應用快取記憶體提高即時遷移虛擬機器之效能

陳莊智*[a]、李維聰[a]、楊堯強[a]、孫天文[b]
淡江大學電機工程研究所[a]
中山科學研究院[b]
eric1817041@gmail.com*

摘要

雲端運算是一種透過網際網路技術將服務提供給外部使用者，能夠很方便的隨著使用者需要做存取設定的一種模式。虛擬機器(virtual machine)常用於雲端系統中，當系統遷移時，可以利用虛擬機器做即時遷移，即時遷移是虛擬機器的重要功能之一，即時遷移的停機時間長短更是使用者評估虛擬機器效能的重要指標，因此本研究試圖縮短虛擬機器在即時遷移的停機時間。本研究在虛擬機遷移過程中加入快取(cache)作為改善技術，以期待達到解決高停機時間及高總遷移時間的問題。

關鍵字：快取(cache)、即時遷移(Live Migration)、預先複製(Pre-copy)、後複製(Post-copy)

一、 前言

雲端運算(Cloud Computing)是目前政府組織、學術研究、科技產業都相當重視且積極推動的科技話題。什麼是雲端運算呢？聽起來雲端運算是一個新興的話題，但實際上，雲端運算並不是一種新的技術，它代表的是透過網際網路使得在各處使用者的電腦能夠彼此合作，是一種無遠弗屆的服務。

雲端運算服務可分為三種服務層級。軟體即服務(Software as a Service，SaaS)是指運用網際網路提供軟體的一種服務。業者將應用軟體及資料放在雲端供使用者存取使用，使用者不需要安裝任何軟硬體，只要透過網際網路連線即可進行資料的存取。使用者只需負擔授權費，不需要維護硬體設備、軟體開發，應用軟體及資料都放置在系統廠商的伺服器中，由系統廠商負責維護。平台即服務 (Platform as a Service，PaaS)，是指系統廠商建立程式開發平台以及作業系統平台，讓程式設計師或開發人員透過網際網路來撰寫程式、自行建立網路應用程式、開發軟體程式系統，是屬於雲端運算的中層服務。基礎架構即服務(Infrastructure as a Service，IaaS) 指的是使用者不需自行購買或建置硬體及各項基礎設施，只需要向雲端服務廠商透過租用的方式，便可使用儲存容量、網路、處理器等運算設施。雲端基礎架構即服務(Infrastructure as a Service，IaaS)是將硬體基礎作為服務，給使用者虛擬化的網路資源、運算資源及儲存空間，由雲端服務廠商做網站及主機的代管，可以隨時且方便的進行需求擴充。因為使用者不需花錢購買底層的雲端基礎設施及管理，因此可以讓使用者減少購買硬體的支出及建構設施的成本。

虛擬機像實體電腦一樣，可以執行各種應用程式及作業系統，是一套軟體容器，而虛擬化技術可使一台實體機器上執行多台虛擬機，每台虛擬機都可以在不同環境中共用實體電腦的資源。虛擬機的系統遷移是指把來源端電腦的操作系統及應用程序移動到目標端電腦，並且可以在目標端主機正常運作。虛擬機的系統遷移時間往往是用戶評估虛擬機效能的因素之一，良好的遷移工具要達到最短的總遷移時間以及停機時間，讓使用者不覺得有服務中斷的感覺，通常虛擬機遷移的性能指標有以下三項：

1.總遷移時間：是指資料從來源端遷移至目標端所經歷的總時間。

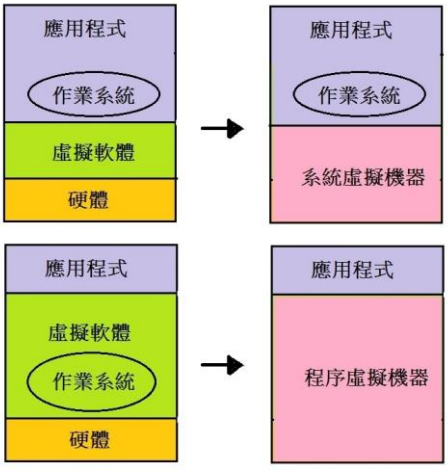
2.停機時間：遷移過程中，來源端主機及目標端主機同時不可使用的時間。

3.應用程序的性能影響：資料遷移後在目標端主機使用的服務性能影響程度。

二、 相關技術與背景介紹

2.1 虛擬機器

虛擬機器不是實體機器而是一種軟體，利用軟體模擬硬體的方式，在電腦平台與用戶端電腦之間建立一種環境，使得用戶可以利用這個軟體所建立的環境來操作軟體，虛擬機器中所執行的所有軟體，只能運用虛擬機器中的資源，是一種可以像真實機器一樣執行各種程式的電腦軟體。虛擬機器的目的是以虛擬的方式提供用戶所期望使用到的機器架構，為一種軟體與實體硬體的夾層[1]。



