

國立交通大學

網路工程研究所

碩士論文



為虛擬遠端桌面環境所發展的效能量測及診斷

工具

Development of a Performance Benchmarking and Diagnostic
Tool for Virtual Desktop Infrastructure

研究 生：張文哲

指 導 教 授：王 協 源 教 授

中 華 民 國 1 0 1 年 7 月

為虛擬遠端桌面環境所發展的效能量測及診斷工具
Development of a Performance Benchmarking and Diagnostic Tool for
Virtual Desktop Infrastructure

研究 生：張文哲

Student : Wen-Jhe Chang

指 導 教 授：王 協 源

Advisor : Shie-Yuan Wang

國 立 交 通 大 學

網 路 工 程 研 究 所

碩 士 論 文



Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

July 2012

Hsinchu, Taiwan, Republic of China

中 華 民 國 101 年 7 月

為虛擬遠端桌面環境所發展的效能量測及診斷工具

學生：張文哲

指導教授：王協源 教授

國立交通大學

網路工程研究所

碩士班

摘要

隨著網路速度越來越快與手持裝置持續進步與發展，使用者操作應用程式的習慣開始有了一些改變，很多廠商開始提出一種虛擬桌面架構，也就是人們手邊不再需要運算速度較快或是儲存容量較大的個人電腦，而只需擁有顯示介面和夠快的網路，就能享有和以前相同或更佳的效能。透過此種架構，程序的執行及檔案儲存都是在伺服器端進行，而使用者端只負責畫面和聲音的接收以及滑鼠或鍵盤的輸入。虛擬遠端桌面很可能在不久的將來，將會徹底改變人們現在使用電腦的習慣，而量測虛擬遠端桌面的效能對於虛擬桌面服務提供業者，將是最主要的一項挑戰，雖然這種架構發展得越來越成熟，但還是鮮少有人研究與開發效能量測工具，我們相信這樣的工具與研究方法對於虛擬桌面中伺服器的佈建或是演算法開發與使用的傳輸協定都有極大的幫助。

在本論文的研究中我們實作了一種評估工具及診斷工具，並設置不同的伺服器資源及網路環境，試圖觀察各種因素對於使用者的效能會有怎麼樣的影響，並利用這些資訊建立資料庫，做為診斷工具的參考指標。我們設置了各種造成使用者操作虛擬桌面效能低落的情境，並利用診斷工具診斷，結果顯示我們的診斷工具具有高準確率且可以診斷同時多種效能低落的情況。

關鍵字：虛擬遠端桌面架構、桌面虛擬化、效能量測、診斷工具

Development of a Performance Benchmarking and Diagnostic Tool for Virtual Desktop Infrastructure

Student: Wen-Jhe Chang

Advisor: Shie-Yuan Wang

Institute of Network Engineering
National Chia Tung University

Abstract

The recent advances in portable devices and the push to transition user's desktop delivery to cloud environments have changed how we use traditional computers today. Many companies start to develop virtual desktop infrastructure. By this technology, people do not need to use a traditional PC with a high clock rate CPU and a large storage device. Instead, they just need to use a portable device with the monitor and network function, but can still finish the same job with better performance. With this architecture, executing programs and saving files all occur in the server. Users just need to receive the screen and voice sent from the server with their portable device. However, there are still many challenges in VDI. One of them is how to estimate the performance. With the performance benchmarking tool developed in this thesis, people can estimate the performance of the algorithm or transport protocol used in a VDI. We also develop a diagnostic tool for VDI. This tool can help users find out the factors that make them experience bad performance while using VDI.

In this paper we develop a diagnostic and performance benchmarking tool for VDI. First we execute our performance benchmarking tool under different server loading and network conditions. Then, we use the measured data to develop our diagnostic tool. The experiment result shows that our diagnostic tool could find out the factors that make users experience bad VDI performance.

Keywords: Virtual Desktop Infrastructure, Desktop Virtualization, Performance Benchmarking, Diagnostic Tool

誌 謝 辭

首先，我要感謝我的指導教授王協源老師兩年來的指導，老師除了在專業領域上給了我很多的教導，包含了做研究的方法、態度，都讓我成長許多，使我在課業、研究或生活上都學到了許多寶貴的經驗和知識，相信這些學習的過程對未來一定有很大的幫助。

此外，我也要感謝郭耀煌老師、周承復老師、趙禧綠老師，感謝你們撥冗蒞臨交通大學擔任這次碩士論文口試的口試委員，提供了我許多寶貴的意見，讓我也能夠更加強不足及缺失之處。

另外，我也很感謝實驗室學長姐、同學及學弟妹們，除了平常生活的照料，以及課業研究上面的幫助，都讓我在兩年的碩士生涯中覺得豐富而且精彩，也勉勵學弟妹們能夠繼續努力，並期許你們都能夠順利完成學業。

最後，我要感謝我的家人，謝謝你們在我求學過程中的支持，讓我也能無後顧之憂的專心在課業上，很開心沒辜負你們的期望，能夠順利的完成碩士學程。

目錄

目錄	I
圖目錄	VI
表目錄	VII
一、介紹	1
1.1 研究動機與研究目標	1
1.2 章節概述	2
二、背景	3
2.1 相關研究	3
2.2 Virtual Desktop Infrastructure (VDI) 平台介紹	4
2.2.1 Virtual Desktop Infrastructure (VDI) 介紹	4
2.2.2 Citrix XenDesktop 介紹	6
2.3 EstiNet 網路模擬器介紹	8
2.3.1 EstiNet 網路模擬器	8
2.3.2 EstiNet Emulator	11
三、VDI 評估工具開發	12
3.1 開發動機	12
3.2 評估工具設計與架構說明	12
3.3 評估工具實作	14
四、VDI 診斷工具開發	20
4.1 開發動機	20
4.2 診斷工具設計與架構說明	20
4.3 診斷工具實作	22
五、評估工具實驗環境與實驗結果	26
5.1 研究與實驗方式	26

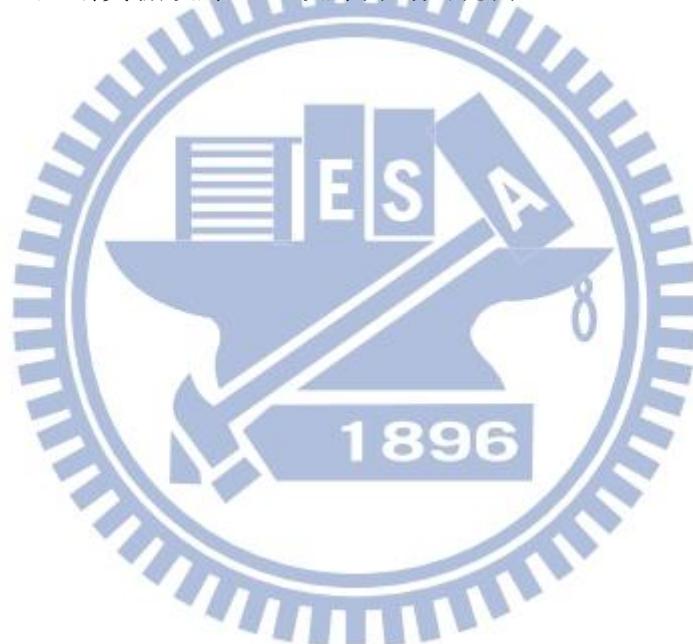
5.2 實驗情境參數	27
5.2.1 處理器資源	27
5.2.2 記憶體資源	27
5.2.3 磁碟資源	27
5.2.4 線路延遲	27
5.2.5 封包遺失率	28
5.2.6 實驗環境說明	28
5.3 效能實驗設計	30
5.3.1 評估實驗設計 A	30
5.3.2 評估實驗設計 B	31
5.3.3 評估實驗設計 C	32
5.4 評估實驗結果與分析	33
5.4.1 評估實驗設計 A 之結果	33
5.4.2 評估實驗設計 B 之結果	38
5.4.3 評估實驗設計 C 之結果	43
六、診斷工具實驗環境與實驗結果	46
6.1 研究與實驗方式	46
6.2 實驗情境參數	46
6.3 診斷實驗設計	48
6.3.1 診斷實驗設計 A	48
6.3.2 診斷實驗設計 B	48
6.4 診斷實驗結果與分析	50
6.4.1 診斷實驗設計 A 之結果	50
6.4.2 診斷實驗設計 B 之結果	53
七、結論	55
八、未來工作	56
參考文獻	57

圖 目 錄

圖 1、XenDesktop 平台示意圖	7
圖 2、kernel re-entering 模擬方法	9
圖 3、EstiNet 架構圖	10
圖 4、畫面捕捉示意圖	14
圖 5、伺服器與接收裝置互動情形	15
圖 6、評估工具執行流程圖	17
圖 7、動作執行前 XenDesktop Receiver 畫面	19
圖 8、動作執行後，預期畫面呈現結果	19
圖 9、實驗環境說明	29
圖 10、實驗設計 A，伺服器處理器資源	34
圖 11、實驗設計 A，伺服器記憶體資源	34
圖 12、實驗設計 A，伺服器磁碟資源	36
圖 13、實驗設計 A，線路延遲	36
圖 14、實驗設計 A，封包遺失率	37
圖 15、實驗設計 B，伺服器記憶體資源	39
圖 16、實驗設計 B，伺服器處理器資源	39
圖 17、實驗設計 B，伺服器磁碟資源	41
圖 18、實驗設計 B，線路延遲	41
圖 19、實驗設計 B，封包遺失率	42
圖 20、實驗設計 C，伺服器處理器資源	44
圖 21、實驗設計 C，伺服器記憶體資源	44
圖 22、實驗設計 C，伺服器磁碟資源	45
圖 23、診斷結果說明	47
圖 24、診斷實驗設計 A、記憶體資源不足	50
圖 25、診斷實驗設計 A、處理器資源不足	51
圖 26、診斷實驗設計 A、磁碟資源不足	51
圖 27、診斷實驗 A、線路延遲過長	52
圖 28、診斷實驗 A、封包遺失率過高	52
圖 29、診斷實驗 B、處理器資源不足與線路延遲過長	53
圖 30、診斷實驗 B、磁碟資源不足與封包遺失率過高	54
圖 31、診斷實驗 B、記憶體資源不足與線路延遲過高	54

表目錄

表 1、評估工具實驗相關診斷資料	23
表 2、伺服器資源相關診斷資料	24
表 3、診斷工具診斷依據表	25
表 4、實驗參數分類表	26
表 5、實驗設計 A 之設計目的與說明	30
表 6、實驗設計 B 之設計目的與說明	31
表 7、實驗設計 C 之設計目的與說明	32
表 8、診斷實驗設計 A 之設計目的與說明	49
表 9、診斷實驗設計 B 之設計目的與說明	49



一、介紹

1.1 研究動機與研究目標

在網路越來越快且普及的現在，使用者操作應用程式的習慣開始有了一些改變，很多廠商開始提出一種 VDI (Virtual Desktop Infrastructure)架構，也就是人們手邊不再需要運算速度較快或是儲存容量較大的個人電腦，而只需擁有顯示介面及夠快的網路，就能享有和以前相同或更佳的效能，透過此種架構，程序的運算及檔案儲存都是在伺服器端進行，而使用者端只負責畫面和聲音的接收以及滑鼠或鍵盤的輸入，而且對於硬體資源及電源消耗比以前使用傳統個人電腦時更少，因為由中央伺服器集中管理這些虛擬桌上型電腦，等使用者想使用的時候再進行分配即可。

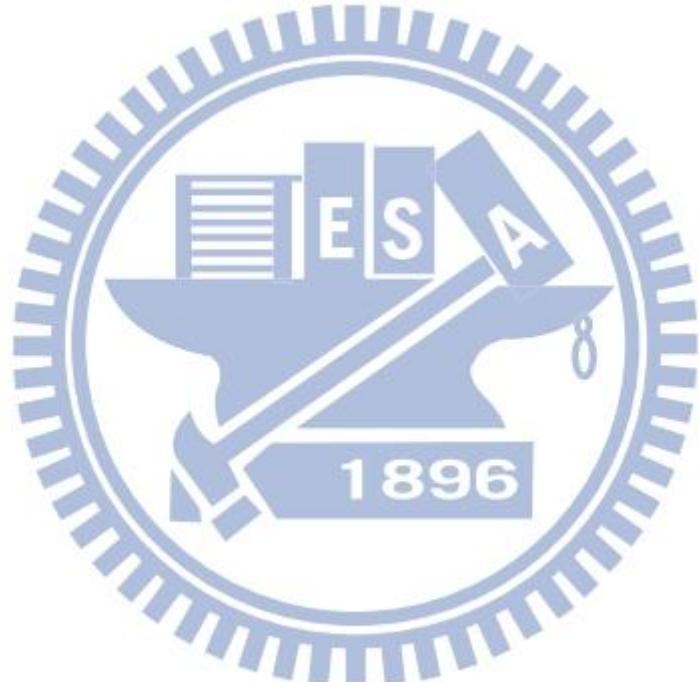
但在此架構下，如何能夠正確的評估(Benchmarking)使用者的感受，或是建立一個診斷工具，在使用者感覺效能不好時，能讓伺服器端查覺並及時改善，對提供服務的伺服器端是很重要的一項研究。更進一步來說，使用者也需要這樣一個工具，當感覺效能不好時，能夠利用此工具精準的診斷出造成延遲的原因，以便讓使用者知道是何種原因造成此種情況，伺服器端也同樣的需要此種工具，因為如果遇到使用者反映在使用 VDI 時，有畫面延遲或停頓等情形，能夠利用診斷工具回報是何種原因造成延遲，讓伺服器端能夠即時的調整硬體資源，解決使用者的延遲現象，或是將原因歸咎於 ISP (Internet Service Provider)業者，所以如果有這樣一個工具能夠讓 VDI 未來發展更順利也更值得信賴。

但對於如何彈性以及有效率的評估一個 VDI 的好壞，或是進一步開發出一個診斷工具是一個很大的挑戰，VDI 比傳統桌面更難評估及診斷有以下原因，首先要高速重播使用者的動作是困難的，因為使用者送出下一個動作之前，必須先確認對應前一個動作的畫面已經正確呈現，否則將造成錯誤的結果，第二是 VDI 的效能好壞比起傳統的個人電腦更難定義，因為不只時間上的考量，還有接收到畫面的品質，第三是造成使用者感覺效能不好的原因可能有很多種，比起以往的個人電腦將更複雜，所以如何能剖析出何種因素是造成延遲的主要原因也會是很大的一個挑戰。

在本篇論文當中，使用的 VDI 平台為 Citrix 所提出的 XenDesktop 商業版，將在此平台提出一種方法來建立 Benchmark，並更進一步利用蒐集到的資訊來建立一個診斷工具，並營造出不同的惡劣網路環境及系統負擔，先進行 XenDesktop 的效能量測，再診斷出造效能不佳的原因為何。

