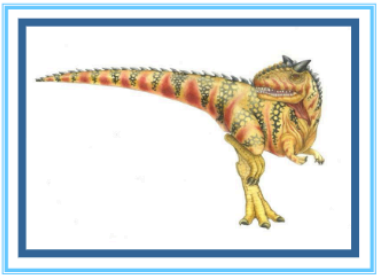
第 11 章：大容量存储系统



第 11 章：大容量存储系统

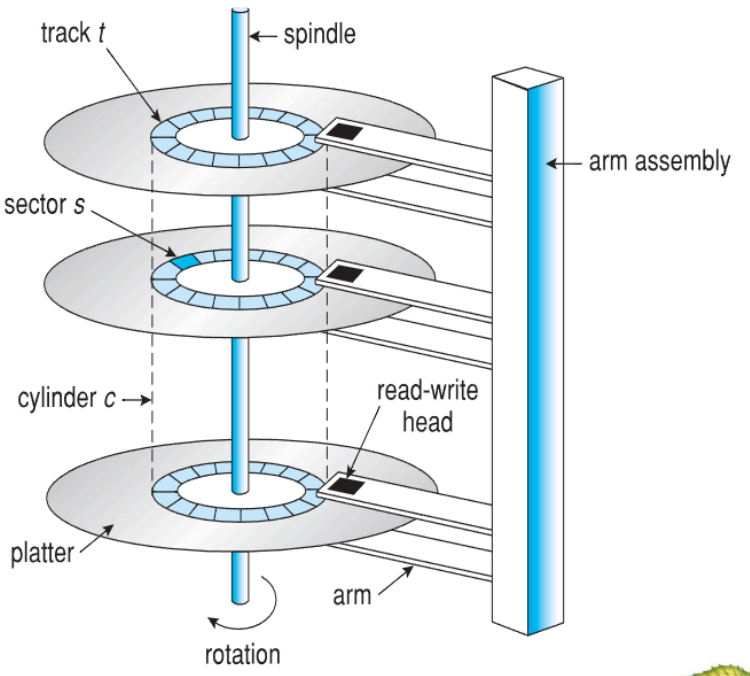
- 大容量存储结构概述 磁盘结构 磁盘调度 RAID 结构

目标

- 描述辅助存储设备的物理结构以及设备结构对其用途的影响
- 解释大容量存储设备的性能特征评估 I/O 调度算法
-
- 讨论为大容量存储提供的操作系统服务，包括 RAID

动头盘机构

- 每个圆盘盘片都有一个扁平的圆形，直径分别为 1.8 英寸、2.5 至 3.5 英寸
- 盘片的两个表面覆盖有用于存储信息的磁性材料
- 读写头“飞”在每个盘片的每个表面上方
- 磁头连接到圆盘臂上，圆盘臂将所有磁头作为一个整体移动
- 盘片表面在逻辑上被划分为圆形轨道，这些轨道又细分为数百个扇区（每个轨道）
- 位于一个臂位置的一组轨道构成一个圆柱体 – 磁盘驱动器中的数千个同心圆柱体
- 一个磁盘驱动器中可能有数千个同心圆柱体



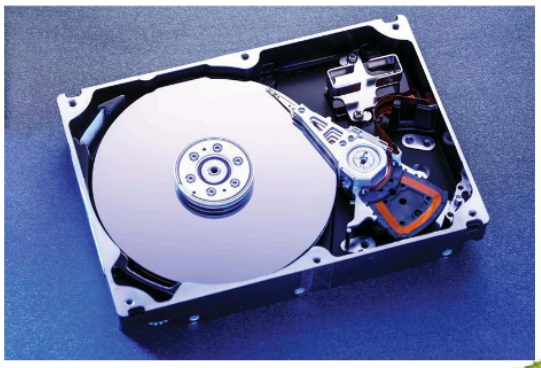
大容量存储结构概述

- 磁盘提供现代计算机的大量辅助存储
  - 驱动器以每秒 60 到 250 次的速度旋转 传输速率是驱动器和计算机之间数据流的速率 定位时间（随机访问时间）是 （1） 将磁盘臂移动到所需的柱面的时间（寻道时间）和 （2） 所需扇区在磁盘磁头下旋转的时间（旋转延迟） 磁头崩溃是由于磁盘磁头与磁盘表面接触而导致的 - 这很糟糕！
- 磁盘是可移动的 驱动器通过 I/O 总线连接到计算机
  - 总线各不相同，包括 EIDE、ATA、SATA、USB、光纤通道、SCSI、SAS、Firewire计算机中的主机控制器使用总线与驱动器或存储阵列中内置的磁盘控制器通信

