

主機規劃與磁碟分割

陳建良



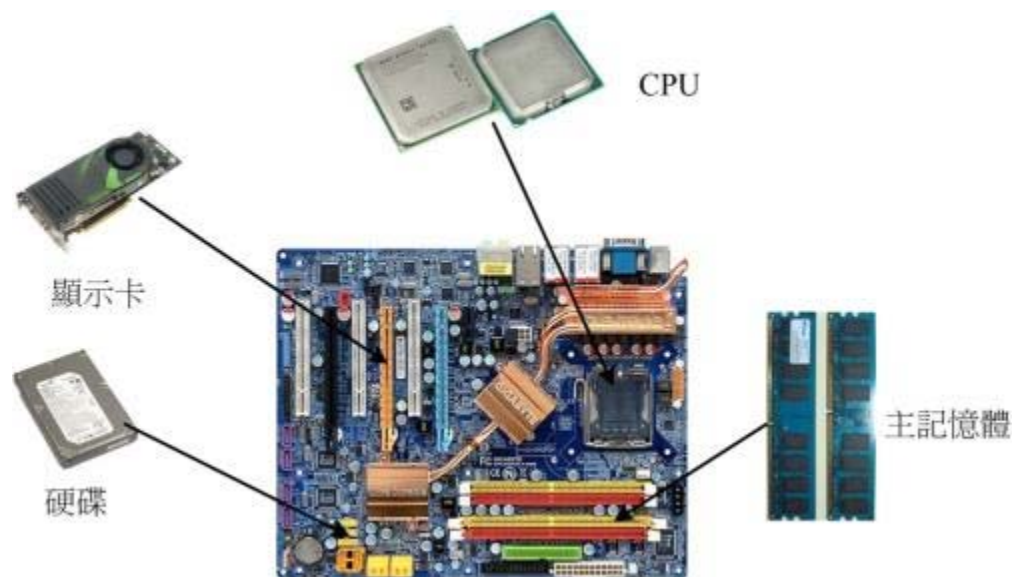
內容

- Linux與硬體的搭配
- 磁碟分割
- 安裝Linux前的規劃

LINUX與硬體的搭配

■ 認識電腦的硬體配備

- 遊戲機/工作機的考量
- 效能/價格比的考量
- 支援度的考量



個人電腦各元件的相關性

選擇與Linux搭配的主機配備

- CPU
- RAM
- Hard Disk
- VGA
- Network Interface Card
- 光碟、軟碟、鍵盤與滑鼠

一般小型主機且不含X WINDOW系統：

- 用途：家庭用NAT主機(IP分享器功能)或小型企業之非圖形介面小型主機。
- CPU：Intel i3等級以上即可。
- RAM：至少512MB，不過還是大於1GB以上比較妥當！
- 網路卡：一般的乙太網路卡即可應付。
- 顯示卡：只要能夠被Linux捉到的顯示卡即可，例如NVidia或ATI的主流顯示卡均可。
- 硬碟：20GB以上即可！

桌上型(DESKTOP)LINUX系統/含X WINDOW：

- 用途：Linux的練習機或辦公室(Office)工作機。(一般我們會用到的環境)
- CPU：最好等級高一點，例如 Intel I5, I7 以上等級。
- RAM：一定要大於1GB比較好！否則容易有圖形介面停頓的現象。
- 網路卡：普通的乙太網路卡就好了！
- 顯示卡：使用256MB以上記憶體顯示卡！(入門級的都這個容量以上了)
- 硬碟：越大越好，最好有60GB。

中型以上Linux伺服器：

- 用途：中小型企業/學校單位的FTP/mail/WWW等網路服務主機。
- CPU：最好等級高一點，例如 I5, I7 以上的多核心系統。
- RAM：最好能夠大於 1GB 以上，大於 4GB 更好！
- 網路卡：知名的broadcom或Intel等廠牌，比較穩定效能較佳！
- 顯示卡：如果有使用到圖形功能，則一張64MB記憶體顯示卡是需要的！
- 硬碟：越大越好，如果可能的話，使用磁碟陣列，或者網路硬碟等等的系統架構，能夠具有更穩定安全的傳輸環境，更佳！
- 建議企業用電腦不要自行組裝，可購買商用伺服器較佳，因為商用伺服器已經通過製造商的散熱、穩定度等測試，對於企業來說，會是一個比較好的選擇。

各硬體裝置在Linux中的檔名

裝置	裝置在Linux內的檔名
IDE硬碟機	/dev/hd[a-d]
SCSI/SATA/USB硬碟機	/dev/sd[a-p]
USB快閃碟	/dev/sd[a-p](與SATA相同)
軟碟機	/dev/fd[0-1]
印表機	25針: /dev/lp[0-2] USB: /dev/usb/lp[0-15]
滑鼠	USB: /dev/usb/mouse[0-15] PS2: /dev/psaux
當前CDROM/DVDROM	/dev/cdrom
當前的滑鼠	/dev/mouse
磁帶機	IDE: /dev/ht0 SCSI: /dev/st0

例題：

- 如果你的PC上面有兩個SATA磁碟以及一個USB磁碟，而主機板上面有六個SATA的插槽。這兩個SATA磁碟分別安插在主機板上的SATA1, SATA5插槽上，請問這三個磁碟在Linux中的裝置檔名為何？

答：

由於是使用偵測到的順序來決定裝置檔名，並非與實際插槽代號有關，因此裝置的檔名如下：

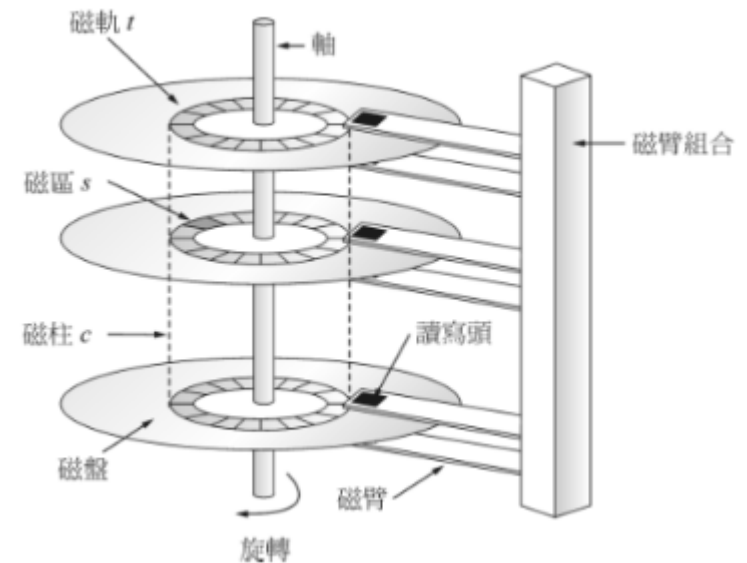
SATA1插槽上的檔名：/dev/sda

SATA5插槽上的檔名：/dev/sdb

USB磁碟(開機完成後才被系統捉到)：/dev/sdc

磁碟的組成

- Track (磁軌)
- Sector (磁區)
- Cylinder (磁柱)
 - 不同面之相同Tracks形成之集合
- Read/Write Head (讀寫頭)

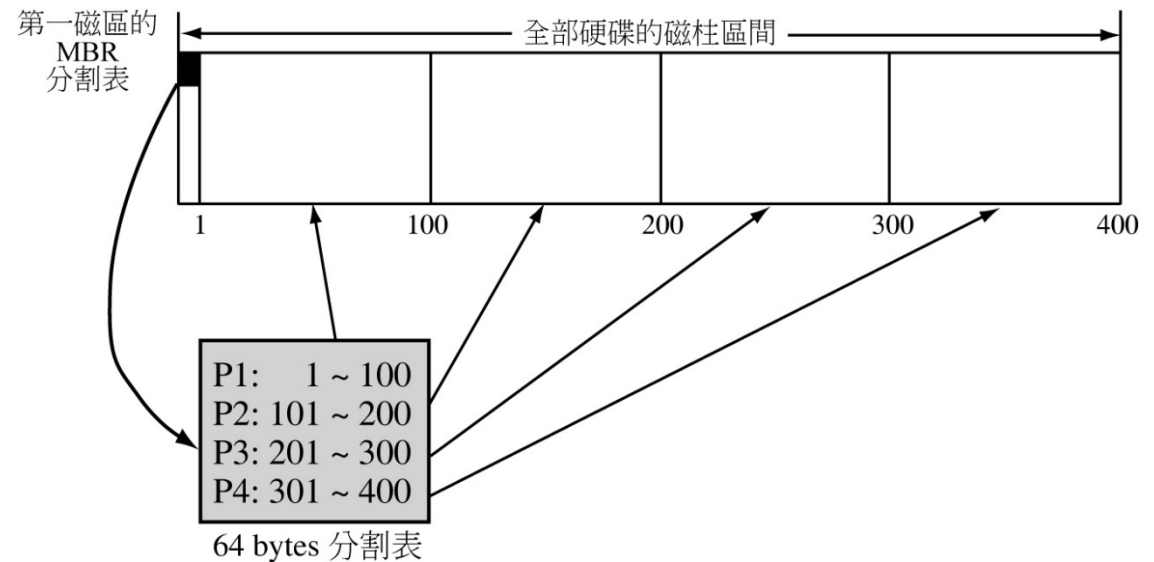


磁碟分割 -- MBR

- 那麼是否每個磁區都一樣重要呢？其實整顆磁碟的**第一個磁區特別的重要**，因為他記錄了整顆磁碟的重要資訊！磁碟的第一個磁區主要記錄了重要的資訊，分別是：
 - 主要開機記錄區(Master Boot Record, MBR)：可以安裝開機管理程式的地方，有446 bytes
 - 分割表(partition table)：記錄整顆硬碟分割的狀態，有64 bytes
 - 結束符號: 共2 bytes
 - MBR定址能力只到「2TB」容量
 - 由於MBR原始設計上採用32位元來代表邏輯磁區(Logical sector)，而每個磁區大小為512位元組，因此磁碟定址大小受限於2TB ($2^{32} \times 512$) 位元組。

