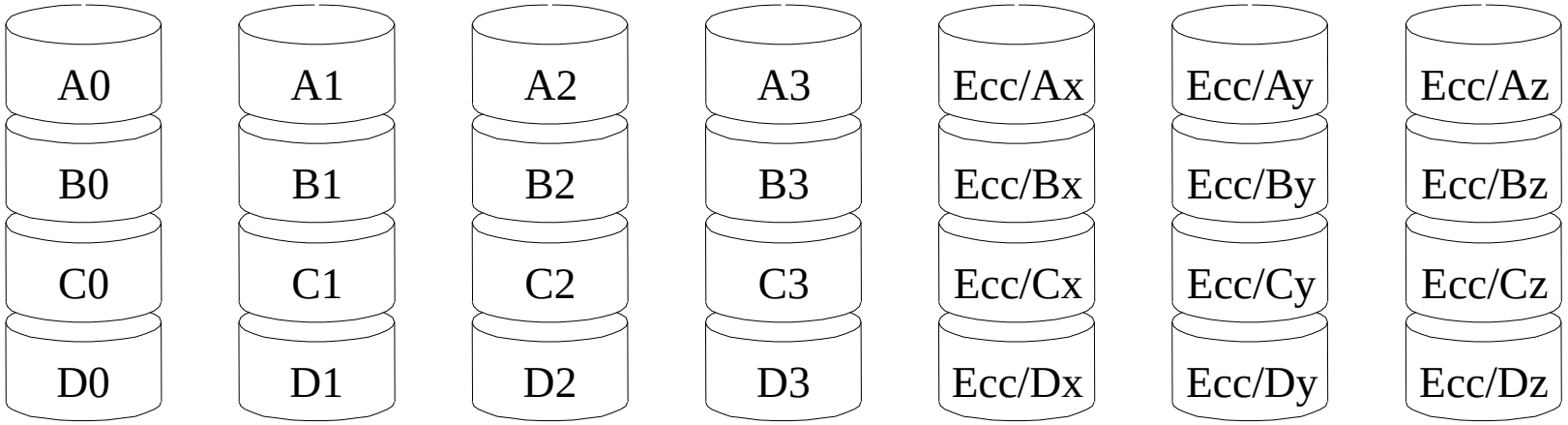
RAID1

- 镜象（Mirror）存储。它是通过磁盘数据镜像实现数据冗余，在成对的独立磁盘上产生互 为备份的数据。当原始数据繁忙时，可直接从镜像拷贝中读取数据，因此RAID 1可以提高读取性能。RAID 1是磁盘阵列中单位成本最高的，但提供了很高的数据安全性和可用性。当一个磁盘失效时，系统可以自动切换到镜像磁盘上读写，而不需要重组失效的数据。



RAID2

- 海明码（Hamming Code）校验条带存储。将数据条块化地分布于不同的硬盘上，条块单位为位或字节，使用称为海明码来提供错误检查及恢复。这种编码技术需要多个磁盘存放检查及恢复信息。



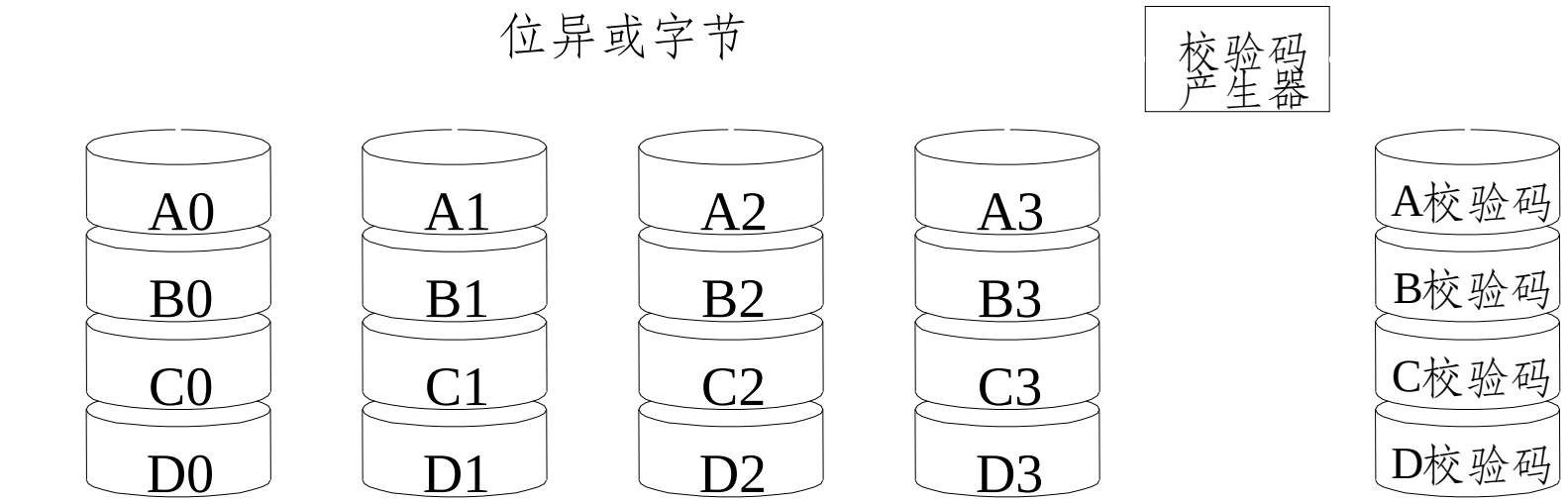
海明码长度：$2^P \geq P+D+1$

RAID2的特点

- 并行存取，各个驱动器同步工作。
- 使用海明编码来进行错误检测和纠正，数据传输率高。
- 需要多个磁盘来存放海明校验码信息，冗余磁盘数量与数据磁盘数量的对数成正比。
- 是一种在多磁盘易出错环境中的有效选择，并未被广泛应用，目前还没有商业化产品。

RAID3

奇偶校验（XOR）条带存储，共享校验盘，数据条带存储单位为字节。它同RAID 2非常类似，都是将数据条块化分布于不同的硬盘上，区别在于RAID 3使用简单的奇偶校验，并用单块磁盘存放奇偶校验信息。如果一块磁盘失效，奇偶盘及其他数据盘可以重新产生数据；如果奇偶盘失效则不影响数据使用。RAID 3对于大量的连续数据可提供很好的传输率，但对于随机数据来说，奇偶盘会成为写操作的瓶颈。

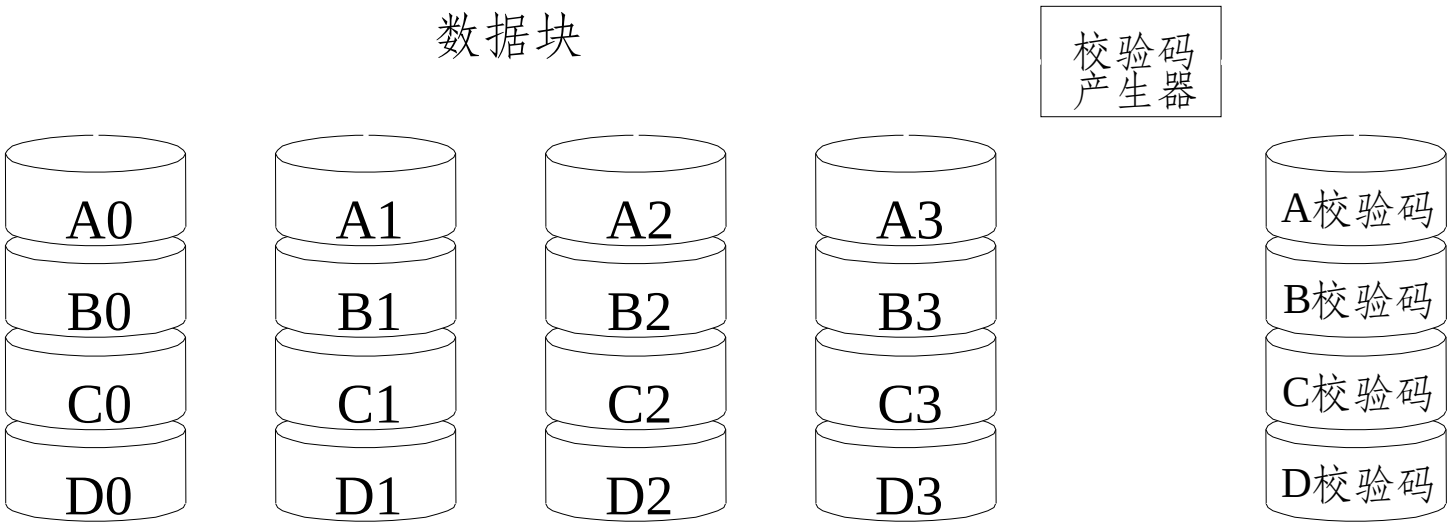


RAID3的特点

- 将磁盘分组，读写要访问组中所有盘，每组中有一个盘作为校验盘。
- 校验盘一般采用奇偶校验。
- 简单理解：先将分布在各个数据盘上的一组数据加起来，将和存放在冗余盘上。一旦某一个盘出错，只要将冗余盘上的和减去所有正确盘上的数据，得到的差就是出错的盘上的数据。
- 缺点：恢复时间较长。

RAID4

- 奇偶校验（XOR）条带存储，共享校验盘，数据条带存储单位为块。RAID 4同样也将数据条块化并分布于不同的磁盘上，但条块单位为块或记录。RAID 4使用一块磁盘作为奇偶校验盘，每次写操作都需要访问奇偶盘，这时奇偶校验盘会成为写操作的瓶颈，因此RAID 4在商业环境中也很少使用。