硬盘驱动器

- 盘片的范围从 .85 英寸到 14 英寸（历史上）
  - 通常为 3.5 英寸、2.5 英寸和 1.8 英寸
- 每个驱动器的性能范围从 30GB 到 3TB 不等
  - 传输速率 – 理论 – 6 Gb/秒 有效传输速率 – 实际 – 1Gb/秒 寻道时间从 3ms 到 12ms – 9ms 通常用于台式机驱动器
  - 根据 1/3 的磁道测量或计算的平均寻道时间
  - RPM 通常为每分钟 5,400、7,200、10,000 和 15,000 转
  - 基于主轴速度的延迟
    - 1/（RPM/60） = 60/RPM 平均延迟 = 1/2 延迟 例如，对于 7200 rpm，即 120 rps，平均延迟为 1/240 = 4.17 微秒

Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2



硬盘性能

- 访问延迟或平均访问时间 = 平均寻道时间 + 平均延迟
  - 对于快速磁盘 3ms + 2ms = 5ms 对于慢速磁盘 9ms + 5.56ms = 14.56ms
  -
- 平均 I/O 时间 = 平均访问时间 +（传输量 / 传输速率）+ 控制器开销
- 例如，要在 7200 RPM 磁盘上传输一个 4KB 数据块，平均寻道时间为 5 毫秒，则 1Gb/s 传输速率和 0.1 毫秒的控制器开销 =
  - 5 毫秒 + 4.17 毫秒 + 4KB / 1Gb/秒 + 0.1 毫秒 =

    4KB = 4 * 2^10 * 8 = 32 * 2^10 比特
    1Gbps = 1 * 2^30 比特/秒
  - 9.27 毫秒 + 32 / 2^20 秒 = 9.27 毫秒 + 0.0305 毫秒 ≈ 9.3 毫秒因此，传输 4KB 平均需要 9.3 毫秒，因此有效带宽仅为 4KB/9.3 毫秒≈仅 3.5 Mb/秒（考虑到开销，传输速率为 1 Gb/秒）。
  -
  -

硬盘性能（续）

- 随机工作负载和顺序工作负载之间的硬盘驱动器性能存在巨大差距 考虑一个容量为 300GB 的磁盘，平均寻道时间为 4 毫秒（4 毫秒），RPM 为 15,000 RPM 或 250 RPS（平均轮换时间为 2 毫秒），传输速率为 125MB/s，
- 并且在随机位置发生 4KB 读取， 召回

平均访问时间 = 平均寻道时间 + 平均延迟 平均 I/O 时间 = 平均访问时间 +（传输量/传输速率）+ 控制器开销（忽略）= 4 毫秒 + 2 毫秒 + 30 微秒

-
- 有效带宽或传输速率为 4KB/6ms = 0.66MB/s
- 对于一个 100 MB 文件的顺序访问，假设只有一个寻道和轮换，这将产生接近 125MB/s 的有效带宽或传输速率

固态硬盘 （SSD）

- SSD 是非易失性存储器 （NVM），使用方式与硬盘驱动器类似许多技术变体，例如，从带有电池的 DRAM 以在电源故障中保持其状态，到单级单元 （SLC） 和多级单元 （MLC） 芯片等闪存技术
  - SSD 比 HDD 更可靠，因为它们没有移动（机械）部件，它们的速度要快得多，因为它们没有寻道时间或旋转延迟。
  - 它们消耗更少的功率 – 能效但它们每 MB 更昂贵，容量更小，并且使用寿命可能更短
- 因为它们比磁盘驱动器快得多，所以标准总线接口可能太慢，从而导致吞吐量的重大限制
  - 有些直接连接到系统总线（例如 PCI），有些将它们用作新的缓存层，在磁盘、SSD 和内存之间移动数据以优化性能