1.2 章節概述

本篇論文的章節組成如下，第二章將介紹與 VDI 效能量測相關的研究與 EstiNet 網路模擬器介紹，讓讀者能夠更加了解 VDI 平台和我們的實驗環境，以期能夠了解我們之後的實作想法；第三章與第四章將分別先介紹我們如何實作評估工具與診斷工具；第五章及第六章我們先對實驗環境及參數做介紹，之後再對我們的實驗結果進行分析及討論，第七章為我們這篇論文下個結論；第八章則是提述在撰寫論文的過程中，我們所發現未來能夠加強之處。



二、背景

我們在此篇論文主要開發一個診斷工具，來評估造成使用者感覺效能不好最主要的原因，但評估原因的過程與量測 VDI 平台效能息息相關，為了量測 VDI 平台效能，不同的研究機構提出了不同的 Benchmark 方法，在此章節中我們將介紹與 VDI 量測相關的研究，接著將介紹我們所選擇的 VDI 平台— Citrix XenDesktop，以及為了產生不同網路環境所使用的 EstiNet 模擬器。

2.1 相關研究

隨著網路速度發展越來越快速的現在，VDI 也開始受到研究機構及企業的重視，為了能夠準確量測 VDI 平台的效能，各家研究機構也紛紛提出不同的方法在各種 VDI 平台上。在[1]中，作者提出了一種評估 VDI 平台的方法—DeskBench，所評估的 VDI 平台為 VMWare ESX，作者藉著錄製及重播使用者的操作來產生滑鼠或是鍵盤的動作，配合 Linux OS 上的 XLib 常式庫，判斷畫面是否已經正確呈現，藉由一邊重複播放使用者行為，一邊監控畫面呈現來量測 VDI 平台的效能，作者在 ESX 伺服器上安裝了八台虛擬機器，所使用的作業系統為 Microsoft Windows XP，並在另一台伺服器上遠端登入並執行 DeskBench 工具來量測效能，但作者只對使用者同時登入數量作為系統負荷的指標，沒有更進一步的探討各式各樣的系統負擔及網路環境會對使用者將會造成什麼影響。

另一個也很相近的研究是[2]和[3]，作者提出了 VDBench 在 VMware ESXi 4.0 上進行評估，作者在伺服器端執行腳本來產生各式各樣的使用者操作，以及在使用者與伺服器之間製造不同惡劣程度的網路環境，然後提出一個 Slow-Motion 的量測方法，這方法是將伺服器傳給使用者端的封包捕捉，然後再以慢速播放，如此一來便可解決使用者與伺服器端的同步問題，也可量測對於使用者發出的輸入，伺服器端花了多少時間來進行畫面的回覆，這個方法好處是不需在用戶端執行任何程式，但缺點是無法以高速度產生使用者的操作，輸入是由伺服器端自己產生，評估上有可能會忽略了惡劣的網路環境對使用者造成的影响，也只對使用者同時登入數量作為系統負荷的指標，並沒有更細部的探討其他系統因素對使用者所造成的影响。

此外像[8][9][10]也是做效能量測相關的研究，但不是在 VDI 平台上，而現在也尚未有人開始研究 VDI 環境的診斷工具相關議題。

2.2 Virtual Desktop Infrastructure (VDI) 平台介紹

2.2.1 Virtual Desktop Infrastructure (VDI) 介紹

桌面虛擬化技術(Desktop Virtualization)是一種將傳統個人電腦(PC)從實體機器轉換成 Client-Server 架構的一種技術。更進一步的定義桌面虛擬化，桌面虛擬化是包含封裝或是交付一個系統環境存取，用戶端的作業系統及硬體架構可以與預計使用的桌面環境完全不同，Desktop Virtualization 這種架構可以允許多個使用者各自使用自己的桌面環境在一個單一集中的電腦或是伺服器上，而這些中央伺服器可能位於商業或是數據中心，而使用者在地理上可以是分散的位於各個不同的地方，但都必須透過區域網路，廣域網路，或在公共互聯網連接到中央伺服器。

而 Virtual Desktop Infrastructure (VDI)，有時也稱做 Virtual Desktop Interface，此種架構是一種主要運算都交給伺服器來進行，並且提供桌面虛擬化技術，包含軟體及硬體系統需要支援虛擬的環境。許多企業都開始使用這項技術，使用者在公司透過網路使用中央伺服器所提供的虛擬桌面，使用的過程中使用對應用程式的修改或是正在開啟中的程序的狀態及資料都會儲存在中央伺服器上，而在 VDI 架構下這些資料或作業系統的狀態，都不是儲存在使用者的實體機器上，而是透過網路在操作時就儲存在中央伺服器上，如此一來，使用者在任何地方都可以使用像智慧型手機或是隨身的裝置存取，在不同地方但是都能執行相同的環境的作業系統。

目前已有很多研究機構或是企業在 VDI 發展，如 Citrix 推出的 VDI 產品叫做 XenDesktop、Microsoft 的 virtual desktop infrastructure in Windows Server 2008、Oracle 的 Oracle Virtual Desktop Infrastructure、Red Hat 的 Red Hat Enterprise Virtualization RHEV、Univention 的 UCS Desktop Virtualization Services (UCS DVS)、VMware 的 VMware View，由此可以看出 VDI 技術隨著網路與硬體資源的發展，已經受到很多國際級的企業及研究機構重視。

VDI 架構在共享資源的部份承襲了桌面虛擬化原本已有的優點，而且其中每一台電腦都有著自己的作業系統、應用程式、或是周邊設備，而且整體對於硬體資源的消耗可能可以減少，因為每位使用者可以根據他們的需要來進行分配及可，桌面虛擬化潛在的提高了使用者資料的完整性，因為所有資料都可以在中央伺服器維護及備份。

桌面虛擬化潛在的優點包含：當要提供一個新的 Desktop 變的更容易、降低部屬一個新的應用程式所需的花費、管理 Desktop Image 的能力較強及簡單、因為將資料儲存在中央伺服器，加強了資料的安全性、可以更安全的遠端存取位於公司裡的 Desktop。

雖然好處很多但也因此多了一些限制，因為虛擬化所以即使在相同硬體資源下，效能可能會比直接使用傳統 Desktop 低落、如果網路上沒有適當的設定也有安全上的疑慮、而設定周邊設備的驅動程式也是一大挑戰(如印表機等等)，某些應用程式在桌面虛擬化架構下執行變得困難(如多媒體)，由於網路的問題可能增加了系統無法正常運作的時間、需要在有網路的環境才能正常使用、在 VDI 架構下環境的部屬及管理也是一大花費。

VDI 架構基本上由於使用的方式不同主要分成四種類型

- Hosted (delivered as a service)
- Centralized
- Remote Synchronization
- Client-Hosted

Hosted 和 Centralized 都必須靠著網路連接到中央伺服器，這兩種模式的概念都很像 Thin Client 架構，使用者的裝置只是負責呈現由中央伺服器所送來的畫面，因此一定需要在有網路的環境。Remote Synchronization 允許使用者將 Desktop 複製到自己的裝置上，因此就可以不透過網路來執行虛擬桌面，使用者平常可能是像一般集中式 VDI 使用虛擬桌面，但當需要移動在無網路環境下就可 Desktop 複製到自己的裝置上。Client-hosted 只是由中央伺服器來管理虛擬機的映像檔，虛擬機的執行則由用戶端的裝置負責運行。

在本篇論文中所採用的 Citrix XenDesktop 使用的是 Centralized 架構，在這種模式下，虛擬桌面及使用者的數據資料都是由一台或是多台中央伺服器負責管理，在這種架構下，又可細分成兩種類型，靜態或是動態，在靜態類型中，使用者與虛擬機器是一對一對應的，每個使用者都會取得唯一的一台虛擬機器，而在動態類型中，存在一個主要的映像檔，使用者自己的資料則是與映像檔分開儲存，當使用者需要使用虛擬機器時，在將主要映像檔複製一份，並結合使用者自己的資料部份。

2.2.2 Citrix XenDesktop 介紹

在本論文中所用來診斷及評估的 VDI 平台為 Citrix 公司的 XenDesktop，Citrix 公司始建於 1989 年由前 IBM 開發 Ed Iacobucci 斥資三億美元所創立，此平台在全球各地都已有企業開始使用，企圖將桌面環境進行轉型，從以設備為中心的管理轉變為以使用者為中心的私有雲端，將桌面和應用程式作為一種服務，當使用者有需求時才傳送給使用者。以下將對 XenDesktop 平台做一個概括性的介紹。

Citrix XenDesktop 是 Citrix 公司推出的桌面虛擬化平台，把虛擬桌面當成一種服務透過網路傳送給任何地方的使用者，Citrix 公司宣稱，如此一來比起傳統管理桌面的方式，XenDesktop 可將成本降低高達 40%。

隨著使用時間的增加，傳統桌面 (Desktop) 管理上變得更加複雜，且成本上也是一大考量。一直以來，因為每台設備都必須個別的更新檢查及修復，且如果有最新版本的作業系統或是應用程式，也必須在每台設備上個別安裝，還有資料備份上比起虛擬桌面也是困難許多，XenDesktop 有著桌面虛擬化的好處，能夠克服這個問題。

XenDesktop 將透過網路連接把桌面當成一種服務，讓使用者可以得到高解析度使用環境和個人化的桌面。有別於其他桌面虛擬化平台，XenDesktop 將個人化桌面影像化再傳送給各地使用者，因而比起傳統桌面能簡化桌面管理，內建有虛擬桌面效能監控工具也比起傳統桌面更能掌握使用者的使用狀況。

XenDesktop 利用 Citrix HDX 技術，在各種網路情況下包含高延遲網路情況下，使用者都還是享受高解析度使用環境。HDX 技術支援圖形、多媒體、語音和週邊設備的，最佳化圖形、多媒體、VoIP Soft Phone 和 USB 設備，這些技術是為了應付各種惡劣的網路環境及中央伺服器的效能低落時，對於使用者使用虛擬桌面及其應用程式都還是能盡量讓使用者察覺且正常使用。XenDesktop 可為隨時為使用者傳送個人化設定的作業系統和應用程式的虛擬桌面，不管使用者使用何種裝置接收由中央伺服器傳送的畫面。XenDesktop 提供使用者高效能的個人化虛擬桌面，虛擬桌面作業系統的版本可隨時更新，供使用者隨時隨地存取。

透過建立 Active Directory 使用者帳號，並將帳號連結虛擬桌面映像檔，當有新的使用者加入，XenDesktop 比起傳統桌面可快速、簡易地設置新使用者虛擬桌面。當應用程式或作業系統需要更新時，只需更新一個桌面映像檔，因此可快速更新及修補作業系統和應用程式，大大減輕桌面管理的負擔。

在 XenDesktop 主要可分成兩大部分，中央伺服器及使用者裝置，在中央伺服器部分主要負責管理及派送虛擬桌面，而使用者部分只負責接收由中央伺服器傳來的影像及聲音，虛擬桌面由使用者個人設定的資料、安裝在作業系統上的應用程式、加上作業系統所組成，使用者需要使用時在透過 Citrix ICA 傳遞協定傳送到使用者裝置。



圖 1、XenDesktop 平台示意圖



2.3 EstiNet 網路模擬器介紹

2.3.1 EstiNet 網路模擬器

在本論文中實驗中為了測量 Citrix XenDesktop 在不同網路環境下對於畫面的反應時間，因此需要引入網路模擬器做為變化網路情境的工具，在本論文中使用的是思銳科技公司所開發的 EstiNet 網路模擬器，EstiNet 的前身是NCTUns[4][5][6]，是一款具有高真實性和擴充性的網路模擬器和擬真器，具有模擬各種有線和無線網路設備和協定的能力，其核心技術是利用 Kernel re-entering 方法，這套方法是交通大學王協源教授 1990 年在美國哈佛大學時所設計的。由於這套創新的方法，EstiNet 有著許多獨一無二的優點，這些獨特的優點是傳統的網路模擬器無法達到的，像是 ns2、OPNET。EstiNet 還具有以下優點。

1. 利用真實世界中的傳輸層與網路層的協定堆疊 (Protocol Stack)，能產生更精準的模擬結果。
2. 真實世界的應用程式可以不經修改直接運行於此模擬器上。
3. 具有高度整合性的使用者圖型化介面 (Graphical User Interface)，提供便捷的操作方式進行模擬情境的設計、執行與模擬結果檢視。

Kernel re-entering 方法如圖 2 所示，透過這個方法 EstiNet 在模擬的時候，可以利用真實世界中 Linux kernel 的 TCP/IP 協定來協助模擬，因此可以得到更真實的模擬結果。在圖 2 中，TCP/IP 協定堆疊是真實世界 Linux kernel 的協定。雖然圖 2 中有兩個 TCP/IP 協定堆疊，但它們都是表示同一個 Linux kernel 的協定堆疊。而通道介面 (tunnel interface)可以看成是一個虛擬的網路卡介面 (pseudo network interface)，但是沒有對應到真實世界的網路裝置。由於我們的通道介面 1 設定成封包的輸出設備，所以封包會被插入到通道介面 1 的 output queue 裡面。模擬引擎將立即偵測到這個事件，並透過執行 read() 系統呼叫，從通道介面 1 獲得這個封包。封包在模擬引擎中會模擬 transmission delay 和 propagation delay 的情況，之後模擬引擎會透過執行 write() 系統呼叫將封包插入通道介面 2 的 input queue 裡面。然後，核心會藉由執行軟體中斷，並把封包傳到 TCP/IP 協定堆疊。接著將封包放入 TCP 接收端的 receiver queue 裡面。最後，TCP 接收端執行 read() 系統呼叫接收封包。

