**【HUST2025】信息存储理论与技术**

##### 内容通览

**名词解释**

AFAs、AHCI、CXL、CoW、DAS、DFTL、disk、FC、FCIP、FDP、IB、IFCP、iSCSI、iSNS、HPDA、MultiStream、NAS、NVM、NVMe、OBS、OCSSD、RPO、RTO、SAM、SAN、SAS、SATA、SBP、SCSI、SDS、SMR、SNIA、SSD、SVC、tape、WAF、ZNSSSD、

**计算题**

**PPT 01 存储概述**

1. 算力网络三大要素
2. 信息存储技术的本质
3. 什么物理现象具有时间稳态
4. 满足哪三个条件才可以作为存储介质
5. 目前所用的主要存储器
6. 当代信息存储技术的特征
7. 网络存储系统的两种资源
8. 存储器类型（存储方式、功能、物理类型、物理特性）
9. 半导体主存储器（分类、可写一次读多次的存储器、闪存单元结构）
10. 磁存储器（种类、时间计算、硬盘、Flash/SSD、冷存储系统）
11. 光存储器
12. 存储器的分层结构/设计的三个问题/存储级别
13. 网络存储

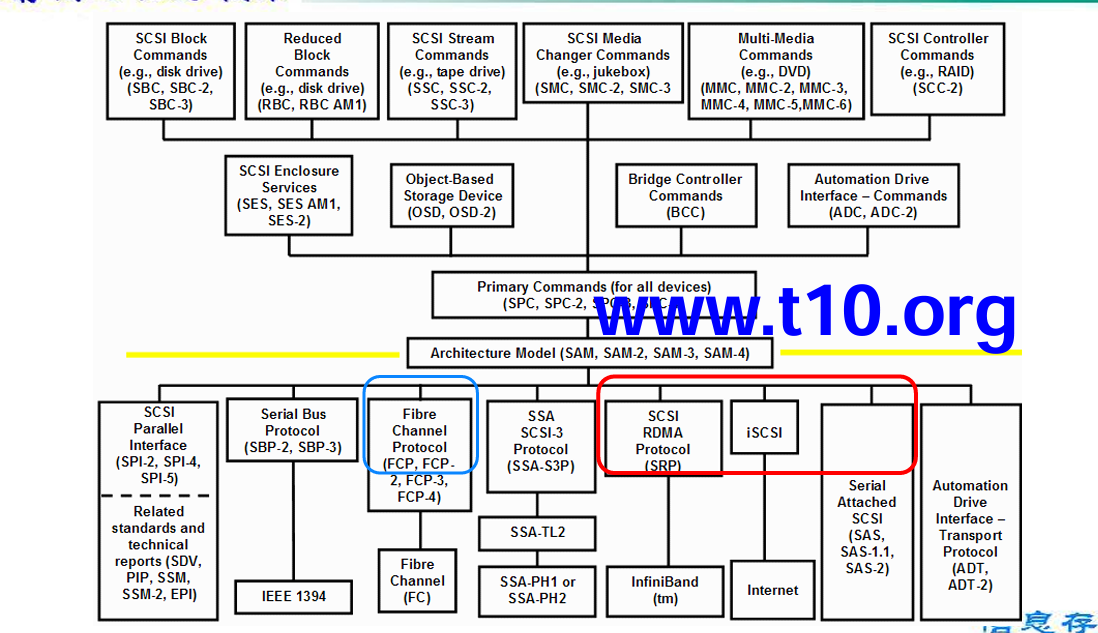
**PPT 02 RAID技术原理**

1. Why RAID

PPT 03 固态盘技术  
PPT 04 RAID遇上SSD  
PPT 05存储接口技术  
PPT 06 对象存储

PPT 07 存储管理与属于保护

##### 一、名词解释

****

1. AFAs（All Flash Arrays）全闪存阵列

全部使用SSD作为存储介质的企业级存储系统，相比传统磁盘阵列提供更高的性能、更低的延迟和更小的物理空间需求。

1. AHCI（Advanced Host Controller Interface）高级主机控制器接口

是一种由Intel开发的标准，用于规范操作系统与SATA设备之间的通信，简化了存储子系统的驱动程序编写，并提高了兼容性和性能。

1. CXL（Compute Express Link）计算快速链接

一种高速互联协议，用于连接 CPU 与设备（如内存、加速器、存储），支持缓存一致性，旨在提升数据中心性能和资源池化能力。

1. CoW（Copy-on-Write）写时复制

一种存储技术，在修改数据时不会直接覆盖原数据，而是先创建一个副本进行修改。这种方法广泛应用于快照和版本控制系统中，以减少对源数据的影响并提高安全性。

1. DAS（Direct-Attached Storage）直连存储

存储设备直接连接到单台主机（如通过 SATA、SAS 或 USB），不通过网络共享，常见于本地硬盘或外接存储。

1. DFTL (Dynamic Flash Translation Layer) 动态闪存转换层

是一种优化的闪存管理算法，旨在提高闪存存储设备的性能和寿命。它动态调整闪存块的映射关系来优化读写操作。

1. Disk磁盘

泛指用于数据存储的旋转磁介质设备（如 HDD），也常作为存储设备的通用术语，包括 HDD 和 SSD。

1. FC（Fibre Channel）光纤通道

一种高速网络技术，专为存储区域网络（SAN）设计，提供低延迟、高带宽的块级存储访问，常用于企业级存储。

1. FDP（Flexible Data Placement）灵活数据放置

NVMe SSD 中的一项特性，允许主机更精细地控制数据在 NAND 闪存上的物理布局，以提升写入效率和寿命，常用于 ZNS SSD。

1. FTL (Flash Translation Layer) 闪存转换层模块

使闪存看起来像磁盘驱动器的软件层

1. IB（InfiniBand）无限带宽

一种高性能、低延迟的网络互连技术，常用于高性能计算（HPC）和 NVMe over Fabrics（NVMe-oF）等场景。

1. IFCP（Internet Fibre Channel Protocol）互联网光纤通道协议

一种将FCP（Fibre Channel Protocol）封装于TCP/IP之上的协议，使得FC SAN可以通过互联网或任何IP网络连接。

1. iSCSI（Internet Small Computer System Interface）互联网小型计算机系统接口

一种基于 TCP/IP 网络传输 SCSI 命令的协议，用于在以太网上实现块级存储访问，是 IP-SAN 的核心技术。

1. iSNS（Internet Storage Name Service）互联网存储名称服务

提供了一种机制，用于发现、注册和查询iSCSI节点和服务的位置，支持大规模分布式存储网络的自动化配置和管理。

1. HPDA（High Performance Data Analytics）高性能数据分析

利用高性能计算资源处理大量数据集的一种方法，强调实时或接近实时的数据分析能力。

1. MultiStream多流写入

SSD 的一项技术，允许主机将不同类型或生命周期的数据分配到不同的“流”（Stream），以优化垃圾回收和写入放大，提升性能与耐久性。

1. NAS（Network-Attached Storage）网络附加存储

通过网络（通常为以太网）提供文件级存储服务的设备或系统，使用 NFS、SMB/CIFS 等协议，适合文件共享。

1. NVM（Non-Volatile Memory）非易失性内存

断电后仍能保留数据的存储介质，如 NAND 闪存、3D XPoint 等，是 SSD 和新型存储器的基础。

1. NVMe（Non-Volatile Memory Express）非易失性内存主机控制器接口规范

一种为 PCIe SSD 设计的高性能存储协议，大幅降低延迟、提升并行度，支持 NVMe over Fabrics（NVMe-oF）实现网络化访问。

1. OBS（Object-Based Storage / Object Storage Service）对象存储

以“对象”为单位管理数据的存储架构，每个对象包含数据、元数据和全局唯一ID，通过 RESTful API 访问，适用于海量非结构化数据（如云存储）。

1. OCSSD（Open-Channel SSD）开放通道固态硬盘

一种特殊的SSD类型，允许操作系统或中间件直接控制底层闪存管理功能，如垃圾回收和磨损均衡，以实现更高效的数据管理和更好的性能。

1. RPO（Recovery Point Objective）恢复点目标

在灾难发生后，企业能够容忍的最大数据丢失量的时间点指标。即数据恢复到最近备份时间点的目标。

1. RTO（Recovery Time Objective）恢复时间目标

灾难发生后，业务必须恢复运行前可接受的最大中断时间长度。

1. SAN（Storage Area Network）存储区域网络

专用高速网络，用于连接服务器与块级存储设备，通常基于 FC 或 iSCSI，提供类似本地磁盘的访问体验。

1. SAM SCSI Architecture Model（SCSI 架构模型）

由 ANSI 和 INCITS 定义的 SCSI 标准体系结构文档（如 SAM-5），用于规范 SCSI 命令集、设备模型和通信协议。这是 SNIA 和存储标准中常见的术语。

1. SAS（Serial Attached SCSI）串行连接 SCSI

一种基于 SCSI 命令集的高速串行存储接口，兼容 SATA，常用于企业级硬盘和服务器存储。【详细见PPT2-8】

1. SATA（Serial ATA）串行高级技术附件

一种主流的硬盘接口标准，用于连接 HDD 和 SATA SSD，成本低但性能低于 SAS 和 NVMe。

1. SBP Serial Bus Protocol（串行总线协议）

特指 SBP-2（Serial Bus Protocol 2），是 IEEE 1394（FireWire/火线）接口上用于传输 SCSI 命令的协议，允许通过 FireWire 连接 SCSI 设备（如外置硬盘）。

1. SCSI（Small Computer System Interface）小型计算机系统接口

一种用于计算机与存储设备通信的命令集和接口标准，是 SAS、iSCSI、FC 等技术的基础。【详细见PPT2-7】

1. SDS（Software-Defined Storage）软件定义存储

通过软件抽象、池化和管理底层物理存储资源，与硬件解耦，存储功能（如快照、复制、去重），可运行在通用 x86 服务器上，无需专用存储阵列，由软件实现，灵活性高、成本低、易于扩展和自动化。

1. SMR（Shingled Magnetic Recording）叠瓦式磁记录

是一种增加硬盘面密度的技术，通过部分重叠轨道来提高存储容量，但可能会影响写入速度和耐久性。

1. SNIA（Storage Networking Industry Association）存储网络行业协会

是一个致力于推动存储和信息管理行业发展的全球性组织，制定标准、教育计划和技术路线图。

1. SSD（Solid State Drive）固态硬盘

使用 NAND 闪存作为存储介质的存储设备，无机械部件，具有高速度、低功耗、抗震等优点。

1. SVC（Storage Virtualization Controller）存储虚拟化控制器

是一种硬件或软件解决方案，用于抽象化物理存储资源，使其看起来像单一逻辑存储池，便于管理和分配

1. Tape磁带

一种顺序访问的磁性存储介质，主要用于数据备份、归档和冷存储，具有高容量、低成本、长寿命的特点。

1. WAF（Write Amplification Factor）写放大系数

描述了实际写入闪存的数据量与主机要求写入的数据量的比例，是衡量SSD性能的一个重要指标。较高的写放大系数意味着更多的写操作和更快的磨损。

1. ZNSSD（Zoned Namespace SSD）分区命名空间固态硬盘

一种新型 NVMe SSD，将存储空间划分为多个“区域”（Zone），主机需按顺序写入，以减少写放大并提升效率，适用于日志、数据库等场景。

##### 计算题

1. **RAID相关计算题**

由8块1TB磁盘组成的RAID5阵列，采用左不对称的配置方式，每个磁盘

的块大小为32KB，磁盘编号为0–7。

1. 该系统的实际可见容量？

2. 若用户对该阵列发出写请求，起始扇区为640（十进制，每个扇区大小为

512B），数据长度为64KB，那么为了完成该写请求最少需要多少次磁盘I/O？

3. 若磁盘3出现故障不能读写，系统进入恢复阶段，则此时要读出题目2中

的数据比正常状态下要多执行多少次磁盘I/O？  
  
1. 可用容量 = (N - 1) × 单盘容量 = 7 × 1 TB = 7 TB

2. 步骤1：将用户请求转换为字节偏移和覆盖的块范围

起始扇区 = 640，每扇区 = 512 B，起始字节偏移 = 640 × 512 = 327,680 字节

请求长度 = 64 KB = 65,536 字节，结束字节 = 327,680 + 65,536 = 393,216 字节

步骤2：确定涉及哪些“数据块”（以32KB为单位）

块大小 = 32 KB = 32 × 1024 = 32,768 字节

起始块号（全局逻辑块号）= floor(327,680 / 32,768) = 10

结束块号（含）= floor((393,216 - 1) / 32,768) = floor(393,215 / 32,768) = 11

因为：块10：字节 [327,680, 360,447]；块11：字节 [360,448, 393,215]

✅ 所以该写请求覆盖 2个连续的数据块：块10 和 块11

步骤3：确定这两个块在RAID 5阵列中的物理分布（磁盘编号）

RAID 5（8盘，左不对称）的条带结构：

每个条带组（stripe set）包含 8 个块：7个数据块 + 1个校验块

条带组编号 s = floor(全局块号 / 7) （因为每组7个数据块）

在组内数据块位置 d = 全局块号 mod 7

对于左不对称（Left Asymmetric）布局（见原文图2-3(b)）：

校验块位于 磁盘号 = (7 - s) mod 8

数据块按顺序填充除校验盘外的其他磁盘，从校验盘的下一个磁盘开始（循环）

但更简单的方法是参考原文对“左不对称”的定义（图2-3(b)）：

因此，条带 s 的校验盘号 = (7 - s) mod 8

计算块10 和 块11 所属条带：

每条带含 7 个数据块 →

块10：s = floor(10 / 7) = 1，组内序号 = 10 % 7 = 3

块11：s = floor(11 / 7) = 1，组内序号 = 11 % 7 = 4

→ 两块同属 条带 s = 1

条带1的校验盘号 = (7 - 1) mod 8 = 6

即：Disk6 是校验盘，其余 Disk0–5、7 存放数据块

数据块在条带中按顺序分配给非校验盘，从校验盘的下一块开始（左不对称特性：数据从 校验盘+1开始顺时针放置）

根据原文图2-3(b)（左不对称）示例：

条带0：P在Disk4，数据D0→Disk0, D1→Disk1, D2→Disk2, D3→Disk3

实际上，在左不对称中，数据块按自然顺序分配到非校验盘，校验盘位置循环右移

更可靠的方式（标准定义）：

对于左不对称（Left Asymmetric）：

数据块依次放在 除校验盘外的磁盘上，从磁盘0开始顺序跳过校验盘

即：构建一个磁盘列表 [0,1,2,3,4,5,6,7]，去掉校验盘6 → [0,1,2,3,4,5,7]

组内数据块0→Disk0, 1→Disk1, 2→Disk2, 3→Disk3, 4→Disk4, 5→Disk5, 6→Disk7

所以：

块10（组内序号3）→ Disk3

块11（组内序号4）→ Disk4

✅ 两个数据块分别位于 Disk3 和 Disk4

步骤4：判断写操作类型

写入长度 = 64 KB = 2 × 32 KB → 正好覆盖2个完整数据块

但这2个块不在同一个条带组的所有数据块中（条带有7个数据块），因此不是“满条带写”

所以属于 “小写”（Partial Stripe Write）

根据原文：

RAID 5 小写采用 RMW（Read-Modify-Write） 或 RCW（Reconstruct-Write）

RMW 更优：需 读旧数据 + 读旧校验 → 计算新校验 → 写新数据 + 写新校验，共 4次 I/O

但注意：这里有2个数据块，且它们属于同一个条带（s=1），共享同一个校验块（在Disk6）

因此，不能分别处理，而应合并处理整个条带更新。

最优策略：一次读取旧校验 + 两个旧数据块 → 计算新校验 → 写两个新数据块 + 新校验

I/O 操作如下：

读 Disk3 上的块10（旧数据）；读 Disk4 上的块11（旧数据）

读 Disk6 上的校验块（旧P）；写 Disk3 块10（新数据）；写 Disk4 块11（新数据）

写 Disk6 校验块（新P）

✅ 共 6 次磁盘 I/O

是否可更少？

若使用 RCW（读取同条带所有其他5个数据块来重算校验），则需读5块+写3块=8次，更差。

所以 RMW 是最少的。

但注意：如果这两个块恰好覆盖了整个条带的数据（7块），则是满条带写，只需写7数据+1校验=8次，无需读。但此处只写2块，故必须 RMW。

然而，是否可以只更新校验一次？ 是的，因为两块同属一条带，只需一次校验更新。

所以最小 I/O 次数 = 2（读旧数据） + 1（读旧校验） + 2（写新数据） + 1（写新校验） = 6 次

✅ 答：最少需要 6 次磁盘 I/O

1. 正常状态下读取：

块10 在 Disk3

块11 在 Disk4

直接读 Disk3 + Disk4 → 2 次 I/O

磁盘3故障时：

块10 无法直接读取（所在盘故障）

需通过重构（Reconstruct）：读取同一条带（s=1）中其余6个可用块（6个数据块中的5个好盘 + 1个校验盘），通过异或恢复块10

条带 s=1 包含：

数据块：块7~13（共7块），其中块10在Disk3（故障）

校验块在 Disk6（正常）

要恢复块10，需读取：

同条带其他 6 个块：即除块10外的 6 个块（5个数据块 + 1个校验块）

具体包括：

块7,8,9,11,12,13（数据块，分布在除Disk3、Disk6外的盘）

校验块 P（在Disk6）

但注意：我们只需要块10和块11

块11 仍在 Disk4，可直接读 → 1 次 I/O

块10 需重建 → 需读取同条带其余 6 个块（6次 I/O），然后在内存中异或计算出块10

原文：“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”

因此，故障状态下读取这两块数据需：

读块11：1 次（Disk4）

读同条带其他6块（用于恢复块10）：6 次

总计：7 次 I/O

正常状态：2 次 I/O

故障状态：7 次 I/O

✅ 多执行：7 - 2 = 5 次

但注意：是否必须读全部6块？

是的，因为 RAID 5 校验为：

P = D0 ⊕ D1 ⊕ … ⊕ D6

⇒ D\_fault = P ⊕ (所有其他 D\_i)

