

提交读书报告主题

1. 截止时间：10月11日（本周5）

2. 提交方式：

发邮件至：chenjx@hust.edu.cn

邮件主题：组长姓名+报告题目

邮件内容：（1）读书报告题目

（2）小组成员

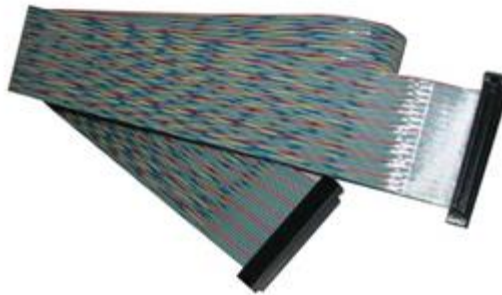
（3）摘要

（4）主要参考文献列表

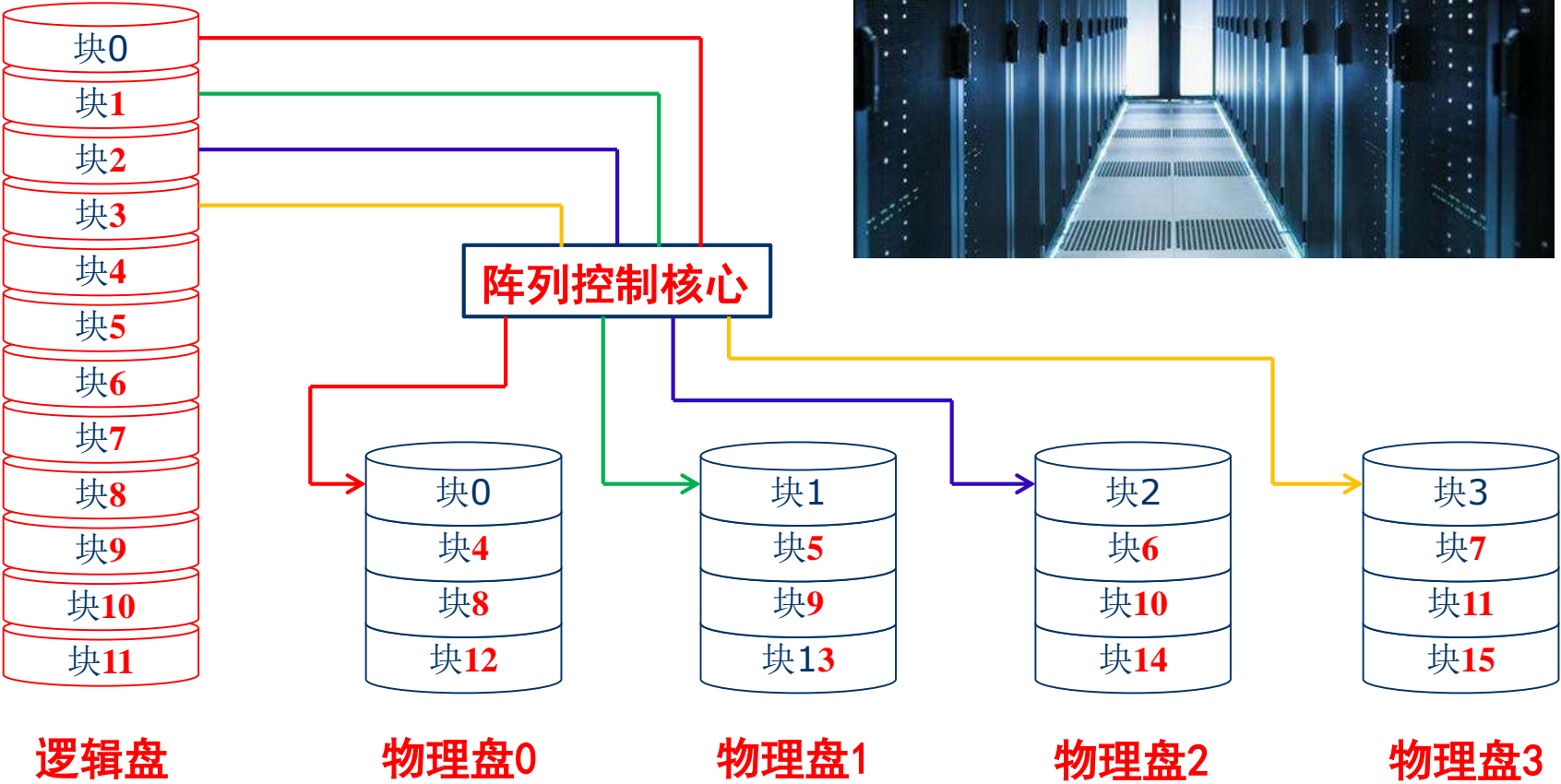
（参考文献按一标准格式，包含作者、题目、期刊/会议名称、发表时间）

3. 正式报告提交：课程考试结束后1周内
（包括报告word文档和PPT）

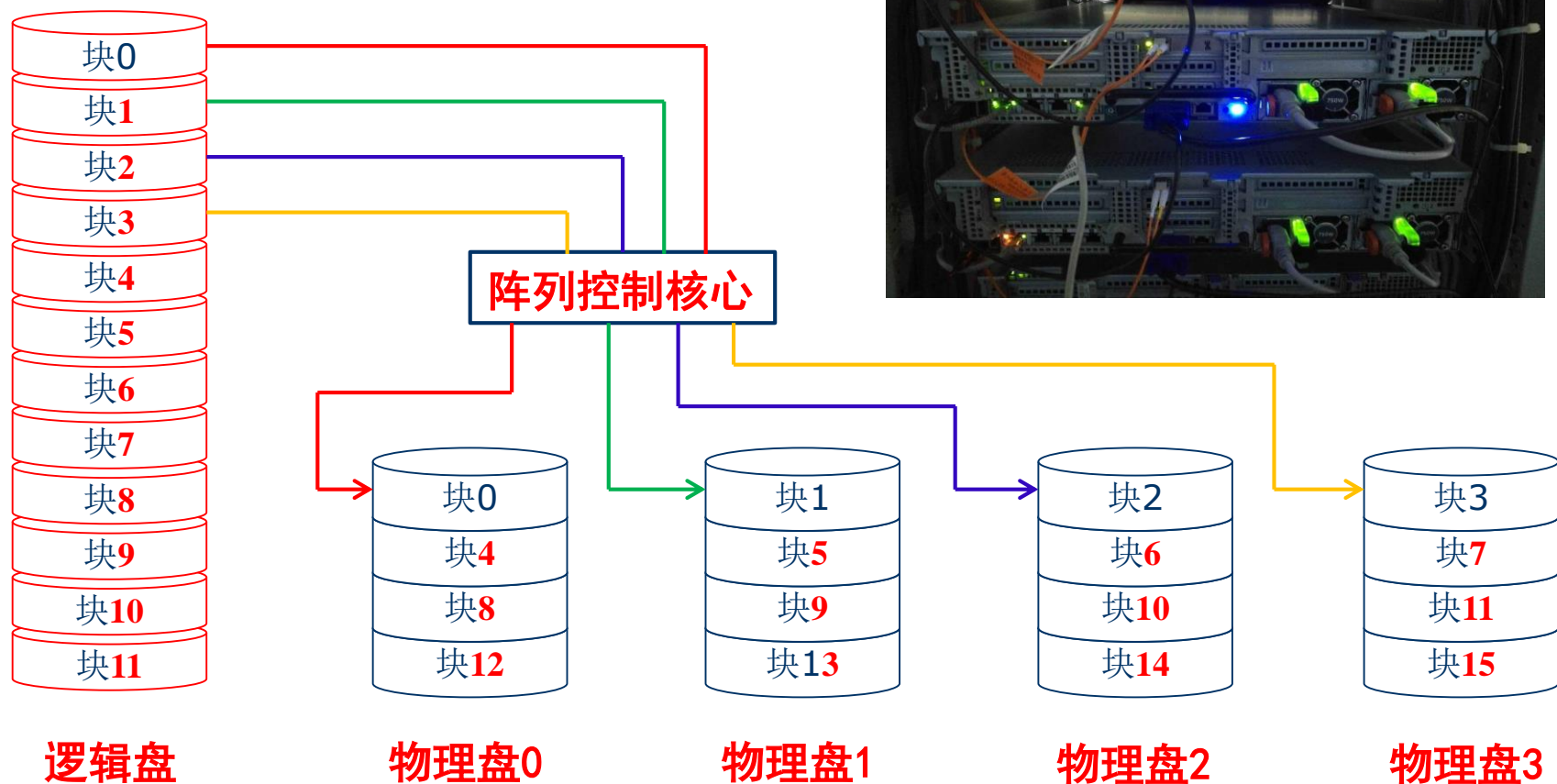
RAID & SCSI



RAID逻辑上构成一个盘设备



RAID逻辑上构成一个盘设备



RAID命令处理过程中的关键问题

- I/O分解

磁盘阵列接收到主机的I/O请求命令，
派生出对应于各个磁盘上的子I/O命令

I/O命令的形式？

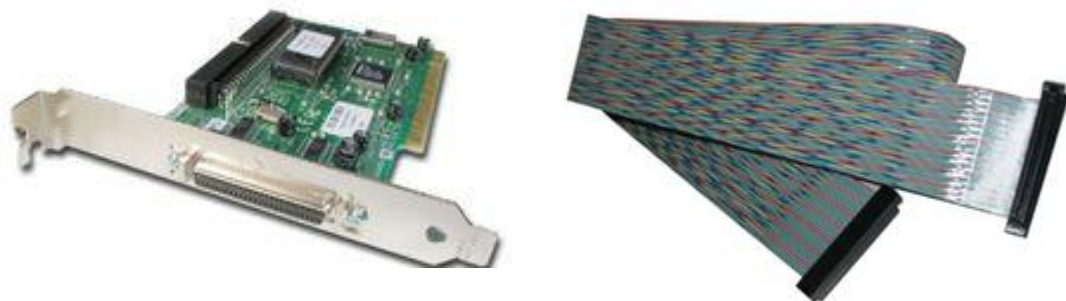
SCSI协议

支撑存储发展的脊梁

- **SCSI**: **S**mall **C**omputer **S**ystem **I**nterface, 计算机与外部设备（特别是存储设备）间系统级接口的标准；
- SCSI标准定义了命令、通信协议、实体电器特性等（物理层，连接层，通信层，应用层）

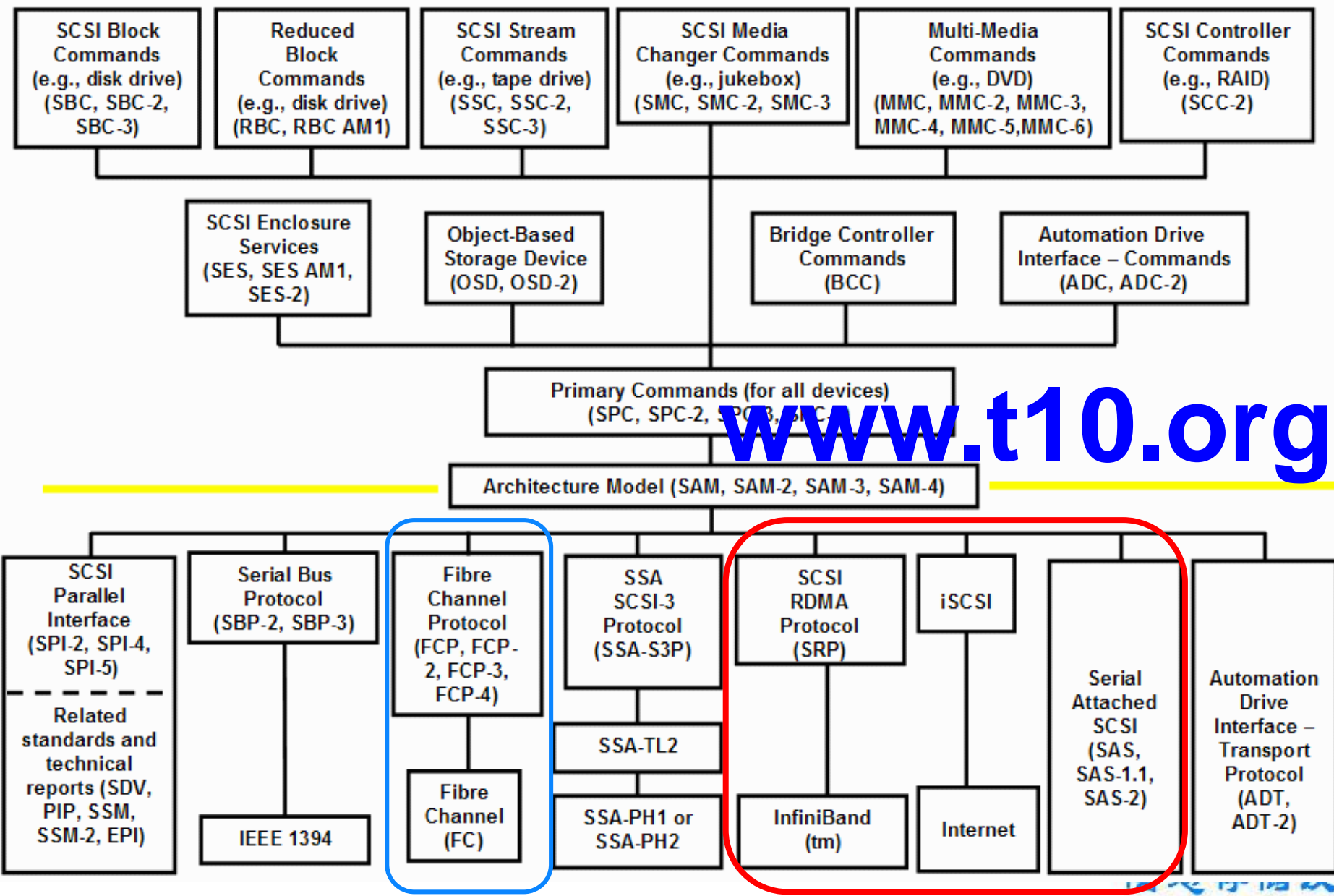
SCSI历史

- **SCSI协议V1版本：5MB/s，8位总线带宽，最多连接7个设备，25针**
- **SCSI协议V2版本：20MB/s（Fast-Wide），16位数据带宽、15个设备，50针或68针。高主频的SCSI存储设备陆续出现并成为市场的主流产品，也使得SCSI技术牢牢地占据了存储市场**



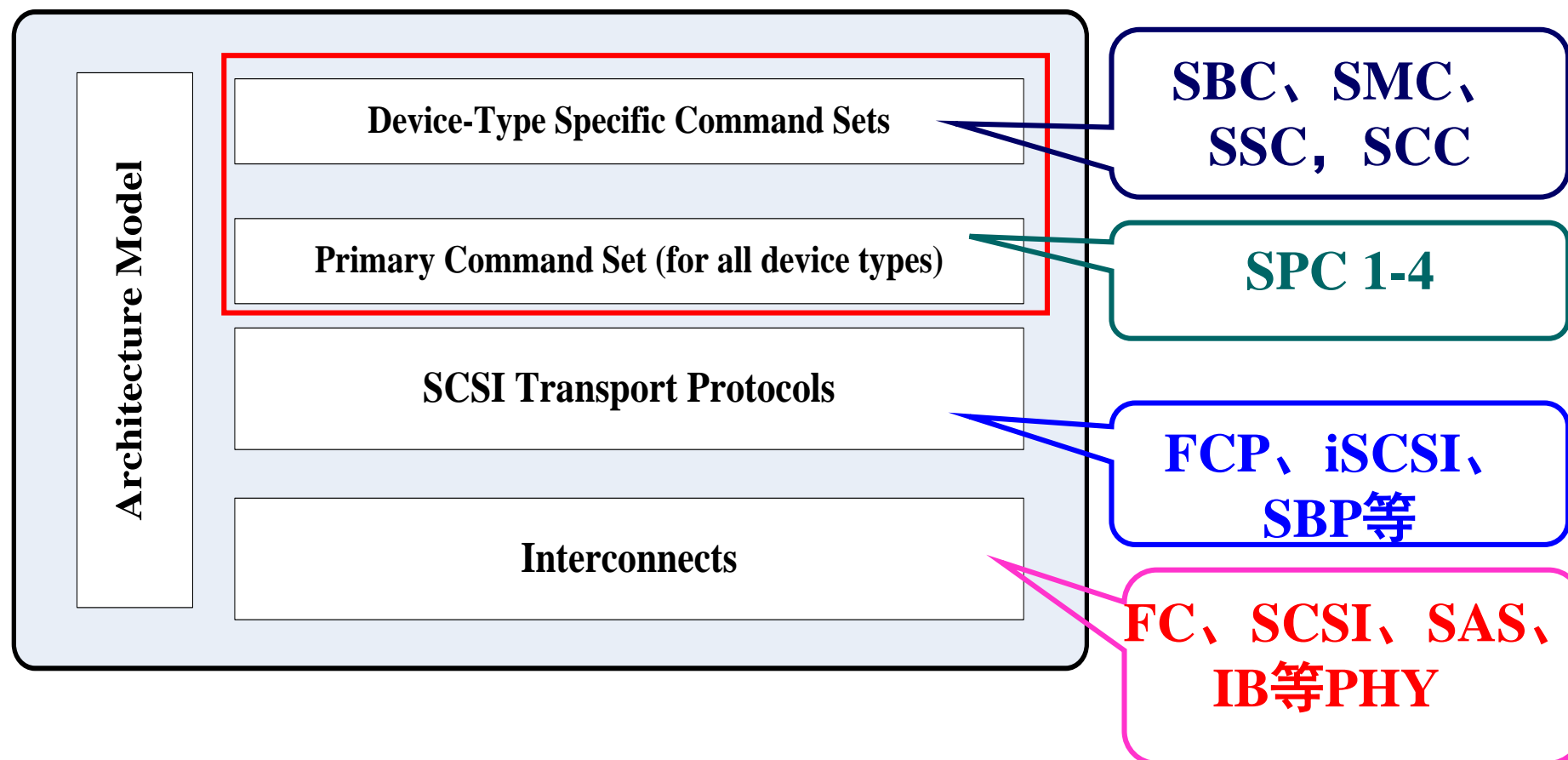
SCSI历史

- **SCSI-3协议**：320MB/s或更高，增加了能满足特殊设备协议所需要的命令集，使得SCSI协议既适应传统的并行传输设备，又能适应最新出现的一些串行设备的通信需要，如光纤通道协议（FCP）、串行存储协议（SSP）、串行总线协议等
- **串行SCSI——SAS**
 - 第一代SAS 1.5Gbps
 - 第二代SAS 3.0Gbps
 - 第三代SAS 6.0Gbps