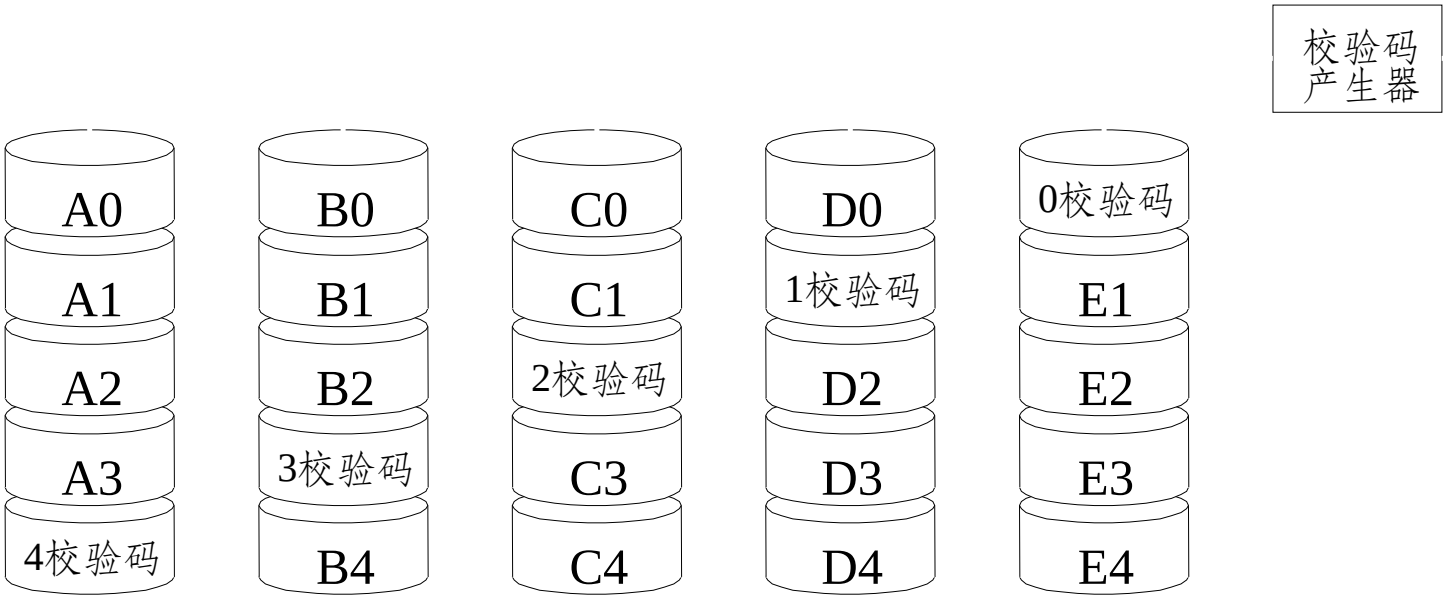


RAID4的特点

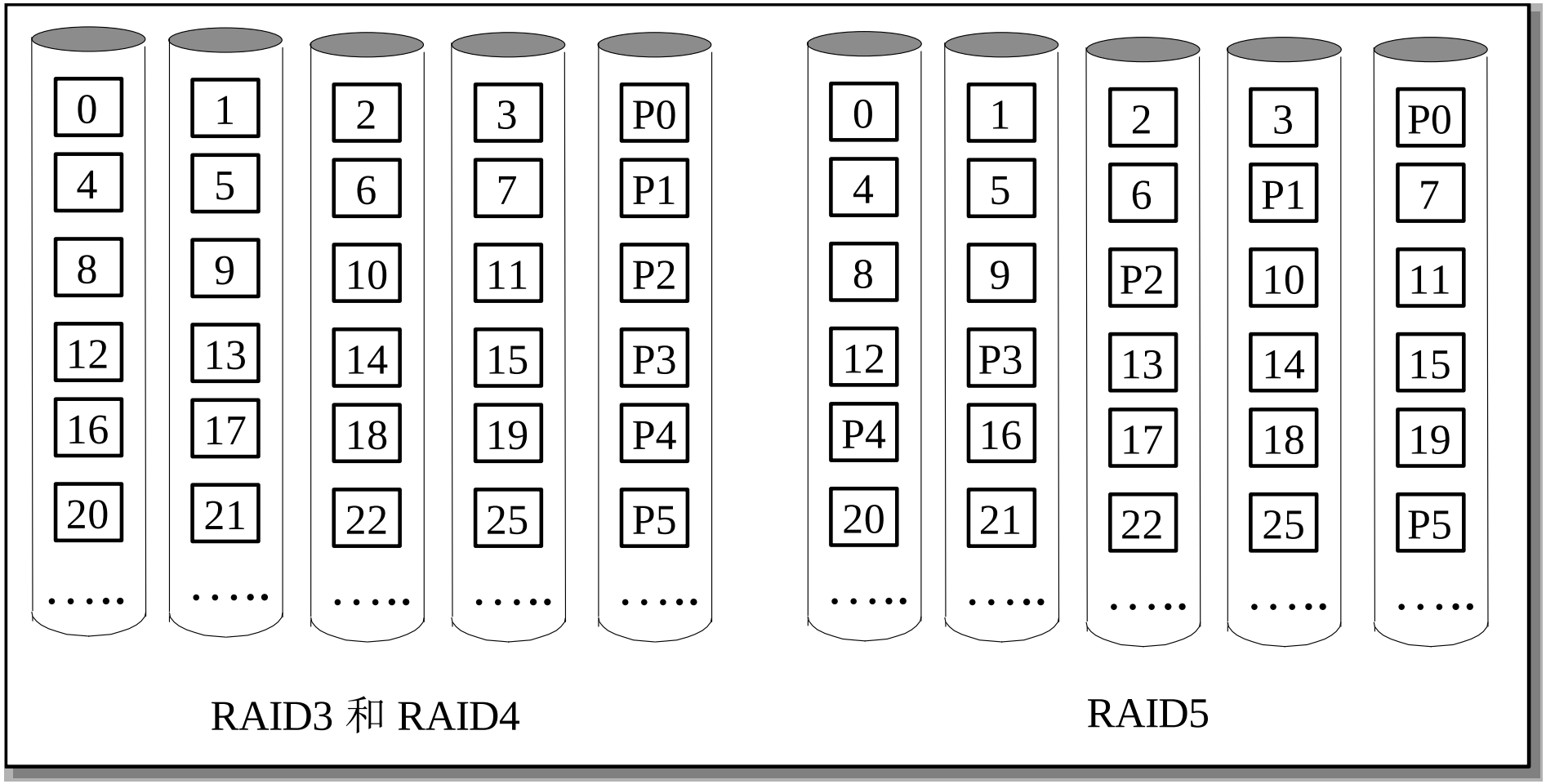
- 冗余代价与RAID3相同
- 访问数据的方法与RAID3不同
- 在RAID3中，一次磁盘访问将对磁盘阵列中的所有磁盘进行操作。
- RAID4出现的原因：希望使用较少的磁盘参与操作，以使磁盘阵列可以并行进行多个数据的磁盘操作。

RAID5

- 奇偶校验（XOR）条带存储，校验数据分布式存储，数据条带存储单位为块。RAID 5不单独指定的奇偶盘，而是在所有磁盘上交叉地存取数据及奇偶校验信息。在RAID 5上，读/写指针可同时对阵列设备进行操作，提供了更高的数据流量。



RAID3 RAID4和RAID5中的信息分布

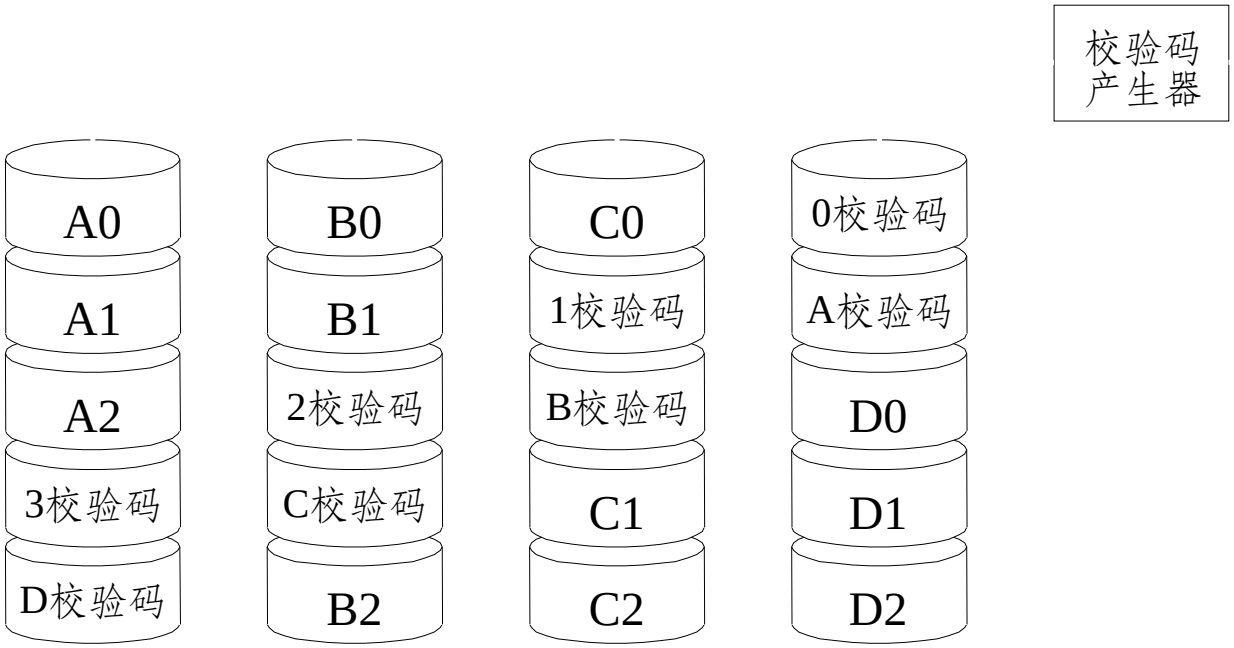


RAID5的特点

- RAID 5更适合于小数据块和随机读写的数据。
- RAID 3与RAID 5相比，最主要的区别在于RAID 3每进行一次数据传输就需涉及到所有的阵列盘；而对于RAID 5来说，大部分数据传输只对一块磁盘操作，并可进行并行操作。
- 在RAID 5中有“写损失”，即每一次写操作将产生四个实际的读/写操作，其中两次读旧的数据及奇偶信息，两次写新的数据及奇偶信息。
- 当进行恢复时，如我们需要恢复图中的A0，这里就必须需要B0、C0、D0加0 parity才能计算并得出A0，进行数据恢复。所以当有两块盘坏掉的时候，整个RAID的数据失效。

