

分散式檔案系統於虛擬化環境下效能比較 -以 MOOSEFS 分散式檔案系統為例

呂泓縉
開南大學資訊管理
研究所
m10107001@mail.knu.edu.tw

賴仲哲
元智大學資訊工程
研究所
s1016026@mail.yzu.edu.tw

葉俊雄
開南大學資訊科技
中心主任
cye@mail.knu.edu.tw

李盛安
開南大學資訊科技中心
雲端服務組組長
shengan@mail.knu.edu.tw

摘要

MooseFS 運作於實體機器架構下，可有效達成資料節點自動備援、後端資料伺服器維護狀態仍可維持資料服務不中斷、快速動態擴展儲存空間等優點，搭配虛擬化技術後，可節省大量實體資料服務節點修復與故障排除時間，更可運用虛擬化技術之各項優點，透過差異化技術快速啟動並擴展資料節點、虛擬化硬碟空間可事先規劃超過實體硬碟儲存空間大小等特性，快速有效設定高可用性的分散式檔案架構。

本研究將 MooseFS 運作於虛擬機器內，並將虛擬機器映像檔儲存於虛擬 MooseFS 內部，透過量測虛擬機器於不同條件下開機的完成時間，由此評估虛擬 MooseFS 分散式檔案系統運作架構與同等架構於實體 MooseFS 檔案服務之效能差異，此結果可作為虛擬化分散式檔案系統回應服務的參考依據。在各種大型資料儲存體需求環境，如虛擬化教室、動態擴展 WEB 網站伺服器、海量資料分析平台等，本架構皆可有效作為新一代資料儲存體的運作參考。

關鍵詞：分散式檔案系統、虛擬化技術、虛擬化分散式檔案系統

Abstract

In physical machine structures, MooseFS can efficiently support high availability of data nodes, consistent data services from back-end servers, and on-demand expansion of storage space. In virtual machines, MooseFS can shorten the time for data node repairing and fail-over. Because the sizes of virtual hard drives can be larger than physical hard drives with proper design, MooseFS can enhance the usability of hard drives using differential techniques to effectively initiate and expand data nodes in distributed.

This study applied MooseFS in virtual machines and stored images of virtual machines in the distributed file structures of virtual MooseFS. By measuring the initiation times of virtual machines under various conditions, we compared the initiation time for similarly structured virtual machines and physical machines. Our results can be used as a reference for the efficiency of response in virtual

distributed file structures. This study can also be used to estimate the requirements of large data storages, such as virtual classrooms, dynamic expansion of WEB servers, and big data analysis platforms..

Keywords: Distributed File System, Virtualization, Virtualization Distributed File System

1. 前言

近年來雲端環境興起，網路設備商採用更高速度傳輸設備，10G 乙太網路儲存設備日益普及，大型雲端基礎服務商開始採用 SAN 儲存架構，企業資料與雲端虛擬主機映像檔案儲存至 SAN 儲存架構，未來 10G 高速網路與 SAN 如果能平價化，對一般民眾資訊利用方式將有機會大幅度的改變，此架構已使輕量級前端大量興起，雲端加上輕量級前端的環境，開始朝向成熟階段。

然而，對一般中小型企業以及研究單位的儲存策略上，SAN 儲存架構仍存在著高投資、低存取利用率、存取架構複雜等成本考量。中小型企業因商業資料機密性的考量，若是要導入雲端虛擬化[1]，目前多採私有雲或混合雲形式來建構本身的資料及計算資源。如果希望有效進行資料的備份，同時應付未來資料挖掘、即時分析變動資料的議題時，往往可能發生資料儲存、備份、大量同時存取等三方面的挑戰，雖然目前已有硬體的解決方案，然大型儲存設備的維護成本對於中小企業仍處於高單價的狀態，同時也代表高額の資料維護與分析費用。

分散式檔案系統[2]的發展日漸成熟，許多軟體套件產品，由過去實驗室獨立發展推廣至社群發展，使用者日漸普及同時相對的應用環境也更多元，目前有許多單位測試與採用 MooseFS 以及 GlusterFS 分散式檔案系統，作為資料中心的前導測試計畫，GlusterFS 於 2010 由 RedHat 收購主導發展成為企業級網路儲存方案，而 MooseFS 由 2008 年發展至今，對於中小型企業的需求環境也日趨成熟。國家高速電腦中心 Ezilla 計畫也採用了 MooseFS 作為虛擬電腦化教室的虛擬機器映像檔案儲存架構[3]，顯示 MooseFS 的架構足以承擔分散式檔案系統的運作架構。

