柱面-磁頭-扇區尋址的一些舊事

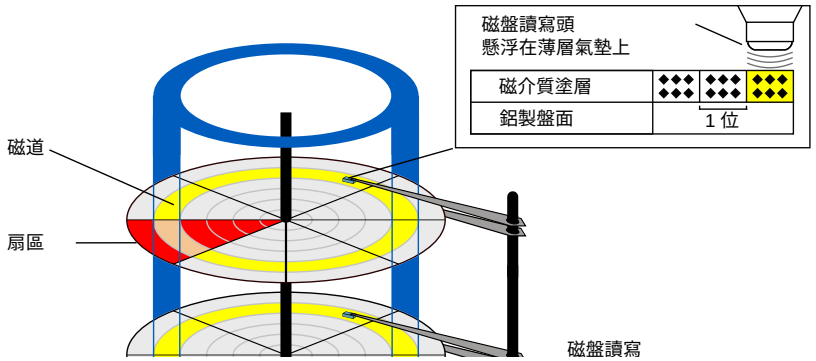
目錄

在 SSD 這種新興存儲設備普及之前，很長一段時間硬盤是個人計算機的主要存儲設備。 更往前的磁帶機不常見於個人計算機，軟盤的地位很快被硬盤取代，到 SSD 出現爲止像 MiniDisc 、 DVD-RAM 等存儲設備也從未能挑戰過硬盤的地位。硬盤作爲主要存儲設備，自然也影響了文件系統的設計。

這篇筆記稍微聊一聊硬盤這種存儲設備的尋址方式對早期文件系統設計的一些影響，特別是 柱面-磁頭-扇區尋址（Cylinder-head-sector addressing, 簡稱CHS尋址）的起源和發展。 大部分內容來自維基百科 Cylinder-head-sector 詞條 這裏只是記錄筆記。現今的硬盤已經不再採用 CHS 尋址，其影響卻還能在一些文件系統設計中看到影子。

柱面、磁頭、扇區以及相關術語

磁盤示意圖（來自維基百科 Cylinder-head-sector 詞條 ）



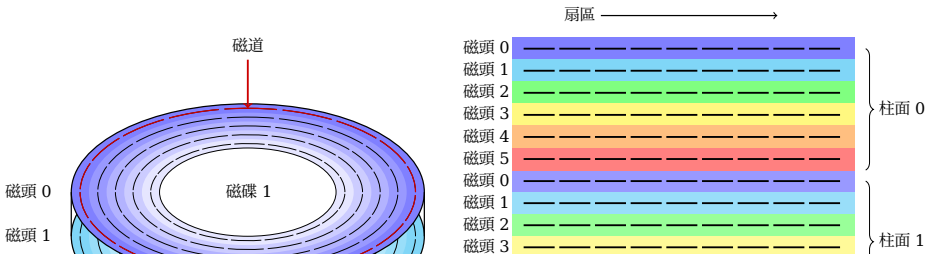
如右圖所示，一塊硬盤(Hard Disk Drive, HDD)是一個圓柱體轉軸上套着一些磁碟片(plate)， 然後有一條磁頭臂(actuator arm)插入磁碟片間的位置，加上一組控制芯片（controller）。 每個磁碟片有上下兩面塗有磁性材質，磁頭臂上有一組磁頭（head），每個磁頭對應磁盤的一個面， 所以比如一個 3 碟的硬盤會有 6 個磁頭。

每個磁碟片上定義了很多同心圓的磁頭軌道，叫做磁道（track），磁道位於盤面上不同半徑的位置， 通過旋轉磁碟臂能讓磁頭移動到特定的半徑上，從而讓讀寫磁頭在不同的磁道間跳轉。 不同磁頭上同磁道的同心圓共同組成一個柱面（cylinder），或者說移動磁碟臂能選定磁盤中的一個柱面。 磁道上按等角度切分成多個小段，叫做扇區（sector），每個扇區是讀寫數據時採用

的最小單元。 早期在 IBM 大型機之類上使用的硬盤的扇區大小比較小，到 IBM PC 開始個人計算機用的硬盤扇區基本被統一到 512 字節。現代硬盤內部可能採用 Advanced Format 使用 4K 字節扇區。

