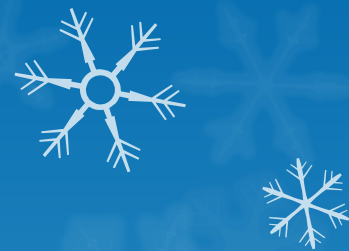


虛擬化於教學環境的應用

崑山科技大學-資訊傳播系

蔡德明(鳥哥) dmtsai@mail.ksu.edu.tw

2014/05/25



什麼是虛擬化？

- 虛擬化技術：
 - 透過虛擬化技術，在一部實體主機上面模擬出多部邏輯上完全獨立的電腦系統(機器)
 - 所模擬出來的硬體我們稱為『虛擬機器』
 - CPU
 - 記憶體
 - 硬碟
 - 網路卡
 - USB
 - 音效
 - 其他的其他...

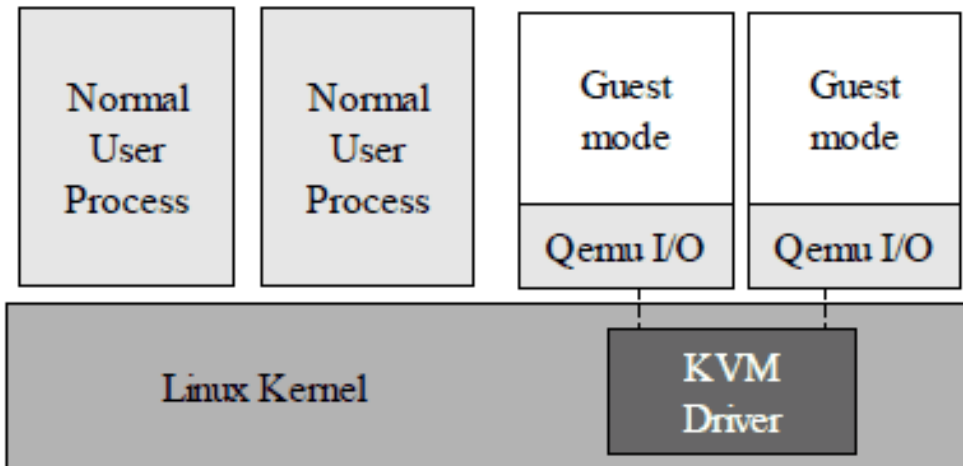
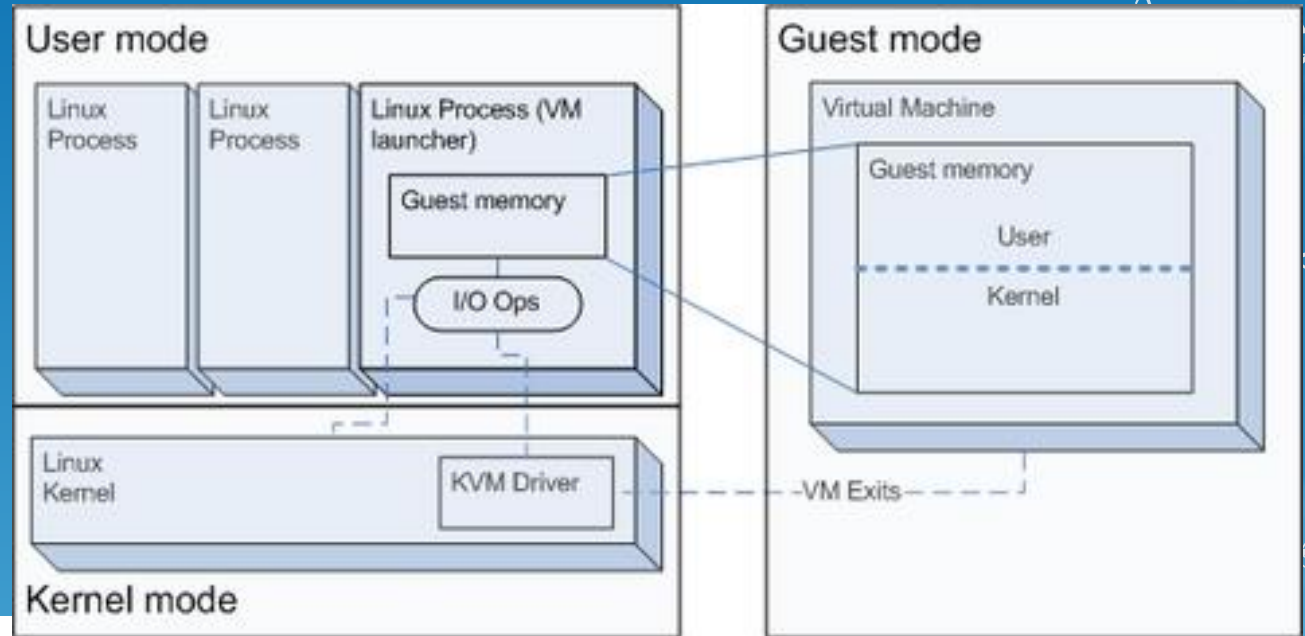
用來模擬虛擬化的軟體