虛擬化環境日益成熟，已有許多虛擬機房的建制案例，顯示虛擬化環境已為多數的企業所接受。虛擬化環境過去給人效能不彰、多工分配不均等印象。近年來虛擬化技術較過去更為成熟，無論在虛

擬網路卡的效能提昇、虛擬機器對實體硬碟的讀寫效率、虛擬機器記憶體管理等議題皆較過去更為有效，且效能較過去已有大幅度的改進，本論文嘗試將 MooseFS 運作於虛擬化機器內部，而虛擬機器的映像檔案儲存於虛擬 MooseFS 的運作架構，透過實際量測虛擬機器運作於於虛擬化 MooseFS 架構的效能，瞭解分散式檔案系統於虛擬化架構運作的可能性，以此作為虛擬化架構分散式檔案系統服務架構於未來提供服務參考依據[4]。

2. 文獻探討

傳統的網路式檔案系統，多以微軟視窗環境之網路芳鄰與 UNIX 之 NFS 架構為主，單點形式網路檔案系統在日漸擴展後，於檔案管理、搜尋與安全認證架構上常會成為資源使用的一大挑戰，如何在不同的網路資源中，快速存取所需的資料，消除單點故障、並具有即時動態擴充存取空間，同時能解決存取效能的問題，是企業應用上最大的挑戰。目前常用的硬體作法是雙節點備援與 RAID 磁碟陣列技術，透過網路備援機制，達成系統故障轉移並利用 RAID 磁碟陣列存取技術，達成資料傳輸並行化的目的，唯此種技術在需要不間斷服務環境下，多需雙倍資源規劃，且後續維護與升級需高額成本。

以下將分三小節討論虛擬分散式檔案系統，第一節為分散式檔案系統的比較與名詞說明，第二節為虛擬化環境介紹，針對虛擬化環境進行說明，第三節說明 MooseFS 虛擬化分散式檔案系統運作架構的優缺點。

2.1 分散式檔案系統

網路檔案系統的出現，讓使用者能如原生磁碟檔案的存取遠端系統的檔案，此服務讓使用者樂於在網路上傳輸各式資料，如微軟的網路芳鄰、UNIX 的 NFS 協定等，無須在每台主機儲存相同檔案，能避免掉檔案內容不同步，此優點對經常變動的資料尤為重要，在安全的考量上也是優點。然而，單點服務的網路檔案系統，最大的問題是單點網路流量，當網路檔案伺服器同時服務過多掛載節點時，用戶端要求增加會加重硬碟讀取及寫入負擔，此為影響網路檔案伺服器效能之主因。另外，當某個網路檔案系統節點停止服務，用戶端會產生無限循環嘗試的當機現象，此時使用者也無法存取伺服器端其他共用資源。

為解決單點式網路檔案系統的面臨的困難，分散式網路檔案系統[5][6]與操作介面紛紛被提出，如 Microsoft DFS(MSDFS)、OpenAFS、ADFS、WebDAV，其他如新興正在發展中的 MooseFS、GlusterFS、Ceph、FhGFS、Lustre、HDFS 等，部份分散式檔案系統，需要特殊命令或 API 才能進行資料存取，如 Hadoop Distributed File System (HDFS) 等，因為需 API 以及程式開發才能有效運用分散式檔案系統內部資料，因此無法廣為大眾採用普及。

Filesystem in Userspace(FUSE)於 2005 年由

Linux 2.6.14 核心支援後，在無需理解檔案系統內部運作架構下，也不用學習核心模組太多相關知識，即可開發存取使用者檔案系統的存取介面，FUSE 有效簡化作業系統提供新檔案系統所需之開發成本，特別適用於各種虛擬檔案系統和網路檔案系統，目前許多分散式檔案服務漸漸都支援 FUSE 操作存取方式，雖然在使用者模式下實作檔案系統服務會產生額外核心/使用者態切換的效能花費成本，但此方式節省許多網路檔案系統維護程式碼版本的時間。本論文主要希望能夠實際測試虛擬環境運作架構下，透過 FUSE 用戶前端，各 MooseFS 資料存取前端能否有效運作於 MooseFS 虛擬資料節點主機，並比較此架構與實體 MooseFS 主機環境效能差異變化情形。

