# 柱面-磁头-扇区寻址的一些旧事

目录

目录

- 柱面、磁头、扇区以及相关术语
- 物理 CHS 寻址
- 逻辑 CHS 寻址
- 区位记录（Zone bit recoding, ZBR）
- 从 CHS 到 LBA
- 叠瓦磁记录（Shingled Magnetic Recording,

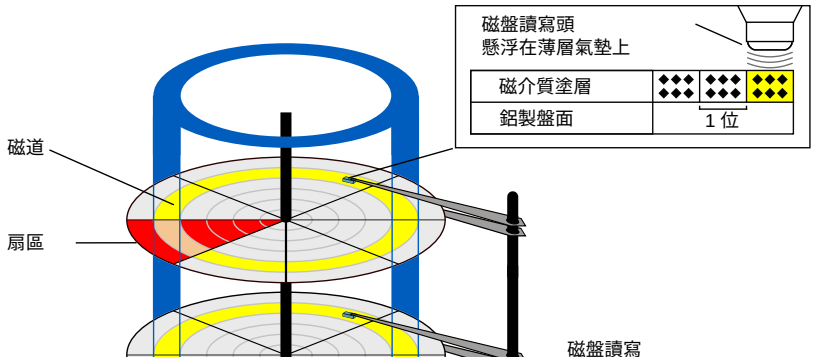
SMR）
- 4KiB 扇区大小
- 结论（TL;DR）和预告

在 SSD 这种新兴存储设备普及之前，很长一段时间硬盘是个人计算机的主要存储设备。 更往前的磁带机不常见于个人计算机，软盘的地位很快被硬盘取代，到 SSD 出现为止像 MiniDisc 、 DVD-RAM 等存储设备也从未能挑战过硬盘的地位。硬盘作为主要存储设备，自然也影响了文件系统的设计。

这篇笔记稍微聊一聊硬盘这种存储设备的寻址方式对早期文件系统设计的一些影响，特别是 柱面-磁头-扇区寻址（Cylinder-head-sector addressing, 简称CHS寻址）的起源和发展。 大部分内容来自维基百科 Cylinder-head-sector 词条 这里只是记录笔记。现今的硬盘已经不再采用 CHS 寻址，其影响却还能在一些文件系统设计中看到影子。

# 柱面、磁头、扇区以及相关术语

磁盘示意图（来自维基百科 Cylinder-head-sector 词条 ）



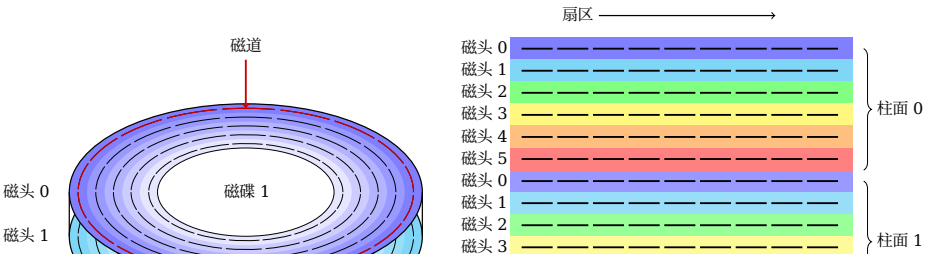

如右图所示，一块硬盘(Hard Disk Drive, HDD)是一个圆柱体转轴上套着一些磁碟片(plate)， 然后有一条磁头臂(actuator arm)插入磁碟片间的位置，加上一组控制芯片（controller）。 每个磁碟片有上下两面涂有磁性材质，磁头臂上有一组磁头（head），每个磁头对应磁盘的一个面， 所以比如一个 3 碟的硬盘会有 6 个磁头。

每个磁碟片上定义了很多同心圆的磁头轨道，叫做磁道（track），磁道位于盘面上不同半径的位置， 通过旋转磁碟臂能让磁头移动到特定的半径上，从而让读写磁头在不同的磁道间跳转。 不同磁头上同磁道的同心圆共同组成一个柱面（cylinder），或者说移动磁碟臂能选定磁盘中的一个柱面。 磁道上按等角度切分成多个小段，叫做扇区（sector），每个扇区是读写数据时采用