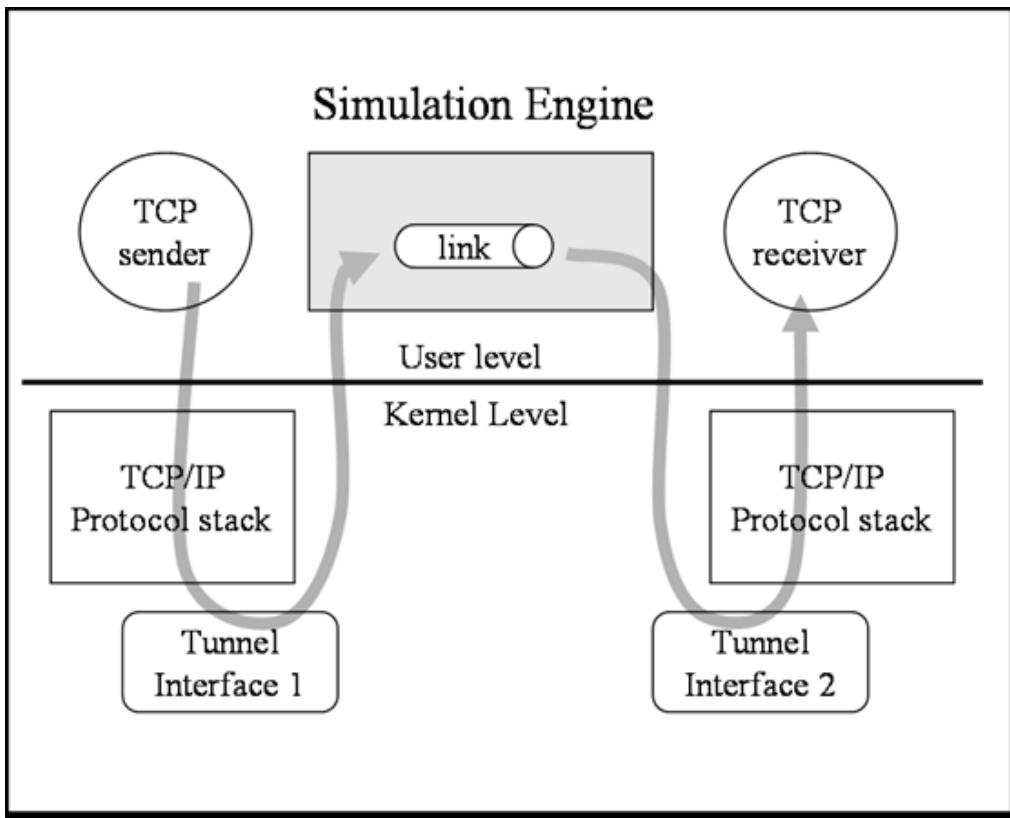


圖 2、Kernel re-entering 模擬方法



EstiNet 網路模擬器的架構如圖 3 所示，裡面包含了幾個基本元件，分別是模擬引擎 (Simulation Engine)、協定模組 (Protocol Module) 及圖形化使用者介面 (Graphical User Interface)。模擬引擎可以想成是一個小型的作業系統核心，它是由許多複雜的功能所組成，包含事件的排程、模擬時間的管理、行程通訊…等，它也提供許多 API 程式，協定模組可以藉由執行這些 API 程式取得系統資訊和要求模擬引擎提供服務。EstiNet 網路模擬器採用模組化設計，每一種網路型態設備的協定堆疊採用了模組化設計，像是 ARP 模組則是對應到 ARP 協定，而 FIFO 模組則是實作 FIFO 封包排程和緩衝區管理。因此當有新的網路型態出現之後，只需要實作並更新其內部的協定模組，就可以對新的網路型態進行模擬。圖形化使用者介面提供使用者一個方便使用的網路模擬環境，使用者可以簡單隨意地規劃出想要模擬的網路拓樸。此外 EstiNet 對 Linux 的核心做了一些修改，使得利用 Socket 介面所撰寫的網路應用程式，不需要經過任何修改就可以直接在 EstiNet 上面執行。

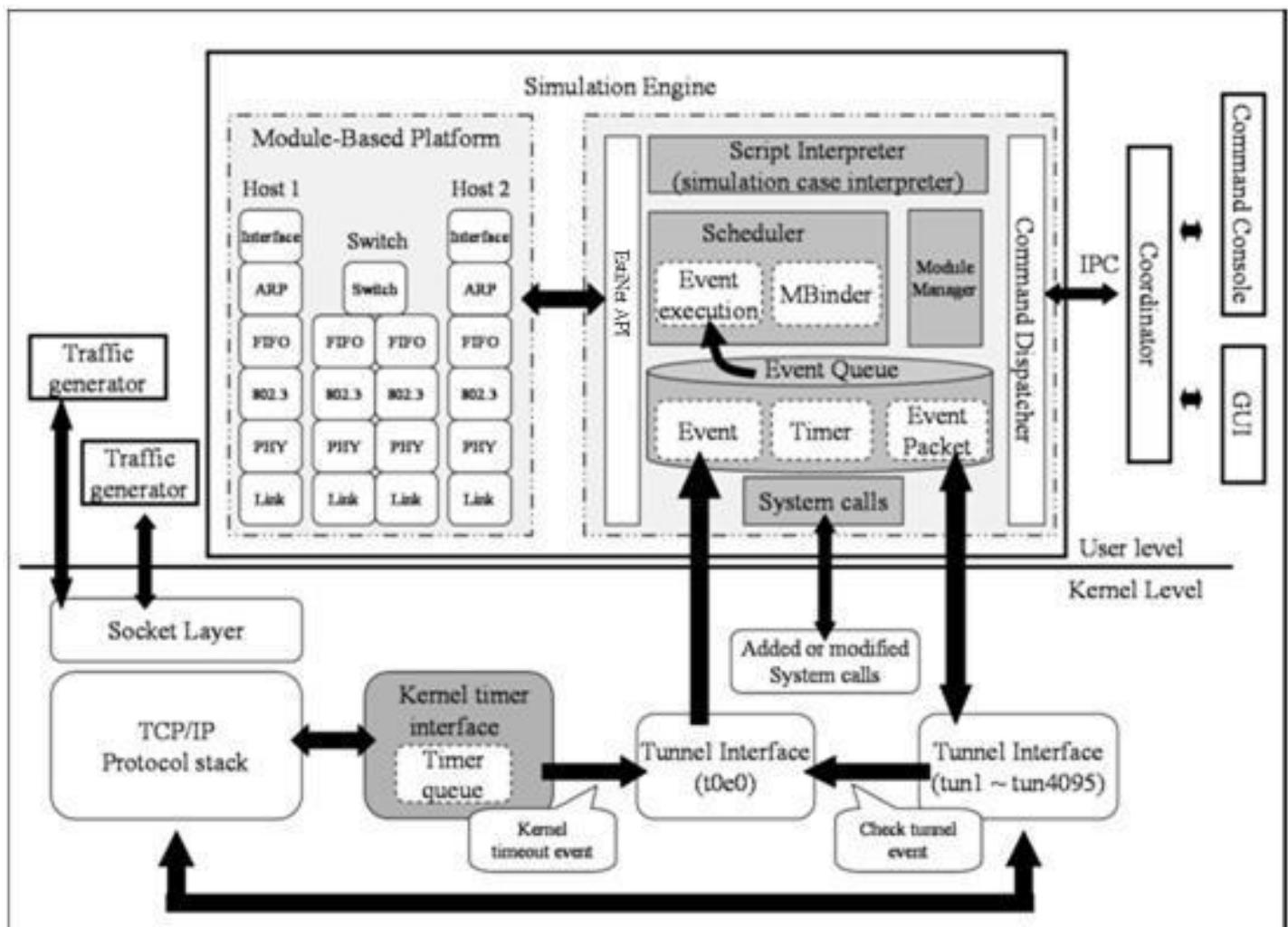


圖 3、EstiNet 架構圖

2.3.2 EstiNet Emulator

EstiNet 網路模擬器可以輕易的轉換成 Emulator，Emulator 是指 EstiNet 裡模擬的網路環境可以和真實世界的裝置互動，強迫真實世界的封包導入使用者指定的虛擬網路環境裡。因此 EstiNet Emulator 拿來測試真實世界的應用程式的效能或是真實世界的裝置在各式各樣的網路環境下是非常有用的。在本論文中就是使用這個特別的功能，來產生各式各樣的網路環境，來觀察 Citrix XenDesktop 的反應會是如何。

在 EstiNet Emulation 環境中，一個外部的裝置可以和模擬網路環境裡的裝置交換 TCP/UDP/ICMP 封包，除此之外還有一個用途，兩個真實世界的裝置也可在模擬的網路環境下交換它們的封包，舉例來說，兩個裝置可以建立一條 TCP 連線，並且設定它們封包傳送的路徑經過設定好的模擬網路環境下。Emulator 提供了許多好處，像是真實世界的裝置可以和模擬網路互動，還有可以設定真實世界網路封包的遺失率、延遲、重排、排程方法等等…，因此我們可以拿來測試應用程式或是裝置在各式各樣網路拓樸或是網路狀況下的表現如何。

EstiNet 使用一個模擬核心模組(Emulation Kernel Module)來橋接真實世界的裝置與模擬的網路環境，此模組最主要的功能是執行了 Network Address Translation (NAT)的功能，它將使用者裝置輸出或輸入的封包攔截，並適當的修改此封包IP及Port，最後再將其導入模擬的網路環境中，最後再根據封包的目的地將它導到虛擬網路或是真實世界的裝置上。在設定完模擬的拓樸之後，在 EstiNet 網路模擬器裡有一支叫做EstiNetclient 程式，會自動的產生一些命令來設定模擬核心模組，然後當模擬開始時，EstiNet 模擬引擎會自動的觸發Linux Kernel 載入及透過系統呼叫來初始化模擬核心模組。此外由於模擬核心模組會消耗一些CPU時脈來執行NAT功能，所以為了達到更精準的模擬結果，執行模擬的機器規格需要一定程度。EstiNet更詳細的實作及設計方式可參考[7]。

三、VDI 評估工具開發

3.1 開發動機

由於我們的最終目標是開發出一個 Citrix XenDesktop 上的診斷工具，因此先開發一個評估(Benchmark)工具是很重要的，因為我們必須先量測此 VDI 平台在不同的系統負擔或是網路環境下，使用者的效能感受會是如何。也就是說觀察到使用者對於不同狀況所接收到的效能表現，診斷時才有所依據，能分辨使用者接收畫面時的效能是好或壞。所以我們先設計出一個 Citrix XenDesktop 平台上的評估工具，再藉由評估工具所量測到的資料來設計診斷工具，而評估工具所得的資料也必須精準，否則將會導致診斷結果錯誤。

3.2 評估工具設計與架構說明

根據我們觀察了 Citrix XenDesktop 使用者使用此 VDI 平台時的操作，對於使用者而言，只有鍵盤或是滑鼠的輸出，且在意的只有由伺服器是否能及時的回送對應的畫面。而影響伺服器端的效能的因素，除了伺服器本身的效能以外，還有網路環境因素。我們開發的評估工具必須能自動的產生模仿使用者使用 VDI 平台的操作，也就是滑鼠左右鍵的點擊或是輸入鍵盤的動作，且還必須能判斷送出的動作以後，由伺服器傳送回來對應此動作的畫面是否已正確呈現。在使用 VDI 平台時畫面已正確呈現的時間，可以說是使用者在使用 VDI 平台時最在意的效能指標，如果呈現的時間夠短的話，使用者操作時的感受與在本地執行無太大差別。

我們實作的評估工具的方法是，利用 Microsoft 所開發的程式語言 C# 提供的 API 來產生使用者滑鼠或是鍵盤的操作，也可以捕捉由伺服器傳送給使用者的畫面。我們觀察了使用者一般使用虛擬桌面的情形，並且建立了一連串的滑鼠及鍵盤的操作行為來模仿使用者使用虛擬桌面的行為，但我們在設計完使用者的動作後發現，這些滑鼠及鍵盤的執行間隔是一大問題，如果以程式能執行的最快速度執行，將會有可能發生以下情形。

在使用者還沒接收到對應的畫面，例如：第一個動作是開啟 Microsoft Word 應用程式，而第二個動作是在開啟好的 Microsoft Word 應用程式上鍵入文字，但如果以最快的速度執行，將產生結果與預期的不符合，也就是開啟的 Microsoft Word 應用程式上並未正確的鍵入文字。第二種方法是先量測 Microsoft Word 應用程式需要開啟的時間，同樣在上述情況下，第一個開啟動作執行完後，等待量測的時間，接著再執行鍵入的動作，如此一來看似可避免掉

結果的不正確，但開啟 Microsoft Word 應用程式的時間可能隨著伺服器的負擔而有所變化，如果把估計開啟的時間增加，可以有更高的機率避免結果錯誤，但產生的代價就是因為增加估計時間而產生的誤差，會造成模仿使用者行為的不正確，且對於每個動作都需一一預估也是一大困難，經過以上討論我們決定採用更智慧的方法。

我們的做法是先將所有鍵盤及滑鼠的動作分成兩類，需要等待特定畫面呈現之後才能執行的動作，我們稱之為需要建立同步的動作，反之則是不需要建立同步的動作，不需要等待特定畫面的動作之前，執行之前可選擇是否加上一段等待時間，此等待時間是模擬使用者在操作時的思考時間，而對於需要建立同步的動作，我們會先將動作執行完的畫面記錄下來，且在需建立同步的動作之後的下一個動作，都必須等到之前先記錄下的畫面與現在使用者接收到的畫面一致，才能執行下一個動作，如此一來便可確保執行動作後畫面呈現結果正確。

所以在我們的評估工具實驗中，將會根據以下步驟來執行

1. 規劃一個實驗裡需要有幾個滑鼠或鍵盤，並記錄每個動作的在螢幕上的座標位置。
2. 將每個動作分類，分成需要建立同步的動作以及不需要建立同步的動作。
3. 把需要建立同步的動作，動作執行完預期畫面儲存。
4. 根據不同實驗設定不同的伺服器負擔或是網路環境。
5. 根據步驟 1 所記錄的動作開始執行，若此動作是需要建立同步的動作則做步驟 6，否則直接執行此動作之下一個動作。
6. 若遇到需建立同步動作，則開始捕捉 XenDesktop Receiver 應用程式裡所呈現的畫面，直到畫面與步驟 3 與此動作紀錄的畫面相同。
7. 確認所有動作是否已結束，並記錄從執行動作後到確認畫面相同的時間，若尚未全部結束則繼續步驟 5。

3.3 評估工具實作

在這一小節我們將詳細說明評估工具的運作細節，以及實作後遭遇的問題，並提出解決方法，我們開發選擇的平台是 Microsoft C#，並利用其提供的函數 User32.dll 來實作我們所需的操作，像是鍵盤滑鼠或是畫面捕捉等等動作。

首先我們對需要建立同步的動作進行動作完成的畫面的擷取，我們先藉由接收裝置上 Citrix XenDesktop Receiver 上得到使用者畫面呈現的座標位置，再將座標位置所包含的區域暫存，做為之後動作執行完比較之用，如圖 4 所示，此畫面為接收裝置在開啟 Citrix XenDesktop Receiver 時的桌面畫面，右下角是 Receiver 應用程式部分，而我們只在意 Receiver 內的內容，所以藉由提取應用程式的座標來達成。



圖 4、畫面捕捉示意圖

接著我們藉由圖 5 來簡單分析在使用者在使用 Citrix XenDesktop 時伺服器與接收裝置互動情形，來分析評估工具何時能執行正確的動作，虛線表示的是由接收裝置送給伺服器的封包，主要是滑鼠及鍵盤的操作，細短的實線是由伺服器發送給接收裝置的封包，也就是傳送給 Citrix XenDesktop Receiver 應用程式的畫面更新，而粗長的實線是指確認需要建立同步之動作執行之後，XenDesktop Receiver 應用程式的畫面與先前儲存的畫面已相符，之後的動作便可接續執行。以下將舉一個簡單實例，在 tcc 時刻之前，接收裝置已執行許多操作，伺服器端也送回對應的畫面給接收裝置，此時都尚未有同步動作出現，而在 tcc 此刻接收