磁碟分割表(PARTITION TABLE)

- 利用參考對照磁柱號碼的方式來處理：
- 在分割表所在的64 bytes容量中，總共分為四組記錄區，每組記錄區記錄了該區段的起始與結束的磁柱號碼。若將硬碟以長條形來看，然後將磁柱以直條圖來看，那麼那64 bytes的記錄區段有點像右方的圖示



磁碟分割表(PARTITION TABLE)

- 假設上面的硬碟裝置檔名為/dev/sda時，那麼這四個分割槽在Linux系統中的裝置檔名如下所示，重點在於檔名後面會再接一個數字，這個數字與該分割槽所在的位置有關喔！

- P1:/dev/sda1

- P2:/dev/sda2

- P3:/dev/sda3

- P4:/dev/sda4

磁碟分割

- 為啥要分割啊？基本上你可以這樣思考分割的角度：
 1. 資料的安全性
 2. 系統的效能考量
- 你可以將一顆硬碟分割成十個以上的分割槽的！那又是如何達到的呢？
 1. 在Windows/Linux系統中，我們可以透過的延伸分割(Extended)的方式來處理！
 2. 延伸分割的想法是：既然第一個磁區所在的分割表只能記錄四筆資料，那我可否利用額外的磁區來記錄更多的分割資訊？

磁碟分割

■ 右圖的分割槽在Linux系統中的裝置檔名分別如下：

■ P1:/dev/sda1

■ P2:/dev/sda2

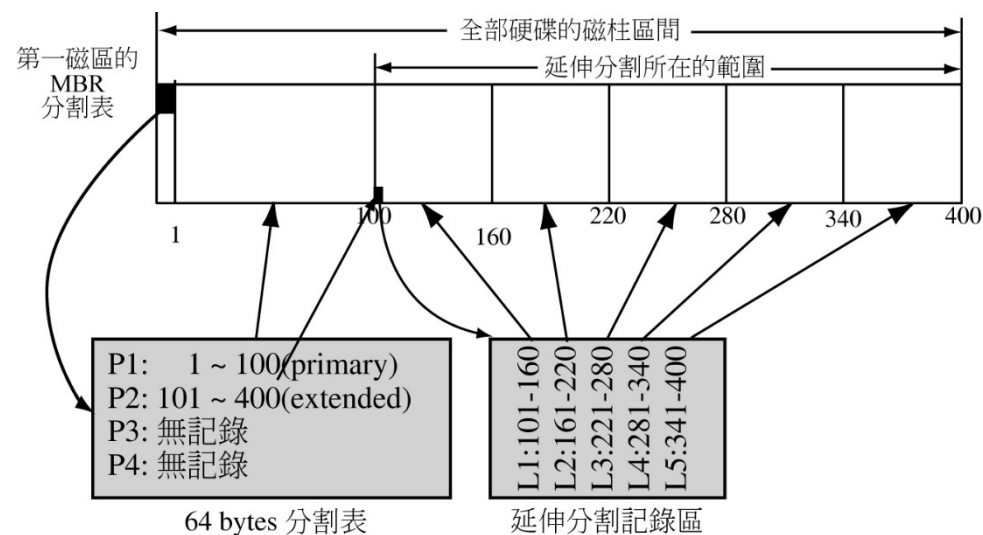
■ L1:/dev/sda5

■ L2:/dev/sda6

■ L3:/dev/sda7

■ L4:/dev/sda8

■ L5:/dev/sda9



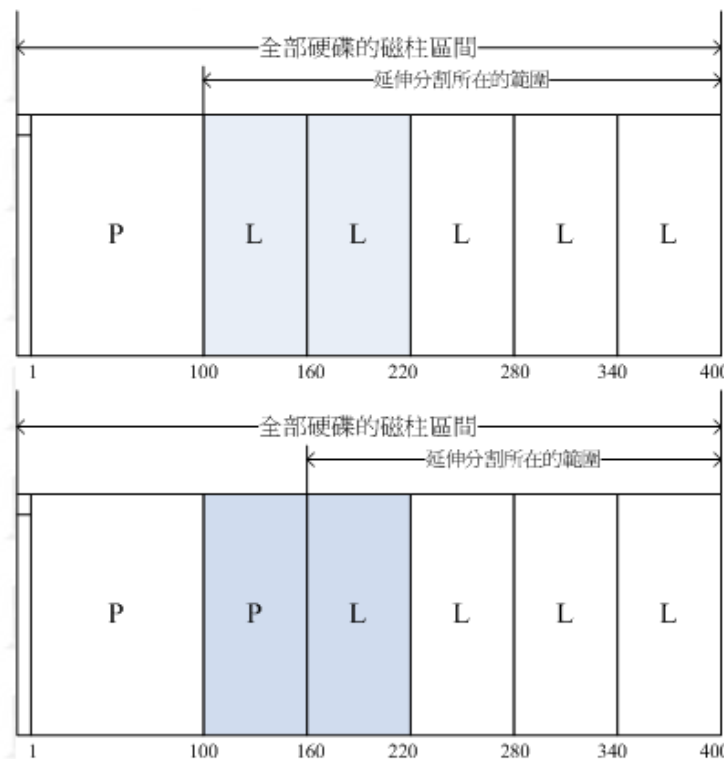
磁碟分割表的作用示意圖

磁碟分割

- 主要分割、延伸分割與邏輯分割的特性我們作個簡單的定義：
 - 主要分割與延伸分割最多可以有**四筆**(硬碟的限制)
 - 延伸分割最多只能有**一個**(作業系統的限制)
 - 邏輯分割**是由延伸分割持續切割出來的分割槽**；
 - 能夠被格式化後，作為資料存取的分割槽為主要分割與邏輯分割。延伸分割無法格式化；
 - 邏輯分割的數量依作業系統而不同，在Linux系統中，IDE硬碟最多有**59個邏輯分割(5號到63號)**，SATA硬碟則有**11個邏輯分割(5號到15號)**。

磁碟分割

- 在Windows作業系統當中，如果你想要將D與E槽整合成為一個新的分割槽，而如果有兩種分割的情況如下圖所示，圖中的特殊顏色區塊為D與E槽的示意，請問這兩種方式是否均可將D與E整合成為一個新的分割槽？



磁碟分割

- 上圖可以整合：因為上圖的D與E同屬於延伸分割內的邏輯分割，因此只要將兩個分割槽刪除，然後再重新建立一個新的分割槽，就能夠在不影響其他分割槽的情況下，將兩個分割槽的容量整合成為一個。
- 下圖不可整合：因為D與E分屬主分割與邏輯分割，兩者不能夠整合在一起。除非將延伸分割破壞掉後再重新分割。但如此一來會影響到所有的邏輯分割槽，要注意的是：如果延伸分割被破壞，所有邏輯分割將會被刪除。因為邏輯分割的資訊都記錄在延伸分割裡面嘛！

例題：如果我想將一顆大硬碟『暫時』分割成為四個partitions，同時還有其他的剩餘容量可以讓我在未來的時候進行規劃，我能不能分割出四個Primary？若不行，那麼你建議該如何分割？

- 由於Primary+Extended最多只能有四個，其中Extended最多只能有一個，這個例題想要分割出四個分割槽且還要預留剩餘容量，因此P+P+P+P的分割方式是不適合的。因為如果使用到四個P，則即使硬碟還有剩餘容量，因為無法再繼續分割，所以剩餘容量就被浪費掉了。
- 假設你想要將所有的四筆記錄都花光，那麼P+P+P+E是比較適合的。所以可以用的四個partitions有3個主要及一個邏輯分割，剩餘的容量在延伸分割中。
- 如果你要分割超過4槽以上時，一定要有Extended分割槽，而且必須將所有剩下的空間都分配給Extended，然後再以logical的分割來規劃Extended的空間。另外，考慮到磁碟的連續性，一般建議將Extended的磁柱號碼分配在最後面的磁柱內。

我能不能僅分割出一個Primary與
一個Extended即可？

- 當然可以，這也是早期Windows作業系統慣用的手法！此外，邏輯分割槽的號碼在IDE可達63號，SATA則可達15號，因此僅一個主要與一個延伸分割即可，因為延伸分割可繼續被分割出邏輯分割槽嘛！

假如我的PC有兩顆SATA硬碟，我想在第二顆硬碟分割出6個可用的分割槽(可以被格式化來存取資料之用)，那每個分割槽在Linux系統下的裝置檔名為何？且分割類型各為何？至少寫出兩種不同的分割方式。

P+P+P+E的環境：

- 實際可用的是
/dev/sdb1, /dev/sdb2,
/dev/sdb3, /dev/sdb5,
/dev/sdb6, /dev/sdb7這
六個，至於/dev/sdb4
這個延伸分割本身僅
是提供來給邏輯分割
槽建立之用。

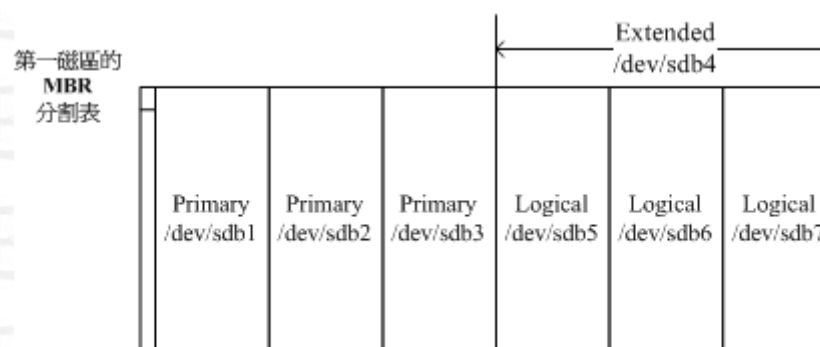


圖2.3.4、分割示意圖



P+E的環境：

- 注意到了嗎？因為1~4號是保留給主要/延伸分割槽的，因此第一個邏輯分割槽一定是由5號開始的！再次強調啊！所以/dev/sdb3, /dev/sdb4就會被保留下來沒有用到了！