RAID6

- 奇偶校验（XOR）条带存储，两个分布式存储的校验数据，数据条带存储单位为块。与RAID 5相比，RAID 6增加了第二个独立的奇偶校验信息块。两个独立的奇偶系统使用不同的算法，数据的可靠性非常高，即使两块磁盘同时失效也不会影响数据的使用。



RAID6的特点

- RAID 6需要分配给奇偶校验信息更大的磁盘空间
- 写入数据要访问1个数据盘和2个冗余盘，相对于RAID 5有更大的“写损失”，因此“写性能”非常差；
- 可容忍双盘出错；
- 存储开销是RAID5的两倍（多一个冗余盘）。
- 较差的性能和复杂的实施方式使得RAID 6很少得到实际应用。

RAID比较

级别	说明	特点	磁盘数量	最少磁盘数量
0	条带化	速度加倍，可靠性减半，容量不变	n	2
1	镜像	速度不变，可靠性加倍，容量减半	n*2	2
0+1	组内条带化、组间镜像	速度加倍，可靠性加倍，容量减半	n*2	4
1+0	组内镜像，组间条带化	速度加倍，可靠性加倍，容量减半	2*n	4
2	海明码校验	校验独立存储，数据位交叉（条带化），自动纠正1个盘错	n+log(n)	7*
3	奇偶校验	校验独立存储，数据位交叉（条带化）容1块盘错	n+1	3
4	奇偶校验	校验独立存储，数据块交叉，容1块盘错	n+1	3
5	奇偶校验	校验交叉存储，数据块交叉，容1块盘错	n+1	3
6	奇偶校验	校验交叉存储，数据块交叉，容2块盘错	n+2	4

RAID小结

- 条带化：一个字节块可能存放在多个数据盘上
  - 优点：并行存取，性能好，磁盘负载均衡
  - 缺点：可靠性、不同IO请求需要排队
- 镜像：数据完全拷贝一份
  - 优点：可靠性
  - 缺点：存储开销
- 校验：数据通过某种运算（异或）得出，用以检验该组数字的正确性
  - 优点：可靠性，快速恢复
  - 缺点：开销

内容提要

- 磁盘的历史及工作原理
- 磁盘的组织与调度算法
- 磁盘空间的管理
- RAID
- 提高I/O速度
- 磁盘管理实例

提高I/O速度的主要途径

- 选择性能好的磁盘
- 并行化
- 采用适当的调度算法
- 设置磁盘高速缓冲区

提高磁盘I/O速度——缓存

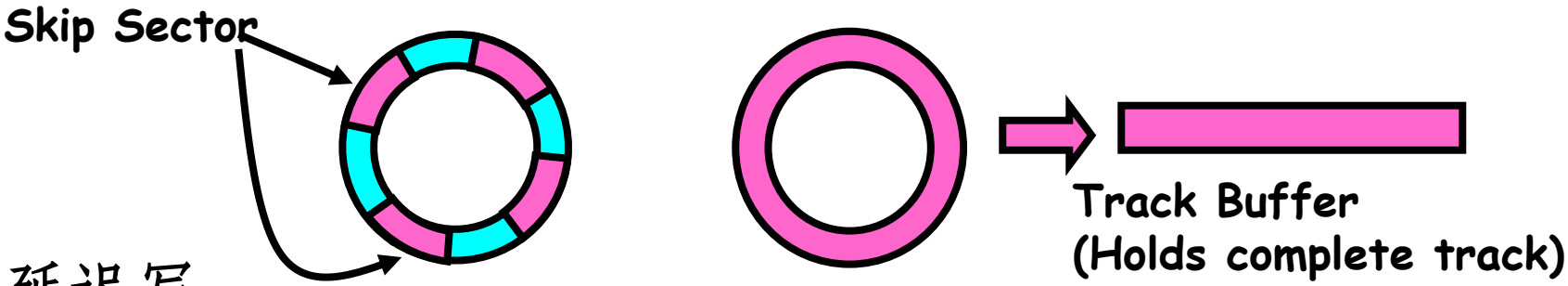
- 磁盘高速缓存的形式
  - 独立缓存（大小固定）
  - 虚拟内存为缓存（弹性）
- 数据交付
  - 直接交付（copy开销）
  - 指针交付（内存管理复杂）
- 置换算法（LRU）
- 周期性写回
  - 周期性地将disk cache中被修改过的内容写回磁盘