裝置此使是開啟 Microsoft Word 應用程式，根據實驗設計，此動作是一需建立同步動作，所以評估工具將暫停執行下一動作，並且持續比對 XenDesktop Receiver 應用程式的畫面與先前對應此動作所儲存的畫面是否相符，在 t3 伺服器端送出對應接收裝置開啟 Microsoft Word 應用程式的畫面，而時間 t4 才比對成功，此時 t4 稱為同步點，同步點之後才可執行下一動作，t3 減去 tcc 的時間便是精準由接收裝置執行動作後，畫面呈現所需的時間，但是由於我們無法正確掌握伺服器何時將畫面回傳，因此 t4 減去 tcc 的時間便是我們量測到接收裝置送出動作後，伺服器回傳畫面所需的時間，此時間與我們比對面畫是否相同的頻率有很大的關係，根據我們的設計，量測的頻率約 100 毫秒一次，因此我們所測的畫面回應時間可能約有 0 至 100 毫秒的誤差，但此誤差我們認為是非常小的，認為算是使用者可以接受的誤差，等到後面實驗我們會做簡單的驗證，來驗證量測從接收裝置送出滑鼠或鍵盤的動作，到接收到對應的畫面回傳時間是否準確。

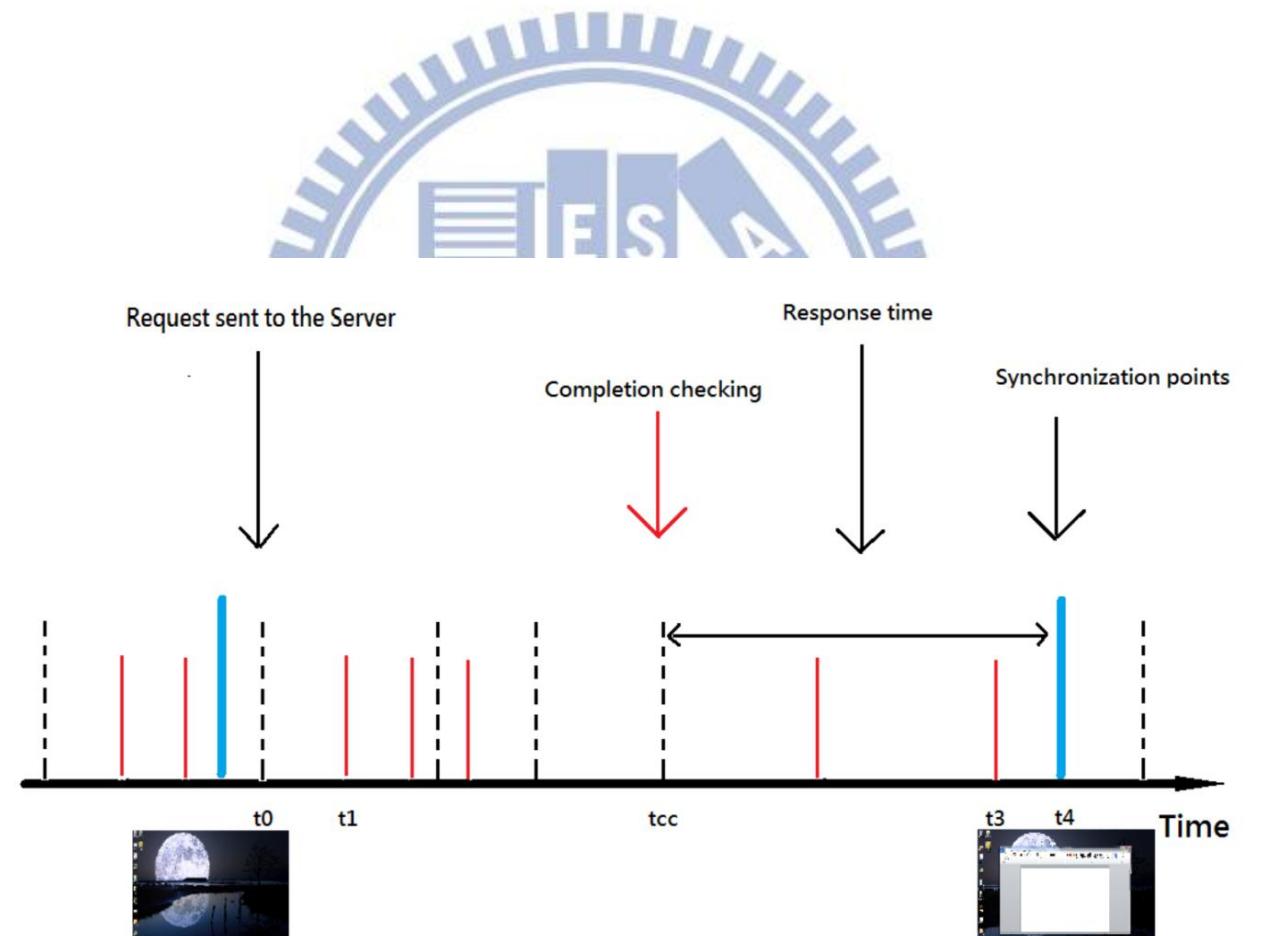


圖 5、伺服器與接收裝置互動情形

圖 6 是評估工具的執行流程圖，我們將藉由這張流程圖來介紹評估工具的執行過程，首先會先接受到一連串輸入，輸入會告知共有幾個輸入需要執行，還有每個輸入所做的動作為何，動作包含鍵盤輸入的內容，以及有關滑鼠的操作左鍵右鍵滾輪的輸入，以及根據輸入所描述的座標來指定每個動作是在螢幕上何處進行。

首先評估工具當還有動作需要執行時，便會將這個動作根據輸入時所描述的狀態來執行，緊接著啟動一個計時器，此計時器是之後用來計算執行動作到畫面回傳所需要的時間之用。接著再查看此動作是否為需要建立同步的動作，如果不是需要建立同步的動作，則直接開始下一個動作，若此動作是需要建立同步的動作，則開始進入畫面比對的流程，評估工具將持續比對 XenDesktop Receiver 畫面內的視窗，與一開始建立此動作完成後的畫面是否相同，直到比對到畫面相同為止，評估工具都將不會開始下一個動作，而比對畫面是否相同的動作約 100 毫秒執行一次，接著如果比對相同的話，便停止計時器，記錄下所量測的時間，我們稱此一時間為回應時間(Respond Time)，接著判斷這是否為最後一個需建立同步動作，如果是的話便完成了一個實驗，否則就開始下一個動作，根據這個簡單的流程，我們就可以掌握接收裝置每個動作需花多久時間得到由伺服器回應的畫面，讓我們可以精準的掌握接受裝置，在伺服器或網路環境惡劣下，接收裝置感受到的延遲情形，我們就可以對伺服器的狀態做評估，也可以再進一步的做為診斷工具的診斷依據。



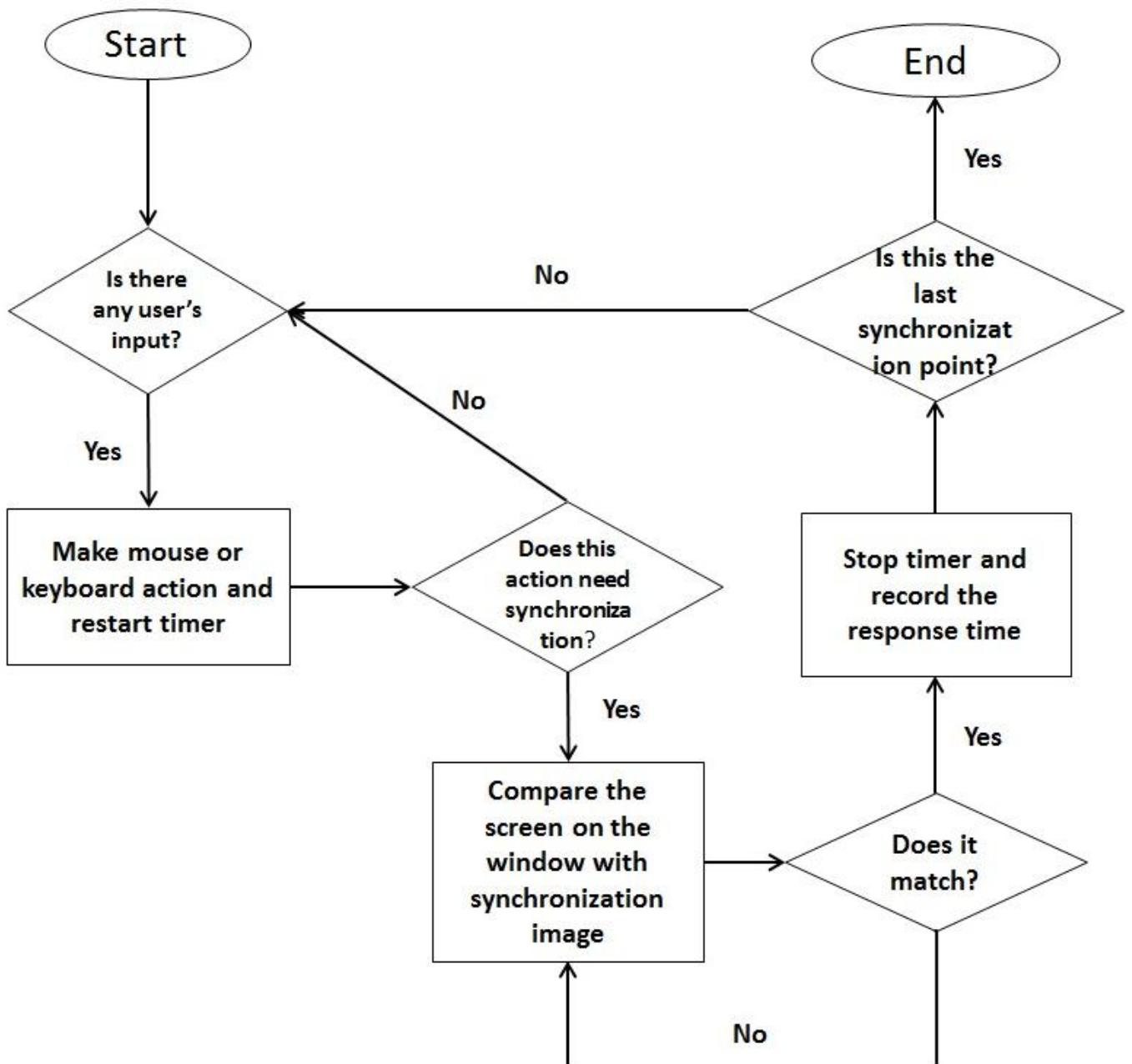


圖 6、評估工具執行流程圖

根據我們介紹評估工具流程後，可以發現，比對 XenDesktop Receiver 畫面內的視窗，與一開始建立此動作完成後的畫面是否相同這個流程，佔據了評估工具執行流程中最長的一段時間，也因此影響了我們量測回應時間的正確性。

所以我們想了一個簡單的方法，來增加量測回應時間的正確性，以下舉一個簡單的例子來說明，我們假設現在要執行開啟 Microsoft Word 的動作，且此動作是一需要建立同步的動作，因為之後的動作都必須等到畫面開啟完成才能執行，否則將導致結果不正確，圖 7 是尚未執行開啟動作前的狀態，而圖 8 是我們預期開啟 Microsoft Word 後的畫面。在之前的做法我們將 XenDesktop Receiver 畫面內的視窗，與一開始建立此動作完成後的畫面拿來比較，但根據不同的動作，其實畫面並不是全部都會更動，所以在這例子中，我們只需要比對 Microsoft Word 開啟的位置如圖 8 所示，因為只要開啟位置內的畫面相同，也代表了 Microsoft Word 已順利開啟，便可繼續執行下一動作。因為需要比較畫面是否相同的範圍縮小許多，也加快了畫面比較的時間與正確性，所以後來我們都使用此做法，也就是根據執行每個需要建立同步動作之後，只對螢幕會產生改變的範圍來比較，所以在開始實驗之前的準備工具，也必須記錄畫面比較的範圍，在執行工具執行的時候，才能判斷需建立同步動作需要比對的範圍，使用此方法的優點是加快了畫面比較的速度與正確性，會提升正確性是因為如果擷取的畫面越大，不穩定的因素就越多，例如在鍵入文字時閃爍的游標等等…，而提升畫面比較速度，則能夠讓評估工具能在更接近畫面回傳的時間就進行，因此能提升評估工具在測量回應時間的準確性，以上是對評估工具的改進說明。