- Linux base
 - KVM (kernel-base Virtual Machine)
 - Xen
- Windows
 - Hyper-V
- Others
 - VirtualBox
- 重點就是在一部硬體上面模擬出多個『假的硬體』

虛擬化技術示意

- 名詞解釋：
 - 實體機器的系統： Host OS
 - 虛擬機器的系統： Guest OS 或 VM
- KVM主要的功能：模擬 CPU、主記憶體等主要的系統單元，但輸入輸出、磁碟等，則可能交由第三方軟協助：
 - qemu：週邊界面如磁碟的虛擬軟體
 - libvirt：管理界面，方便控制虛擬機器

虛擬化技術示意



那...我能用虛擬化幹啥東西做啥事情

一鳥哥專題生的應用

- 實體機器：
 - Intel i7 3770 CPU (8 threads)
 - 64GB RAM
 - 2 Network Interface Card (NIC)
 - 4 x 2TB HDD
- 虛擬機設定：
 - 1, 2, 4 CPU
 - 1, 2, 4 RAM
 - 100GB HDD (用檔案模擬成硬碟)
 - 2 NIC
 - 最多可同時啟動 40 個 VM 以上存在，畢業專題時提供 10VM 展覽

鳥哥在 server 上的應用

- 實體機器：
 - Intel E5 1620 v2 CPU (8 threads)
 - 32GB RAM
 - 2 Network Interface Card (NIC)
 - 5 x 2TB HDD (RAID card)
- 虛擬機設定 (就是鳥站耶!)：
 - 8 core CPU
 - 16GB RAM
 - 500GB HDD (用檔案模擬成硬碟)
 - 實做某些軟體、未來移機與備份都很容易

鳥哥在學術上的應用

- 實體機器：
 - 2 x Intel E5 2620 v2 CPU (total 24 threads)
 - 96GB RAM
 - 2 Network Interface Card (NIC)
 - 8 x 2TB HDD (RAID card)
- 虛擬機設定 (air quality modeling)：
 - 幾乎全部的資源都給這系統
 - 不需要對外的 public IP，亦可不須網路
 - 同樣在備份或安裝額外軟體時，比較安心(可以重來的
 ㄉ！)
 - 系統效能其實與實機差不多

鳥哥在教學上的應用

- 一個 file server 共用 (i7 系統)
- 四台 KVM host (i7 系統)
- KVM hosts 透過網路連結 file server
 - 重點在取得虛擬機器的磁碟檔案
 - 四台 KVM hosts 各 32GB 記憶體
 - 可同時啟動 60 個以上的 VM (各給 1.5GB 記憶體)
 - 每位同學可擁有多個磁碟檔案
 - 每個磁碟檔案可擁有不同的作業系統(依課程而定)