优化数据布局

- 优化物理块的分布
  - 连续摆放
- 优化索引节点的分布
  - 减少与数据块的距离
  - 与数据块结合

提高磁盘I/O速度的其他方法

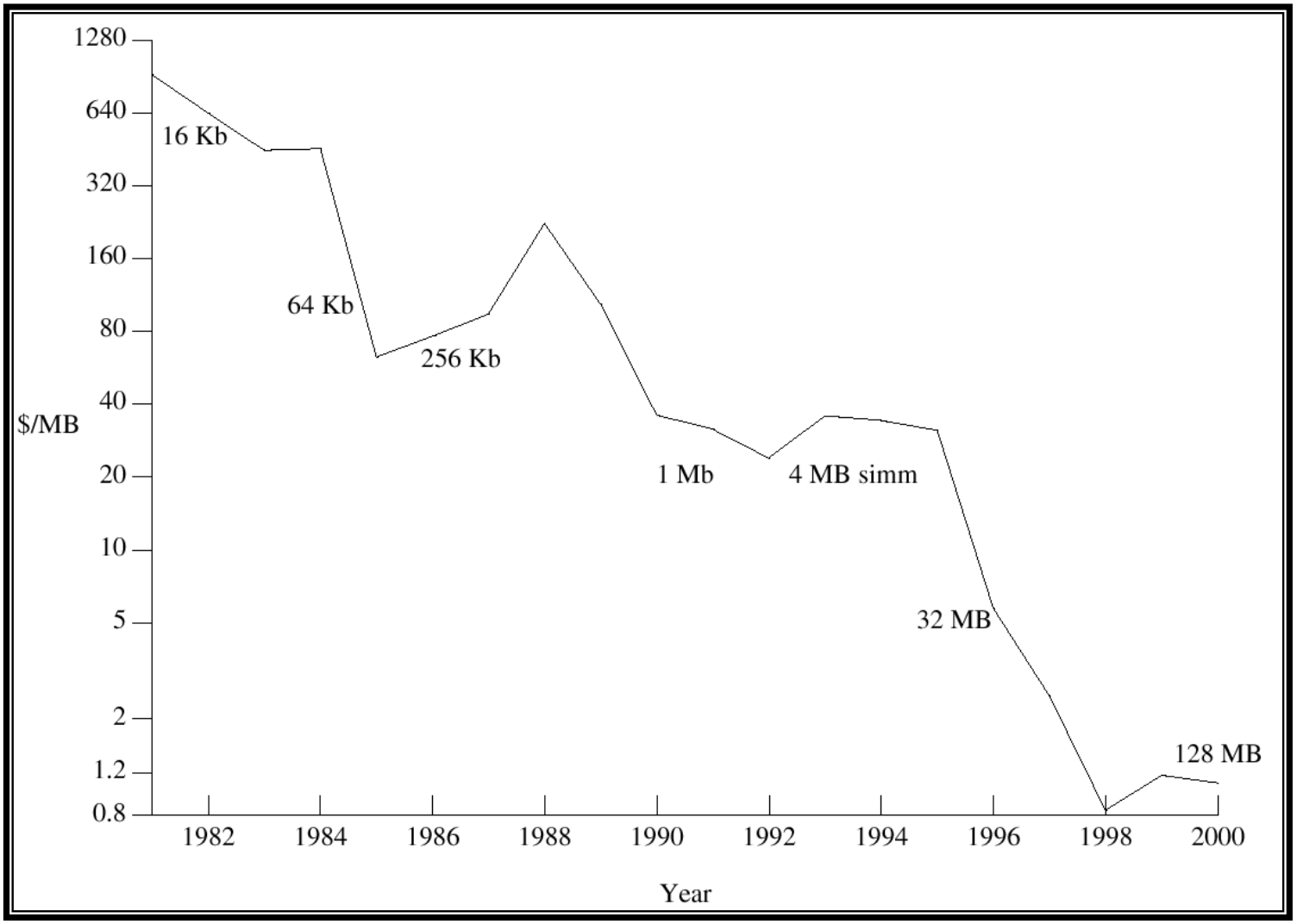
- 提前读
  - 顺序访问时,常采用提前读入下一块到缓冲区中



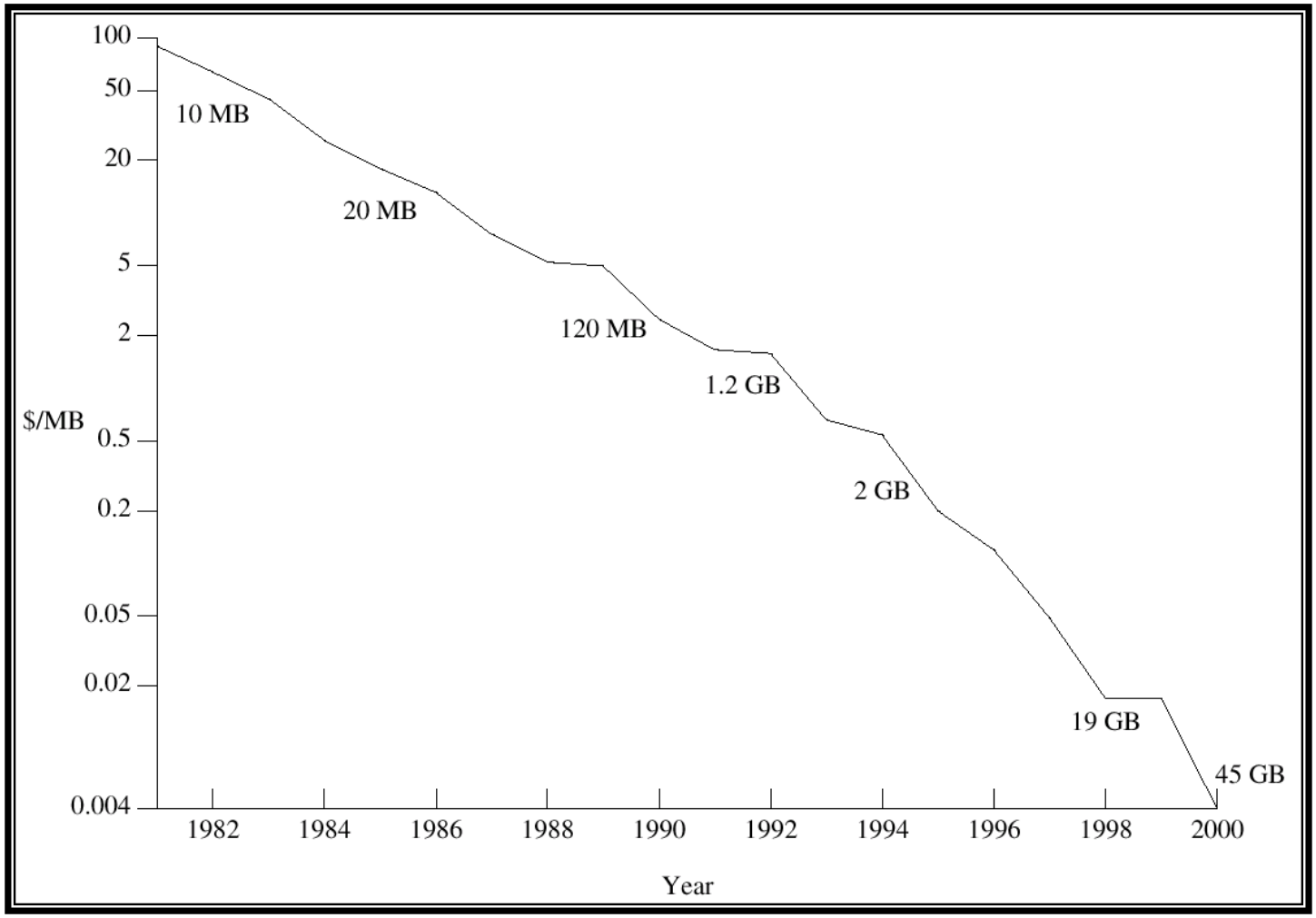
- 延迟写
  - 将本应立即写回磁盘的数据挂到空闲缓冲区的队列的末尾。直到该数据块移到链头时才将其写回磁盘，再作为空闲区分配出去（可靠性？）

- 虚拟盘
  - 利用内存空间去仿真磁盘（RAM盘）
  - Virtual disk 与disk cache的区别是：
    - Virtual disk的存放的内容由用户完全控制
    - Disk cache中的内容完全是由操作系统控制

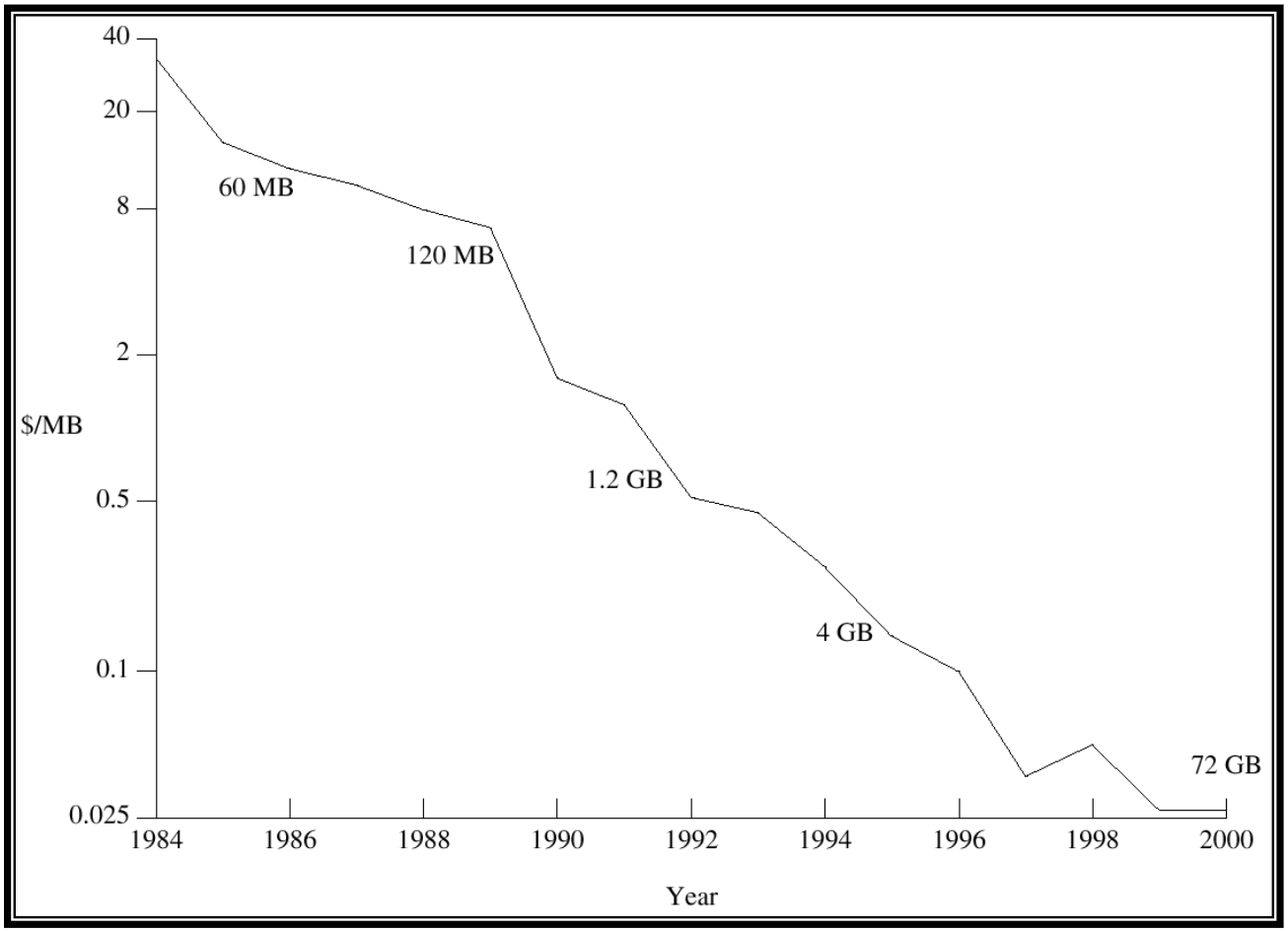
Price per Megabyte of DRAM, From 1981 to 2000



Price per Megabyte of Magnetic Hard Disk, From 1981 to 2000



Price per Megabyte of a Tape Drive, From 1984-2000



内容提要

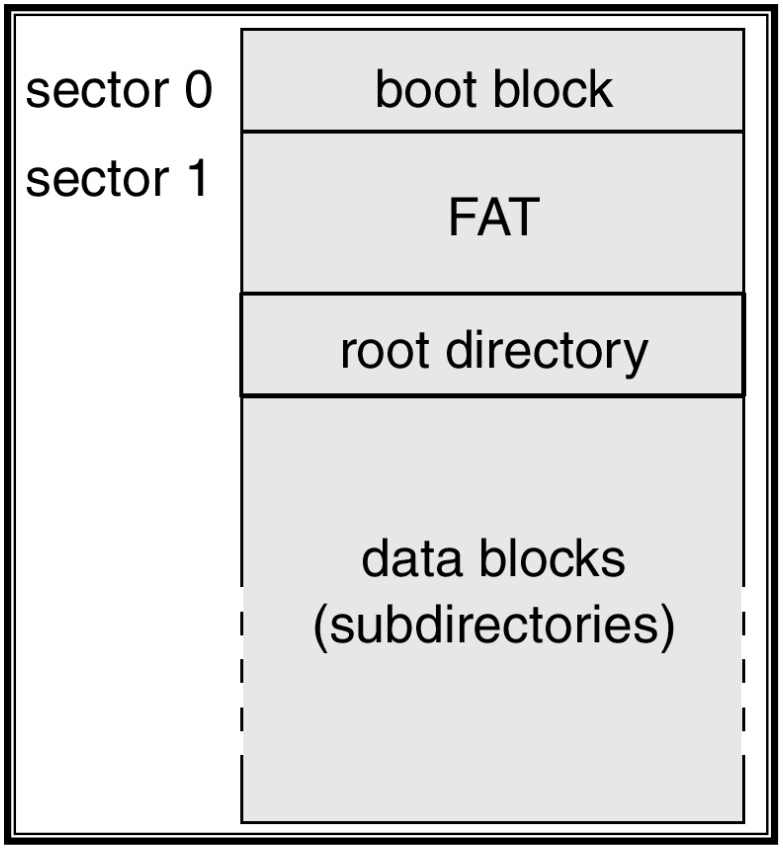
- 磁盘的历史及工作原理
- 磁盘的组织与调度算法
- 磁盘空间的管理
- RAID
- 提高I/O速度
- 磁盘管理实例

Windows2000磁盘管理系统

- 跨磁盘管理（DISK SPANNING）
- 容错（FAULT TOLERANCE）

MS—DOS的分区方式缺点

- 对大多磁盘设置的改变需要重起操作系统才能生效。
- WINDOWS NT的注册表中为MS—DOS方式的分区保存了多分区盘的配置信息
- 每个卷有一个唯一的从A到Z的驱动器名



基本术语

- 盘（DISKS）一种物理存储设备，如硬盘，3.5英寸软盘，光盘。
- 盘被分为扇区（SECTOR）,可寻址的大小固定的块。扇区的大小是由硬件决定的。所有现在的X86处理器的硬盘扇区大小为512字节。光盘的扇区一般为2048字节。未来的X86系统可能支持更大的硬盘扇区。
- 分区是盘上连续扇区的集合，分区表或其他盘管理数据库中保存了分区的起始扇区，大小，和其他特性。
- 简单卷（SIMPLE VOLUME）是这样一种对象，他代表文件系统驱动程序所作为一个独立单元所管理的，来自一个分区的扇区。
- 多分区卷（MULTIPATITION VOLUME）是这样一种对象，他代表文件系统驱动程序所作为一个独立单元所管理的，来自多个分取得扇区。多分区卷提供简单卷所不支持的更好的性能，可靠性，大小的特性。