在早期軟盤和硬盤的尋址方式被稱作「柱面-磁頭-扇區尋址」，簡稱 CHS 尋址， 是因爲這三個參數是軟件交給硬件定位到某個具體扇區單元時使用的參數。 首先柱面參數讓磁頭臂移動到某個半徑上，尋址到某個柱面，然後激活某個磁頭，然後隨着盤面旋轉， 磁頭定位到某個扇區上。

「柱面-磁頭-扇區」這個尋址方式，聽起來可能不太符合直覺，尤其是柱面的概念。直覺上， 可能更合理的尋址方式是「盤片-盤面-磁道-扇區」，而柱面在這裏是同磁道不同盤片盤面構成的一個集合。 不過理解了磁盤的機械結構的話，柱面的概念就比較合理了，尋址時先驅動磁頭臂旋轉， 磁頭臂上多個磁頭一起飛到某個磁道上，從而運動磁頭臂的動作定義了一個柱面。 柱面和磁頭（CH）組合起來能定位到某個特定的磁道，畫張圖大概如下圖所示：



上圖中值得注意的是磁道的編號方式，我用相同的顏色畫出了相同的磁道。因爲按照 CHS 的順序尋址，所以先定位柱面，然後選定磁頭。磁盤上按半徑從外向內定義柱面的編號，最外圈的磁道位於 0號柱面，由0號磁頭開始。隨着柱面編號增加，逐步從外圈定位到內圈。

物理 CHS 尋址

以上術語中，柱面號和磁頭號直接對應了硬盤上的物理組成部分，所以通過在物理 CHS 尋址方式下，通過扇區地址的寫法能對應到扇區的具體物理位置。之所以

這樣描述扇區， 是因爲早期的軟盤和硬盤驅動器沒有內置的控制芯片，可以完全由宿主系統執行驅動程序驅動。

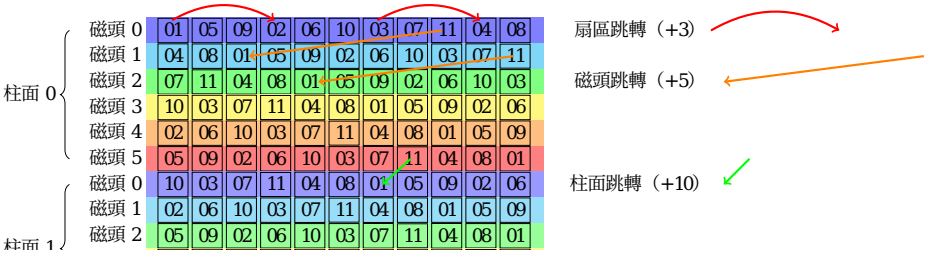
在 IBM PC 上，驅動軟盤和硬盤的是 CPU 執行位於主板 BIOS (Basic Input/Output System) 中的程序，具體來說操作系統（比如DOS）和應用程序調用 INT 13H 中斷，通過 AH=02H/03H 選擇讀/寫操作，BIOS 在中斷表中註冊的 13H 中斷處理程序執行在 CPU 上完成讀寫請求。調用 INT 13H 讀寫扇區的時候，CPU 先通過 INT 13H AH=0CH 控制硬盤的磁頭臂旋轉到特定磁道上，然後選定具體磁頭，讓磁頭保持在磁道上讀數據， 通過忙輪訓的方式等待要讀寫的扇區旋轉到磁頭下方，從而讀到所需扇區的數據。在 DOS 之後的操作系統， 比如早期的 Windows 和 Linux 和 BSD 能以覆蓋中斷程序入口表的方式提供升級版本的這些操作替代 BIOS 的程序。

以上過程中可以看出兩點觀察：

1. CHS 尋址下，跨磁道的尋址（不同 CH 值），和磁道內的尋址（同 CH 不同 S ），是本質上不同的操作。跨磁道的尋址有移動磁頭臂的動作，會比磁道內尋址花費更多時間。
2. 通過扇區號的磁道內尋址是個忙輪訓操作，需要佔用完整 CPU 週期。這也隱含扇區號在一個磁道內的物理排列不必是連續的。