雲端虛擬化的應用—話當年

- 沒Linux要教Linux
 - PIII 1GB RAM
 - 一個 Linux server 一起玩
- 沒Linux要教Linux
 - 4core+4GB
 - 開30VM
- 有 Linux 要考試
 - 實體系統環境真難搞～

現在開始來談談教學應用面吧！！

校園虛擬化最原始的想法

- 目的
 - 可以讓學生回家去練習 Linux
 - 才不會每次考試都後，**都說沒有 Linux 可以練習**
- 衍生想法
 - 學生在校的操作環境與在家不同，常常無法複習
 - 很多學校的軟體學生無法帶回家使用
 - 如果能夠讓學生在上面考試就更好了! ^_^
 - 再也不用復原實體系統環境

虛擬化前一先思考教室環境是怎樣

- 通常有超過 30 部以上的電腦啊~
- 作業系統與資料並不重要，砍掉重練沒關係
 - 一定要能夠開機給學生使用
- 能夠越快速復原越好
 - 因為常常需要砍砍殺殺~
- 網路能用就好，但最好能可以管理開放與否
- 最好是每一部電腦都能夠有學生自己的環境
 - 座位就不用重調~

虛擬化前一思考學生操作的行為

- 進教室第一件事情就是按下 Power
 - 管他之前有沒有開機
- 使用過程中，給我滑鼠直接按下螢幕上面的登入，就讓我使用電腦啊！
- 離開教室前，如果嫌關機太慢，按下Power四秒鐘不就好了？
- 能不能盡量不要讓我敲指令啊？我好怕...
- 能不能讓我回家還能擁有跟學校一樣的作業系統環境，這樣才能練習啊！

虛擬化前—思考老師操作的行為

- 能不能讓我自己安裝作業系統(image)
- 能不能讓我管理自己的教學環境?
- 能不能讓我安裝自己所需要的軟體?
- 能不能讓我快速的將自己的環境複製給學生?
- 能不能讓我管理學生的網路?
- 能不能讓我考前可以快速復原考試環境?
- 能不能讓我在家裡管理教室的images?

虛擬化前一思考管理者的行為

- 能夠越快復原系統越好
- 能夠讓老師操作的方便最好
- 能夠讓老師自己安裝作業系統與軟體最好
- 能夠讓老師管理學生的電腦最好
- 能夠滿足不同老師的需求的電腦環境最好
 - 影像處理是誰要?鳥哥就是要搞死人的Linux...
 - 就是，多重開機開到死那種環境啦!
- 最好不要來煩我最好!

所以需要考慮的部分有？

- 硬體架構設計；
- VM 效能優化的設計；
- 操作介面設計；
- 一般用戶操作功能設計；
- 管理方面之功能設計；

硬體架構部分：

- 考慮要將 VM 放置於區網內，這樣模擬電腦教室內部環境較佳，比較容易控制網路操作的行為
 - 需要一部主控電腦，除了管理使用者與系統整體資料庫之外，還充當整個區網的防火牆
 - 需要多部 KVM host，可以提供 VM 運算功能，同時使用多張網卡連接到 file server，以增加網路頻寬，讓 VM 的虛擬磁碟效能提升
 - 需要一部可以提供快速的磁碟系統的 file server，亦須提供大量的頻寬。
- 由於 KVM host 均連線到 file server 取得 VM 所需磁碟，因此 KVM 的數量是可調整的，依據 VM 所需而變。理論上，硬體配置會較具有彈性。

需要思考的地方

- KVM host 要有幾部？效能如何？網路連接方式呢？
 - 沒有用到的淘汰電腦，能不能有利用的空間？
- File server 與 KVM host 連線要用什麼協定？
 - iSCSI
 - NFS
- File Server 的 filesystem 與 I/O 要如何設計？
 - LVM
 - RAID (level ??)

