文件系統的宏觀結構

前兩篇筆記分別記錄了 硬盤 和 閃存 兩大類存儲設備的物理特性。 這些存儲設備，在操作系統中被抽象爲塊設備（block device）， 是因爲他們不同於隨機存儲設備（Random Access Memory），需要按塊爲單位讀寫。 文件系統構建在塊設備抽象之上，需要根據存儲設備的物理特性做設計和優化，滿足性能和安全性兩方面的考量。

對傳統硬盤而言，尋址時間依賴於地址跳轉間隔，地址跳轉越遠尋址時消耗的時間越長。 因此設計文件系統時要儘量避免遠距離的地址跳轉，讓讀寫儘量連續。對 SSD 而言，雖然沒有了跳轉時間的考量，但是隨機尋址的寫入仍然會加重 FTL 地址翻譯的負擔， 並且增加垃圾回收時寫放大的程度，所以對 SSD 也是同樣，按地址連續寫入時的性能遠優於隨機寫入。 現代硬盤上內外圈的讀寫性能也會影響到文件系統設計，外圈柱面的連續讀寫能比內圈柱面上快很多倍， 甚至一些硬盤的外圈連續讀取速度能高於 SSD 的讀取速度， 所以在硬盤上寫入數據時將經常讀寫的數據（比如文件系統元數據）放入低端地址能有效提升性能。

現代文件系統通過合理安排存儲設備上的數據結構佈局，來爲這些讀寫情況優化性能，這種優化叫地理佈局優化（ geometry based optimization）。 本篇筆記想討論一下文件系統在存儲設備上安排數據時如何做地理佈局優化。文件系統設計中這種優化分兩方面： 安排數據佈局的「宏觀結構（macro structures）」和實際分配地址的「分配器（allocators）」算法。

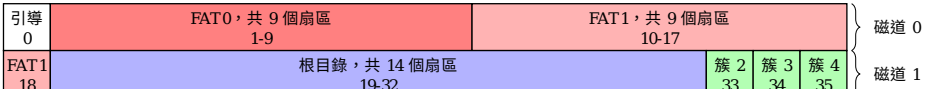
「宏觀結構（macro structures）」是盤上數據結構（on-disk structures）中， 關於如何分割並分配整個地址空間的那些數據結構。「分配器（allocators）」則是在文件系統代碼中負責 在宏觀結構裏分配地址安放數據的具體算法。數據結構和算法相互配合而最優化性能， 所以考察文件系統性能時也應當從這兩者的設計開始考慮。

值得區別的是「宏觀結構」是特定文件系統記錄的方式，所以和文件系統代碼的實現無關， 不同操作系統下實現同一個文件系統要支持同樣的宏觀結構佈局，也難以在保持兼容性的前提下修改宏觀結構。 而「分配器算法」則屬於具體文件系統實現的代碼，可能在使用同樣盤上結構的情況下優化算法、改善分配器性能。 不同操作系統下實現的同一個文件系統也可以採用不同的算法。

DOS文件系統與文件分配表（FAT12 FAT16 VFAT FAT32）

FAT 系文件系統最初爲軟盤驅動器設計，

比如 3.5 吋 1.44M 軟盤使用 FAT12 ，每簇對應一個 512B 扇區， FAT表項是 12bit ，於是最多可以有 \(2^{12}=4096\) 個簇。 之前的文章講過 軟盤上共有 \(2 \times 80 \times 18 = 2880\) 個扇區，所以減去最開始的 33 個非數據區的扇區，軟盤上的 FAT12 有 2847 個簇，於是存一個文件分配表需要至少 \(\frac{2847 \times 12\texttt{bit}}{8\texttt{bit}}=4270.5\) 字節的空間，不到 9 個扇區。按正確大小畫出軟盤上的 FAT12 大概這樣：



注意上圖 FAT12 中非數據區佔用空間，引導扇區+兩份 FAT 表+根目錄全部位於軟盤的首個柱面上， 這是微軟有意如此設計的，從而在讀寫 FAT 的時候磁頭位於最外圈的磁道，不需要移動磁頭。 FAT12 用於軟盤，可用空間不會很大，所以可以一個簇對應一個扇區也不會導致很大的 FAT 表。 相比之下用於硬盤的 FAT16 的大小可以很大，於是需要增加每簇中包含的扇區數量，讓 FAT 表儘量放入首個柱面。

V Size	FAT16			FAT32			exFAT		
V Size									
7 MiB–8 MiB	N/A	N/A	N/A	N/A	N/A	N/A	4KiB	2K	8KiB	4KiB
8 MiB–16 MiB	512B	32K	64KiB	N/A	N/A	N/A	4KiB	4K	16KiB	8KiB
16 MiB–32 MiB	512B	64K	128KiB	512B	64K	256KiB	4KiB	8K	32KiB	8KiB
32	1KiB	64K	128KiB	512B	128K	512KiB	4KiB	16K	64KiB	8KiB