www.t10.org

SAM-3结构

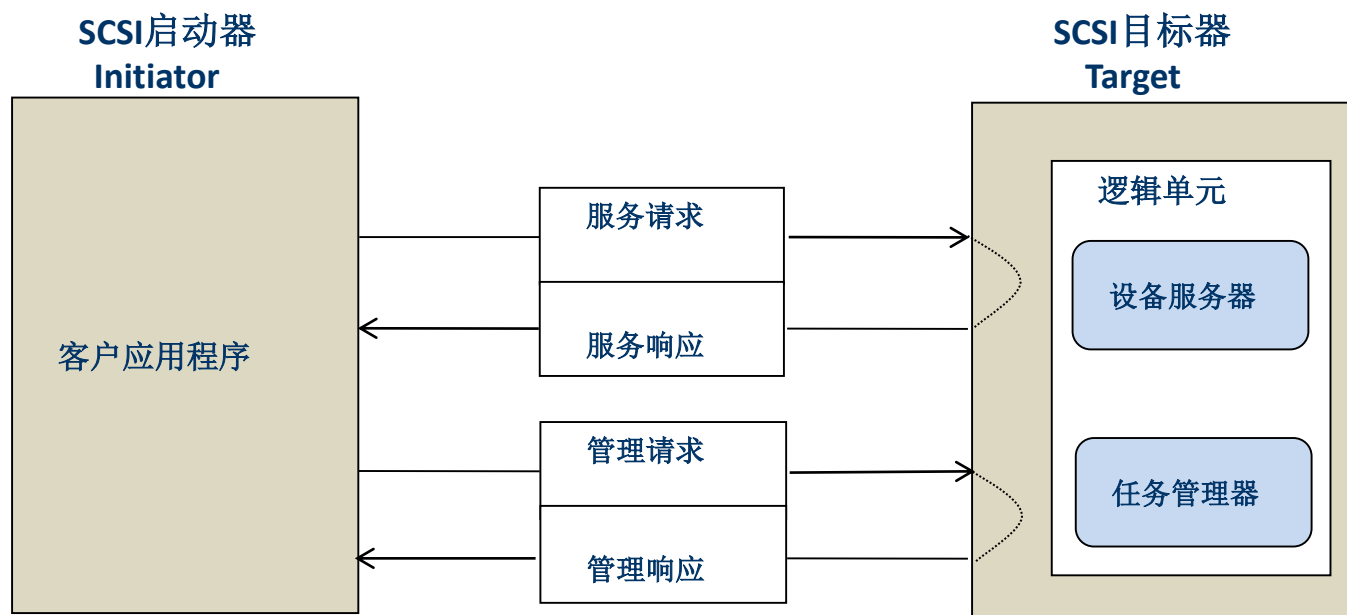


SCSI在系统中的地位

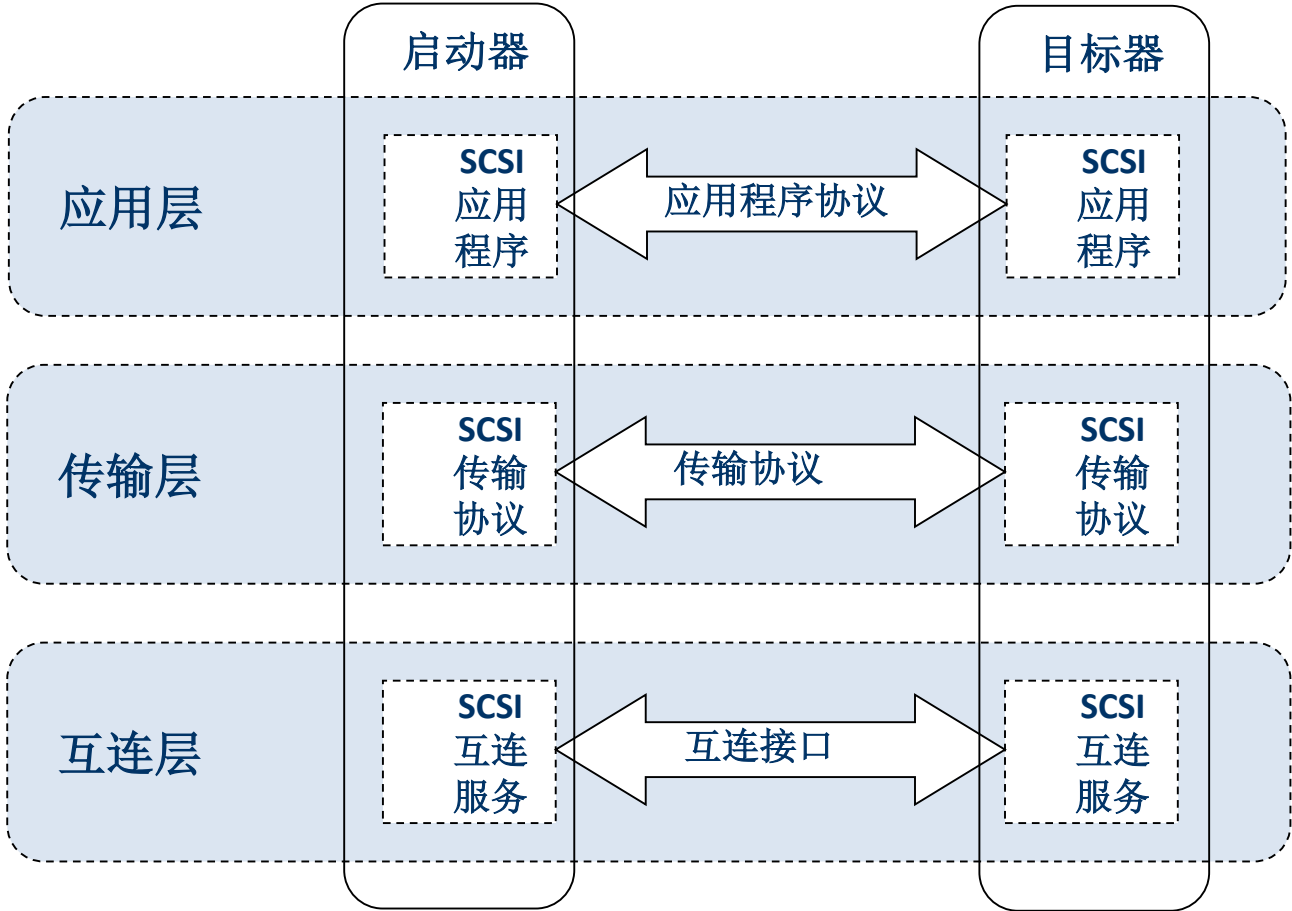


SCSI通信服务模型

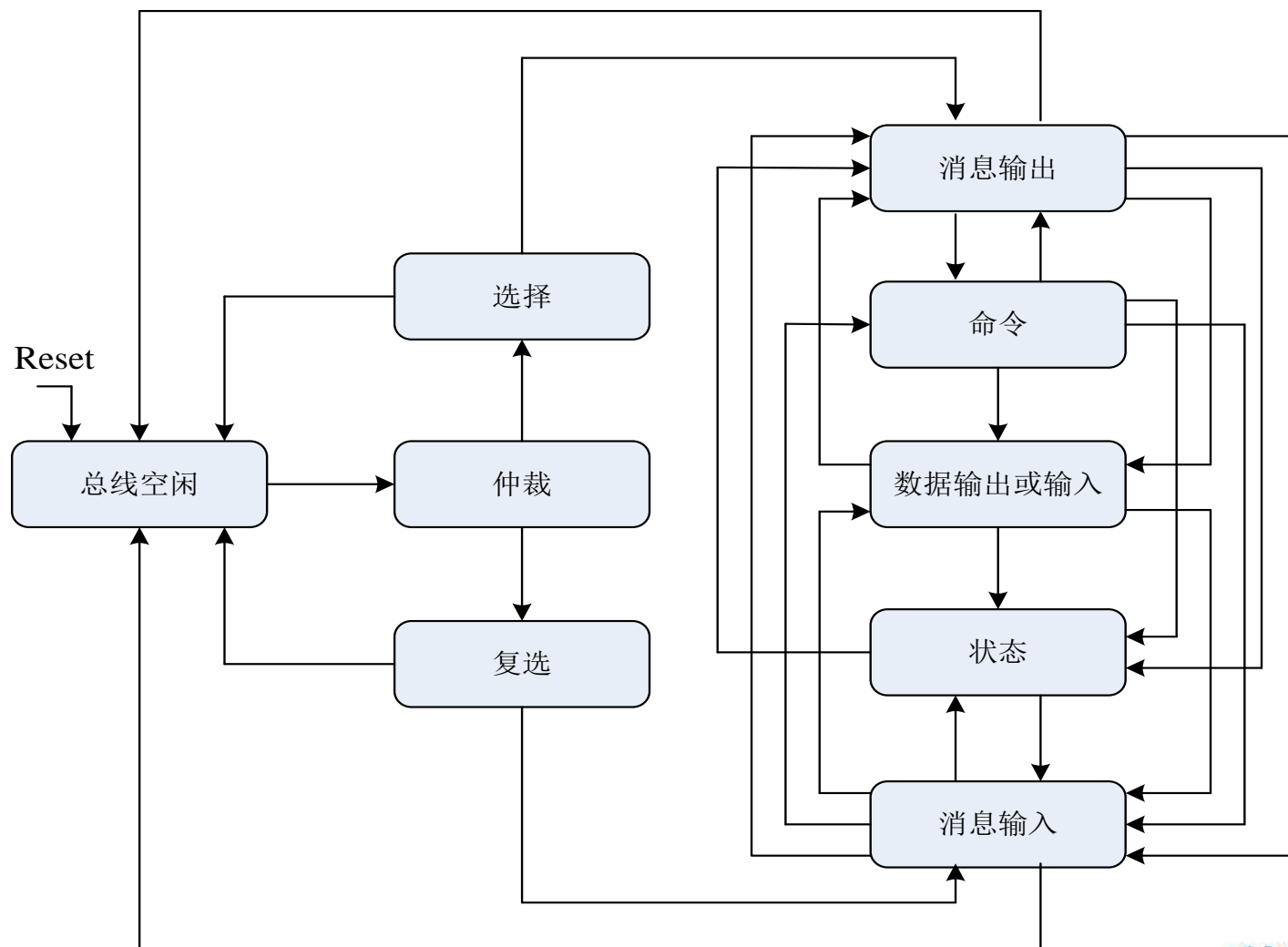
- “客户-服务器”模型



SCSI通信模型



并行SCSI状态变迁



SAS(Serial Attached SCSI)简介

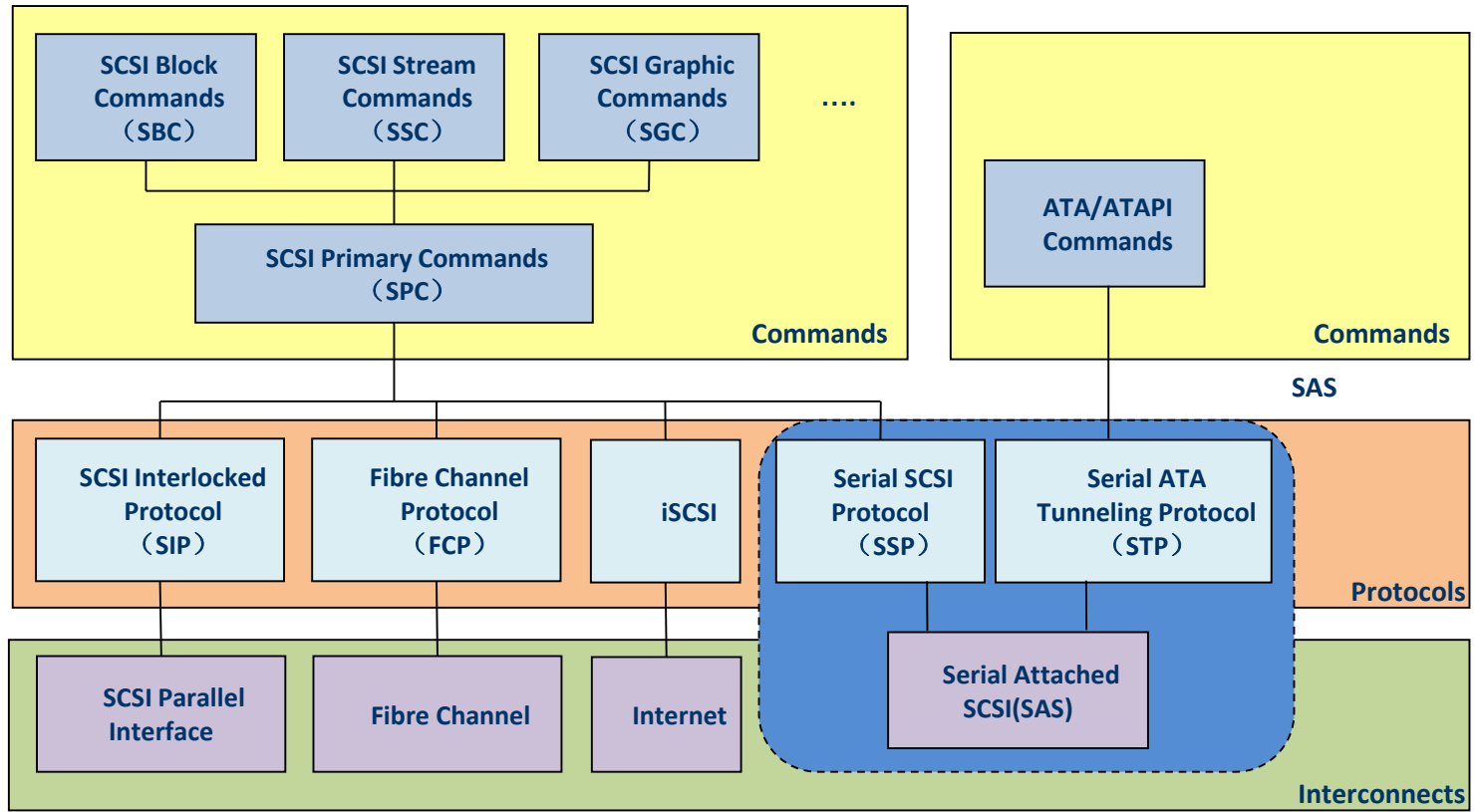
- 产生

- 并行SCSI发展到Ultra320，没有提升空间
- 低端的SATA性能、可靠性受限

- 特点

- SCSI向下兼容性、串行点对点互连、双端口、寻址性和向小型化的扩展能力于一身
- 可提供大数量设备、高带宽、可扩展性支持

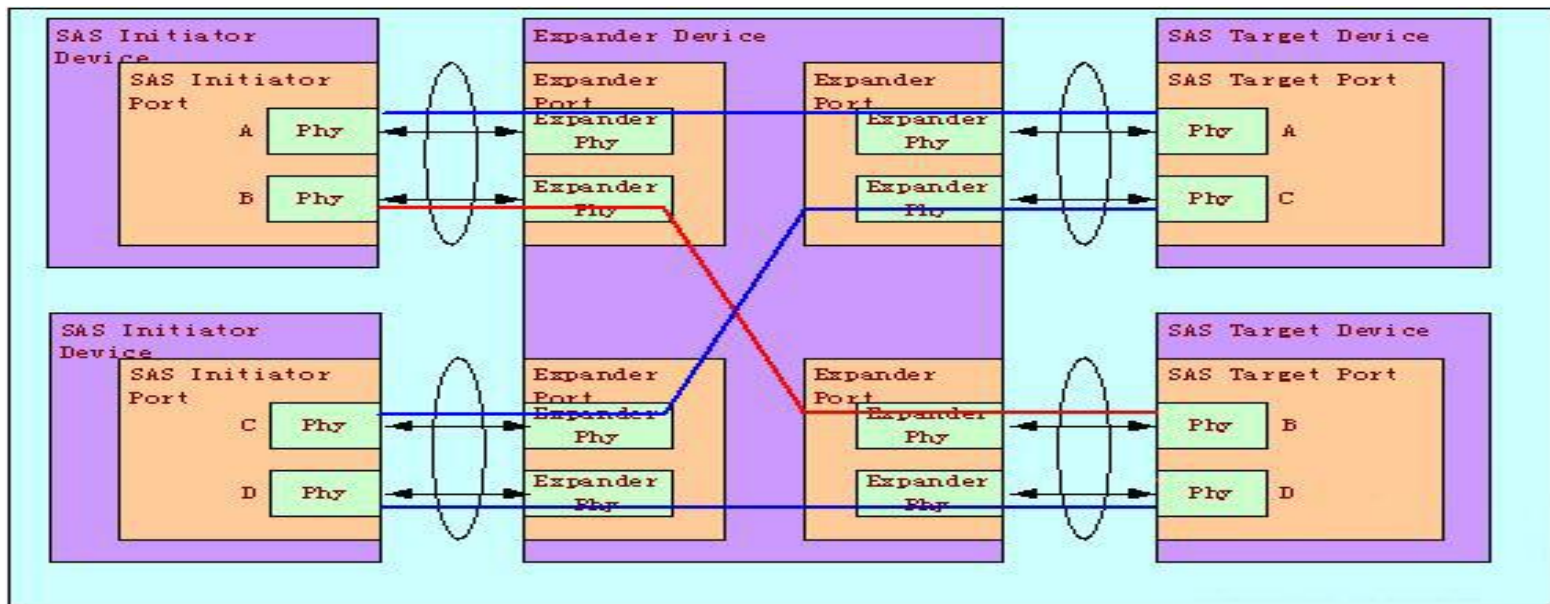
SAS在SAM中的位置



SAS连接

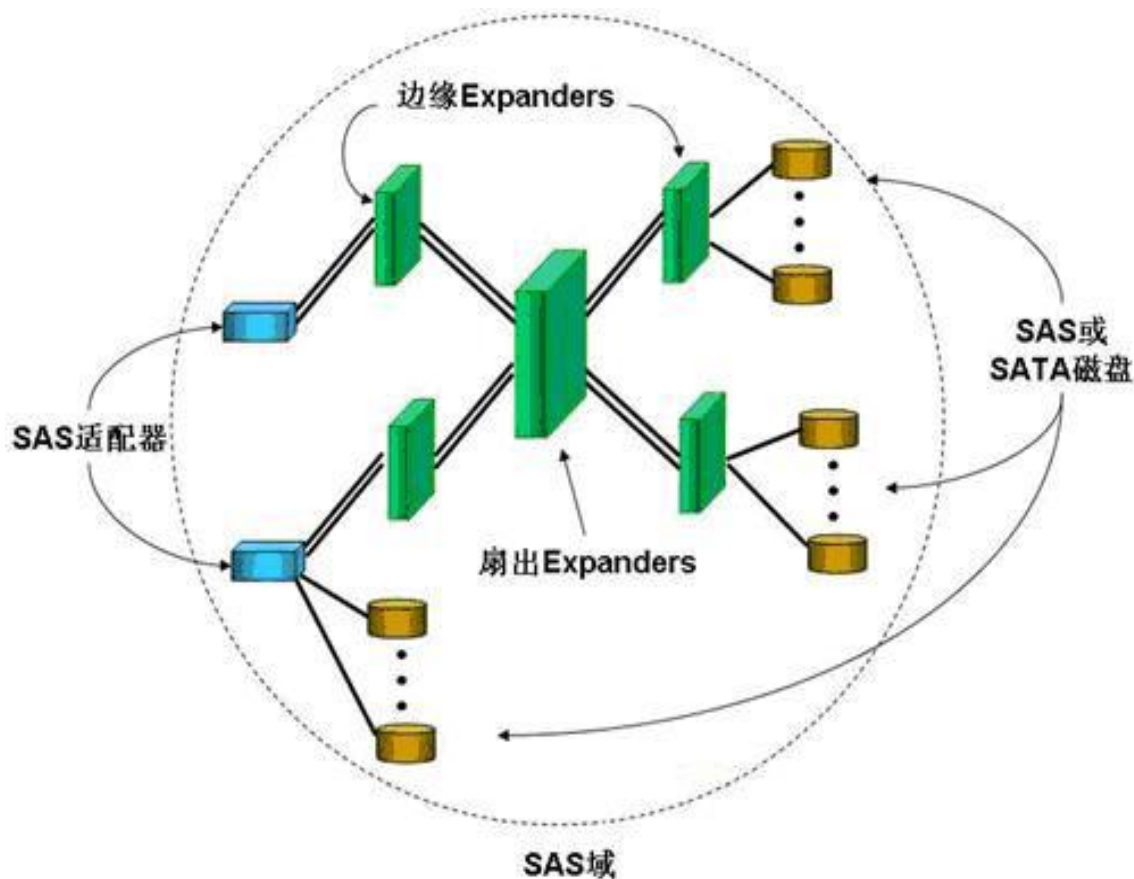
● SAS域

- 一个SAS域主要由SAS初始设备(SAS Initiator Device), 扩展设备 (Expander Device), 以及SAS目标设备(SAS Target Device)组成



SAS连接

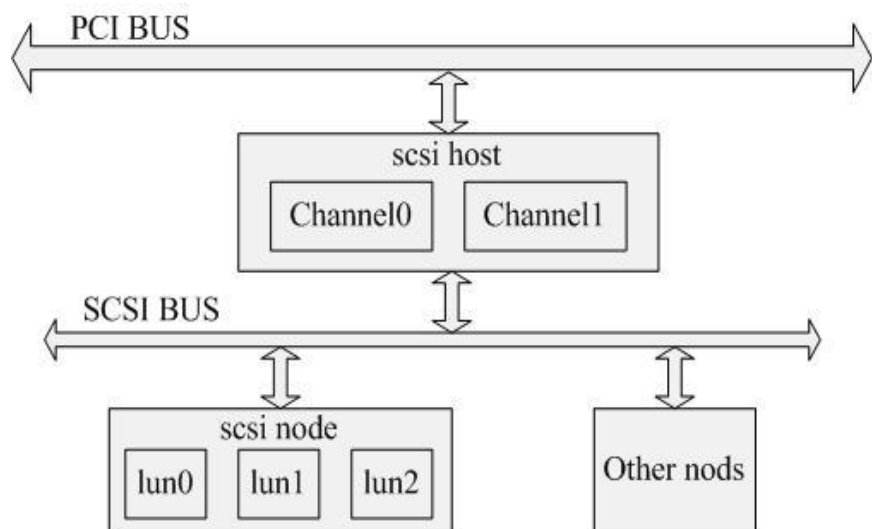
● 连接实例



理论上，每个“边缘Expander”可以支持128个端口，每个SAS域可以有128个“边缘Expander”，因此，每个SAS域中最多可以有 $128 \times 128 = 16384$ 个端口。当然，内部互联至少要占用若干个端口。

对SCSI设备的访问通过三元组：

- 总线（Bus, Channel）
- 目标（Target, ID）
- LUN（Logical Unit Number）



Linux系统SCSI中间层
（SCSI Middle Level）定
义了“scsi device”的数据
结构

SCSI CDB (Command Descriptor
Block)

SCSI命令

Typical CDB for 6-byte commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	miscellaneous CDB information			(MSB)				
2	LOGICAL BLOCK ADDRESS (if required)							
3								
4	TRANSFER LENGTH (if required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)							
5	CONTROL							

TEST UNIT READY(0x00)
INQUIRY(0x12)
READ (0x08)

REPORT LUNS(0xA0)
WRITE (0x0A)

SCSI命令

Typical CDB for 10-byte commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	miscellaneous CDB information			SERVICE ACTION (if required)				
2	(MSB)							
3	LOGICAL BLOCK ADDRESS (if required)							
4								
5								
5								
6	miscellaneous CDB information							
7	(MSB)							
8	TRANSFER LENGTH (if required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)							
9	(LSB)							
	CONTROL							

READ_CAPACITY (0x25)
WRITE_10 (0x2a)
VERIFY (0x2f)

READ_10 (0x28)
WRITE_VERIFY (0x2e)
SYNCHRONIZE_CACHE (0x35)

SCSI命令


Typical CDB for 12-byte commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	miscellaneous CDB information			SERVICE ACTION (if required)				
2	(MSB)							
3	LOGICAL BLOCK ADDRESS (if required)							
4								
5								
5								
6	(MSB)							
7	TRANSFER LENGTH (if required)							
8	PARAMETER LIST LENGTH (if required)							
9	ALLOCATION LENGTH (if required)							
9	(LSB)							
10	miscellaneous CDB information							
11	CONTROL							

- READ_12 (0xa8)
- WRITE_12 (0xaa)
- WRITE_VERIFY_12 (0xae)
- SEARCH_HIGH_12 (0xb0)
- SEARCH_EQUAL_12 (0xb1)
- SEARCH_LOW_12 (0xb2)

SCSI命令

Typical CDB for long LBA 16-byte commands

Bit Byte	7	6	5	4	3	2	1	0	
0	OPERATION CODE								
1	miscellaneous CDB information								
2	(MSB)		LOGICAL BLOCK ADDRESS						
3									
4									
5									
6									
7									
8									
9									(LSB)
10	(MSB)	TRANSFER LENGTH (if required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)							
11									
12									
13								(LSB)	
14	miscellaneous CDB information								
15	CONTROL								

READ_16(0x88) write_16(0x8a)
Readcapacity_16(0x9e)