硬體架構-廉價硬體的應用

- 廉價硬體的應用方面：
 - 想透過十部左右的雙核心 (1.86GHz) 機器組成cluster架構來提供很多記憶體的KVM host虛擬化環境
 - 想透過一部雙核心主機搭配陣列卡與五顆以上硬碟來提供共享的檔案系統(提供給guest使用的硬碟)
 - 上述主機全部都用一條GB線路連上同一個 switch
 - 結果：
 - 很耗電、雙核系統效率真糟糕！
 - File server光是一條GB的網卡根本無法負荷龐大的流量需求

硬體架構-主機系統確定，但是網路？

- 所以，硬體的需求應該要：
 - 主機系統部份：至少 i7 以上等級的硬體系統比較妥當啊！
 - CPU 四核八緒以上
 - 記憶體插爆為止！（現在至少可上 32G 以上啊！）
 - 網路系統部份：一條 GB 給 file server 是不夠用的！
 - 要用 bonding 嗎？
 - 要用 10G 網卡嗎？
 - 要考慮到 switch 嗎？

硬體架構-還是 10 G好！

- 透過 bonding 嗎？
 - 傳統的 bonding 針對單一 IP 來源，僅可使用到一個 NIC 收發
 - 因此 file server \leftrightarrow KVM host 永遠只有一張 NIC 在動作！
- 所以只用 GB 的作法
 - 透過不同的 LAN 提供網段的連線
 - file server \leftrightarrow KVM host 終於可以擁有數張 NIC 的流量了。
 - 但 KVM host 數量越多，file server 的網卡數量就得要跟著增加
- 10G 網卡
 - 超棒！但是 switch 真是貴！
 - 用網卡對接法處理，省略 switch 的龐大負擔
 - 但是 file server 的 pcie 的 8x 插槽就要足夠才行

網路連線協定選擇：

- iSCSI好像很棒棒？
 - iSCSI 掛載在各個 KVM 上頭，看起來就是一個單純的裝置
 - 但是由於每個 KVM 的 iSCSI 都具有獨立的檔案系統 metadata 控制，因此同一個 VM 在不同的 KVM 上面運作後，將會造成 filesystem 損毀~
- 分散式檔案系統，例如 pNFS
 - 似乎很棒棒！讓 NFS 好像磁碟陣列一樣
 - 但機制方面鳥哥還是很擔心！而且俺的主機數量不夠啊....單純用 RAID 系統來處理，說不定還比較好！
- 所以，老的還是很優秀：
 - NFS 搭配 file server 的 RAID 來處理吧！

LVM 還是 qemu 的 qed/qcow2 呢

- LVM + iSCSI 的嚴重問題：
 - LVM 必須是系統上面的裝置啊！
 - 透過 file server 來提供，則只能使用 iSCSI ！肯定掛點！
- LVM + NFS
 - 在 LVM 裝置上面建立大檔案再處理成 VM 可讀寫格式
 - 如此則可透過 NFS 來傳輸檔案囉！但是....
 - LVM 的快照都獨立，因此讀取 original 磁碟時，將不會進行快取
 - 鳥哥層偵測到 2.5GBytes/s 的讀寫速度....
- 所以，LVM 搭配快照，就被捨棄了
 - Netman 師父...我對不起您啊！

Qemu 的 qcow2/qed

- VM 檔案系統最重要的任務：
 - 需要可以快速大量建立相同的磁碟
 - 所以原本考量 LVM 的快照功能啊！
 - 但是 qemu 的 `qemu-img` 指令，可以提供 `qcow2/qed` 這種格式的 `backing file` 功能，此功能與快照相仿！
- Backing file 的好處與缺點
 - Backing file 與 original 檔案間並不會持續產生連結
 - 只有在 backing file 要讀寫時，才會對照 original 磁碟
 - 因為是單純的檔案系統，因此 original 可以被快取喔！節省很多磁碟 I/O 的問題！對效能的影響相當大
 - 因為是 filesystem 磁碟而非裝置層級，因此設計 `KVM` \leftrightarrow `file server` 時，必須將兩者通通掛載起來才行！
- 總的來說，單一選項，沒得選擇啊～