實際上扇區號的物理排列的確不是連續的，每個物理扇區中除了用512字節記錄扇區本身的數據， 還有扇區的開始記錄和結束記錄，寫有扇區編號和扇區校驗

碼。每讀到一個扇區， CPU 可能需要做一些額外操作（比如計算比對校驗、寫入內存緩衝區、調整內存段頁映射） 後纔能繼續讀下一個扇區，如果物理排列上連續編號扇區，可能等 CPU 做完這些事情後磁頭已經旋轉到之後幾個扇區上了。所以出廠時做磁盤低級格式化的時候， 會跳躍着給扇區編號，給 CPU 留足處理時間。比如下圖：



上圖中假設有3個柱面，每個柱面6個磁頭，每個磁道內11個扇區，並且畫出了三種不同的扇區編號跳轉情況， 分別是磁道內的扇區跳轉（+3），柱面內的磁頭跳轉（+5），以及柱面間跳轉（+10）。 實際磁盤上的柱面數、扇區數要多很多，尋址時需要跳轉的距離也可能更長，這裏只是舉例說明。 圖中和實際情況相同的是，柱面號和磁頭號從 0 開始編號，而扇區號從 1 開始編號， 所以做邏輯地址換算的時候要考慮編號差異。

早期 IBM PC 的 BIOS 使用 24bit 的 CHS 地址，其中 10bit 柱面(C)、 8bit 磁頭(H)、 6bit 扇區(S)。從而用物理 CHS 尋址方式的軟盤和硬盤驅動器最多可以尋址 1024

個柱面，256 個磁頭， 63 個扇區，其中扇區數因爲從 1 開始編號所以少了 1 個可尋址範圍。比如 3.5 吋高密度（HD）軟盤有雙面， 出廠時每面 80 磁道，每磁道 18 扇區，從而能算出 1,474,560 字節的容量。

如此跳躍編號扇區之後，不是總能給磁道中所有扇區編號，可能在磁道的末尾位置留幾個沒有使用的扇區空間， 這些是磁道內的保留扇區，可以在發現壞扇區後使用這些隱藏扇區作爲替代扇區。當然讀寫替代扇區的時候 因爲扇區尋址不連續可能會有一定性能損失。

因爲物理 CHS 尋址下，磁盤由 CPU 執行驅動程序來驅動，所以以上扇區跳躍的長短實際是由 CPU 的速度等因素決定的，理論上 CPU 越快，跳躍間隔可以越短，從而磁盤讀寫速度也能加快。磁盤出廠時， 廠商並不知道使用磁盤的計算機會是怎樣的性能，所以只能保守地根據最慢的 CPU 比如 IBM 初代 PC 搭配的 8086 的速度來決定跳躍間隔。所以在當年早期玩家們流傳着這樣一個操作：買到新硬盤， 或者升級了電腦配置之後，對硬盤做一次 低級格式化(Low level formating) ，聰明的低級格式化程序能智能安排扇區編號，提升硬盤讀寫速度，也能跳過已知壞道位置繼續編號， 甚至可能將更多保留扇區暴露成可用扇區。這對現代有硬盤控制器的硬盤而言已經沒有意義了。

邏輯 CHS 尋址

隨着硬盤容量不斷增加， BIOS 中用來 CHS 尋址的地址空間逐漸不夠用了。早期 24bit 地址按 *C H S* 的順序分爲 *10 8 6* 的位數，用 8bit 來尋址磁頭最多可以有 256 個磁頭，而只有 10bit 來尋址柱面，就只能有 1024 個柱面。最初 IBM 這麼劃分是因爲早期用於 IBM 大型機之類的硬盤可以有 厚厚一疊的盤片組，同樣的尋址方式就直接用於了 IBM PC 。而 PC 用的硬盤迫於硬盤倉空間大小， 有厚度限制，硬盤中物理盤面可能只有四五個盤片，硬盤容量增加主要是增加盤片表面的數據密度而非增加盤片數量。

於是逐漸地，硬盤廠商開始對 CHS 尋址的地址空間做一些手腳。比如最初的簡單想法是重新定義 CH ，將一些磁頭數挪用做柱面數。從而有了邏輯 CHS 尋址，其中 CH 是固定一組，通過簡單換算從 CH 值找到物理的柱面和磁頭數。結合 CH 而不映射 S 的優勢在於，從操作系統和文件系統來看依然能根據邏輯 CHS 地址估算出地址跳轉所需大概的時間，只是原本一次切換磁頭的動作可能變成一次短距離的切換柱面。