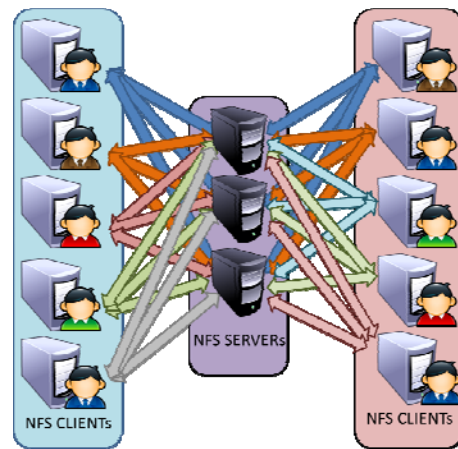


圖 1 傳統 NFS 掛載架構

MooseFS 主要透過 Master、Metalog、Chunk、FUSE 用戶端等部份組成完整之運作架構，MooseFS 目前尚未推出解決 Master 單點失效，此時分散式檔案系統即失效的問題，主要原因在使用者放置的檔案資料需要透過 Master 伺服器節點，決定將使用者資料放置於哪些 Chunk 節點，當 Master 節點失效時，會造成 FUSE 前端節點失效，此時使用者所執行的 mfsmount 服務將會無法回應，唯一解決方式是重新修復 Master 節點，Chunk 節點將會自動與 Master 節點重新產生正確連結。已有文獻嘗試將 MooseFS 建立為多點形式 Master 節點架構[7]，而目前在業界常見作法是搭配同步鏡像架構，將 Master 節點即時鏡像一完整複製，透過 DRBD 與 HeartBeat 搭配，有效達成 Master 節點失效後即時回復服務之目的[8]。

GlusterFS 最新版本為 3.4，與 MooseFS 極為不同，GlusterFS 架構無須具備實體 Master 中央控制服務節點，僅分為 Client 端及 Server 端，Server 端主要靠 Brick 來儲存各式資料，再透過各式服務模組，將資料提供至不同種類前端服務，如 NFS、Samba 等前端，無 Master 節點最大好處在單點故障即失效的機率大大降低，唯新版之穩定性與效能尚須進一步的驗證。

2.2 虛擬化環境

虛擬化在近幾年成為建置雲端熱門討論之一，目前已有許多文章對 XEN、VMware、Hyper-v 等虛擬化環境測試執行效能分析，藉由虛擬化技術將分散式檔案系統建構於內部運作，現階段相關文獻與測試資料尚不普遍，而虛擬化技術目前較成熟的應用方案為虛擬機器建構網頁及資料庫伺服器[9]，達成 24 小時不間斷服務環境。

在商業版本中 VMware、Hyper-V、XEN、KVM 是目前較常被用來選擇作為商業虛擬化服務環境，如 Hi-Cloud、Windows Azure、Amazon EC2、Google Computing Engine 等雲端服務環境，客戶端可選擇自己喜歡擅長的虛擬化平台，選擇租用虛擬機器數量與計算能量，越來越多文獻針對不同 Hypervisor 間不同情境下運作效能提供測試結果。

在這兩年間，各種虛擬化技術的改良與進步，如硬體設計針對虛擬化架構的改良與支援，各式虛擬化運作軟體環境，由驅動程式、作業系統核心針對虛擬化架構的效能改良、網路虛擬化橋接器軟體效能提昇、日趨成熟的虛擬機器管理介面，使過去只能在實體機房處理與控制的情境，越來越容易在實體機房架構建構出對應虛擬機房的管理策略。

網路卡虛擬化技術的大幅度進步，Virtual Switching 與 Virtual Bridge 的支援，有效解決實體網卡與虛擬機器間資源共享與管理等問題[10][11]，雖然目前虛擬化硬碟讀寫效能仍是虛擬環境執行瓶頸點，相對於過去已有大幅度提昇[12]，除虛擬化檔案系統格式更為優異外，針對差異化存取版本控制與讀寫效能也有顯著提昇，這些進步若能有效應用於分散式檔案系統運作環境中，相信可有效提昇資料儲存、存取效能、備份與管理等多方面議題。

2.3 虛擬化分散式系統

分散式檔案系統運作於虛擬化架構下相關研究，目前已有文獻[13]提供相關觀念與架構比較。最後真正重要資料才備份回系統的 NAS，相信這樣的運作架構，對未來資料與計算資源管理上，將會提供非常大的效能與便利性。