硬體配置的變更-第一代的設計

- 1部 core 2 login server 搭配兩張網卡給內外連線使用
- 3部 core i7 2600 + 32 GB 記憶體의 KVM host，搭配4port 的GB網卡
- 1部 雙 Opteron 雙核 CPU + 舊式 RAID card 2T * 8 組成 RAID 5 提供磁碟 + 6 port GB 網卡
 - 除 login server 有一張網卡對外，其餘所有網卡均連線於同一個 switch/hub 上面。
 - KVM host 到 file server 透過 bonding mode 6 連線，所有的網卡均綁定於同一個 bonding 介面，所有的 IP 與 bridge 均透過這個 bonding 介面所提供

硬體配置的變更-第一代設計的問題

- Bonding 的問題：
 - 頻寬：KVM host 與 file server 之間僅透過一個 bonding 介面連線，由於 bonding 僅能針對單一 IP 提供一張實體網卡的頻寬，因此總頻寬並沒有增加
 - 對 VM 的問題：VM 的 bridge 如果設定在 bonding 上，由於我們提供的是會自動學習 ARP 的 mode 6，此種模式架設在 VM 上時，他的 MAC 與實體對應將會經常性的變動，導致 VM 之間無法順利連線
- 問題總結：效能瓶頸卡在網路上~需要重新設計

硬體配置的變更-第一代設計的改善

- 透過多層迴路設計：
 - 每張 KVM host 網卡直接連線到 File server 的網卡，設計為一個獨立區網(ex> 10.1.0.0/16, 10.2.0.0/16)，因此若有兩對網卡 (四張)，則對 VM 來說，將可使用到兩個 GB 的網路頻寬，越多迴路效能理論上越佳。
- 本系統的安全性問題：
 - KVM host 提供一張網卡給 VM 使用，該網卡與上述迴路界面皆於同一個 switch 上面
 - 若 VM 使用者技術較高，設定到錯誤的 IP 使與 KVM host 或與 file server 相同，甚至嘗試入侵內部伺服器，將會導致整體系統的損毀，相當嚴重。

硬體配置的變更-第一代設計的改善

- 解決安全性問題的手段：
 - 透過分層，將 KVM host 分享給 VM 使用的 bridge 網卡連接到另一個獨立的 switch 上頭，同時讓 Login server 提供另一張網卡到這個 switch 上，KVM host 的 bridge 網卡並不給予 IP，此時每部 VM 感覺上，將在一個獨立的網路區段，甚至可以沒有網路參數喔。
- 依舊產生的問題：
 - 理論 VM 的磁碟頻寬為網路頻寬，應可達 600Mbytes/s，但一般常用情況，磁碟讀寫速度僅有大約 200~300Mbytes/s，情況很糟。

硬體配置的變更-第二代的設計

- 僅修改了 File server 的硬體，使用較新的 CPU 並使用伺服器等級的主機板，提供了 32GB 的記憶體，使用了共 10 port 的 GB 網卡，同時使用 Dell H700 磁碟陣列，並使用了速度較快的 RAID0 作為磁碟提供者。
- 理論速度：
 - 網路共三迴路，理論上可以到達約 900Mbytes/s 的傳輸量
 - 磁碟陣列假設每顆硬碟 120bytes/s 時，總共亦可達約 $8 * 120 = 1\text{Gbytes/s}$ 的傳輸量才對
- 問題：
 - 效能依舊不彰，最快速度大約僅能達到 400~500Mbytes/s 的速度
 - 思考，恐怕是傳統硬碟的延遲效應所致吧。

硬體配置的變更-第二代設計的改良

- 針對 RAID，換插兩顆 SSD 固態硬碟，以提供較快速的讀寫效能，但是在 SSD 的腳色方面，卻有部分問題：
 - Dell H700 具有 SSD 第二層快取功能，原本的想法是讓兩顆 SSD 作第二層快取，如此一來，未來 VM 的資料將通通由 SSD 提供，效能理論上可以改善不少
 - 結果：由於 H700 內部的快取功能是記憶常用資料，每部 VM 的資料都是獨立的，因此沒有進入到快取的運算機制上，整體效能反而降低非常多。
- 進一步改善：
 - 將磁碟陣列分為兩部分，SSD 組成原生硬碟放置區，傳統硬碟組成快取磁碟放置區，並修改程式碼，讓 LVM 與此設計搭配
 - 效能提升非常多!全部 60 個 windows 7 開機時，瞬間磁碟存取效能可達 1GBytes/s，網路、磁碟的理論極速達成。