圖一 系統虛擬機器 VS 程序虛擬機器

虛擬機器根據服務的對象不同大致可分為兩類：系統虛擬機器(System Virtual Machine)以及程序虛擬機器

(Process Virtual Machine)[2]，如圖一所示。系統虛擬機器是提供一個可以執行作業系統的平台，用戶可以藉由系統虛擬機器所提供的平台執行一個課端作業系統(Guest OS)；程序虛擬機器只能對單一程序服務執行某個特定的程序。

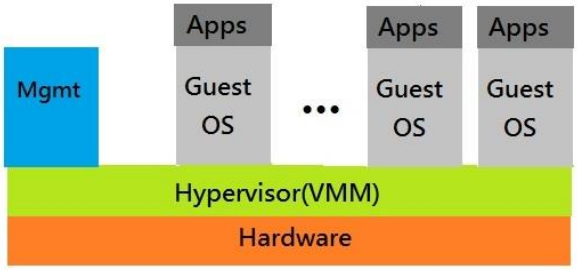
2.2 虛擬化技術

系統虛擬機器又因為虛擬技術的不同，而有所不同。例如，虛擬化技術分成全虛擬化與半虛擬化，不同的虛擬技術會影響虛擬機器所能使用的硬體資源與作業系統的選擇。

2.2.1 全虛擬化(Full Virtualization)

亦稱原始虛擬化技術，利用虛擬化產生的作業系統及硬體之間可以自由操作，課端作業系統所能使用的硬體資源，會因虛擬出來的硬體資源而受限，較無法挪用實體電腦的硬體。全虛擬化的基本輸出輸入系統(Basic Input/Output System，BIOS)、顯示卡等等硬體皆可以被模擬，只要安裝專屬的驅動程式即可，但處理器(Central Processing Unit，CPU)、記憶體(Random Access Memory，RAM)、主機板則無法模擬，此種虛擬技術是目前最廣泛被使用的一種。

全虛擬化的優點是無論在怎樣的硬體環境中，課端作業系統都能維持一致的相容性，而且能和實體機器使用不同的作業系統，轉換工作動態時無需修改課端的作業系統，每個虛擬器都完全獨立[3]，全虛擬化架構如圖二所示。



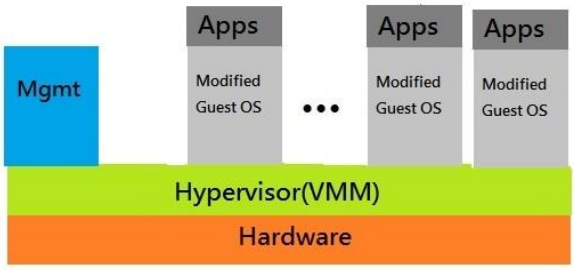
圖二 全虛擬化架構圖

由於全虛擬化可以在原作業系統下執行另外一個作業系統，因此全虛擬化的缺點則是會造成實體機器較大的負擔。

2.2.2 半虛擬化(Para Virtualization)

半虛擬化的主要目的是改善傳統平台下對硬體支援較弱的問題[3]，半虛擬化是指將好幾個記憶體位置的程式，設定成可在不同時間呼叫的情形而不是在硬體設置虛擬層。半虛擬化修改過後可以知道目前是在虛擬環境中執行程式，因此必須與虛擬平台密切的相容，否則無法在實體機器上執行。也就是半虛擬化必須和虛擬化平台互相支援，才能正常運作。