命令码

Bit	7	6	5	4	3	2	1	0
	GROUP CODE			COMMAND CODE				

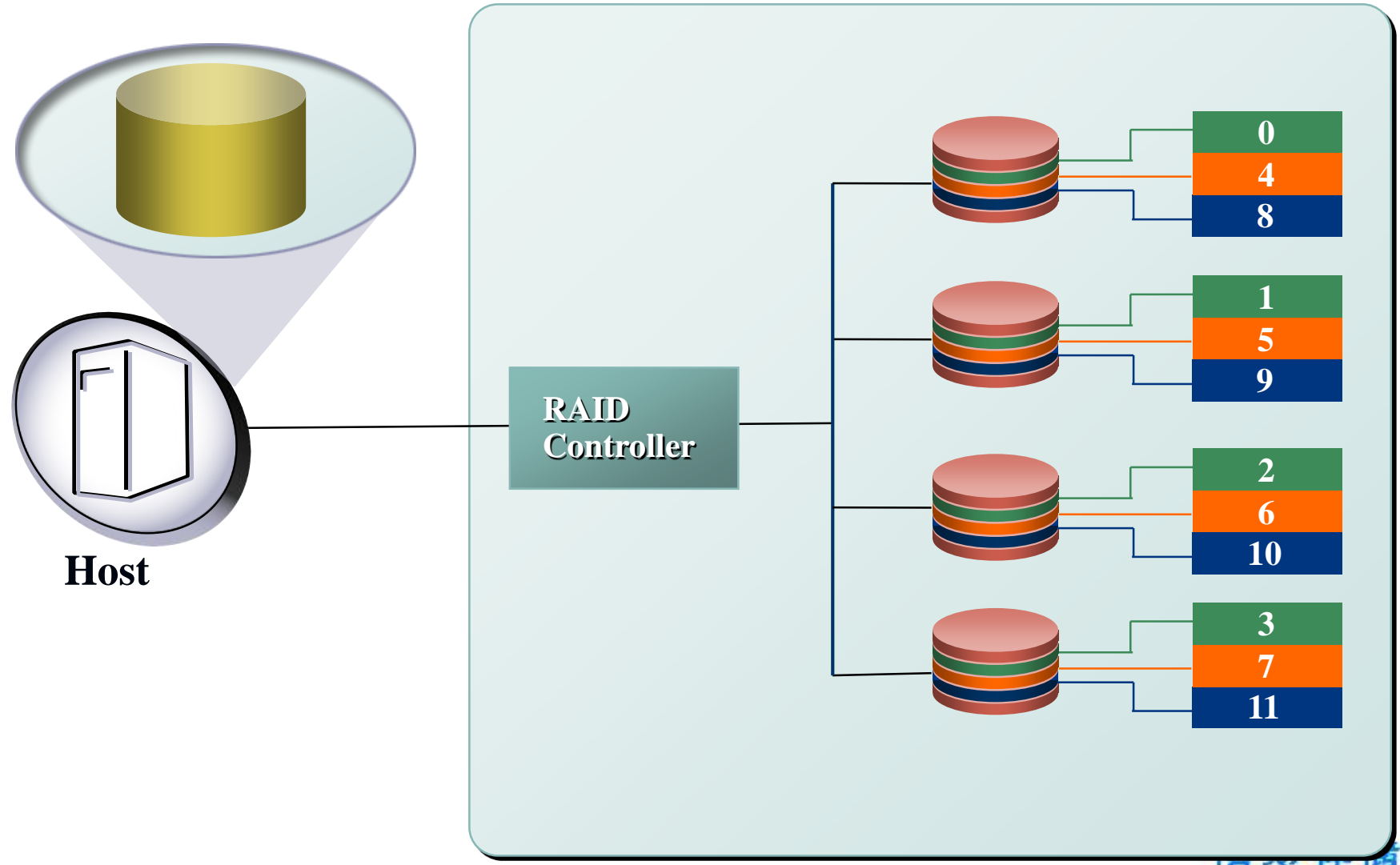
0x

A

B

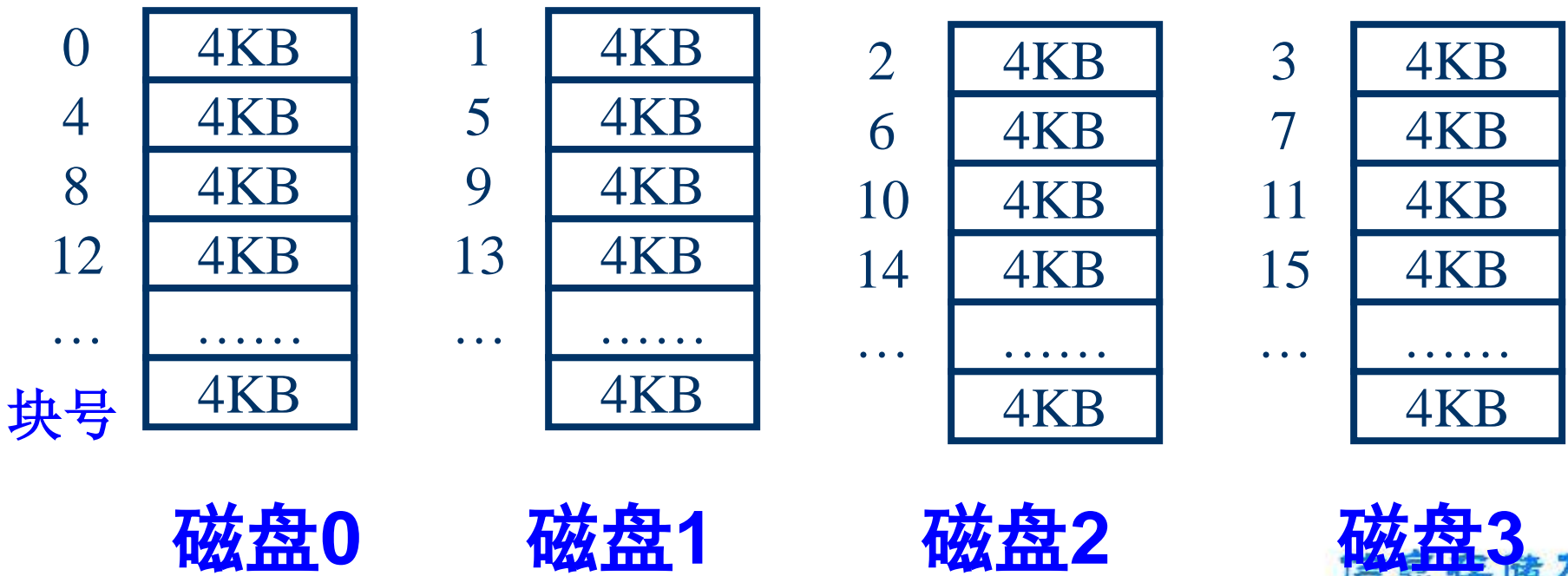
Group code	A	Meaning	example
000	0, 1	6字节命令	0x08, 0x12
001	2, 3	10字节命令	0x25, 0x2a,0x35
010	4, 5	10字节命令	0x55, 0x5a
011	6, 7	reserved	
100	8, 9	16字节命令	0x88, 0x8a,x09e
101	A, B	12字节命令	0xa8, 0xaa
110	C, D	Vendor specific	
111	E, F	Vendor specific	

命令分解举例 (RAID 0)

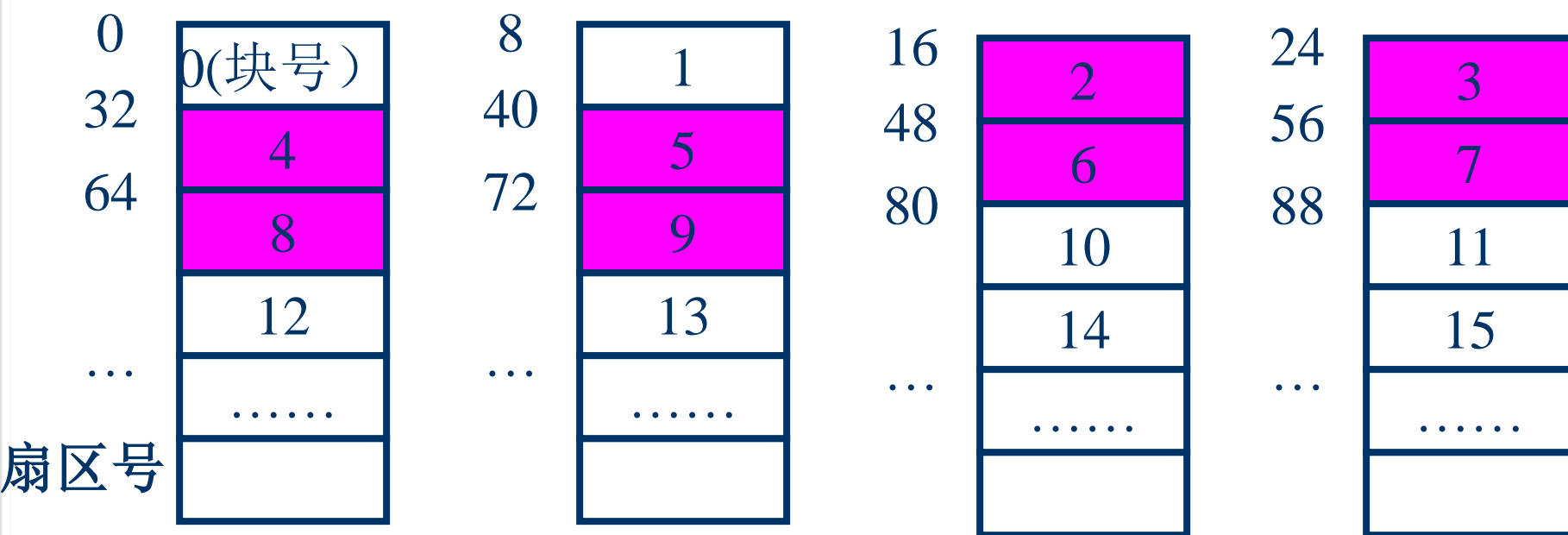


命令分解举例 (RAID 0)

设阵列由4个磁盘构成，每个磁盘容量为C，则主机用户看到的阵列为一个容量为4C的大磁盘；数据分块大小为4KB，构成的阵列结构如下：



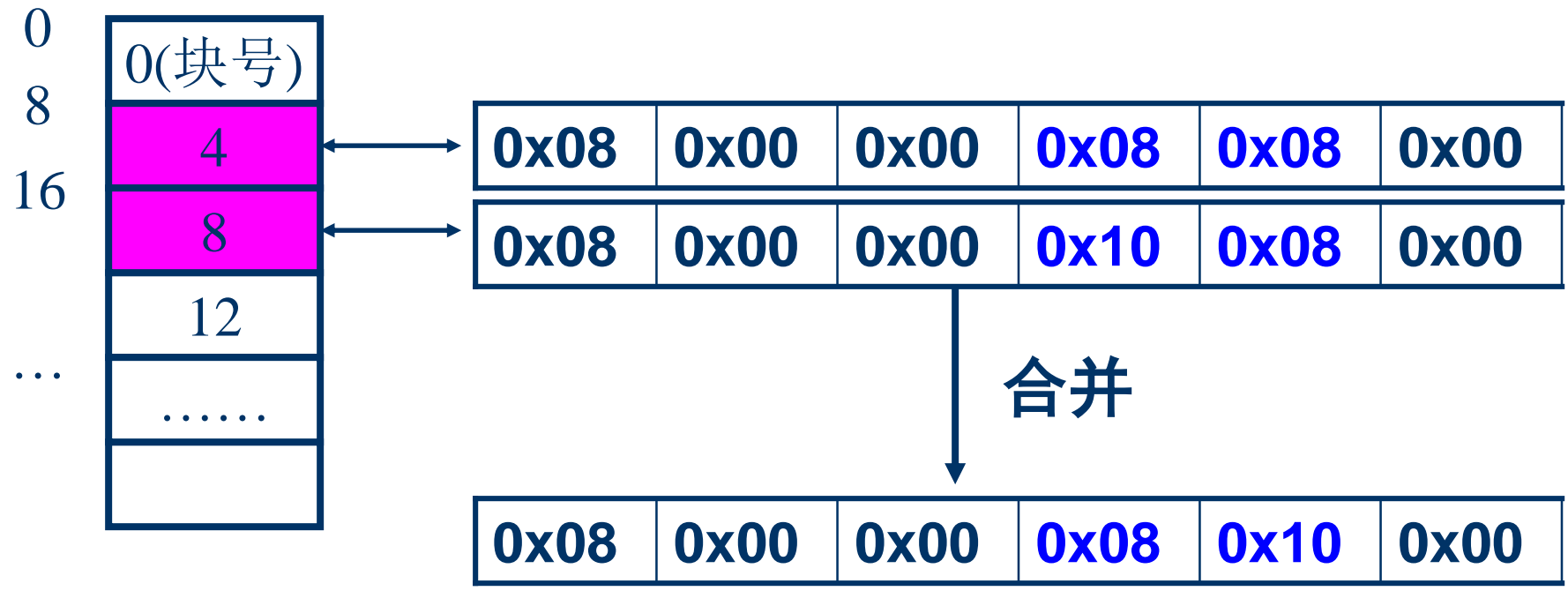
设主机请求读32KB的数据，起始地址为16（单位：扇区），SCSI命令为：
(0x08, 0x00, 0x00, 0x10, 0x40, 0x00)



要读的数据在磁盘上的分布

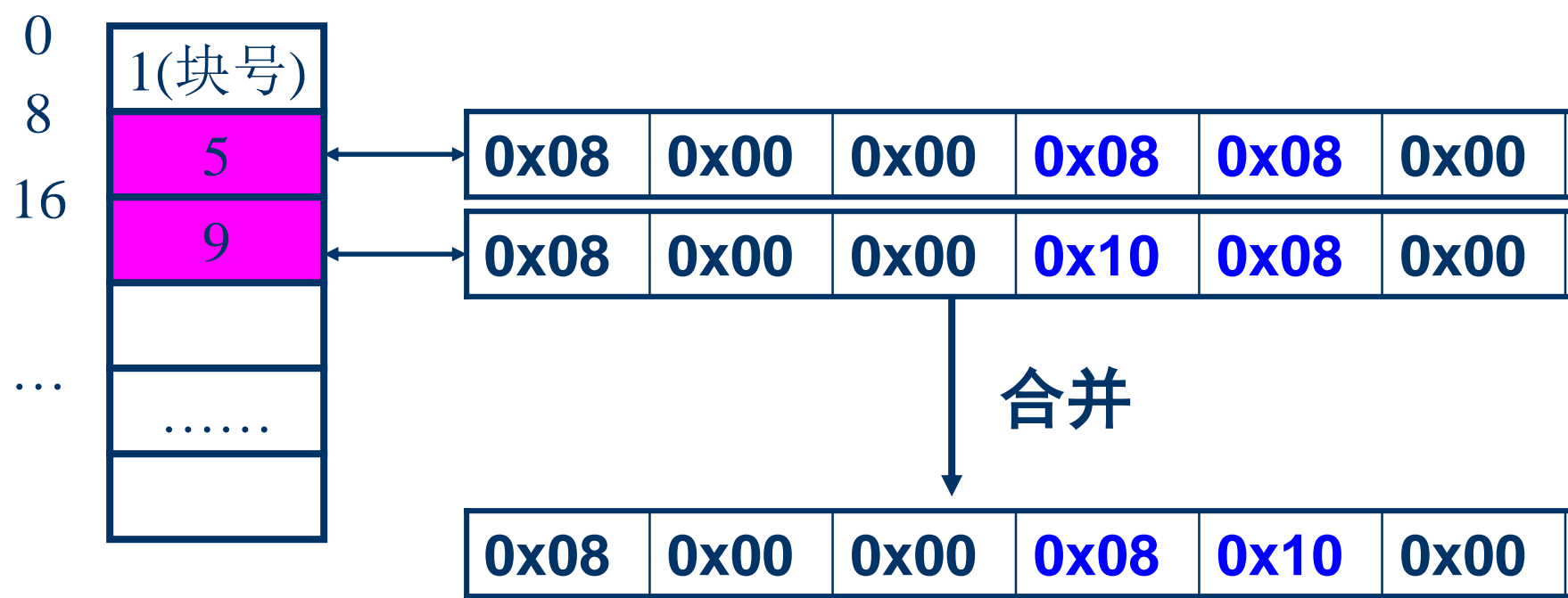
实际上主机要读的数据就是上图中的2~9块，并且是按照2, 3, 4, ..., 9的顺序。

对应单个磁盘上的子命令：
磁盘1：

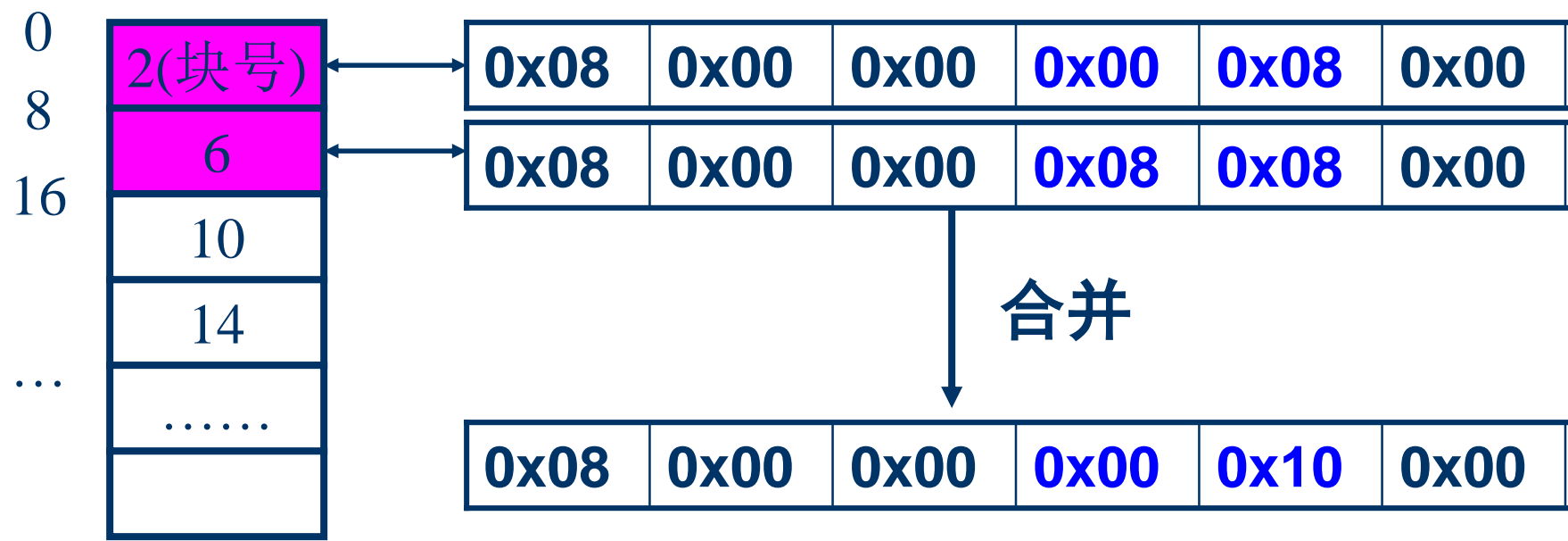


磁盘2

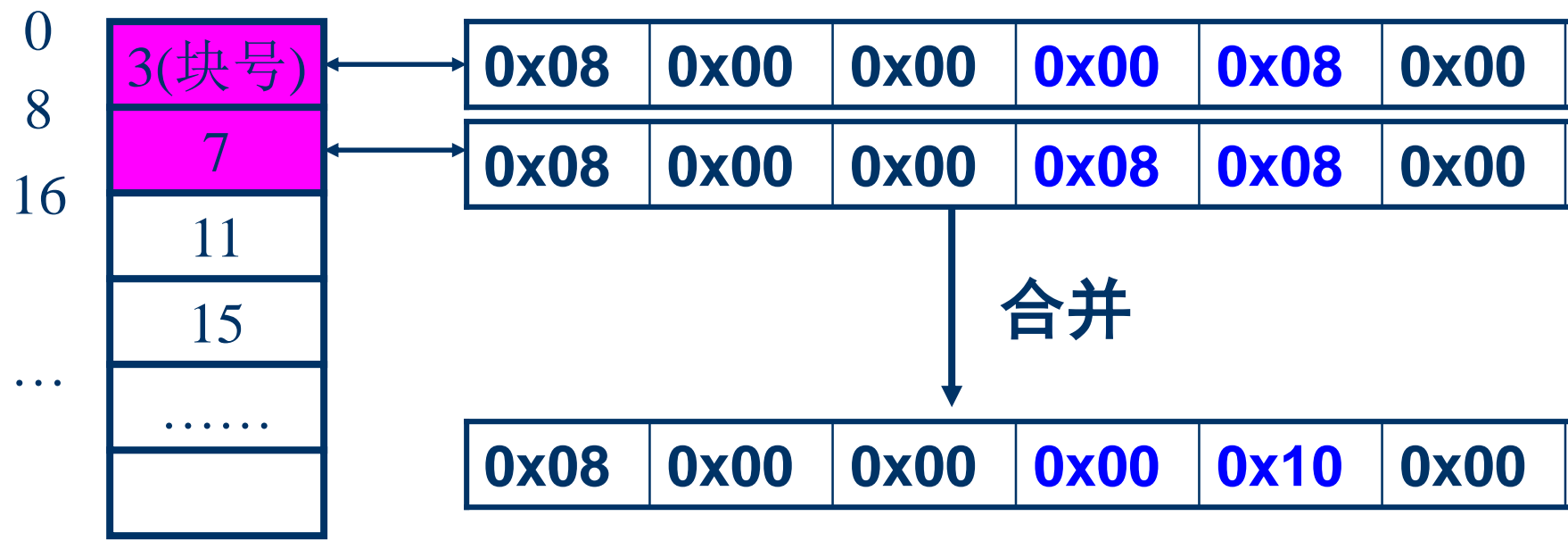
单个磁盘上的扇区号



磁盘3



磁盘4



Next

数据在内存中的分布：

经过命令合并（减少I/O次数），采用普通DMA方式从磁盘读出数据在内存中分布如下：

磁盘1

磁盘2

磁盘3

磁盘4

...	4k	4k	...	4k	4k	...	4k	4k	...	4k	4k	...
-----	----	----	-----	----	----	-----	----	----	-----	----	----	-----

块号： 4, 8

5, 9

2, 6

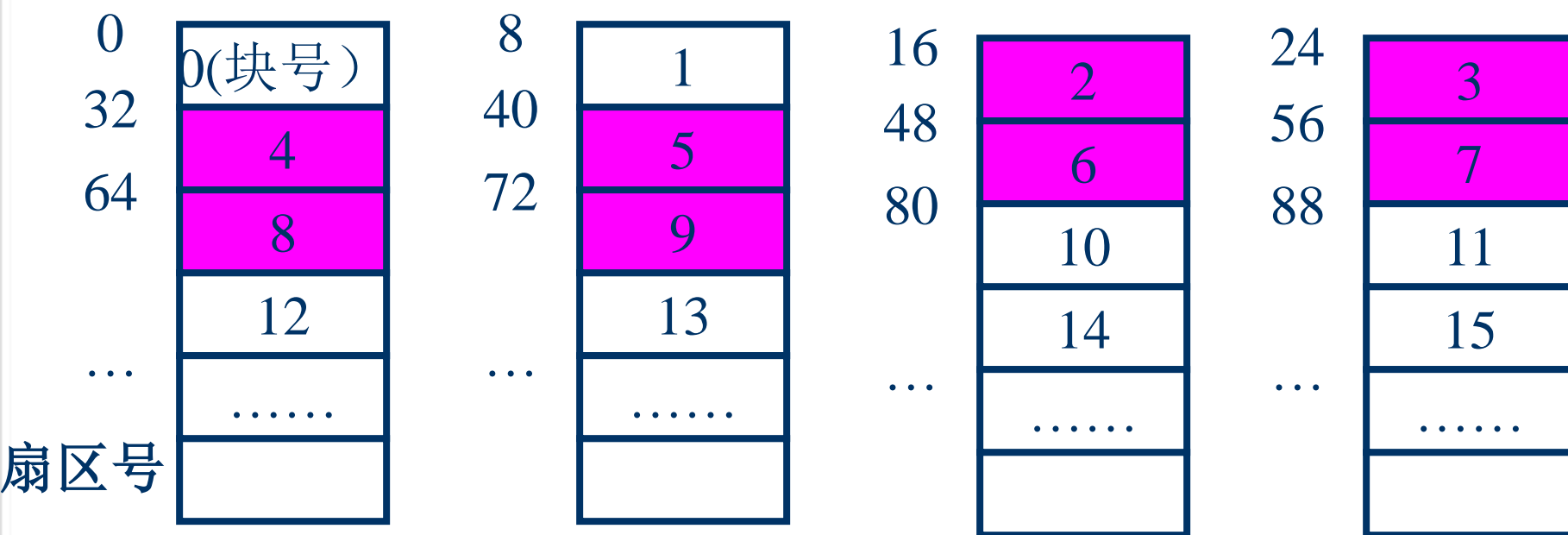
3, 7

重组

...	4k	4k	4k	4k	4k	4k	4k	4k	...
-----	----	----	----	----	----	----	----	----	-----