磁带

- 磁带是一种早期的辅助存储介质
  - 它是相对永久的，可以保存大量数据 访问时间很慢，因为移动到磁带上的正确位置可能需要几分钟 随机访问比磁盘慢 ~1000 倍，因此它们对于现代计算机系统中的辅助存储不是很有用
- 主要用于备份、存储不常用数据，或作为将信息从一个系统传输到另一个系统的媒介
- 磁带容量差异很大，具体取决于特定类型的磁带驱动器，当前容量超过数 TB，通常在 200GB 到 1.5TB 之间

磁盘结构

- 磁盘驱动器被寻址为大型的一维逻辑块数组，其中逻辑块是最小的传输单位。换句话说，磁盘由多个磁盘块表示，每个块都有一个唯一的块号 – 磁盘地址
  - 逻辑块的大小通常为 512 字节低级格式化在物理介质上创建逻辑块
  -
- 逻辑块的一维数组按顺序映射到磁盘的扇区：
  - Sector 0 （扇区 0） 是最外侧圆柱体上第一个轨迹的第一个扇区映射按顺序通过该轨迹，然后是该圆柱体中的其余轨迹，然后从最外层到最内层穿过其余圆柱体
  -
  - 逻辑到物理地址（由柱面编号、柱面内的磁道编号和带有磁道的扇区号组成）应该很容易，但
    - 有缺陷的扇区，映射通过替换磁盘上其他位置的备用扇区来隐藏这一点 在某些设备上，每个磁道的扇区数可能不是恒定的。非常量 #
      - 通过恒定角速度的扇区数

磁盘调度

- 操作系统负责有效地使用硬件 – 对于磁盘驱动器，这意味着具有快速的访问时间和较大的磁盘带宽
- 寻道时间是磁盘磁头臂将磁头移动到包含所需扇区的相应柱面的时间，这可以通过寻道距离（以柱面/磁道的数量表示）来衡量。
- 旋转延迟是磁盘将所需扇区旋转到磁盘磁头的额外时间。
- 磁盘带宽 是传输的总字节数除以第一次服务请求和最后一次传输完成之间的总时间。
- 我们可以通过管理磁盘 I/O 请求的服务顺序来改善访问时间和带宽。

磁盘调度（续）

- 磁盘 I/O 请求的来源有很多，包括操作系统、系统进程和用户进程
  - I/O 请求包括输入/输出模式、磁盘地址、内存地址、要传输的扇区数
- 操作系统为每个磁盘或设备维护一个请求队列
  - 在具有许多进程的 multiprogramming 系统中，磁盘队列通常有几个待处理的请求。
- 空闲磁盘可以立即处理 I/O 请求，繁忙磁盘意味着请求必须排队。

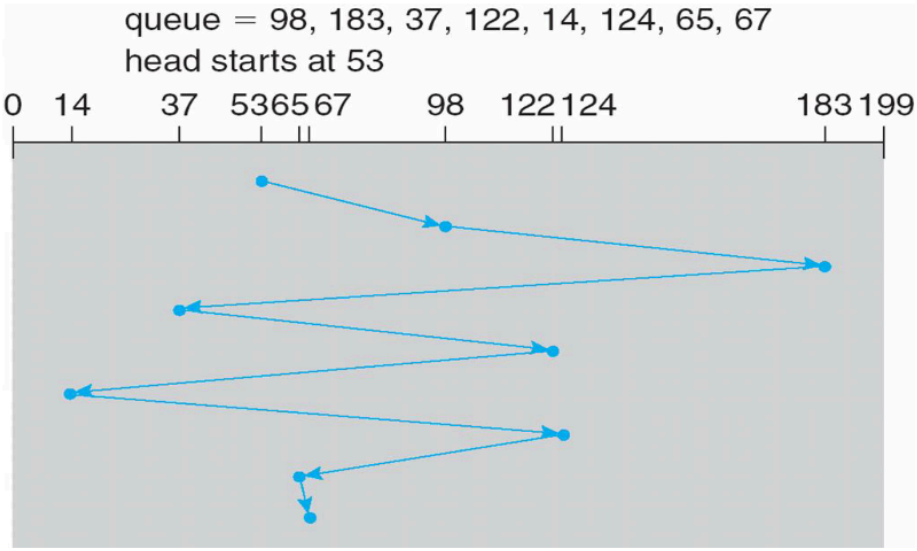
  仅当存在 I/O 请求队列时，优化才有意义。磁盘驱动器控制器具有较小的缓冲区，并管理 I/O 请求队列（具有不同的“深度”）。
  - 当一个请求完成后，接下来选择为哪个待处理请求提供服务？磁盘调度
- 接下来，我们用请求队列 （0-199） 来说明调度算法，0-199 是柱面编号。