半虛擬化的優點是需要把硬體的效能花費在模擬硬體層上面，客端系統可以但缺點就是虛擬機器中的作業系統必須與實體機器一樣才行。



圖三 半虛擬化架構圖

2.3 虛擬機器的遷移

虛擬機器的系統移是指把來源端主機上的操作系統和應用程序移至目標端主機上，並且能夠在目標端主機正常的運作。虛擬機技術的發展越趨成熟，因此系統的遷移可以更加靈活及便利，在之前沒有虛擬機器時，只能靠系統備份以及恢復技術，將來源端的操作系統及應用程序先複製到儲存介質，再將介質上的資料複製在目標端主機，進行恢復的動作。現在只需運用遷移技術即可以達到相同的效果。之前的電腦服務器體積較為龐大，現在的服務器體積縮小很多，因此節省用戶大量存放機房的空間，更可以節省用戶管理資金、維修設備等費用，系統遷移的優勢在於簡化系統的維護及管理、提高系統負載均衡、增強系統容錯率等等。

虛擬機器遷移可以分為非即時遷移和即時遷移[4]。而在虛擬機器的系統遷移過程中，會有兩種時間的描述：停機時間(downtime)及總遷移時間(total migration time)[5]。停機時間是指系統在來源端暫停開始一直到在目的端啟動所經過的時間。總遷移時間是指整個遷移過程從傳輸開始一直到完全傳輸結束使得來源端可以停止服務所經過的時間。一個好的遷移工具，就是能達到最小的停機時間和最小的總遷移時間，並且在遷移至目標端主機後執行效果要能像在來源端主機上執行能有一樣的效能。

2.3.1 非即時遷移(Non Live Migration)

又稱為靜態遷移、離線遷移。遷移開始之前先將虛擬機器暫停運作，然後將主機上的系統狀態拷貝及複製記憶體的內容後，複製到目標端主機後，在恢復目標端主機的運作。非即時遷移不在乎停機時間的長短，對用戶來說會有一段停止服務的時間，虛擬機器暫停運作，應用程式不會執行，因此如果是需要即時反應的服務，則不適用此種遷移方式。在這段停機時間中，來源端主機開始複製記憶體內容和處理器的狀態，再將複製好的狀態在目標端主機恢復。非即時遷移因為會先將虛擬機器暫停運作，再傳送每一個分頁到目標端主機，因此會有高停機時間的缺點；不過在傳送過程每一個頁面只會傳送一次，因此會有最小總遷移時間的優點。

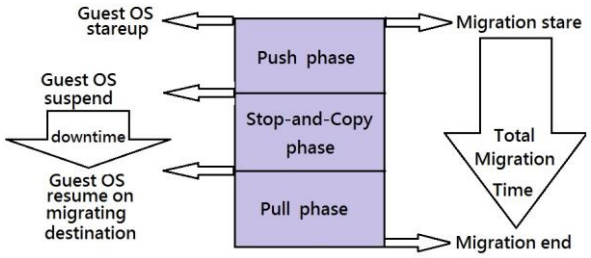
2.3.2 即時遷移(Live Migration)

即時遷移又稱為在線遷移或實時遷移，指的是虛擬機服務在正常運作沒有中斷的同時，虛擬機在不同主機之間進行遷移。即時遷移和非即時遷移的過程其實是一

樣的，不同的地方是即時遷移的停機時間要比非即時遷移短很多，短暫到使用者沒有察覺到主機的服務有中斷。遷移過程的前半段，來源端主機仍然繼續運作，但目標端主機已經漸漸具備運行系統需要的資源，經過一個短暫時間的切換，使得控制權轉移到目標端主機，因此即時遷移重視停機時間的長短。

2.4 即時遷移技術

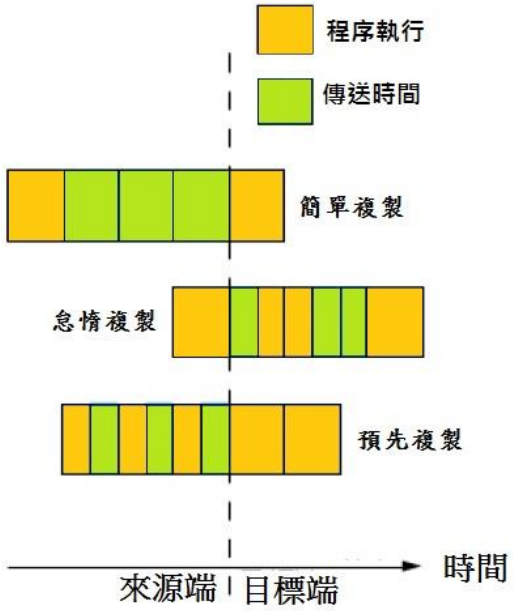
虛擬機在遷移過程通常可以分為三個階段[5]，如圖四所示，分別為推進階段、暫停-複製階段及拖行階段。