要恢复一个数据块，必须读取其余6个块（6个 XOR 项）

因此，恢复1个故障块需6次读 I/O

所以：

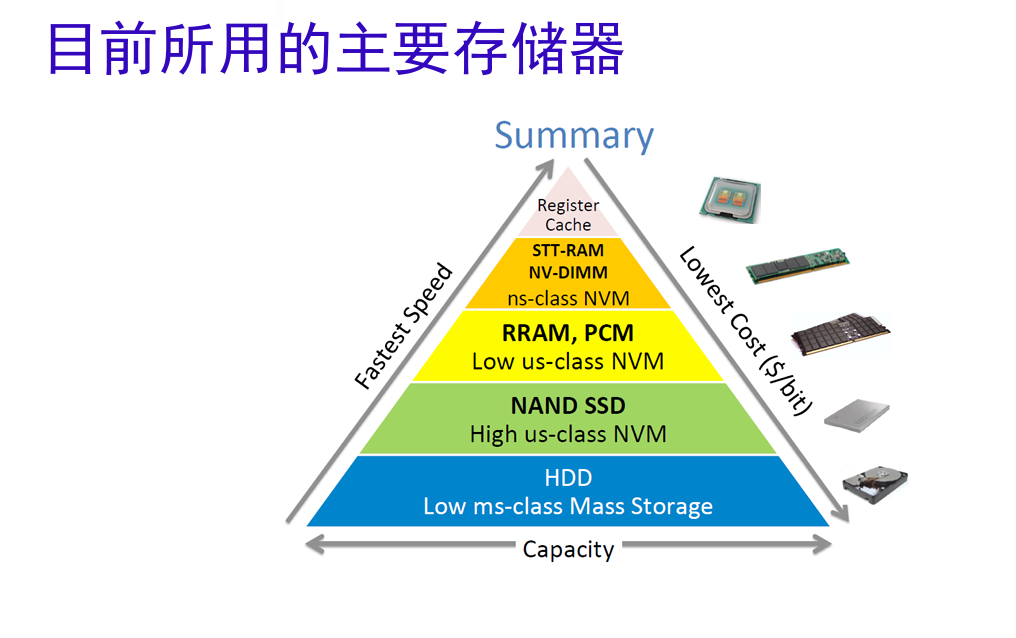
正常读块10+11：2次

故障时：读6块（恢复块10） + 读块11 = 7次

多出：5次

✅ 答：多执行 5 次磁盘 I/O

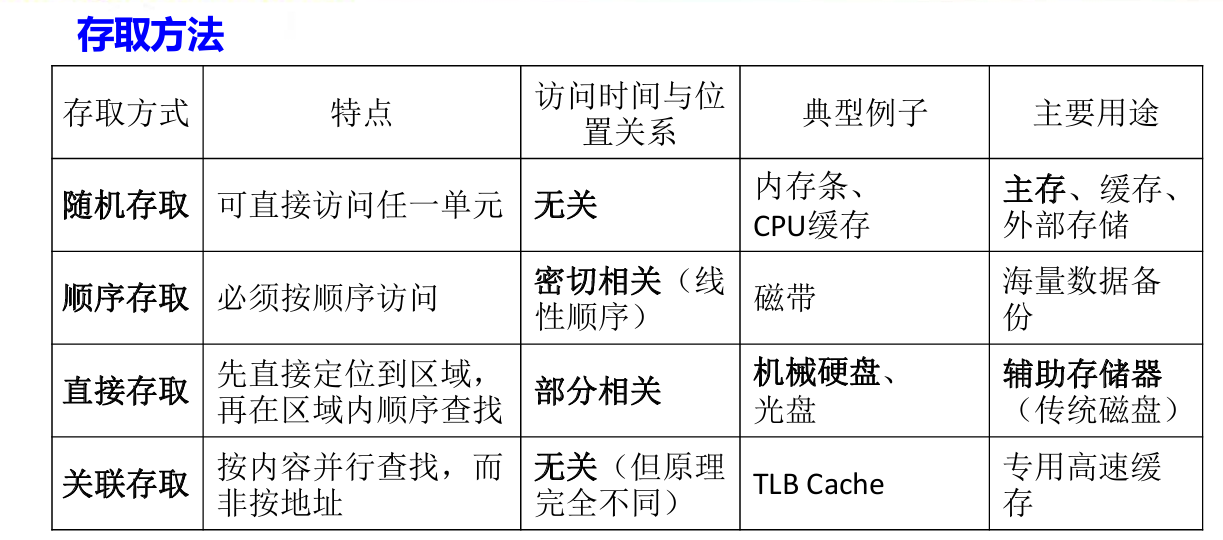
##### PPT 01 存储概述

1. 算力、存力、运输力
2. 信息存储技术的本质是利用具有时间稳态的物理原理和现象， 实现跨越时间的信息传递
3. 什么物理现象具有时间稳态： 形变/色变对光的反射和透射，磁稳态，半导体稳态，量子稳态，生物稳态
4. 满足三个条件： 具有2个或以上状态，状态可识别，状态可改变（磁 光 电 量子 DNA）
5. 目前所用的主要存储器 ****
6. 当代信息存储技术的特征：  
   数字化： 一切形式的信息统一转换为0，1的集合而存储   
   网络化： 所有的存储器都可通过网络（广义的网络：有线、无 线、互联总线）互连起来， 首次实现了时间和空间的二维任意传送
7. 网络存储系统的两种资源   
   存储资源（容器）： 半导体存储（RAM、NVRAM、Flash)、磁存储（硬盘、磁带）、 光存 储（CD-ROM、DVD、BD、MO)   
   传输资源（管道）： 总线、网络和交换、路由设备 SCSI, FC, IP, iSCSI, IB, SATA, SAS, NVMe，CXL…
8. 存储器类型
9. 存取方法：

顺序存取：访问时间与存储单元的物理位置密切相关，磁带

随机存取：每一位置有唯一的寻址机制直接达到，RAM

直接存取：块间直接到达、块内顺序存取，磁盘

关联存取：一个字通过其部分内容而不是地址进行访问， Cache、TLB   


【固态硬盘SSD属于随机存取】

1. 功能：   
   只读存储器：ROM,CD-ROM   
   可重写的存储器：Disk   
   可擦除的存储器：EPROM，Flash
2. 物理类型： 半导体、磁表面、光
3. 物理特性： 易丢失/不易丢失、可擦除/不可擦除
4. 半导体主存储器
5. 分类   
   随机存储器RAM：数据易失

静态RAM：触发器中逻辑门，快；

动态RAM：电容充电存储数据，需刷新

只读存储器ROM：数据永久保存

微程序设计、常用函数库、系统程序、功能表 （批量生产）

可编程ROM（PROM）： 写一次，数据不丢失 （少量生产）

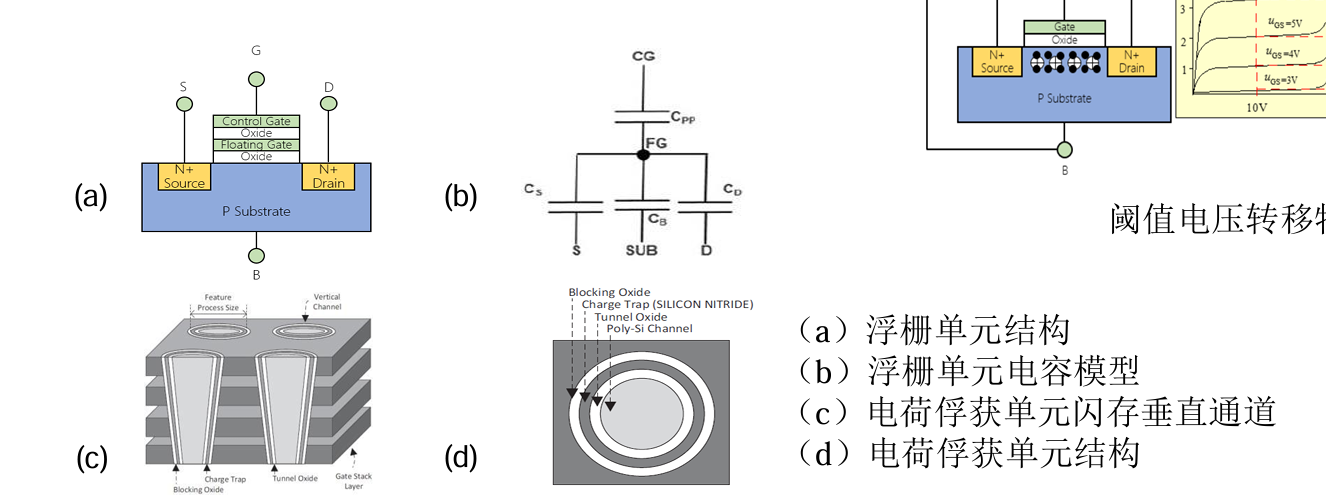
1. 可写一次读多次：

PROM（EPROM）： 紫外线擦除、电写、数据不易失

闪存（FLASHMemory）： 一至几秒内被擦除，数据不易失

电可擦PROM（EEPROM） 存放配置信息

1. FLASH MEMORY闪存单元结构  
   目前主流闪存单元分为浮栅单元和电荷俘获单元；

  
 • 通过一定外加电场作用，使闪存单元俘获/排出电荷；改变存储单元阈值电压的高低， 表示逻辑0、1；

– 写：字线与位线间施加较大电压(20V)

– 读：字线与位线间施加判断电压(≈2V)

– 擦除：字线与位线间施加反向电压(20V)

• 闪存阵列结构 【容量更大】

从2D阵列转向3D堆叠阵列

–工艺尺寸缩小到达物理极限

–垂直堆叠进一步增加存储密度，堆叠层数逐年增加

单元存储更多逻辑比特

–存储密度进一步提高（MLC->TLC->QLC）

–阈值电压区间被压缩，可靠性和编程速度需要进一步研究

1. 磁存储器

借助磁性材料的两种剩余磁化状态，或磁化与非磁化的两种材料状态，或有磁化翻转和无磁化翻转的两种状态记录二进制数据信息。

1. 种类：   
   磁芯存储器：存储单元是铁氧体圆环

磁泡存储器存储单元是圆柱形的磁畴，称为磁泡

磁面存储器磁层上记录的是正、负磁化状态或磁化状态的变化，即磁化翻转。按媒体基 底材料的不同，分为两类。使用柔性基底材料的设备有磁带机，软磁盘机。使用刚性 基底材料的是硬磁盘驱动器。

1. 时间计算

Disk access time = seek time (disk arm) ~ 8ms （寻道时间）  
 + rotational delay (disk platter) ~ 3 ms @ 10000 rpm （旋转延迟）

+ transfer time (media transfer rate) ~ 60 MB/s （传送时间）

• Today: Processor performance increases roughly 60% per year.

• Today: Disk capacity increases 100% per year.

• Today: Disk performance increases only 20% per year

• 旋转延迟补充：最坏情况 a complete rotation 7,500 rpm = 8 ms 15,000 rpm = 4 ms

1. 硬盘：在大规模存储系统中仍占重要地位

➢ 耗能在不同状态下有很大差别：   
 休眠（0.8W)、空闲(5-11w)、读写(10-21w)   
 随机读写，顺序读写，内圈读写，外圈读写：转速不同

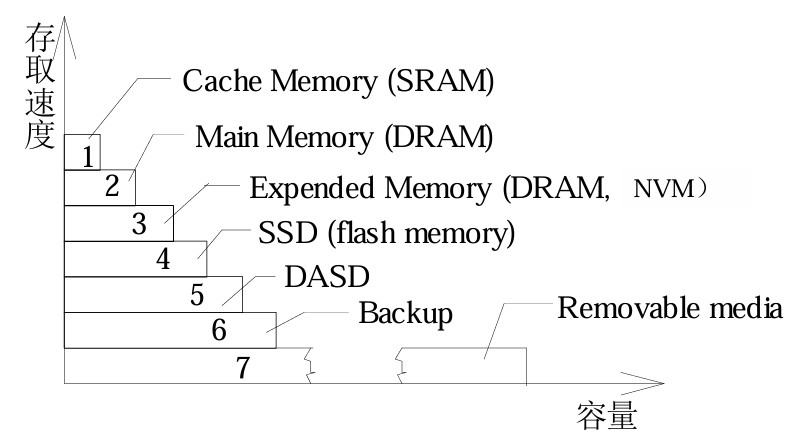
1. 闪存和固态盘（Flash/SSD）

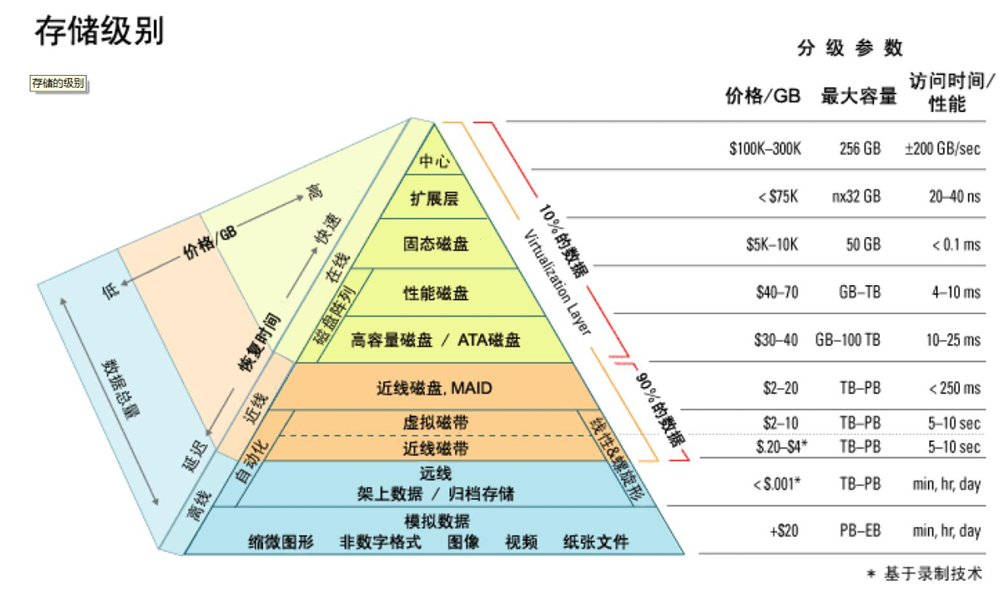
用于移动和高端存储，能耗为硬盘的十分之一 （ZNSSSD、FDP、MultiStream…）

1. 冷存储系统 ➢ 硬盘/固态盘/光盘？
2. 光存储器

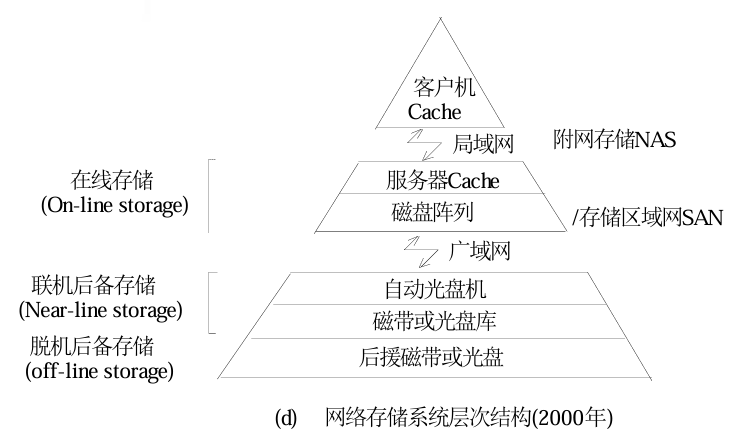
利用微小的激光束照射光记录媒体上，使被照射部位发生 热效应或光效应，从而改变媒体的光学（或光磁）性质以记录 信息的一类存储设备。读出时，媒体表面的状态转变为反射光 强或偏振光的偏转角旋转，还原出记录的信息。种类： 只读光盘存储器 、只写一次读多次光盘存储器 、可擦光盘存储器

1. 存储器分层结构

存储器设计的三个问题：容量、速度、价格 ****

****

1. 网络存储

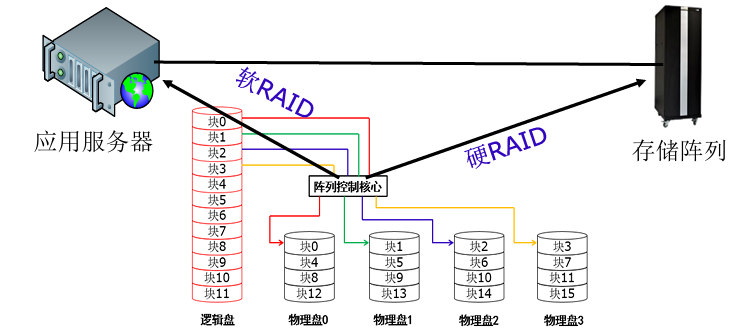
****

##### PPT 02 RAID技术原理

1. Why RAID存储容量需求不断增加、存储数据可靠性要求越来越高、存取速度要求越来越快RAID设计借用大型机中Striping 和Interleaving的概念
2. 盘阵列  
   把多块独立的存储盘按某种方式组织起来形成一个或多个逻辑盘，从而提供比单个存储盘更高性能和更高可靠性的存储技术。

◆ 条带化（Striping） ◆ 镜像（Mirroring） ◆ 奇偶校验（Parity）

RAID：构成一个逻辑盘 【阵列控制核心 → 映射，也可称为虚拟化 实例：Linux DM:MD】

1. RAID分类软件式与硬件式(Software RAID & Hardware RAID)  
   区分的标准是看实现RAID是否占用主机的资源（CPU， Memory） ****
2. RAID基本原理与特征数据以条块化（striping）分布于多个磁盘

存储容量扩展，I/O性能提升

冗余机制获得较高的数据可用性

**可用性（Availability）**: 即使某些部件故障仍能够为用户提供服务

通过冗余信息实现数据恢复（Reconstructed & Rebuild)

不足之处： 容量损失：存储冗余信息

带宽损失：冗余信息的读写

计算资源损失：冗余信息的更新,恢复

1. RAID级别

RAID 0：数据分割，无容错能力

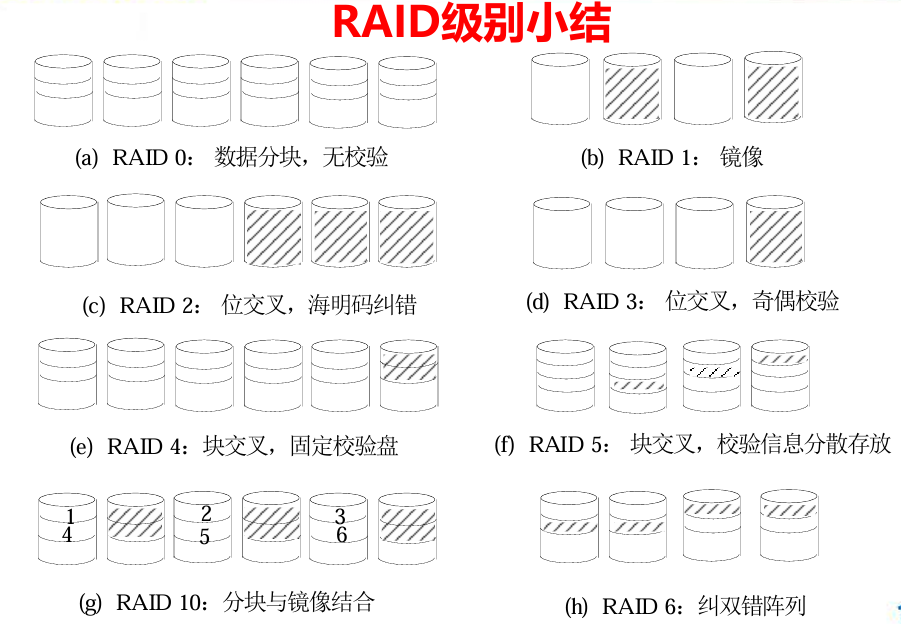
RAID 1：镜像(双拷贝)

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，块交叉，循环校验盘

RAID 6：容双盘错，块交叉，近年被广泛重视   
  
➢ RAID 0

多个盘构成阵列，数据分块轮流存储，提高容量。解决单盘无法并行工作的问题，多盘同时操作，提高性能。数据分块，无冗余校验。  
存在的问题：可靠性降低，无容错能力。由N个盘构成的盘阵的故障率为单盘的N倍