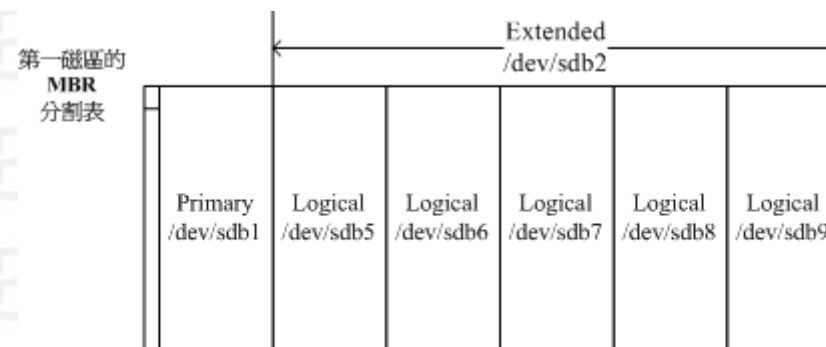


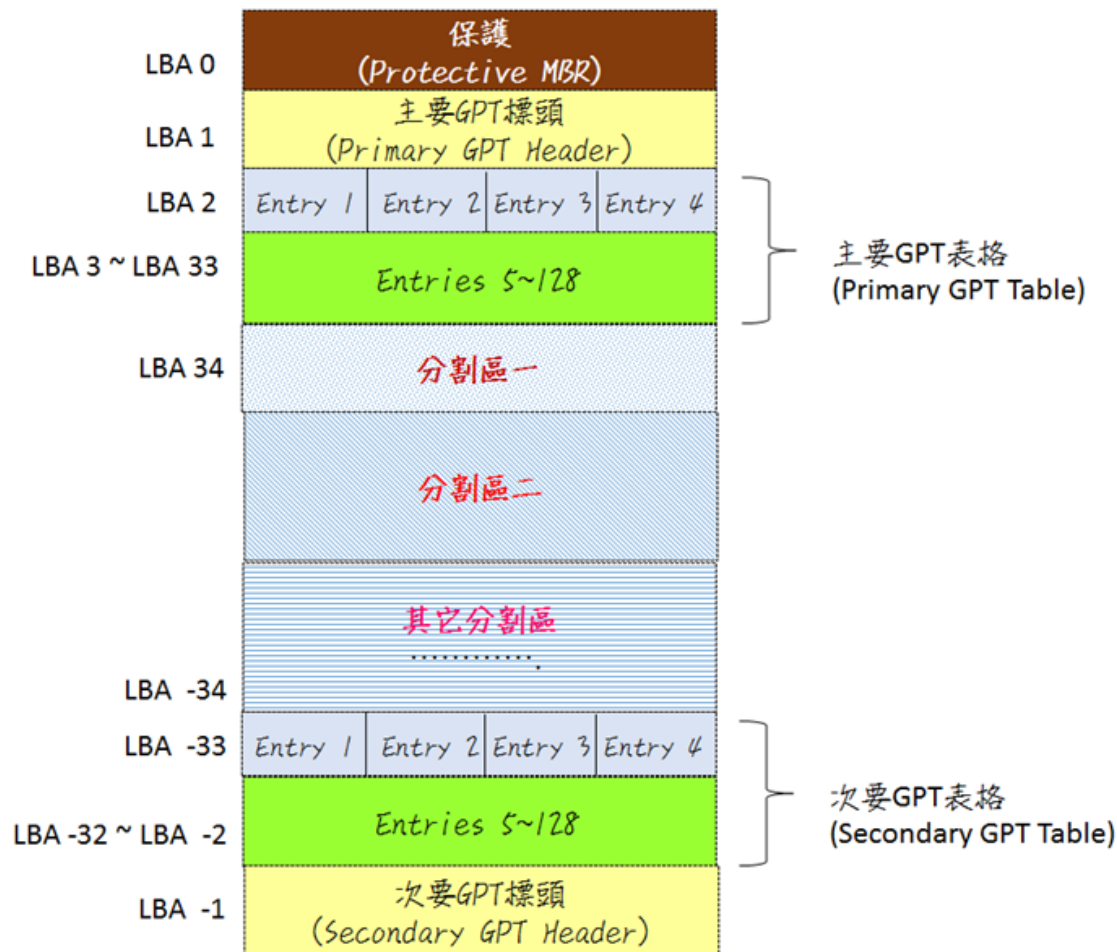
圖2.3.5、分割示意圖



GUID PARTITION TABLE, GPT 磁碟分割表

- 64 位元邏輯區塊定址(Logical Block Address, LBA)
 - GPT 不再採用傳統的磁柱-磁頭-磁區(cylinders-heads-sectors ; CHS)定址模式，而採用64位元的邏輯區塊位址(Logical Block Address, LBA)，因此不再受到2TB的大小限制，最大理論空間為8 ZB
- 改進的分割技術
 - GPT可以支援的分割數量不再受限於四個，預設上可支援高達128個分割區。
 - 傳統的MBR只設計一個位元組來定義分割類型，GPT則利用16個位元組(128位元)的GUID來辨識每個分割區，並採用GUID和屬性定義分割類型，有效的降低的分割類型的識別衝突。
 - 每個GPT分割區的名稱可利用一個36字元(72位元組)長度，支援萬國碼(unicode)的友善名稱來表達。
- 強化容錯能力
 - 採用主要和備援表格以支援容錯能力，將GPT資料結構儲存於開頭以及結尾二份，另一方面，利用32位元的CRC檢查總和提升標頭和分割欄位的資料完整性。

GUID PARTITION TABLE, GPT 磁碟分割表



GPT 在每筆紀錄中分別提供了 64bits 來記載開始/結束的磁區號碼，因此，GPT 分割表對於單一分割槽來說，他的最大容量限制就會在『 $2^{64} * 512\text{bytes} = 2^{63} * 1\text{Kbytes} = 2^{33} * 1\text{TB} = 8\text{ZB}$ 』，要注意 $1\text{ZB} = 2^{30}\text{TB}$ 啦！

GPT包含了下列重要的邏輯分割區塊

- **Protective MBR (LBA 0)**：這是為了相容性考量而設計的第一個邏輯磁區(LBA 0)，包含一個0xEE 類型的主要分割欄位以定義整個磁碟大小，主要設計目的在用來避免那些不支援GPT的硬碟管理工具由於錯誤識別而破壞硬碟中的資料，因此稱為保護MBR。
- **Primary GPT Header (LBA 1)**：包括了一個唯一的磁碟GUID，主要分割表的位置，分割表的可用欄位數量，本身的CRC32值和次要GPT標頭的位置 (Secondary GPT Header)。
- **Primary GPT Table**：包括128個分割區欄位，每個欄位有128位元組長度，包含了16位元組的Partition type GUID和另一個16位元組的Unique partition GUID，以及第一個與最後一個邏輯磁區位址(First and Last LBA)。
- **Secondary GPT Table (LBA -1)**：備援容錯之用，假如主要的GPT表格毀損，可利用來恢復。

開機流程與主要開機記錄區(MBR)

■ 簡單的說，整個開機流程到作業系統之前的動作應該是這樣的：

1. BIOS：開機主動執行的韌體，會認識第一個可開機的裝置；
2. MBR：第一個可開機裝置的第一個磁區內的主要開機記錄區塊，內含開機管理程式；
3. 開機管理程式(boot loader)：一支可讀取核心檔案來執行的軟體；
4. 核心檔案：開始作業系統的功能...

開機流程與主要開機記錄區(MBR)

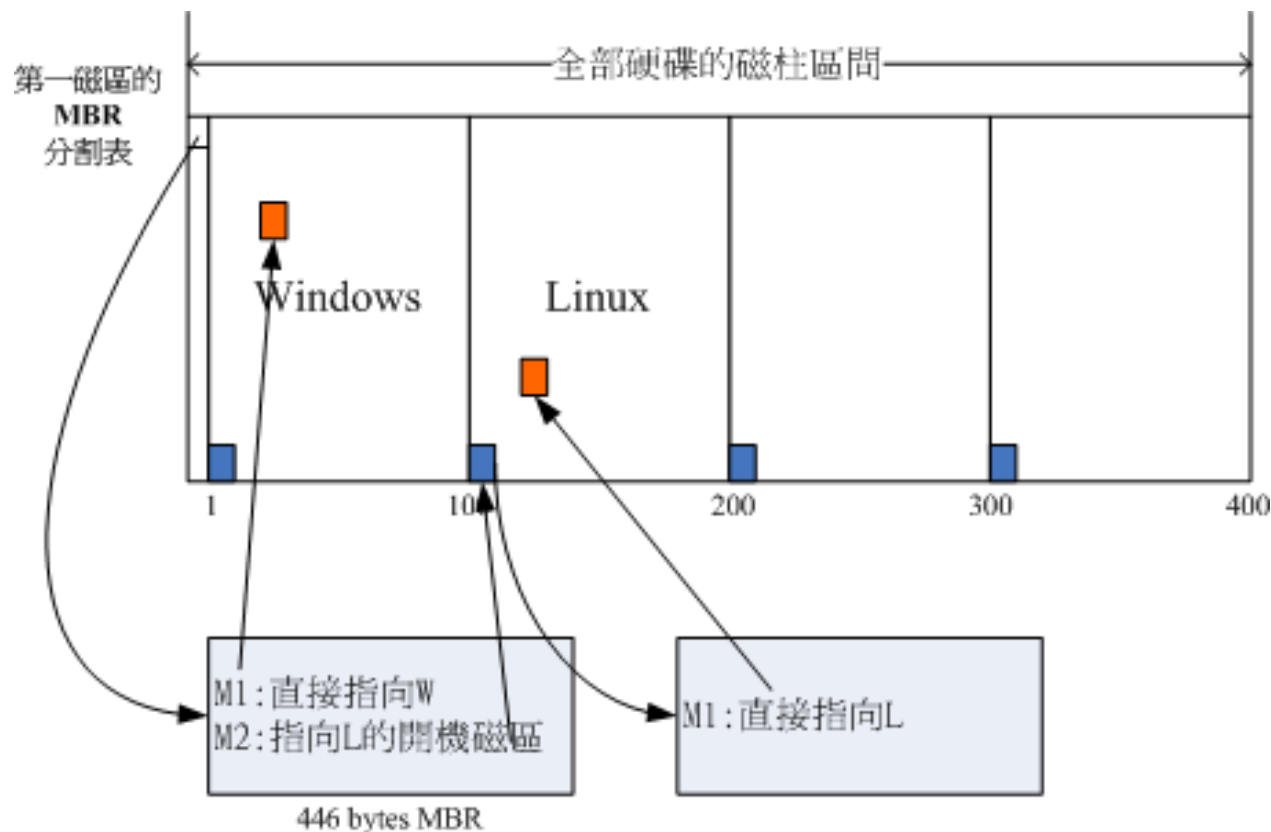
- 如果你的分割表為 **GPT** 格式的話，那麼 **BIOS** 也能夠從 **LBA0** 的 **MBR** 相容區塊讀取第一階段的開機管理程式碼，如果你的開機管理程式能夠認識 **GPT** 的話，那麼使用 **BIOS** 同樣可以讀取到正確的作業系統核心喔！
- **BIOS**與**MBR**都是硬體本身會支援的功能，至於**Boot loader**則是作業系統安裝在**MBR**上面的一套軟體了。由於**MBR**僅有**446 bytes**而已，因此這個開機管理程式是非常小而美的。