圖四 動態遷移三階段示意圖

推進階段(Push phase)是指來源端主機系統運作時在同一時間將系統內容搬遷；而暫停-複製階段(Stop and Copy phase)是指來源端系統停止運作，系統內容進行最後的搬遷，當搬遷結束之後目標端的系統即開始運作；至於拖行階段(Pull phase)則是指從來源端搬遷至目標端的系統開始運作時，如果目標端系統存取到尚未複製來到目標端的內容時，則會從來源端立即將內容傳送至目標端。

目前在動態遷移中常用的遷移技術如下[7]，以下將說明各種常用的動態遷移技術。



圖五 常見的動態遷移技術示意圖

2.4.1 簡單複製(Simple copy)

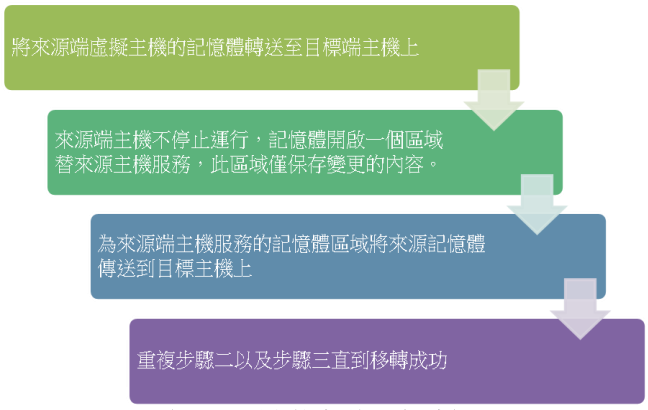
簡單複製是指先將來源端虛擬主機暫停運作，然後將欲複製的內容進行複製，再將複製好的內容放至目標端主機上，接著啟動目標端的虛擬系統。這是一個極為簡單的遷移方式，但所花的停機時間會和遷移內容的多寡有相對的長度。

2.4.2 怠惰複製(Lazy copy)

怠惰複製是指系統遷移時，只搬移足夠讓目標端虛擬主機可以運作的最少內容。這種遷移方式的好處是和簡單複製相比，怠惰複製可以省下較多的停機時間，但可能因複製的內容不夠完整，因此常會出現內容遺失，像是頁錯誤(page fault)的情形。當出現頁錯誤情形時，再去來源端複製需要的內容移送至目標端，因此會產生更高的總遷移時間。

2.4.3 預先複製(Pre-copy)

預先複製是指在來源端虛擬機在不停止運行的情形下，將所有的系統內容標記成唯讀模式然後傳送到目標端。預先複製首先在推進階段中，以疊代(iterative)的形式，使用回合制的系統內容搬移。當系統試著要修改內容的時候，標記該內容為髒頁(dirty page)。當搬遷開始之後，標記為髒頁的內容會再次被標記成唯讀並傳送到目標端。這樣的動作會一直持續，直到標記為髒頁的內容非常少的時候，來源端虛擬機才暫停，並傳送剩下的內容，然後在目標端恢復運行。此種遷移方式系統暫停無法回應的時間只有在最後搬遷髒頁內容的階段，但是同一區塊的內容可能會有多次搬遷，因此此種遷移方式可得到一極短的停機時間，但可能會增加總遷移時間。預先複製大致可以分為四個步驟，如圖六所示。



圖六 預先複製的四個步驟

2.4.4 後複製(Post-copy)

後複製的遷移程序剛好和預先複製相反，後複製是指先暫停來源端的虛擬主機，然後複製必要的處理器狀態到目標端主機，接著恢復目標端主機的運作，從來源端存取所需的記憶體頁面。一旦虛擬機在目標端主機恢復，如果連接尚未傳輸過來的內存頁面就會引起網頁異