的最小单元。 早期在 IBM 大型机之类上使用的硬盘的扇区大小比较小，到 IBM PC 开始个人计算机用的硬盘扇区基本被统一到 512 字节。现代硬盘内部可能采用 Advanced Format 使用 4K 字节扇区。

在早期软盘和硬盘的寻址方式被称作「柱面-磁头-扇区寻址」，简称 CHS 寻址， 是因为这三个参数是软件交给硬件定位到某个具体扇区单元时使用的参数。 首先柱面参数让磁头臂移动到某个半径上，寻址到某个柱面，然后激活某个磁头，然后随着盘面旋转， 磁头定位到某个扇区上。

「柱面-磁头-扇区」这个寻址方式，听起来可能不太符合直觉，尤其是柱面的概念。直觉上， 可能更合理的寻址方式是「盘片-盘面-磁道-扇区」，而柱面在这里是同磁道不同盘片盘面构成的一个集合。 不过理解了磁盘的机械结构的话，柱面的概念就比较合理了，寻址时先驱动磁头臂旋转， 磁头臂上多个磁头一起飞到某个磁道上，从而运动磁头臂的动作定义了一个柱面。 柱面和磁头（CH）组合起来能定位到某个特定的磁道，画张图大概如下图所示：




上图中值得注意的是磁道的编号方式，我用相同的颜色画出了相同的磁道。因为按照 CHS 的顺序寻址，所以先定位柱面，然后选定磁头。磁盘上按半径从外向内定义柱面的编号，最外圈的磁道位于 0号柱面，由0号磁头开始。随着柱面编号增加，逐步从外圈定位到内圈。

# 物理 CHS 寻址

以上术语中，柱面号和磁头号直接对应了硬盘上的物理组成部分，所以通过在物理 CHS 寻址方式下，通过扇区地址的写法能对应到扇区的具体物理位置。之所以

这样描述扇区， 是因为早期的软盘和硬盘驱动器没有内置的控制芯片，可以完全由宿主系统执行驱动程序驱动。

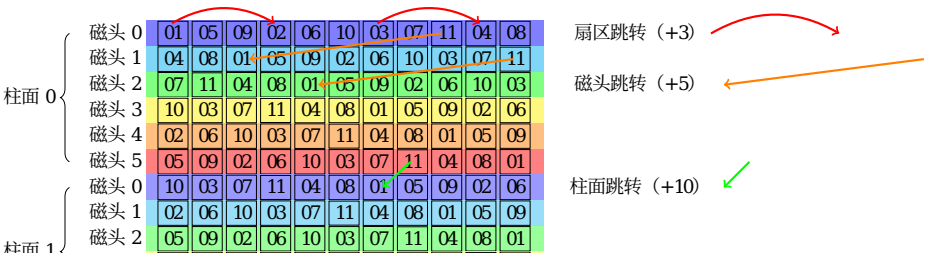
在 IBM PC 上，驱动软盘和硬盘的是 CPU 执行位于主板 BIOS (Basic Input/Output System) 中的程序，具体来说操作系统（比如DOS）和应用程序调用 INT 13H 中断，通过 AH=02H/03H 选择读/写操作，BIOS 在中断表中注册的 13H 中断处理程序执行在 CPU 上完成读写请求。调用 INT 13H 读写扇区的时候，CPU 先通过 INT 13H AH=0CH 控制硬盘的磁头臂旋转到特定磁道上，然后选定具体磁头，让磁头保持在磁道上读数据， 通过忙轮训的方式等待要读写的扇区旋转到磁头下方，从而读到所需扇区的数据。在 DOS 之后的操作系统， 比如早期的 Windows 和 Linux 和 BSD 能以覆盖中断程序入口表的方式提供升级版本的这些操作替代 BIOS 的程序。

以上过程中可以看出两点观察：

1. CHS 寻址下，跨磁道的寻址（不同 CH 值），和磁道内的寻址（同 CH 不同 S ），是本质上不同的操作。跨磁道的寻址有移动磁头臂的动作，会比磁道内寻址花费更多时间。
2. 通过扇区号的磁道内寻址是个忙轮训操作，需要占用完整 CPU 周期。这也隐含扇区号在一个磁道内的物理排列不必是连续的。