  98, 183, 37, 122, 14, 124, 65, 67

  当前 Head 职位为 53

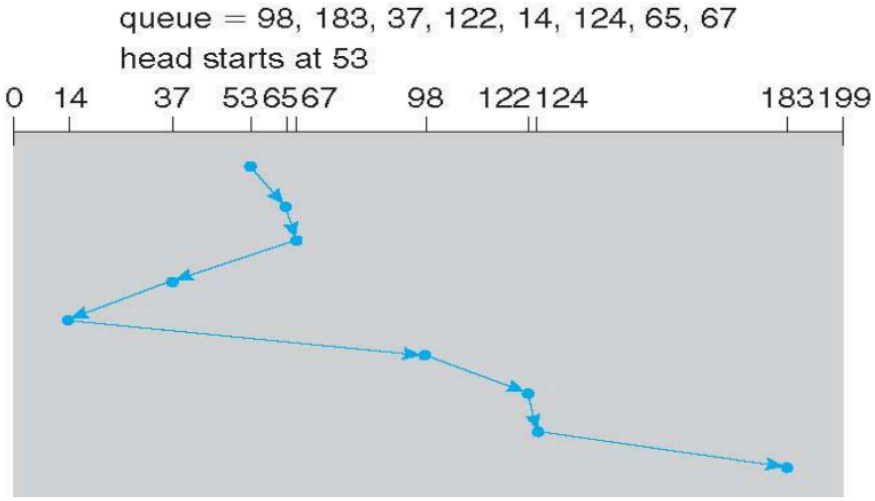
FCFS 先到先得

- FCFS 本质上是公平的，但它通常不提供最快的服务插图显示了 640 个气缸的总扬程运动



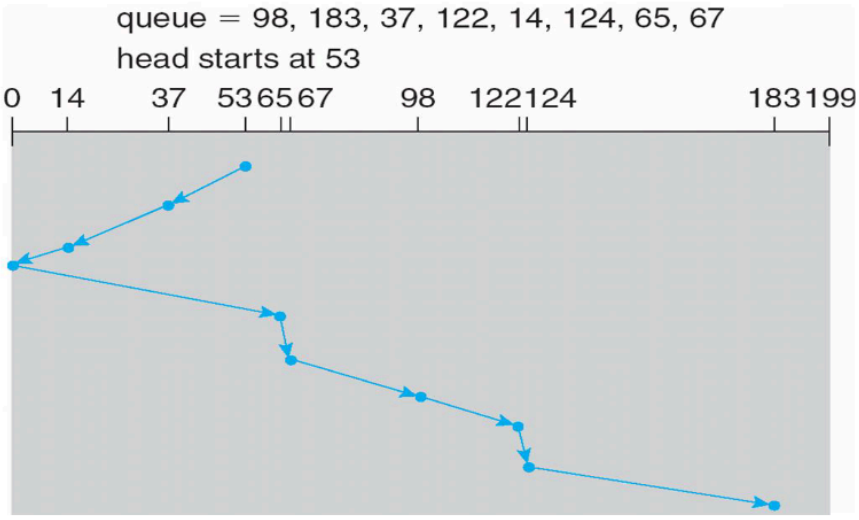
SSTF 最短搜索时间优先

- 最短搜索时间优先 （SSTF） 选择从当前头部位置开始具有最短搜索时间的请求，即选择最接近当前头部位置（任一方向）的待处理请求
- SSTF 调度是 SJF 调度（贪婪算法）的一种形式;可能会导致某些请求匮乏，因为请求可能随时动态到达
- 图示为 236 个气缸的总扬程移动