➢ RAID 1

每个盘上的数据均在镜像盘上有一个完整的复制副本；高可用性

写入带宽缩减：Logical write = two physical writes Reads may be optimized

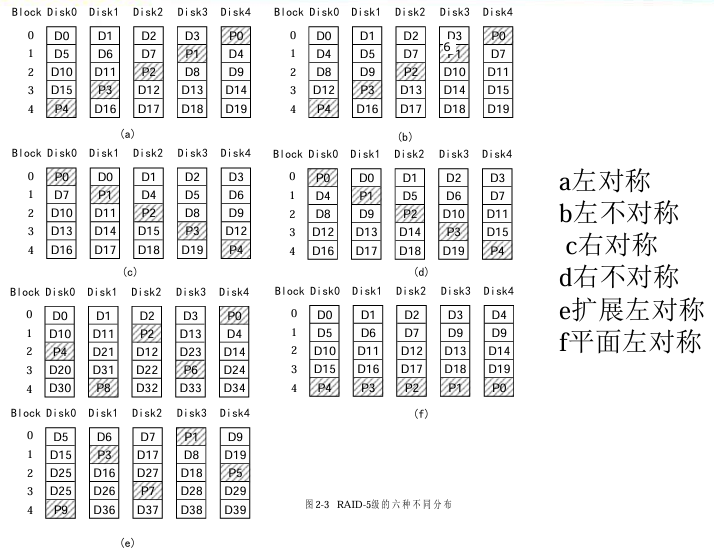
代价较高的解决方案：一半的容量损失

➢ RAID 3、4 奇偶校验

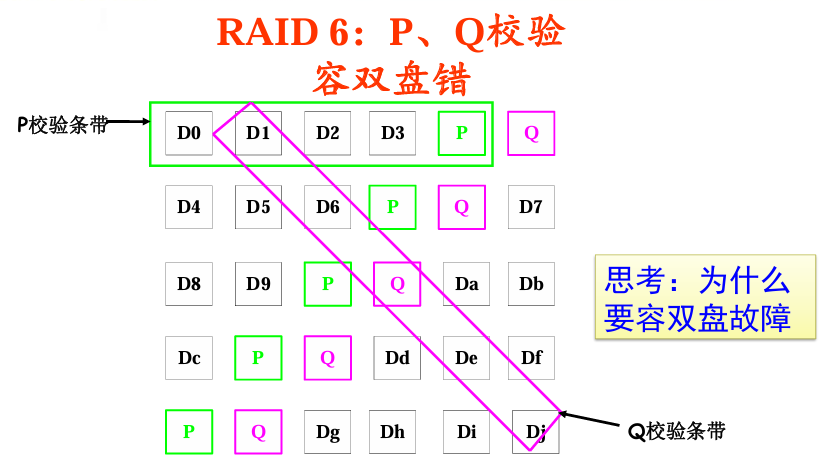
* 区别：RAID 3：位交叉，细粒度；RAID 4：块交叉，粗粒度
* RAID 4校验瓶颈问题：在同一时刻DISK P只能有一个I/O操作，因此作业B只能在作业A完成后才能开始
* RAID 4改进→RAID 5：RAID 4对读比较有效；并发写操作受限于校验盘：Write to D0, D5, both also write to P disk

➢ RAID 5 Rotate Parity

* 在同一时刻可以同时完成作业A及作业B
* 适用于小组，随机存取；写速度缓慢







➢ RAID 6  
【重视原因】存储容量需求不断增加；硬盘故障是常态；RAID重建时间随容量增长而显著延长；重建期间系统处于降级状态，风险极高。因此，随着单盘容量增大：单次重建所需读取的数据量剧增，导致重建时间大幅延长；在漫长的重建过程中，再次发生磁盘故障的概率显著提高；RAID 5仅能容忍单盘故障，在此场景下系统可靠性急剧下降；而RAID 6支持双盘容错，有效提升了在大容量磁盘环境下的系统可用性与数据安全性，故“近年被广泛重视”。

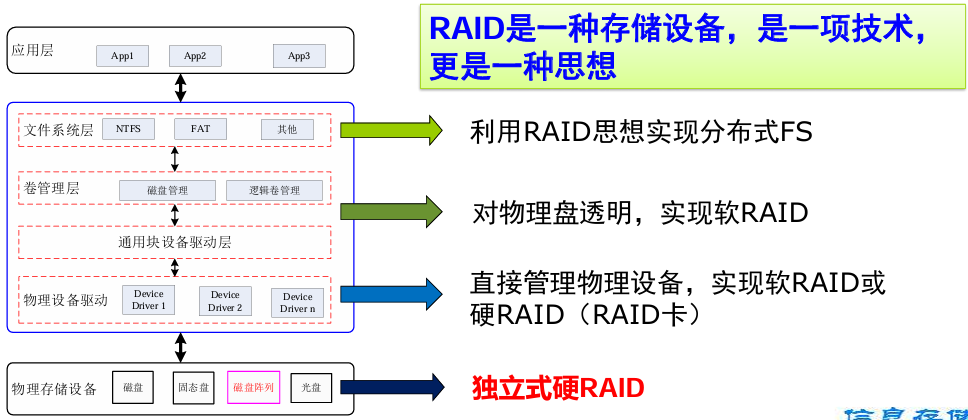
➢ 组合级别：RAID 10: Striping & Mirroring

RAID技术从0到5很好地解决了阻碍计算机发展的几个重要问题

1. 通过多个盘构成阵列，解决了容量问题
2. 通过冗余算法，解决了存储设备的可靠性问题
3. 通过多盘并行，解决IO性能问题

不足： 存取速度仍是处理器的瓶颈

1. STRIPING技术并不能有效地提高速度分块的大小与磁盘的机械特性有关文件的大小也不同
2. 并行I/O提高速度的能力有限多次读写后,数据散列在各个盘上, 磁盘的机械运动 成为阻碍速度提高的主要因素
3. 并非备份解决方案，无法抵御文件损坏、病毒感染或人为误删等逻辑错误
4. RAID实现

****

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？】

1. SCSI协议 支撑存储发展的脊梁计算机与外部设备（特别是存储设备）间系统级接口的标准；

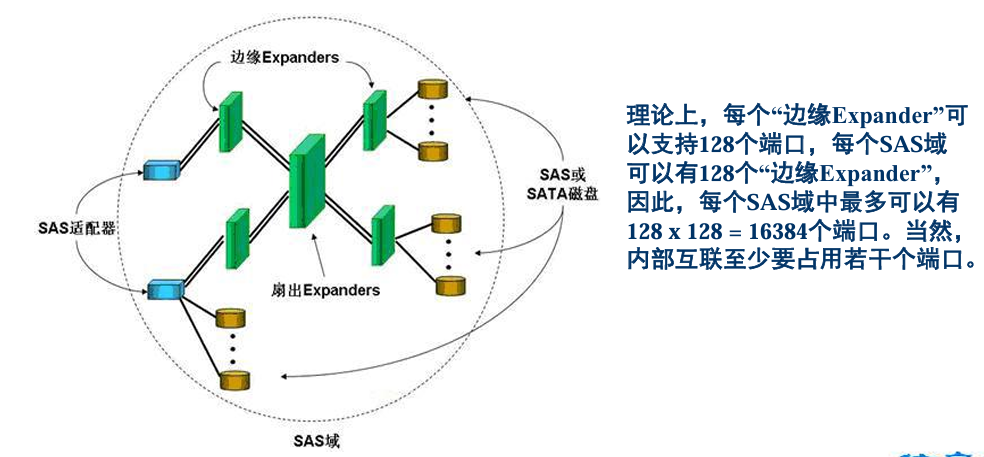
SCSI标准定义了命令、通信协议、实体电器特性 等（物理层，连接层，通信层，应用层）

1. SAS（串行的SCSI）

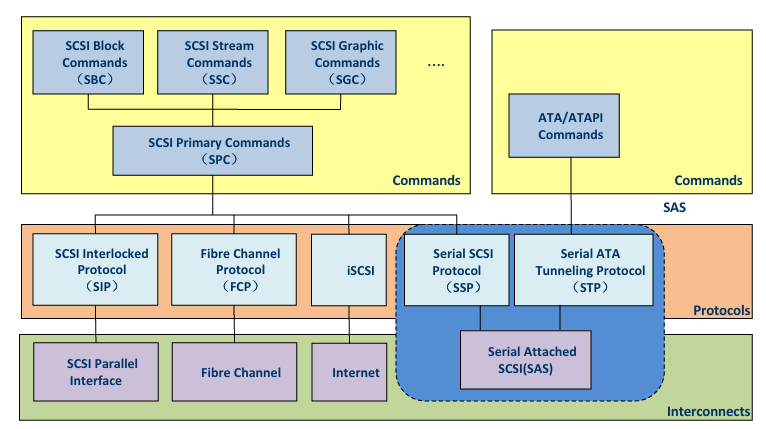
产生：并行SCSI发展到Ultra320，没有提升空间；低端的SATA性能、可靠性受限

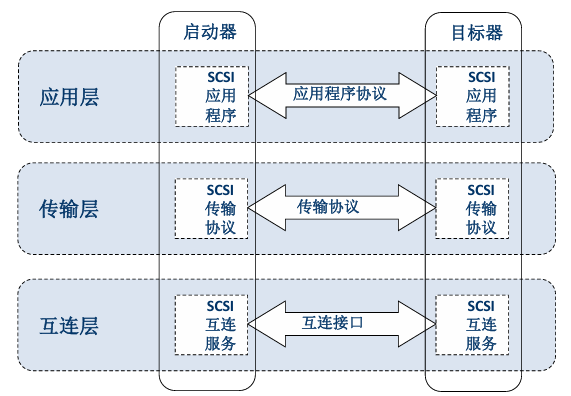
特点：SCSI向下兼容性、串行点对点互连、双端口、寻址性和向小型化的扩展能力于一 身；可提供大数量设备、高带宽、可扩展性支持

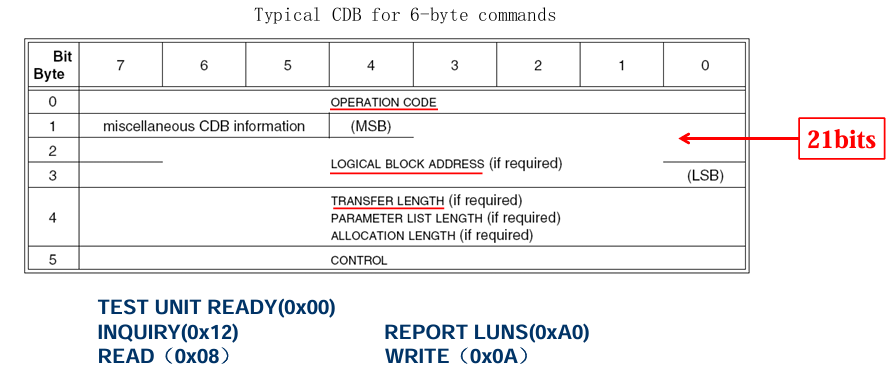
SAS域：一个SAS域主要由SAS初始设备(SAS Initiator Device)，扩展设备 (Expander Device)，以及SAS 目标设备(SAS Target Device)组成 SAS连接：

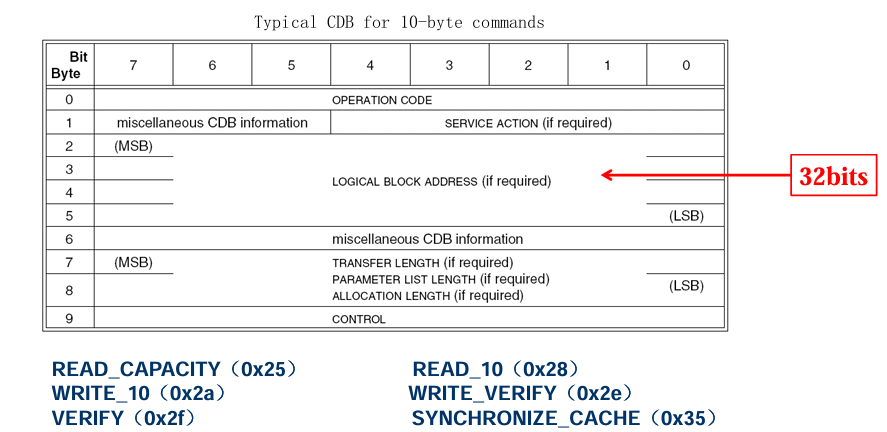


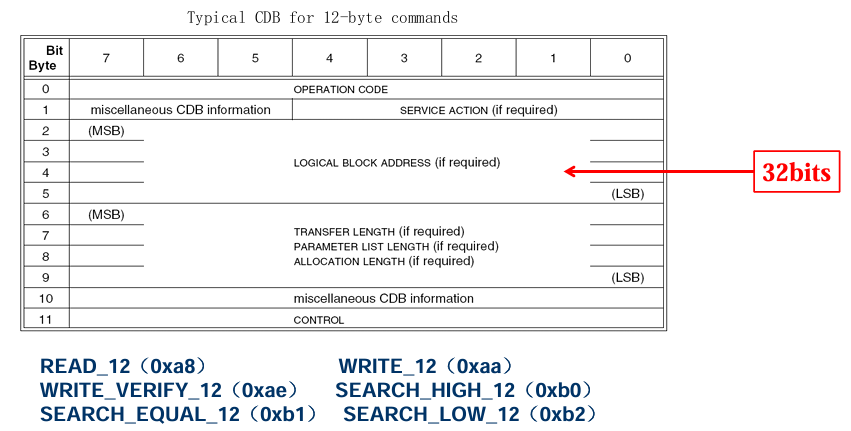
SAM-3结构、SAS在SAM中的位置

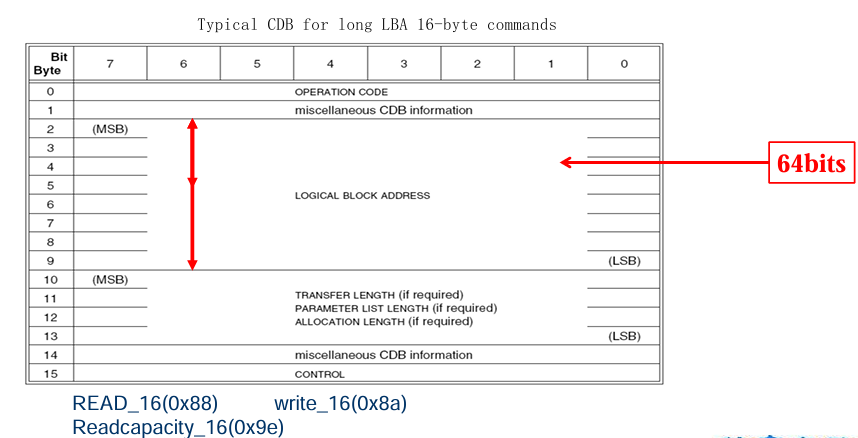
  


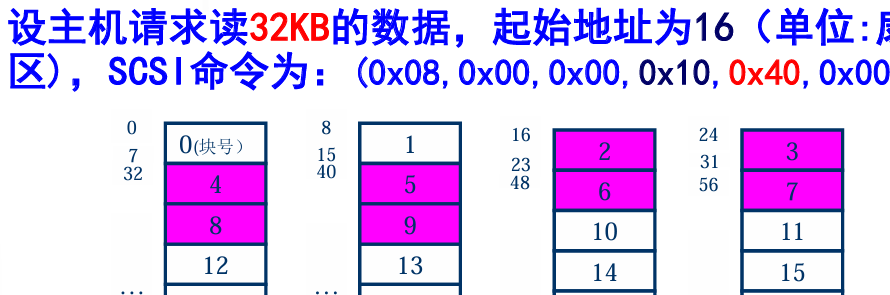
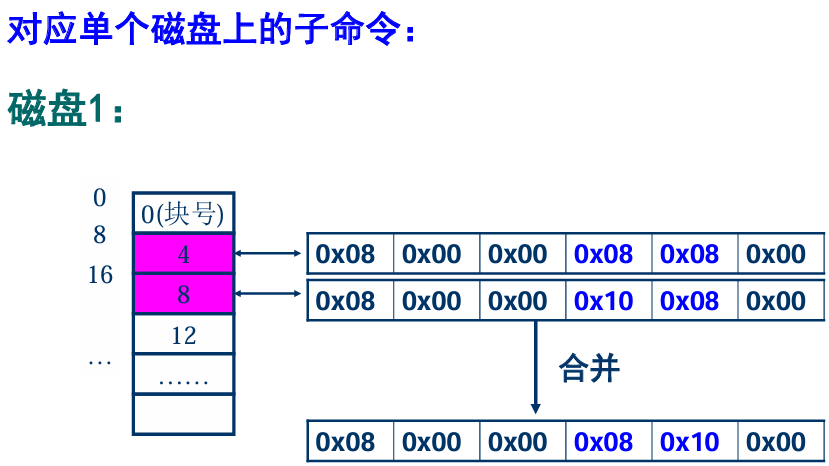
1. SCSI通信服务模型“客户-服务器”模型 ****对SCSI设备的访问通过三元组：总线、目标、LUNLinux系统SCSI中间层(SCSI Middle Level) 定义了“scsi device”的数据结构（SCSI CDB）