此時的操作系統和文件系統已經開始出現針對 CHS 尋址特點的優化方式， 儘量減少跨磁道的尋址能一定程度提升讀寫速度，跨磁道時的磁道間距離也會影響尋道時間， 文件系統可能會根據CHS地址來安排數據結構，優化這些尋址時間。

即便使用沒有針對 CHS 尋址方式優化過的操作系統和文件系統，比如侷限在早期 Windows 和 FAT 系文件系統上，早期這些桌面系統用戶們仍然能自己優化磁盤讀寫性能：通過分區。 分區是硬盤上連續的一段空間，

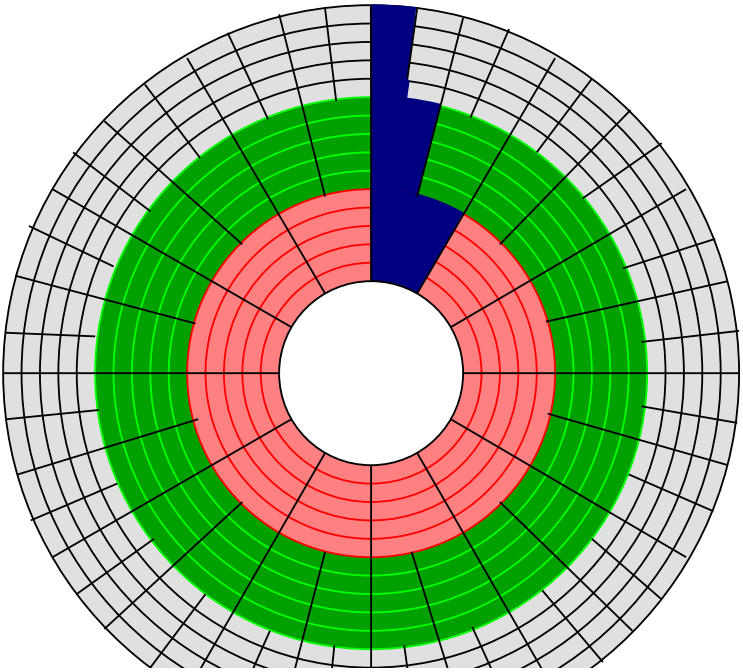
早期由於 BIOS 和 bootloader 的一些技術限制， 每個分區必須對齊到柱面大小上。早期 PC 玩家們通過把一個大硬盤切分成多個小分區， 使用時儘量保持近期讀寫針對同一個分區，就可以減少尋址時的額外開銷，改善讀寫速度。

於是隱含地，CHS 尋址導致底層硬盤和上層操作系統之間有一層性能約定： **連續讀寫保證最快的讀寫速度** 。硬盤實現 CHS 尋址時，調整扇區編號方式讓連續的 CHS 地址有最快讀寫速度，文件系統也根據這個約定，按照 CHS 地址的跳躍來估算讀寫速度耗時並針對性優化。

區位記錄（Zone bit recoding, ZBR）

以上物理 CHS 尋址，其實依賴一個假設： **每個磁道上有同樣數量的扇區** 。早期硬盤上也的確遵循這個假設， 所以我們上面的圖示裏纔能把一個盤面上的扇區展開成一張長方形的表格，因爲每個磁道的扇區數是一樣的。 實際上當時的硬盤都是恆定角速度（constant angular velocity, CAV）的方式讀寫，無論磁頭在哪兒，盤片都旋轉保持恆定的轉速，所以對磁頭來說在單位時間內轉過的角度影響讀寫二進制位的數量， 而磁頭掃過的面積在這裏沒有影響。

區位記錄（來自維基百科 Zone bit recording 詞條）



不過隨着硬盤容量增加，盤面的數據密度也隨之增加，單位面積中理論能容納的二進制位數量有限。 理論上，如果保持相同密度的話，盤片外圈能比內圈容納更多數據。因此硬盤廠商們開始在盤面上將軌道劃分出 區塊（zone），外圈區塊中的軌道可以比內圈區塊中的軌