SCAN 调度

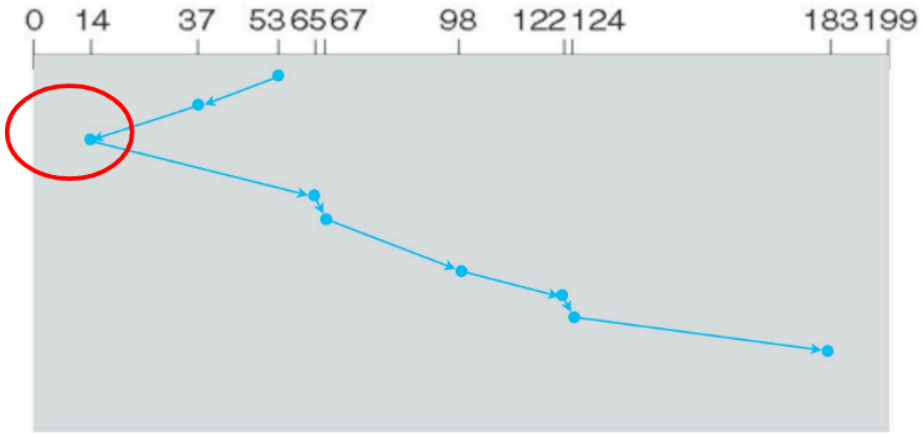
- 磁盘臂从磁盘的一端开始，然后向另一端移动，在到达每个柱面时为请求提供服务，直到它到达磁盘的另一端。在另一端，头部移动的方向相反，维修继续进行。这有时称为电梯算法。请注意，如果请求在柱面之间均匀分布，则请求密度最高的是磁盘的另一端，并且等待时间最长。此外，我们需要知道头部运动的方向。图示为 236 个气缸的总扬程移动



LOOK 调度

- LOOK Scheduling：与 SCAN 调度类似，但磁盘臂仅到达每个方向的最终请求（而不是磁盘的末端）
- 在以下示例中，总磁头移动为 208 个圆柱体

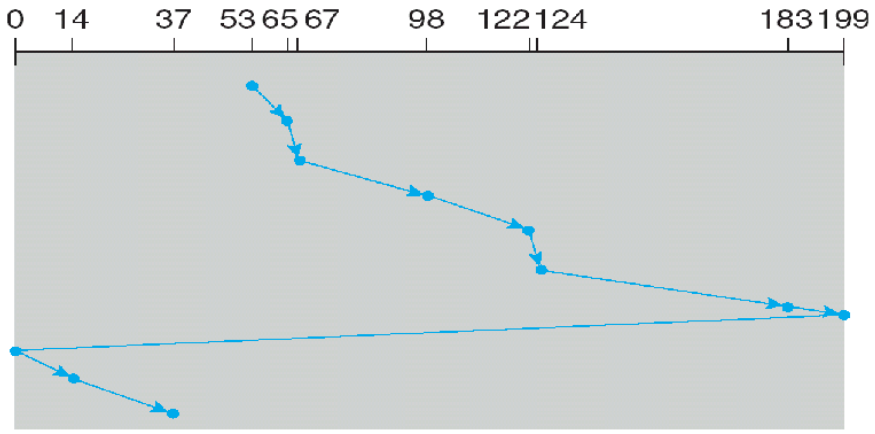
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



C-扫描

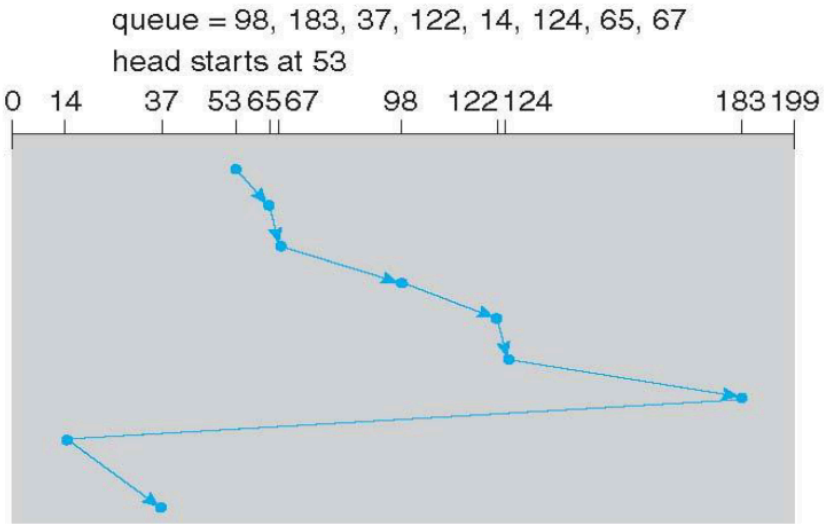
- C-SCAN，Circular-SCAN，SCAN 的一种变体，提供比 SCAN 更均匀的等待时间磁头从磁盘的一端移动到另一端，在移动过程中为请求提供服务。但是，当它到达另一端时，它会立即返回到磁盘的开头，而不在回程中处理任何请求
- 将圆柱体视为一个循环列表，该列表从最后一个圆柱体环绕到第一个圆柱体。圆柱体总数 - 382