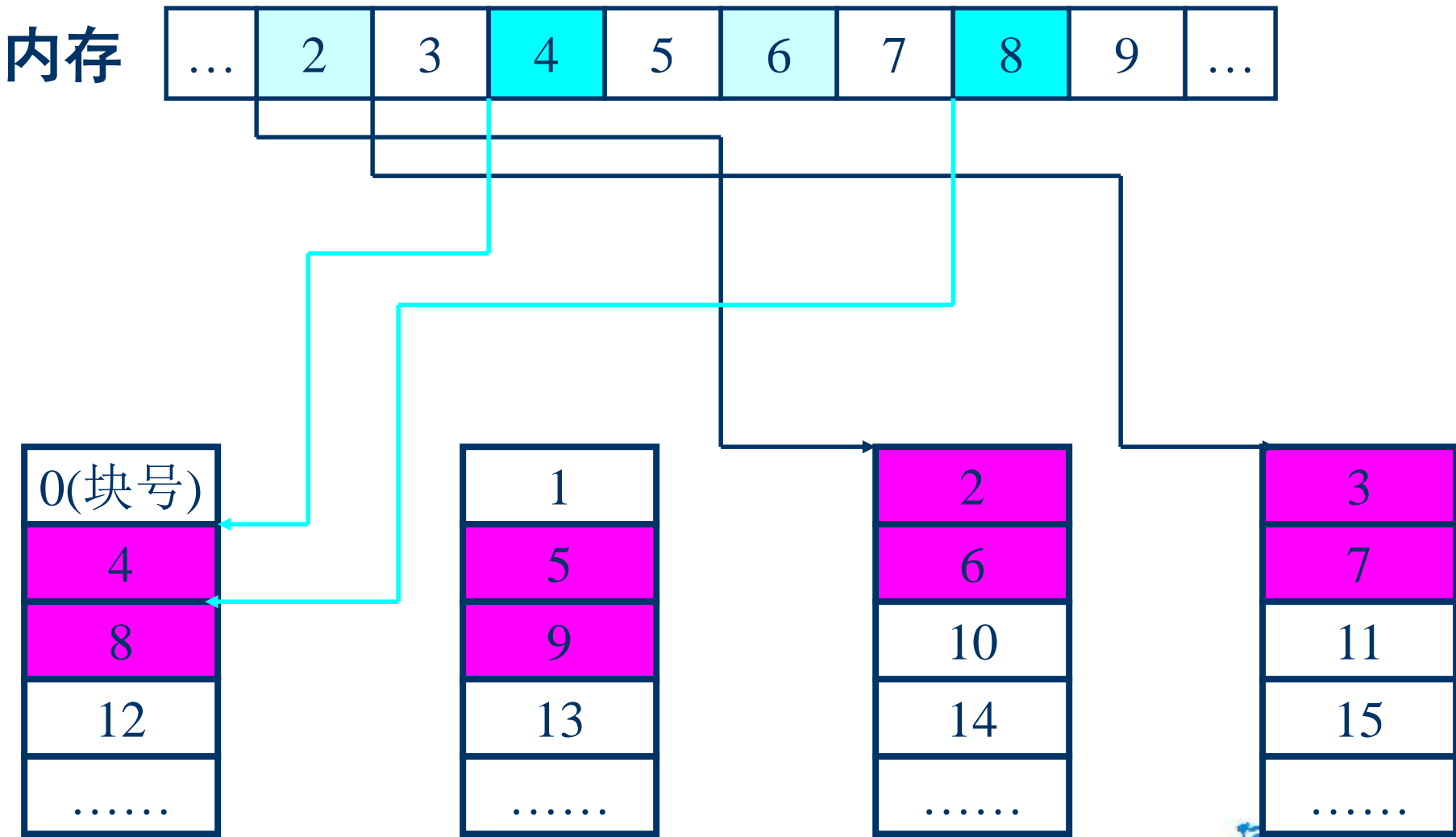
块号： 2, 3, 4, 5, 6, 7, 8, 9

写过程：32KB的数据程度，起始地址为
16（单位：扇区），SCSI命令为：
(0x0A, 0x00, 0x00, x10, 0x40, 0x00)

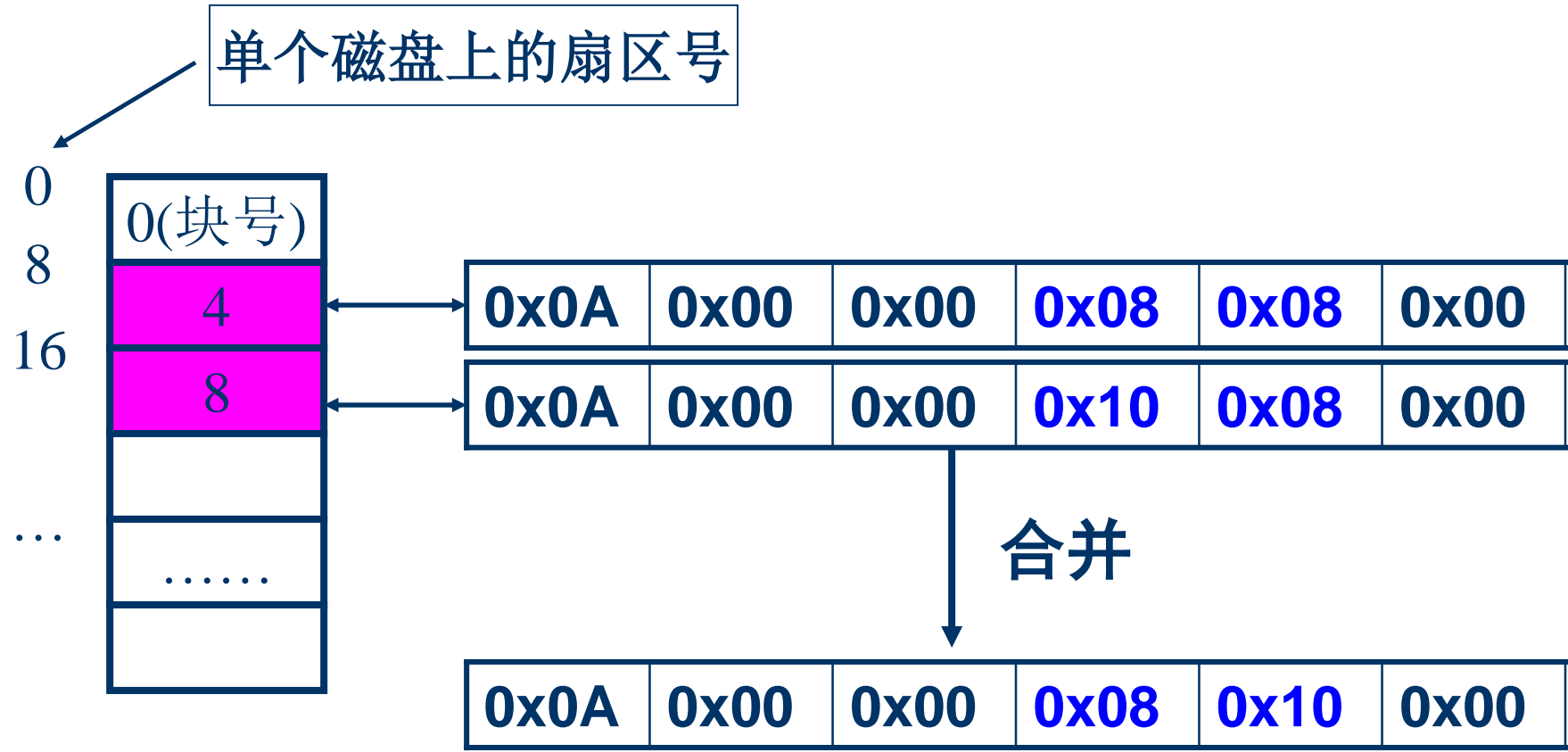


要写的数据在磁盘上的分布

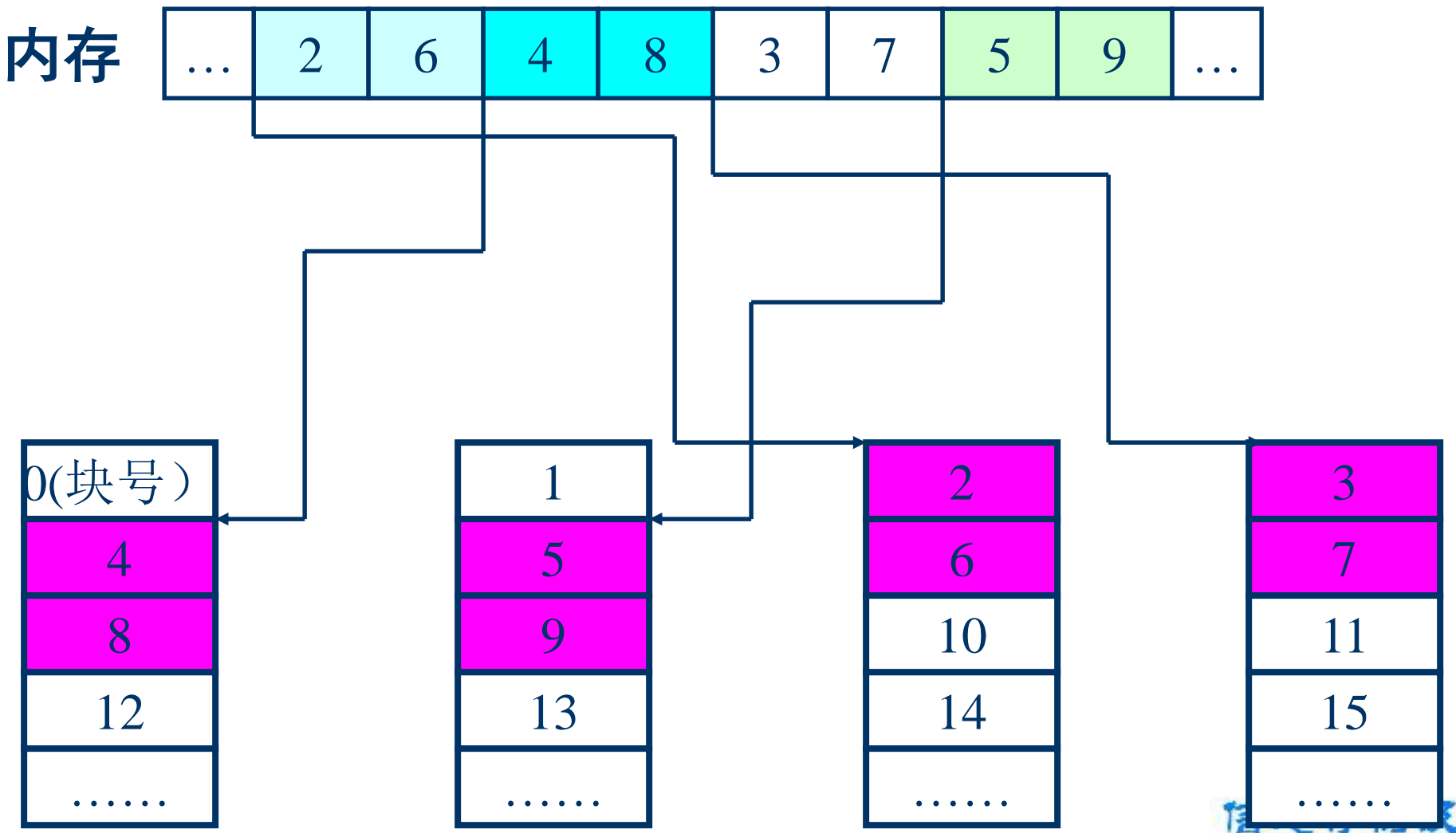
内存中分块数据—磁盘数据对应关系：



对应磁盘1上的写命令：



分块重组后内存—磁盘对应关系：



数据分块重组带来的问题—数据移动

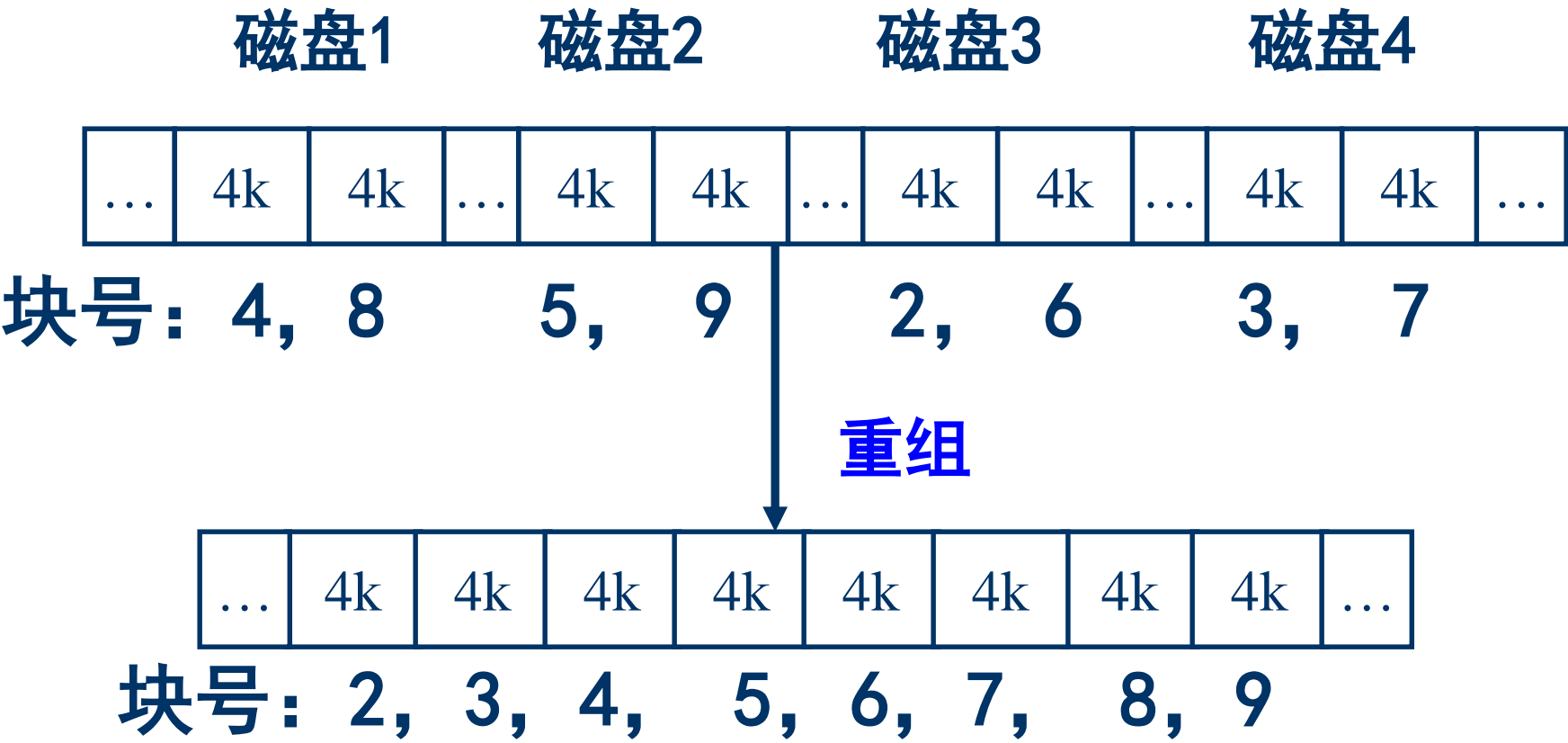
分块前

...	2	3	4	5	6	7	8	9	...
-----	---	---	---	---	---	---	---	---	-----

重组后

...	2	6	4	8	3	7	5	9	...
-----	---	---	---	---	---	---	---	---	-----

读命令同样存在这个问题



时间开销问题：

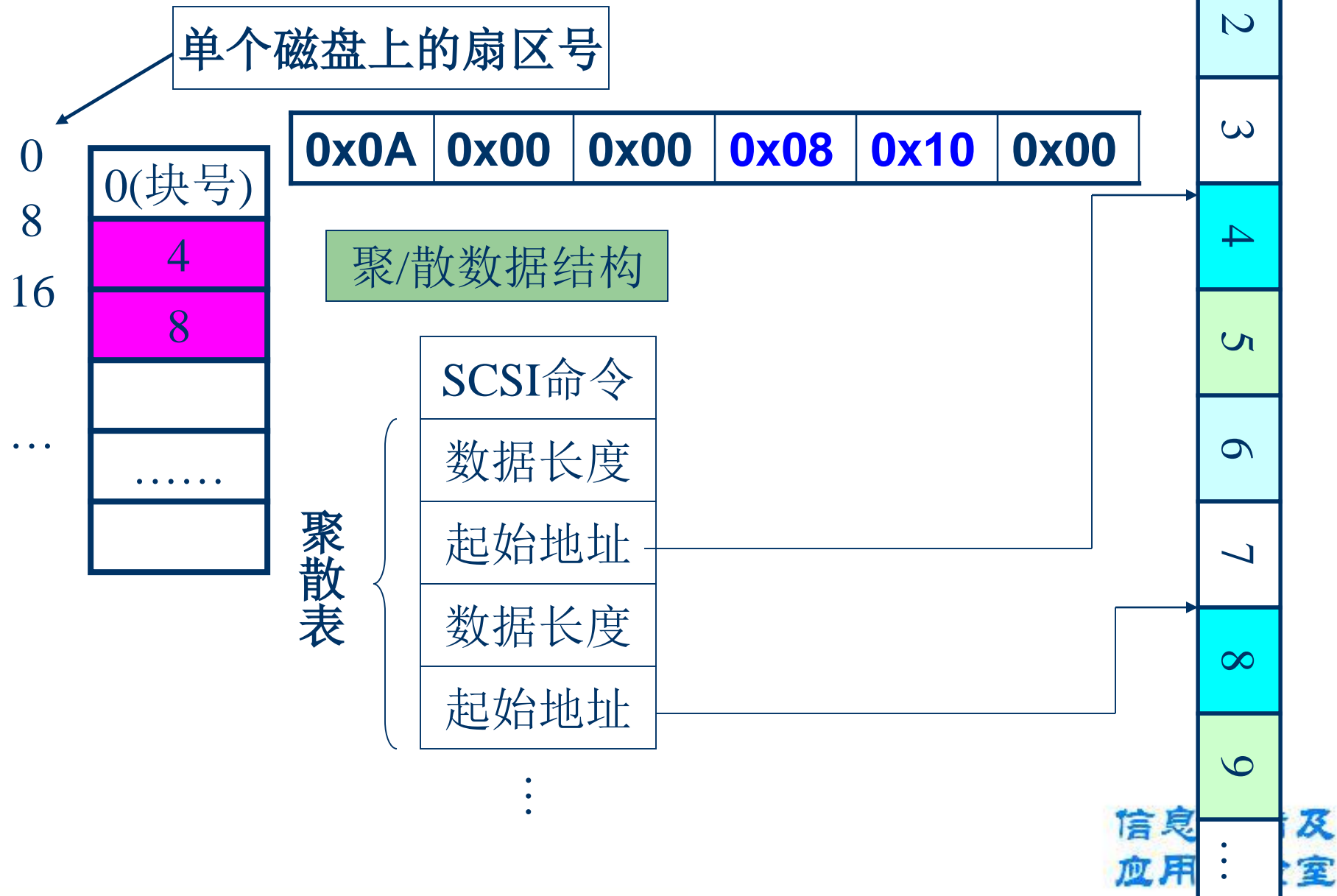
减少磁盘I/O操作的次数极大地提高阵列的性能，所以采用了I/O合并技术，I/O合并引起数据在内存中的移动，这种移动开销相对于磁盘I/O开销要小的多，但性能的影响还是相当大的（近25%）。如何消除这种不必要的开销？