道多放入一些扇區。這種方式下生產出的硬盤叫 區位記錄硬盤（Zone bit recoding, ZBR），相對的傳統固定軌道中扇區數的硬盤就被叫做恆定角速度（CAV） 硬盤。

如右圖所示，區位記錄在硬盤上將多個柱面組合成一個區塊，區塊內的磁道有相同數量的扇區， 而不同區塊的磁道可以有不同數量的扇區，外圈區塊比內圈區塊有更多扇區。

顯然要支持 ZBR ，物理 CHS 尋址方式不再有效，於是 ZBR 硬盤將原本簡單的地址換算電路升級爲更複雜的磁盤控制器芯片，替代 CPU 來驅動硬盤，把來自文件系統的邏輯 CHS 地址通過換算轉換到物理 CHS 地址，並且驅動磁頭做跳轉和尋址。 從而有了獨立的控制芯片之後，硬盤讀寫扇區的速度不再受 CPU 速度影響。有了完整的邏輯-物理地址轉換後， 邏輯扇區編號不再對應物理扇區編號，上述編號跳轉和壞扇區處理之類的事情都由磁盤控制芯片代爲完成。 從而 CHS 地址已經喪失了物理意義，只留下 **連續讀寫保證最快的讀寫速度** 這樣的性能約定。

有了 ZBR 之後，硬盤讀寫速度也不再恆定，雖然仍然保持恆定轉速，但是讀寫外圈磁道時單位時間掃過的扇區 多於讀寫內圈磁道時掃過的扇區。所以 ZBR 硬盤的低端地址比高端地址有更快的讀寫速度， 通過硬盤測速軟件能觀察到階梯狀的「掉速」現象。

邏輯地址轉換也會造成邏輯 CHS 尋址能訪問到的扇區數少於物理 CHS 尋址的現象， 磁盤中扇區被重新編號後可能有一些扇區剩餘，於是 ZBR 硬盤的出廠低級格式

化可能會均分這些訪問不到的扇區 給每個磁道作爲保留扇區，留作壞扇區後備。

另外有了獨立磁盤控制器芯片之後，扇區內的校驗算法也不再受制於 BIOS INT 13H 接口。 原本 BIOS 的 INT 13H 接口定義了每個扇區 512 字節，額外配有 4 字節校驗， 32bit 的校驗碼對 4096bit 的數據來說，只能允許一些簡單的校驗算法，比如 漢明碼 對 4096bit 的數據需要 13bit 的校驗，突破了校驗算法限制後硬盤可以在物理扇區中放更多校驗位，使用更複雜的 ECC 算法，提供更強的容錯性。

通過 ZBR ，邏輯 CHS 尋址不再侷限在具體每磁道扇區數等物理限制上，但是仍然侷限在 CHS 總位數。 24bit 的 CHS 地址能尋址 \(1024*256*63 = 16515072\) 個扇區，也就是 8064MiB 的空間。 於是早期很多操作系統有 7.8G 硬盤大小的限制。後來 ATA/IDE 標準提升了 CHS 尋址數量，從 24bit 到 28bit 到 32bit ，不過在系統引導早期仍然依賴 BIOS 最基本的 24bit CHS 尋址能力，於是那時候安裝系統時要求引導程序裝在前 8G 範圍內也是這個原因。

從 CHS 到 LBA

隨着硬盤大小不斷提升，無論是操作系統軟件層，還是硬盤廠商硬件層，都逐漸意識到邏輯 CHS 尋址是兩邊相互欺騙對方的騙局：文件系統根據假的 CHS 地址的提示苦苦優化，而硬盤控制器又要把物理 CHS 模擬到假的 CHS 地址上以兼容 BIOS 和操作系統。和 CS 領域太多別的事情一樣， CHS 尋址過早地暴露出太多底層抽象細節，而上層軟件又轉而依賴於這些暴露出的細節進行優化， 底層細節的變動使得上層優化不再是有意義的優化。