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



C-LOOK外观

- LOOK 是 SCAN 的一个版本，C-LOOK 是 C-SCAN 磁盘臂的一个版本，每个方向的最后一个请求都只到最后一个请求，然后立即反转方向，而不是先一直到磁盘的末端
- 气缸总数？ – 322（用于 C-LOOK）和 308 用于 LOOK



选择磁盘调度算法

- SSTF 很常见，并且具有天然的吸引力，因为它比 FCFS 提高了性能。SCAN 和 C-SCAN 对于在磁盘上放置重负载的系统性能更好，因为它们不太可能导致匮乏问题。
- 调度性能还取决于请求的数量和类型。如果只有一个请求，则所有调度算法的行为都应该相同（如 FCFS 调度）
- 对磁盘服务的请求受文件分配方法的很大影响（有待讨论）
  - 连续分配的文件将在磁盘上生成多个请求，从而导致磁头移动受限，而链接或索引文件可能包含广泛分散在磁盘上的块，从而导致磁头移动更大。
- 目录和索引块的位置也很重要，它们经常被访问。
  - 不同柱面上的目录条目和文件数据导致头部移动过多在内存中缓存目录和索引块 帮助
- disk-scheduling 算法应作为 os 的单独模块编写，以便在必要时将其替换为其他算法。SSTF 或 LOOK 作为默认算法是合理的选择。

RAID – 通过冗余提高可靠性

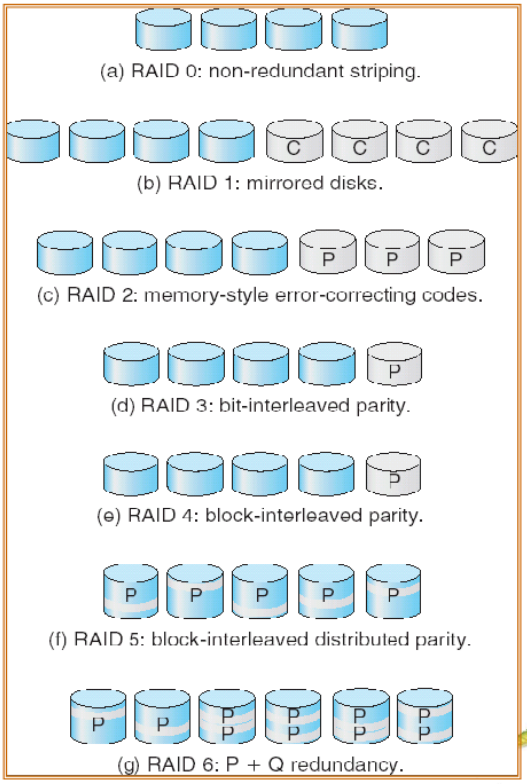
- RAID – 独立磁盘的冗余阵列
  - 过去，由小型、廉价磁盘组成的 RAID 被视为大型、昂贵磁盘（曾经称为廉价磁盘冗余阵列）的经济高效替代方案
  - 现在，RAID 通过冗余和更高的数据传输速率（并行访问）实现更高的可靠性
- 增加平均故障时间 N 个磁盘中某个磁盘发生故障的几率远高于特定单个磁盘发生故障的几率。假设单个磁盘的平均故障时间为 100,000 小时，则 100 个磁盘阵列中某个磁盘的平均故障时间为 100,000/100 = 1,000 小时，即 41.66 天！
- 如果我们只存储一个数据副本，则数据丢失率是不可接受的
  - 解决方案是引入冗余;最简单（但最昂贵）的方法是复制每个磁盘，称为镜像。每次写入都在两个物理磁盘上执行。只有在更换第一个故障磁盘之前第二个磁盘发生故障时，数据才会丢失。
- 平均修复时间是更换故障磁盘并恢复其上数据所需的时间 - 其他故障可能导致数据丢失的暴露时间
  - 假设单个磁盘的平均故障时间为 100,000 小时，平均修复时间为 10 小时。数据丢失的平均时间为 100， 0002 / （2 * 10） = 500 * 106 小时，即 57,000 年！