國家高速網路中心為打造雲端機房，建置名為 Ezilla 簡單龍雲端管理軟體，藉由 Ezilla 由底層 IaaS 到中間平台管理層 PaaS，使資料中心能擁有彈性且具延展性雲端基礎架構，提供更完善軟體上層服務。Ezilla 運作架構，底層採用 MooseFS 分散式檔案系統，目前採用實體機器之 MooseFS 運作架構。

圖 2 分別列出目前常用之 (A) SAN 資料服務 (B) MooseFS 實體機器運作 (C) MooseFS 運作於虛擬機器等三種運作架構示意圖，此三種架構之特性列於表 1，於表 1 中可觀察出 MooseFS 底層進行虛擬化後相關的優缺點。

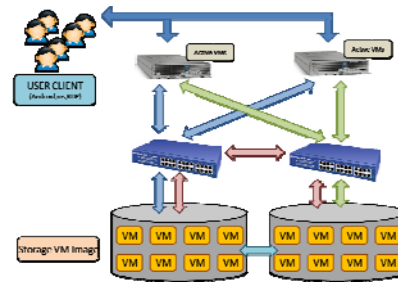


圖 2 架構(A)

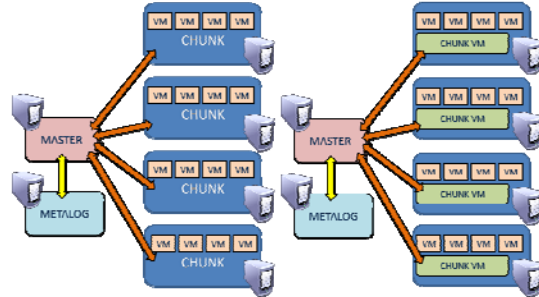


圖 2 架構(B)

圖 2 架構(C)

表 1 - 圖 2 架構比較表

架構	架構(A) SAN	架構(B) MooseFS	架構(C) MooseFS in VM
效能	由 1G、10G 與光纖網路及各式儲存體建構資料服務環境	理論上能使用實體硬碟及網卡效能	僅能使用虛擬硬碟及網卡所呈現的效能，讀寫速度較低
管理	MIS 人員須熟悉 SAN 架構及網路防火牆設定，安全控管集中化是優點	操作人員須熟悉分散式檔案架構操作方式，單點資料伺服器故障時，前端不受任何影響。單點資料硬碟需要維護升級方便	操作人員除熟悉分散式檔案操作方式，還須熟悉虛擬化架構操作。單點資料伺服器故障不影響外，透過虛擬化技術，立即啟動新的差異化單點伺服器即可進行故障回復
硬碟空間	需一次採購至足夠的需求單位大小	由所有實體硬碟組成 Chunk Server 之合併總容決定(可動態掛載卸除)	由 Chunk Server 虛擬化大小決定，最大好處是可預先配置理論上所需的硬碟空間，且無實體硬碟大小限制
成本	採購成本較高，資料大量傳輸環境需網路設備硬體及軟體升級	無須任何虛擬化硬體支援，舊 PC 掛上大硬碟即可服務	須具備虛擬化桌上型 PC
維護	須整體汰換及批次升級	僅需抽換硬碟或是周邊設備	僅需維護映像檔
備援	如需建立備援架構，需購置同套軟硬系統作為其備援，成本加倍	設定 Goal 份數，系統自行備份，但若實體節點損毀仍需再手動建置節點	設定 Goal 份數做備援，並自行同步，但 chunk 或是 master 節點可任意變動至任何實體節點上

3. 研究流程與架構

3.1 KVM+MOOSEFS

利用 KVM 作為虛擬化運作環境，透過分散式檔案系統有效管理每台虛擬機器的儲存空間，讓所有虛擬機器的硬碟空間有效分配與利用，本論文採用 MooseFS 作為實現分散式檔案系統，作為大量儲存虛擬機器映像檔空間，KVM 過去常用的映像檔格式為 QCOW2、RAW、QED 等格式，其中 QED 格式為 KVM Hypervisor 中虛擬硬碟讀寫速度較佳的虛擬硬碟格式，虛擬機器其硬碟檔案格式以及分散式檔案系統的資料讀寫速度是主要影響整體虛擬機器作業系統及前端使用者感受最大的兩個因素。

本論文中所測量的方式是將已安裝作業系統的映像檔，並利用差異化硬碟，大量差異出虛擬硬碟，因此無須再重新安裝虛擬機器映像檔，而各種 KVM 硬碟格式所展現出來的速度及效能方面相關的測試已由 IBM 公司提出相關測試及報告[14]。