聚/散技术

消除数据移动带来的额外开销，聚散技术的基本思想是将散落在内存的各块数据聚集起来传输。

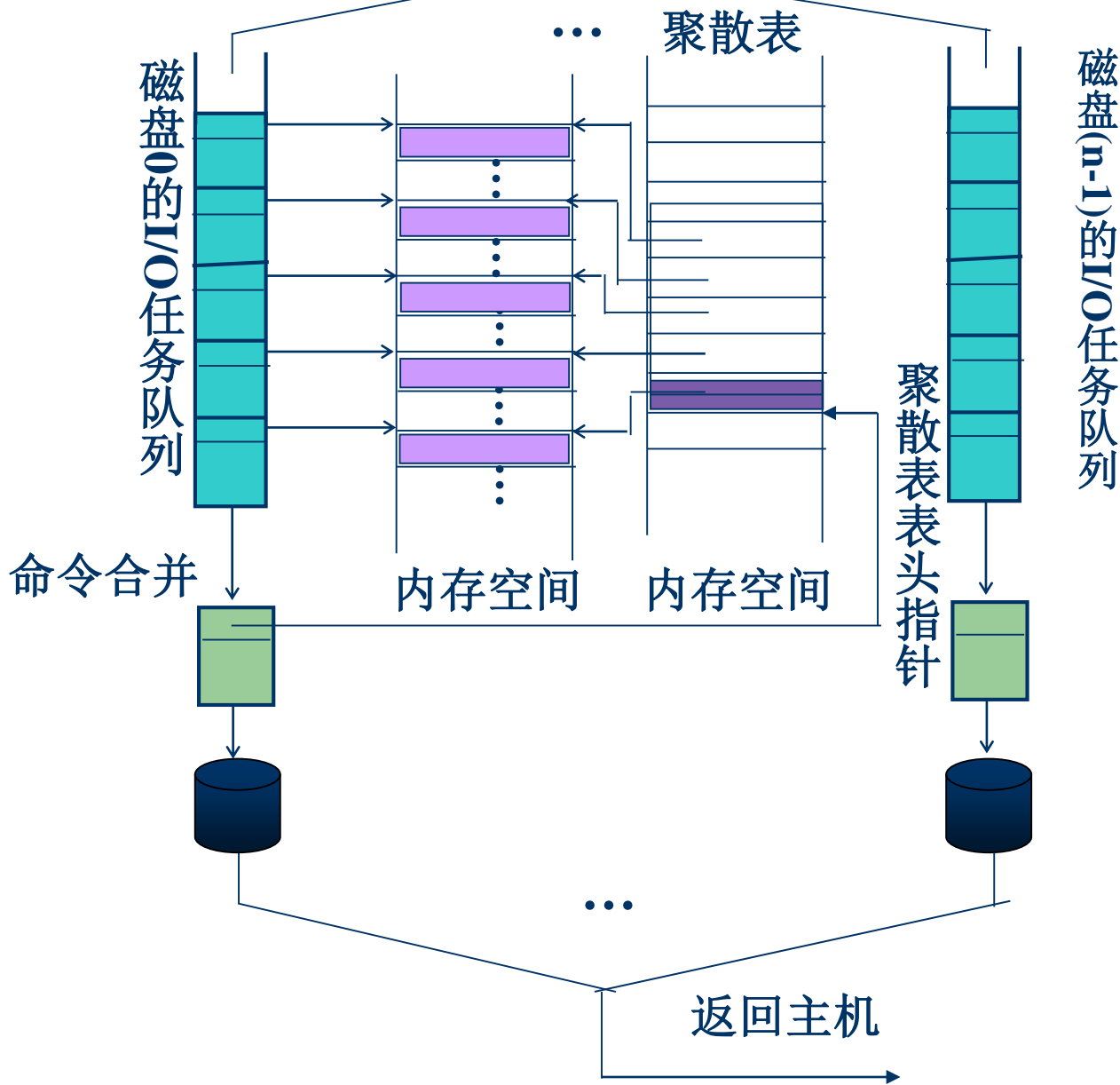


对应磁盘1上的写命令聚散格式：



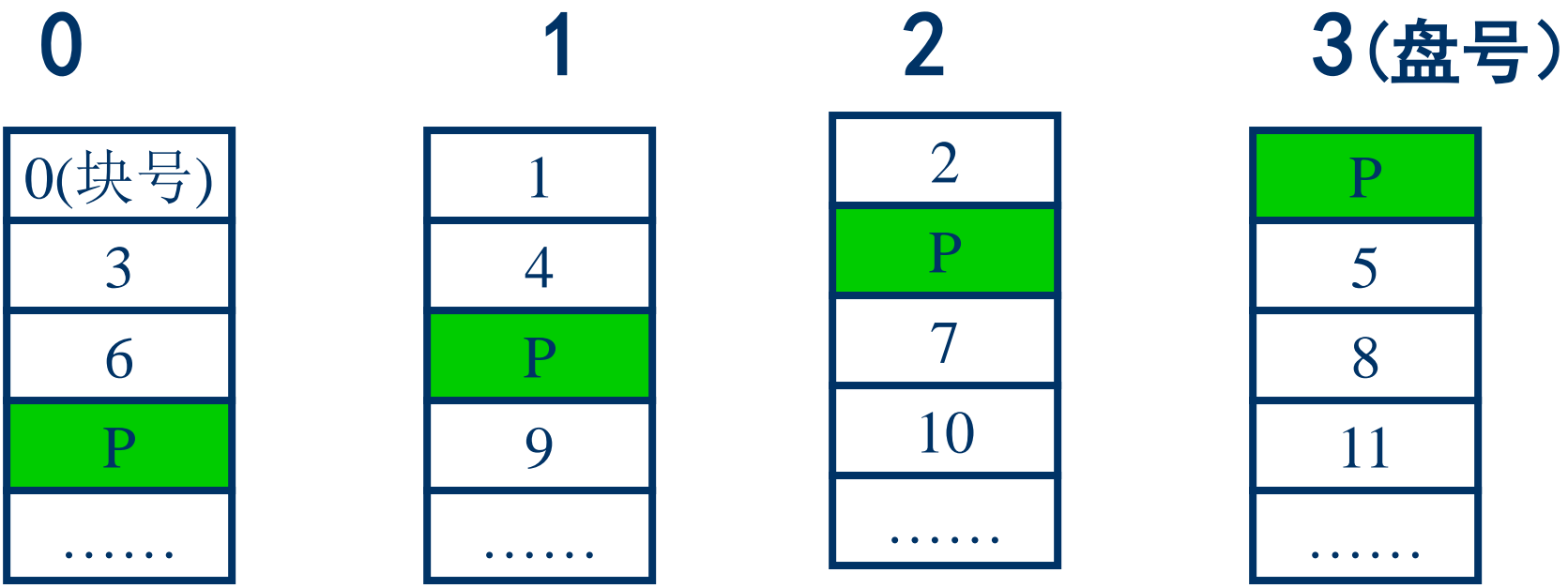
主机I/O请求

命令分解



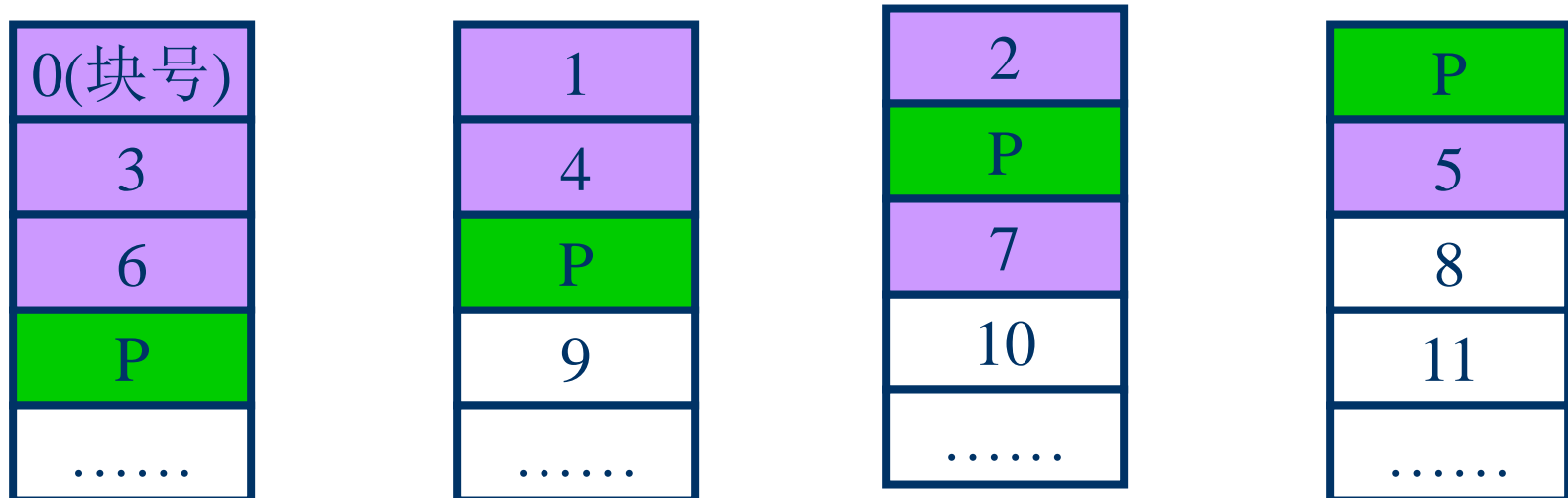
使用聚散技术的阵列控制流 程模型

RAID5命令分解过程:



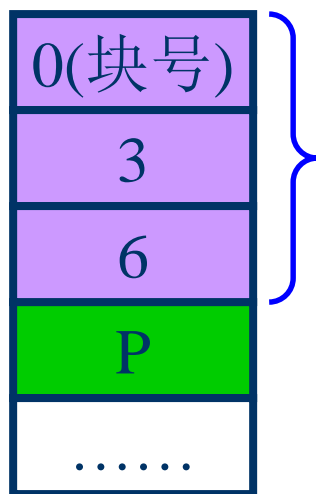
阵列参数：左不对称，4块盘，分块大小为2K

读命令： 0x08, 数据长度16k, 起始地址为0



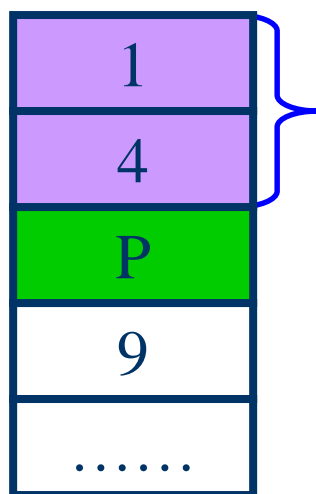
则要读的有效数据为图中的0~7块

0号盘上的子命令：



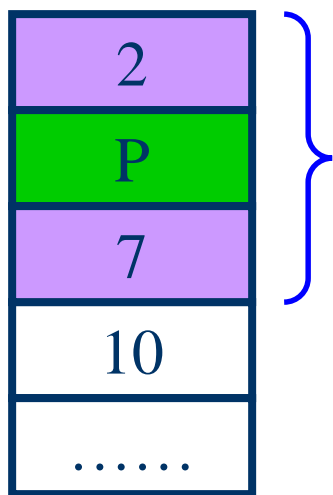
顺序读出0，3，6块，可以合并为一个读命令，数据长度为6k

1号盘上的子命令:



顺序读出1, 4块, 可以合并为一个读命令, 数据长度为4k

2号盘上的子命令：



读2号块和7号有效数据块，为了减少I/O次数，把它们合并再一起读，把中间夹的校验块也读出来，数据总长度为6k，有效的为4k

3号盘上的子命令：

P
5
8
11
.....

} 只有5号块，长度为2k

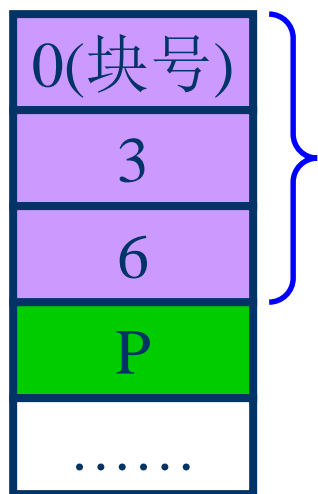
RAID5 块大小2KB

写命令： 0x0a, 数据长度16k， 起始地址为0

	0(块号)		1		2		P	满条
	3		4		P		5	满条
	6		P		7		8	大写
	P		9		10		11	
	

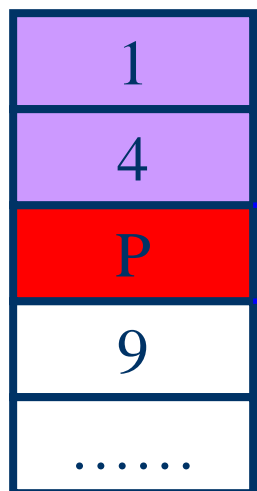
则要写的有效数据为图中的0~7块

0号盘上的子命令：



顺序写出0，3，6块，可以合并为一个读命令，数据长度为6k

1号盘上的子命令：

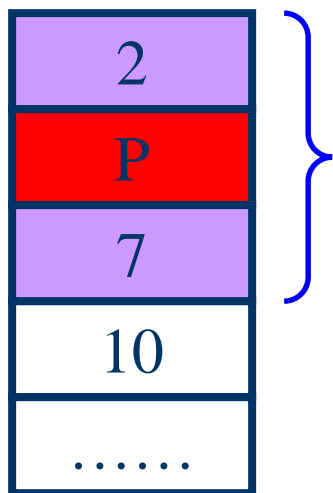


顺序写1，4块，

计算校验命令；
写新校验命令，它可以和上面的
写合并，但要放在计算校验之后

1. 计算校验P
2. 写数据和校验命令（合并为1条命令）

2号盘上的子命令：



1. 计算校验
2. 写2, P, 7块

1. 计算校验P
2. 写数据和校验命令(合并为1条命令)

3号盘上的子命令：

P
5
8
11
.....

2. 计算校验命令

3. 写数据和校验（合并为1命令）

1. 读旧的数据命令

写命令： 0x0a, 数据长度38k, 起始地址为0

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

则要写的有效数据为图中的0~18块

小写的过程(Read-Modify-Write, RMW)

- 读第18号块的旧数据
- 读图中红色的旧校验数据
- 计算校验: $P = D18_{old} \oplus D18 \oplus P_{old}$
- 写入新的数据D18和新的校验P

- $P_{old} = D1 \oplus D2 \oplus D3 \oplus \dots Dn$

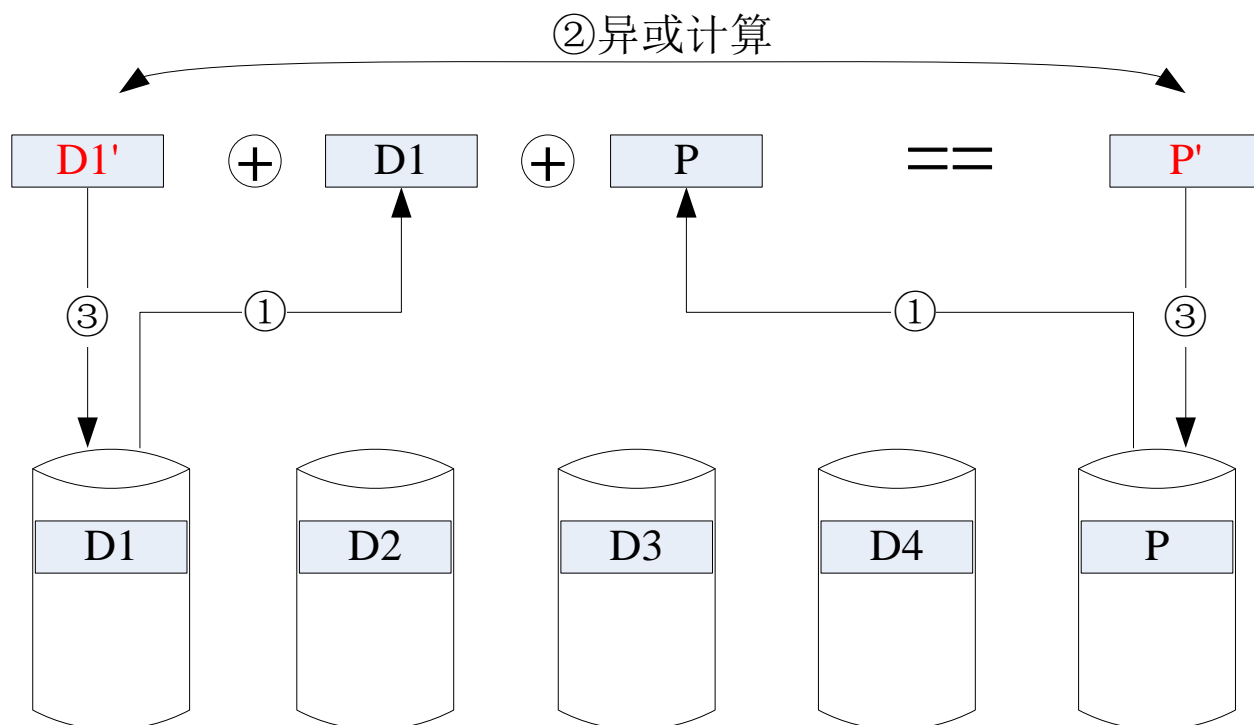
$$P = D1 \oplus D1' \oplus P_{old}$$

$$P = \text{D1} \oplus D1' \oplus \text{D1} \oplus D2 \oplus D3 \oplus \dots Dn$$

$$P = D1' \oplus D2 \oplus D3 \oplus \dots Dn = P_{new}$$

RAID5 Write Hole Problem

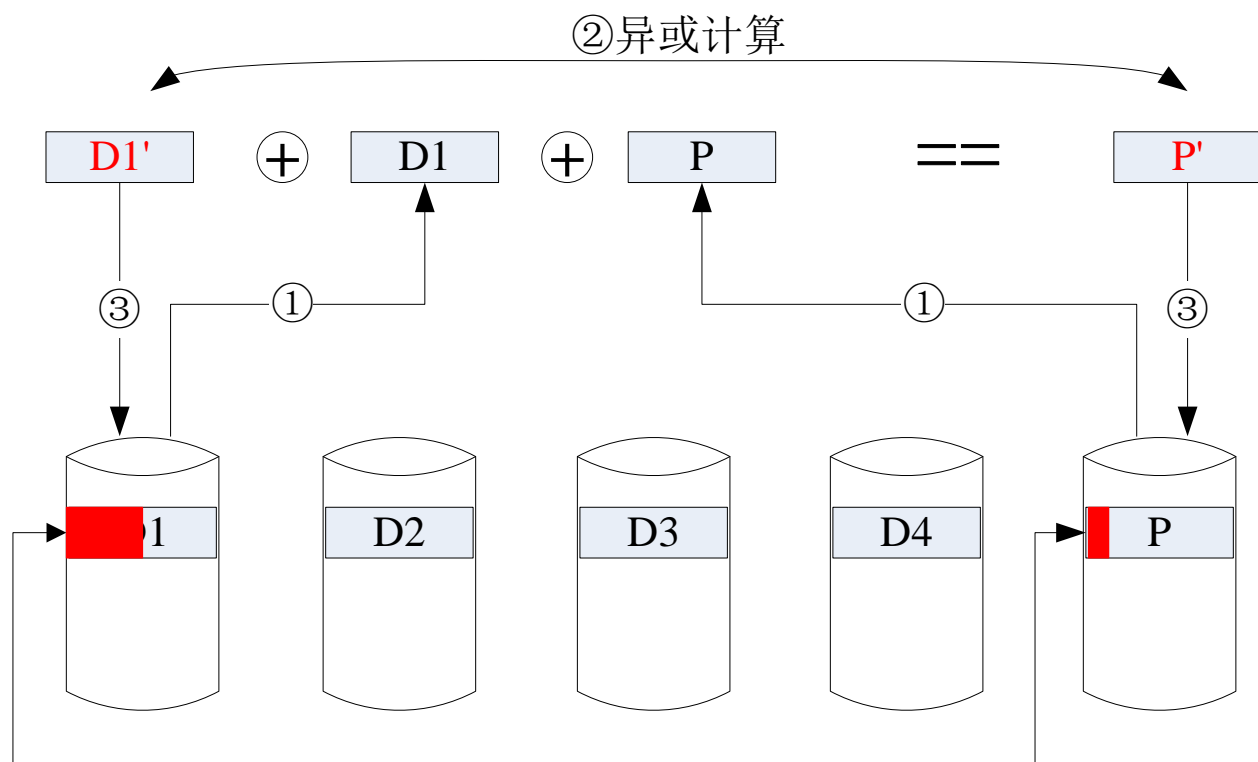
- 奇偶校验盘阵列（RAID5/6）中校验数据丢失（污染）问题（Parity Pollution）



- ①：读取旧数据 D1，旧校验 P；
- ②：异或计算得到新的校验值 P'；
- ③：写入新数据 D1' 和新校验值 P'

RAID5 Write Hole Problem

● 异常状态下（突然掉电）小写请求处理流程

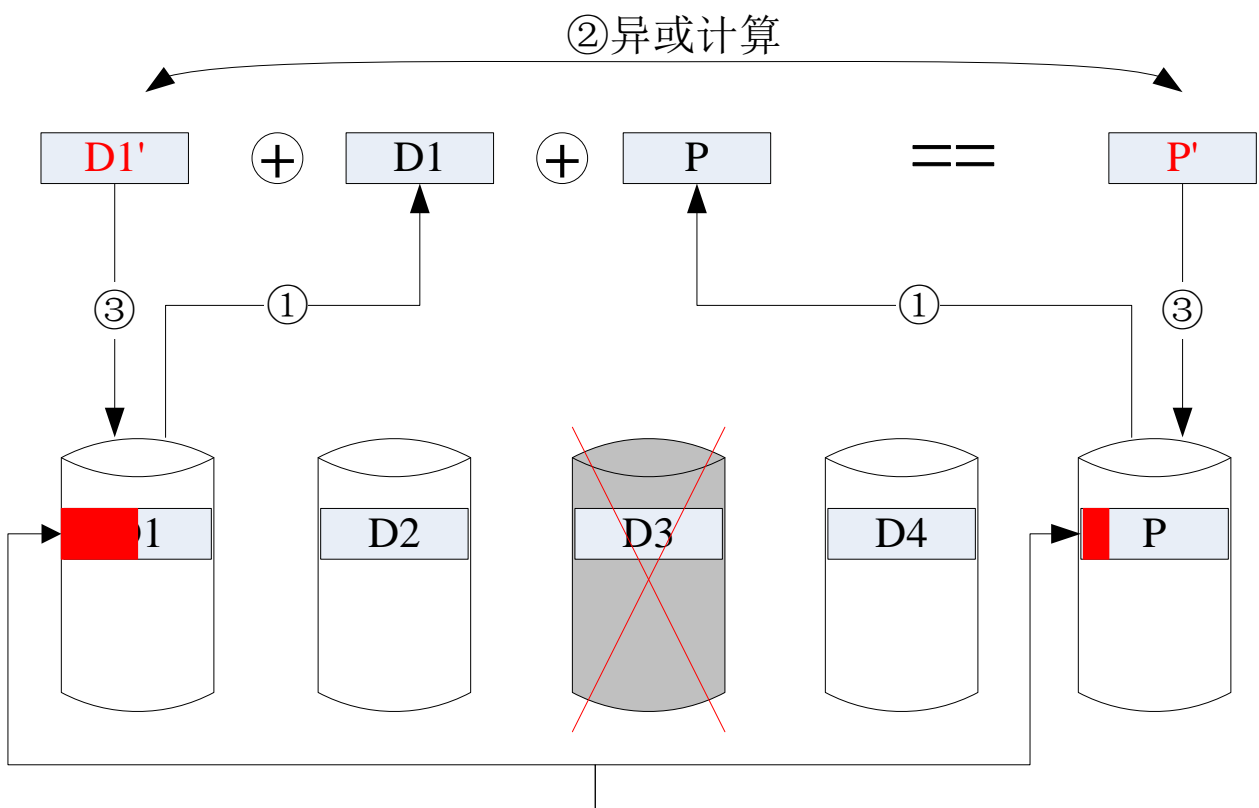


④系统断电，数据部分写入，且两个盘中写入的数据长度不一定相等（校验不一致）

⑤：系统重启后，再次写入D1'，此时只有RCW (Read-Reconstruct-Write)机制才能恢复数据，否则会导致条带不一致现象，进而会丢失数据（再次发生磁盘故障）；

RAID5 Write Hole Problem

● 降级模式下（突然掉电）小写请求处理流程



⑤：系统重启后，
如何恢复？？？



④系统断电，数据部分写入，且两个盘中写入的数据长度不一定相等（校验不一致）

课堂交流报告入选题目

序 号	题 目	成 员
1	非易失性存储器概述	马玮良、张晓辉、熊倩、彭达
2	非易失性存储器在主存和存储系统中应用的软件技术综述	黄维周、胡广行、周焜
3	忆阻器交叉阵列的非理想特性与解决方案	朱蔚霖、王舒虹、吴登辉
4	键值存储发展研究综述	余瑞、周翔、郑媛方
5	针对新型非易失存储器寿命问题的研究	张兴锐、黄创、董升育
6	基于ReRAM的新型体系结构的调查	辛杰、王可、邓运洋
7	区块链存储模型概述	王超凡、韩睿、熊浩辰
8	SSD新型技术综述	钟芳郅、零贤亮、徐阳
9	分布式存储系统文献综述	王程锦、徐心兰、郑银婷
10	存储可靠性研究综述	徐佳、徐伟光
11	数据去重相关技术研究综述	鲍瑞祺、程欢、周恒
12	分布式存储的一致性协议综述	黄炜宸、刘晋通、董子豪

RAID5的重建与初始化

- **重建(Rebuild):** 读出非故障盘上的数据, 按照校验信息的计算方法计算得到故障盘上的数据, 然后写入到新替换的盘上
- **重构 (Reconstruct) :** 重建的微观操作过程

校验信息的正确完整性问题

- 磁盘上初始信息是随机的
- 满条块写和大写 (RCW) 可以保证校验信息的正确性
- 若原始校验信息不正确，则小写不能保证校验信息的正确，例如：

1	0	1	0	1	1	1	0	(正确值1)
0	0	1	0	1	1	1	?	(应该是0)

根据小写计算得： $0 \oplus 1 \oplus 0 = 1$

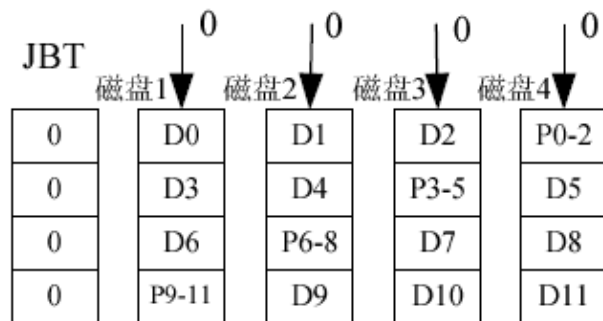
若刚写入的盘出现了故障，需要数据恢复：

恢复的数据 = $0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1$

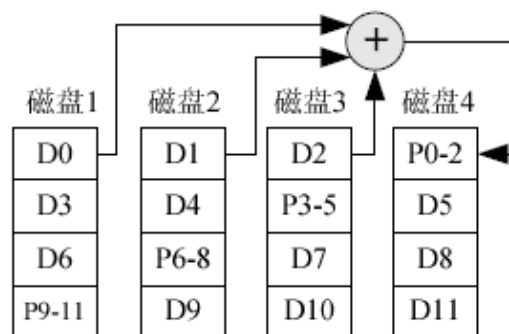
依据错误的校验信息进行重建得出的数据也是不正确的，
所以需要**初始化**过程

RAID5 初始化过程

- 原则：保证校验信息一致正确
- 方法：
 - 全部写“0”
 - 计算校验



(b) 改进的数据同步过程



(a) 传统的数据同步过程

RAID高可靠、高可用技术

评估磁盘阵列可用性的方法

- 从**SLA**(Service-Level Agreement)角度看，系统在两种状态间切换：
 - 服务完成，即服务按照指定的SLA交付；
 - 服务中断，即提交的服务不满足SLA的要求。
- 评估指标：
 - 可靠性：评估服务完成，指从初始状态起，持续完成服务的时间，一般用**MTTF**评估；
 - 可用性：也评估服务完成，当系统从“完成”切换到“中断”时，同样认为系统可用性降低；传统的磁盘阵列可用性形式化定义为**MTTF**和**MTTR**的函数：

$$\frac{MTTF}{MTTF + MTTR} \times 100\%$$

RAID高可靠、高可用技术

评估磁盘阵列可用性的方法（续）

- 可用性的其它定义方法
 - 系统除了“运行”和“宕机”两个状态外还存在着许多“降级”状态，衡量可用性必须捕获这些“降级”状态，除了测量系统是“运行”或“宕机”外还要评估它所提供服务的质量
 - 可用性的定义不能仅仅局限于某个时间点而应该是全部时间内系统服务的质量，系统的可用性评估还需要测量全部时间内系统**服务质量**的变化
- 适用于大部分服务器系统的两个典型标准是“**容错能力**”和“**性能**”，即存储子系统能够容忍的故障数和每秒能响应的请求数（或请求的响应时间）