於是 ATA 標準 引入了 邏輯塊尋址（Logical Block Addressing, LBA） 來替代 CHS 尋址，解決其中的混亂。LBA 的思路其實就是邏輯 CHS 尋址的簡單換算，因爲 CHS 尋址下 S 從 1 開始計算，而 LBA 使用連續扇區編號，從 0 開始編號，所以換算公式如下：

$$\begin{equation*} \text{LBA 地址} = (\text{C} \times \text{磁頭數} + \text{H}) \times \text{每磁道扇區數} + (\text{S} - 1) \end{equation*}$$

使用 LBA 尋址，操作系統和文件系統直接尋址一個連續地址空間中的扇區號， 不應該關心柱面和磁頭之類的物理參數，將這些物理細節交由磁盤控制器。 對操作系統和文件系統這些上層軟件而言，LBA尋址的抽象仍然保證了 **連續讀寫提供最快的讀寫速度** ，文件系統仍然會嘗試根據 LBA 地址優化，儘量連續讀寫從而減少尋道時間。

從 CHS 尋址切換到 LBA 尋址，需要硬盤和操作系統兩方面的努力，所以很長一段時間， 硬盤同時支持兩種尋址方式，在控制器內部做轉換。最後需要放棄支持的

是深植了 CHS 尋址的 BIOS，使用 BIOS 引導的 MBR 引導程序還在用 CHS 尋址方式讀取數據加載操作系統，直到大家都切換到 UEFI。

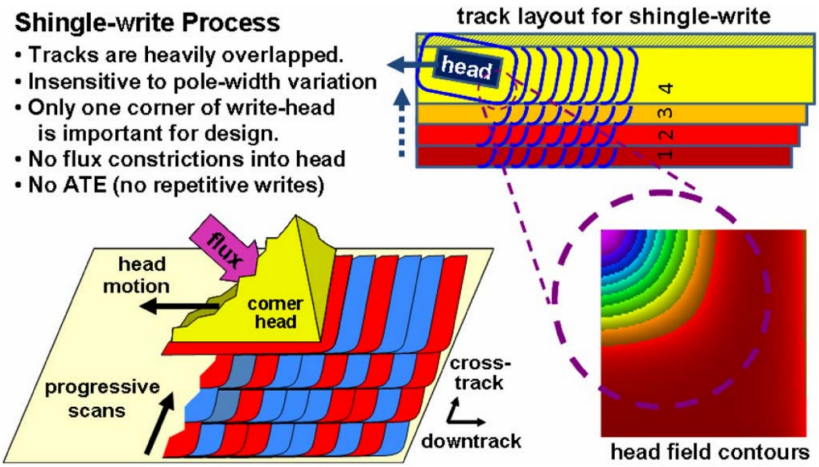
並且隨着硬盤使用 LBA 尋址，導致上層軟件很難預測底層硬件實際切換柱面切換磁頭之類的時機， 潛在得導致一些性能不確定性。於是硬盤控制器在除了負責實際驅動物理磁盤之外， 還開始負責維護一塊盤內緩衝區，實現盤內的 IO 隊列。緩衝區的存在允許磁盤控制器同時接收更多來自上層軟件 的讀寫請求，轉換成實際物理佈局參數，並根據磁盤物理佈局來調整讀寫順序，增加總體吞吐率。 當然有緩衝區的存在也使得突然斷電之類的情況下更難保證數據一致性，於是 SCSI/SATA 標準開始約定特殊的請求，從操作系統能發送命令讓底層設備清空自己的讀寫隊列。

疊瓦磁記錄（Shingled Magnetic Recording, SMR）

逐漸從歷史講到了現在，隨着硬盤記錄密度的不斷增加，硬盤廠商們也在不斷發明新技術嘗試突破磁盤記錄的物理極限。 因爲有了在硬盤上獨立的控制器，並且切換到了邏輯塊地址（LBA）的尋址方式， 操作系統大

部分時候不用再關心底層硬盤的物理技術革新，比如垂直寫入技術（perpendicular magnetic recording, PMR）將磁頭記錄方式從水平轉換成垂直記錄，增加了記錄密度，但不影響尋址方式。

疊瓦磁記錄（來自 The Feasibility of Magnetic Recording at 10 Terabits Per Square Inch on Conventional Media ）