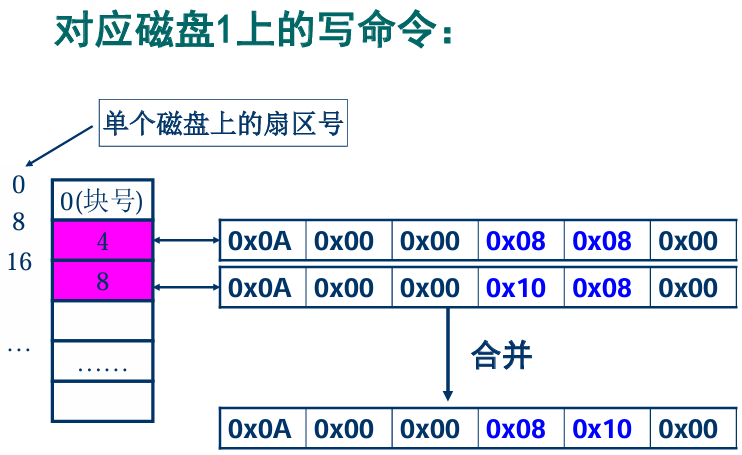
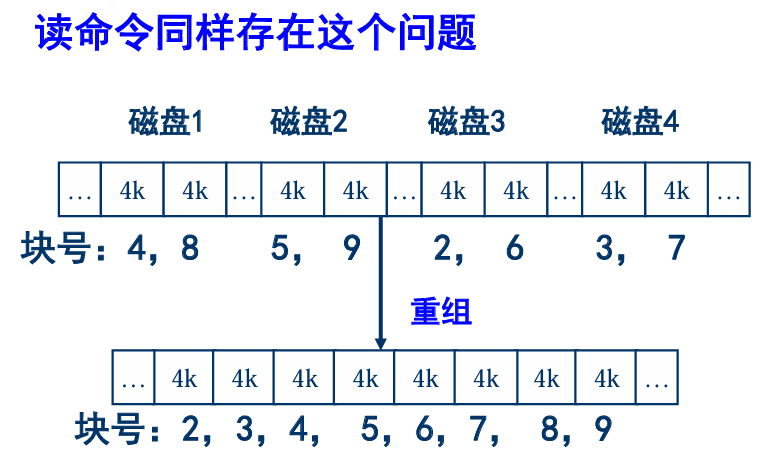
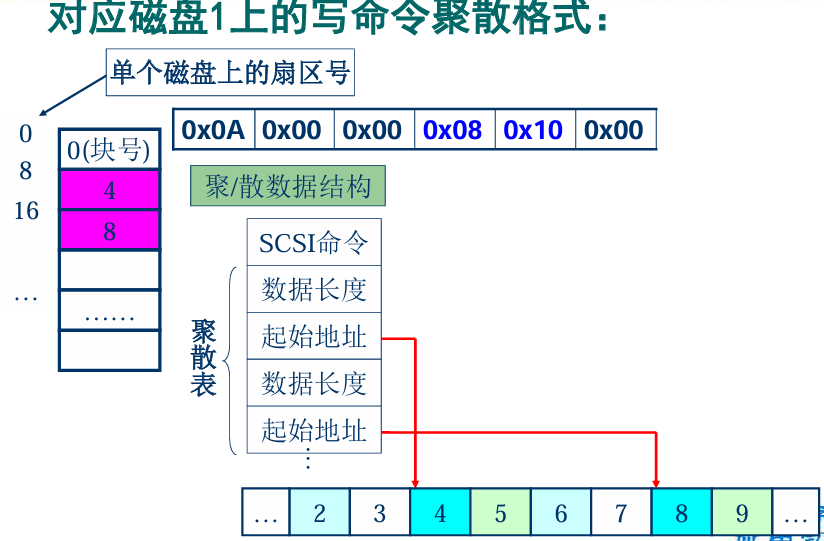
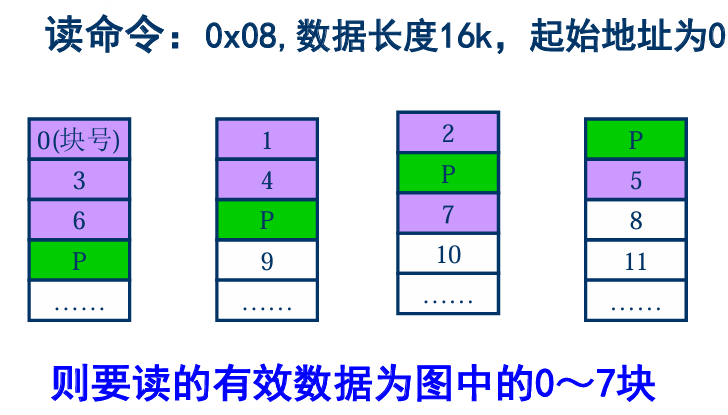
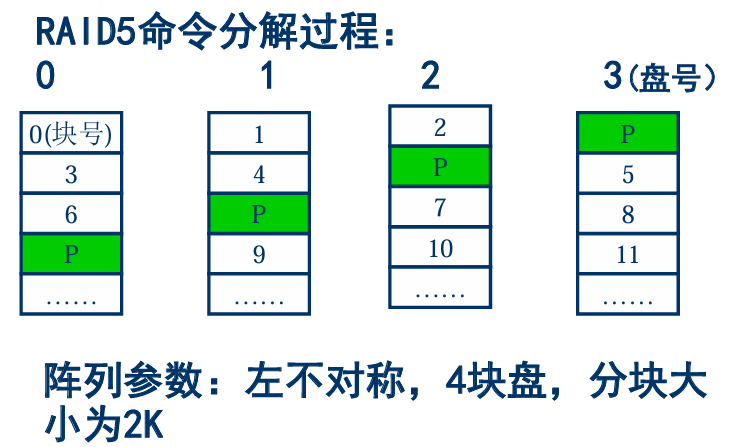
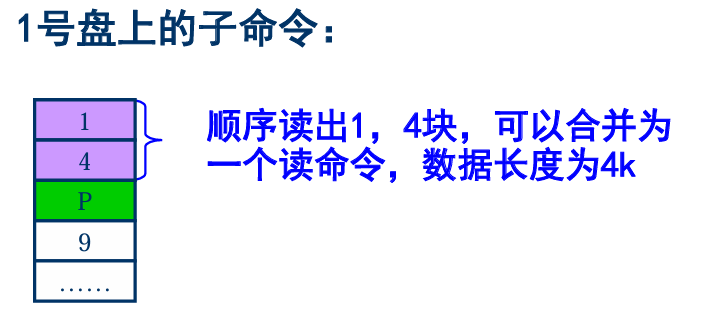
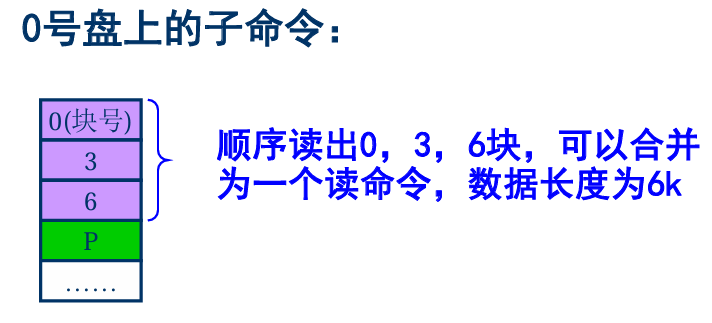
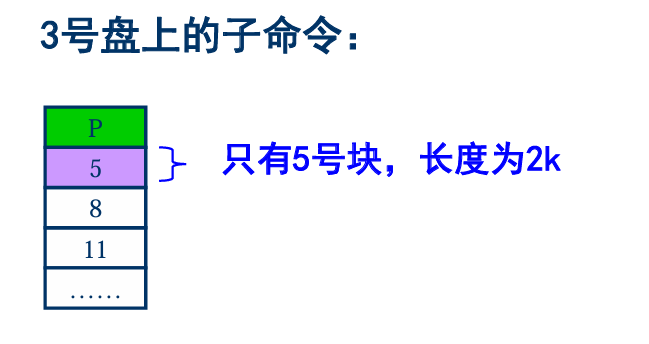
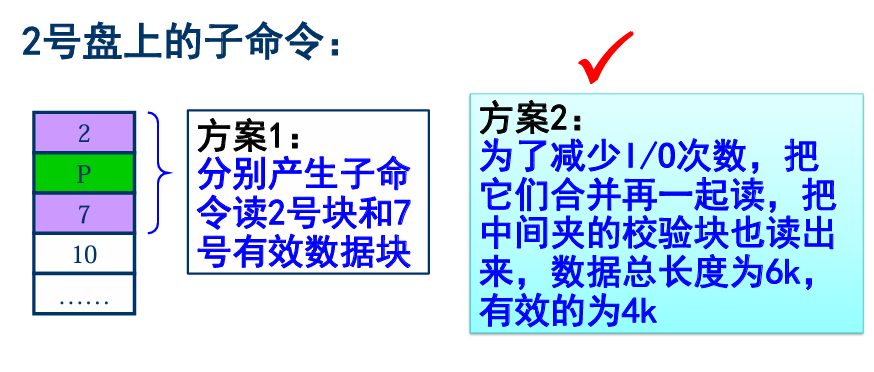
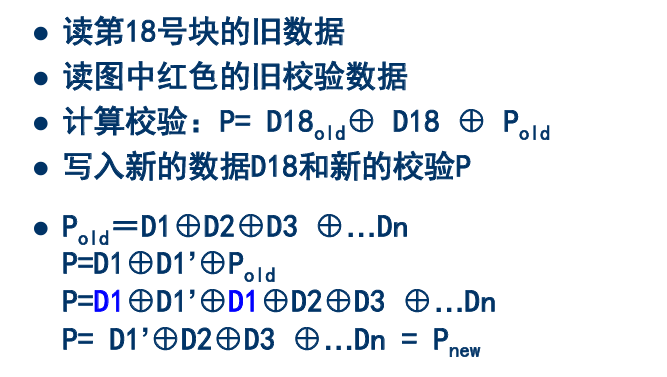
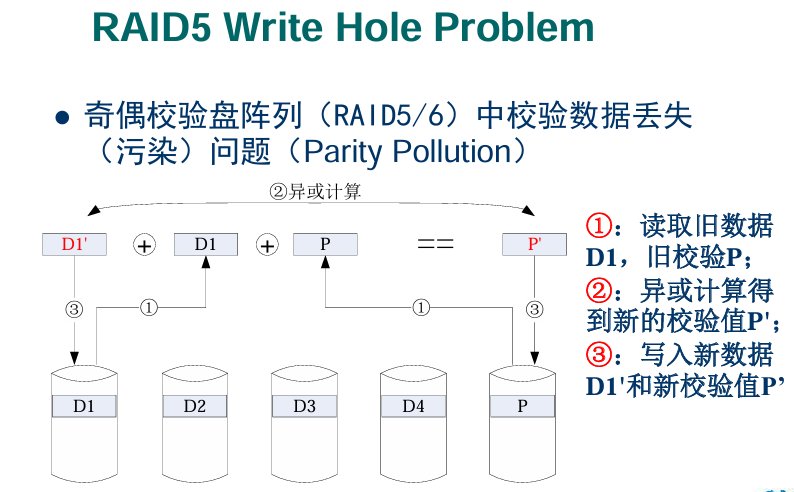
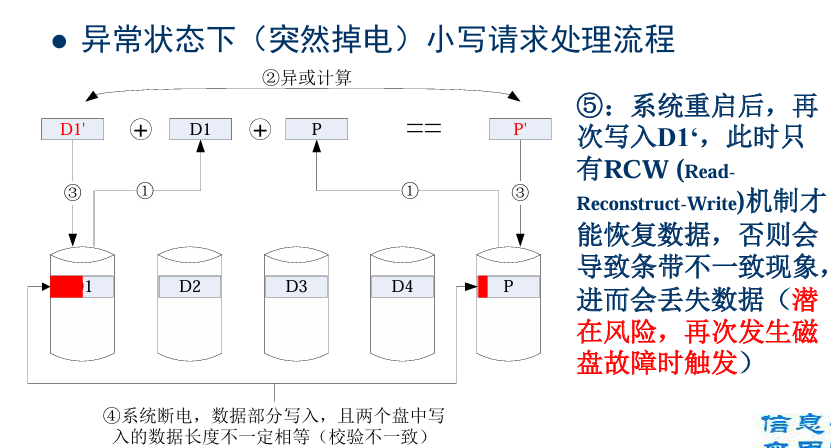
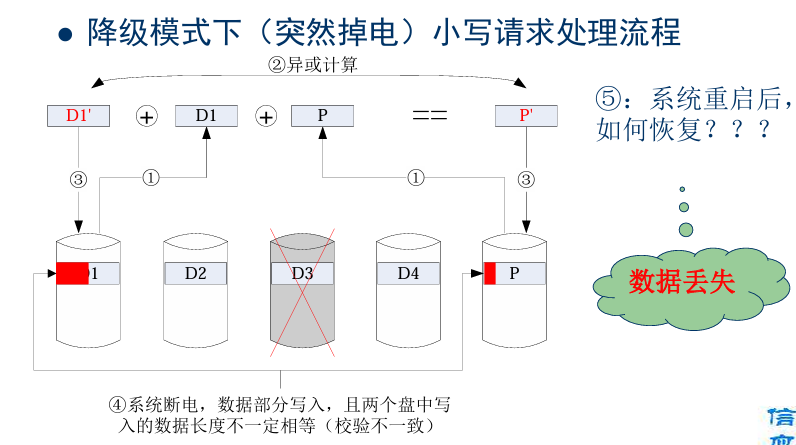
* SCSI命令 ****

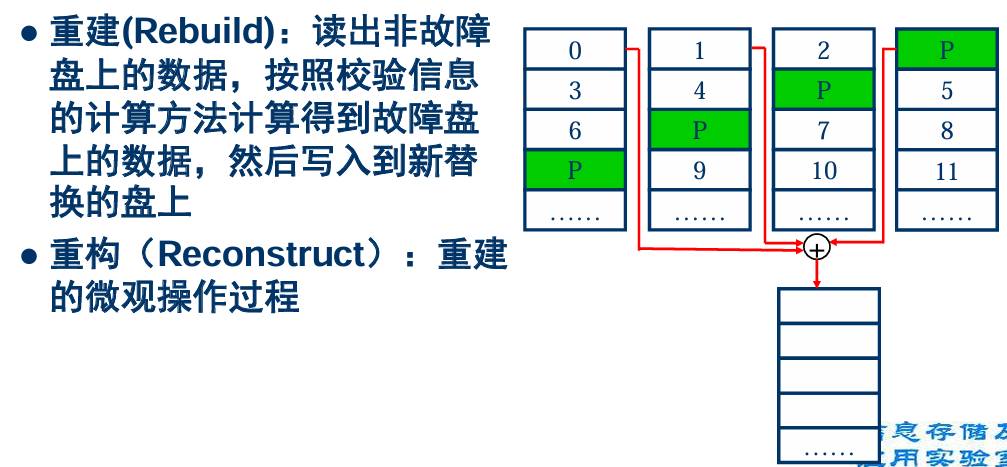
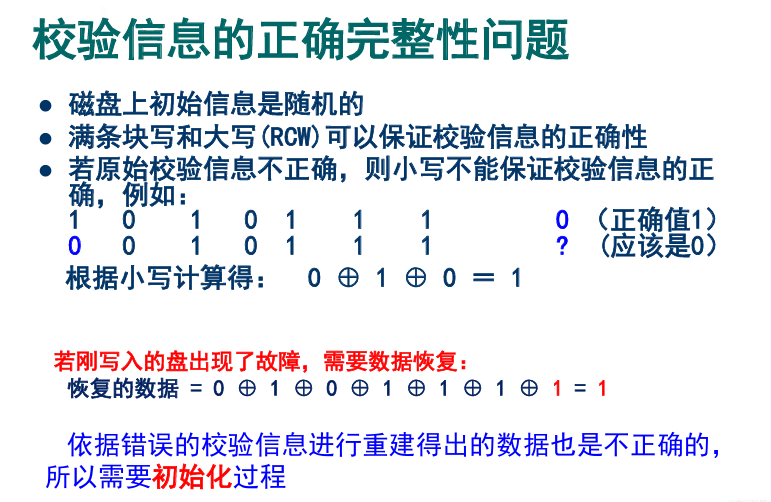
****

****

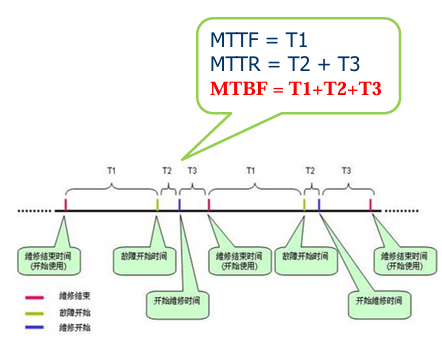
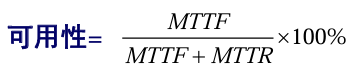
****

* 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： **  
  **

**  
** ➢ 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。  
  
  
  
  
 ➢ 小写的过程(Read-Modify-Write, RMW) **  
  
  
** ➢ RAID5的重建与初始化

**  
  
**

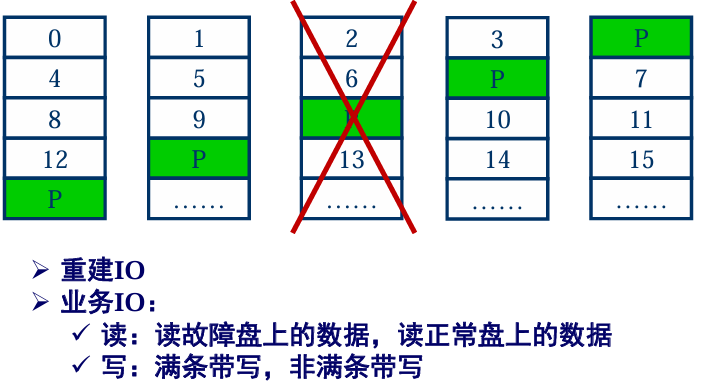
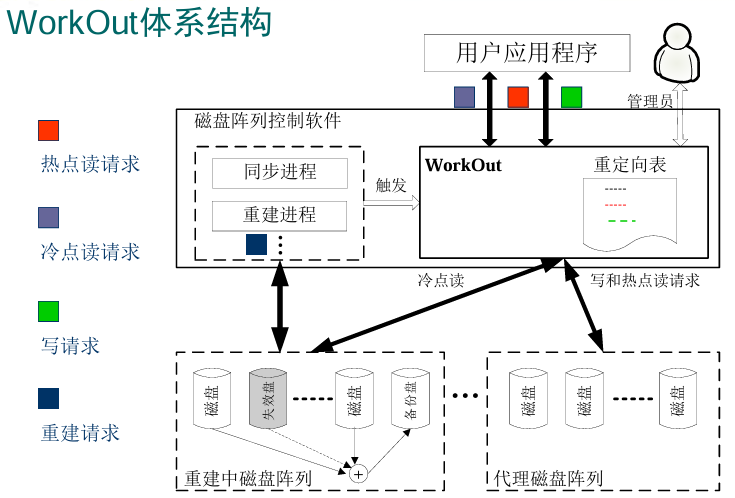
1. RAID高可靠、高可用技术

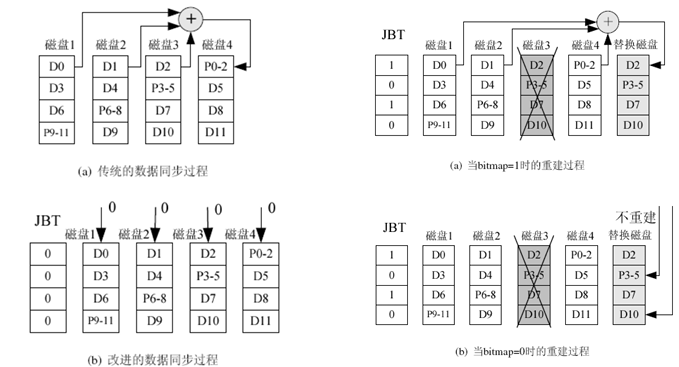
* 评估磁盘阵列可用性的方法➢ 从SLA(Service-Level Agreement)角度看，系统在两种 状态间切换：  
   – 服务完成，即服务按照指定的SLA交付；  
   – 服务中断，即提交的服务不满足SLA的要求。   
  ➢ 评估指标：  
   – 可靠性：评估服务完成，指从初始状态起，持续完成服务的时间，一般用MTTF评估  
   – 可用性：也评估服务完成，当系统从“完成”切换到 “中断”时，同样认为系统可用性降 低；传统的磁盘 阵列可用性形式化定义为MTTF和MTTR的函数 ****
* 提高磁盘阵列可用性的相关技术磁盘阵列数据重建算法研究：

– 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

* 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

1. 重建过程的优化方法（PRO）

RAID5➢ 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，体验差重建时间和用户I/O响应时间都随 着用户请求的强度的增加而增加  
用户I/O强 度对在线重 建的影响➢ 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➢ 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– 提高数据中心中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 ****