觀察 MooseFS 執行效能瓶頸點，主要發生於 Master 節點與單位時間內大量存取某特定 Chunk Server，而過去對虛擬化環境的負面印象，如檔案存取效能低落、網路卡驅動程式無法有效負荷等議題，本研究 MooseFS 採用之虛擬化運作環境為 Linux KVM，為加速虛擬機器的運作效能，同時維持日後虛擬映像檔的管理與維護方便，Linux KVM 採用 QED 虛擬映像檔案格式，作為各 MooseFS 虛擬機器的開機映像檔，MooseFS 之虛擬機器運作於各實體機器之單機硬碟，為了讓 KVM 能有效率的支援硬碟存取驅動程式與網路卡驅動程式，KVM 採用 Virtio 驅動程式存取硬碟與網路卡，以利各虛擬機器存取網路卡與實體硬碟之效能。

3.2 虛擬化 MooseFS 測試架構與流程

研究流程透過虛擬機器映像檔案儲存於實體機器 MooseFS 檔案系統以及虛擬機器 MooseFS 檔案系統，量測虛擬機器儲存於兩種不同 MooseFS 環境時，開啟虛擬機器所需要開機時間，比較兩者之時間差異，評估 MooseFS 運作於虛擬機器環境下，是否可滿足實際運作需求。實際運作需求參考指標為虛擬機器開機時間，同時觀察虛擬機器成功開機後是否可正常運作。本研究採用之虛擬化 MooseFS 架構，運作環境如圖 3。

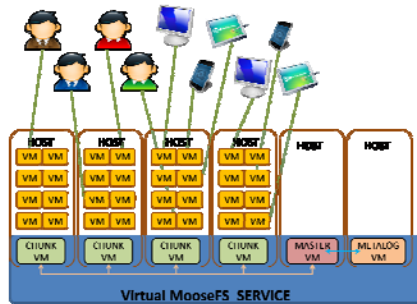


圖 3 - 虛擬 MooseFS 分散式檔案系統架構

3.3 軟體與硬體環境配置

3.3.1 測試環境

表 2 為本研究實體機器測試軟體與硬體環境，規格如下：

表 2 - 本研究採用之實體機器規格

設備	作業系統	CPU	硬碟	記憶體	數量	說明
伺服器 (a)(b)	Ubuntu 13.04	3.2G, 6Core	500G	64G	2	Host
HP (1)(2)	Ubuntu 13.04	2.4G, 4Core	500G	24G	2	Host
伺服器 (3)(4)	Ubuntu 13.04	3.4G, 4Core	500G	24G	2	Host

MooseFS 分別於伺服器(a)運作 Master 及伺服器(b)運作 Metalog Server，伺服器(1)(2)(3)(4)運作 Chunk Server，而對等運作架構於虛擬 MooseFS 運作架構時，分別將各實體機器所提供之 MooseFS，於各自機器上建立對應的虛擬機器提供同樣的服務，例如伺服器(a)即建立一虛擬機器提供 MooseFS Master 服務、伺服器(b)建立虛擬機器提供 Metalog 服務，伺服器(1)(2)(3)(4)即各自建立一 Chunk Server 虛擬機器。為方便測試編號，將伺服器(1)(2)之虛擬機器編號為 1-16、伺服器(3)(4)之虛擬機器編號為 21-36，伺服器(1)(2)(3)(4)共組成 32 台虛擬機器。

上述過程嘗試進行同樣的硬體條件下，實體機器運行 MooseFS 與虛擬機器運行 MooseFS 服務的效能變化。MooseFS 虛擬機器軟硬體規格如表 3，可以看出虛擬硬碟的大小於初始化設定時，大於實體機器所提供之硬碟大小，若硬碟實際上不敷使用時，可分別將虛擬機器映像檔案再搬移至更大的實體硬碟提供服務。

表 3 - MooseFS 虛擬機器軟硬體規格

設備	作業系統	CPU	硬碟	RAM	數量	用途說明
伺服器 vm1,vm2 vm3,vm4	Ubuntu 13.04	4Core	1000G	4G	4	Chunk VM
伺服器 vm5	Ubuntu 13.04	4Core	1000G	4G	1	Master VM
伺服器 vm6	Ubuntu 13.04	4Core	1000G	4G	1	Metalog VM