基本盘和动态盘的概念

- WINDOWS 2000 把基于MS—DOS分区方式的盘称为基本盘。
- 动态盘对WINDOWS 2000来说是一个新的概念，他实现了比基本盘更具适应性的分区机制。

基本盘和动态盘之间不同点

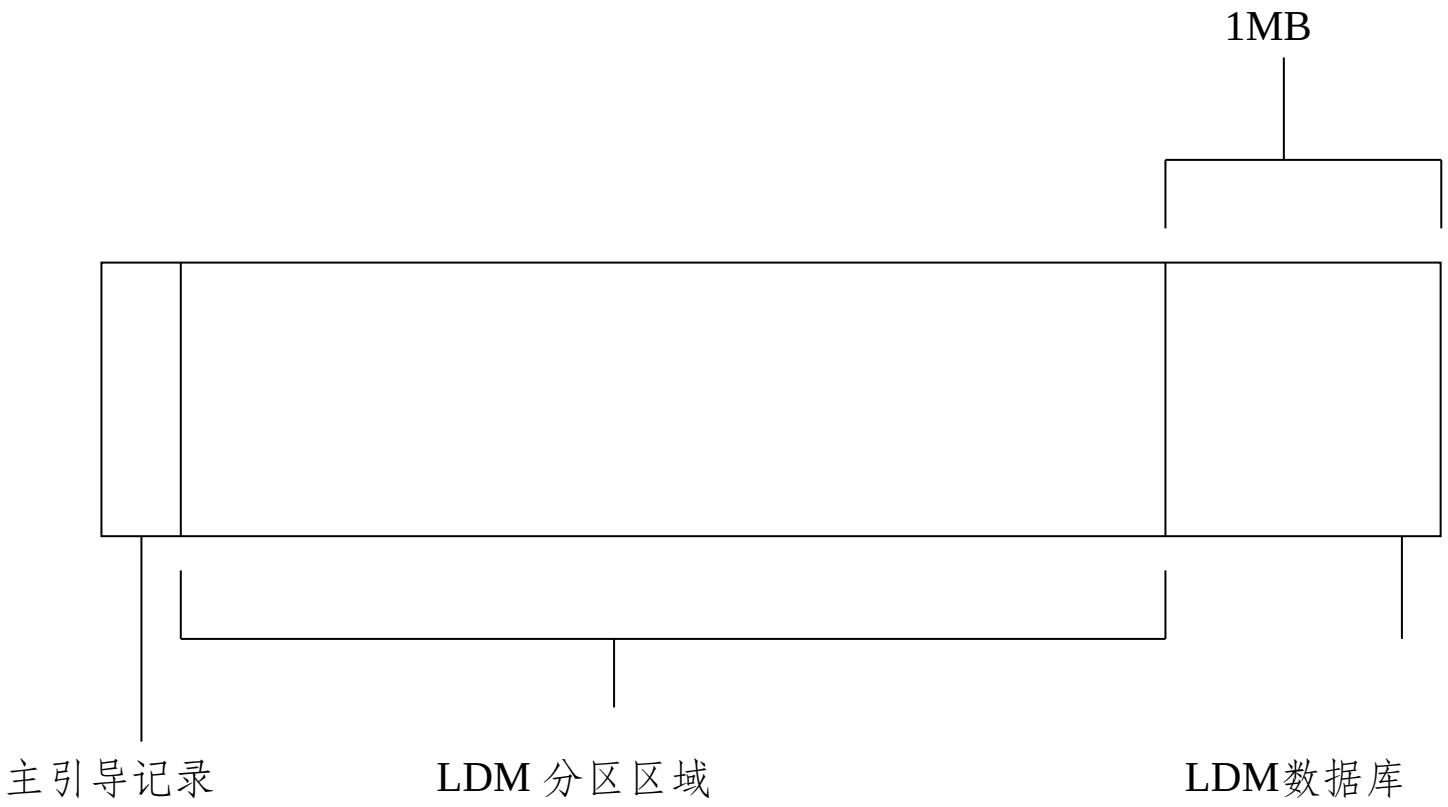
- 动态盘支持创建新的多分区卷。
- 基本盘的多分区卷的配置信息保存在注册表中。动态盘的多分区卷的配置信息保存在磁盘中。

动态盘的缺点

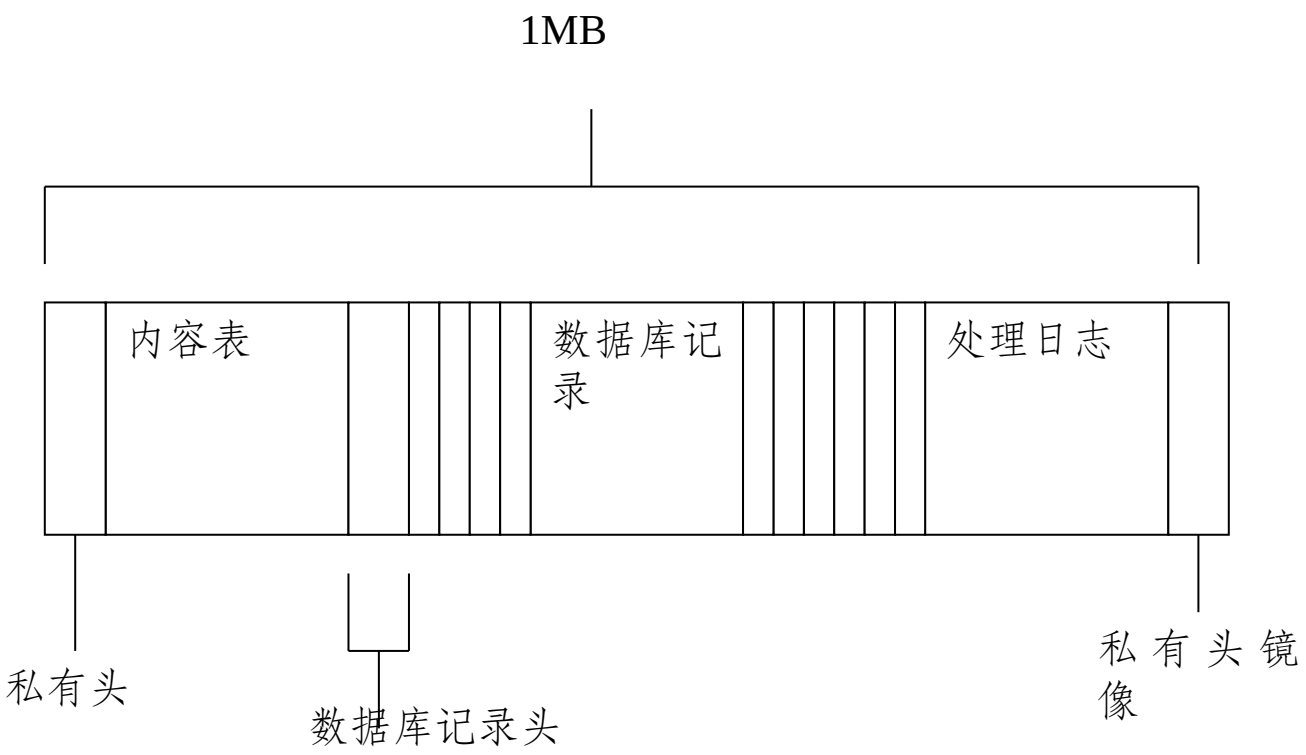
- 它所应用的分区的格式是专有的，并且不和其它的操作系统兼容

- 系统卷是WINDOWS2000存放引导文件的地方，包括引导程序（NTLDR）和NTDETECT，
- 引导卷是WINDOWS2000存放系统文件，如NTOSKRNL.EXE核心内核文件的地方

- LDM的分区在一个盘的MSDOS分区表中并没有体现出来，所以被称为软分区，而MSDOS分区被称为硬分区。



动态盘的内部组织


1MB
内容表
数据库记录
处理日志
私有头
数据库记录头
私有头镜像

数据库结构

- 私有头：GUID，磁盘组的名字（该名字是由Dg0和计算机的名字一起组成，例如SusanDg0，意味着计算机的名字是Susan）和一个指向数据库内容表的指针。为了保证可靠性，LDM在磁盘的最后一个扇区保存了私有头的拷贝。
- 数据库内容表有16个扇区大小，其中包含关于数据库布局的信息。
- 数据库记录区域紧接着内容表，并将内容表后第一个扇区作为数据库记录头。这个扇区中存储了数据库记录区的信息，包括其所包含的记录个数，数据库相关的磁盘组的名字和GUID，以及LDM用于创建下一项的序列号。

- 数据库中的每一项可以是如下四种类型之一：分区，磁盘，组件，卷。LDM把每一项与内部对象的标识符联系到一起。在最低的级别，分区项描述软分区，它是在一个盘上的连续区域。存储在分区项中的标识符把这个项与一个组件和一个磁盘项联系起来。磁盘项代表一个磁盘组中的动态盘，包括磁盘的GUID。组件项像一条链子把一个或多个分区项和与分区相连的卷项联系起来。卷项存放这个卷的GUID，卷的大小和状态，驱动器的名字。比一个数据库记录大的磁盘项占用多个记录的空间，分区项、组件项和卷项很少占用多个记录的空间。

- LDM需要三个项来描述一个简单卷：分区项、组件项和卷项。分区项描述系统分配给某个卷的磁盘上的一个区域，组件项把一个分区项和一个卷项联系起来，卷项中包含Windows Server 2003内部用来识别卷的GUID。多分区卷需要的项数多于三个。例如，一个条带卷包括最少两个分区项，一个组件项和一个卷项。唯一一种含有一个以上组件项的卷的类型是：镜像卷。镜像卷含有两个组件项，每个只表示这个镜像的一半。LDM为每个镜像卷使用两个组件项的目的是：当一个镜像破坏时LDM能够在组件一级将他们分割开来，并创建两个各含有一个组件项的卷。因为简单卷需要三个项，而1MB数据库空间大约可以容纳8000个项，所以在Windows Server 2003中可以创建的卷数目的有效上界大约是2500个。