圖 8、動作執行前 XenDesktop Receiver 畫面

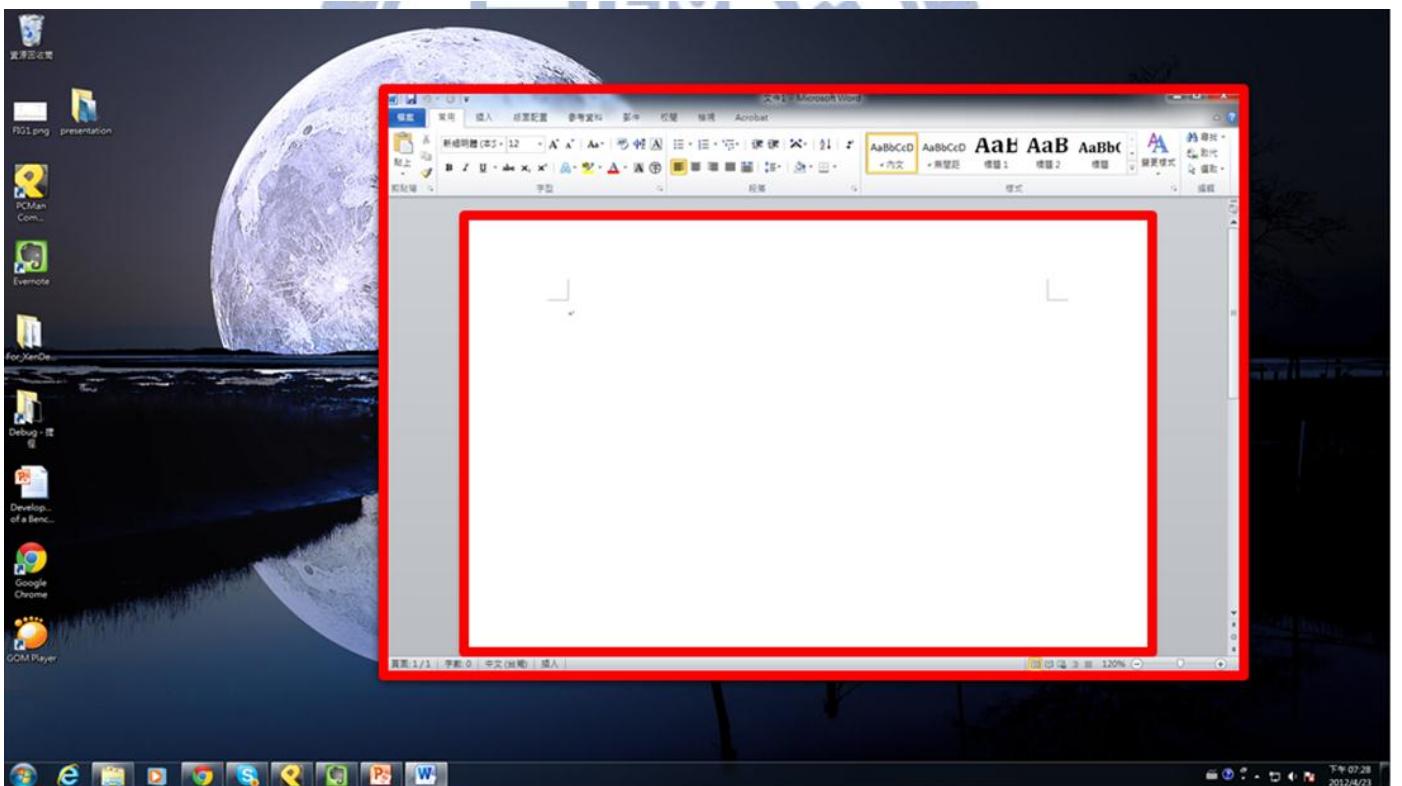


圖 7、動作執行後，預期畫面呈現結果

四、VDI 診斷工具開發

4.1 開發動機

在不同 VDI 平台上，使用者與 VDI 服務提供業者都很關心一個問題，就是當使用者在使用 VDI 服務時，如果發生畫面停頓，操作延遲等等現象，是因為什麼原因所造成的，使用者必須知道是什麼因素才能知道如何解決，如果是網路問題則該找尋 ISP 服務提供業者來解決，如果是 VDI 伺服器的問題，則該向 VDI 服務提供業者反映，而 VDI 服務提供業者也需要這樣一套工具來知道是什麼因素造成 VDI 使用者感覺效能低落，才能進一步改善，所以我們認為開發診斷工具，對於 VDI 平台的發展有著極重要的貢獻，有了這樣的工具能讓使用者更願意使用 VDI 平台，因為當查覺效能低落，就能知道是何種因素造成，能夠即時進行改善，也可以在使用平台之初，能先做一個全面的評估，來決定自己作業的環境是否適合使用 VDI 平台。

4.2 診斷工具設計與架構說明

首先我們先設定可以診斷的項目，主要分成網路環境與伺服器資源，根據我們的觀察，這兩大原因如果情況惡劣，都可能會造成使用者效能低落，網路環境部分則分成，連線延遲(link delay)和封包遺失(packet loss)，這兩者都會造成使用者在使用 VDI 服務感受不佳，但感受程度又隨著環境惡劣程度而有所不同。而伺服器資源則是分成處理器(CPU)、記憶體(Memory)、磁碟(Disk)，與傳統桌面環境不同，VDI 使用到的硬體資源是與多位使用者共享的，所以雖然使用者覺得自己並未將硬體資源耗盡，但由於可能與多位使用者同時競爭，而造成了效能低落，處理器造成效能低落，例如伺服器端由於處理器使用率太高，加上多位使用者同時競爭，造成 VDI 使用者因為處理器效能不足而造成執行使用者操作變慢。記憶體部分則是由於大部分 VDI 伺服器在配置記憶體都是採取動態配置，當使用者越多時則配置的記憶體也將變少，如果記憶體不足，當使用者在執行時，伺服器會一直進行記憶體交換(Memory Swap)，此動作牽扯到磁碟的存取，所以 VDI 使用者也會感受到效能低落。最後是磁碟資源不足造成 VDI 使用者感覺效能低落，這邊指的並不是磁碟空間，而是由於多位使用者同時進行磁碟存取，因為每位使用者存取的區塊不一定是同一部分，當磁碟讀寫頭忙碌移動時，VDI 使用者在進行磁碟相關的操作時，也會感覺到效能低落。所以我們期望使用者在執行診斷工具後，能夠精準的知道是每種因素目前的狀況如何，能夠協助 VDI 伺服器端而使用者來解決及排除問題，盡快給使用者一個無延遲的 VDI 環境。

為了達成上述目標，我們利用我們在評估工具實驗中得到的結果，得知不同類型操作，需要不同的伺服器資源，所以我們的想法是，診斷工具會先設定特定的執行動作，與評估工具類似，接著將量測到不同動作所得到的畫面回應時間傳給伺服器端，因為只有伺服器端能得知伺服器資源的狀態，而網路部分的狀態我們藉由伺服器端執行 Ping 這個指令來探測伺服器端與接收裝置之間的網路狀況。結合這些資訊我們便可推論出是何種因素造成使用者感受到效能低落，推論的過程將在下一小節詳述。

診斷工具分為伺服器端以及接收裝置端，在接收裝置端部分，主要功能與評估工具類似，都負責量測使用者執行動作後接到畫面的時間，另外還需要擔任伺服端與接收裝置建立的 TCP 連線用戶端(Client)的角色，需要將量測的時間傳送給伺服器端。而伺服器端主要負責查看伺服器資源的使用情形、探測接收裝置與伺服器間網路狀況，以及利用上述資料進行診斷工作，並擔任伺服端與接收裝置建立的 TCP 連線伺服器端(Server)的角色，將診斷結果回傳給使用者。以上就是診斷工具主要組成以及接收裝置和伺服器端負責的功能。

以下是診斷工具執行流程

1. 執行設定好的診斷工具動作(開啟關閉 Microsoft Word、鍵盤輸入、解壓縮檔案)。
2. 連線到伺服器建立 TCP 連線。
3. 將量測到各個動作的回應時間傳送給伺服器端診斷工具。
4. 伺服器端診斷工具利用 Ping 探測與 VDI 使用者之間的網路狀況，以及查看目前伺服器各項資源的使用情形。
5. 進行診斷流程，伺服器端診斷工具利用所得到的回應時間以及資源使用情形產生診斷結果。
6. 將診斷結果回傳給診斷工具 VDI 使用者端。

4.3 診斷工具實作

在這一小節我們將介紹診斷工具的實作，以及說明如何診斷是何種因素造成使用者感受不佳，由於診斷工具在接收裝置上的實作，大部分沿用評估工具之內容，因此不多加敘述，只增加了建立 TCP 連線的動作，我們希望接收裝置只需要按一個按鍵就能執行診斷工作並得到結果，因為一般使用者並不一定能進行複雜的操作。而主要診斷資料的收集和診斷流程都由伺服器端負責，所以在這一小節，我們將會介紹診斷工具在伺服器端如何收集資料以及利用這些資料進行診斷。

我們實作的診斷工具在伺服器端所需要收集的資料主要分成兩大部分，一部分如表 1 為評估工具實驗相關的診斷資料，利用在評估工具所做的實驗得知，使用者執行不同的動作時效耗不同的伺服器資源，所以我們的想法是藉由接受裝置所回傳執行不同動作的畫面回應時間，來得知伺服器目前的狀況。第二部分是如表 2 所示，是伺服器端利用 Citrix XenServer 所提供的指令 Xentop 來得知目前系統資源分配情形，我們只收集我們關心的部分也就是伺服器整體處理器、記憶體和磁碟的使用情形，以及分配給接收裝置的虛擬機器。藉由這兩部分的交叉比對，我們可以更精準的診斷使用者是由於何種因素造成功能低落，若只依據表 1 或是表 2 單獨的資訊都無法有效判斷，舉例來說，若我們由伺服器資料收集得知伺服器整體處理器高達 95% 以上但根據我們實驗結果，處理器使用率在相同忙碌情況下，由於不同個數的使用者競爭，對單一使用者而言也有著不同的影響，所以我們需要參考表 1 所量測的數據，才能更精準且有效的評估。

在診斷的過程中，若只有一項因素，也就是伺服器資源不足或是網路環境惡劣，根據我們評估實驗的觀察，可以輕易的分辨是哪一項因素，但我們在這裡考慮到，如果有多重因素造成使用者感受不佳，一項與多項因素的在診斷作法是不相同的。如果只需診斷出一項最主要的因素，我們只需要由表 1 以及表 2 的資料，就可以推論出是何種因素目前嚴重影響使用者的功能，但如果有多項因素的話，由於表 1 內的結果，如壓縮檔案的效率會被記憶體、處理器或是磁碟影響，如果有幾項因素，只根據表 1 以及表 2 的資料無法直接判斷出，是哪幾項因素影響了使用者感受到的功能不佳，所以為了能夠診斷多種項目。

我們的想法是，對每一項因素個別診斷，再根據評估工具實驗結果，將此項因素所會影響到的表 1 內的實驗回應時間縮減，如此一來便能夠達到檢測多種項目的目的。舉例來說，若現在由於網路延遲以及處理器資源缺乏，由實驗結果我們得知，網路延遲對表 1 內的三個動作都有影響，但處理器資源只對開啟 Microsoft Word 以及壓縮檔案有影響，但因為有多項因素，如果直接根據接收裝置量測開啟 Microsoft Word 的反應時間以及處理器效能做診斷，可能會高估了處理器資源所造成的惡劣情形，因為網路延遲也會影響開啟 Microsoft Word 的畫面回應時間，所以我們會將開啟 Microsoft Word 的回應時間減去因為網路延遲所造成的估計值，此一估計值是由評估工具所量測的結果來估計。

而表 3 則是我們將診斷工具所診斷的項目與需要的資料做總整理，透過評估工具的實驗我們可以得知何種使用者操作與較需要資源的對應，所以當某一動作執行變慢時代表對應需要的資源有可能處於缺乏情形，加上伺服器資料我們便可進一步確認，所以診斷依據分為執行動作畫面回應時間與伺服器資源。

診斷的流程如下

1. 由表 1 鍵盤輸入特定字串以及表 2 線路延遲(link delay)來診斷是否有線路延遲問題。並根據評估工具實驗結果對表 1 內三個動作的畫面回應時間減去估計值。
2. 由表 1 鍵盤輸入特定字串以及表 2 封包遺失率(packet loss rate)來診斷是否有線路延遲問題。並根據評估工具實驗結果對表 1 內三個動作的畫面回應時間減去估計值。
3. 由表 1 開啟 Microsoft Word 以及表 2 使用者虛擬機器分配記憶體以及使用者虛擬機器磁碟讀寫效能，來診斷是否有記憶體不足問題。並根據評估工具實驗結果對表 1 內開啟 Microsoft Word 以及壓縮檔案的畫面回應時間減去估計值。
4. 由表 1 開啟 Microsoft Word 以及表 2 伺服器整體處理器使用率來診斷是否有處理器資源不足問題。並根據評估工具實驗結果對表 1 內壓縮檔案的畫面回應時間減去估計值。
5. 由表 1 壓縮特定檔案以及表 2 伺服器整體磁碟讀寫效能來診斷是否有磁碟資源不足問題。
6. 將診斷結果回傳給 VDI 使用者。

表 1、評估工具實驗相關診斷資料

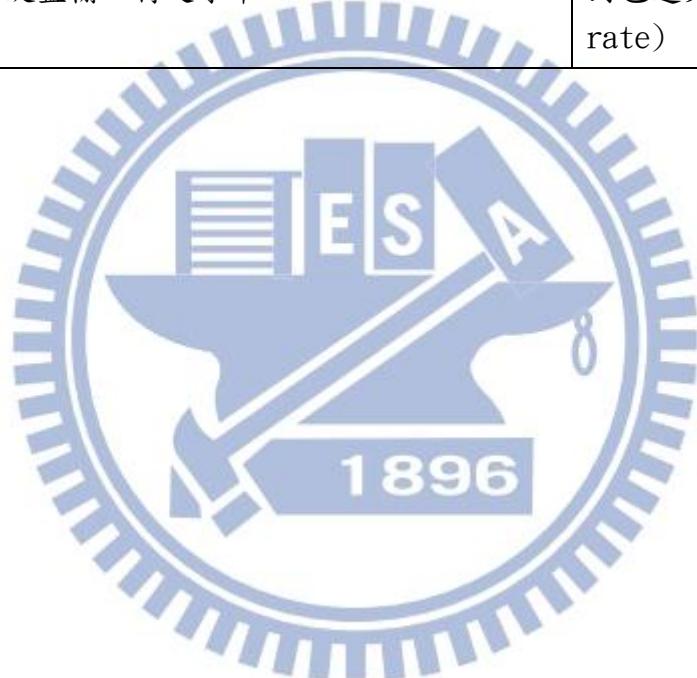
對應評估工具實驗項目	說明
開啟 Microsoft Word	此動作是指使用者點擊滑鼠左鍵兩次之後，等待 Microsoft Word 畫面完全開啟所需要的時間。根據評估工具實驗結果，我們得知伺服器資源 CPU 及 Memory 的多寡會嚴重影響此時間的快慢，而網路環境也會影響，但影響程度較小。
鍵盤輸入特定字串	此動作是指使用者輸入特定字串，等待字串在 XenDesktop Receiver 視窗內畫面正確呈現所需要的時間。根據評估工具實驗結果，我們得知伺服器資源對此時間並未有太大影響，而網路環境決定了此時間的快慢。
壓縮特定檔案	此動作是指使用者對特定檔案點擊壓縮動作後，等待在 XenDesktop Receiver 視窗內畫面壓縮完成的時間。根據評估工具實驗結果，我們得知伺服器資源 Disk 將嚴重影響此時間快慢，Memory 及 CPU 也有著不小的影響。

表 2、伺服器資源相關診斷資料

伺服器資源項目	說明
線路延遲(link delay)	在伺服器端收到由接收裝置所送出的診斷要求後，伺服器會利用 Ping 這個指令來量測伺服器端與接收裝置之間的線路延遲。
封包遺失率(packet loss rate)	在伺服器端收到由接收裝置所送出的診斷要求後，伺服器會利用 Ping 這個指令來量測伺服器端與接收裝置之間的封包遺失率。
伺服器整體 CPU 使用率	在伺服器端收到由接收裝置所送出的診斷要求後，伺服器會利用 xentop 這個指令來取得 10 秒內整體伺服器 CPU 使用率。
伺服器整體已分配記憶體	在伺服器端收到由接收裝置所送出的診斷要求後，伺服器會利用 xentop 這個指令來取得伺服器目前已分配和剩餘記憶體資源。
伺服器整體磁碟讀寫效能	在伺服器端收到由接收裝置所送出的診斷要求後，伺服器會利用 xentop 這個指令來取得 10 秒內整體伺服器磁碟的讀寫效能。
使用者虛擬機器 CPU 使用率	在伺服器端收到由接收裝置所送出的診斷要求後，伺服器會利用 xentop 這個指令來取得 10 秒內使用者虛擬機器 CPU 使用率。
使用者虛擬機器分配記憶體	在伺服器端收到由接收裝置所送出的診斷要求後，伺服器會利用 xentop 這個指令來取得使用者虛擬機器目前已分配和剩餘記憶體資源。
使用者虛擬機器磁碟讀寫效能	在伺服器端收到由接收裝置所送出的診斷要求後，伺服器會利用 xentop 這個指令來取得 10 秒內，使用者虛擬機器磁碟的讀取效能。