RAID高可靠、高可用技术

提高磁盘阵列可用性的相关技术

- 磁盘阵列数据重建算法研究
 - 数据布局的重新组织
 - 重建工作流优化
 - 重建顺序优化
 - 理论研究及其它
- 提高校验磁盘阵列小写性能的相关技术
 - 改变磁盘阵列的数据布局或校验更新方法
 - 缓存技术

RAID高可靠、高可用技术

问题：

- 存储系统规模变大，磁盘故障成为经常性事件，**RAID重建过程过长**，系统可靠性成为问题
- 磁盘阵列在线重建时，用户应用请求和重建请求争夺磁盘资源，**用户应用请求将延长重建时间**，降低重建效率，而重建又会影响用户应用请求，降低用户应用的性能

基于热度的多线程重建调度优化算法PRO

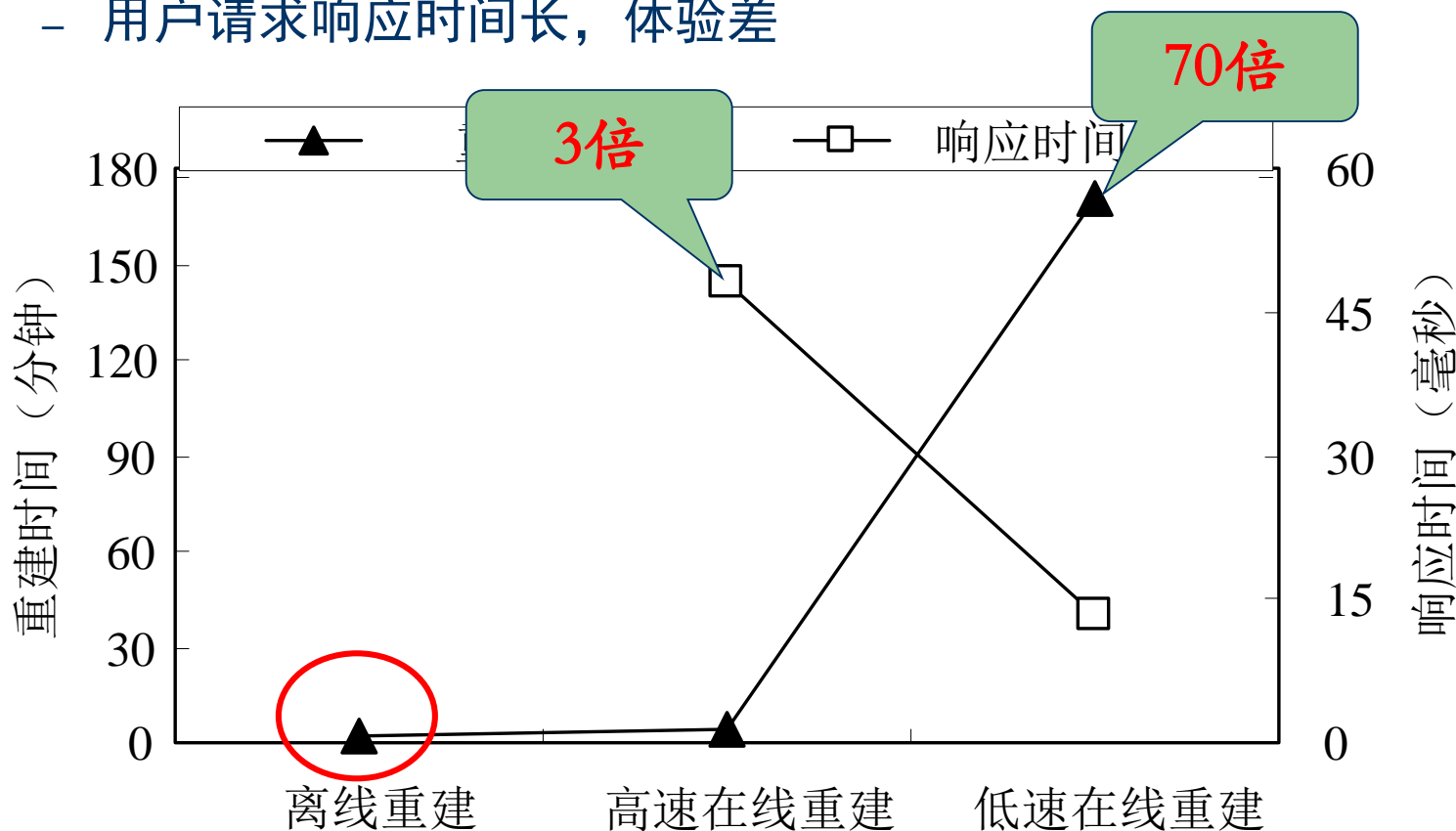
——FAST' 07

基于I/O负载重定向的重建优化算法WorkOut

——FAST' 09

重建过程的优化方法（PRO）

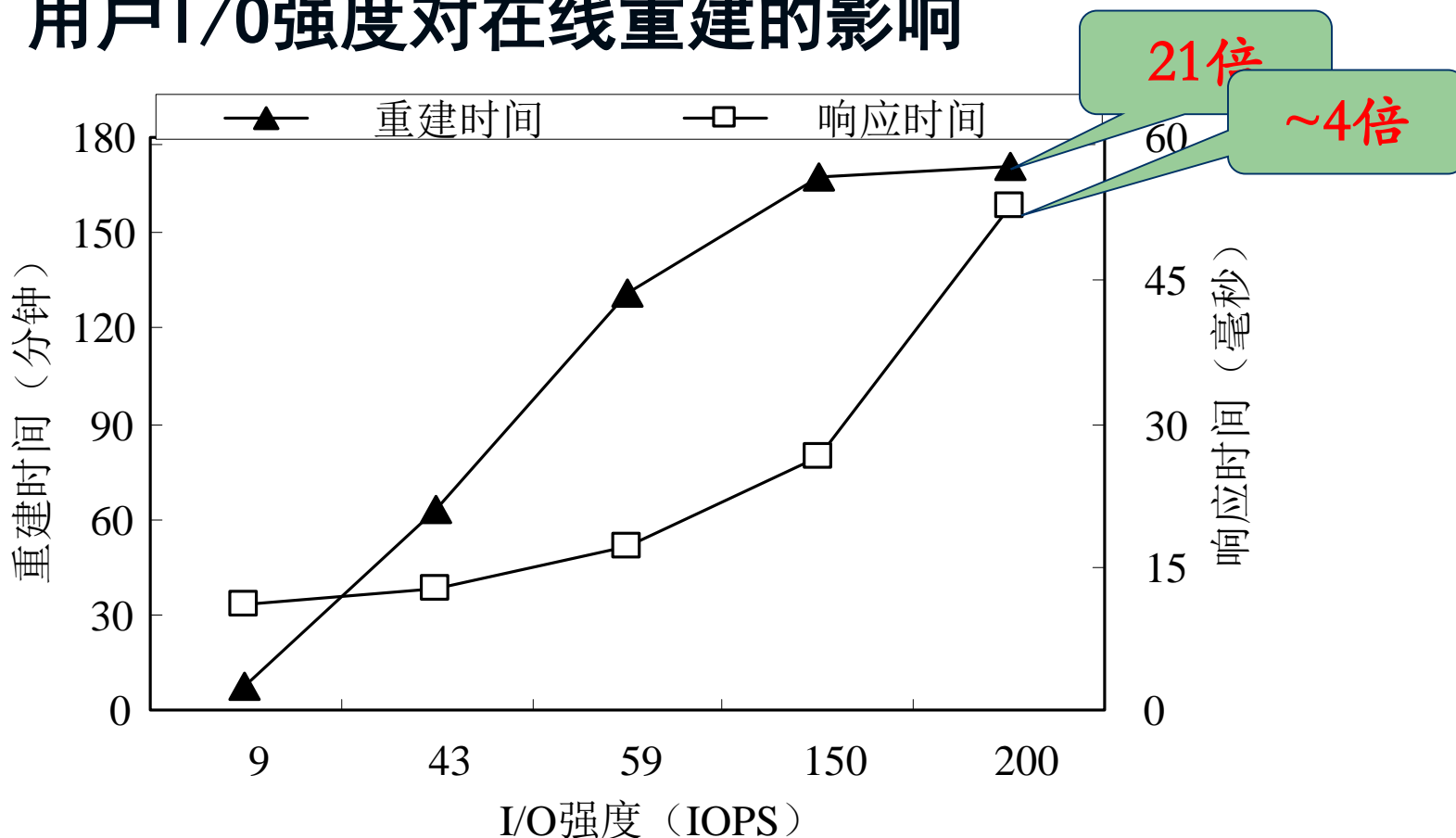
- 重建面临的挑战：重建I/O与用户I/O争用资源
 - 时间长，速度慢
 - 用户请求响应时间长，体验差



重建I/O和用户I/O的相互影响

重建过程的优化方法（PRO）

用户I/O强度对在线重建的影响



重建时间和用户I/O响应时间都随着用户请求的IOPS（每秒的I/O数）的增加而增加

重建过程的优化方法（PRO）

- 重建面临的挑战：重建I/O与用户I/O争用资源
 - 时间长，速度慢
 - 用户请求响应时间长，体验差
- 解决办法1
 - 利用20—80原理
 - 热点数据区优先重建，减少磁头移动
 - 减少IO请求的延迟，加快重建速度

Lei Tian, et al. "PRO: A Popularity-based Multi-threaded Reconstruction Optimization for RAID-Structured Storage Systems". In Proceedings of the 5th USENIX Conference on File and Storage (FAST'07). pp. 277~290, 2007

解决办法2:

I/O负载重定向的重建算法 (WorkOut)

- 基本思想

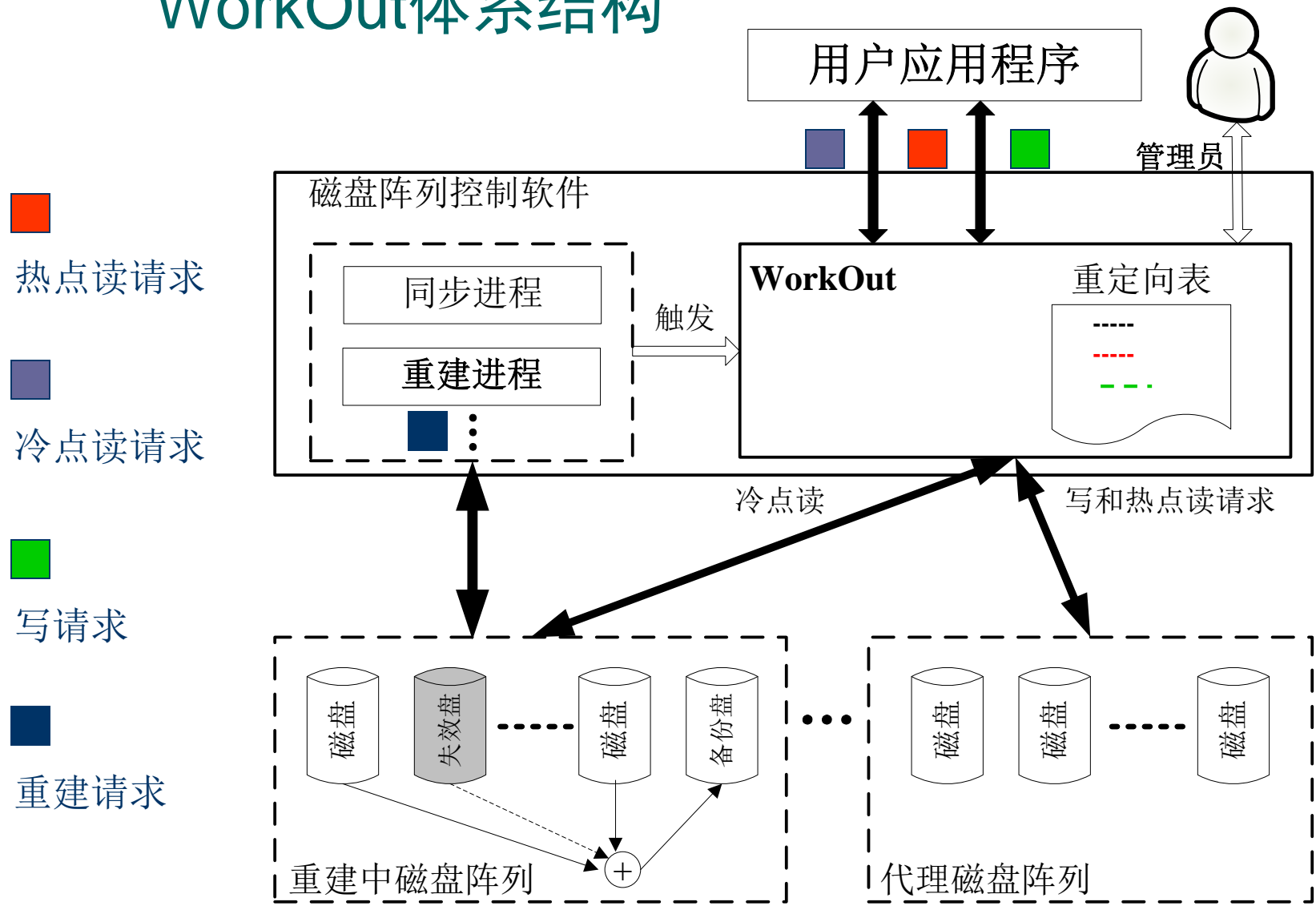
- 通过将重建中磁盘阵列收到的所有写请求和热的读请求重定向到代理磁盘阵列中，从而减少重建中磁盘阵列的用户I/O负载强度，使重建进程得到更多的磁盘资源，进而提高了重建效率

- 目标

- 提高数据中心的存储系统的可靠性和可用性。即通过I/O负载重定向技术减少磁盘阵列的数据恢复时间，同时减少重建进程对用户I/O响应性能的影响

Suzhen Wu, et al. **Improving Availability of RAID-Structured Storage Systems by Workload Outsourcing**. *IEEE Transactions on Computers*. 60(1):64-79, 2011

WorkOut体系结构



解决办法3:

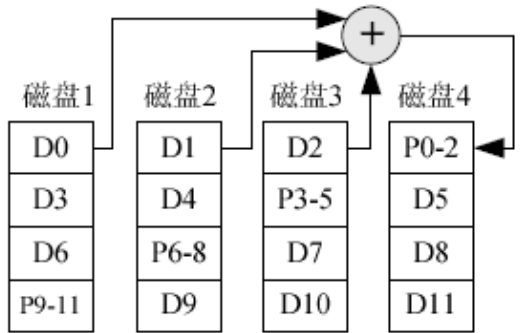
基于日志的重建

● 原理:

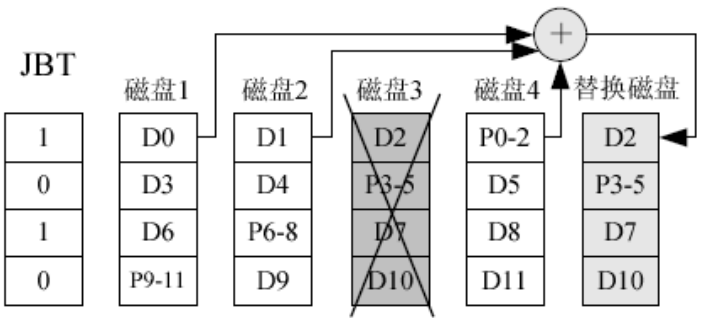
- 初始化全部写0
- 存储空间的空闲达50%，相当一部分空间一次都没修改过
- 记录修改过的条块，重建的时候，对修改的条块重新计算，未修改的部分直接写0

Suzhen Wu, et al. JOR: A Journal-guided Reconstruction Optimization for RAID-Structured Storage Systems. In Proceedings of the 15th International Conference on Parallel and Distributed Systems (ICPADS'09)

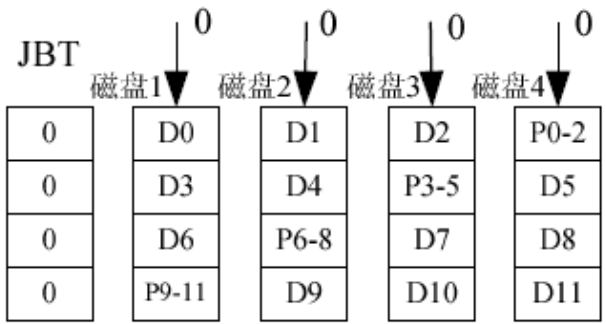
解决办法3：
基于日志的重建



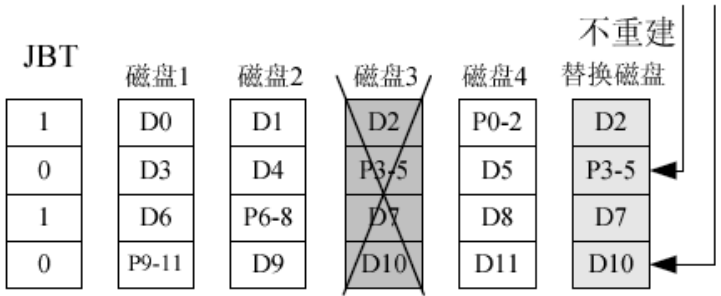
(a) 传统的数据同步过程



(a) 当bitmap=1时的重建过程



(b) 改进的数据同步过程



(b) 当bitmap=0时的重建过程

企业需求实例：

1.2TB的数据，30分钟内要求完成重建：

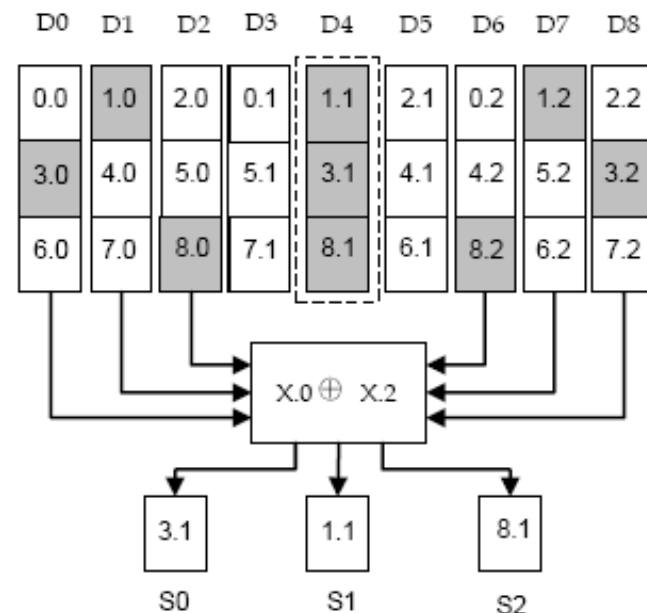
$$1.2\text{TB}/30*60\text{s} = 667\text{MB/s}$$

解决途径：

多设备并行重建

S²-RAID: Parallel RAID Architecture for Fast Data Recovery

- The idea is to divide each large disk in the RAID into small partitions. Partitions on these disks form sub-arrays. The sub-arrays are skewed among the disks in the RAID in such a way that conflict-free parallelism is achieved during a RAID reconstruction when any disk fails.
- Recovered data that was on the failed disk is stored in parallel on multiple disks consisting of spare disks and available space of good disks.



Jiguang Wan, Jibin Wang, Changsheng Xie, Qing Yang, "S²-RAID: Parallel RAID Architecture for Fast Data Recovery," *IEEE Transactions on Parallel and Distributed Systems*, 20 Nov. 2013.

RAID for SSD

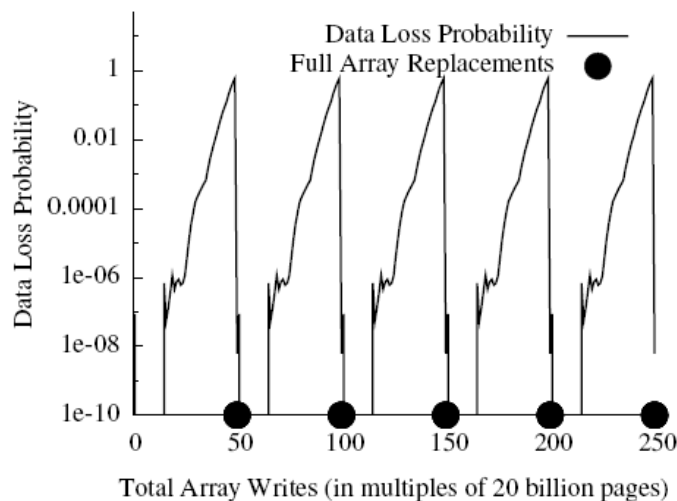
特点:

- SSD擦写次数受限
 - SLC: 100,000次, MLC: 5,000-10,000次
- RAID中数据尽量均匀分布到各成员盘
 - RAID0, RAID5
- RAID5中校验数据块的擦写更频繁
 - N-device RAID-5, 任何一个成员盘的数据更新都会触发校验数据更新

RAID for SSD

问题：

- 成员盘故障集中爆发
 - 多个SSD会同期达到写入次数极限，发生不可恢复的bit err
- 数据不可恢复
 - RAID5在修复过程中再次发生SSD故障的概率高
- 不仅仅RAID5才会出现
 - RAID1、10、6同样会出现



RAID for SSD

问题：

校验块数据为热点更新数据，导致介质损耗问题更为严重

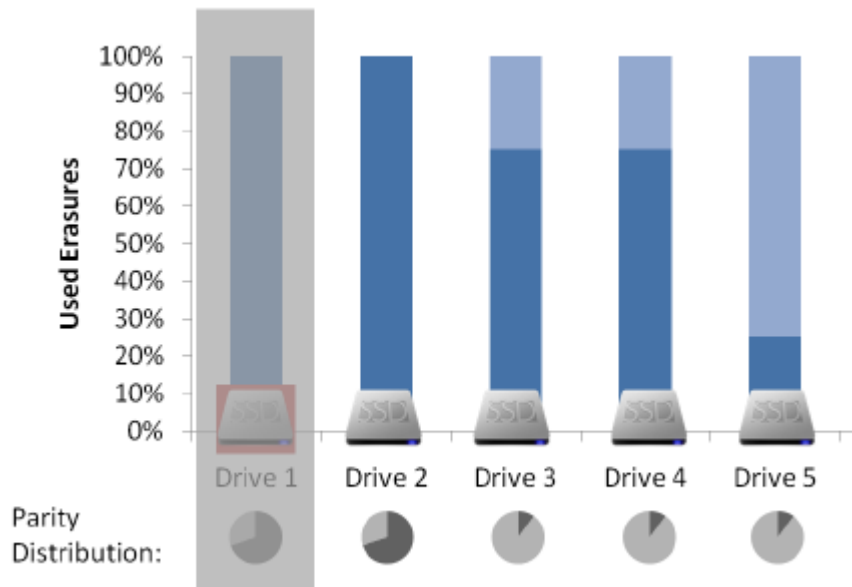
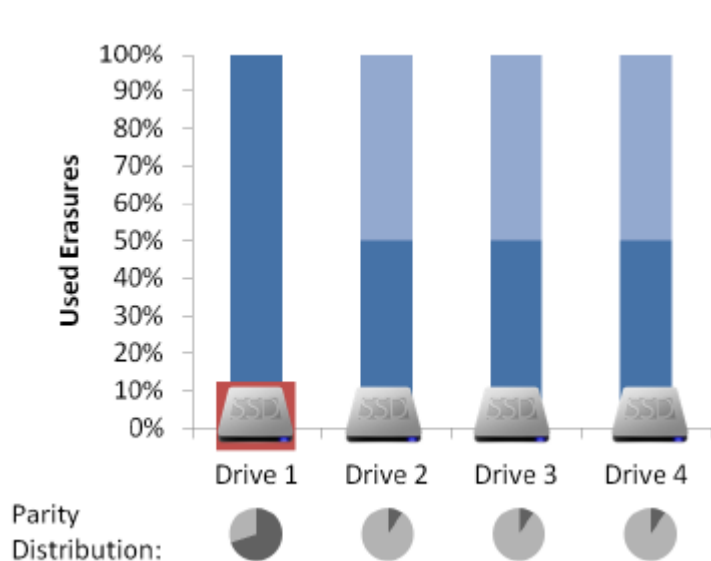
小写请求性能差伴随着校验更新导致小写性能更加严重

级别	可靠性	性能	价格
RAID0	低（无冗余）	中（小写性能问题）	低
RAID1/10	高（双冗余）	低（小写性能问题）	高（双冗余）
RAID5/6	低（校验信息频繁更新）	低（小写性能更加严重）	低
HPDA	高（镜像和校验保护）	高（基于日志的缓存）	低

RAID for SSD

Solution: **Differential RAID**

- 打破平衡，让校验信息非均匀分布



Mahesh Balakrishnan, Asim Kadav. Differential RAID: Rethinking RAID for SSD Reliability. In Proc. of EuroSys 2010.