- LDM数据库的最后部分是事务处理日志区，它包含的几个扇区在数据库信息改变时用来存储备份信息。这样确保在系统崩溃或断电时，LDM能够利用日志把系统恢复到一个正确的状态。

驱动程序

- 系统卷中引导扇区中的代码负责执行**Ntldr**。
- **Ntldr**从系统卷中读取**Boot.ini**文件，把计算机的引导选项显示给用户。**Boot.ini**指定分区名为**mult(0)disk(0)rdisk(0)partition(1)**的形式。
- **Ntldr**把**Boot.ini**中用户指定的项转换为正确的引导分区，然后将**Windows Server 2003**系统文件（从注册表、**Ntoskrnl.exe**、引导驱动程序开始）装入内存，继续引导过程。

- **Windows Server 2003的存储驱动程序**
  - 类：实现所有存储设备共同的功能
  - 端口：基于某种特定总线设备的共同功能，如**SCSI、IDE**
  - 小端口：**OEM**提供

- 磁盘的类驱动程序使用I/O管理器的IoReadPartitionTable函数识别表示分区的设备对象
- 设备名
  - **\Device\Harddisk0\DP(1)0x7e000-0x7ff50c00+2**

- **Windows Server 2003**保存了两个不同的名字空间子目录供**Win32**使用，其中之一是：**\??**子目录（另一个是**\BaseNamedObjects**子目录）。
- 在**\??**子目录中，**Windows Server 2003**创建了一些与**Win32**程序交互的硬件对象，包括串口和并口，还有磁盘。

- 由于磁盘对象实际上存在于其它的子目录中，所以Windows Server 2003使用符号链接，把在\??子目录下的名字与在名字空间其它地方的对象联系起来。I/O管理器为系统中的每一个物理盘都创建一个\??\PhysicalDriveX的链接，指向\Device\HarddiskX\Partition0（从零开始的数字来替代X）。
- 那些直接访问磁盘扇区的WIN 32应用程序可以调用Win32函数CreateFile，通过指定\\.\PhysicalDriveX（X是一个磁盘的号码）作为参数来打开磁盘。Win32 的应用层先把名字转化为\??\PhysicalDriveX，然后在把名字提交给Windows Server 2003对象管理器。

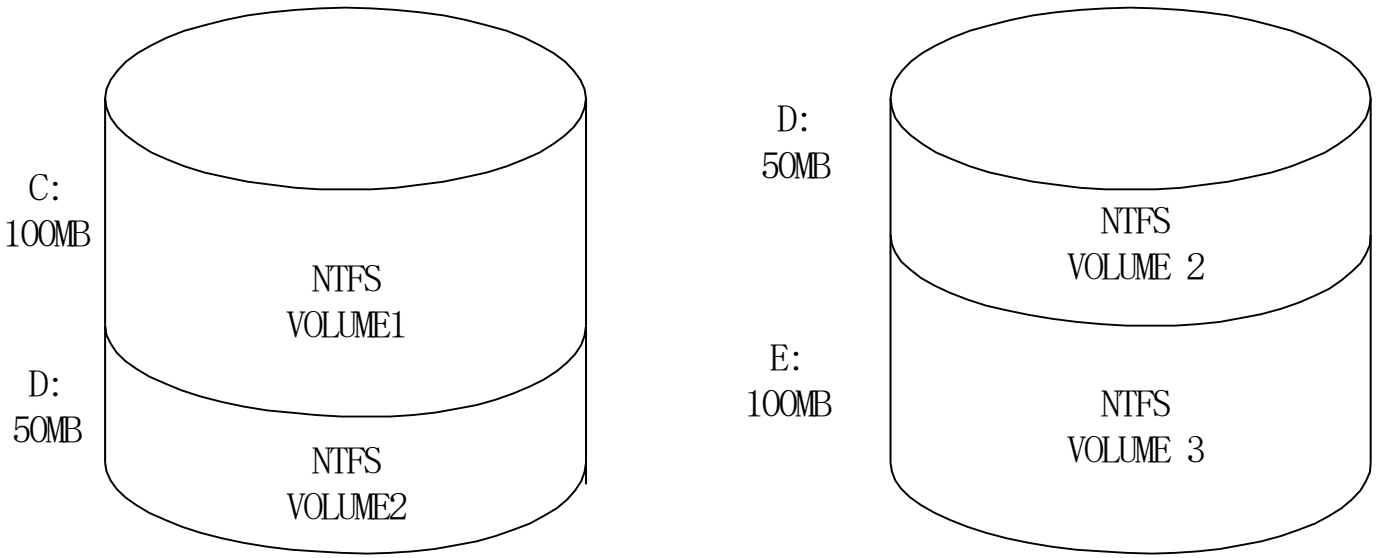
管理工具

- FtDisk和 DMIO负责识别文件系统驱动程序管理的卷，并将I/O直接从卷映射到组成卷的底层分区。
- 对简单卷来说，通过把卷的偏移量加上卷在磁盘中的起始地址，卷管理器可以保证卷的偏移量被转换成盘的偏移量。
- 对于多分区卷这就复杂多了，因为组成卷的分区可以是不邻接的分区，甚至可以在不同的磁盘中。有一些多分区卷使用数据冗余技术，所以它们需要更多的卷到磁盘的转换工作。

多重分区管理

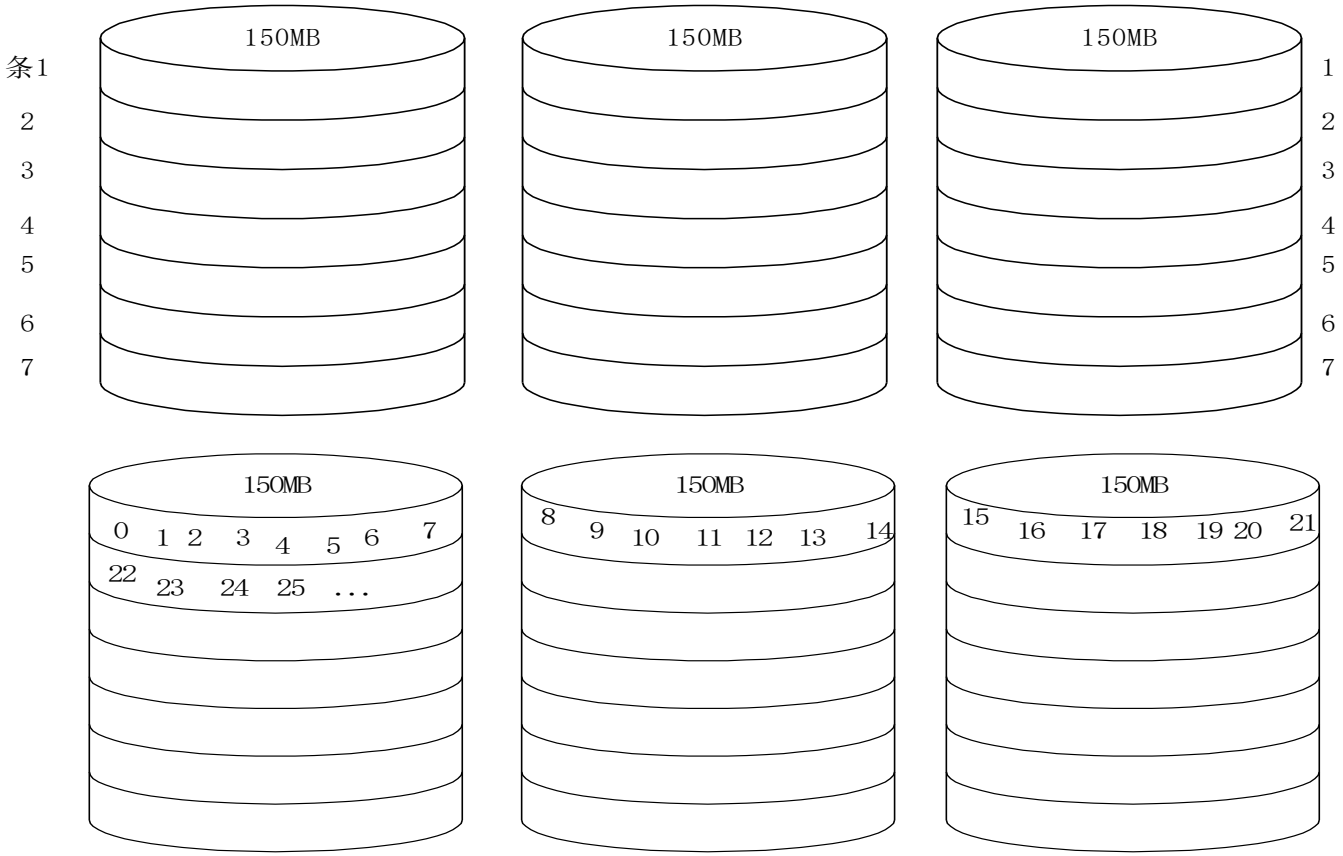
- 跨分区卷(spanned volume)
- 条带卷（striped volume）
- 镜像卷（mirrored volume）
- 廉价冗余磁盘阵列5卷（RAID-5 volume）