硬體配置的變更-第三代的設計

- 針對網路：
 - KVM 與 File server 均換插雙 port 10G 網卡，其中 File server 更安插了 3 張 10G 網卡，已達成雙迴路共六對的 10G 傳輸
- 針對磁碟系統：
 - 使用 5 顆 SSD 組成原生磁碟放置區，使用 5 顆 2TB 硬碟安插於主機板南橋(software raid)，使用於快照磁碟區，混合使用。
- 檔案管理此時依舊使用 LVM + NFS 喔！
 - 效能最高可達 2.5 Gbytes/s 的磁碟讀寫率，VM 內部的磁碟速度使用 dd 測試時，可高達 300~400 Mbytes/s 的速度，效能相當好
 - 但很奇怪的是，過了兩三週，同樣的磁碟測試卻發現效能變糟了！而且是很糟很糟！
 - 因為快照量變大了，所以讀寫 original 的機會變少，且兩者間對照的時間延長，故效能變得很糟糕

硬體配置的變更-第三代設計的改良

- 針對檔案系統處理方式：
 - 開始透過 qemu 的 qed 對應 backing file 進行設計
 - 建立與刪除類似快照的 backing file 速度超快！比 LVM 快很多
- 針對磁碟系統：
 - 兩張RAID card。
 - SSD 5 顆 raid 5 放 original，有保全，速度 OK！可被快取
 - HDD 8 顆 raid 10 放 backing file，有保全，讀/寫速度效能 OK！
- 結果：
 - 檔案系統管理變得很簡單！但 original 不可隨意修改
 - 效能瓶頸可能變成 KVM 上面而非 file server 上了
 - VM 磁碟用的越久，與 original 關係越低，此時對 HDD 所在的 RAID 壓力越大！因此效能瓶頸會隨時間而變換位置喔！
 - 同時開啟 60 windows7，只要 2 分鐘內搞定



那原本的 Linux 參數就很棒了嗎？

自己手動調整參數還是需要的

硬體配置與 CentOS 參數搭配

- File server 的磁碟參數設計：

- 預設的磁碟佇列處理使用 cfq 機制，此機制對於磁碟陣列來說，效能相當有問題，因此即使第三代使用 5 顆 SSD 時，一開始的效能竟然僅可達 800Mbytes/s 左右
- 若有磁碟陣列卡，建議使用 noop 機制
- 若有混合式磁碟 (SSD+HDD)，則建議使用 deadline 機制

硬體配置與 CentOS 參數搭配

- File server 的磁碟參數設計：

- for hname in sda
- do
- echo "deadline" > /sys/block/\${hname}/queue/scheduler
- echo 65545 > /sys/block/\${hname}/queue/nr_requests
- echo 0 > /sys/block/\${hname}/queue/rotational
- echo 0 > /sys/block/\${hname}/queue/rq_affinity
- echo 16384 > /sys/block/\${hname}/queue/read_ahead_kb
- done
- for hname in sdb sdc sdd sde sdf sdg
- do
- echo "deadline" > /sys/block/\${hname}/queue/scheduler
- echo 16384 > /sys/block/\${hname}/queue/nr_requests
- echo 8192 > /sys/block/\${hname}/queue/read_ahead_kb
- done

硬體配置與 CentOS 參數搭配

- 網路效能參數設計：

- 預設的核心網路參數並無法使用於高負載的環境中，因此，調整了 `deadline` 之後，有陣子效能還是僅能達到 1.25Gbytes/s 的性能而已
- 參考 IBM 紅皮書與 Red Hat 工程師的 `paper`，了解需要調整 `sysctl.conf` 的內容，才有辦法提升 GB 或 10G 的網卡效能。