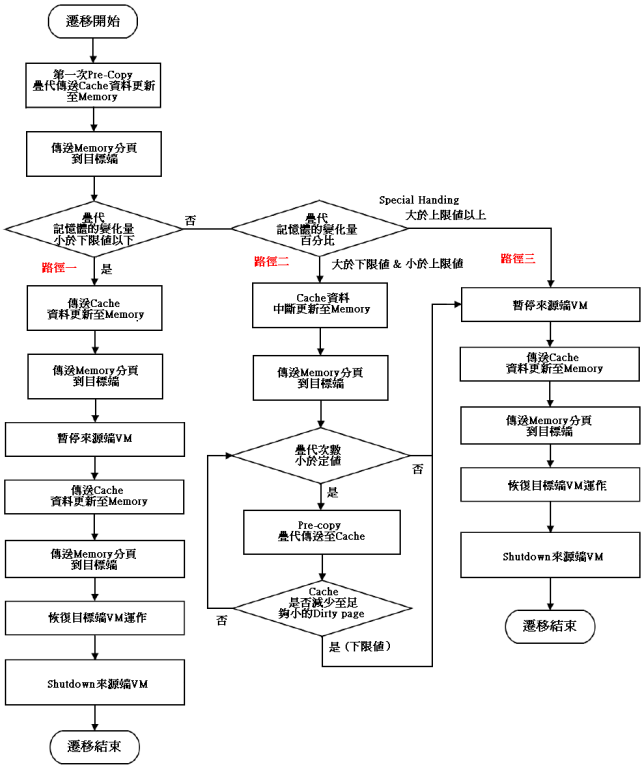
常，再從來源端主機獲取需要的頁面。這樣的好處是可以縮短遷移的時間，但因拖行階段常會出現許多網頁異常現象，當網頁異常現象產生時在網頁傳完成之前系統將無法執行，因此會產生較長的停機時間。

2.5 快取(Cache)

快取原意是指一種存取速度較記憶體更為快的一種記憶體，快取記憶體使用的是較為昂貴也較為快速的靜態隨機存取存儲器（Static Random Access Memory，SRAM)技術。先前在主機操作過的資料會被暫時儲存在快取記憶體中，主機中的處理器在處理資料時，會先到快取記憶體尋找，而不需要去主記憶體中讀取資料。透過資料快取的方式，讓使用者不需每次都要進入主記憶體資料中心讀取被儲存的資料取常用的程式或是資料，而是直接讀取快取記憶體中的常用資料，因此可以縮短整個處理資料的時間。

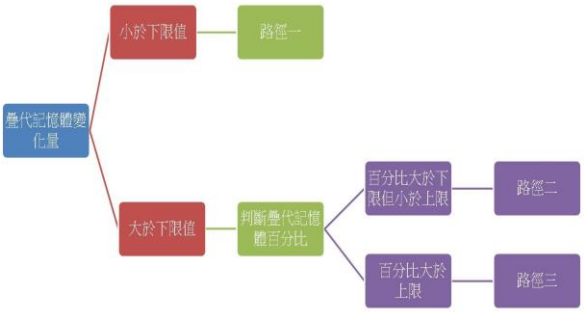
三、 系統設計與架構

在本研究中，我們利用快取融入雲端做虛擬機的系統即時遷移，一共分成三種路徑，以期待同時縮短停機時間及總遷移時間，其系統設計架構如圖七所示。



圖七 快取融入虛擬機系統動態遷移過程

在遷移開始時，先進行第一次的pre-copy，疊代傳送快取資料更新至記憶體(memory)，根據疊代記憶體的變化量來判斷接下來的路徑。若疊代記憶體變化量小於下限值，則進行路徑一；若沒有小於下限值，則判斷疊代記憶體變化量的百分比，若百分比大於下限值但小於上限值則進行路徑二；若百分比大於上限值以上則進行路徑三，系統設計流程示意如圖八所示。



圖八 系統設計流程示意圖

一、路徑一

當疊代記憶體變化量小於下限值(10%)以下時傳送快取資料更新至memory，再將memory分頁傳送至目標端，然後暫停來源端虛擬機，傳送快取資料更新至memory，再將memory分頁傳送至目標端，恢復目標端虛擬機運作，shutdown來源端虛擬機，完成遷移。