跨分区卷


C:
100MB
NTFS
VOLUME1
D:
50MB
NTFS
VOLUME2
D:
50MB
NTFS
VOLUME 2
E:
100MB
NTFS
VOLUME 3

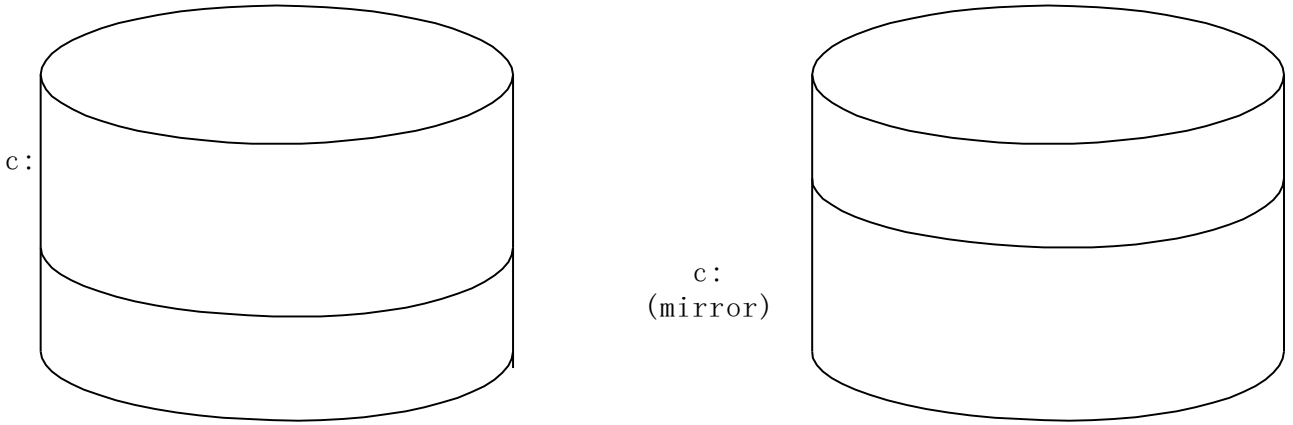
- 一个单独的逻辑卷，最多由在一个或多个磁盘上的32个空闲分区组成。
- 跨分区卷可以用来把小的磁盘空闲区域，或者把两个或更多的小磁盘组成大的卷。
- 卷管理器对Windows Server 2003的文件系统隐藏了磁盘物理配置信息。

条带卷（RAID-0卷）


150MB
150MB
150MB
条1
2
3
4
5
6
7
1
2
3
4
5
6
7
150MB
0 1 2 3 4 5 6 7
22 23 24 25 ...
150MB
8 9 10 11 12 13 14
150MB
15 16 17 18 19 20 21

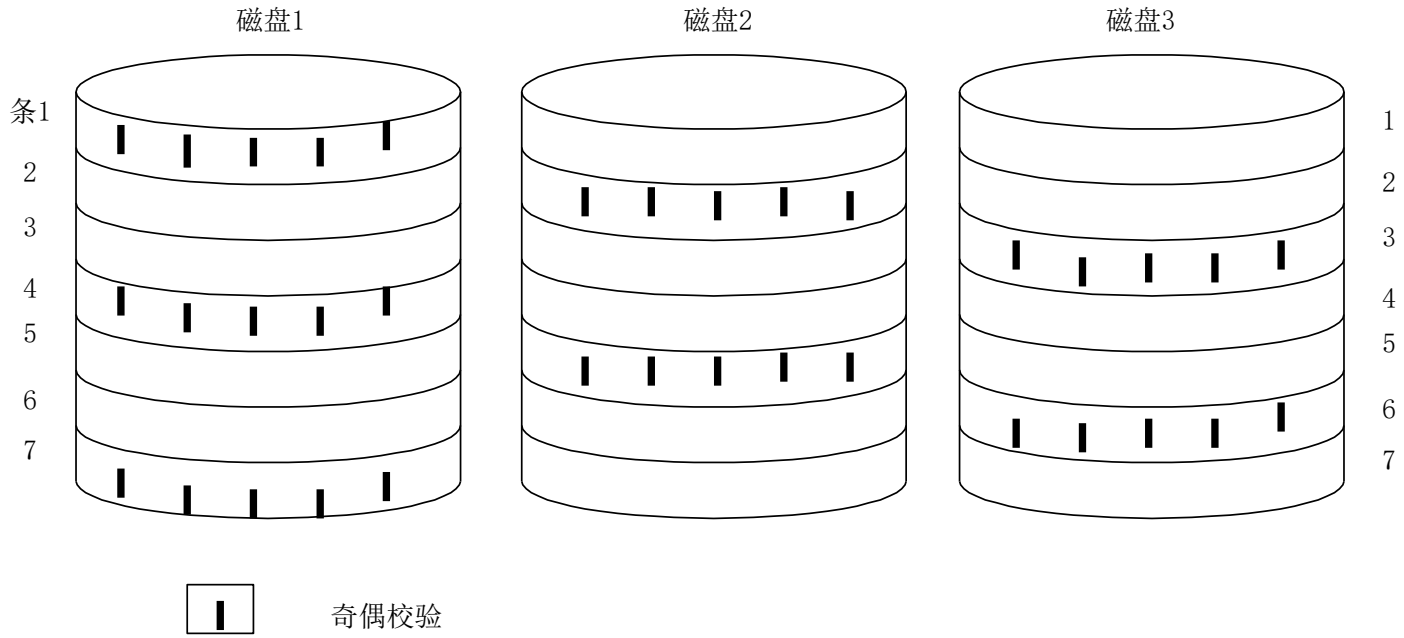
- 一系列分区组成的单独的逻辑卷，最多有32个分区并且每个盘一个分区。
- 条带卷中的一个分区不需要占据整个磁盘，唯一的限制是每个盘上的分区大小相同
- 数据能够被平均分配到每个磁盘上

镜像卷



- 一个磁盘上分区的内容被复制另一个磁盘与它等大小的分区中。镜像卷有时也被称为RAID-1。
- 镜像卷能够可以在主分区和镜像分区之间平衡I/O操作。两个读操作可以同时进行，所以理论上只用一半时间就可以完成。当修改一个文件时，必须写入镜像卷的两个分区，但是磁盘写操作可以异步进行，所以用户态程序的性能一般不会被这种额外的磁盘更新所影响。
- 镜像卷是唯一一种支持系统卷和引导卷的多分区卷。

廉价冗余磁盘阵列5卷



卷名字空间

- 安装管理器
- 安装点
- 卷安装

安装管理器

- 安装管理器（Mountmgr.sys）是Windows Server 2003中新驱动程序，为在Windows Server 2003安装后创建的动态磁盘卷和基本磁盘卷分配驱动器名。
- 卷管理器创建卷时都将通知它。当接到通知时，确定新的卷GUID或者磁盘标记；
- 安装管理器使用卷GUID（或者标识）在内部数据库中进行查询
- 安装管理者使用第一个未分配的驱动器名，为这次分配创建一个符号链接（例如，\??\D:）

安装点

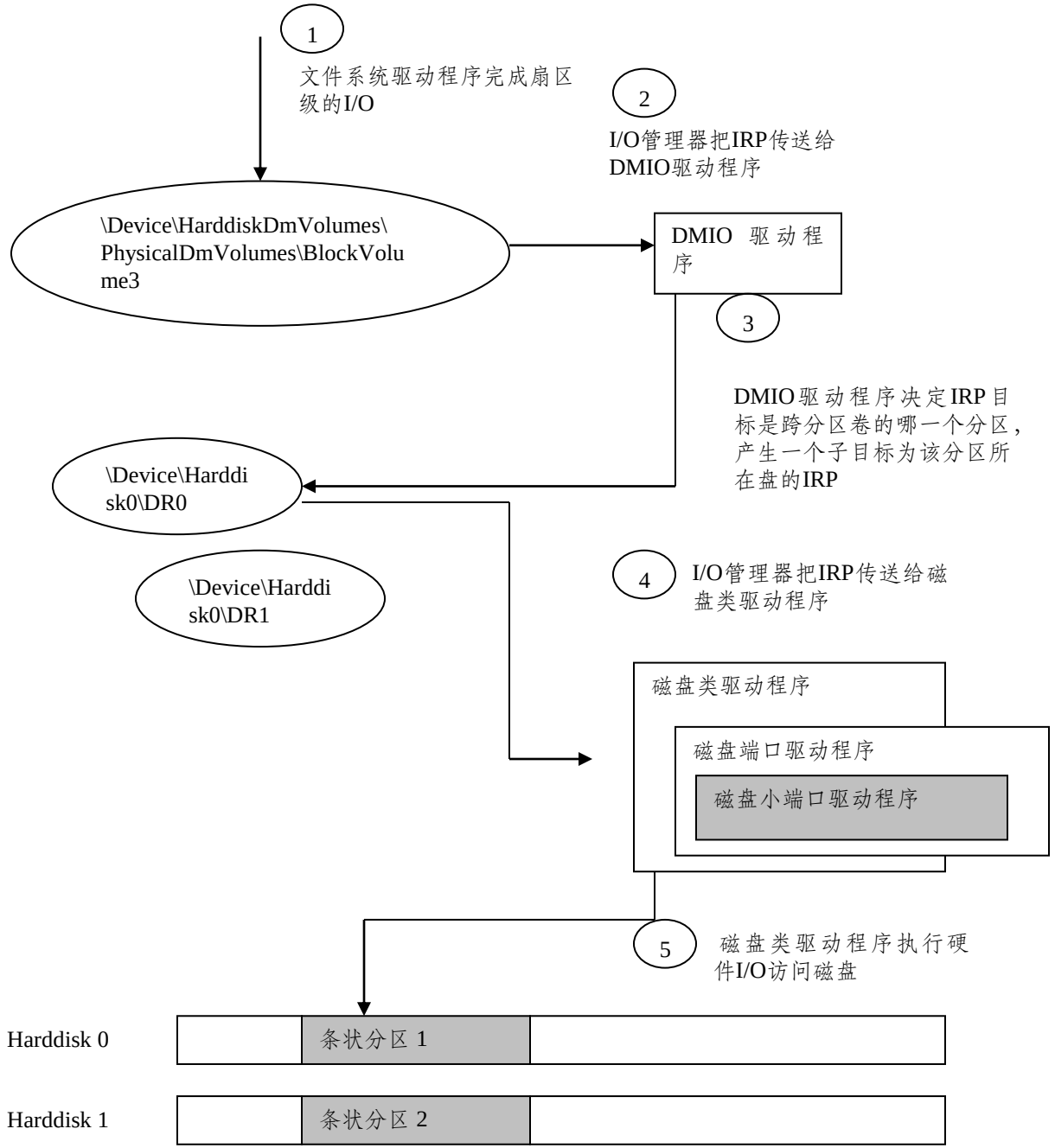
- 实现安装点的技术是再解析点(Reparse Point)技术。
- C:\Project　　CurrentProject\Description.txt
- C:\Projects\CurrentProject\Description.txt