实际上扇区号的物理排列的确不是连续的，每个物理扇区中除了用512字节记录扇区本身的数据， 还有扇区的开始记录和结束记录，写有扇区编号和扇区校验

码。每读到一个扇区， CPU 可能需要做一些额外操作（比如计算比对校验、写入内存缓冲区、调整内存段页映射） 后才能继续读下一个扇区，如果物理排列上连续编号扇区，可能等 CPU 做完这些事情后磁头已经旋转到之后几个扇区上了。所以出厂时做磁盘低级格式化的时候， 会跳跃着给扇区编号，给 CPU 留足处理时间。比如下图：




上图中假设有3个柱面，每个柱面6个磁头，每个磁道内11个扇区，并且画出了三种不同的扇区编号跳转情况， 分别是磁道内的扇区跳转（+3），柱面内的磁头跳转（+5），以及柱面间跳转（+10）。 实际磁盘上的柱面数、扇区数要多很多，寻址时需要跳转的距离也可能更长，这里只是举例说明。 图中和实际情况相同的是，柱面号和磁头号从 0 开始编号，而扇区号从 1 开始编号， 所以做逻辑地址换算的时候要考虑编号差异。

早期 IBM PC 的 BIOS 使用 24bit 的 CHS 地址，其中 10bit 柱面(C)、 8bit 磁头(H)、 6bit 扇区(S)。从而用物理 CHS 寻址方式的软盘和硬盘驱动器最多可以寻址 1024

个柱面，256 个磁头， 63 个扇区，其中扇区数因为从 1 开始编号所以少了 1 个可寻址范围。比如 3.5 吋高密度（HD）软盘有双面， 出厂时每面 80 磁道，每磁道 18 扇区，从而能算出 1,474,560 字节的容量。

如此跳跃编号扇区之后，不是总能给磁道中所有扇区编号，可能在磁道的末尾位置留几个没有使用的扇区空间， 这些是磁道内的保留扇区，可以在发现坏扇区后使用这些隐藏扇区作为替代扇区。当然读写替代扇区的时候 因为扇区寻址不连续可能会有一定性能损失。

因为物理 CHS 寻址下，磁盘由 CPU 执行驱动程序来驱动，所以以上扇区跳跃的长短实际是由 CPU 的速度等因素决定的，理论上 CPU 越快，跳跃间隔可以越短，从而磁盘读写速度也能加快。磁盘出厂时， 厂商并不知道使用磁盘的计算机会是怎样的性能，所以只能保守地根据最慢的 CPU 比如 IBM 初代 PC 搭配的 8086 的速度来决定跳跃间隔。所以在当年早期玩家们流传着这样一个操作：买到新硬盘， 或者升级了电脑配置之后，对硬盘做一次 低级格式化(Low level formating) ，聪明的低级格式化程序能智能安排扇区编号，提升硬盘读写速度，也能跳过已知坏道位置继续编号， 甚至可能将更多保留扇区暴露成可用扇区。这对现代有硬盘控制器的硬盘而言已经没有意义了。

# 逻辑 CHS 寻址

随着硬盘容量不断增加， BIOS 中用来 CHS 寻址的地址空间逐渐不够用了。早期 24bit 地址按 *C H S* 的顺序分为 *10 8 6* 的位数，用 8bit 来寻址磁头最多可以有 256 个磁头，而只有 10bit 来寻址柱面，就只能有 1024 个柱面。最初 IBM 这么划分是因为早期用于 IBM 大型机之类的硬盘可以有 厚厚一叠的盘片组，同样的寻址方式就直接用于了 IBM PC 。而 PC 用的硬盘迫于硬盘仓空间大小， 有厚度限制，硬盘中物理盘面可能只有四五个盘片，硬盘容量增加主要是增加盘片表面的数据密度而非增加盘片数量。