BOOT LOADER的主要任務有底下這些項目

- 提供選單：使用者可以選擇不同的開機項目，這也是多重開機的重要功能！
- 載入核心檔案：直接指向可開機的程式區段來開始作業系統；
- 轉交其他loader：將開機管理功能轉交給其他loader負責。

開機管理程式除了可以安裝在MBR之外，還可以安裝在每個分割槽的開機磁區(boot sector)喔

假設你的個人電腦只有一個硬碟，裡面切成四個分割槽，其中第一、二分割槽分別安裝了Windows及Linux，你要如何在開機的時候選擇用Windows還是Linux開機呢？假設MBR內安裝的是可同時認識Windows/Linux作業系統的開機管理程式，那麼整個流程可以圖示如下：



- MBR的開機管理程式提供兩個選單，選單一(M1)可以直接載入Windows的核心檔案來開機；選單二(M2)則是將開機管理工作交給第二個分割槽的開機磁區(boot sector)。
- 當使用者在開機的時候選擇選單二時，那麼整個開機管理工作就會交給第二分割槽的開機管理程式了。
- 當第二個開機管理程式啟動後，該開機管理程式內(上圖中)僅有一個開機選單，因此就能夠使用Linux的核心檔案來開機囉。這就是多重開機的工作情況啦！

那現在請你想一想，為什麼人家常常說：『如果要安裝多重開機，最好先安裝WINDOWS再安裝Linux』呢？這是因為：

- Linux在安裝的時候，你可以選擇將開機管理程式安裝在MBR或各別分割槽的開機磁區，而且Linux的loader可以手動設定選單(就是上圖的M1, M2...)，所以你可以在Linux的boot loader裡面加入Windows開機的選項；
- Windows在安裝的時候，他的安裝程式會主動的覆蓋掉MBR以及自己所在分割槽的開機磁區，你沒有選擇的機會，而且他沒有讓我們自己選擇選單的功能。

UEFI BIOS 搭配 GPT 開機的流程

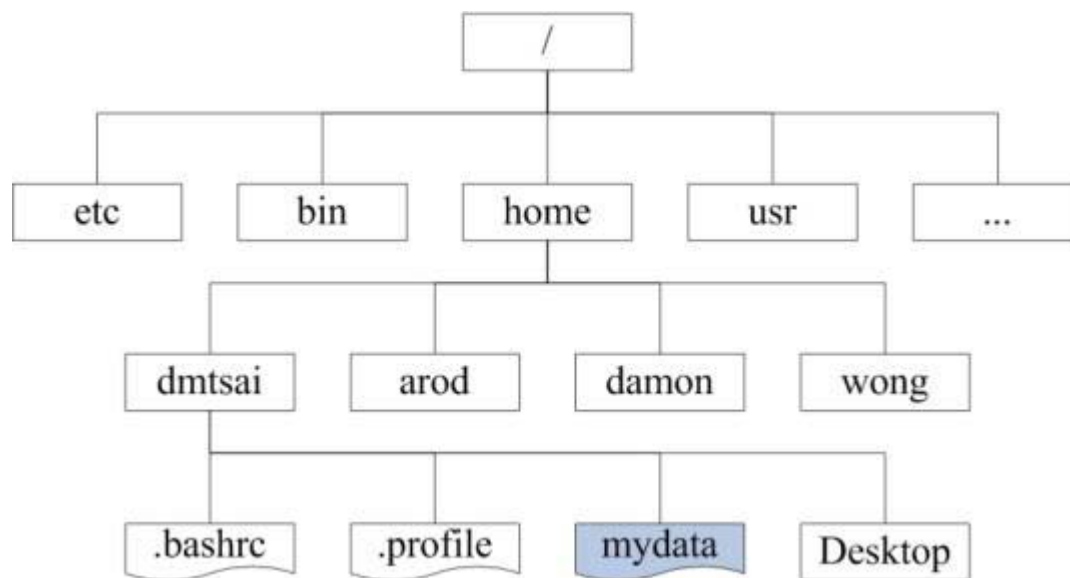
- GPT 可以提供到 64bit 的定址，然後也能夠使用較大的區塊來處理開機管理程式。但是 BIOS 其實不懂 GPT 耶！還得要透過 GPT 提供相容模式才能夠讀寫這個磁碟裝置
- BIOS 僅為 16 位元的程式，在與現階段新的作業系統接軌方面有點弱掉了！
- 為了解決這個問題，因此就有了 UEFI (Unified Extensible Firmware Interface) 這個統一可延伸韌體界面的產生。
- UEFI 主要是想要取代 BIOS 這個韌體界面，因此我們也稱 UEFI 為 UEFI BIOS 就是了。

比較項目	傳統 BIOS	UEFI
使用程式語言	組合語言	C 語言
硬體資源控制	使用中斷 (IRQ) 管理 不可變的記憶體存取 不可變得輸入/輸出存取	使用驅動程式與協定
處理器運作環境	16 位元	CPU 保護模式
擴充方式	透過 IRQ 連結	直接載入驅動程式
第三方廠商支援	較差	較佳且可支援多平台
圖形化能力	較差	較佳
內建簡化作業系統前環境	不支援	支援

1. 與傳統的 BIOS 不同，UEFI 簡直就像是一個低階的作業系統～甚至於連主機板上面的硬體資源的管理，也跟作業系統相當類似，只需要載入驅動程式即可控制操作。
2. 同時由於程式控制得宜，一般來說，使用 UEFI 介面的主機，在開機的速度上要比 BIOS 來的快上許多！

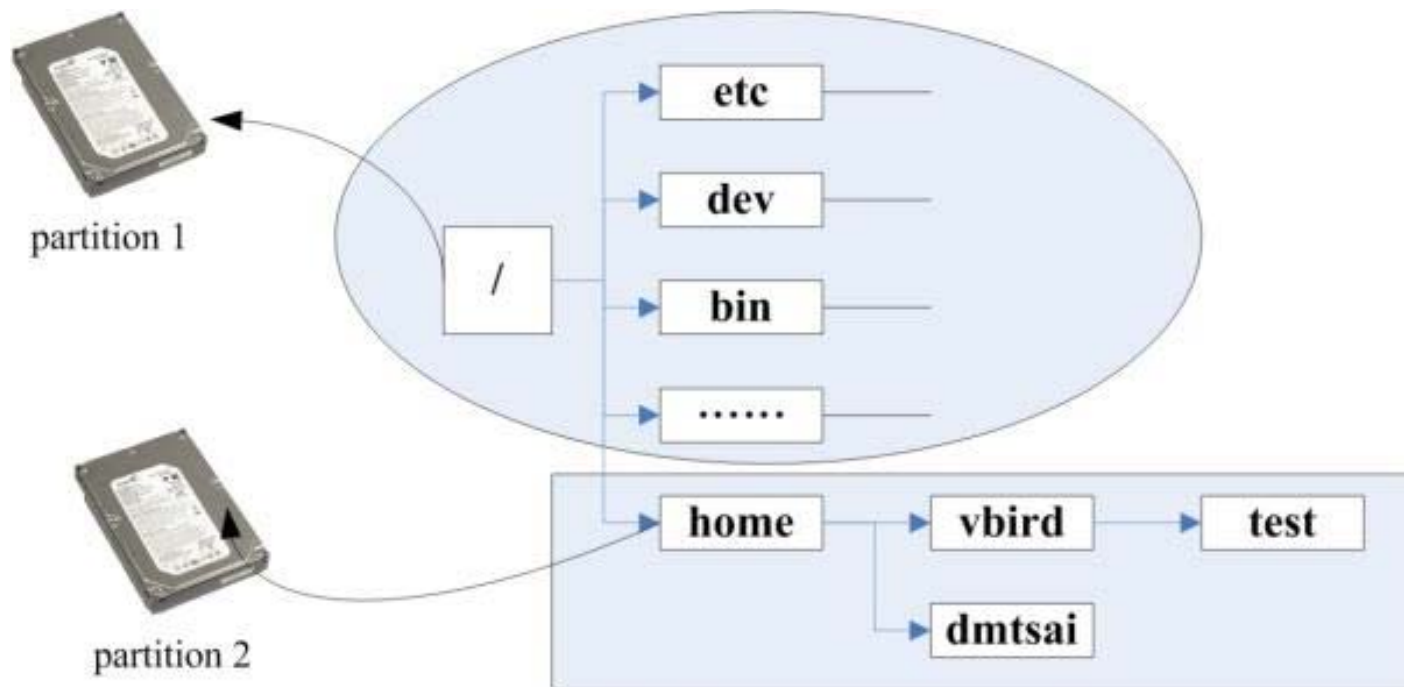
LINUX安裝模式下，磁碟分割的選擇(極重要)