解决办法3： 基于日志的重建 JOR  
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 ****

企业需求实例： 1.2TB的数据，30分钟内要求完成重建: 1.2TB/30\*60s = 667MB/s   
解决途径: 多设备并行重建 ➢ 多设备并行重建 S^2-RAID: 用于快速数据恢复的并行 RAID 架构

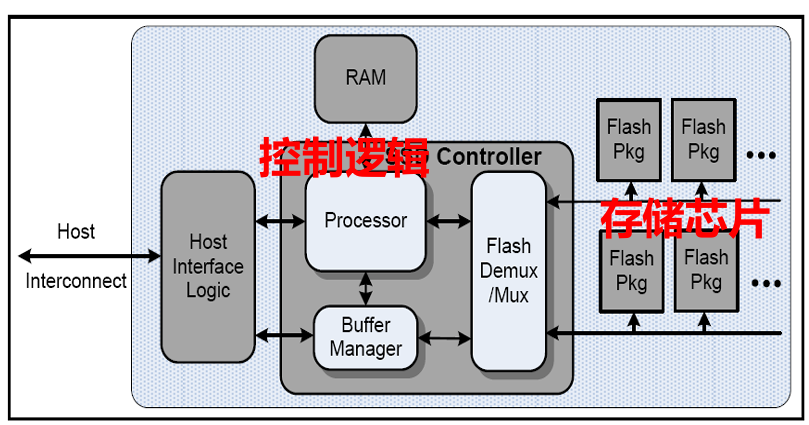
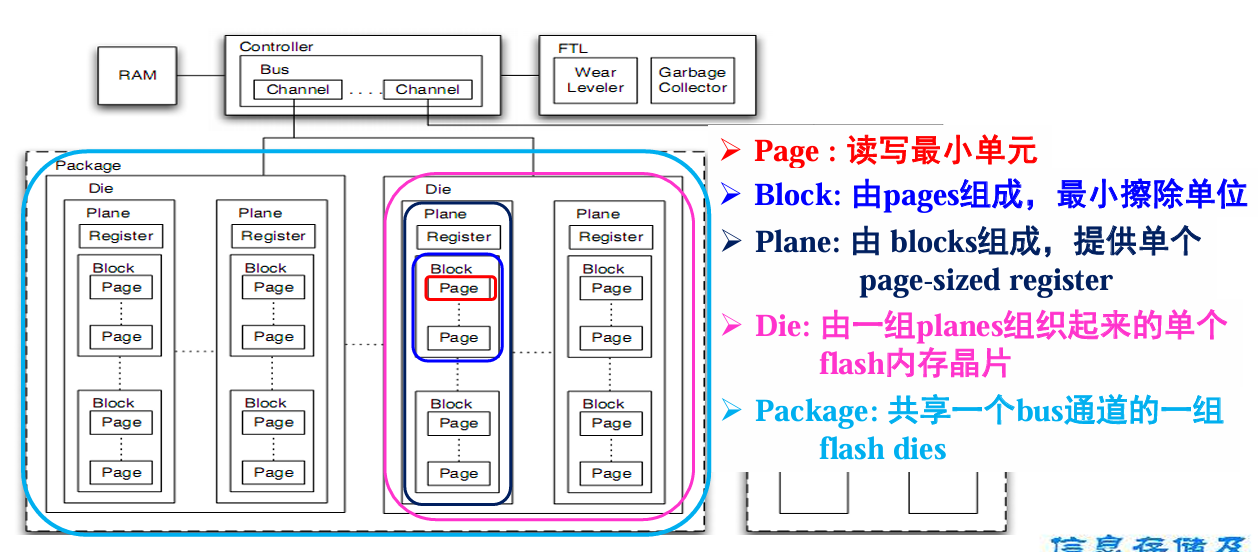
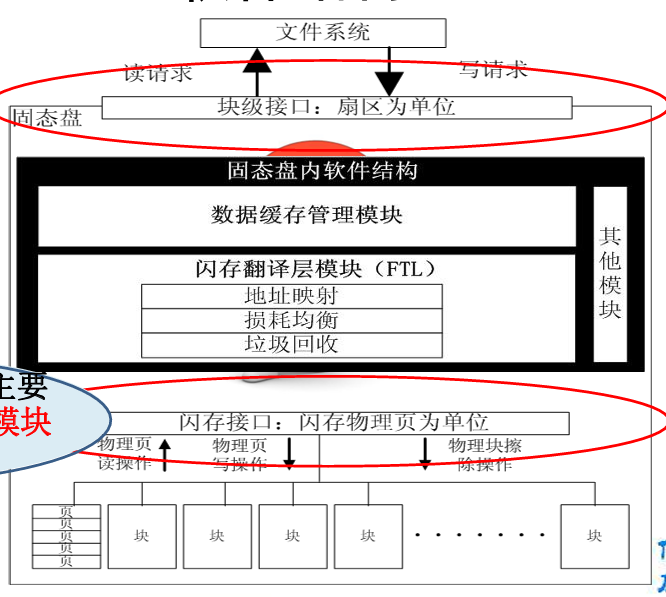
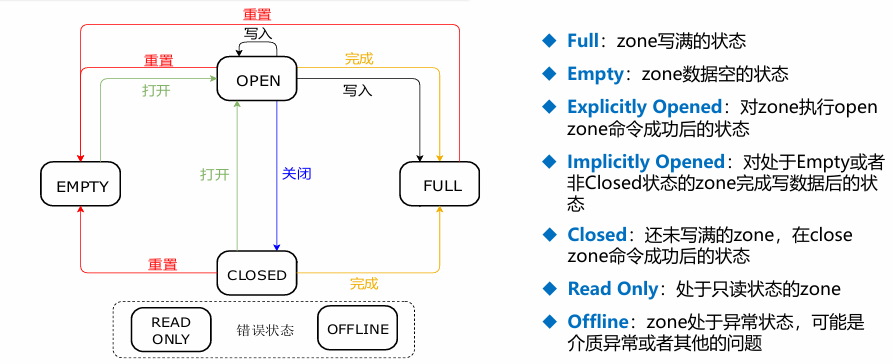
其核心思想是将 RAID 中的每块大容量磁盘划分为多个小分区。这些磁盘上的分区组成若干子阵列（sub-arrays）。这些子阵列以一种特殊的方式在 RAID 的所有磁盘间交错分布（skewed），从而在任意磁盘发生故障进行 RAID 重建时，能够实现无冲突的并行操作。

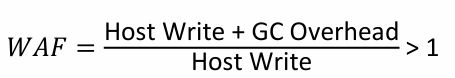
原本存储在故障磁盘上的数据，在恢复过程中会并行地写入多块磁盘，包括专用的备用磁盘（spare disks）以及正常磁盘上的可用空间

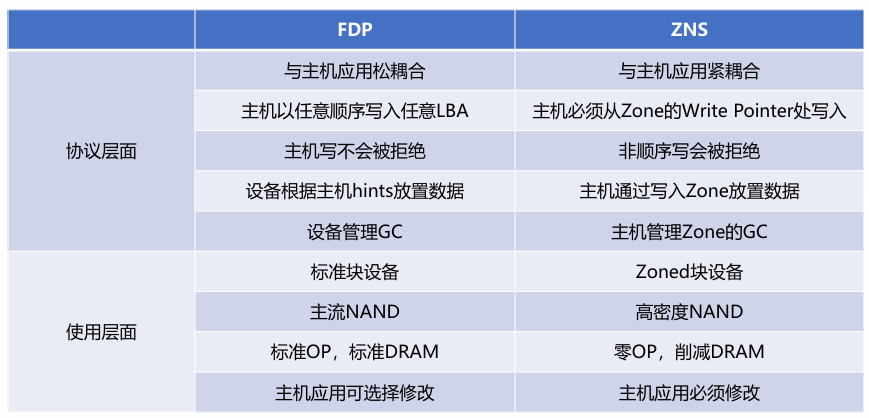
##### PPT 03 固态盘技术

1. 基于存储介质分类：   
   ➢ 基于闪存的固态盘：数据能够持久保持，掉电也能保持数据， 随机读性能好   
   ➢ 基于新型NVM的固态盘：数据能够持久保持，随机读写性 能好，读写延迟低，如傲腾固态盘   
   ➢ 基于DRAM的固态盘：读写速度快，但需要独立的电源来 保持数据安全，需要备份硬盘来长久地存储数据

➢ 混合介质：由DRAM、闪存、NVM混合构成的固态盘

1. SSD内部结构 【处理器；DMA；主机接口控制器；通道控制器】 ****
2. 【2020】NAND FLASH 连接 ****
3. SSD优点   
   • 在便携式环境中的可靠性、无噪音 – 无电机马达   
   • 快速启动 – Does not need spin up   
   • 低读延迟（low read latency）– 没有寻道时间(xus per page/4KB)   
   • 确定性的读性能 – The performance does not depends on the location of data   
   • 静态功耗低、发热量低
4. SSD缺点   
   • 单位容量成本更高 – 0.1$/GB vs. 0.05$/GB   
   • 有限的写入擦除次数(Lifetime)  
    – 100000 writes for SLC (MLC is even fewer)  
    – high endurance cells may have an 1-5 million  
    – But some files still need more  
    – Weaver leaving to spread writes all over the disk   
   • 由于擦除块而导致的写入速度越来越慢(1.5 msper erase)   
   • 延迟不可预测性(Unpredictable Latency)   
   • 写放大(Write Amplification)
5. SSD软件结构  
   固态盘内部软件有两个主要模块：1. 数据缓存管理模块 2. 闪存转换层模块（FTL） ****
6. Flash Translation Layer （FTL）  
   SSD是以硬盘的替代者的姿态出现的，为了与现有系统无缝对接，SSD必须对外提供的是块接口，作为主机端，所看到的SSD是一个和HDD一样的块设备。为了达到模拟块设备的目的，SSD中需要FTL作为中间层。FTL从主机文件系统接收块级请求（LSN, size），经过FTL的处理，产生flash的各种控制命令。a. 定义 –使闪存看起来像磁盘驱动器的软件层  
   b. FTL的挑战(来自闪存介质）  
    – 三大基本操作单位不同：读（页）、写（页）、擦除（块）   
    – 不能就地覆盖：闪存必须先擦除才能写入（Program） 【来自上层的写请求】
7. FTL由三部分组成：地址映射、损耗平衡、垃圾回收
8. Address mapping (地址映射） **-** 上层文件系统发送给SSD的任何读写命令包括两个部分（LSN，size）   
   - LSN是逻辑扇区号，对于文件系统而言，它所看到的存储空间是一个线性的连续空间。例如，读请求（260，6）表示的是需要读取从扇区号为260的逻辑扇区开始，总共6个扇区。   
   - 请求到达SSD后，需要经过地址转换，将逻辑扇区转换成NAND FLASH 中的物理页号  
    <package，die，plane，block，page> **-** 地址映射分类 ：页映射(page-level) 块映射(block-level) 混合映射（hybrid-level)  **a.** 页映射算法思想：构建一个逻辑页(page/sector) 和物理页映射表来记录映射关系。 (类似全相关 cache)   
    优点:能够将任何逻辑页映射到物理页– 有效的提高flash页利用率   
    缺点: 映射表过大  
    – 2TB flash, 4KB flash page，映射表大小 超4GB  
    – 当flash更大时, SRAM要扩展  
    – 价格昂贵 **b.** 块映射算法  
    思想：构建逻辑块和物理块的映射表，逻辑页和物理 页在对应块中偏移相同。   
    优点：映射表小 – 映射表减小N倍（N = block size / page size )   
    缺点：页在块中偏移固定，灵活性差，利用率问题 – 垃圾回收开销增大
9. 高性能地址映射方法（DFTL 隐藏翻译过程映射算法） **  
   -** 页级映射算法的性能最佳，在高性能的固态盘设计中，大多采用或者基于页级映射。为了减少映射表大小，DFTL出现了。DFTL是基于页级映射的映射算法，是目前性能、寿命、成本综合最优的闪存转换层算法；性能、寿命基本接近或等于页级映射算法 。  
   - DFTL是基于负载的局部性原理，将经常访问的数据的映射关系存放在内存中，通过这种方式减少映关系占用内存的容量。DFTL依赖于局部性，当负载的局部性下降，将导致系统性能急剧下降。 **-** DFTL出现性能下降的根本原因：映射关系数据的访问路径与用户数据的访问路径是 同一条路径，因此产生冲突 。 **-** DFTL的核心思想：将映射关系数据的访问路径与用户数据的访问路径分离。
10. 损耗平衡（Wear-Leveling）➢ Flash中每个块都有一定的擦写次数限制。故不能让某一个块被写次数较多，而其他块被写的次数较少，需要找一种方法：使flash中每个块被擦写的次数基本相同。  **a.** 动态损耗平衡：在请求到达时，选取擦除次数较少的块作为请求的物理地址。 **b.** 静态损耗平衡：在运行一段时间后，有些块存放的数据一直没有更新(冷数据）， 而有些块的数据经常性的更新(热数据)。那些存放冷数据的块的擦除次数远小于存放热数据的块。将冷数据从原块取出，存放在擦除次数过多的块，原来存放冷数据的块被释放出来，接受热数据的擦写。
11. 垃圾回收（GC） **a.** 为什么要回收：标记作废的数据虽然无效但仍然占据物理存储空间；物理存储单元在其被擦除之前不能被重新使用；主机对底层物理存储空问的占用情况一无所知。 **b.** 什么是垃圾回收：选择一个包含较多无效页的块，将该块中的有效页迁移到一个干净块中，然后将该块擦除以备再次使用。  
    c. 垃圾回收策略 ：➢空闲块阀值回收 ➢实时回收 ➢无效数据阀值回收 **d.** 垃圾回收方式 ➢选取哪些块作为回收块 ➢选取哪些clean块容纳回收块中有效数据 **e.** TRIM指令 ：告诉NAND闪存固态存储设备要擦除哪些数据的SATA接口指令  **-** TRIM 是从操作系统发送到SSD 的命令，它通知SSD 哪些数据块不再使用，允许 SSD 高效地执行垃圾回收，提高写入性能并延长使用寿命。  
     - 废弃文件在（inode/host表）中标记为无效，指令改变控件分配位图的设置。 **-** TRIM支持：操作系统和SSD，在某些情况下可能默认禁用（在某些RAID配置中） **-** TRIM的好处： 提高写入性能；延长SSD使用寿命；减少SSD的磨损均衡工作负载
12. Open-Channel SSD （SSD中buffer策略） **- 传统SSD**面临的问题：性能&空间利用率不足；I/O延迟不可预测&尾延迟问题；功能冗余问题；语义隔离/Semantic Gaps；【根本问题：SSD的硬盘式黑盒架构设计】补充：【I/O延迟不可预测问题】主机端发起的读写请求可能与SSD控制器的GC、WL 过程产生碰撞，从而导致主机的I/O延迟不可预测。这对于实时应用或延迟敏感的工作 是不可 接受的。【尾延迟问题】SSD在进行GC时，会阻塞主机的I/O，导致尾延迟问 题。【功能冗余问题】 元数据管理、地址映射、垃圾回收等功能模块在应用层、文件系 统层和设备层出现三重冗余；冗余问题会引起严重的写放大和性能损失 ；写放大问题 进一步导致SSD磨损和寿命问题。【语义隔离】由于冗长的I/O栈，底层设备与上层应 用之间 存在语义鸿沟，一方面底层flash设备的内部并行性不能被有效的开发利用另一 方面，上层应用不能结合本身的负载特性和专业知识针对底层flash进行优化。 **- Open-Channel SSD**指的是闪存转换层（FTL）不在硬件/固件设备上实现，而 是由主机（Host）负责维护的一类SSD。核心思想：将SSD内部并行性暴露给主机，并由主机进行管理。 **- NANDFlash的结构构成**：Packages->Dies/chips->Planes->Blocks->pages   
     Package：共享一个总线channel的一组flash dies/chips组成   
     Die：由一组planes组织起来的单个flash晶粒  **  
      
    - 存在的问题**：对于主机端管理的OCSSD，无法给用户提供设备担保（Device Warranty）；直接将介质特性暴露给主机，增加主机负担，可能导致编程效率低下，并且限 制了介质抽象 ；OCSSD缺乏统一的标准，尚无成熟的商业产品，接口的兼容性问题
13. ZNS Zoned Namespaces SSD  **- ZNSSSD**区命名空间固态硬盘，将SSD的存储空间划分为逻辑分区Zone，与擦除块对齐。每个Zone具有独立的读写特性和管理策略，每个zone可以按任意顺序读 取，但必须按顺序写入。如果需要更新分区，需要先将整个分区重置才能再次开始从头写入。 **-** ZNS 将区域大小（Zone Size）定义为区域内逻辑块的总数；区域容量(Zone Capacity)是每个区域的附加属性，指示每个区域内可用逻辑块的数量，从每个区域的第一个逻辑块开始（区域容量始终小于或等于区域大小）；Active zones 活动分区数量有限制：由于基于闪存的SSD的性质，同时活动的区域数量有严格限制，从而对并发性能有一定影响。 **- Zone的几种状态  
    - 优势** ：改善内部数据放→更高的写吞吐量、更低的QoS开销；无需内部GC→更大的容量，更低的成本 ；减少GC对性能的干扰和写放大的影响；性能可预测；不同Zone之间的I/O操作相互独立，实现性能隔离 ；减少了写放大和擦除次数，延长了SSD的使用寿命  
    - **研究热点** ：性能调优；文件系统和数据库支持；ZNS与其他存储技术的结合；安全性与数据持久性；硬件与成本的优化。
14. FDP lexible Data Placement **- 目标**：期望把SSD盘的的写放大能够降低到1  **- 是什么在损耗SSD的性能与寿命**？  
    写入放大（**WAF**）= SSD实际写入量/ 主机请求写入量；理想情况:WAF = 1 (写多少就是多少) ，现实情况:WAF > 1 (写了1GB，SSD内部可能写了2-3GB甚至更多) ；根本原因:擦除与写入粒度不对称，垃圾回收（GC）需要搬运有效数据，产生额外写入。