不過技術革新中也有影響尋址方式的技術，比如 疊瓦磁記錄技術（Shingled Magnetic Recording, SMR）。SMR 試圖讓相鄰磁道的寫入有部分重疊，從而增加記錄密度。有了重疊之後，讀取磁道還是能隨機定位， 而寫入磁道會破壞它後面疊加上的磁道，所以寫入磁道必須嚴格按地址順序寫入。爲了滿足隨機順序寫入的需要， SMR 硬盤把連續的幾個磁道組織成區塊（zone），

在一個區塊內必須按順序寫入。 這裏的區塊可以和區位記錄（ZBR）是同樣的區塊，也可以獨立於 ZBR 做不同大小的區塊分割。

這種區塊內連續寫入的要求，很像是 SSD 這種基於閃存介質的記錄方式， SMR 硬盤也同樣像 SSD 一樣在磁盤控制器內引入 日誌結構式的記錄方式，採用類似的 GC 算法 ，收到隨機寫入請求的時候，在區塊間執行 GC 搬運數據塊，對操作系統提供可以任意寫入的抽象接口。

當然這種類似閃存介質的 FTL 的抽象有對讀寫性能的直接影響。SMR 硬盤可以將這些細節完全隱藏起來（Device Managed），或者完全暴露給宿主系統（Host Managed ），或者隱藏細節的同時在宿主想查詢的時候提供細節（Host Aware）。和 SSD 一樣，消費級的 SMR 硬盤通常選擇隱藏細節只在需要的時候暴露，完全暴露細節的設備通常只在企業服務器級別 的產品中看到。

可以期待，隨着 SMR 硬盤的逐漸普及，文件系統設計中也將更多考慮 SMR 的特性加以優化。這些優化可能參考 對 SSD 的優化（比如儘量連續寫入），但是又不能完全照搬（比如 SSD 需要考慮寫平衡而 SMR 硬盤不需要，比如 SSD 不用擔心隨機尋道時間而 SMR 硬盤需要）。這些對現在和未來文件系統的設計提供了更多挑戰。

4KiB 扇區大小

不侷限於硬盤，存儲設備發展中另一個方向是增加扇區大小。如前所述，在應用於 PC 之前的硬盤設計也曾有過比 512 字節更小的扇區大小，而自從 PC 普及之後 512 字節扇區逐漸成爲主流， 甚至到了揮之不去的地步。隨着硬盤容量提升，直接尋址 512 字節的扇區顯得不再那麼高效， 文件系統內部也早已把多個扇區合併成一個邏輯簇（cluster）或者塊（block），按簇或塊的粒度管理。 在底層硬件同樣也是按照 512 字節大小劃分扇區，每個扇區都要獨立計算校驗，如果能增大扇區大小到比如 4KiB，將能更經濟地安排扇區校驗碼，從而得到更多可用容量。可見 512 字節扇區大小這一設計，和 CHS 尋址一樣，逐漸成爲了操作系統和硬盤廠商彼此間互相努力維護的謊言。

硬盤物理扇區提升爲 4KiB 大小的設計，叫做「先進格式化（Advanced Format）」，這樣的硬盤叫做先進格式化硬盤（AFD）。在此基礎上，硬盤控制器可以提供模擬 512 字節扇區的模擬層， 叫做 512e ，也可以直接提供 4K 大小的扇區給操作系統，叫做 4K native (4Kn)。 操作系統和文件系統要儘量避免依賴 512e 以提供最優性能，支持 4Kn 扇區尋址也是現在和未來 文件系統設計中一個重要挑戰。

結論（TL;DR）和預告

軟件層面的優化與硬件層面的革新一直是一組矛盾。長久以來文件系統和硬盤設備在關於尋址方式的磨合中， 逐漸演化出一條真理，也是我文中一直在強調的： **連續讀寫提供最快的讀寫速度** 。文件系統總是能根據底層設備暴露出的一些抽象泄漏，比如物理 CHS 佈局，比如 512 字節扇區大小， ，針對性做更多優化，但是隨着底層設備的技術革新這些優化也隨之成爲泡影。

從 SMR 技術中也能看出， 硬盤的讀寫接口也在逐漸向 SSD 的接口靠攏，從而文件系統的「優化」也在逐漸向這種「傾向順序寫入」的方向優化。關於這些發展趨勢待我有空再談。