混合式盘阵列

■ 磁盘

- ❑ 高能耗
- ❑ 非对称的顺序随机性能
- ❑ 无介质损耗问题

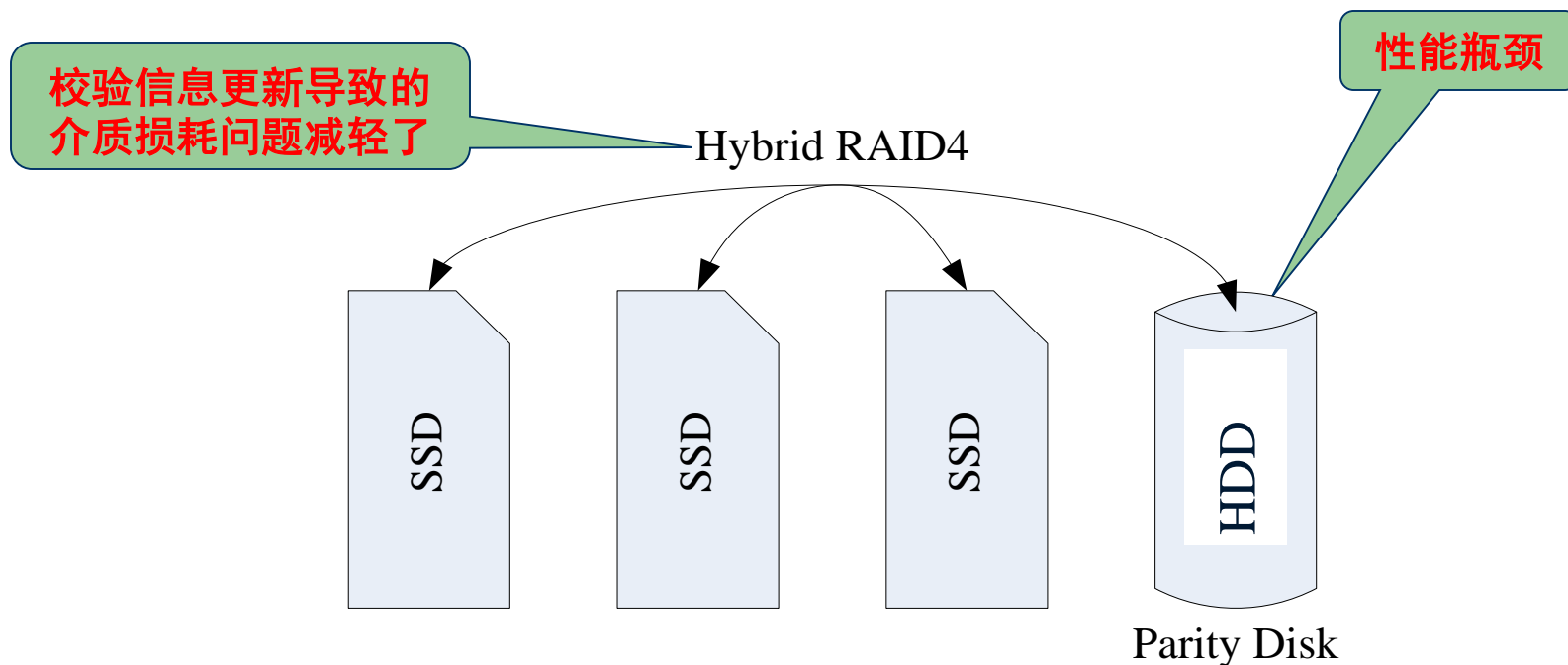


■ 固态硬盘

- ❑ 低能耗
- ❑ 非对称读写性能
- ❑ 介质损耗问题



混合式盘阵列

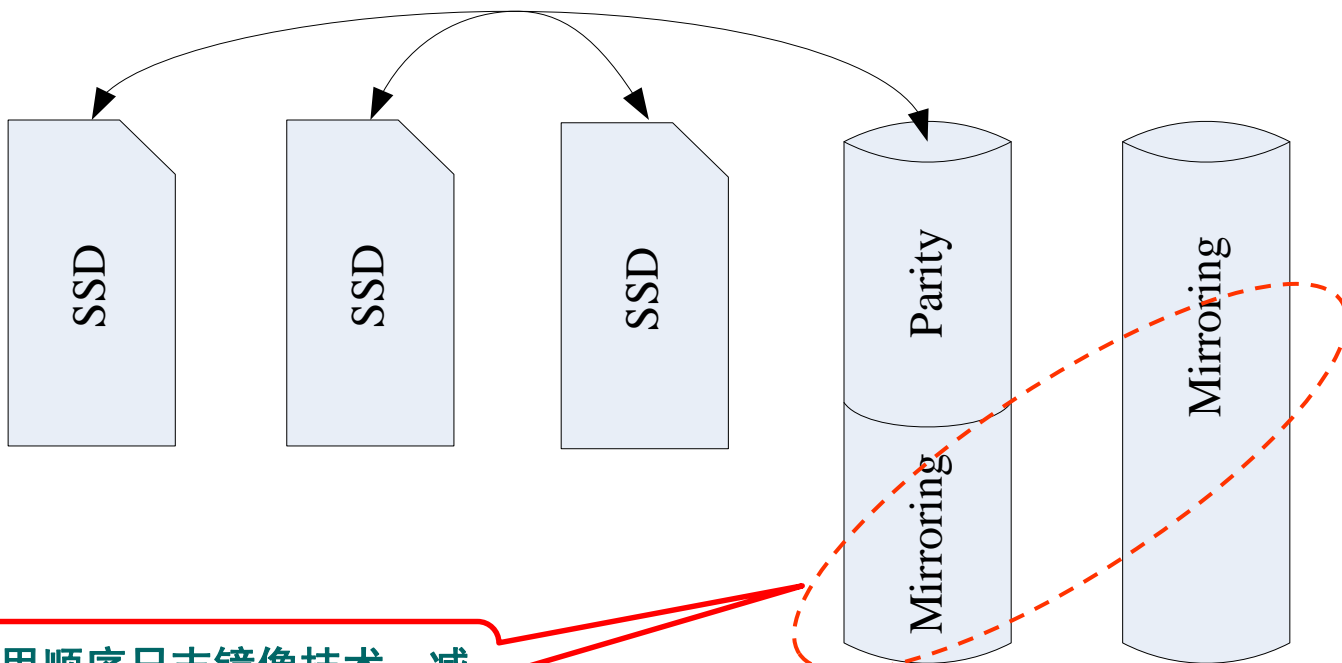


- 随机小写性能问题依旧存在
- 数据恢复过程更为漫长（读取磁盘）

混合式盘阵列

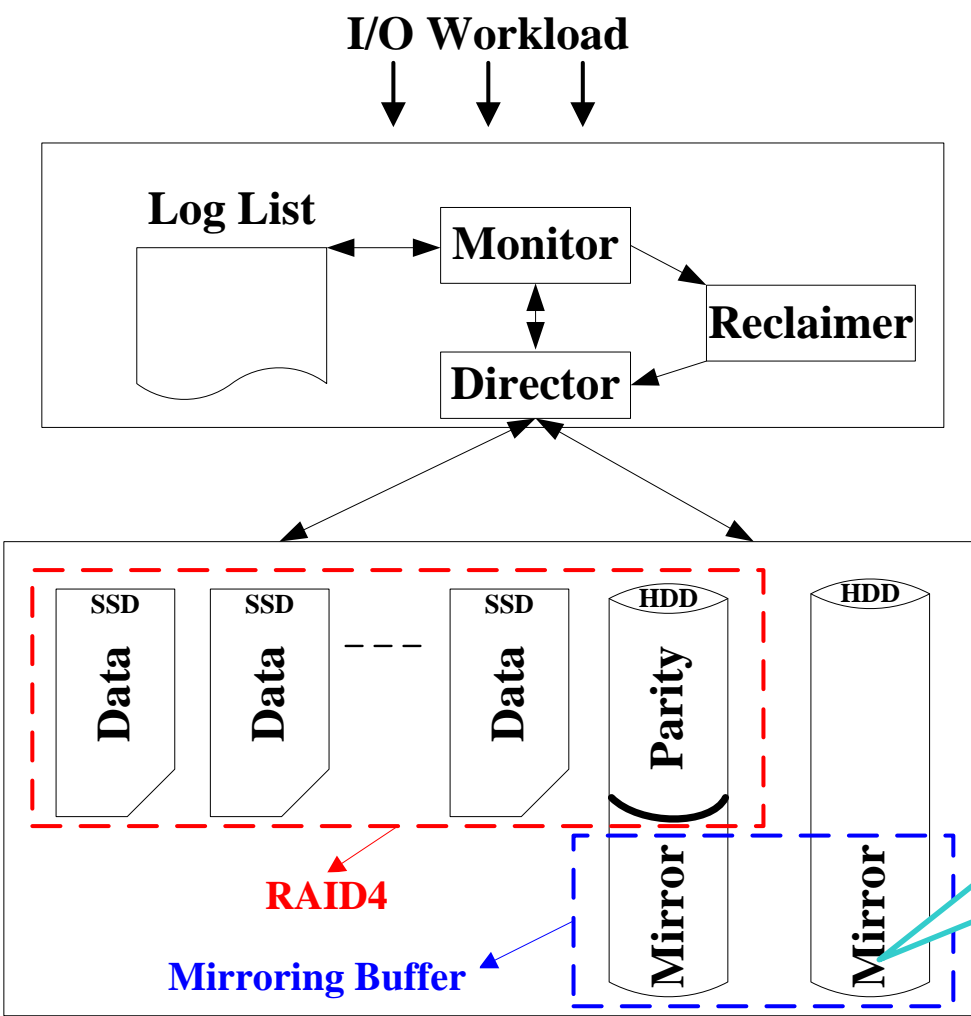
校验信息更新导致的
介质损耗问题减轻了

Hybrid RAID4



由于采用顺序日志镜像技术，减
轻了随机小写性能问题

混合式盘阵列



暂时缓存来自用户的随机写请求数据，并在数据恢复过程中充当代理空间加快盘阵列重建

A Case of Hybrid RAID: SSD + HDD

- **SSD :**
 - High performance for read, especially for random read
 - Small random write means low performance and wearing
- **HDD**
 - Good performance for sequential read and write
 - Very low performance for random read and write
 - Almost no wearing

	Random read	Sequent read	Random write	Sequent write	wearing
SSD	High	Very high	Low	High	Yes
HDD	Low	High	Low	High	No

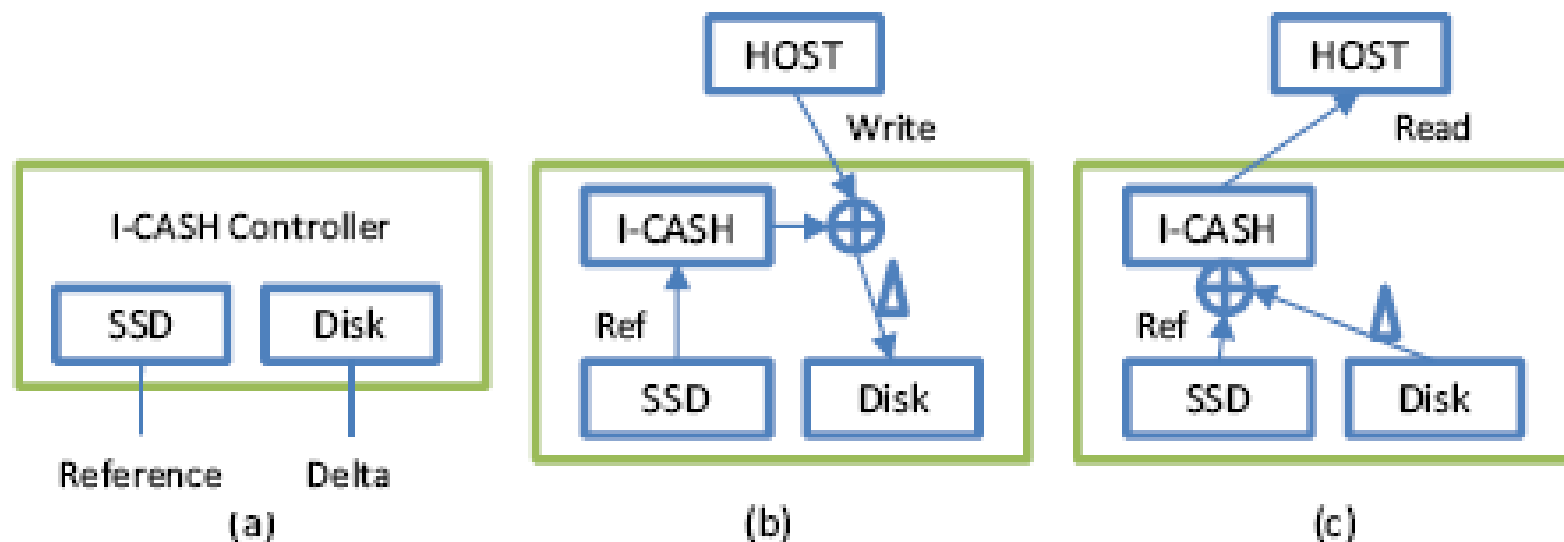
Content Locality:

Recent research literature has reported strong content locality in many data intensive applications with only 5% to 20% of bits inside a data block being actually changed on a typical block write operation

A Case of Hybrid RAID: SSD + HDD

I-CASH: Intelligently Coupled Array of SSD and HDD

- SSD stores seldom-changed and mostly read reference data blocks
- HDD stores a log of **deltas** between currently accessed I/O blocks and their corresponding reference blocks in the SSD
- Store **deltas** in a compact form



A Case of Hybrid RAID: SSD + HDD

I-CASH: Intelligently Coupled Array of SSD and HDD

- Random writes are not performed in SSD during online I/O operations
- High speed read performance of reference blocks stored in SSDs
- Potentially large number of small deltas packed in one delta block stored in HDD and cached in the RAM
- **Exploit the fast read performance of SSDs and the high speed computation of modern multi-core CPUs to replace and substitute the mechanical operations of HDDs**
- Avoid runtime SSD writes that are slow and wearing

Selected Publications

- ISCA • ["TRAP-Array: A Disk Array Architecture Providing Timely Recovery to Any Point-in-time"](#) in The 33rd Annual International Symposium on Computer Architecture, 2006 (ISCA'06). Qing Yang, Weijun Xiao, and Jin Ren
- ISCA • ["DCD---Disk Caching Disk: A New Approach for Boosting I/O Performance."](#) The 23rd Annual International Symposium on Computer Architecture, Philadelphia PA May, 1996. (ISCA'96). Y. Hu and Qing Yang
- ISCA • ["Caching Address Tags: A technique to reduce chip area cost for on-chip caches."](#) The 22nd Annual International Symposium on Computer Architecture, Santa Margherita Ligure, Italy, June, 1995. (ISCA'95). H. Wang, T. Sun, and Qing Yang
- ISCA • ["A novel cache design for vector processing."](#) The 19th International Symposium on Computer Architecture, May 1992, pp. 362-371. Gold Coast, Australia. (ISCA'92). Qing Yang and Liping W. Yang
- HPCA • ["I-CASH: Intelligently Coupled Array of SSD and HDD"](#) in The 17th IEEE International Symposium on High Performance Computer Architecture, 2011 (HPCA'11), San Antonio, TX, Feb 2011. Jin Ren and Qing Yang
- HPCA • ["RAPID-Cache --- A Reliable and Inexpensive Write Cache for Disk I/O Systems"](#), in The 5th International Symposium on High Performance Computer Architecture (HPCA-5). Orlando, Florida. Jan. 1999. Y. Hu, Qing Yang, and T. Nightingale

存储器件与设备变迁



...

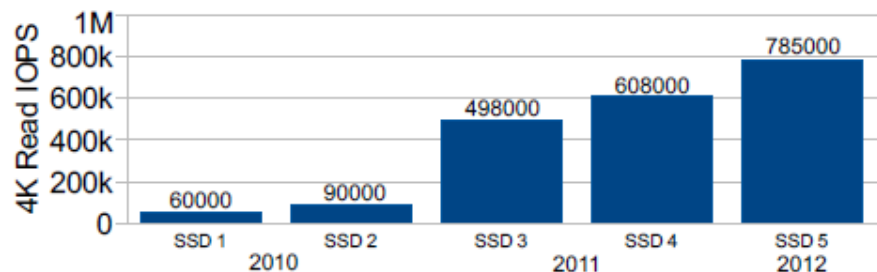
IOPS: 百级
带宽: 100MB/s
延迟: 若干毫秒 (ms)

IOPS: 千级
带宽: 数百MB/s
延迟: 低于1ms

IOPS: 百万级
带宽: x GB/s
延迟: 微秒级 (us)

新型固态硬盘:


- 采用NVM (Flash、3D Flash等)
 - 顺序/随机访问性能差距不再明显
 - 内部高并发, 高带宽, 高IOPS
 - 无机械部件, 低延迟 (us级)
- 需要高速物理接口 (PCIe)
- 需要高性能软件及协议
 - 优化IO软件栈
 - NVMe



2016年7月 国家集成电路产业 中国高端芯片联盟于 7 成员包括了集成电路产业链上重点企业及高校、研
业发展领导小组 月 31 日正式成立 究所等 27 家单位，如紫光集团、长江存储、中芯
办公室 国际、中国电子、华为、中兴、联想；高校研究所
则有中科院微电子所、工信部电信研究院、清华大
学、北京大学等。

2016年8月 国务院 “十三五”国家科技创新 基本形成核心电子器件、高端通用芯片和基础软件
规划 产品的自主发展能力，扭转我国基础信息产品在安
全可控、自主保障方面的被动局面；形成 28—14

全球首款：长江存储启动64层3D NAND闪存量产

 澎湃新闻
发布时间：09-02 20:40 | 上海东方报业有 限公司

9月2日，紫光集团旗下长江存储科技有限责任公司（长江存储）宣布，已开始量产基
于Xtacking架构的64层256 Gb TLC 3D NAND闪存，以满足固态硬盘、嵌入式存储等
主流市场应用需求。

作为中国首款64层3D NAND（三维闪存）闪存，该产品将亮相IC China 2019紫光集
团展台。Xtacking技术的研发成功和64层3D NAND闪存的批量生产，标志着长江存储
已走出了一条高端芯片设计制造之路。

作者最新文章

无缘武网决赛令李娜愿望落空，
科维托娃依旧值得为自己骄傲

最早登顶珠峰的中国攀登者们，
他们的信念和牺牲当被铭记

存储接口/协议变迁

传统存储接口

接口类型	传输方式	传输率	备注
IDE/PATA	并行	UDMA/33, UDMA/66, UDMA/100, UDMA/133	淘汰
SATA	串行	150MB/s, 300MB/s, 600MB/s (SATA 3.0)	
SCSI	并行	5MB/s, 10MB/s, ... 320MB/s	淘汰
SAS	串行	3Gbps, 6Gbps, 12Gbps, ...	
FC	串行	1Gbps, 2Gbps, ... 16Gbps, ...	