- Linux系統最重要的地方就是在於目錄樹架構。
- 所謂的目錄樹架構(directory tree)就是以根目錄為主，然後向下呈現分支狀的目錄結構的一種檔案架構。
- 整個目錄樹架構最重要的就是那個根目錄(root directory)，這個根目錄的表示方法為一條斜線『/』



檔案系統與目錄樹的關係(掛載)

- 所謂的『掛載』就是利用一個目錄當成進入點，將磁碟分割槽的資料放置在該目錄下；也就是說，進入該目錄就可以讀取該分割槽的意思。
- 這個動作我們稱為『掛載』，那個進入點的目錄我們稱為『掛載點』。
- 由於整個Linux系統最重要的是根目錄，因此根目錄一定需要掛載到某個分割槽的。
- 至於其他的目錄則可依使用者自己的需求來給予掛載到不同的分割槽。



1. 假設我的硬碟分為兩槽，partition 1是掛載到根目錄，至於partition 2則是掛載到/home這個目錄。
2. 當我的資料放置在/home內的各次目錄時，資料是放置到partition 2的
3. 如果不是放在/home底下的目錄，那麼資料就會被放置到partition 1了！

安裝時，掛載點與磁碟分割的規劃

- 自訂安裝『Custom』：
 - 初次接觸Linux：只要分割『/』及『swap』即可
 - 或選擇Linux安裝程式提供的預設硬碟分割方式
- 『Expert 專家模式』
 - 針對幾個重要的目錄掛載分割磁區
 - 例如: /usr , /var/log, /home ... etc.

選擇適當的Distribution – 以CentOS為例

- 最新的版本是CentOS 7.2版。不過，從 CentOS 7.0 版本開始，安裝光碟已經不再提供 386 相容版本了，亦即僅有 64 位元的硬體才能夠使用該安裝光碟來裝系統了！
- 下載的檔名會是 CentOS-7-x86_64-Everything-1503-01.iso 這樣的格式？那個 1503 是啥東西啊？其實從 CentOS 7 之後，版本命名的依據就跟發表的日期有關了！那個 CentOS-7 講的是 7.x 版本，x86_64 指的是 64 位元作業系統，Everything 指的是包山包海的版本，1503 指的是 2015 年的 3 月發表的版本，01.iso 則得要與 CentOS7 搭配，所以是 CentOS 7.1 版的意思！

主機的服務規劃與硬體的關係

- 打造Windows與Linux共存的環境
- NAT(達成IP分享器的功能)
- SAMBA(加入Windows網路上的芳鄰)
- Mail(郵件伺服器)
- Web(WWW伺服器)
- DHCP(提供用戶端自動取得IP的功能)
- Proxy(代理伺服器)
- FTP

打造WINDOWS與LINUX共存的環境

- 在某些情況之下，你可能會想要在『一部主機上面安裝兩套以上的作業系統』
- 那麼剛剛我們談到的開機流程與多重開機的資料就很重要了。
- 如果你的Linux主機已經是想要拿來作為某些服務之用時，那麼務必不要選擇太老舊的硬體喔！前面談到過，太老舊的硬體可能會有電子零件老化的問題～
- 另外，如果你的Linux主機必須要全年無休的開機著，那麼擺放這部主機的位置也需要選擇啊！

NAT(達成IP分享器的功能)

- 在這種環境中，由於Linux作為一個內/外分離的實體，因此網路流量會比較大一點。此時Linux主機的網路卡就需要比較好些的配備。其他的CPU、RAM、硬碟等等的影響就小很多。
- 好處 -- Linux NAT還可以額外的加裝很多分析軟體，可以用來分析用戶端的連線，或者是用來控制頻寬與流量，達到更公平的頻寬使用呢！
- 缺點 -- Linux作為NAT主機來分享IP是很不智的～因為PC的耗電能力比IP分享器要大的多～

SAMBA(加入WINDOWS網路上的芳鄰)

- 可以使用Linux上面的SAMBA這個軟體來達成加入Windows網芳的功能！
- SAMBA的效能不錯，也沒有用戶端連線數的限制，相當適合於一般學校環境的檔案伺服器(file server)的角色呢！
- 這種伺服器由於分享的資料量較大，對於系統的網路卡與硬碟的大小及速度就比較重要。
- 如果你還針對不同的使用者提供檔案伺服器功能，那麼/home這個目錄可以考慮獨立出來，並且加大容量。

MAIL(郵件伺服器)

- 雖然免費的信箱已經非常夠用了，老實說，我們也不建議您架設mail server了。
- 問題是，如果你是一間私人單位的公司，你的公司內傳送的email是具有商業機密或隱私性的，那你還想要交給免費信箱去管理嗎？
- 因此在mail server上面，重要的也是硬碟容量與網路卡速度，在此情境中，也可以將/var目錄獨立出來，並加大容量。

WEB(WWW伺服器)

- WWW伺服器幾乎是所有的網路主機都會安裝的一個功能，因為他除了可以提供Internet的WWW連線之外，很多在網路主機上面的軟體功能(例如某些分析軟體所提供的最終分析結果的畫面)也都使用WWW作為顯示的介面
- CentOS使用的是Apache這套軟體來達成WWW網站的功能，在WWW伺服器上面，如果你還有提供資料庫系統的話，那麼CPU的等級就不能太低，而最重要的則是RAM了！要增加WWW伺服器的效能，通常提升RAM是一個不錯的考量。

DHCP(提供用戶端自動取得IP的功能)

- 如果你是個區域網路管理員，你的區網內共有**20**部以上的電腦給一般員工使用，這些員工假設並沒有電腦網路的維護技能。那你想要讓這些電腦在連上**Internet**時需要手動去設定**IP**還是他可以自動的取得**IP**呢？
- 硬體要求可以不必很高

FTP

- 架設**FTP**去進行網路資料的傳輸
- 對於**FTP**的硬體需求來說，硬碟容量與網路卡好壞相關性較高。

主機硬碟的主要規劃

- 系統對於硬碟的需求跟剛剛提到的主機開放的服務有關，那麼除了這點之外，還有沒有其他的注意事項呢？
- 那就是資料的分類與資料安全性的考量。所謂的『資料安全』並不是指資料被網路cracker所破壞，而是指『當主機系統的硬體出現問題時，你的檔案資料能否安全的保存』之意。
- 網路上有些人在問『我的Linux主機因為跳電的關係，造成不正常的關機，結果導致無法開機，這該如何是好？』呵呵，幸運一點的可以使用fsck來解決硬碟的問題，麻煩一點的可能還需要重新安裝Linux呢！

基本硬碟分割的模式

■ 最簡單的分割方法

- 僅分割出根目錄與記憶體置換空間(/ & swap)即可。

■ 稍微麻煩一點的方式

- 先分析這部主機的未來用途，然後根據用途去分析需要較大容量的目錄，以及讀寫較為頻繁的目錄，將這些重要的目錄分別獨立出來而不與根目錄放在一起，那當這些讀寫較頻繁的磁碟分割槽有問題時，至少不會影響到根目錄的系統資料，那挽救方面就比較容易。

基本硬碟分割的模式

■ CentOS環境中，底下的目錄是比較符合容量大且(或)讀寫頻繁的目錄

■ /boot

■ /

■ /home

■ /usr

■ /var

■ Swap

案例一：家用的小型Linux伺服器，IP分享與檔案分享中心：

■ 提供服務：

- 提供家裡的多部電腦的網路連線分享，所以需要NAT功能。提供家庭成員的資料存放容量，由於家裡使用Windows系統的成員不少，所以建置SAMBA伺服器，提供網芳的網路磁碟功能。

■ 主機硬體配備：

- CPU使用 AMD Athlon 4850e 省電型 CPU
- 記憶體大小為 4GB
- 兩張網路卡，控制晶片為常見的螃蟹卡(Realtek)
- 只有一顆 640GB 的磁碟
- 顯示卡為 CPU 內的內建顯卡 (Radeon HD 3200)
- 安裝完畢後將螢幕,鍵盤,滑鼠,DVD-ROM等配備均移除，僅剩下網路線與電源線。

案例一：家用的小型Linux伺服器，IP分享與檔案分享中心：

■ 硬碟分割：

- 分成 /boot, /, /usr, /var, /tmp等目錄均獨立；
- /home獨立出來，放置到那顆640GB的磁碟，提供給家庭成員存放個人資料；
- 1 GB的Swap；

提供Linux的PC叢集(CLUSTER)電腦群

■ 提供服務

- 提供研究室成員對於模式模擬的軟、硬體平台，主要提供的服務並非網際網路服務，而是研究室內部的研究工作分析。

■ 主機硬體配備

- 利用兩部多核系統處理器 (一部 20核 40緒，一部 12核 24緒)，搭配 10G 網卡組合而成
- 使用內建的顯示卡
- 運算用主機僅兩顆磁碟，儲存用主機提供 8 顆 2TB 磁碟組成的磁碟陣列
- 一部 128GB 記憶體，一部 96GB 記憶體

提供Linux的PC叢集(CLUSTER)電腦群

■ 硬碟分割

- 運算主機方面，整顆磁碟僅分 /boot, / 及 swap 而已
- 儲存主機方面，磁碟陣列分成兩顆磁碟，一顆 100G 給系統用，一顆 12T 給資料用。
- 系統磁碟用的分割為 /boot, /, /home, /tmp, /var, /usr 等分割，資料磁碟全部容量規劃在同一個分割槽而已。

結論

- 案例一是屬於小規模的主機系統，因此只要使用預計被淘汰的配備即可進行主機的架設！唯一可能需要購買的大概是網路卡吧！
- 案例二中，由於需要大量的數值運算，且運算結果的資料非常的龐大，因此就需要比較大的磁碟容量與較佳的網路系統了。

General

Manufacturer:

IBM

Model:

IBM System x3650 M4 HD: -...