**  
 -**  **核心思想**：从混乱到有序。传统SSD:所有数据扔进一个垃圾桶->混合混乱->回收麻烦； FDP SSD: 数据在扔之前就分好类-> 专人专桶-> 回收高效。

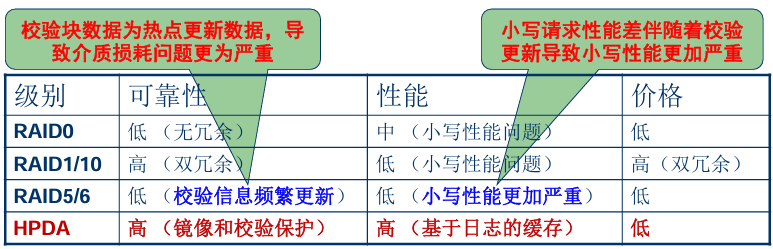
基本思路：Host侧的应用对数据进行识别 ↓ 写入SSD时，分类写入不同的空间 数据废 弃后，以Reclaim Group为粒度进行 擦除回收SSD的NAND空间 ↓ 垃圾集中起来回收 的效果 减少了对NAND FALSH的block做GC过程 中对有效数据的搬移操作，提高 效 率，降低写放大WAF。  
 - FDP的“**工具箱**”：回收单元(RU) 一个专用的“垃圾桶”，由多个 NAND块组成；回收单 元句柄 (RUH) 垃圾桶的编号和写入指针。主机通过选择RUH来选择垃圾桶；放置句柄 (PH) 一个命名空间 能使用的所有RUH的列表。  
 - 具体实现：①SSD可以将并发NAND拓扑划分为多个Reclaim Group，实现性能隔离 ②将SSD内的NAND物理block（或者Super block），划为Reclaim Unit ③Reclaim Unit Handle指向不同的Reclaim Unit，可以理解成写入不同的NAND物理block的写指针。  
 从系统应用看来，就是在标准IO写时，通过DSPEC标识，指定写入到特定性能隔离区域 Reclaim Group里面，放置到按类别（不同事务类型或者冷热度）区分的Reclaim Unit （即NAND block）   
 - **主机与SSD**如何协同工作？  
 **主机**：智能决策，知道数据是热是冷，100%决定数据放到哪。下发NVMeWrite命令，并 携带RUH参数。**SSD：**高效执行，负责物理存放。将数据顺序追加 到RUH指定的RU 中，并更新地址映射表。SSD对外暴露RUH概念，但隐藏RU内部的物理管理细节。主机: “随机写入”，可以不顾LBA顺序，任意指定LBA，向同一个RUH发送，灵活性高SSD:“顺序追加”，将这些LBA混乱的数据，按其到达的时间顺序，紧密地、顺序地排列在 同一个RU的物理位上。保证了同一RU内的数据物理上集中存放，为高效回收奠定基础。 **- FDP SSD**: 由于 单个RU内所有数 据生命周期相似， 它们几乎同时失效（变成垃圾） SSD无需搬运任何数据，可直接擦除整个RU。  
 - **GC特点**：①FDP写方式不与Reclaim Unit对齐，且允许SSD盘内GC；FDP定义 Reclaim Unit Open时间，超过时间会造成Reclaim Unit切换 且SSD盘内由于异常处理 等原因造成的Reclaim Unit切换都要事件上报 （FPD要求更严格的多流或者IOD模式） ②FDP通过一些机制，避免与系统应用GC冲突，尽量减少SSD盘内GC： FDP定义 Estimated Reclaim Unit Time Limit (ERUTL)(用于表示Reclaim Unit写入 后到被SSD盘 内主动回收的时间) 在未到时间前，系统应用主动回收，就不会触发SSD盘内GC 热数据 /前台写入数据：一定时间内系统应用进行整理回收 冷数据：系统应用不会搬移，SSD内 部会根据NAND特性主动进行GC和Wear Leveling 对于SSD应用能力较强的系统，可 以通过感知Reclaim Unit粒度，根据Reclaim Unit 制定系统GC策略，从而获得更好的效 果收益：FDP通过与系统间的模糊策略交互，使SSD盘只有弱GC，减少写放大，降低 SSD盘 的OP。同时也减少系统和SSD两层GC冲突，保障系统侧的时延和QoS  
 - 思路对比：FDP的思路：SSD需要重构FTL表映射的架构 不再是以4K、8K等方式映 射，至少以single block为粒度分配空间，与新增 的Reclaim Group对应，这样才能将生 命周期类似的一类数据写入相同的Reclaim Group-->block Group，以达到同时擦除的目 的 ①与open channel/ZNS相比：FDP的Host的应用和驱动无需看到非常多的SSD FW 的实现细节 (NAND空间管理、SUPER BLOCK组织、GC等) 实现上会简化一些。但是 对FW的架构影响较大 ②与多流相比：多流是以SUPER BLOCK与一条流对应，粒度较 大，一个SUPER BLOCK的空间都是以GB为单位 而FDP可以用更小的粒度对数据进 行分组，比如以single block分组，可以 在MB级别。  
 

- 总结：ZNS和FDP代表了两种前进方向上的最新标准   
 - 将SSD内的闪存阵列结构暴露给主机：Open-Channel、ZNS等   
 减少了映射表所需DRAM空间、预留配 (OP)空间，硬件成本低   
 更改了块接口，应用需要严格遵循Open-Channel或ZNS的写入约束， 软件成本高 - 主机提供建议，SSD视自身情况采纳建议：Multi-Streams、IOD、FDP等 •  
 保留了映射表、OP空间，硬件成本不变，仅通过数据布局减少写放大， 效果依赖软 件建议有效程度和负载特征   
 看系统应用的向下兼容，SSD可以拒绝FDP的数据布局建议，FDP软件成本低.  
 FDP硬件成本不变， 软件成本降低，可 选择开启或不开启。

1. SchInFS  
   【动机】：现有文件系统在 ZNS 上的并发写效率低下；并发写效率低的根本原因（无法有效利用多 Zone 并发带宽；单 Zone 内部并发写效率低）。【设计目标与建议】使用简单的 Zone Write 而非 Zone Append；采用多头日志（Multi-head Logging）设计；最小化 I/O 调度开销。【SchInFS 核心架构与机制】：多头日志设计；日志状态感知的块分配器；每日志合并队列；块 I/O 提交控制器。【结论】SchInFS 通过深入分析 ZNS 文件系统的并发瓶颈，识别出两大根本问题：无法有效利用多 Zone 带宽；单 Zone 内部调度效率低下。提出并实现了 集成块 I/O 调度功能的新型文件系统：基于 多头日志、每日志合并队列、BSC 控制器；实现高效、低开销的并发写入。

##### PPT 04 RAID 遇上 SSD

1. RAID for SSD 当写不友好遇上写不友好  
   SSD擦写次数受限，延迟波动大，尾延迟问题明显   
   RAID中数据尽量均匀分布到各成员盘   
   RAID5中校验数据块的擦写更频繁

* Differential RAID **问题：**成员盘故障集中爆发；数据不可恢复；RAID1、5、10、6都会出现 **方法：**打破平衡，让校验信息非均匀分布，造成寿命差异。在整个阵列中 不均匀地分布 奇偶校验块， 利用它们较高 的更新率控制 不同设备的老 化速率。为了在旧设备被 新设备替换时保 持这种年龄差异， 每次驱动器更换 时重新调整奇偶 校验分布。
* 混合式盘阵列 HPDA磁盘(HDD) ❑ 高能耗 ❑ 非对称的顺序/随机性能 ❑ 介质损耗可忽略固态盘(SSD) ❑ 低能耗 ❑ 非对称读/写性能 ❑ 介质损耗问题 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） **方法：**校验信息更新导致的 介质损耗问题减轻了；由于采用顺序日志镜像技术，减轻随机小写性能问题 新加入的磁盘和校验盘中的剩余 空间组成一个镜像缓冲区，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 ****
* I-CASH  **-** SSD（固态硬盘）：

读取性能高，尤其在随机读取方面表现优异

小规模随机写入会导致性能较低，并加剧磨损

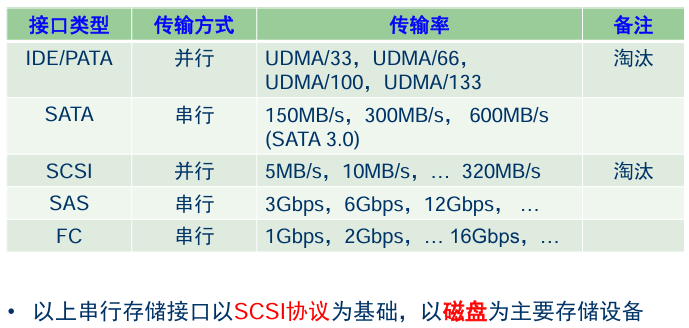
**-** HDD（机械硬盘）：顺序读写性能良好；随机读写性能极低；几乎不存在磨损问题

- 内容局部性（Content Locality）：近期研究文献指出，在许多数据密集型应用中存在 很强的内容局部性：在典型的块更新操作中，一个数据块内仅有5%到20%的比特位 实际发生了变化。  
 - I-CASH：智能耦合的SSD与HDD阵列。  
 SSD 存储很少更改、主要用于读取的参考数据块

HDD 存储当前访问的 I/O 块与其在 SSD 中对应参考块之间的差异日志（deltas）  
 - 在线 I/O 操作期间，SSD 不执行随机写入；存储在 SSD 中的参考数据块具有高速 读取性能；大量小规模的差异（deltas）可被打包进一个存储在 HDD 中的差异块， 并缓存在 RAM 中；利用 SSD 的快速读取性能和现代多核 CPU 的高速计算能力， 替代 HDD 的机械操作；避免运行时对 SSD 的写入操作，因为这类写入速度慢且会 加剧 SSD 磨损。

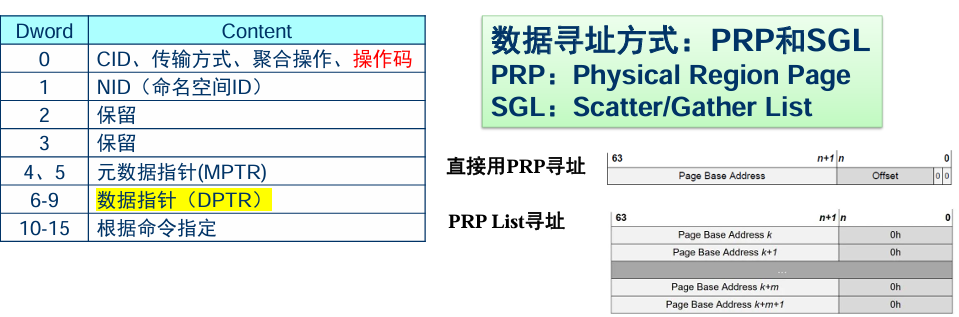
1. 全闪存阵列 AFAs All-FlashArrays **-** 研究聚焦 ➢缓解尾部延迟 ➢提高性能（Performance） ➢改进可靠性（Availability）
2. FusionRAID **-** SSD RAID性能问题：与HDDRAID相比，延迟抖动厉害；尾延迟突出；随盘的老化问题加剧 **-** 实验观察：工作负载通常不规则，交错突发（interleaving bursts），All-for-all 映射模式优于物理分区；SSD RAID 写入操作软件开销严重，相对开销比HDDRAID高，绝对开销远高于写RAM(Disk) ；SSD性能异常常见，写尤其突出（幅度和持续时间），延迟峰值高且持久，足以在运行时感知。 **-** Write Overhead in SSD RAID 1.软件开销在HDD RAID写过程中对延迟的影响可以忽略不计 2. SSD RAID 写入操作软件开销是写盘时间的2.9倍 •相对开销比HDDRAID高，绝对开销远高于写RAM（2个数量级） •软件开销还使最慢的1% 请求延迟增加为平均延迟的10 倍 3.在最慢的1% 请求中，SSD的写延迟对尾延迟贡献也很大，是平均 写时延的20.7倍【原因在于写同步 ，需要缩短写路径】 **-** 缩短写操作路径，减少依赖关系，可大大降低SSD RAID访问延迟, 包括平均延迟和最坏场景下的延迟
3. Asym-RAID （非对称RAID）  
   - 核心理念是异构性感知；不对称地分布数据，充 分利用每个SSD的容量 和性能。 1. 最大化向主机导出的有效容量 2. 性能优化的放置 **-** 问题：传统RAID架构面对异质性SSD的性能瓶颈，SSD无法在长时间内保持一致的性能，必须充分考虑SSD异构性对整个系统性能的影响

##### PPT 05 存储接口技术

1. 存储器件与设备变迁  
   - 新型固态盘：   
    ➢ 【特点】采用NVM（Flash、3D Flash、X3D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，CXL）   
    ➢ 需要高性能软件及协议：优化IO软件栈；高效的接口协议NVMe   
    ➢【优势】高性能表现：IOPS可达百/千万级、带宽达 x GB/s 级别、延迟降至微秒 级 （us）；访问特性优化：顺序与随机读写性能差异显著缩小，突破传统磁盘“顺序快、随 机慢”的限制；物理结构优势：无机械运动部件，可靠性更高，功耗更低，抗震性更好  
    ➢【需要解决的问题】需要高速物理接口：传统SATA/SAS接口存在带宽和延迟瓶颈， 无法匹配NVM性能。需要高性能软件及协议栈支持：优化IO软件栈：传统Linux块层 存在单一队列锁、中断处理开销大等问题，限制了高IOPS发挥（“单一CPU核只能承 受80万IOPS”，“传统块层受限于百万IOPS”）；高效的接口协议NVMe：替代AHCI， 实现更低延迟、更高并行度（多队列、深队列）、更简化的命令路径。协议与生态适配： 需要从SCSI/AHCI体系转向专为NVM设计的NVMe协议；支持NVMe over Fabrics 以实现网络化高性能存储（如RDMA、FC等传输）。数据寻址与传输效率：NVMe引入 PRP和SGL两种数据寻址方式，需合理使用以避免跨页访问开销，提升DMA效率。  
   - 传统存储接口 以SCSI协议为基础，以磁盘为主要存储设备   
     
   - SSD接口变迁  
   ➢SATA/SAS 简单、兼容性好；带宽、延迟瓶颈 SSD接口变迁；AHCI

➢ PCIe 缩短存储访问物理通路；高带宽、低延迟；NVMe；  
 PCIe 消除控制器延迟 NVMe 降低了软件延迟

1. NVMe接口/协议

* NVMe命令集，所有命令都是64B，具有相同的格式，某些字段根据命令的不同有不同的定义  
  
* 数据寻址方式PRP和SGL的区别：  
  ➢PRP只能映射到一个个物 理页，不能跨页   
  ➢SGL可以映射到任意大小的连续物理空间，具有更大的灵活性   
  ➢Admin命令的数据地址只能采取PRP的方式，I/O命令的数据地址既可以采取 PRP的方式，又可以采取 SGL的方式
* NVMe命令队列管理：  
  SQ由Host来更新， CQ则由NVMe Controller更新   
  IO SQ/CQ中可以有最多64K对IO SQ/CQ （可以多对一）  
  系统中只有一对Admin SQ/CQ (一对一）  
  1. Host提交新的Command 2. Host通知Controller提取Command 3. NVMe Controller从SQ提取Command 4. NVMe Controller执行从SQ提取的 Commands 5. NVMe Controller将Commands的完成 状态写入CQ 6. NVMe Controller通知Host检查 Commands的完成状态 7. Host检查CQ中的Completion信息 8. Host告知Controller已处理完成 Completion信息
* NVMe vs AHCI：更低延迟，更高IOPS，更低功耗
* 传统的I/O软件栈瓶颈：  
  ➢ 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 精简I/O路径；消除系统软件瓶颈；支持多队列、深度队列；随机读写4KB性能优势显著
* 支持存储网络 — NVMe over Fabrics：NVM Express over Fabrics, 不依赖于底层物理网络协议与连接。协议层支持：Fibre Channel Transport Protocol Layers；RDMA Transport Protocol Layers
* NVMe 的优势  
  IO吞吐率(IOPS)优势 ➢并行：多队列，队列深度；低延迟优势 ➢硬件 ➢软件；低功耗优势 ➢自动电源状态转换➢动态电源管理机制；扩展性优势 ➢NVMe over Fabric，不依赖于底层物理网络协议与连接

1. 存储系统体系架构

* DAS－Direct Attached Storage 直 接 连 接 存 储  
  ➢ 存储设备(通常为磁盘、固态盘、磁带等)通过 电缆直接与计算机相连, 系统存取访问I/O请求 (又称为协议或命令)直接在计算机和存储设备 间进行

➢ DAS 连接方式：独占式（dedicated）

➢ DAS 面临的挑战  
存储与主机必须直接连接 ：数据可用性问题– Single points of failure– Share data and resource ；扩展性有限–Number of connectivity ports to hosts–Number of addressable disks–Distance limitations ；停机维护问题

➢ 网络存储系统的必要性  
任何单个的存储设备都无法满足当今网络时代数字 化信息爆炸性增长的需求。通过网络（广义）将存储设备连接起来，并通过软 件来管理，才能满足容量、性能、价格、安全性等 方面的综合需求。【SAN, NAS, OBS, Cluster, Grid, Cloud Storage】

* SAN: Storage Area Network 存储区域网  
  ➢ 传统SCSI的弱点：距离的限制：小于12米；设备数量的限制：15／通道 ；带宽的限制；没有网络拓扑结构  
  ➢ 扩展受限制：无法满足对存储空间呈几何级数增长的需求；存储设备被服务器完全约束、无法共享；不适合作大规模的集群应用；备份必须通过局域网进行、或每台服务器连接独享的带库。  
  ➢ 为什么不使用LAN：【性能无保障，产生的背景A good LAN doesn’t make a good SAN】TCP/IP是为不可靠传输设计的网络，大量的校验；LAN 适合于短的、突发性的数据传输，不适合大数据量传输；LAN的用户产生大量广播数据包，不利于某些基于数据流的应用；LAN是不安全的、易受攻击；性能无保障

➢ 定义：【百度百科】是一种新的以数据存储为中心 的网络存储体系结构，采用可伸 缩的网络拓扑结构，通过光纤通 道连接方式为SAN内部任意节点提 供多路可选择的数据交换，并且 将数据存储管理集中在相对独立 的存储区域网内。【SNIA】SAN 是一种专用网络，其主要目的是在计算机系统与存储设备之间，以及存储设备彼此之间传输数据。【部分】它包含两部分：通信基础设施：提供物理连接；管理层：对这些连接、存储设备和计算机系统进行组织，以确保数据传输的安全性和可靠性。通常，术语 “SAN” 指的是提供块级 I/O 服务的网络，而非文件访问服务（如 NFS 或 SMB）  
➢ 优势：几乎无限的扩展能力；非常强健的设备连接；高速度、极高的传输效率；异构平台数据共享（需软件支持）；传输距离更远，可达数十公里服务器。