表 3、診斷工具診斷依據表

診斷項目	執行何種動作畫面回應時間	何種伺服器量測資料
處理器	開啟 Microsoft Word	伺服器整體 CPU 使用率
記憶體	開啟 Microsoft Word	使用者虛擬機器磁碟讀寫效能。 使用者虛擬機器分配記憶體。
磁碟讀寫	壓縮特定檔案	伺服器整體磁碟讀寫效能
線路延遲	鍵盤輸入特定字串	線路延遲(link delay)
封包遺失率	鍵盤輸入特定字串	封包遺失率(packet loss rate)



五、評估工具實驗環境與實驗結果

5.1 研究與實驗方式

在本論文的研究中，我們想藉由我們設計的評估工具，來得知 VDI 平台在使用者在執行不同操作，在各種伺服器負擔狀況以及網路環境下，會如何影響使用者感受到的效能，藉由所測得的數據，一來可以讓 VDI 服務提供業者得知伺服器在什麼樣的負擔下，對於使用者感受到不同動作的反應時間為何，再者可以讓我們的診斷工具得到參考指標，結合查詢伺服器資源的使用狀況，能產生正確的診斷結果。

我們首先對於不同的實驗參數進行分類，如表 4 所示，將各種不同的伺服器資源或是網路環境，搭配我們想要研究的實驗設計，定義出各種實驗案例，之後針對各個案例的實驗結果進行整理、分析與歸納。

表 4、實驗參數分類表

實驗參數	說明
處理器資源	我們在伺服器裡的虛擬機器中，執行消耗處理器的程序，由不同數量的虛擬機器執行程序再搭配使用者送出給伺服器端不同的動作，來觀察使用者感受到的效能會是如何。
記憶體資源	我們藉由分配給使用者分配到的虛擬器不同的記憶體資源，再搭配使用者送出給伺服器端不同的動作，來觀察使用者感受到的效能會是如何。
磁碟資源	我們在伺服器裡的虛擬機器中，執行消耗磁碟的程序，由不同數量的虛擬機器執行程序再搭配使用者送出給伺服器端不同的動作，來觀察使用者感受到的效能會是如何。
線路延遲	我們利用 EstiNet Emulation 的功能變化伺服器端與接收裝置之間的線路延遲，由不同程度的線路延遲再搭配使用者送出給伺服器端不同的動作，來觀察使用者感受到的效能會是如何。
封包遺失率	我們利用 EstiNet Emulation 的功能變化伺服器端與接收裝置間產生封包遺失的動作，由不同程度的封包遺失率再搭配使用者送出給伺服器端不同的動作，來觀察使用者感受到的效能會是如何。

5.2 實驗情境參數

在定義本論文所要探討的實驗案例前，我們先將各種不同的實驗參數進行分類，在設定實驗案例時，將依所要探討的各個議題之實驗設計搭配各種實驗參數，定義出不同的實驗案例。在本節中，我們將介紹我們使用的實驗參數，包含伺服器資源和網路環境進行說明。

5.2.1 處理器資源

在本論文中，我們藉由在伺服器端新增虛擬機器，每一台虛擬機器將執行我們所撰寫的一支應用程式，此應用程式將消耗伺服器端處理器資源約 1/24，在本論文實驗中所採用的伺服器有 24 核心。我們一次增加兩台虛擬機器執行消耗處理器資源程序，從 0 台到 32 台虛擬機器，如此一來我們便可觀察使用者在不同處理器資源下執行不同的動作，所需的畫面回應時間。

5.2.2 記憶體資源

在本論文中，我們藉由同時在伺服器中開啟的虛擬機器數量，來對我們實驗中使用者的接收裝置，進行記憶體資源配置的變化，由於大部分 VDI 平台都是採取動態記憶配置記憶體的方式，在開啟較少的虛擬機器時，使用者能分配到的記憶體也較多，反之則使用者分配到的記憶體則較少，為了在實驗過程中更明顯看出使用者對記憶體資源的需求，我們額外在使用者的虛擬桌面上，執行了消耗記憶體資源的程序，整體約使用了 1G 的虛擬記憶體，而配置上使用者的虛擬桌面隨著開啟虛擬機器多寡，可配置 1~4G 的記憶體。如此一來我們便可觀察使用者在不同記憶體資源下執行不同的動作，所需的畫面回應時間。

5.2.3 磁碟資源

在本論文中，我們藉由在伺服器端新增虛擬機器，每一台虛擬機器將執行需要大量磁碟讀取的工作，其中在只有一台虛擬機器執行此工作時磁碟讀取效率約 6000KB/s。我們一次增加兩台虛擬機器執行需要磁碟讀寫資源的工作，從 0 台虛擬機器到 10 台，如此一來我們便可觀察使用者在不同磁碟資源下執行不同的動作，所需的畫面回應時間。

5.2.4 線路延遲

在本論文中，我們藉由在接收裝置和伺服器間，引入 EstiNet 的 Emulation 功能，如此一來我們便可輕易的變化網路環境，我們將接收裝置和伺服器之間的其中一條線路延遲，從單向 1 毫秒變化到 500 毫秒。如此一來我們便可觀察使用者在不同線路延遲下執行不同的動作，所需的畫面回應時間。

5.2.5 封包遺失率

在封包遺失率實驗參數中，我們一樣是在接收裝置和伺服器間，引入 EstiNet 的 Emulation 功能，在 EstiNet 中有一個功能是能設定封包遺失率，我們將接收裝置和伺服器之間的傳送的封包攔截並設定不同程度的封包遺失率，一次增加 2%，從 0% 增加到 20%。如此一來我們便可觀察使用者在不同封包遺失率下執行不同的動作，所需的畫面回應時間。

5.2.6 實驗環境說明

另外我們利用圖 9 對我們實驗環境進行說明，我們實驗環境一共使用了三台實體機器分別為接收裝置端、伺服器端，和變化網路環境安裝 EstiNet 的主機，另外上半部分為虛擬網路拓墣，而下半部分為一簡單示意圖，主要是接收裝置與裝有 EstiNet 的主機連接，在透過網路連接上伺服器端，以下我們再分別對這三台主機分別說明。

接收裝置所使用的處理器為 Intel I7 3.4GHz，並安裝 Windows7 及 Citrix XenDesktop Receiver，並負責執行我們的評估工具，此外實驗中我們並未變化接收裝置上的硬體資源，只單純考慮網路環境和伺服器端的資源變化，然後我們將接收裝置的網路連接到安裝 EstiNet 的主機上，藉此產生網路環境的變化。

伺服器端使用的主機為 Asus rs 500A-E6，一共有 24 顆 CPU (1900GHz)，48GB 的記憶體，並安裝 XenServer 以及在 3.3.1 所有負責 XenDesktop 運行的套件，扮演了虛擬桌面分配管理的角色，此外我們除了提供一個虛擬桌面作為接收裝置分配的虛擬桌面外，還額外建立了 32 台虛擬機器，主要是為了執行我們實驗所需的動作，藉以產生伺服器負擔之目的。

而安裝 EstiNet 的主機，是我們為了設定網路環境所需要額外增加作為 Emulator 之用，如圖 9 上半部分所示，虛擬拓墣裡最左邊 IP 1.0.1.1 代表了接收裝置，而 1.0.2.1 為伺服器端。我們變化線路延遲為節點 3 和節點 4 中間的線路，而節點 3 又提供產生封包遺失率的功能。所以藉由設定這台主機的虛擬網路拓墣，我們便可讓接收裝置和伺服器的網路傳遞情形產生變化。以上就是我們在評估工具所使用的實驗環境以及虛擬拓墣說明。

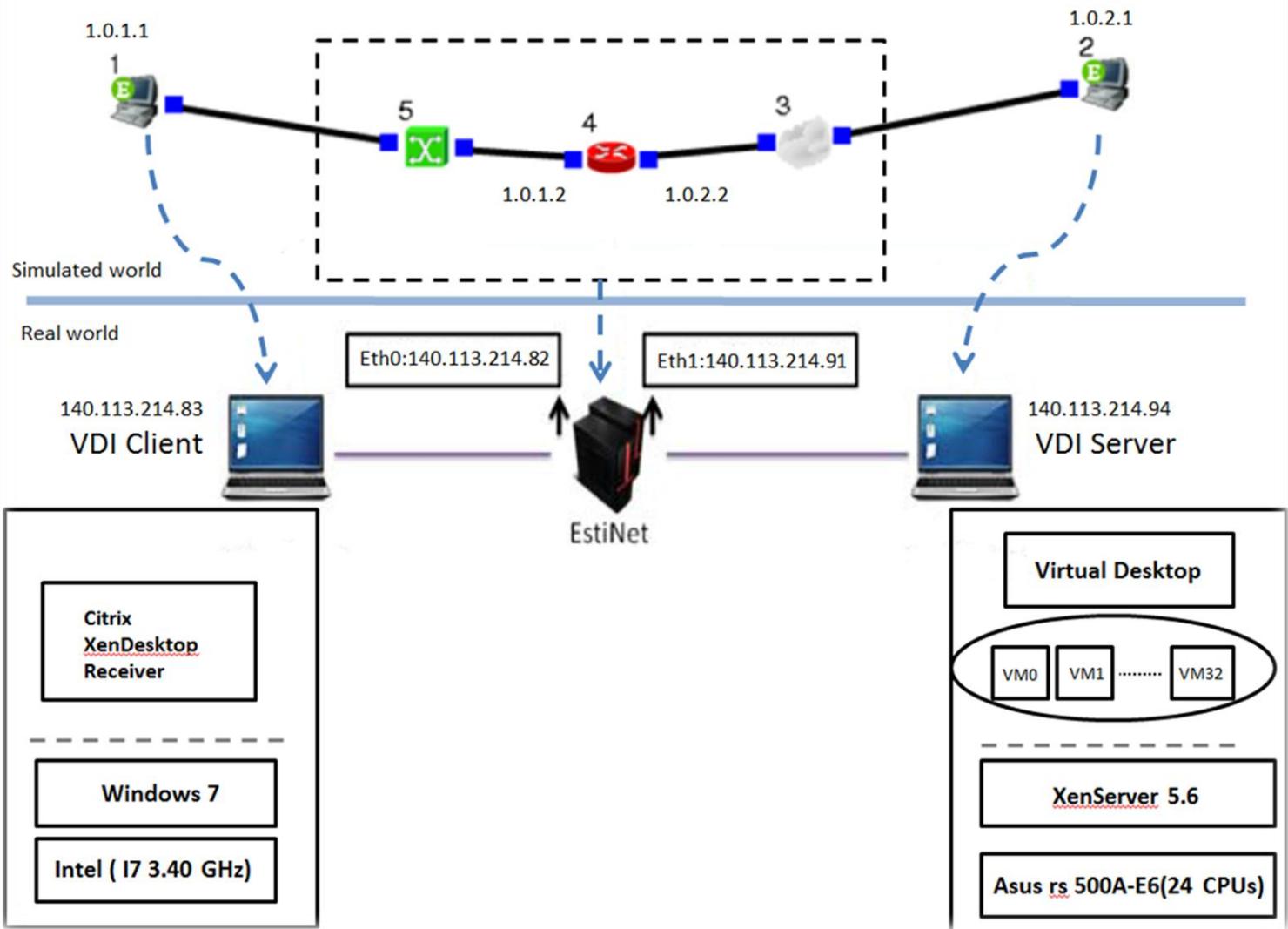


圖 9、實驗環境說明

5.3 效能實驗設計

在本論文的規劃中，我們將幾個想要探討的使用者操作，分別設計了不同的實驗案例。各種實驗案例的設計目的、說明、以及該設計中使用的實驗情境參數種類分別說明如下：

5.3.1 評估實驗設計 A

實驗設計 A 是為了觀察，對使用者最常使用的應用程式，在開啟和關閉應用程式時，當伺服器或網路環境處於各種情況時，對使用者感受到的效能會有什麼樣的影響，關於實驗設計 A 的設計目的以及說明請見表 5。

表 5、實驗設計 A 之設計目的與說明

模擬設計編號	A
設計目的	我們觀察了使用者在使用 VDI 平台時，最常做的操作就是文書處理的應用程式，因此我們選擇了 Microsoft Word，並藉由產生不同的系統資源負擔，或是網路環境的變化，藉此可以觀察出使用者感受，可以讓伺服器端得知，什麼樣的系統配置，可以在即使有大量使用者湧入，還能維持使用者可以接受的畫面回應時間。
實驗設計說明	我們在使用者端執行了我們的評估工具，安插了開啟 Microsoft Word 及關閉的動作，並搭配了不同的系統資源負擔，或是網路環境的變化，並量測使用者所接收到開啟或是關閉動作畫面回應的時間。各種實驗參數請見 5.2。

5.3.2 評估實驗設計 B

實驗設計 B 是為了對使用者最常做的另一個操作，也是使用者最關心的，就是當使用者送出鍵盤動作後，由伺服器所送回的回應畫面的時間，因為使用者在使用 VDI 平台時，一定伴隨著大量的鍵盤或滑鼠操作，而因為與傳統桌面不同，多增添了網路環境，所以此動作的反應時間就更難評估了。關於實驗設計 B 的設計目的與說明請見表 6。

表 6、實驗設計 B 之設計目的與說明

模擬設計編號	B
設計目的	我們希望觀察使用者在鍵盤輸入動作之後，在不同的伺服器負擔或是網路環境下，得到所作的鍵盤輸入在畫面上呈現的時間為何，因為此動作是使用者在 VDI 平台上最常做的動作，所以這個實驗所得到的畫面回應時間，最能直接代表使用者是否會感覺到操作不順暢。
實驗設計說明	我們在接收裝置上，執行了我們所設計的評估工具，並設定開啟 Microsoft Word，等待開啟完成後，依序總共輸入五個字串，使用者一次輸入一個字串後，等待 Citrix XenDesktop Receiver 應用程式內畫面已正確顯示後，才開始下一個字串的輸入，之後將五個字串所得到的回應時間平均。我們也一樣搭配 5.2 的實驗情境參數，如此一來我們就可以觀察，是哪些伺服器資源或是網路環境對此動作會有較大的影響。

5.3.3 評估實驗設計 C

實驗設計 C 我們主要是為了對診斷工具資料收集而設計的實驗，由於實驗設計 A 與實驗設計 B，我們觀察到，磁碟讀取效能並未開啟或關閉 Microsoft Word 及鍵盤輸入產生太大的影響，所以我們期望藉由實驗設計 C，來看出何種動作較需要磁碟讀取資源，以提供診斷工具一個參考數值，讓我們可以產生更準確的診斷結果。