- 以上串行存储接口以SCSI协议为基础，以磁盘为主要存储设备

存储接口/协议变迁

SSD接口

➤ SATA/SAS

- 简单、兼容性好
- 带宽、延迟瓶颈
- AHCI (Advanced Host Controller Interface)



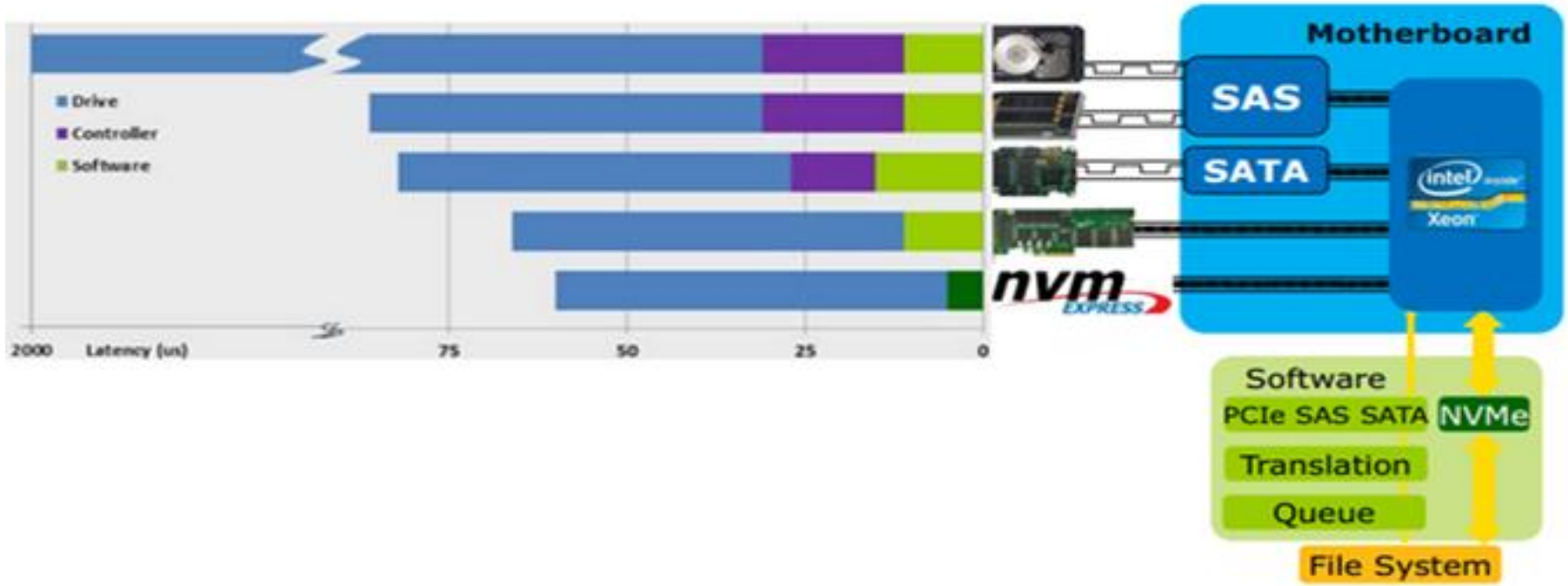
➤ PCIe

- 缩短存储访问物理通路
- 高带宽、低延迟
- NVMe (Non-Volatile Memory Express)



存储接口/协议变迁

SSD接口变迁



PCI Express* (PCIe) removes controller latency
NVM Express (NVMe) reduces software latency

NVMe接口/协议 NVMe命令集

- 所有命令都是 64 B
- Admin Command Set 与 NVM Command Set

Delete I/O Submission Queue
 Create I/O Submission Queue
 Get Log Page
 Identify
 Abort
 Set Features
 Get Features
 Asynchronous Event Request
 Namespace Management
 Firmware Commit
 Firmware Image Download
 Namespace Attachment
 I/O Command Set specic
 Vendor specic
 Format NVM
 Security Send
 Security Receive

Flush
 Write
 Read
 Write Uncorrectable
 Compare
 Write Zeroes
 Dataset Management
 Reservation Register
 Reservation Report
 Reservation Acquire
 Reservation Release
 Vendor Specic

SCSI命令集

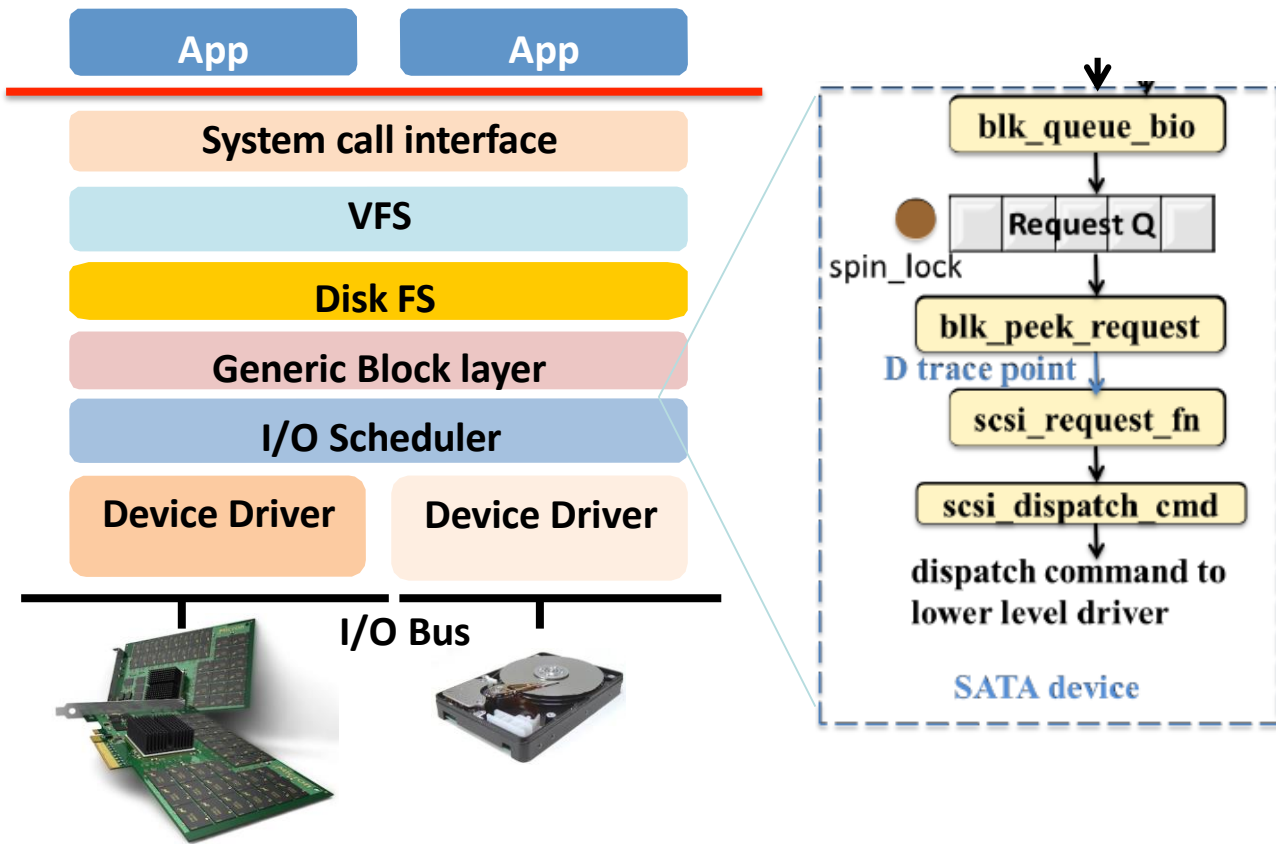
A SCSI to NVMe Translation Layer (SNTL) document was also created to define a mapping between NVMe and some SCSI commands.

SCSI/ATA



NVMe

传统I/O软件栈瓶颈



软件瓶颈:

➤ 请求队列锁

- 单一队列
- 队列操作需先上锁

➤ IO中断处理

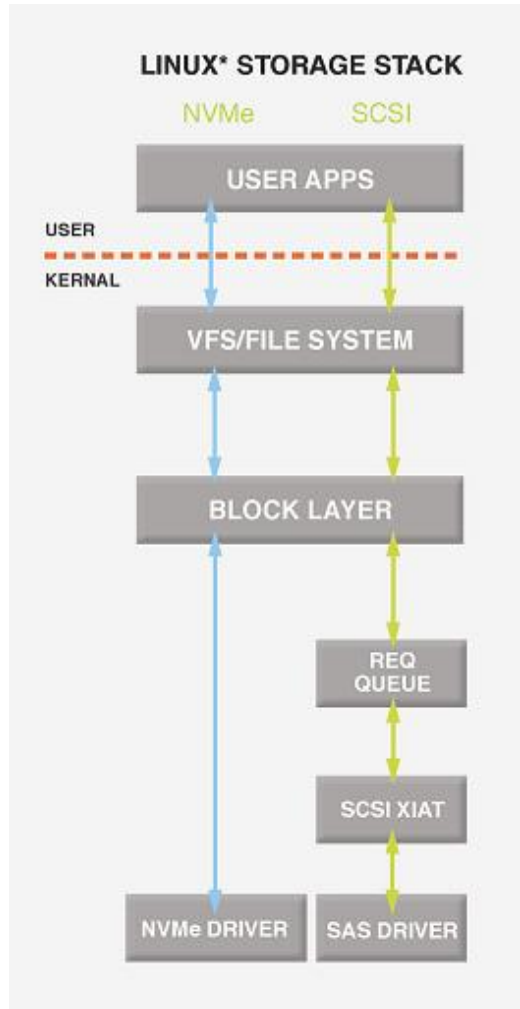
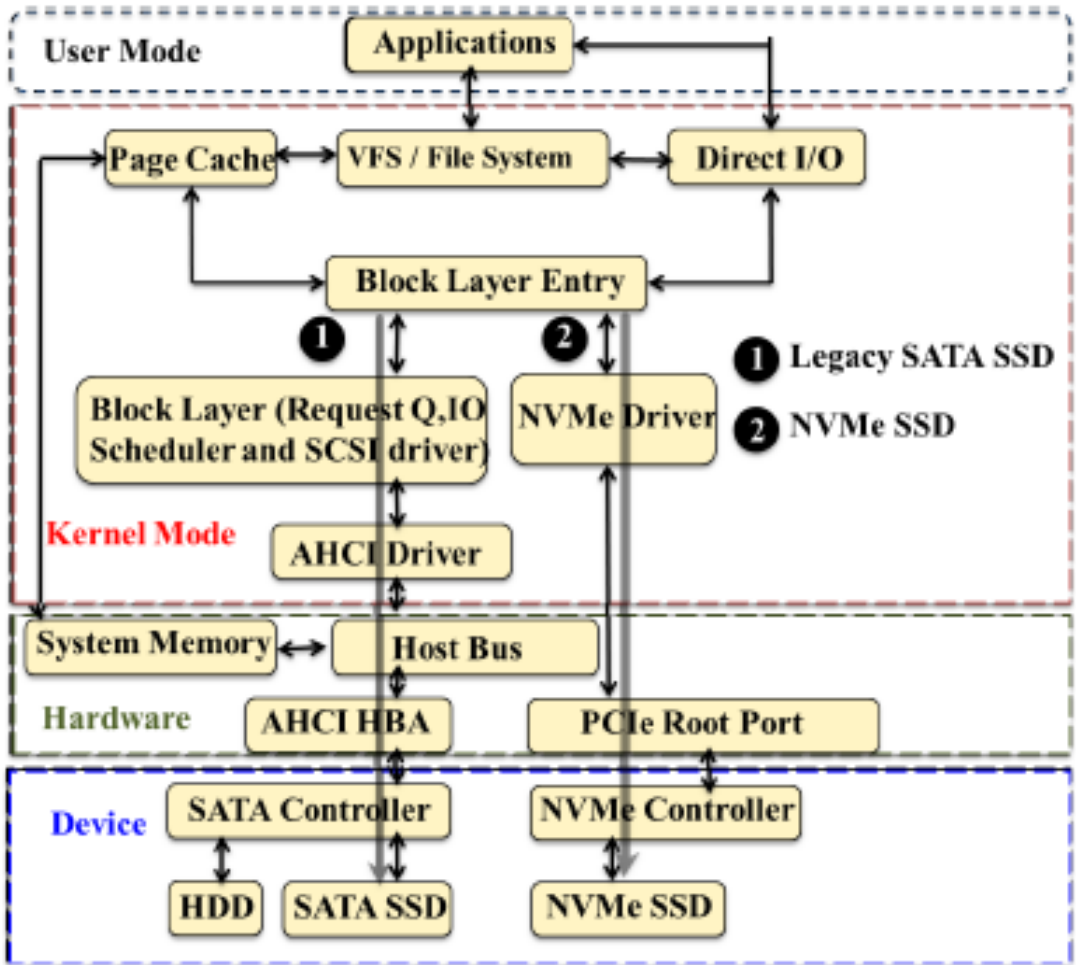
- 高IOPS意味着高中断数
- 在NUMA结构中中断开销更大

在传统Linux系统中，单一CPU核只能承受80万IOPS。不管使用多少核，传统块层受限于百万IOPS[1].

[1] Matias Bjørling, Jens Axboe, David Nellans, Philippe Bonnet. Linux Block IO: Introducing Multi-queue SSD Access on Multi-core Systems. SYSTOR '13.

NVMe接口/协议

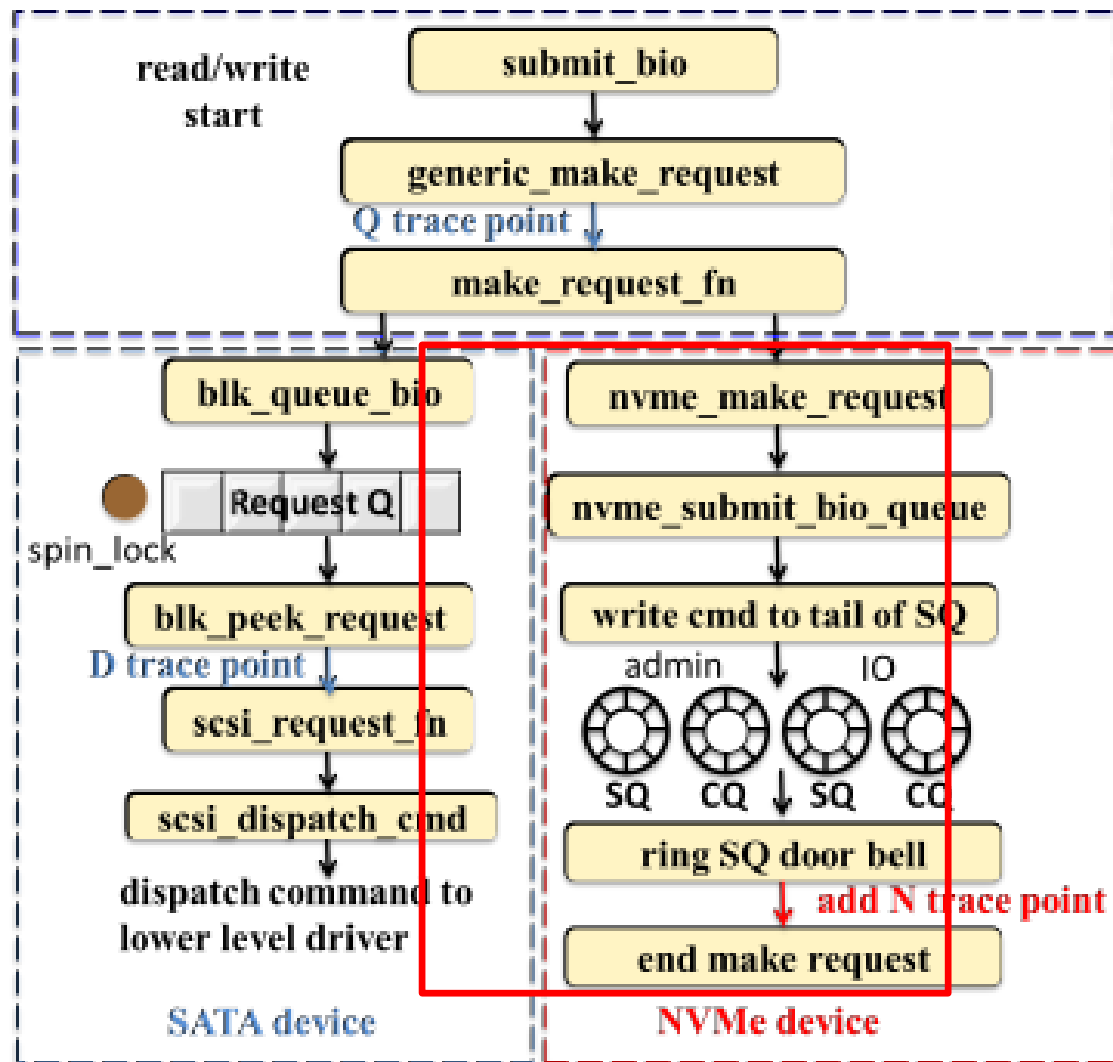
NVMe 精简I/O路径



[1] Qiumin Xu, Huzefa Siyamwala, Mrinmoy Ghosh. Performance Analysis of NVMe SSDs and their Implication on Real World Databases. SYSTOR '15.

NVMe接口/协议

NVMe 消除系统软件瓶颈



软件瓶颈:

➤ 请求队列锁

- 单一队列
- 队列操作需先上锁

➤ IO中断处理

- 高IOPS意味着高中断数
- 在NUMA结构中中断开销更大

NVMe接口/协议

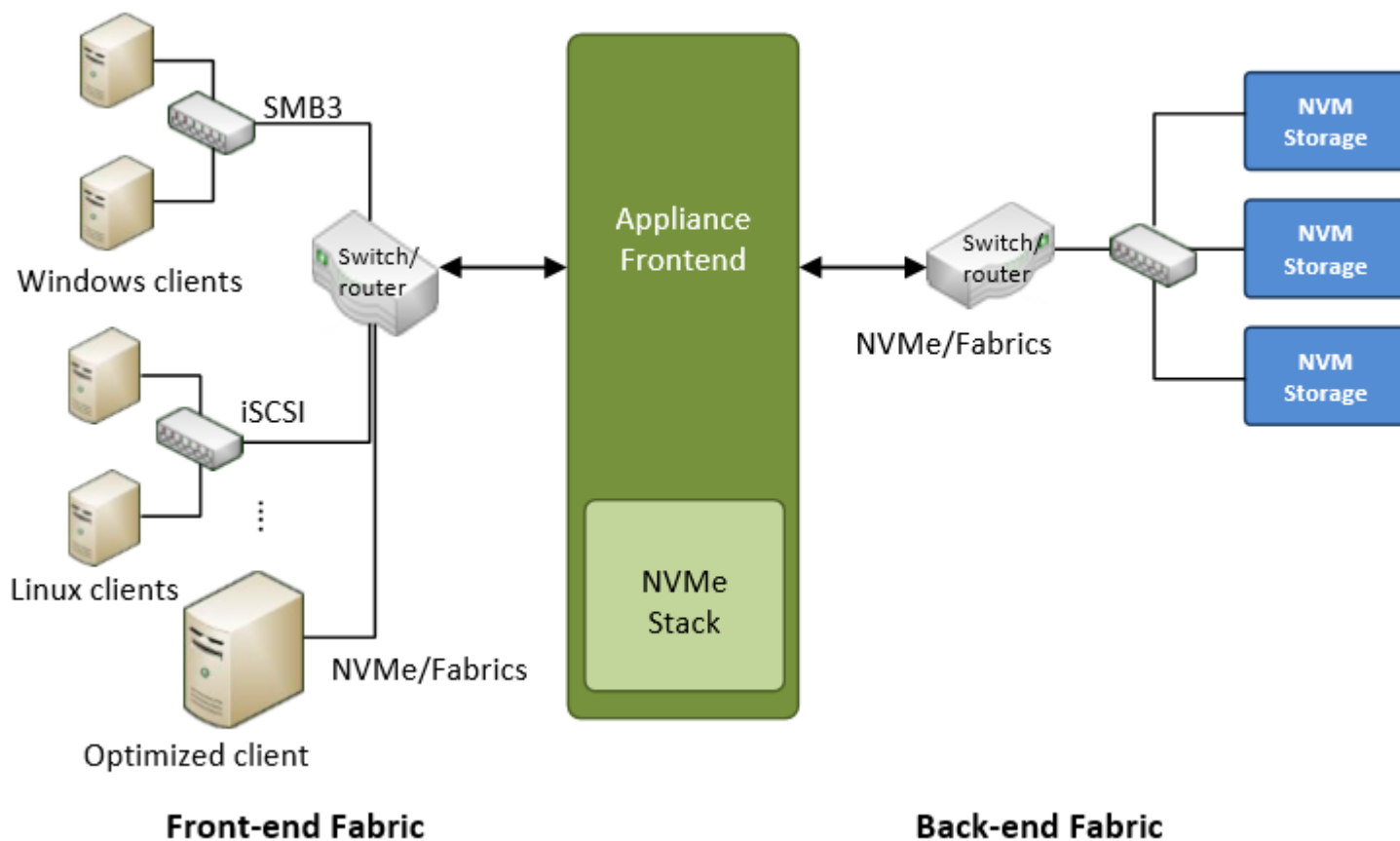
NVMe vs AHCI

	AHCI	NVMe
Maximum Queue Depth	1 command queue 32 commands per Q	64K queues 64K Commands per Q
Un-cacheable register accesses (2K cycles each)	6 per non-queued command 9 per queued command	2 per command
MXI-X and Interrupt Steering	Single interrupt; no steering	2K MSI-X interrupts
Parallelism & Multiple Threads	Requires synchronization lock to issue command	No locking
Efficiency for 4KB Commands	Command parameters require two serialized host DRAM fetches	Command parameters in one 64B fetch

- NVMe: 更低延迟，更高IOPS，更低功耗

NVMe接口/协议

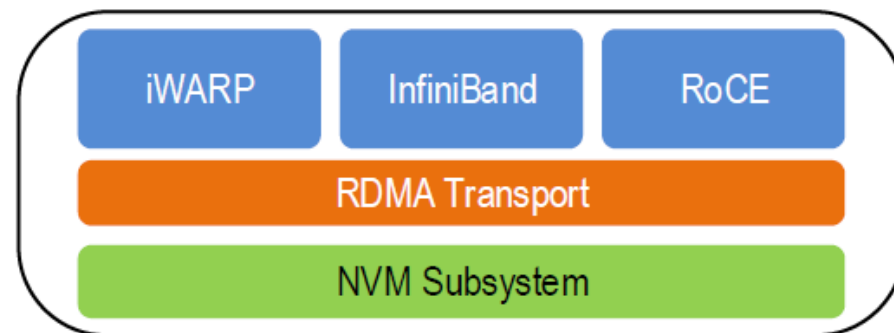
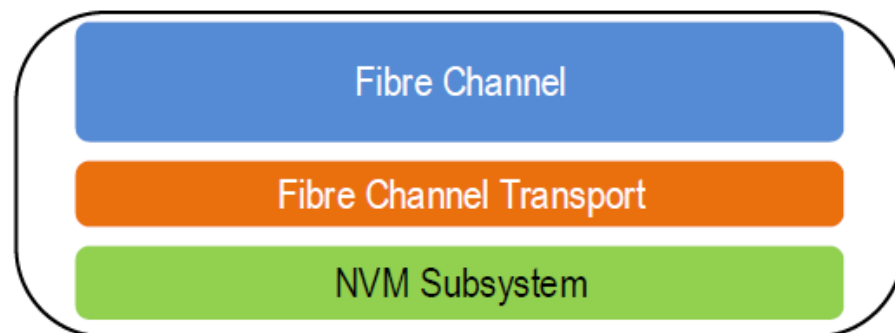
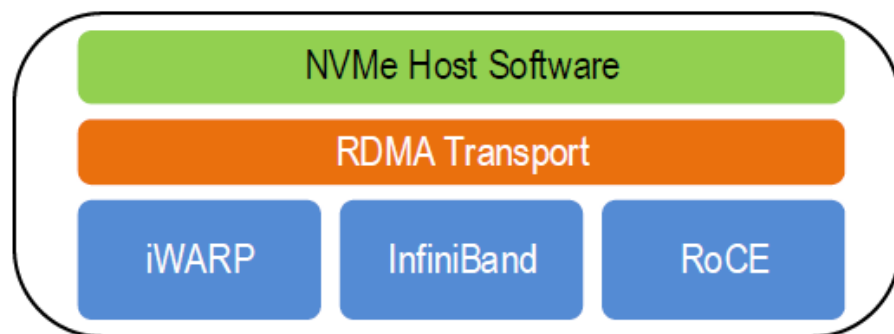
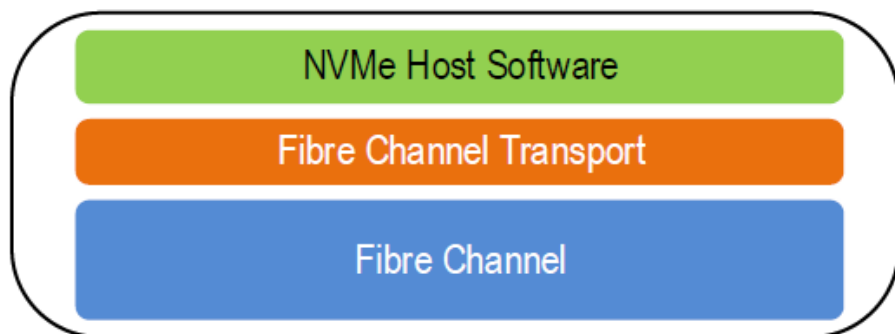
NVMe支持存储网络—NVMe over Fabrics



- **NVM Express over Fabrics, Revision 1.0, June 5, 2016**
- **不依赖于底层物理网络协议与连接**

NVMe接口/协议

NVMe支持存储网络—NVMe over Fabrics



Fibre Channel Transport Protocol Layers **RDMA Transport Protocol Layers**

The End