于是逐渐地，硬盘厂商开始对 CHS 寻址的地址空间做一些手脚。比如最初的简单想法是重新定义 CH ，将一些磁头数挪用做柱面数。从而有了逻辑 CHS 寻址，其中 CH 是固定一组，通过简单换算从 CH 值找到物理的柱面和磁头数。结合 CH 而不映射 S 的优势在于，从操作系统和文件系统来看依然能根据逻辑 CHS 地址估算出地址跳转所需大概的时间，只是原本一次切换磁头的动作可能变成一次短距离的切换柱面。

此时的操作系统和文件系统已经开始出现针对 CHS 寻址特点的优化方式， 尽量减少跨磁道的寻址能一定程度提升读写速度，跨磁道时的磁道间距离也会影响寻道时间， 文件系统可能会根据CHS地址来安排数据结构，优化这些寻址时间。

即便使用没有针对 CHS 寻址方式优化过的操作系统和文件系统，比如局限在早期 Windows 和 FAT 系文件系统上，早期这些桌面系统用户们仍然能自己优化磁盘读写性能：通过分区。 分区是硬盘上连续的一段空间，

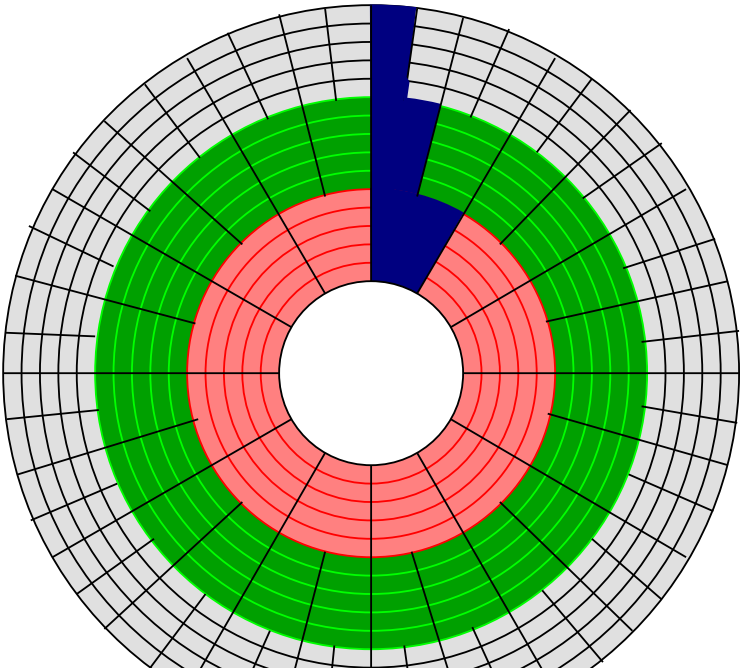
早期由于 BIOS 和 bootloader 的一些技术限制， 每个分区必须对齐到柱面大小上。早期 PC 玩家们通过把一个大硬盘切分成多个小分区， 使用时尽量保持近期读写针对同一个分区，就可以减少寻址时的额外开销，改善读写速度。

于是隐含地，CHS 寻址导致底层硬盘和上层操作系统之间有一层性能约定： **连续读写保证最快的读写速度** 。硬盘实现 CHS 寻址时，调整扇区编号方式让连续的 CHS 地址有最快读写速度，文件系统也根据这个约定，按照 CHS 地址的跳跃来估算读写速度耗时并针对性优化。

# 区位记录（Zone bit recoding, ZBR）

以上物理 CHS 寻址，其实依赖一个假设： **每个磁道上有同样数量的扇区** 。早期硬盘上也的确遵循这个假设， 所以我们上面的图示里才能把一个盘面上的扇区展开成一张长方形的表格，因为每个磁道的扇区数是一样的。 实际上当时的硬盘都是恒定角速度（constant angular velocity, CAV）的方式读写，无论磁头在哪儿，盘片都旋转保持恒定的转速，所以对磁头来说在单位时间内转过的角度影响读写二进制位的数量， 而磁头扫过的面积在这里没有影响。