* FC-SAN  
  FC（Fibre Channel）是介于通道和局域网之间的一种接口，构成存储专用网络的基础。  
  ➢ FC的特点：➢ 五个协议层次：物理层（FC-0）、传输层（FC-1）、帧构造协议 和流控层（FC-2）、普通服务层（FC-3）、协议映射层（FC-4） ➢ 物理介质可以是光纤或电缆（宽带和基带电缆或屏蔽双绞线）， 连接距离为10米到10公里 ➢ 高性能：数据传输率高达8GBps以上，双通道数传率达到16GBps ➢ 具有网络拓扑结构：点对点，仲裁环，交换架构  
  ➢ 光纤通道拓扑结构：【点到点结构】：简单易用。【环路结构】：共享 100/200/400/800/1600MBps 带宽；每个 Loop 最多支持 126 个节点和 1 个 FL\_Port；使用 LIP 给每个设备指定 FC AL 地址；使用仲裁环路协议控制对介质的访问。【光纤交换】：每端口独享接口传输带宽 1/2/4/8/16Gbps；可以直接连接节点或接仲裁环路 集线器；支持各种高级服务，用于发现和监控设备；使用 3 个字节地址，可支持 1600 万个设备   
  ➢ FC 访问控制：  
  软分区（Soft Zoning）：利用光纤通道（Fibre Channel）网络中的 名称服务器（Name Server） 来限制响应发起端（initiator）查询时返回的信息。分区内的设备可通过以下任一方式标识：全局节点名（World Wide Node Name, WWNN）、全局端口名（World Wide Port Name, WWPN）、设备所连接交换机的 域/端口号（domain/port）。软分区仅控制“可见性”，不阻止非法通信；若设备获知其他设备地址，仍可能直接通信。硬分区（Hard Zoning）：由 Fabric（光纤通道网络结构）强制执行。交换机会监控通信流量，并阻断任何不符合当前有效分区配置的数据帧。阻断操作发生在目标设备所在端口的发送侧（即数据到达目标前被拦截），从而提供更强的安全性和隔离性。  
  ➢ 冗余、可用性与故障切换（Redundancy, Availability, and Failover）：  
  多路径（Multi-Path）：通过在主机与存储设备之间建立多条物理或逻辑路径，实现 I/O 负载的分发和路径冗余，提升性能与可靠性。完全冗余（Full Redundancy）：整个存储连接架构（包括主机总线适配器 HBA、交换机、存储控制器、电源等）均采用冗余设计，确保任一组件发生故障时系统仍可正常运行。路径故障切换（Path Failover）：当某条活动路径发生故障时，系统自动将 I/O 请求无缝切换至备用路径，保障数据访问的连续性和业务高可用性。
* IP SAN   
  ➢ Why IP-SAN  
  SAN 互联的市场驱动力——光纤通道（Fibre Channel）SAN 面临的挑战：高昂的采购成本、独立的基础设施、独立的管理模式。  
  IP SAN 的推动因素：可轻松利用现有的 IP 设备和专业技能，与光纤通道 SAN 协同管理数据。对于可容忍一定延迟的应用，IP 网络在成本最低的前提下提供了最大的灵活性。非常适合在校园网、城域网（MAN）或广域网（WAN）上进行远程备份。  
  ➢ IP SAN 的优势：  
  扩展 SAN 的覆盖范围： 标准光纤通道（Fibre Channel）传输距离有限； 借助 IP 网络，可将光纤通道应用延伸至区域级甚至全球范围；在更高链路带宽下，IP 网络还能支持同步应用（如同步复制）。

成本效益高：大多数组织已具备现成的 IP 网络基础设施，并熟悉传统的网络管理方式；可继续利用现有的光纤通道应用，实现平滑过渡与整合。  
➢ IP Storage Protocols：FCIP – FC over IP (tunnel)、iFCP–Internet FCP (map)、iSCSI、iSNS– internet Storage Name Service

* iSCSI  
  ➢ iSCSI 是一种原生的基于 IP 的存储块级协议：专为终端设备直接使用而设计；其命令仍遵循 T10 SCSI 规范；使用 TCP 作为网络传输协议；利用 IPSec 提供安全性；并非光纤通道（FC）封装协议。  
  ➢ iSCSI协议栈  
  IP Header → TCP Header → iSCSI Header → SCSI Data → IP Payload  
  ➢ iSCSI术语  
  - iSCSI 节点（iSCSI Node）—— 一个独立的、可寻址的 iSCSI 实体，可以是单个 iSCSI 发起端（initiator）或 iSCSI 目标端（target）。  
  - iSCSI 门户（iSCSI Portal）—— 网络实体中的一个组件，具有 TCP/IP 网络地址，可被该网络实体中的 iSCSI 节点用于建立其某个 iSCSI 会话中的连接。（发起端中的网络门户通过其 IP 地址标识；目标端中的网络门户通过其 IP 地址及其监听的 TCP 端口共同标识。）  
  ➢ iSCSI Discovery  
  iSCSI 发现（iSCSI Discovery）是指 iSCSI 发起端（initiator）获知有哪些目标端（target）的 iSCSI 节点名称对其可用的过程。  
  发现方法包括：  
  静态配置（Static Configuration）发起端被手动配置完整的目标准确信息，包括目标名称、门户（portal）地址等。

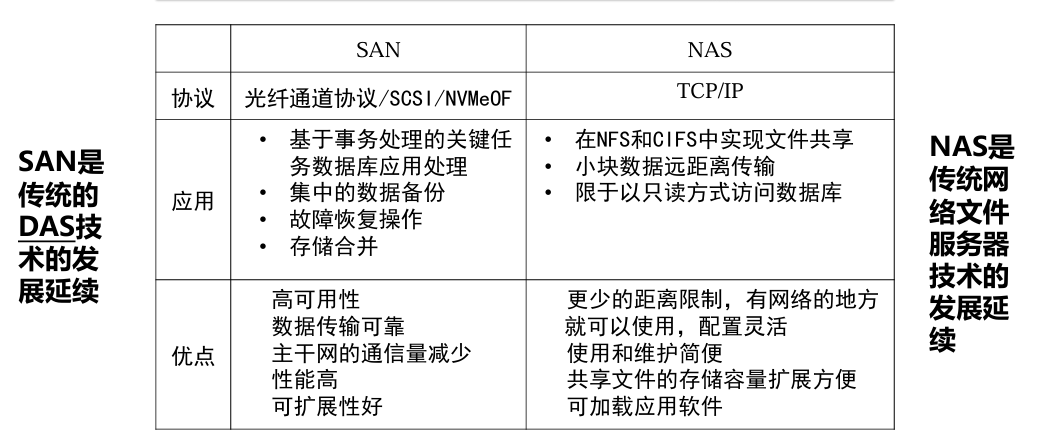
SendTargets：发起端被指定向一个发现用的 IP 地址发送查询请求，以获取可用的目标列表。

Internet 存储名称服务（iSNS, Internet Storage Name Service）：发起端通过 iSNS 服务器自动发现网络中注册的 iSCSI 目标设备。

* iSNS Internet 存储名称服务器  
  ➢ 概述：iSNS 采用客户端/服务器（client/server）模型；iSNS 服务器是被动的（不主动发起通信）；iSNS 客户端向 iSNS 服务器注册并操作其中的对象；iSNS 服务器可部署在目标端（target）、发起端（initiator）或具有指定 IP 地址的独立服务器上。  
  ➢ 控制 IP SANS
* 附网存储-NAS Network Attached Storage   
  ➢ 连接在网络上具备资料存储功能的装置，通过网络共享存储，也称为“网络存储器”  
  ➢ 是一种专用数据存储服务器   
  ➢ 以数据为中心，将存储设备与服 务器彻底分离，集中管理数据， 从而释放带宽、提高性能、降低 总拥有成本、保护投资  
  ➢ NAS系统结构与技术特点  
  【物理连接】上，将存储器直接接到网络上，不再挂 在服务器后端，避免了给服务器增加I/O负载。【技术】上，通过专用软件提供高性能的文件服务。  
  因此，NAS是一个专用的文件服务器，因其“专”而I/O性能高，但从NAS的构成看，其I/O 路径在本质上仍然是传统的服务器－存储设备 I/O方式——文件服务器。

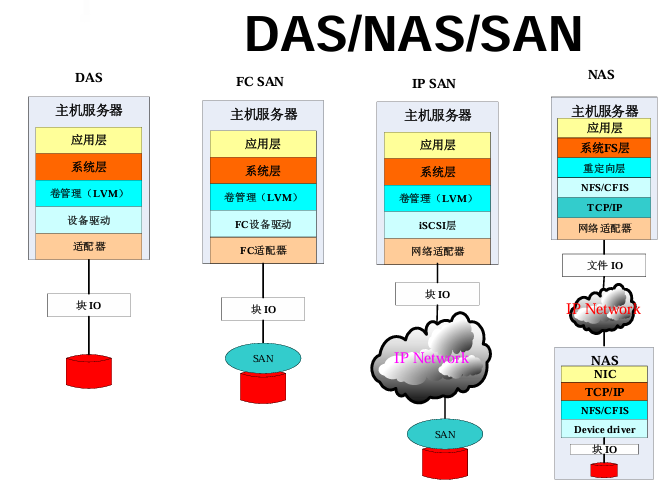
➢ 传统服务器 vs NAS设备  
NAS使用网络和文件共享协议， 实现文件归档和存储等功能   
➢ NAS 文件服务协议: NFS和CIFS  
➢ NAS系统的优点：1. 支持全面信息存取：文件共享更加高效，支持NAS设备和客户端的多对一和一对 多的配置方案 2. 高效率： 消除了通用文件服务器在文件存取过程中的性能瓶颈。因为 NAS使用的是为文件服务特制的操作系统，它将通用服务器 从繁重的文件服务操作中解脱出来，提高了通用服务器的利用率 3. 增强灵活性： 使用工业标准协议，NAS同时兼容UNIX和Windows平台的客户端4. 集中式存储，更安全更可靠 将客户工作站上的数据冗余降低到最小，简化了数据管理，确 保了更强的数据保护能力 5. 管理简单化 提供一个集中化的控制台，使得文件系统的管理更加高效 6. 可扩展性 NAS的高性能和低延迟设计，使得它能够针对不同的使用特征 和商业应用类型进行部署，具有良好的可扩展性，即插即用

➢ 网络连接存储扩展：NAS Gate Way，NAS网关提供了NAS应用扩充的方法，其本身不含 有存储磁盘，但是可以通过连接磁盘阵列整体作为 NAS功能，并可以提供将NAS连入SAN的特性

➢ SAN与NAS融合 SAN正与NAS结合，一起用来构成高性能、大 容量的存储系统，其表现是SAN隐藏在NAS后 面，用来实现关键数据的存储可用性和备份要求  


➢ NAS产业及应用现状：NAS市场规模及趋势预测；AIGC的发展给存储带来新机会；HPC促进高性能NAS市场；HPC促进高性能NAS市场

➢存储技术趋势-数据访问模式改变、网络架构越来越复杂性、所需存储容量成倍增长



##### PPT 06 对象存储

1. 数据访问分类举例

【块范式】：DAS, SAN, SCSI，ATA SAS，SATA FCP，iSCSI，SRP，NVMe...  
【文件范式】：Local FS, Distributed FS:(NFS, CIFS, WAFS) Shared FS:(Cluster FS, SAN FS) Global, Distributed & Parallel FS【对象范式】：OSD, CAS, OSS

1. 传统块范式与新对象范式对比



传统模型：应用程序 → 系统调用层 → 文件系统 → 存储管理 → LBA层 → I/O块管理器 → 介质

OSD模型：应用程序 → 系统调用层 → 文件系统 → OSD层 → OSD存储管理 → I/O块管理器 → 介质

1. Inodes vs. Objects  
   将存储的底层细节从管理员和应用程序中抽象出来。在对象中包含丰富的自定义元数据：支持用户或应用程序提供的特定信息，以实现更高效的索引；支持数据管理策略；跨多个独立节点和集群集中管理存储；独立于数据存储，对元数据的存储、缓存和索引进行优化。
2. 定义：对象存储是一种将数 据作为对象进行管理的计算机数据存储体系结构。与文件存储和块存储不同，每个对象通常包括数据本身，数量可变的元数据和全局唯一 标识符。对象存储可以在多个级别上实现，包括设备级别，系统级别和接口级别。
3. 对象存储的特性：性能优势；存储设备的智能化；数据的共享更容易；管理更方便；更好的安全性。是大数据、AI时代存储组织架构的理想选择
4. 6.对象存储的责任：空间管理、访问控制（身份管理）、服务质量管理、缓存与备份、策略驱动的数据迁移与保留。
5. 对象存储系统实例： Lustre

* 一种用于高性能计算（HPC）集群的共享文件系统；具备极高的元数据处理和 I/O 性能；可扩展至数千个节点
* 主要的设计问题：I/O 吞吐量 —— 如何避免性能瓶颈、元数据可扩展性 —— 如何让数万个客户端同时操作同一目录中的文件、集群恢复能力 —— 若发生故障，如何实现透明的自动恢复、管理运维 —— 包括系统的添加、移除与替换，以及数据迁移和备份

1. 当前一代对象存储

* 云存储（Cloud Storage）是用于计算云基础设施的存储。计算云对 IOPS 要求很高，通常在这些应用中使用块存储。  
  存储云（Storage Clouds） 是“云中的存储”，无论是公有云还是私有云。存储云只是通过互联网提供的存储容量。当今大多数存储云都使用对象存储技术。  
  横向扩展的对象存储（Scale out object storage） 使用简单的 REST API。
* REST API  
  REST 代表 Representational State Transfer（表述性状态转移）。它是一种用于分布式应用环境的软件架构。REST API 已成为云应用连接到云的主流接口。对于以存储为中心的云应用，REST API 是应用与对象存储平台之间的接口。常用命令包括：PUT、GET、DELETE...
* 总结  
  数据以对象的形式存储在一个大型、可扩展的存储池中；对象与其元数据（关于该对象的信息）一起存储；存储一个对象 ID，用于定位数据；REST 是标准接口，应用程序使用简单命令进行操作；对象是不可变的，编辑会保存为一个新对象。

1. 对象存储与键值存储对比

* 相同点：对象ID类似于Key（任意字符串）；数据可以具有任意大小
* 不同点：对象存储还允许将一组有限的属性（元数据）与每个(数据)对象相关联；对象存储针对大量数据（数百兆甚至千兆字节）进行了优化，而对于键值存储，其值则相对较小（千字节）；对象存储通常提供较弱的一致性保证，例如最终的一致性，而键值存储则提供较强的一致性。

1. 块、文件和对象

* 块存储（DAS/SAN）

➢FC、iSCSI协议、NVMe-oF

➢专用系统中，高读写性能和高可靠性   
 ➢一套存储只服务一个应用系统，例如 如交易系统，计费系统。典型行业如 金融、 制造、能源、电信等

* 文件存储（NAS）   
  ✓NFS/CIFS协议，IP网络   
  ✓兼顾多个应用和更多用户访问， 同时提供方便的数据共享手段   
  ✓中小企业市场，CRM系统、 SCM系统、OA系统、HPC、 AIGC等
* 对象存储（Object） ➢OSD，HTTP协议 ➢互联网或者公网 ➢海量数据，高并发访问 ➢常见的适配应用如网盘、媒体娱乐、 医疗PACS、气象、归档等数据量超 大而又相对“冷数据”和非在线处理 的应用类型
* 数据处理的演变：Moving Data to the Processor is Costly；Opportunities for 1000x improvement are increasingly rare；One floating-point calculation: 17 picojoules；Moving data from DRAM to CPU: 17,000 picojoules

1. 近数据处理 Near Data Processing  
   近数据处理技术作为一种存算一体思想的新型算力，有望解决冯·诺依曼架构下存储墙等问题

* 根据计算与存储的距离远近分类： ▶近存计算PNM ▶存内处理 PIM ▶存内计算CIM
* NDP-近存计算

近存计算是一种通过将计算逻辑靠近存储设备实现高效数据处理的计算机架构技术，旨在突破传统冯·诺依曼架构的瓶颈。 其核心是通过减少数据传输距离和频率来降低延迟并提升能 效。--通过芯片封装和板卡组装等方式，在内存或存储设备附近集成计算 单元，增加访存带宽、减少数据搬移

近存计算仍是存算分离架构，计算操作由位于存储外部、独立 的计算单元完成，其技术成熟度较高，主要包括存储上移、计 算下移两种方式

* NDP-存储上移  
  用先进封装技术将存储器向处理器(XPU)靠近，增加计算和存储间的链路数量，提供更高访存带宽。 典型代表：HBM(High Bandwidth Memory)，将内存颗粒通过硅通孔多层堆叠实现存储容量提升， 同时基于硅中介板的高速接口与计算单元互联提供高带宽存储服务
* NDP-计算下移

采用集成技术将数据处理能力卸载到存储器， 由近端处理器进行数据处理，有效减少存储 器与远端处理器的数据搬移开销。典型的方 案为可计算存储CSD(Computational Storage Drives)

CSD在SSD内部引入的计算引擎功能，主要功能包括：压缩和解压缩、加密和解密等。e.g., 压缩和解压缩，会在主控将数据写到后 端NAND颗粒之前对要写入的数据进行压缩；反过来读取的时候从NAND颗粒读出的数据再进行解压缩，节省空间，降低损耗。

* NDP-存内处理

存内处理是在芯片制造的过程中，将存和算集成在同一个晶粒中，使存储器本身具备了一定算的能力。存内处理本质上仍是存算分离，相比于近存计算，“存”与“算”距离更近。

当前存内处理方案大多在内存芯片中实现部分数据处理，较为典型的产品形态 为HBM-PIM和PIM-DIMM，在DRAM Die中内置处理单元，提供大吞吐低延迟 片上处理能力，可应用于语音识别、数据库索引搜索、基因匹配等场景 。

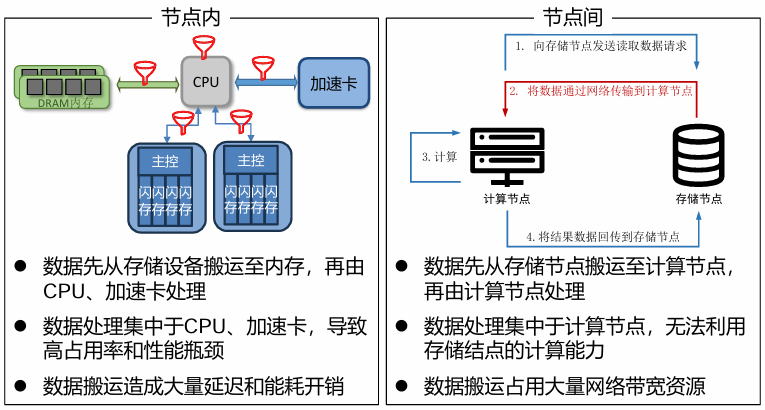
存内计算即狭义的存算一体，在芯 片设计过程中，不再区分存储单元 和计算单元，真正实现存算融合。

存内计算是计算新范式的研究热点， 其本质是利用不同存储介质的物理 特性，对存储电路进行重新设计使 其同时具备计算和存储能力，直接 消除“存”“算”界限，使计算能 效达到数量级的提升。

典型应用场景是AI提供向量矩阵乘 法的算子加速，例如卷积神经网络 CNN，循环神经网络RNN

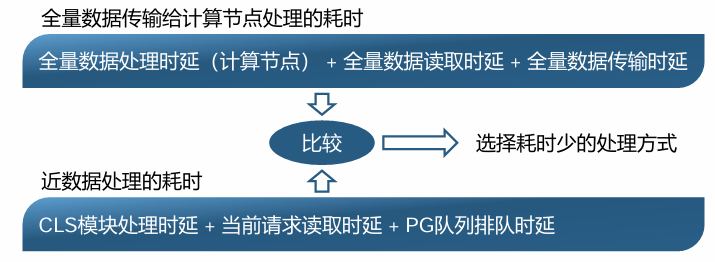
* 总结  
  目标：将计算资源放置在数据存储的更近处  
  优势：减少数据移动开销、降低延迟、提升系统性能等

1. 主动对象存储  
   传统存储系统被动响应服务请求，对象具有智能性，智能的系统能够提供主动服务

* 存储组织结构对比  
  传统模式：只含数据，主机组织和管理，被动响应。效率低，管理复杂。  
  主动对象：含"数据、属性"和"操作"，自组织和自管理，主动服务。性能高，管理简单。
* 主动对象特点  
  自我学习和策略触发机制，实现存储主动服务；自组织与管理，自我优化调节，使系统整体性能最佳；利用对象"封闭"特性，使系统具有安全性；提高系统可靠性，在故障出现时实现快速恢复。【自组织、自优化、自愈】
* 负载模型/预测模型：根据历史负载 预测未来趋势（AI）
* 主动对象实例：近数据处理  
  数据总量爆炸式增长，数据搬运成为瓶颈。问题：如何在缓解集群网络压力同时提高查询算子性能？方法：建立基于系统运行信息的算子收益评估模型，实现算子动态卸载至存储节点的方法。  
   

查询算子执行流程：

librados API对象存储层→Chunk Store→K/V Store→CLS Plug-in→R/WData Object

基于收益评估的算子动态卸载方法：  
算子卸载收益评估：根据系统运行时信息动态判断算子卸载收益；判断方式：预估算子交给计算节点处理和近数据处理耗时；选择耗时更少的处理方式。  
 

##### PPT 07 存储管理与数据保护

1. 四个重点：存储虚拟化；软件定义存储；数据保护；重复数据删除
2. 存储虚拟化   
   对存储硬件资源 进行抽象化表现，一种对细节的抽象，用于分离各层。  
   SNIA: Storage virtualization is an abstraction of storage that separates the host view from the storage system implementation.