二、路徑二

當疊代記憶體變化量大於下限值(10%)但小於上限值(50%)時，則將快取資料中斷更新至memory而將memory分頁傳送至目標端。疊代次數若沒有小於定值則進行路徑三；若小於定值則將pre-copy疊代傳送快取資料更新至memory分頁到目標端，此時若快取沒有減少到足夠小的髒頁(dirty page)則繼續pre-copy疊代傳送快取資料更新至memory分頁到目標端一直到快取減少到足夠小的dirty page(下限值)；若已經減少到下限值時則進行路徑三。

三、路徑三

當疊代記憶體變化量大於上限值(50%)時，則暫停來源端虛擬機，將快取資料更新傳送至memory，再將memory分頁傳送至目標端，恢復目標端虛擬機運作，shutdown來源端虛擬機完成遷移。

四、 範例介紹

4.1 網頁瀏覽

靜態網頁為不包含網頁程式(如 ASP、PHP、ASP.Net…等)及資料庫的純文字及圖片網頁。動態網頁則指有包含 Flash 動畫或 Gif 動畫的網頁，包含網頁程式及資料庫的網頁。Cookies 瀏覽器為了 Web Server 存儲一小段資料訊息。Cookies 動作原理步驟如下圖九所示。

圖九 Cookies 動作原理步驟圖

特性分析：

遷移開始，例如進入奇摩(yahoo)網頁時，點選「新聞」頁面，畫面轉換會將整個頁面刷新，刷新後當滑鼠無點選動作時，則無 Dirty Page。反之滑鼠點選動作時，則會有 Dirty Page，然而產生 Dirty Page 時將會一直刷新快取內資料。過程中第一次 Pre-Copy 疊代傳送快取資料更新至 Memory，且傳送 Memory 分頁到目標端，進入流程圖判斷式，預估第一次疊代記憶體變化量判斷式將小於下限值(10%)以下，所以執行路徑一 (Pre-Copy) 分支。此分支一開始傳送快取資料更新至 Memory，且傳送 Memory 分頁至目標端，目的是讓其 Dirty Page 繼續收斂，而換取縮短遷移時的傳輸時間，再來暫停來源端虛擬機(VM)，傳送快取資料更新至 Memory，再來傳送 Memory 分頁至目標端，恢復目標端虛擬機運作，Shutdown 來源端虛擬機，最後遷移結束。

4.2 矩陣相乘

例如：

$$A=\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}，B=\begin{bmatrix}5 & 7\\ 6 & 8\end{bmatrix}，C=?$$

$$AxB=\begin{bmatrix}1x5+3x6 & 1x7+3x8\\ 2x5+4x6 & 2x7+4x8\end{bmatrix}=\begin{bmatrix}23 & 31\\ 34 & 46\end{bmatrix}$$

特性分析：

遷徙開始，範例 1.矩陣 2x2 中，第一次疊代後將產生三分之一(約 33%)以下 C 矩陣的 Dirty Page，預估再一次疊代也將產生原本三分之一(約 33%)以下 C 矩陣的 Dirty Page，進入流程圖判斷式中，第一次疊代記憶體變化量判斷式預估將大於下限值(10%)&小於上限值(50%)的 Dirty Page 容量。A、B、C 矩陣假設容量都為相同，過程中第一次疊代 VM 需傳輸 3 倍矩陣(A、B、C)容量，疊代後需要再傳輸的資料為 C 矩陣之某一行列相乘之和，所以 Dirty Page 產生必小於單一矩陣容量以下。而 Dirty Page 為某一行程每一列再相加，C 矩陣資訊放置於新的矩陣之行列，而 Dirty Page 可能是三分之一的單一矩陣或其他容量，所以執行路徑二 (Post-Copy&Pre-Copy) 。此分支一開始傳送快取資料中斷更新至 Memory，且傳送 Memory 分頁至目標端，目的是讓 A、B 矩陣資料先行送至目標端。再來進入疊代次數判斷式是否小於定值，如果為「是」，則往下繼續 Pre-Copy 疊代傳送至快取。再來進入判斷式為快取是否減小至足夠小的 Dirty Page，如果為「否」則回到疊代次數判斷式。其目的是讓 C 矩陣所產生 Dirty Page 繼續收斂，而換取縮短遷移時的傳輸時間。如果判斷式中快取是否減小至足夠小的 Dirty Page 為「是(下限值)」，則會與疊代次數判斷式定值為「否」時，一起進入路徑三的起點，暫停來源端虛擬機，傳送快取資料更新至 Memory，再來傳送 Memory 分頁至目標端，恢復目標端虛擬機運作，Shutdown 來源端虛擬機，最後遷移結束。