区位记录（来自维基百科 Zone bit recording 词条）



不过随着硬盘容量增加，盘面的数据密度也随之增加，单位面积中理论能容纳的二进制位数量有限。 理论上，如果保持相同密度的话，盘片外圈能比内圈容纳更多数据。因此硬盘厂商们开始在盘面上将轨道划分出 区块（zone），外圈区块中的轨道可以比内圈区块中的轨

道多放入一些扇区。这种方式下生产出的硬盘叫 区位记录硬盘（Zone bit recoding, ZBR），相对的传统固定轨道中扇区数的硬盘就被叫做恒定角速度（CAV） 硬盘。

如右图所示，区位记录在硬盘上将多个柱面组合成一个区块，区块内的磁道有相同数量的扇区， 而不同区块的磁道可以有不同数量的扇区，外圈区块比内圈区块有更多扇区。

显然要支持 ZBR ，物理 CHS 寻址方式不再有效，于是 ZBR 硬盘将原本简单的地址换算电路升级为更复杂的磁盘控制器芯片，替代 CPU 来驱动硬盘，把来自文件系统的逻辑 CHS 地址通过换算转换到物理 CHS 地址，并且驱动磁头做跳转和寻址。 从而有了独立的控制芯片之后，硬盘读写扇区的速度不再受 CPU 速度影响。有了完整的逻辑-物理地址转换后， 逻辑扇区编号不再对应物理扇区编号，上述编号跳转和坏扇区处理之类的事情都由磁盘控制芯片代为完成。 从而 CHS 地址已经丧失了物理意义，只留下 **连续读写保证最快的读写速度** 这样的性能约定。

有了 ZBR 之后，硬盘读写速度也不再恒定，虽然仍然保持恒定转速，但是读写外圈磁道时单位时间扫过的扇区 多于读写内圈磁道时扫过的扇区。所以 ZBR 硬盘的低端地址比高端地址有更快的读写速度， 通过硬盘测速软件能观察到阶梯状的「掉速」现象。

逻辑地址转换也会造成逻辑 CHS 寻址能访问到的扇区数少于物理 CHS 寻址的现象， 磁盘中扇区被重新编号后可能有一些扇区剩余，于是 ZBR 硬盘的出厂低级格式

化可能会均分这些访问不到的扇区 给每个磁道作为保留扇区，留作坏扇区后备。

另外有了独立磁盘控制器芯片之后，扇区内的校验算法也不再受制于 BIOS INT 13H 接口。 原本 BIOS 的 INT 13H 接口定义了每个扇区 512 字节，额外配有 4 字节校验， 32bit 的校验码对 4096bit 的数据来说，只能允许一些简单的校验算法，比如 汉明码 对 4096bit 的数据需要 13bit 的校验，突破了校验算法限制后硬盘可以在物理扇区中放更多校验位，使用更复杂的 ECC 算法，提供更强的容错性。

通过 ZBR ，逻辑 CHS 寻址不再局限在具体每磁道扇区数等物理限制上，但是仍然局限在 CHS 总位数。 24bit 的 CHS 地址能寻址 \(1024*256*63 = 16515072\) 个扇区，也就是 8064MiB 的空间。 于是早期很多操作系统有 7.8G 硬盘大小的限制。后来 ATA/IDE 标准提升了 CHS 寻址数量，从 24bit 到 28bit 到 32bit ，不过在系统引导早期仍然依赖 BIOS 最基本的 24bit CHS 寻址能力，于是那时候安装系统时要求引导程序装在前 8G 范围内也是这个原因。

# 从 CHS 到 LBA

随着硬盘大小不断提升，无论是操作系统软件层，还是硬盘厂商硬件层，都逐渐意识到逻辑 CHS 寻址是两边相互欺骗对方的骗局：文件系统根据假的 CHS 地址的提示苦苦优化，而硬盘控制器又要把物理 CHS 模拟到假的 CHS 地址上以兼容 BIOS 和操作系统。和 CS 领域太多别的事情一样， CHS 寻址过早地暴露出太多底层抽象细节，而上层软件又转而依赖于这些暴露出的细节进行优化， 底层细节的变动使得上层优化不再是有意义的优化。