表 7、實驗設計 C 之設計目的與說明

模擬設計編號	C
設計目的	由於在實驗 A 以及實驗 B 我們並未看出磁碟讀取對使用者有太大的影響，所以我們想藉由實驗設計 C 來看出磁碟讀取對於使用者會有著什麼樣的影響，並將收集到的實驗結果，提供我們的診斷工具一個參考值，主要希望看出使用者對於磁碟讀寫忙碌情形的反應會是如何，還有對於使用者在執行需要大量讀取操作的行為時，其他伺服器資源對此操作的影響情形。
實驗設計說明	我們在使用者接收裝置上，執行了我們的評估工具，設定的動作是壓縮一個約 600MB 的檔案，並量測壓縮檔案所需要的時間，但與實驗設計 A 和實驗設計 B 不同，我們並未考量網路因素，是由於壓縮 600MB 的檔案所需的時間與網路環境所會影響到的畫面反應時間，有著不小的差距，因此在實驗設計 C 中我們並未考慮網路因素，只有變化了伺服器相關的資源。

5.4 評估實驗結果與分析

在本節中，我們將呈現之前介紹的實驗設計之實驗結果，並且將解釋各種伺服器資源或是網路環境有著什麼樣的影響，並將此結果作為診斷工具診斷的重要參考依據。

5.4.1 評估實驗設計 A 之結果

實驗設計 A 的設計目的是為了看出使用者在進行應用程式的開啟與關閉動作時，隨著伺服器的負擔以及網路環境的不同，會有著什麼樣的影響，因此我們變化了伺服器上處理器、記憶體、磁碟讀取的資源，以及網路環境中，線路延遲和封包遺失率等等，使用者在使用 VDI 平台時，幾種最有可能使得效能被影響的因素。

圖 10~14 是我們評估工具在不同伺服器資源以及網路環境所得到的畫面回應時間，又分為開啟 Microsoft Word 和關閉所需要的時間，實驗數據是將一百次結果平均，以下我們將說明我們所觀察到的情形，這些數據也可以給 VDI 服務提供業者參考，知道伺服器配置與使用者接收到的畫面回應時間有著什麼樣的關聯。

圖 10，橫軸為虛擬機器執行我們設定消耗處理器資源的程序的數量，縱軸為使用者送出開啟或關閉的動作後畫面顯示開啟完成或關閉完成的時間，由圖 10 我們可以發現，在伺服器 32 台虛擬機器啟動時，處理器忙碌的狀況下與無負擔的狀況下相比，開啟 Microsoft Office 約需多花 57% 時間，而關閉需多花費 42% 時間，由此可知開啟與關閉時間對於伺服器上的處理器資源有著高度相關，而另一個發現是在伺服器過於忙碌時，約同時服務 16 台虛擬機器，使用者才會感受到效能變差的影響，另外當伺服器過於忙碌時，使用者感受效能不好的情況將會越來越明顯。

圖 11，橫軸為接收裝置使用的虛擬桌面所配置到的記憶體大小，縱軸為使用者送出開啟或關閉的動作後畫面顯示開啟完成或關閉完成的時間，在這實驗中我們預先在使用者桌面上先執行了消耗記憶體資源的程序，此程序約佔據了 1GB 的記憶體空間，加上作業系統也需要一些記憶體空間，所以可以看出記憶體資源嚴重不足時與無負擔的狀況下相比，開啟與關閉 Microsoft Office 分別需多花費 88% 及 43% 的時間，由此可知開啟與關閉時間對於虛擬桌面所配置的記憶體資源也有著高度相關，因為在記憶體資源不足時，作業系統就需要作 Memory Swap 的動作，因此會大大的影響執行應用程式的速度，當記憶體資源足夠大於所有應用程式所需之記憶體，使用者就感受不到由於記憶體不足所帶來的延遲。

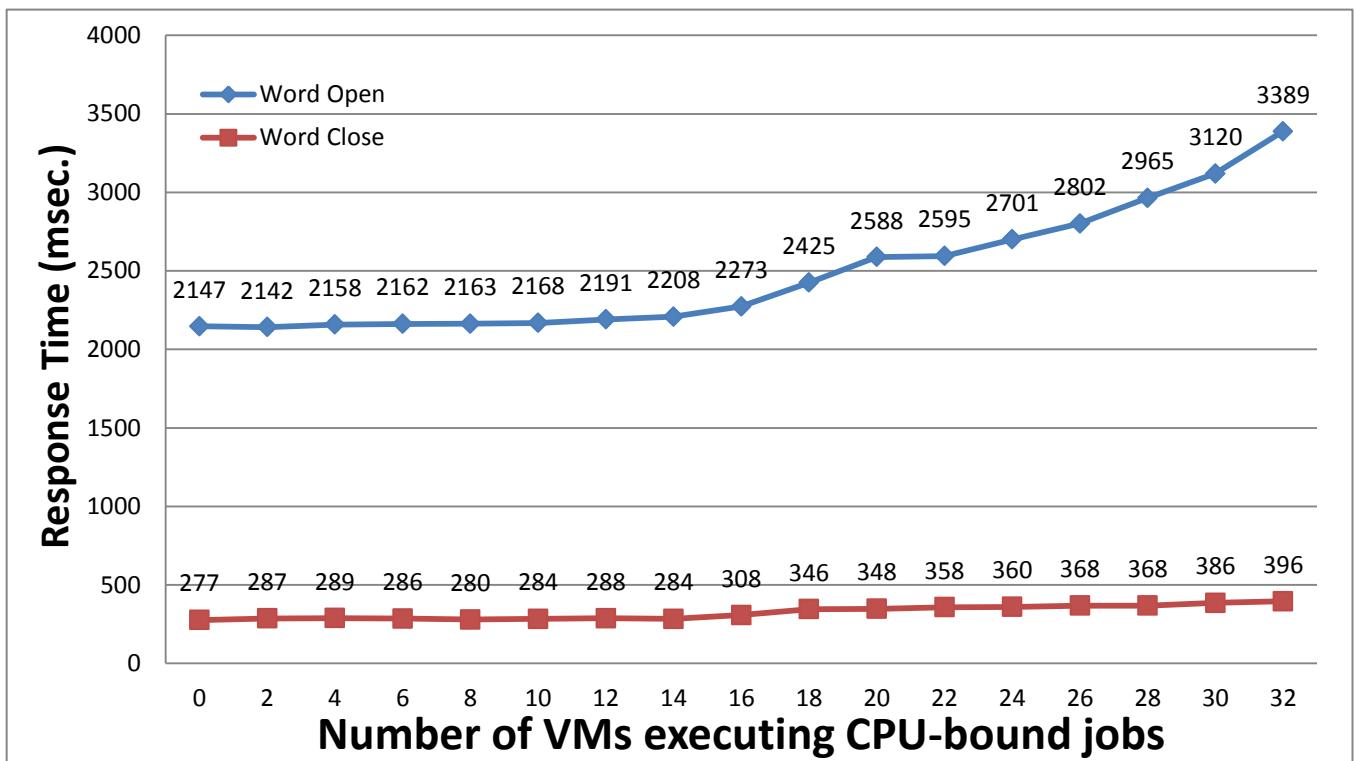


圖 11、實驗設計 A，伺服器處理器資源

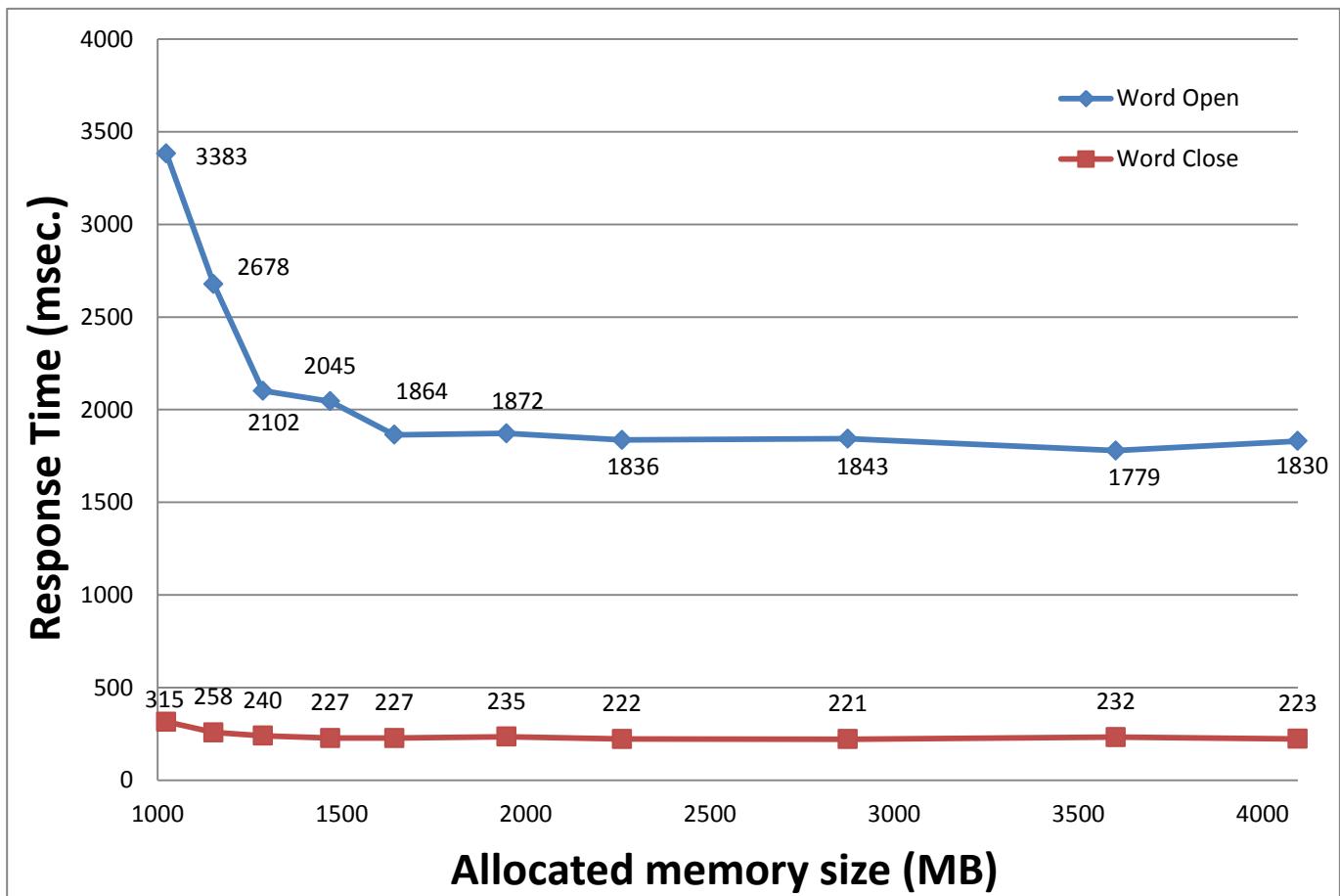


圖 10、實驗設計 A，伺服器記憶體資源

圖 12，橫軸為虛擬機器執行消耗磁碟讀寫的動作的數量，縱軸為使用者送出開啟或關閉的動作後畫面顯示開啟完成或關閉完成的時間，由此圖我們可以發現，磁碟讀寫忙碌與否與開啟或關閉 Microsoft Word 並未有直接關係，因此在我們診斷工具的資料收集中，並無法從開啟或關閉的動作來查覺，伺服器是否處於一個磁碟讀寫忙碌的狀態。

圖 13，橫軸為伺服器與接收裝置之間的線路延遲，縱軸為使用者送出開啟或關閉的動作後畫面顯示開啟完成或關閉完成的時間，此實驗結果是顯而易見的，由於我們設定的線路延遲指的是單向的線路延遲，因為一個畫面的回應，至少需要使用者送出動作透過網路所以需要經過第一段的線路延遲，而接著畫面回應需要經過第二段的時間延遲。在此實驗我們只單純考慮線路延遲的因素，因此可以看到，畫面回應時間隨著線路延遲的增加，也大約多了兩倍線路延遲的時間，此外我們也可以在這個實驗中，簡單的驗證我們評估工具所量測的畫面回應時間的確是精準的，以下是一個簡單的驗證方法。我們以線路延遲 1 毫秒為基準，在分別觀察 100、200、300、400、500 毫秒對於關閉 Microsoft Word 的時間變化，理論上關閉時間應該約增加，200、400、600、800、1000 毫秒，但由於之前提到我們的評估工具會有誤差，加上關閉 Microsoft Word 本身所需的時間也非固定值，但實際量測的結果，線路延遲 100、200、300、400、500 毫秒，關閉時間分別增加，188、393、577、745、946，可以看出評估工具所量測到的畫面回應時間是精準的，也讓實驗數據更具有可靠性，也因此能幫助做為診斷工具作更精準的診斷。

圖 14，橫軸為伺服器與接收裝置之間的封包遺失率，縱軸為使用者送出開啟或關閉的動作後畫面顯示開啟完成或關閉完成的時間，此實驗我們想看出網路上另一重要指標封包遺失率對於 Citrix XenDesktop 使用者的影響，可以觀察到在封包遺失率小於 10% 時，畫面回應時間增加幅度並不大，也是使用者可以接受的範圍，但隨著封包遺失率慢慢提高，反應時間增加的幅度也越變越大，我們猜測由於 Citrix XenDesktop 所使用的傳輸協定為 TCP，因此如果連續的封包遺失，傳輸速度將大幅下降，也因此導致反應時間劇烈增加，所以當封包遺失率增加到 10% 以上，使用者將感受到延遲的情形非常嚴重，也無法很順暢的使用。