4.3 Server Game

特性分析：

遷徙開始，當線上遊戲(On-Line Game) 執行時，Server-A 即時遷移至 Server-B 過程中，因為畫面資訊會一直刷新，進入流程圖判斷式，第一次疊代記憶體變化量判斷式預估將大於上限值(50%)以上，所以執行路徑三(Post-Copy) 分支，此分支一開始將暫停來源端虛擬機，傳送快取資料更新至 Memory，再來傳送 Memory 分頁至目標端，恢復目標端虛擬機運作，Shutdown 來源端虛擬機，最後遷移結束。

五、 結果

1. 硬體環境設定：來源端1GB 記憶體、來源端快取 4MB、目標端1GB 記憶體、網路連線速度：100MB/s
2.負載定義：無負載，為基礎系統，在VM中並無安裝應用軟體。
3. I/O 密集型：通過實驗量測 I/O 密集的狀況對 VM 遷移之影響。
4. 內存密集型：在遷移同時，利用一個寫內存的小軟體持續向 VM 之內存持續寫入，每次數量為 VM 內存 512 MB 一半。

利用[8]中數據之轉換，而快取數據傳送為 Memory 之五倍。

路徑一：利用無負載數據，結果如表 I。

表 I 無負載數據結果

Memory (MB)	Pre-copy		
	疊代次數	停機時間 /ms	總遷移時間 /ms
512	3	348	35,175

路徑二：利用 I/O 數據，結果如表 II。

表 II I/O 數據結果

Memory (MB)	Pre-copy		
	疊代次數	停機時間 /ms	總遷移時間 /ms
512	9(定值)	554	19,989

路徑三：利用內存數據，結果如表 III。

表III 內存數據結果

Memory (MB)	Pre-copy		
	疊代次數	停機時間 /ms	總遷移時間 /ms
512	1	129	50,920

在路徑二中因為持續疊代所產生的 Dirty page 一直在快取內更新，資料要傳送至目標端時，才會將快取內的值傳送至 Memory，會與沒快取時傳輸時間上會有明顯差異，計算如下。

路徑二：

沒快取的停機時間:5+5+5*9+5=60⇔1,384

有快取的停機時間:5+5+1*9+5=24⇔554

沒快取的總遷移時間 :5+5+5*9+5=60⇔49,973

有快取的總遷移時間 :5+5+1*9+5=24⇔19,989

結論

本篇論文應用快取提高即時遷移虛擬機器之效能，一共提供三個路徑(路徑一、路徑二、路徑三)，當遷移開始時，第一次 Pre-Copy 疊代後，進入記憶體的變化量判斷式，依照記憶體的變化量之上下限值選擇相對應執行路徑，根據最後可以提高即時遷移之效能。

參考文獻

[1] B. H. Wellenhoff, H. Lichtenegger and J. Collins, Global Positions System: Theory and Practice, Fourth Edition. Springer Verla

[2] Smith, J.E.; Ravi Nair “The architecture of virtual machines,” Computer, Volume 38, Issue 5, May 2005 Page(s): 32-38

[3]D.Z.Blog, "動態遷移原理與詳解", http://findman.blog.51cto.com/438315/260748

[4] E. Anderson, M. Hobbs, K. Keeton, S. Spence, M. Uysal,and A. Veitch.Hippodrome: running circles around storage administration. In Proceedings of the First Usenix Conference on File and Storage Technologies(FAST）, January 2002.

[5] D. Milojicic, F. Douglis, Y. Paindaveine, R. Wheeler, and S. Zhou. Process migration. ACM Computing Surveys, 32(3):241299, 2000.

[6]王選豪," 以混合式複製遷移技術提高虛擬機器即時遷移之效能",大同大學資訊工程研究所碩士論文,中華民國 101 年 1 月.

[7] Jacob G. Hansen and Asger K. Henriksen. Nomadic operating systems. Master's thesis, Dept. of Computer Science, University of Copenhagen, Denmark, 2002.

[8]江雪、李小勇,"虛擬機動態遷移的研究",計算機應用,vol.28,no.9,pp.2375-2385,2008 年 9 月.