3.3.2 使用者前端虛擬機器測試環境

測試共分兩階段進行，伺服器(1)(2)(3)(4)各自啟動八台虛擬機器。分為兩個啟動條件，第一次先測試於硬碟差異化完成後，即同時於 4 台 Chunk Server 各自啟動 8 台虛擬機器，最後會陸續的完成 32 台虛擬機器的啟動，此方式每次實驗完畢之後，將虛擬機器的差異化硬碟刪除並重新建立新的 32

台差異化硬碟於 MooseFS，然後重新開機再啟動，避免快取機制與其他的變因。

第二個啟動條件為第一次開啟差異化硬碟完成後，隨即重新啟動實體機器，不重新製作差異化硬碟，因為在實驗的過程中，發現第一次虛擬機器啟動時，需要將 SWAP 等相關的虛擬化硬碟資料回寫至 MooseFS，此時對 MooseFS 的存取量相對已經初始化好一個虛擬差異化硬碟的環境，啟動虛擬機器的時間差異會相當大。開機完成後，各 Ubuntu 機器會自動設定 IP，並且回報回後端的資料庫系統進行啟用回報的時間回傳，此時使用者也可以開始登入至各虛擬機器使用。

實驗量測環境分別為在 Chunk Server 的機器上，啟動 8 台 Ubuntu 13.04 的虛擬機器，由各台機器開機完成的時間，作為 MooseFS 提供服務的回應基準，因啟動虛擬機器除了啟動虛擬機器本身機器的效能外，虛擬機器映像檔儲存於分散式檔案系統的環境下，各虛擬機器的運作效率主要變因會發生於虛擬機器運作期間需要與後端檔案伺服器讀取與回寫的次數，期間除網路效能、硬碟存取效率外，主要變因還發生在分散式檔案的演算法與搭配虛擬化以後之虛擬化環境的效能調校。

因此透過直接量測同樣的虛擬機器的數量開啟的條件下各虛擬機器開機完成的時間，即可有效的直接取得真正服務狀態下，前端使用者在連線至虛擬機器後實際的使用感受。前端使用者服務之虛擬機器以 Ubuntu 13.04 為作業系統的映像檔，各虛擬機器之 CPU 數量設定為 2，記憶體設定為 2048M、虛擬硬碟設定為檔案格式 QED，大小為 500GByte，虛擬機器開機完成後，使用者可透過遠端桌面或 SSH 連線至虛擬機器執行相關的運算。

3.3.3 延緩開機策略參數調教

透過延緩開機的策略，可同時進行兩方面的負載緩解，一個部份是虛擬化檔案系統的讀寫效率能得到平衡，另外一個部份是大量同時開啟虛擬機器，對實體機器也會產生相當大的負擔，藉由延緩開機時間策略，使後端 MooseFS 分散式檔案系統與前端的虛擬機器開機環境得到平衡的狀態，目前已觀察到實際的現象是有效的延緩虛擬機器的開機時間，總體的開機時間相對的較短且單位時間總頻寬需求較小。

4. 實驗結果與討論

本研究主要分為實體 MooseFS 及虛擬 MooseFS 兩類測試結果，比較開啟虛擬機器於實體與虛擬 MooseFS 分散式檔案系統可觀察出兩架構間效能與維護之優缺點。

4.1 虛擬機器於實體 MooseFS 開機時間

圖 4 為量測 Ubuntu 13.04 於實體 MooseFS 當映像檔於實體分散式檔案系統所呈現趨勢及整體開機效能，利用不同虛擬機器開機延遲觀察整體開機

時間之變化，1-16(D0)(I)與 21-36(D0)(I)為同時開啟 32 台虛擬機器之開機(延遲時間為 0 秒)，此時 Ubuntu 13.04 映像檔案需要大量 MooseFS 讀寫資料，整體時間拉高至 296 秒。(D 為開機延遲秒數、I 為差異化硬碟第一次開機，未初始化，需建立 SWAP 及其他硬碟資訊、R 為初始化完成之開機時間、H 為實體機、V 為虛擬機)