卷安装

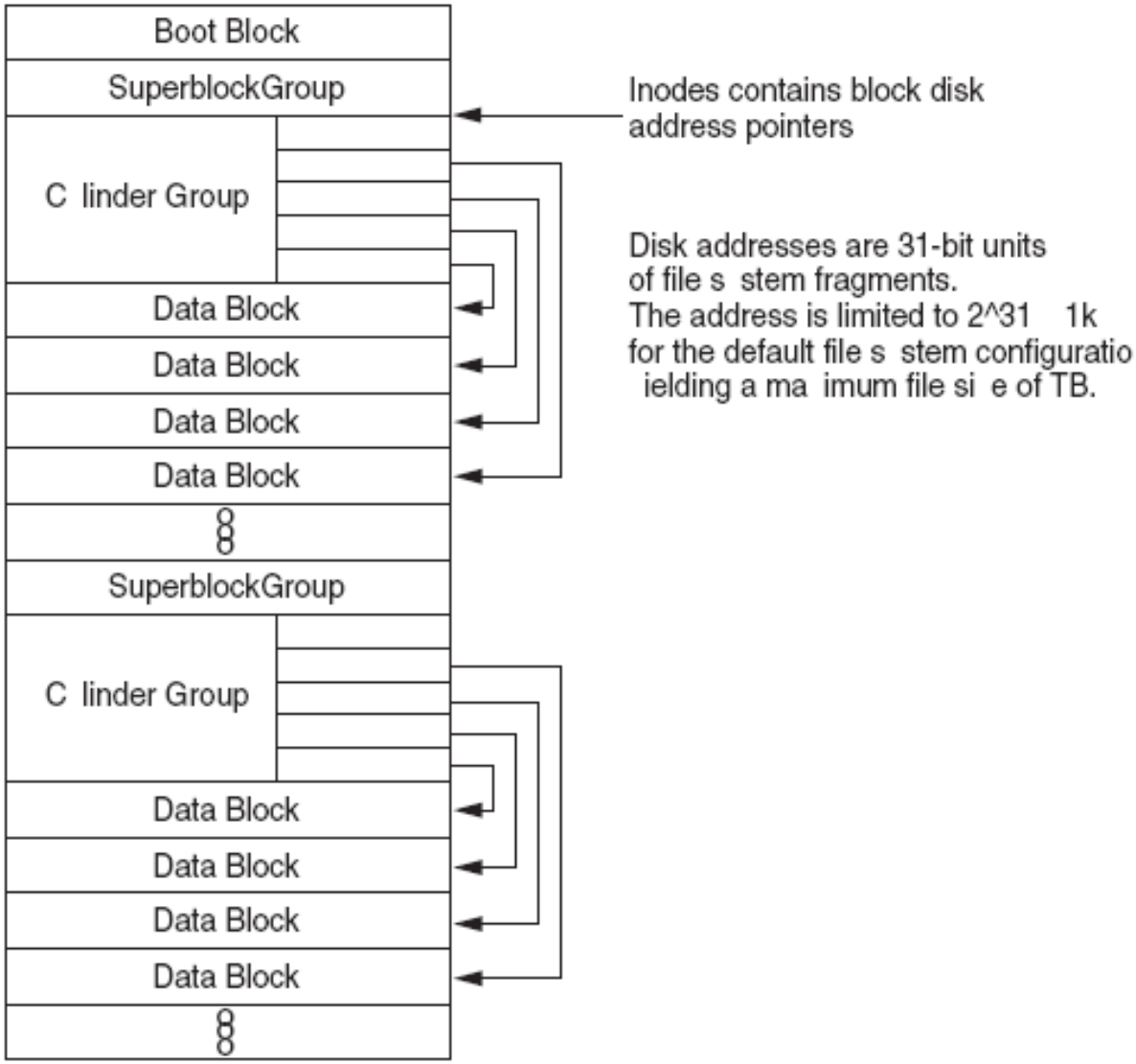
- 每个文件系统驱动程序在初始化时都向I/O管理器注册
- 每个设备对象包含一个卷参数块VPB，但是I/O管理器认为只有卷设备对象的VPB是有意义的
- 安装请求

- 安装管理器把D:分配给系统中的第二个卷
- 它产生符号连接\??\D:指向设备对象\Device\HarddiskVolume2。
- 一个WIN 32应用程序试图打开D:上的文件\Temp\Test.txt时，它将会指定路径D:\Temp\Test.txt。

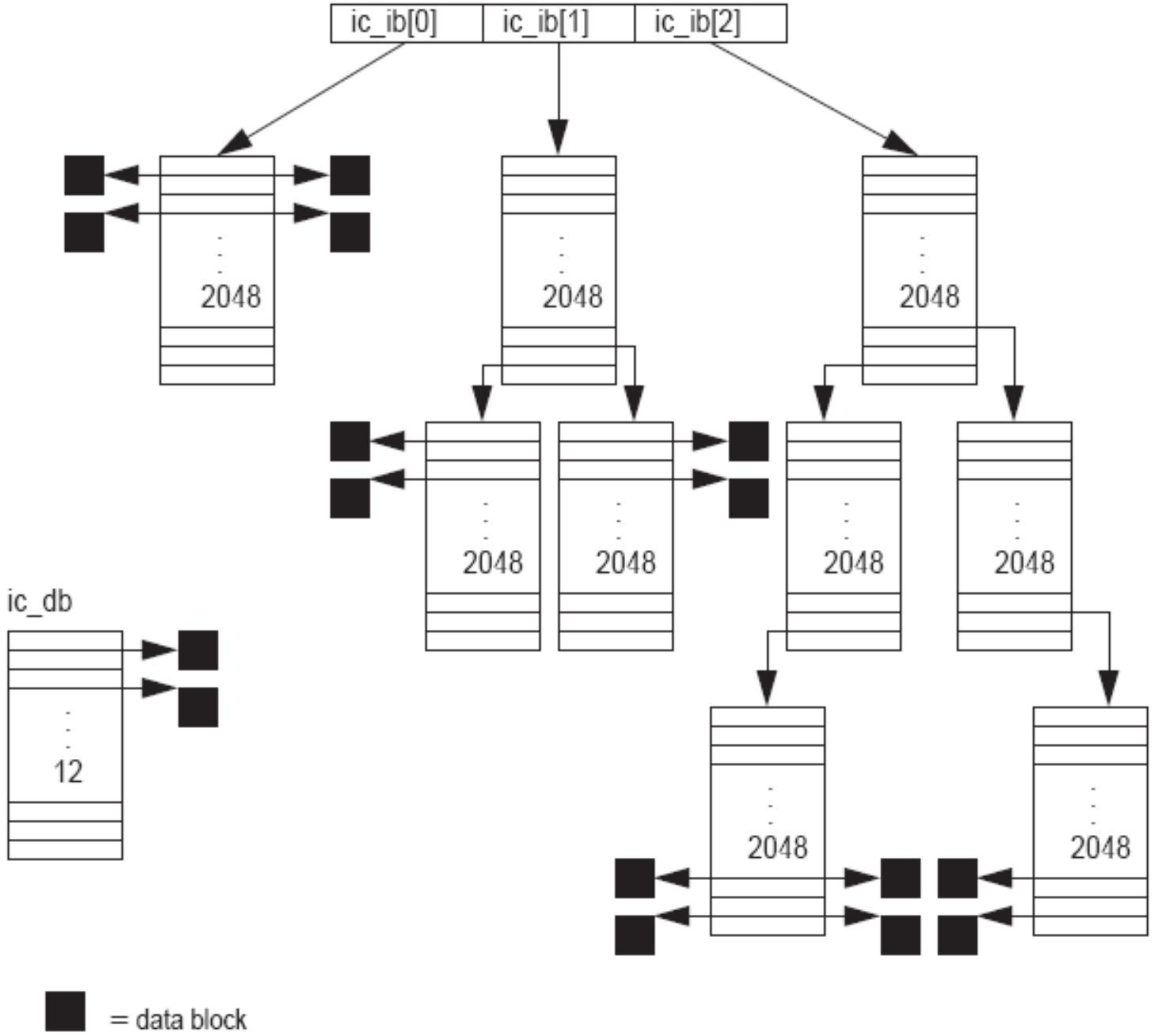
- WIN 32子系统在调用NtCreateFile之前将路径转化为\??\D: \Temp\Test.txt。
- I/O管理器将检查\Device\HarddiskVolume2的VPB是否引用一个文件系统。


1
文件系统驱动程序完成扇区级的I/O
2
I/O管理器把IRP传送给DMIO驱动程序
\Device\HarddiskDmVolumes\PhysicalDmVolumes\BlockVolume3
DMIO 驱动程序
3
DMIO驱动程序决定IRP目标是跨分区卷的哪一个分区，产生一个子目标为该分区所在盘的IRP
\Device\Harddisk0\DR0
\Device\Harddisk0\DR1
4
I/O管理器把IRP传送给磁盘类驱动程序
磁盘类驱动程序
磁盘端口驱动程序
磁盘小端口驱动程序
5
磁盘类驱动程序执行硬件I/O访问磁盘
Harddisk 0
条状分区 1
Harddisk 1
条状分区 2

UFS


Boot Block
SuperblockGroup
C linder Group
Data Block
Data Block
Data Block
Data Block
SuperblockGroup
C linder Group
Data Block
Data Block
Data Block
Data Block
Inodes contains block disk address pointers
Disk addresses are 31-bit units of file s stem fragments.
The address is limited to 2^31 1k for the default file s stem configuratio ielding a ma imum file si e of TB.


fs superblock
struct cg + bitmaps
inodes
data blocks ...


ic_ib[0]
ic_ib[1]
ic_ib[2]
2048
2048
2048
2048
2048
2048
2048
2048
ic_db
12
= data block

- 8K*（（12+2048）+（2048*2048）+（2048*2048*2048））=64Tbytes
- 2^31*1K=2TB