MiB–64 MiB											
64 MiB–128 MiB	2KiB	64K	128K	1KiB	128K	512K	4KiB	32K	128K	4KiB	
128 MiB–256 MiB	4KiB	64K	128K	2KiB	128K	512K	4KiB	64K	256K	4KiB	
256 MiB–512 MiB	8KiB	64K	128K	4KiB	128K	512K	32KiB	256K	64KiB	8KiB	
512 MiB–1 GiB	16KiB	64K	128K	4KiB	256K	1MiB	32KiB	32K	128K	4KiB	
1 GiB–2 GiB	32KiB	64K	128K	4KiB	512K	2MiB	32KiB	64K	256K	4KiB	
2 GiB–4 GiB	64KiB	64K	128K	4KiB	1M	4MiB	32KiB	128K	512K	4KiB	
4 GiB–8 GiB	N/A	N/A	N/A	4KiB	2M	8MiB	32KiB	256K	1MiB	4KiB	
8 GiB–16 GiB	N/A	N/A	N/A	8KiB	2M	8MiB	32KiB	512K	2MiB	4KiB	
16	N/A	N/A	N/A	16KiB	2M	8MiB	32KiB	1M	4MiB	4KiB	

GiB–32 GiB										
32 GiB–16TiB	N/A	N/A	N/A	N/A	N/A	N/A	128K	1G	8G	4KiB
16 TiB–32 TiB	N/A	N/A	N/A	N/A	N/A	N/A	128K	2G	16G	8KiB
32 TiB–64 TiB	N/A	N/A	N/A	N/A	N/A	N/A	128K	4G	32G	16KiB
64 TiB–128 TiB	N/A	N/A	N/A	N/A	N/A	N/A	128K	8G	64G	32KiB
128 TiB–256 TiB	N/A	N/A	N/A	N/A	N/A	N/A	128K	16G	128G	64KiB
> 256 TiB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

從中我們可以算出默認 FAT 表項的最大數量限制：

Volume Size	FAT1	FAT3	exFA
7 MiB–8 MiB	N/A	N/A	2K
8 MiB–16 MiB	32K	N/A	4K
16 MiB–32 MiB	64K	64K	8K

32 MiB–64 MiB	64K	128K	16K
64 MiB–128 MiB	64K	128K	32K
128 MiB–256 MiB	64K	128K	64K
256 MiB–512 MiB	64K	128K	16K
512 MiB–1 GiB	64K	256K	32K
1 GiB–2 GiB	64K	512K	64K
2 GiB–4 GiB	64K	1M	128K
4 GiB–8 GiB	N/A	2M	256K
8 GiB–16 GiB	N/A	2M	512K
16 GiB–32 GiB	N/A	2M	1M
32 GiB–16TiB	N/A	N/A	128M
16 TiB–32 TiB	N/A	N/A	256M
32 TiB–64 TiB	N/A	N/A	512M
64 TiB–128 TiB	N/A	N/A	1G
128 TiB–256 TiB	N/A	N/A	2G
> 256 TiB	N/A	N/A	N/A

和單個 FAT 表大小：

Volume Size	**FAT**	**FAT**	**exFA**
7 MiB–8 MiB	N/A	N/A	8KiB
8 MiB–16 MiB	64KiB	N/A	16KiB
16 MiB–32 MiB	128KiB	256KiB	32KiB
32 MiB–64 MiB	128KiB	512KiB	64KiB
64 MiB–128 MiB	128KiB	512KiB	128KiB
128 MiB–256 MiB	128KiB	512KiB	256KiB
256 MiB–512 MiB	128KiB	512KiB	64KiB
512 MiB–1 GiB	128KiB	1MiB	128KiB

1 GiB–2 GiB	128KiB	2MiB	256KiB
2 GiB–4 GiB	128KiB	4MiB	512KiB
4 GiB–8 GiB	N/A	8MiB	1MiB
8 GiB–16 GiB	N/A	8MiB	2MiB
16 GiB–32 GiB	N/A	8MiB	4MiB
32 GiB–16TiB	N/A	N/A	512MiB
16 TiB–32 TiB	N/A	N/A	1GiB
32 TiB–64 TiB	N/A	N/A	2GiB
64 TiB–128 TiB	N/A	N/A	4GiB
128 TiB–256 TiB	N/A	N/A	8GiB
> 256 TiB	N/A	N/A	N/A

增加簇大小的原因是爲了限制 FAT 表整體的大小，因爲在使用 FAT 表的文件系統中，需要將 FAT 表整體裝入內存才能滿足文件訪問和簇分配時的性能，如果讀寫 FAT 表的範圍需要訪問磁盤，那麼整個文件系統的讀寫性能將暴跌到幾近不可用。到針對閃存優化的 exFAT 上，雖然在 FAT 表外還有額外的簇分配位圖（Cluster Bitmap），但是也同樣要限制 FAT 表整體大小，減少對 FAT 表區域的隨機讀寫。

早期 Unix 文件系統

柱面-磁頭-扇區尋址與地理位置優化

BSD 的 FFS

Linux 的 ext2

物理尋址到邏輯塊尋址

Flash 媒介與 FTL