* 行为：对主机隐藏以下信息 ➢ 物理路径 ➢ 设备特性 ➢ 数据的物理位置

提供位置透明性和实现透明性

支持动态操作：➢ 支持透明的“实时”重新配置   
 ➢ 允许数据位置在主机环境中透明地变更

* 优势：对新型服务器、网络和存储技术具有开放性；显著减少计划内和计划外停机时间；提高存储资源利用率 – 降低存储资本支出 – 降低管理复杂性；（潜在地）提升性能 – 负载分散、负载均衡、多路径访问、启发式数据迁移；改善存储服务质量（QoS）
* No Config/Reboot/Downtime
* 磁盘（驱动器）虚拟化 三十年来在磁盘驱动器中已广泛应用

物理数据布局

– 柱面（Cylinders）、磁头（Heads）、扇区（Sectors）

– 存储介质缺陷（Media defects）

虚拟数据布局

– 编号块组成的线性数组

– “无缺陷”视图（对主机呈现） vs. “缺陷映射表”（内部管理）

实现方式 – 由磁盘固件（firmware）实现

对外接口 – 通过 ATA、SCSI、SAS 等标准协议对外暴露

* 将物理设备转变为“虚拟设备”  
  物理磁盘 – 容量固定；性能受限；偶尔会发生故障

虚拟磁盘（通过块级虚拟化转换）

➢ 容量可按用户需求任意设置：更大、更小，或创建任意数量

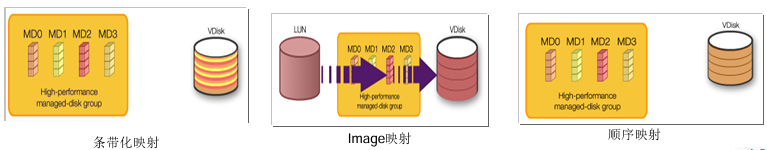
➢ 性能可满足用户和应用程序的需求

➢ 可靠性可根据用户和应用程序的需求进行配置  
 ➢ 支持动态扩展（！）、收缩或变形（morph）

* 存储虚拟化可以哪几个层次实现？

主机（Host）层、网络（Network）层、存储设备（Storage Device）层  
“In-band vs Out-of-band”（带内与带外）

* SVC SAN Volume Controller  
  SVC将多个磁盘存储控制器管理起来，将存储控制器划分的多个LUN  
  （SVC称为MDisk）再次条带化，根据性能的高低划分成不同的磁盘池MDiskGroup (MDG)，再从磁盘池中切割VDisk分给应用服务器。对于应用服务器来说，它们可见的只是逻辑磁盘，也就是VDisk，后端的物理磁盘对其是不可见的。

每个MDisk都被划分为多个Extent，Extend类似于RAID组中的条块，其最小值是16MB，最大值是2GB，具体大小由MDG来定义。这些Extent按照每个MDisk从头到尾被顺序编号。  


1. 软件定义存储（Software Defined Storage） SDS  
    VMware对SDS定义如下：软件定义的存储产品是一个将硬件抽象化的解决方案，它使你可以轻松地将所有资源池化并通过一个友好的用户界面（UI）或API来提供给消费者。一个软件定义的存储的解决方案使得你可以在不增加任何工作量的情况下进行纵向扩展（Scale-Up）或横向扩展（Scale-Out）。  
    IDC对SDS定义如下：SDS将数据中心或者跨数据中心的各种存储资源抽象化、池化，以服务 的形式提供给应用，满足应用按需(如容量、性能、QoS、SLA等)自动化使用存储的需求。  
    SNIA认为: SDS需要满足：提供自助的服务接口，用于分配和管理虚 拟存储空间； SDS应该包括的功能：自动化、标准接口、虚拟数据路径、 扩展性、透明性。

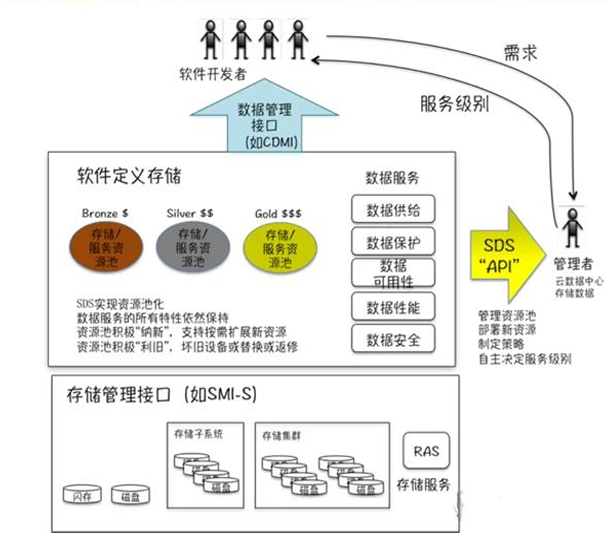
* 特性：

动态性 Responsive to workloads and changing requirements

自动化 Policy based provisioning and optimization

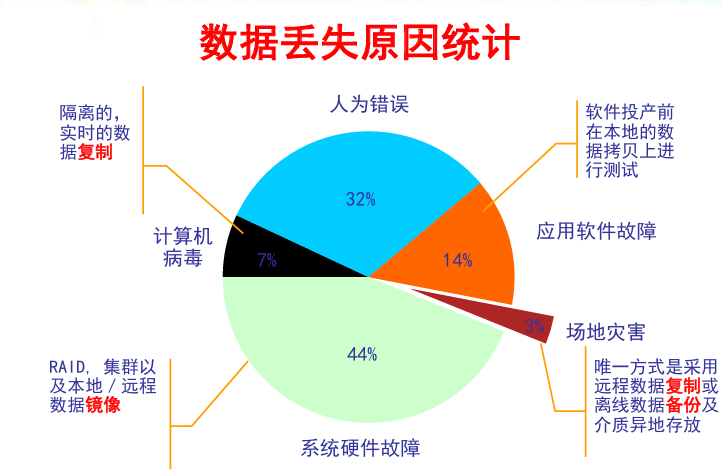
易扩展 Cost-optimized Scaling of Capacity & Throughput

* 存储虚拟化与SDS的区别： 前者把物理存储与逻辑视图分离，后者主要针对功能和服务与存储设备剥离，实现按需定制。
* RAID–First SW Defined Storage:AID软件创建特定的存储能力(高可用性,性能,成本)



1. 数据保护

* 数据丢失的原因：用户误操作；系统错误；电力故障；灾难事故；介质故障



* 数据保护的基本原则： 信息冗余  
  –设备级：RAID1、RAID3、 RAID4、RAID5 …  
  –系统级 ：纠删码 、备份（本地、远程）、快照、连续数据保护  
  –空间上 ：RAID1、RAID3、 RAID4、RAID5 … 、纠删码、远程镜像  
  –时间上 ：备份（本地、远程）、快照 、连续数据保护
* 名词术语  
  RPO-Recovery Point Objectives恢复点目标 指能把数据恢复到过去的那一个时间点   
  RTO-Recovery Time Objectives恢复时间目标 指在出现问题后“什么时候”可恢复数据

RPO定义了企业能够承受的数据丢失量，RTO定义了企业可以承受的停机时间

* 传统数据保护方案：备份与恢复： 低廉 费时、费力 安全性？
* 介质变革：磁带vs. 磁盘

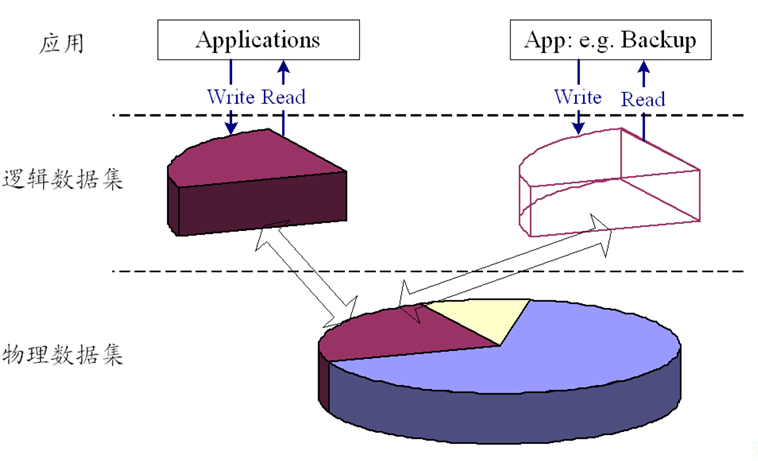
磁带：离线、顺序访问、容量大、速度慢、价格低

磁盘：在线、随机访问、容量↑、速度快↑、价格↓、更适宜构建系统

* 数据备份方法  
  热备份（Hot Backup）: 应用服务不中断  
  冷备份（Cold Backup）: 应用服务被中断.  
  一般通过备份代理管理备份不同类型的数据: 结构化数据 (如：Databases)、半结构 化数据 (如：Email)、非结构化数据 (file systems)

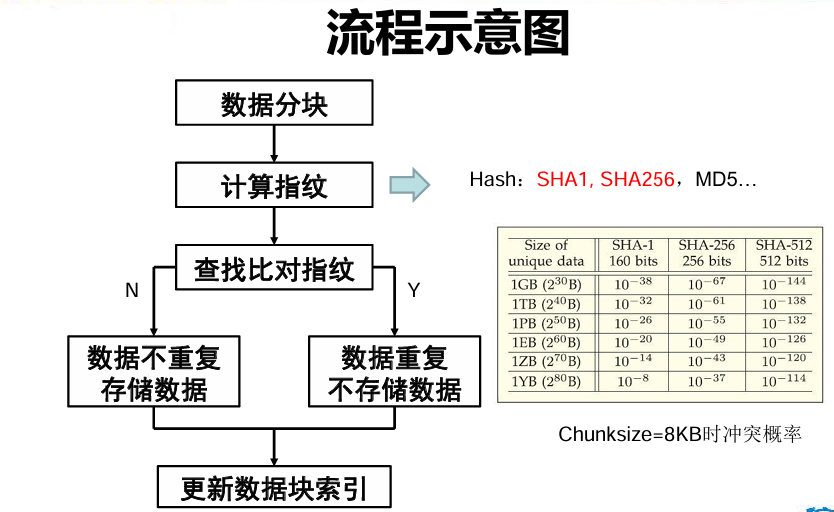
备份的分类和级别：全备份（Full Backup）、累积（差分）备份（Cumulative /Differential）、增量备份（Incremental）

* 一种常用的数据保护技术 快照(Snapshot)   
  ➢ 为什么“快照”：“快”：解决批量数据备份中的长时间窗口问题；“照”：至少在逻辑上 生成独立冗余数据，解决硬件 故障以外的本地数据保护问题；   
  ➢ 存储网络工业协会（SNIA）对快照的定义：  
   – 快照（snapshot）为一个数据对象产生完全可用的副本，它包含对该数据 对象 在某一时间点的映象，快照在快照时间点对数据对象进行逻辑复制操 作，产生数 据对象在该时间点的一致性数据副本，但实际的部分或全部物 理复制过程可能在 复制时间点之外的某些时间进行。  
   – 快照可以在非常短的时间内完成数据对象逻辑副本的创建，其数量级通常 为几 秒，因此也称为零时间复制。  
   – 快照技术可以最大限度减小复制操作对上层应用的影响，同时保证复制操 作语 义的原子性。  
  ➢ 分层与虚拟化：快照背后的思想：  
   “零时间复制产生一致性数据副本” ：独立的逻辑数据视图、关联的物理数据分布  
   “完全可用”的副本：–逻辑的副本具备可用性、一致



➢ 快照技术的主要策略  
 分离镜像  
 缺：空间效率低，可扩展性差，快照不适合写（写双/多份），快照创建时间 受限（同步过程中不能 创建新的快照）  
 优：安全性高，响应快，对系统性能影响小  
 写时拷贝 Copy on Write (CoW)   
 优：空间效率高， 快照可读写，可 扩展性好，易改 进  
 缺：CoW操作影响系统写性能  
 写重定向  
 优： 空间效率高，快照 可读写，响应快， 可扩展  
 缺： 物理数据交叠，映 射关系维护复杂

1. 数据备份中的去重  
   重复数据删除（Deduplication）   
   重删级别：File，Block；数据指纹：MD5,SHA-1, SHA-256, SHA-512；重删后的数据恢复
2. 重复数据删除  
   背景：冗余数据广泛存在；资源浪费，成本高，效率低；在数据备份系统中尤为明显

* 定义：重复数据删除是一种节约数据 存储空间的技术。在计算机中存储了很多重复数据， 这些数据占用了大量硬盘空间，利用重复数据删除技 术，可以只存储一份数据。  
  在存储数据时检查和比较已存在的 数据，如果它们是相同的，那么就 过滤掉这部分数据的存储，然后通 过指针引用已存在的数据，达到消 除冗余数据的目的。
* 意义  
  节省存储空间 – 删除重复数据，大大降低所需存储介质空间大小  
  降低成本 – 存储介质数量减少，存储数据所需成本降低   
  节省网络带宽 – 在客户端去重，减少不必要的数据传输，提升了传输效率  
  提升写入性能 – 磁盘写入性能有限，写入数据时删除重复数据，减少不必要的写入  
   – 特别地，对于SSD而言，减少写入可提升使用寿命
* 去重方法分类  
  去重端点 – 源端去重source based；目标端去重（target based）   
  去重时机 – 在线（inline）；后续/离线（postprocess/offline）   
  去重层次/粒度 – 文件级（file level）；块级（chunk level）   
   分块算法– 定长分块（fixed length block）；变长分块（variable length block）
* 去重端点  
  源端去重：节省网络带宽，去重后传输到target仍可能存在重复数据；目标端去重：提高存储利用率,去重复数据都在目标端，使得管理集中，能够进行全局的数据去重复，可称为一个相对独立的系统
* 去重时间  
  在线去重：数据写入一次处理一次，存在延迟。离线去重：数据先写入后处理再写入；在用户数据上传的过程中，数据去重复并不会发生，而是直接写到存储设备上。当用户数据上传完全结束后，再进行相关的数据去重复。
* 重删的基本步骤（块级）  
  1. 数据分块（Chunking） 2. 计算块标志值（Fingerprinting，指纹) 3. 比较数据块标志值，消除重复数据块，建立索引(Indexing)   
   
* 数据去重存储系统面临的挑战：分块算法计算开销；数据块指纹索引瓶颈；数据块碎片造成数据恢复效率问题 【分块哈希→查找索引→更新恢复】
* 两种数据去重分块算法  
  1、固定分块长度去重   
  优点：定长文件的切片实现和管理比较简单 ；缺点：数据去重复的比率比较低   
  2、可变分块长度去重 （基于内容的分块, CDC）   
  优点：去重率高 ；缺点：实现复杂，计算开销大
* 数据去重索引瓶颈问题  
  磁盘索引瓶颈问题：Venti使用了磁盘指纹索引方案，其系统冗余数据消除吞吐率仅仅为6.5MB/s；高吞吐率是企业数据保护的关键指标，其备份任务通常需要超过100MB/s → 局部性与相似性结合的索引算法SiLo
* 局部性与相似性结合的索引算法SiLo

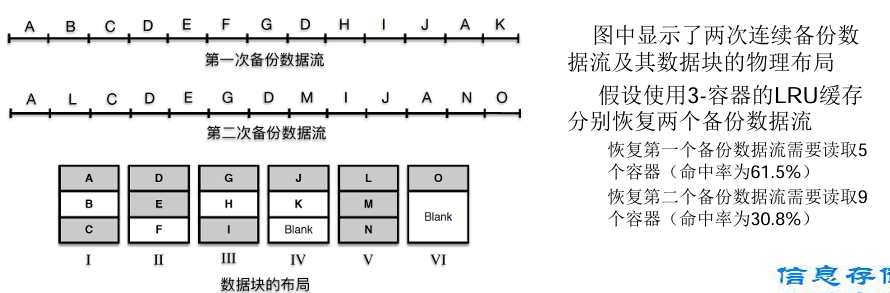
目前的主流解决方案：（挖掘局部性 / 挖掘相似性）  
 

启发：联合挖掘相似性和局部性  
 ➢ 通过对相似性的挖掘，避免了全局遍历索引

➢ 通过对局部性的挖掘，补充相似性查找效果

局部性算法加强联合挖掘相似性与局部性来优化数据去重索引性能   
 SiLo与其他索引算法比较：  
 挖掘备份数据流的局部性与相似性：➢ 使系统达到了近似精确的数据去重效果， 获得很高的去重索引吞吐率，同时仅仅消 耗了很少的内存开销。 ➢ 与Extreme Binning系统和ChunkStash索 引方法比较，在保证去重效果的前提下， SiLo仅 分别消耗它们1/3∼1/90和1/41∼1/60 的索引内存开销。

* 去重数据块碎片问题  
  数据去重通过消除文件间的重复数据块节约存储空间：去重后的文件不再连续存储，造成数据块碎片；去重后的文件读性能下降；数据块碎片导致无法保证数据恢复时间



数据备份的目的是为了数据恢复。  
当遇到数据灾难时，数据恢复的时间决定了业务中断的时间。  
存储成本和恢复性能是否可以取得平衡：不显著增加存储成本；保证稳定的恢复性能 延迟去重技术：基于缓冲区的重写算法：iDedup，CFL-SD，CBR，Capping   
历史感知重写算法HAR

* 基于历史感知重写算法HAR  
  ➢ 数据块碎片示例：观察：稀疏容器的遗传性。随着备份版本的增加，稀疏容器的总数（新增稀疏容器和继承稀疏容器之和）在不断增多；每次备份的新增稀疏容器相对较少；每次备份的继承稀疏容器数量约等于上一次备份的稀疏容器总数  
  ➢ 稀疏容器：容器利用率较低（预取机制失效），例子：容器IV，可能的解决方法：数据块迁移（开销很大）。乱序容器：容器内数据块不是连续被访问（时间局部性较弱），例子：容器V，可能的解决方法：缓存（开销较小）   
  ➢ HAR的基本思想：本次备份的过程中：统计容器的利用率；本次备份结束后：记录稀疏容器的ID；下次备份的过程中：重写继承的稀疏容器中的数据块  
  ➢ 本次备份的新增稀疏容器成为下次备份的继承稀疏容器  
  ➢ 优势：基于第二个观察（每次备份的新增稀疏容器相对较少），每次备份需要处理 的继承稀疏容器很少；基于第三个观察（继承性），每次备份都能准确识别稀疏容器