于是 ATA 标准 引入了 逻辑块寻址（Logical Block Addressing, LBA） 来替代 CHS 寻址，解决其中的混乱。LBA 的思路其实就是逻辑 CHS 寻址的简单换算，因为 CHS 寻址下 S 从 1 开始计算，而 LBA 使用连续扇区编号，从 0 开始编号，所以换算公式如下：

$$\begin{equation*} \text{LBA 地址} = ( C \times \text{磁头数} + H ) \times \text{每磁道扇区数} + ( S - 1 ) \end{equation*}$$

使用 LBA 寻址，操作系统和文件系统直接寻址一个连续地址空间中的扇区号， 不应该关心柱面和磁头之类的物理参数，将这些物理细节交由磁盘控制器。 对操作系统和文件系统这些上层软件而言，LBA寻址的抽象仍然保证了 **连续读写提供最快的读写速度** ，文件系统仍然会尝试根据 LBA 地址优化，尽量连续读写从而减少寻道时间。

从 CHS 寻址切换到 LBA 寻址，需要硬盘和操作系统两方面的努力，所以很长一段时间， 硬盘同时支持两种寻址方式，在控制器内部做转换。最后需要放弃支持的

是深植了 CHS 寻址的 BIOS，使用 BIOS 引导的 MBR 引导程序还在用 CHS 寻址方式读取数据加载操作系统，直到大家都切换到 UEFI 。

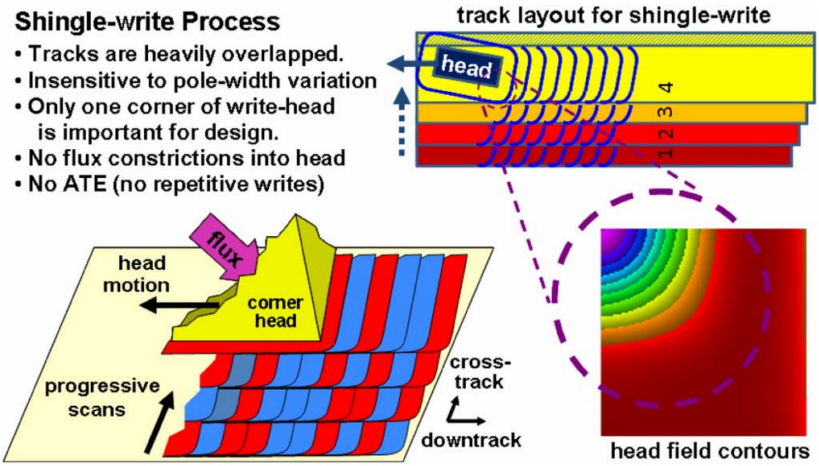
并且随着硬盘使用 LBA 寻址，导致上层软件很难预测底层硬件实际切换柱面切换磁头之类的时机， 潜在得导致一些性能不确定性。于是硬盘控制器在除了负责实际驱动物理磁盘之外， 还开始负责维护一块盘内缓冲区，实现盘内的 IO 队列。缓冲区的存在允许磁盘控制器同时接收更多来自上层软件 的读写请求，转换成实际物理布局参数，并根据磁盘物理布局来调整读写顺序，增加总体吞吐率。 当然有缓冲区的存在也使得突然断电之类的情况下更难保证数据一致性，于是 SCSI/SATA 标准开始约定特殊的请求，从操作系统能发送命令让底层设备清空自己的读写队列。

# 叠瓦磁记录（Shingled Magnetic Recording, SMR）

逐渐从历史讲到了现在，随着硬盘记录密度的不断增加，硬盘厂商们也在不断发明新技术尝试突破磁盘记录的物理极限。 因为有了在硬盘上独立的控制器，并且切换到了逻辑块地址（LBA）的寻址方式， 操作系统大

部分时候不用再关心底层硬盘的物理技术革新，比如垂直写入技术（perpendicular magnetic recording, PMR）将磁头记录方式从水平转换成垂直记录，增加了记录密度，但不影响寻址方式。

叠瓦磁记录（来自 The Feasibility of Magnetic Recording at 10 Terabits Per Square Inch on Conventional Media ）