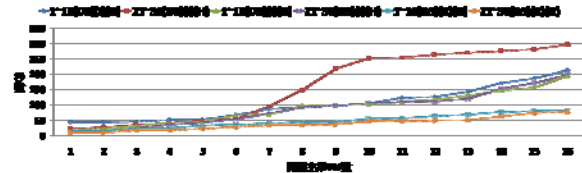


圖 4 - Ubuntu 虛擬機器於實體 MooseFS 開機時間

將虛擬機器開機時間設定為每隔 5 秒啟動 (D5)(I)，發現整體虛擬機器啟動時間均低於 (D0)(I) 開機時間，進一步觀察已初始化且每隔 5 秒啟動虛擬機器之開機情形 (D5)(R) 曲線，明顯的由 296 秒的開機時間縮短至 81 秒。由此可知透過延遲開機策略確實能有效縮短整體開機時間，也助於 MooseFS 檔案系統負載量降低。

4.2 Ubuntu 虛擬機器於虛擬分散式開機之效能變化

與 4.1 節進行之實體 MooseFS 環境與虛擬 MooseFS 進行對照組比較，圖 5 顯示虛擬 MooseFS 開機時間秒數整體拉高接近 490 秒，將實體 MooseFS 與虛擬 MooseFS 各台虛擬機器開機時間加總後平均，時間差異分別為 1.8 倍 (D0(I))，2.2 倍 (D5(I))，1.1 倍 (D5(R)) 倍。延遲 5 秒開機時間在虛擬化 MooseFS 未初始差異化硬碟狀態下，無法明顯提升開機速度，可能原因為延遲 5 秒對虛擬 MooseFS 仍無法有效降低單位時間網路與硬碟頻寬需求，顯示除測試機器效能本身差異，MooseFS 資料排程演算法會影響虛擬機器開機之時間。

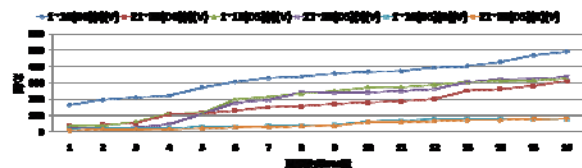


圖 5 - Ubuntu 虛擬機器於虛擬 MooseFS 開機時間

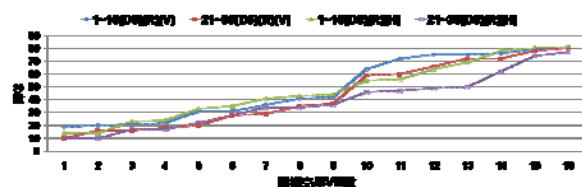


圖 6 - 實體與虛擬 MooseFS 延遲 5 秒開機時間比較

最後透過比較實體 MooseFS 開機與虛擬 MooseFS 之 (D5(R)) 開機時間的差異，可以明顯的發現，當差異化硬碟初始化完成後，無須大量同時讀寫 MooseFS 分散式檔案系統情境下，圖 6 之 (D5(R)(V)) 與 (D5(R)(H)) 各虛擬機器開機總時間之

平均時間差距為 1.1 倍(46/42 秒)，此結果顯示透過適度的開機延遲，讓 MooseFS 單位時間內不致連續維持在高負載條件下，實體 MooseFS 與虛擬 MooseFS 提供之外顯服務效能趨近一致。

4.3 實測案例討論

虛擬化環境伺服器可快速差異化啟動，省下大量系統程式版本同步與更新步驟與時間，只需準備新版虛擬機器，以差異化指令執行影像檔案差異化，即可於不同實體主機間轉移與啟動虛擬機器與相關服務，此特性與 MooseFS 分散式檔案系統結合後，MooseFS 的各 Chunk 節點無須於各實體機器獨立安裝，只需將實體機器安裝虛擬化環境，啟動 MooseFS Chunk 虛擬機器後，實體機器硬碟裝置即可自動成為網路的一服務節點，各 Chunk 節點間也會自動備分同步不同 Chunk 中的檔案，不論各儲存的節點發生單點故障或是發生多點暫態不穩定故障時，整體系統仍舊能提供前端使用者資料服務。

在實作虛擬化分散式檔案系統時，當系統其中一個虛擬化 Chunk 節點產生作業系統故障或硬碟容量不足時，透過虛擬化搬遷技術，將單一節點映像檔搬移至較大實體硬碟空間中，不僅省去主機停機修復的大量時間也省去軟體層設定，且搬移過程依舊不影響整體虛擬化 MooseFS 系統運作，搬移完成後之 Chunk 虛擬機器節點也能立即運作，整體系統帶給資訊人員更省時與便利。