RAID – 通过并行性提高性能

- 通过数据条带化实现磁盘系统中的并行性有两个主要目标： 通过负载平衡增加多个小型访问的吞吐量 减少大型访问的响应时间
- 位级条带化
  - 例如，如果我们有一个包含 8 个磁盘的数组，我们可以将每个字节的位 i 写入磁盘 i。可以将 8 个磁盘的数组视为单个磁盘，其扇区是正常扇区大小的 8 倍。访问率可以提高 8 倍！
  - 位级条带化可以概括为包括 8 的倍数或除以 8 的磁盘数。例如，对于一个 4 个磁盘的数组，每个字节的 bit i 和 4+i 存储在磁盘 i 中
- 块级条带化
  - 文件块在多个磁盘上条带化使用 n 个磁盘时，文件的块 i 进入磁盘 （i mod n） +i

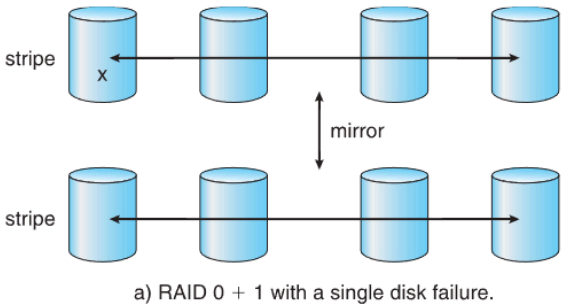
RAID 结构

- 镜像提供高可靠性，但价格昂贵条带化提供高数据传输速率，但不提供可靠性
-
- 许多方案通过使用条带化和“奇偶校验”位（通常分为 6 个 RAID 级别）来以较低的成本提供冗余
- 在 RAID 级别中，存储了 4 个磁盘的数据。P 表示纠错位，C 表示数据的第二个副本
  - RAID 0 是指在非冗余的块级别具有条带化的磁盘阵列
  - 镜像或重影 （RAID 1） 保留每个磁盘的副本
  - 条带化镜像 （RAID 1+0） 或镜像条带化 （RAID 0+1） 提供高性能和高可靠性
  - 块交错奇偶校验（RAID 4、5、6）使用的冗余要少得多

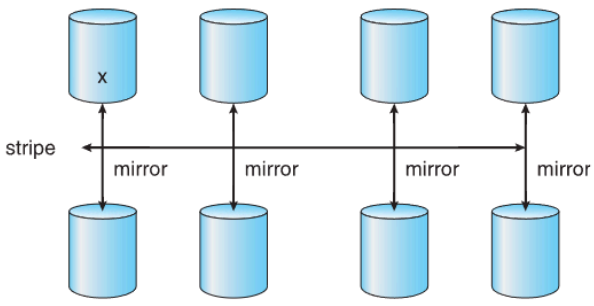


RAID （0 + 1） 和 （1 + 0）

- 条带镜像 （RAID 1+0） 或镜像条带 （RAID 0+1） 都提供高性能 （RAID 0） 和高可靠性 （RAID 1）。（RAID 0+1） 一组驱动器被条带化，然后将条带镜像到另一个等效的条带
-
- （RAID 1+0） 驱动器成对镜像，然后对生成的镜像对进行条带化



a) RAID 0 + 1 with a single disk failure.



b) RAID 1 + 0 with a single disk failure.

其他功能

- 无论实施了什么 RAID，都可以在每个级别添加其他有用的功能
- 快照是上次更新发生之前的文件系统视图（用于恢复）
- 复制是在单独的站点之间自动复制写入，以实现冗余和灾难恢复。这可以是同步的（在认为写入完成之前，每个块都必须在本地和远程写入）或异步的（写入分组在一起并定期写入）
- 热备用磁盘不用于数据，但配置为在磁盘发生故障时用作替换
  - 例如，如果镜像对中的一个磁盘发生故障，可以使用热备件来重建镜像对。通过这种方式，可以自动重新建立 RAID 级别，而无需等待更换/修复故障磁盘。

第十一章结束