CPU Cores:

16 CPUs x 2.599 GHz

Processor Type:

Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz

License:

VMware vSphere 5 Enterprise Plus - Licensed for 2 physic...

Processor Sockets:

2

Cores per Socket:

8

Logical Processors:

32

Hyperthreading:

Active

Number of NICs:

5

Resources

CPU usage: 192 MHz

Capacity

16 x 2.599 GHz

Memory usage: 24763.00 MB

Capacity

155459.20 MB













Storage	Status	Drive Type
<div><div></div>DS1813-25T-NFS-...</div>	<div><div></div>Normal</div>	Unknown
<div><div></div>DS1815-21T-NFS-...</div>	<div><div></div>Normal</div>	Unknown
<div><div></div>ESX-18.cc.s1</div>	<div><div></div>Normal</div>	Non-SSD

<

>

Network	Type
---------	------

General	
Manufacturer:	Dell Inc.
Model:	PowerEdge R720
CPU Cores:	12 CPUs x 1.999 GHz
Processor Type:	Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz
License:	VMware vSphere 5 Enterprise Plus - Licensed for unlimited...
Processor Sockets:	2
Cores per Socket:	6
Logical Processors:	24
Hyperthreading:	Active
Number of NICs:	4

Resources		
CPU usage: 1732 MHz	<div><div></div></div>	Capacity 12 x 1.999 GHz
Memory usage: 50282.00 MB	<div><div></div></div>	Capacity 90066.48 MB
Storage	Status	Drive Type
 DS1812-19T-NFS-...	 Normal	Unknown
 DS1813-25T-NFS-...	 Normal	Unknown
 DS1815-21T-NFS-...	 Normal	Unknown
 EMC-FC-2.1T	 Normal	Non-SSD
 EMC-FC-825G	 Normal	Non-SSD
 EMC-SATA-10T	 Normal	Non-SSD



RAID 介紹



RAID簡介

- 容錯式廉價磁碟陣列 『 Redundant Arrays of Inexpensive Disks, RAID 』。
- RAID 可以透過一個技術(軟體或硬體)，將多個較小的磁碟整合成為一個較大的磁碟裝置。

RAID狀態

RAID狀態	解釋
RAID-0	等量模式, stripe 優點:效能最佳。 缺點:只要有任一磁碟損毀，在 RAID 上面的所有資料都會遺失而無法讀取。
RAID-1	映射模式, mirror 優點:完整備份 缺點:寫入效能不佳
RAID 0+1	Stripe+mirror 優缺點:具有 RAID 0 的優點，所以效能得以提升，具有 RAID 1 的優點，所以資料得以備份。但是也由於 RAID 1 的缺點，所以總容量會少一半用來做為備份

RAID狀態

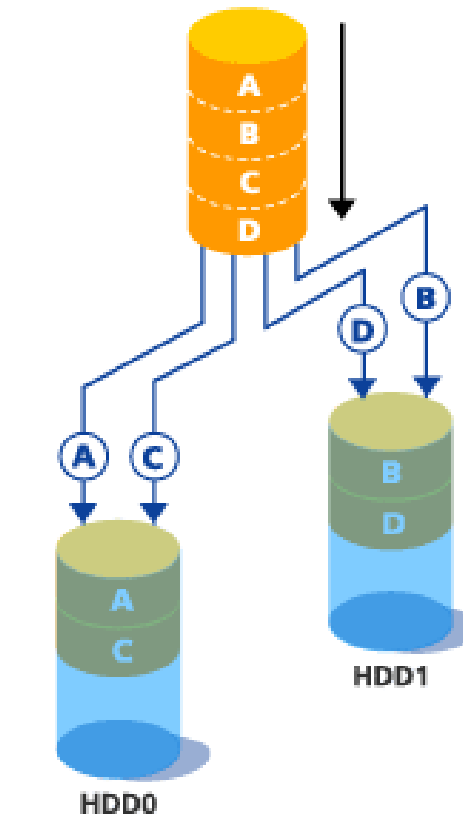
RAID狀態	解釋
RAID 5	效能與資料備份的均衡考量 需要三顆以上磁碟才能夠組成，允許一顆故障。
	每個循環的寫入過程中，在每顆磁碟還加入一個同位檢查資料 (Parity)，這個資料會記錄其他磁碟的備份資料，用於當有磁碟損毀時的救援。
RAID 6	增加了第二個獨立的奇偶校驗信息塊，可靠度高。 需要四顆以上磁碟才能夠組成，允許兩顆故障。
Spare Disk	預備磁碟

RAID 0

figure from: <http://storage-system.fujitsu.com/jp/term/raid/>

- RAID 0: Block-level striping without parity or mirroring
 - It has no (or zero) redundancy.
 - It provides improved performance and additional storage
 - It has no fault tolerance. Any drive failure destroys the array, and the likelihood of failure increases with more drives in the array.

CPUからABCDというデータの書き込み指示

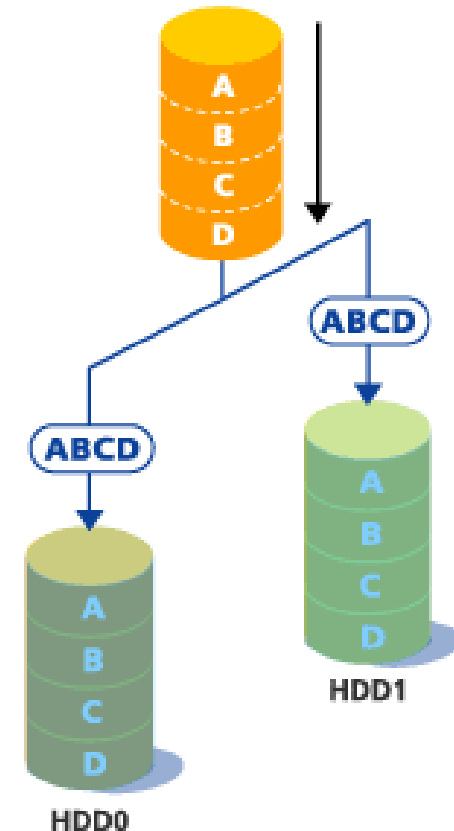


複数ディスクへ分散して書き込む
ストライピング処理を行う

RAID 1

- RAID 1: Mirroring without parity or striping
 - Data is **written identically to two drives**, thereby producing a "mirrored set";
 - A read request is serviced by one of the two drives containing with least seek time plus rotational latency.
 - A write request updates the stripes of both drives. The write performance depends on the slower of the two.
 - At least two drives are required to constitute such an array.
 - The array continues to operate as long as at least one drive is functioning.
- Space efficiency
 - $N / 2$
- Fault tolerance
 - 1

CPUからABCDというデータの書き込み指示

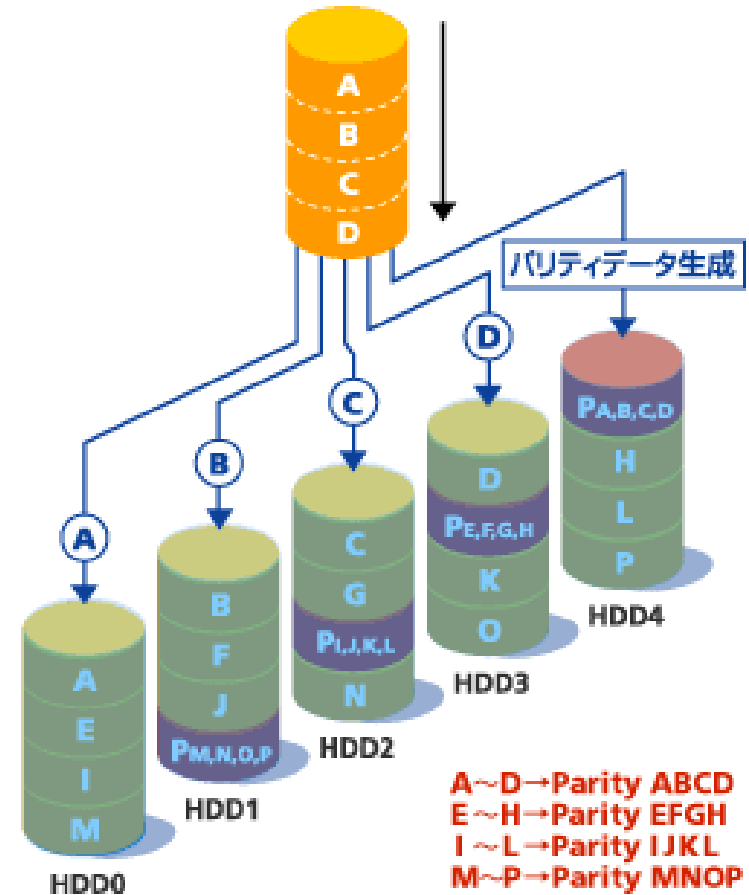


データを2台のディスクに同時に書き込む

RAID 5

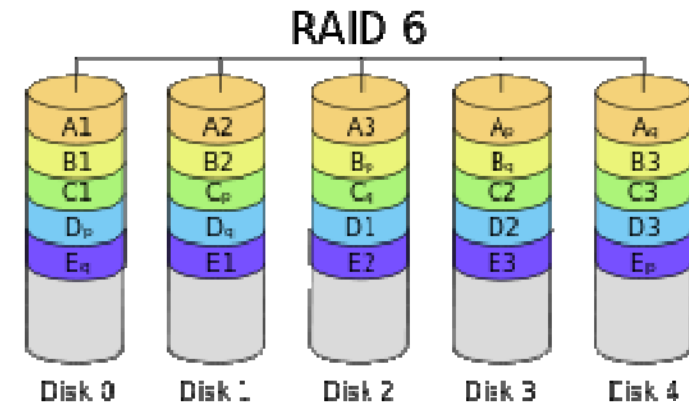
- RAID5: Block-level striping with distributed parity
 - distributes parity on different disk
 - requires at least 3 disks
- Space efficiency
 - N-1
- Fault tolerance
 - 1