不过技术革新中也有影响寻址方式的技术，比如 叠瓦磁记录技术（Shingled Magnetic Recording, SMR）。SMR 试图让相邻磁道的写入有部分重叠，从而增加记录密度。有了重叠之后，读取磁道还是能随机定位， 而写入磁道会破坏它后面叠加上的磁道，所以写入磁道必须严格按地址顺序写入。为了满足随机顺序写入的需要， SMR 硬盘把连续的几个磁道组织成区块（zone），

在一个区块内必须按顺序写入。 这里的区块可以和区位记录（ZBR）是同样的区块，也可以独立于 ZBR 做不同大小的区块分割。

这种区块内连续写入的要求，很像是 SSD 这种基于闪存介质的记录方式， SMR 硬盘也同样像 SSD 一样在磁盘控制器内引入 日志结构式的记录方式，采用类似的 GC 算法 ，收到随机写入请求的时候，在区块间执行 GC 搬运数据块，对操作系统提供可以任意写入的抽象接口。

当然这种类似闪存介质的 FTL 的抽象有对读写性能的直接影响。SMR 硬盘可以将这些细节完全隐藏起来（Device Managed），或者完全暴露给宿主系统（Host Managed ），或者隐藏细节的同时在宿主想查询的时候提供细节（Host Aware）。和 SSD 一样，消费级的 SMR 硬盘通常选择隐藏细节只在需要的时候暴露，完全暴露细节的设备通常只在企业服务器级别 的产品中看到。

可以期待，随着 SMR 硬盘的逐渐普及，文件系统设计中也将更多考虑 SMR 的特性加以优化。这些优化可能参考 对 SSD 的优化（比如尽量连续写入），但是又不能完全照搬（比如 SSD 需要考虑写平衡而 SMR 硬盘不需要，比如 SSD 不用担心随机寻道时间而 SMR 硬盘需要）。这些对现在和未来文件系统的设计提供了更多挑战。

# 4KiB 扇区大小

不局限于硬盘，存储设备发展中另一个方向是增加扇区大小。如前所述，在应用于 PC 之前的硬盘设计也曾有过比 512 字节更小的扇区大小，而自从 PC 普及之后 512 字节扇区逐渐成为主流， 甚至到了挥之不去的地步。随着硬盘容量提升，直接寻址 512 字节的扇区显得不再那么高效， 文件系统内部也早已把多个扇区合并成一个逻辑簇（cluster）或者块（block），按簇或块的粒度管理。 在底层硬件同样也是按照 512 字节大小划分扇区，每个扇区都要独立计算校验，如果能增大扇区大小到比如 4KiB，将能更经济地安排扇区校验码，从而得到更多可用容量。可见 512 字节扇区大小这一设计，和 CHS 寻址一样，逐渐成为了操作系统和硬盘厂商彼此间互相努力维护的谎言。

硬盘物理扇区提升为 4KiB 大小的设计，叫做「先进格式化（Advanced Format）」，这样的硬盘叫做先进格式化硬盘（AFD）。在此基础上，硬盘控制器可以提供模拟 512 字节扇区的模拟层， 叫做 512e ，也可以直接提供 4K 大小的扇区给操作系统，叫做 4K native (4Kn)。 操作系统和文件系统要尽量避免依赖 512e 以提供最优性能，支持 4Kn 扇区寻址也是现在和未来 文件系统设计中一个重要挑战。

# 结论（TL;DR）和预告

软件层面的优化与硬件层面的革新一直是一组矛盾。长久以来文件系统和硬盘设备在关于寻址方式的磨合中， 逐渐演化出一条真理，也是我文中一直在强调的： **连续读写提供最快的读写速度** 。文件系统总是能根据底层设备暴露出的一些抽象泄漏，比如物理 CHS 布局，比如 512 字节扇区大小， ，针对性做更多优化，但是随着底层设备的技术革新这些优化也随之成为泡影。

从 SMR 技术中也能看出， 硬盘的读写接口也在逐渐向 SSD 的接口靠拢，从而文件系统的「优化」也在逐渐向这种「倾向顺序写入」的方向优化。关于这些发展趋势待我有空再谈。