硬體配置與 CentOS 參數搭配

- 網路效能參數設計：

- `net.core.rmem_max` = 8388608
- `net.core.wmem_max` = 8388608
- `net.ipv4.tcp_rmem` = 4096 87380 8388608
- `net.ipv4.tcp_wmem` = 4096 65536 8388608
- `net.ipv4.tcp_tw_reuse` = 1
- `net.ipv4.tcp_tw_recycle` = 1
- `net.core.netdev_max_backlog` = 10000

硬體配置與 CentOS 參數搭配

- 網路效能參數設計：
 - for nic in eth11 eth12 eth21 eth22 eth31 eth32
 - do
 - ifconfig \$nic txqueuelen 5000
 - done
- NFS 分享時的 export 參數設計
 - NFS server :
rw,async,no_acl,no_wdelay,insecure,no_root_squash,no_subtree_check
 - NFS client:
rw,intr,bg,hard,vers=4,tcp,noacl,timeo=600,retrans=2

硬體配置與 CentOS 參數搭配

- 但上述的參數設定不見得適合所有的設備
 - 因為所有的設備均不相同，上述參數需要一項一項的設定後測試
 - 所有的項目都是 case by case 的，苦工很辛苦~

效能優化的選擇與設計

虛擬化技術效能的瓶頸

- 對外頻寬
- 內部頻寬
- 檔案系統效能

VM 效能優化的設計

- 遇到的問題：
 - 顯示的效能很糟糕~過去都用 VNC 之故
- 解決方案：
 - 透過 Red hat Spice 機制，可以同時傳輸聲音，且顯示卡共用記憶體可設定到 64M，對於圖形影像影響相當大
 - 唯一的問題是，你的 guest 如果是 windows 的話，要額外安裝 QXL 驅動程式，否則系統顯示的效能依舊很糟糕

VM 效能優化的設計

- 遇到的問題：
 - 改了 spice 機制，在某些 KVM host 上面運作很好，但在某些 KVM host 則很糟糕~
 - 某些 KVM host 的 CPU 明明很不錯，但卻無法安裝 windows 8
- 解決方案：
 - 本系統使用的 XML 檔案都是預先建置的，因此僅能使用最陽春的 CPU 特徵設定 (CPU feature) 來建立
 - 修改成透過 libvirt 去抓到的 CPU feature 取代 XML 內容。但依舊不夠，因為 libvirt 無法抓到所有支援的特徵
 - 透過分析實體 CPU 的 flags 後，再動態加入 XML 裡面

VM 效能優化的設計

- 修改了 XML 內部的 CPU feature 後，產生的影響有：
 - VM 讀取磁碟的次數降低，過去 60 windows VM 同時啟動需要 4 分鐘，且磁碟讀取幾乎定格在 2.5Gbytes/s 上，資源使用很可怕
 - 在此同時，VM 對 KVM host 的 CPU 存取亦很頻繁，造成 KVM host CPU loading 飆高，產生 VM 的效能低落問題
 - 現在修改之後，60 Windows VM 同時啟動僅需 2 分鐘開機完成，且磁碟讀取降低至大約 1.2~1.5Gbytes/s 左右而已
 - 開機完成後，整個 KVM host CPU 就冷卻了! VM 的效能大增。
 - 案例：在同時開啟 60 windows VM 時，其他已啟動的 VM 的運作幾乎不受影響~感覺就像本機前運作一般。

VM 效能優化的設計

- 小結語：
 - VM 設計上，需要注意顯示卡相關設定；
 - VM 設計上，CPU 的 feature 設計，影響非常大！原因是所有 VM 內的指令均須透過這層 feature 的 CPU 微指令集去與 KVM host 溝通，因此連帶的也會影響到顯示卡的相關行為
 - 其他的部分感覺影響較小，包括硬碟與網路使用 virtio 與否，影響不大。
 - 在音效方面，對 windows VM 而言，不同的版本需要不同的音效晶片。32 位元使用 ac97，而 64 位元則需要使用 ich6 才能聽到聲音

The background is a solid blue color. On the right side, there is a vertical column of white snowflake icons of varying sizes and opacities. The largest and most prominent snowflake is at the top right, with several smaller ones below it, some appearing as faint outlines. The snowflakes have a six-pointed, dendritic structure.