CPUからABCDというデータの書き込み指示



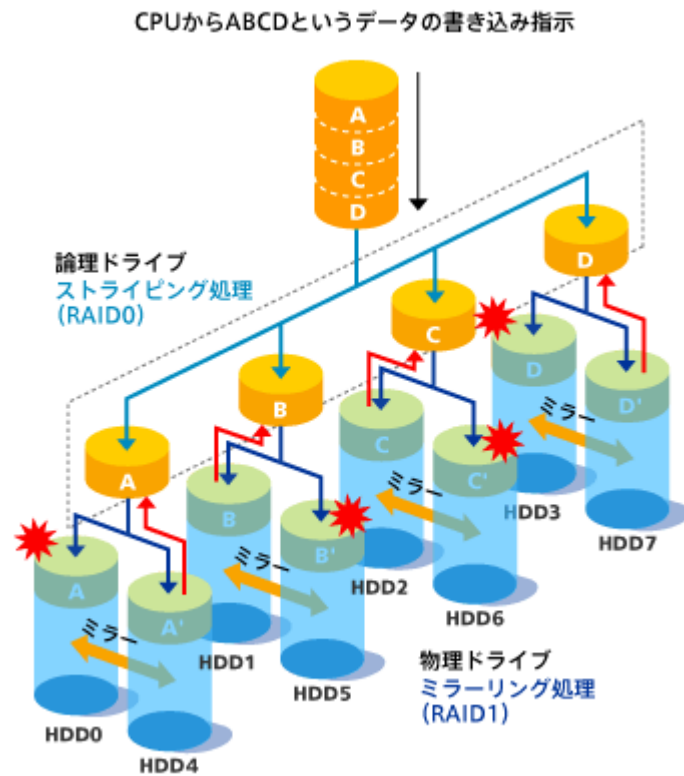
RAID 6

- RAID6: Block-level striping with two distributed parities in two different disks.
 - distributes two parities on different two disks.
 - requires at least 4 disks
- Space efficiency
 - $N-2$
- Fault tolerance
 - 2

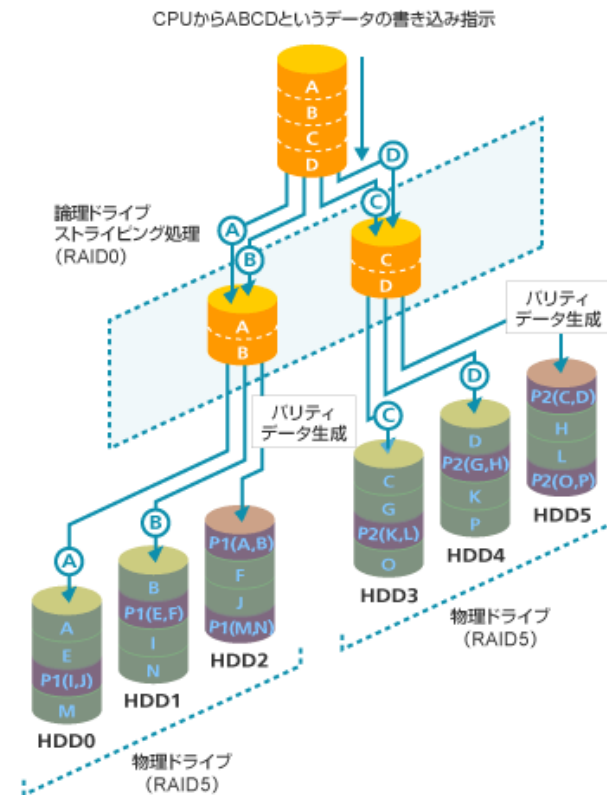


RAID 1+0 / RAID 5+0

- RAID 1+0
- RAID1(mirror) + Stripe



- RAID 5+0
- RAID5(parity) + Stripe



RAID Level Comparison

RAID level	Reliability	Write Performance	Space efficiency
RAID 0	×	○	2/2
RAID 1	◎	○	1/2
RAID 1+0	◎	○	2/4
RAID 5	○	△	2/3
RAID 5+0	○	○	4/6

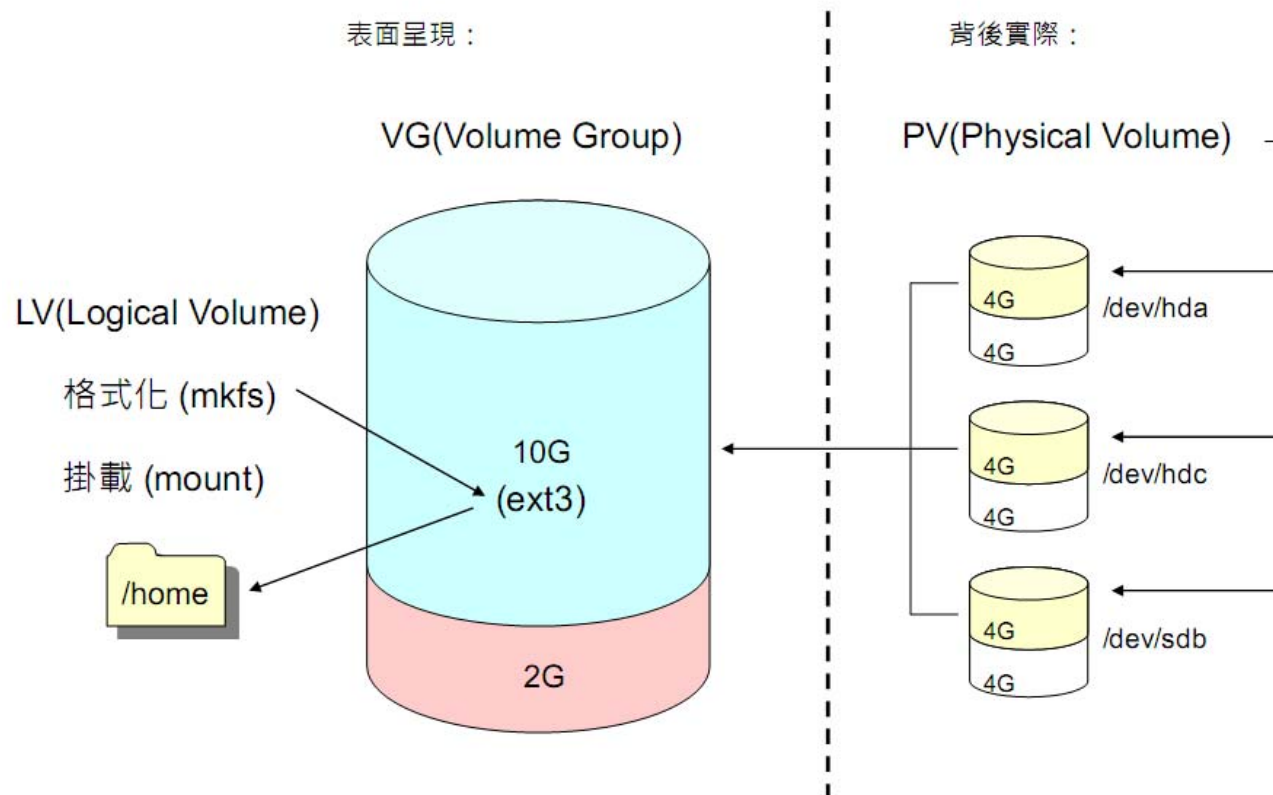
LVM 介紹



LVM介紹

- LVM的全名邏輯磁卷管理,是以磁卷(Volume)為單位,捨棄傳統磁碟以分割(Partition)為磁碟的單位。
- LVM 的重點在於『可以彈性的調整 filesystem 的容量！』而並非在於效能與資料保全上面。