5. 結論

針對於 Linux 環境下的 KVM 進行測試與評估，藉此能夠實際瞭解 KVM 虛擬化環境實際的穩定性與其運作效能，此一虛擬化分散式檔案系統環境的實測實作對未來在企業資料中心，學術單位或是公家機關單位在建置虛擬化能夠理解虛擬化帶來便利性及延展性，充分運用虛擬化技術，對於中小企業或是大企業在建立自身的 datacenter 時，建立檔案系統來存放，能否有助於在整體雲端系統建置方面，而在管理層面能有效減少 MIS 網管人員負擔，企業有機會可以大量建置虛擬機房，且收費機制與服務也較以往多元，透過商業化 IaaS 服務，引發由叢集式主機遷移至虛擬化主機叢集的風潮。

本運作架構顯示未來能夠移植實體機器分散式檔案至虛擬化分散式檔案架構，在虛擬化技術的日趨成熟與效能日益提昇後，過去對虛擬化環境的各種顧慮逐漸的達到實用階段，本研究顯示虛擬化技術與分散式檔案技術的結合，將可進一步的使虛擬化技術的應用邁向新一代儲存體的時代，對於虛擬化教室與中小型企業的辦公室本身亦可達成更方便有效的管理運作架構。

如能將過去需要專用實體機器的分散式檔案系統建立於日常辦公用的桌上型機器，讓每一台辦公用的電腦，透過虛擬化技術形成既是雲又是端的環境，有效透過虛擬化技術將機器維持應用上的軟體環境實體切割，每台辦公用電腦即可各自提供整

體儲存資源的一部分，再透過分散式檔案系統自動達成資料分配與管理，機器的利用率即可有效提昇，實則大大提昇每台機器的利用率與產能[15]。本運作架構可作為實體機器分散式檔案移植至虛擬化分散式檔案運作架構的參考。

參考文獻

- [1] 張保榮, 虛擬化實施評估和配置成本, 民國, 國立高雄大學資工系
- [2] 趙祥景, 一個具備可靠性與安全性的分散式檔案系統之研究, 民國 97 年 7 月, 私立逢甲大學資工所, 碩士論文
- [3] Chang-Hsing Wu, Yi-Lun Pan, Hsi-En Yu, Hui-Shan Chen and Weicheng Huang, Power Saving for Fast Deployment Private Cloud Toolkit – Ezilla with Infrastructure Services, National Center for High-performance Computing, Taiwan, R.O.C. July 16-19, 2012
- [4] 莊富雄, 分散式檔案系統建置於虛擬化平台之研究, 2012, 私立世新大學資管所
- [5] 鐘鈞彥, 具延展性的高效能分散式檔案系統之研究, 民國 100 年 6 月, 私立逢甲大學資工所, 碩士論文
- [6] Benjamin Depardon and Cyril Seguin and Ga el Le Mahec, Analysis of Six Distributed File Systems, February 15, 2013
- [7] Jiongyu Yu, Weigang Wu, Huaguan Li, DMooseFS: Design and Implementation of Distributed Files System with Distributed Metadata Server, 2012 IEEE Asia Pacific Cloud Computing Congress
- [8] Opsren, http://rritw.com/a/bianchengyuyan/_NET/20120612/170023.html 2012.6.12
- [9] 呂泓縉、賴仲哲、石旻諺、李昱宏、呂承義、葉俊雄、李盛安, 虛擬化網頁伺服器負載平衡效能分析, TANET 2012, 2012.10
- [10] 陳柏翰, 實作一個 KVM 雲端基礎建設與監測系統, 民國 100 年 7 月, 私立東海大學資工所, 碩士論文
- [11] 鄭翔耀, 在雲端環境中提供一個動態資源分配模型管理虛擬機器, 100 年 7 月, 私立東海大學資工所, 碩士論文
- [12] 廖本加, 於雲端計算環境上高可用性儲存系統之實作, 民國 100 年 7 月, 私立東海大學資工所, 碩士論文
- [13] 潘介棟, 使用平行虛擬檔案系統建置網格大型儲存系統, 民國 94 年 6 月, 私立東海大學資訊工程與科學所, 碩士論文
- [14] IBM – 2010 Linux Plumbers Conference, 取自, <http://pic.dhe.ibm.com/infocenter/lnxinfo/v3r0m0/topic/laav/LPCKVMSSPV2.1.pdf>
- [15] 吳欣容、呂泓縉、李昱宏、葉俊雄、李盛安, 虛擬化技術應用於手機程式設計課程相較於傳統保護卡教室之成效探討, IETAC2013 第六屆資訊教育與科技應用研討會