操作界面設計的概念

系統的設計概念—操作介面

- 透過網頁提供使用者登入與操作使用
 - 透過資料庫提供學生註冊/資料訂正/連線使用
 - 具有權限設定，分為管理員/教師/學生
 - 透過瀏覽器直接取得連線雲端主機
 - 可以略過不同作業系統的連線問題
 - 登入的是host機器，並非是Guest機器
 - 系統由身分判斷學生的VM，與教室內的實際作業無關
 - 可以使用光碟映像檔，老師可透過瀏覽器直接準備image
 - 可選擇不同的媒體開機，所以可以線上安裝軟體

系統的設計概念—磁碟設計

- 不同的課程可以擁有不同的磁碟
 - 每個老師給予一顆以上的個人擁有的原始碟
 - 該顆原始碟可以讓老師完全掌控
 - 包括自行安裝所有上課所需要的個人化軟體
 - 如果有特殊需求，可有多顆原始碟
 - 上課有上課專用硬碟
 - 考試有考試專用硬碟
 - 老師可以將自己的磁碟派送給課程內的學生
 - 以課程為單位

有網路的話，我們就來操作看看囉！

使用情況

- 全部動作皆在瀏覽器上完成
 - 包括 image 的安裝製作，不用自己製作再上傳
 - 原始硬碟可以對拷，省去自行安裝系統的時間
 - 原始硬碟作為考試環境，上機考可以考的深入一些(每個課程都可以擁有多個硬碟！)
 - 可以直接透過網頁抽換雲端主機的光碟(若教師有提供光碟映像檔的話)
- 學生操作介面簡單
 - 只需要啟動並且取得雲端主機，這是目的。

使用情況

- 目前使用情況：
 - 用來進行 Linux 授課
 - 用來提供學生回家練習
 - 用來作為上機考的環境
 - 再也不需要復原啦！刪掉快照重建即可
 - 再也不需要替學生擔心啦！作錯自己復原再作一次！
 - 再也不需要擔心學生將其他多重系統搞爛咯
 - 壓力測試：提供60位同學同時開機進入考試環境中，幾乎沒有停頓的狀態可以順利考試，就跟實機一樣！

虛擬電腦教室困境



The background is a solid blue color with several white snowflake patterns scattered across it. The snowflakes vary in size and opacity, with some being more prominent than others. The title text is centered horizontally in the lower half of the page.

實體電腦教室的管理

電腦系統的安裝與大量佈署

- 因應教學所需要的環境：
 - 多重作業系統 (windows X2 + Linux + data)
 - 不想要使用還原卡
- 大量佈署方式：
 - 透過國高的 DRBL
- 問題是？
 - DRBL 目前沒有 web 的管理界面
 - 沒有預約復原的功能
- 解決方案：
 - 鳥哥專題組來告訴你怎麼辦！

電腦系統的帳號管理

- 問題：
 - 每個學生都需要自己的帳密、在每部系統上面都可以使用自己的帳密來登入取得自己的工作目錄
 - 不想要使用 Windows 的 AD 啊！
 - 流量好嚇人！帳號管理好恐怖...
- 解決想法：
 - 將全系的所有電腦教室實體網路串接在一起
 - 透過 LACP 加大 switch 間的頻寬
 - 透過 bonding 加大 server 與 client 間的頻寬
 - 使用 LDAP 控管跨平台的帳號（client 用 Pgina）
 - 使用 SAMBA 搭配 windows folder redirection
- 解決方案：有好的學生很重要！ ^_^

多謝聆聽，敬請指教！