LVM架構



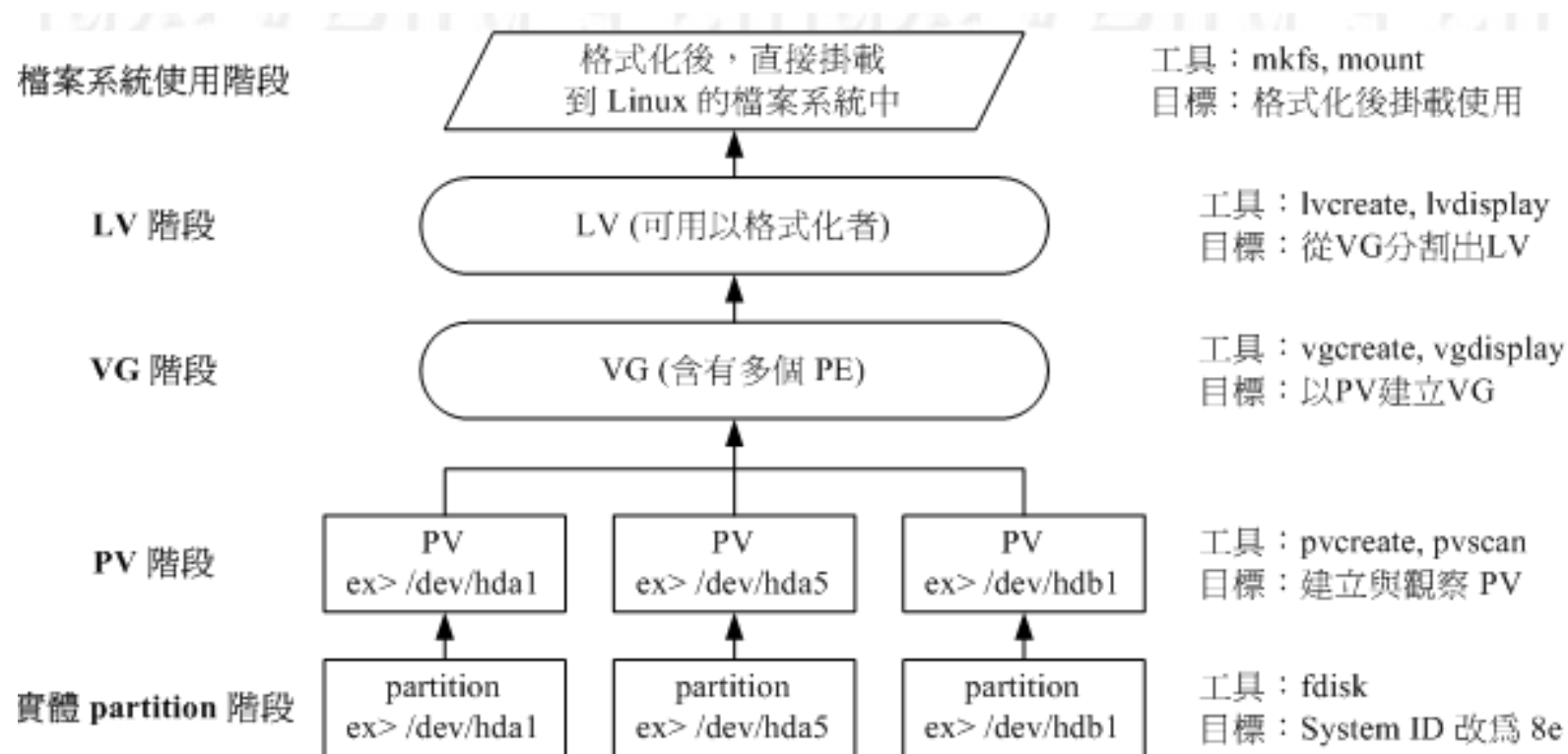
LVM的三個元件

- 實體磁卷PV:(Physical Volume)磁碟分割區;System ID必須標示為8e。
- 磁卷群組VG:(Volume Group)把多割磁碟分割區(實體磁卷)建立成一個磁卷群組。
- 邏輯磁卷LV:(Logic Volume)把邏輯磁卷當作原本的分割區使用。

LVM管理工具

	實體磁卷	磁卷群組	虛擬磁卷
掃描	pvscan	vgscan	lvscan
檢視	pvdisplay	vgdisplay	lvdisplay
新建	pvcreate	vgcreate	lvcreate
移除	pvremove	vgremove	vgremove
放大		vgextend	lvextend
縮小		vgreduce	lvreduce

LVM流程圖



Linux File Systems: ext2 vs ext3 vs ext4



EXT2

- Ext 2 (second extended file system) 檔案系統誕生於西元 1993 年，是為了改善既有的 Ext 檔案系統而設計的，以下是這個檔案系統的特色：
 - 沒有日誌 (journaling) 功能。
 - 因為沒有日誌功能，Ext 2 檔案系統比較適合用於 flash 的儲存設備或一般 USB 隨身碟。
 - 磁碟容量最大可以支援到 32 TB。
 - 單一檔案最大可以支援到 2 TB。

EXT3

- Ext 3 檔案系統誕生於西元 2001 年，顧名思義就是 Ext 2 的下一版，Linux 的版本從 Kernel 2.4.15 開始支援這個檔案系統，其特色如下：
 - 加入日誌功能，日誌功能是在硬碟中規劃出一個區塊，專門用於記錄資料寫入與修改的動作，如果在硬碟寫入的過程發生問題，可以藉由日誌的紀錄加速硬碟的修復，日誌的記錄方式可分為三種：
 - Journal：記錄 Metadata 與內容。
 - Ordered：只記錄 Metadata，在內容寫入磁碟之後，記錄 Metadata，此為預設選項。
 - Writeback：只記錄 Metadata，在內容寫入磁碟之前或之後，記錄 Metadata。
 - 磁碟容量最大可以支援到 32 TB。
 - 單一檔案最大可以支援到 2 TB。
 - Ext 2 的檔案格式可以直接轉換成 Ext 3，不需要經過額外備份與還原的動作。

EXT4

- Ext 4 檔案系統是 Ext 3 的下一版，誕生於西元 2008 年，Linux 的版本從 Kernel 2.6.19 開始支援，特色如下：
 - 支援大容量的磁碟與單一檔案：
 - 磁碟容量最大可以支援到 1 EB (1 EB = 1024 PB , 1 PB = 1024 TB) 。
 - 單一檔案最大可以支援到 16 TB 。
 - 單一目錄最多可以存放 64,000 個子目錄 (在 Ext 3 只能存放 32,000 個) 。
 - 相容於 Ext 3 檔案格式，可以直接以 Ext 4 的檔案格式掛載 (mount) Ext 3 檔案系統的磁碟。
 - 提供將日誌功能關閉的選項。
 - 其餘 Ext 4 新功能有：multiblock allocation、delayed allocation、journal checksum、fast fsck 等，這些都是改善 Ext 3 效率與可靠性的功能。

EXT4

- Linux Kernel自2.6.28開始正式支援新的檔系統Ext4。Ext4是Ext3的改版，修改了Ext3中部份重要的結構，而不像Ext3對Ext2那樣，只是增加了一個日誌功能而已。Ext4可以提供更加的性能和可靠度，還有更為豐富的功能：
- 與Ext3相容
 - 執行若干條命令，就能從 Ext3 線上遷移到 Ext4，而無須重新格式化磁片或重新安裝系統。原有 Ext3資料結構照樣保留，Ext4 作用於新資料，當然，整個檔案系統因此也就獲得了 Ext4 所支援的更大容量。
- 更大的檔案系統和更大的文件。
 - 較之 Ext3 目前所支持的最大 16TB 檔案系統和最大 2TB 文件，Ext4 分別支持 1EB (1,048,576TB，1EB=1024PB，1PB=1024TB) 的檔案系統，以及 16TB 的文件。

■ 無限數量的子目錄。

■ Ext3 目前只支持 32,000 個子目錄，而 Ext4 支持無限數量的子目錄。

■ Extents。

■ Ext3 採用間接塊映射，當操作大檔時，效率極其低下。比如一個 100MB 大小的文件，在 Ext3 中要建立 25,600 個數據塊（每個資料塊大小為 4KB）的映射表。而 Ext4 引入了現代檔案系統中流行的 extents 概念，每個 extent 為一組連續的資料塊，上述檔則表示為“該檔資料保存在接下來的 25,600 個資料塊中”，提高了不少效率。

■ 多塊分配。

■ 當寫入資料到 Ext3 檔案系統中時，Ext3 的資料塊分配器每次只能分配一個 4KB 的資料塊，寫一個 100MB 檔就要調用 25,600 次資料塊分配器，而 Ext4 的多塊分配器“multiblock allocator”（mballoc）支持一次調用分配多個資料塊。

■ 延遲分配

- **Ext3** 的資料塊分配策略是儘快分配，而 **Ext4** 和其它現代檔作業系統的策略是盡可能地延遲分配，直到檔在 **cache** 中寫完才開始分配資料塊並寫入磁片，這樣就能優化整個檔的資料塊分配，與前兩種特性搭配起來可以顯著提升性能。

■ 快速 fsck

- 以前執行 **fsck** 第一步就會很慢，因為它要檢查所有的 **inode**，現在 **Ext4** 給每個組的 **inode** 表中都添加了一份未使用 **inode** 的列表，今後 **fsck Ext4** 檔案系統就可以跳過它們而只去檢查那些在用的 **inode** 了。

■ 日誌校驗

- 日誌是最常用的部分，也極易導致磁片硬體故障，而從損壞的日誌中恢復資料會導致更多的資料損壞。**Ext4** 的日誌校驗功能可以很方便地判斷日誌資料是否損壞，而且它將 **Ext3** 的兩階段日誌機制合併成一個階段，在增加安全性的同時提高了性能。

■ “無日誌”（No Journaling）模式

- 日誌總歸有一些開銷，**Ext4** 允許關閉日誌，以便某些有特殊需求的使用者可以借此提升性能。

■ 線上磁碟重組。

- 儘管延遲分配、多塊分配和 **extents** 能有效減少檔案系統碎片，但碎片還是不可避免會產生。**Ext4** 支援線上磁碟重組，並將提供**e4defrag** 工具進行個別檔或整個檔案系統的磁碟重組。

■ inode相關特性

- **Ext4** 支持更大的 **inode**，較之 **Ext3** 默認的 **inode** 大小 128 位元組，**Ext4** 為了在 **inode** 中容納更多的擴展屬性（如納秒時間戳記或 **inode** 版本），默認 **inode** 大小為 256 位元組。**Ext4** 還支持快速擴展屬性（**fast extended attributes**）和 **inode** 保留（**inodes reservation**）。

■ 持久預分配（Persistent preallocation）

- **P2P**軟件為了保證下載檔案有足夠的空間存放，常常會預先創建一個與所下載檔案大小相同的空檔，以免未來的數小時或數天之內磁碟空間不足導致下載失敗。**Ext4**在檔案系統層面實現了持久預分配並提供相應的 **API**（**libc** 中的**posix_fallocate()**），比應用軟體自己實現更有效率。

■ 默認啟用barrier

- 磁片上配有內部緩存，以便重新調整批量資料的寫操作順序，優化寫入性能，因此檔案系統必須在日誌資料寫入磁片之後才能寫 **commit** 記錄，若 **commit** 記錄寫入在先，而日誌有可能損壞，那麼就會影響資料完整性。**Ext4** 默認啟用 **barrier**，只有當 **barrier** 之前的資料全部寫入磁片，才能寫 **barrier** 之後的資料。（可通過 "**mount -o barrier=0**" 命令禁用該特性。）