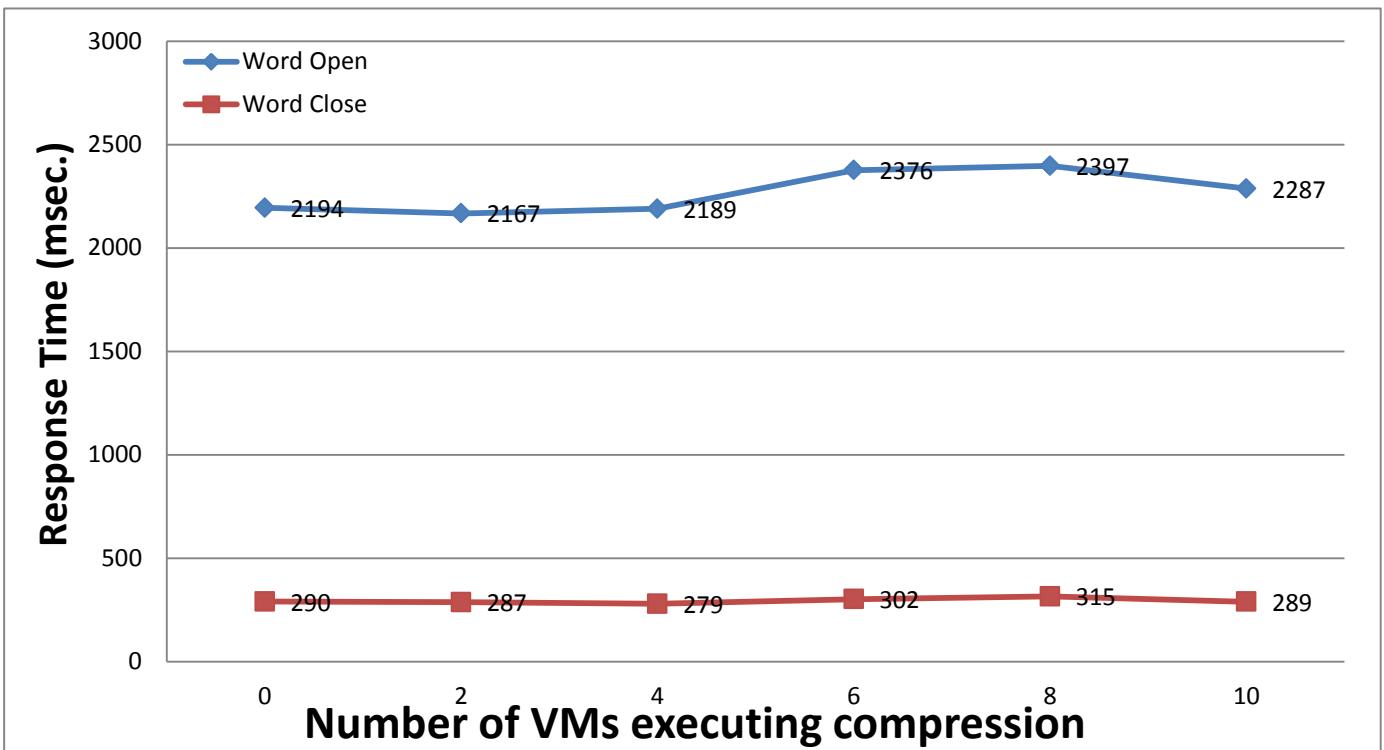


圖 13、實驗設計 A，伺服器磁碟資源

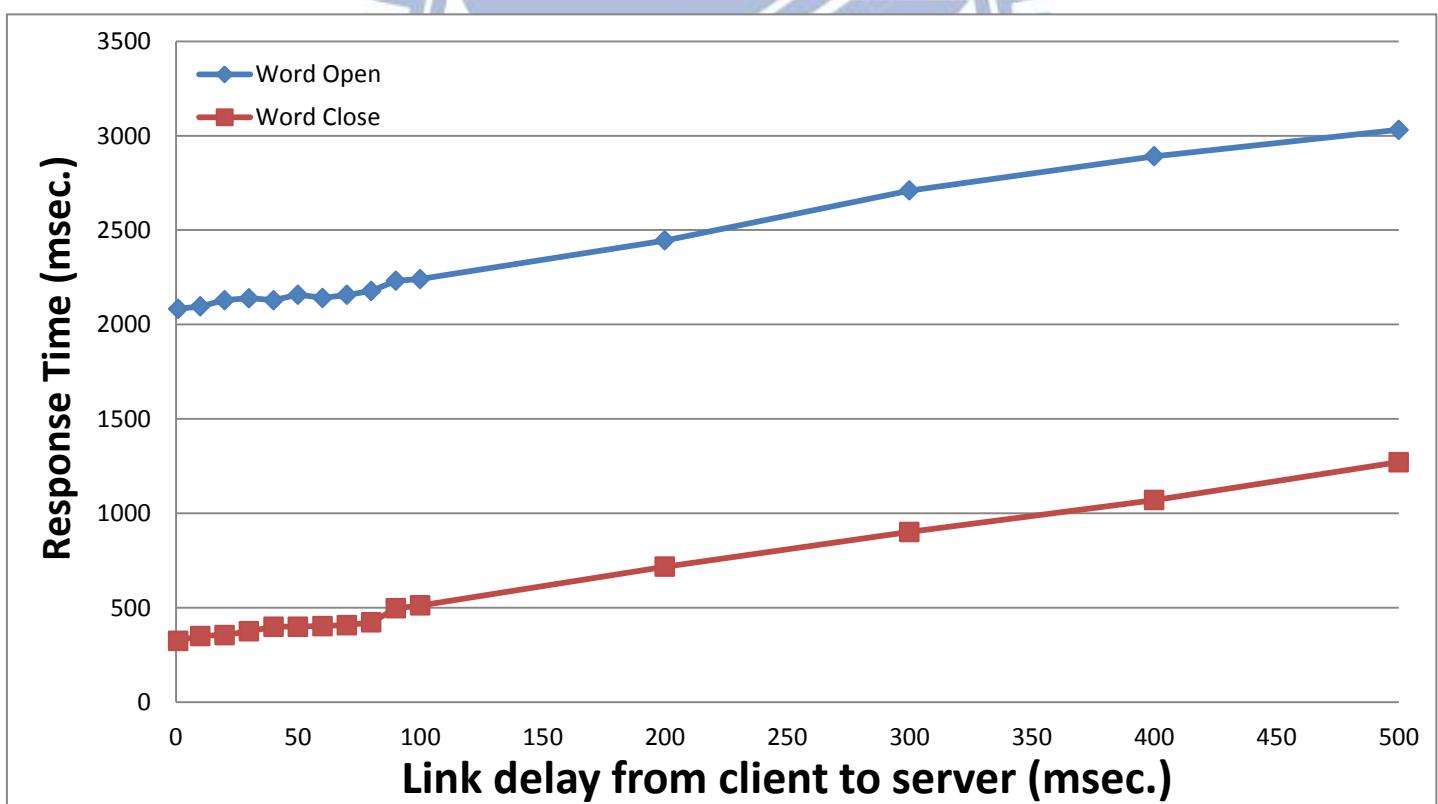


圖 12、實驗設計 A，線路延遲

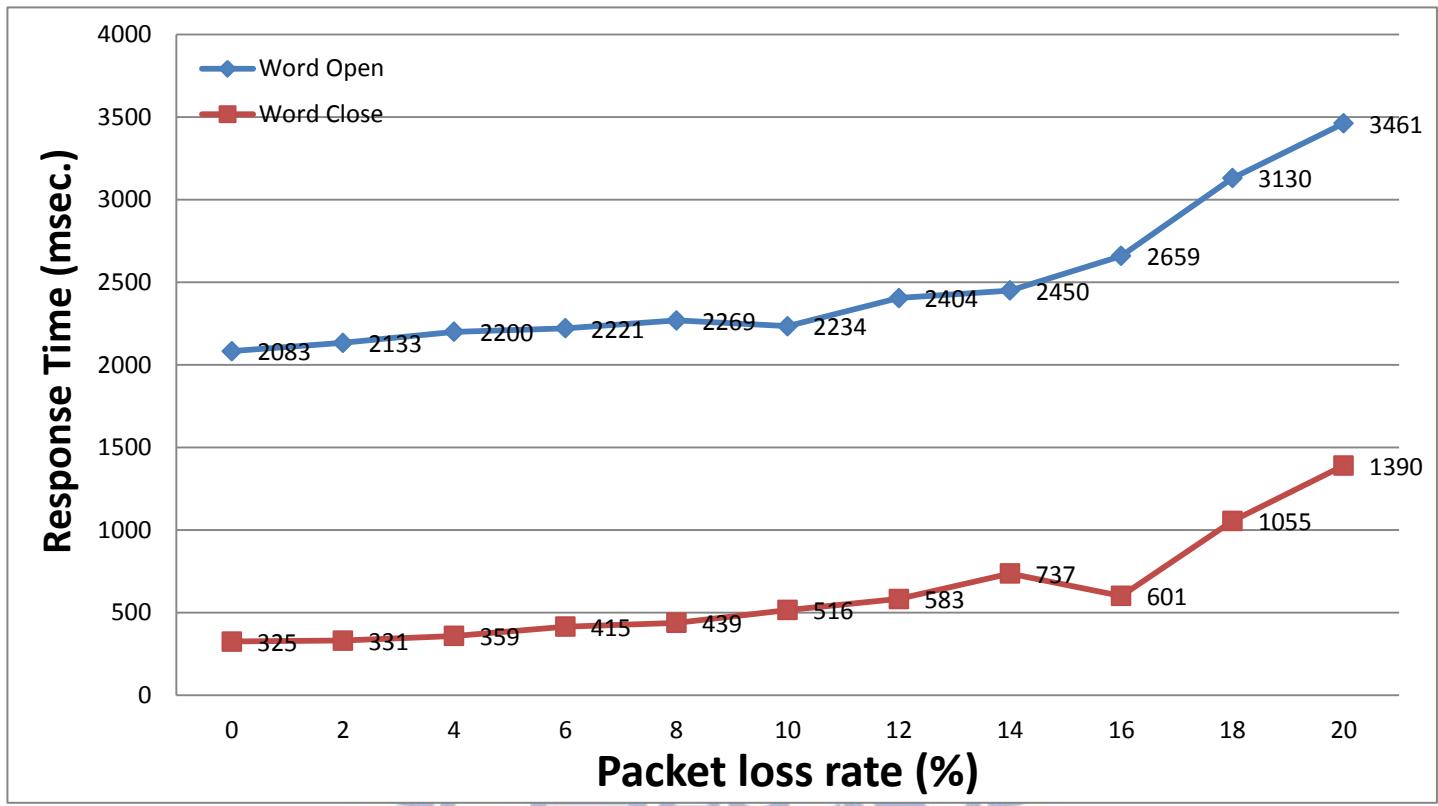


圖 14、實驗設計 A，封包遺失率



5.4.2 評估實驗設計 B 之結果

實驗設計 B 的設計目的是為了看出使用者在進行鍵盤輸入動作時，隨著伺服器的負擔以及網路環境的不同，會有著什麼樣的影響，因此我們一樣變化了伺服器上處理器、記憶體、磁碟讀取的資源，以及網路環境中，線路延遲和封包遺失率等等。

圖 15~19 是我們評估工具在不同伺服器資源以及網路環境所得到的畫面回應時間，一次實驗是將五次鍵盤輸入時間平均，共重複執行一百次將實驗結果取平均值，以下我們將說明我們所觀察到的情形，這些數據也可以給 VDI 服務提供業者參考，知道伺服器配置與使用者執行鍵盤輸入後接受到的畫面回應時間有著什麼樣的關聯。

圖 15，橫軸為開啟虛擬機器數量並執行我們設定消耗處理器資源的程序，縱軸為使用者執行鍵盤輸入動作後畫面將輸入結果正確顯示的時間，由圖 15 可以看出，伺服器處理器資源的忙碌與鍵盤輸入的動作並無太大關聯，可以得知鍵盤輸入動作並不太消耗處理器資源。所以當如果大多數 VDI 平台使用者是屬於鍵盤操作時，VDI 服務業者在配置上可以較不需大量處理器資源。

圖 16，橫軸為接收裝置使用的虛擬桌面所配置到的記憶體大小，縱軸為使用者執行鍵盤輸入動作後畫面將輸入結果正確顯示的時間，由此圖我們可以發現，當記憶體資源不足時，對鍵盤輸入會產生影響，雖然影響程度比起記憶體不足時開啟應用程式來得較不劇烈，我們推測鍵盤輸入雖然不太需要記憶體資源，但由於作業系統如果處於記憶體資源不足的狀況下，我們在操作時，有時並沒辦法直接回應，所以在鍵盤輸入操作時我們也會得到類似的狀況，因此造成反應時間略微上升的情形。

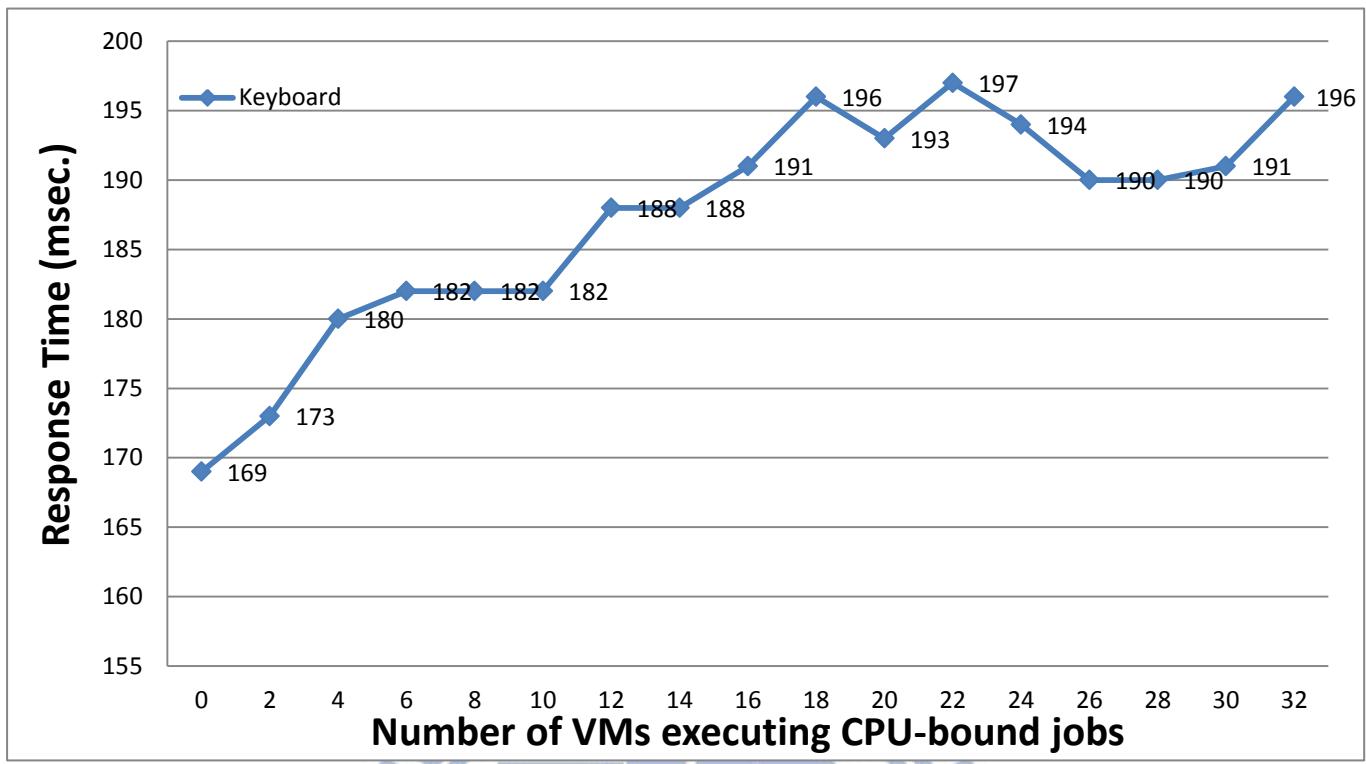


圖 15、實驗設計 B，伺服器處理器資源

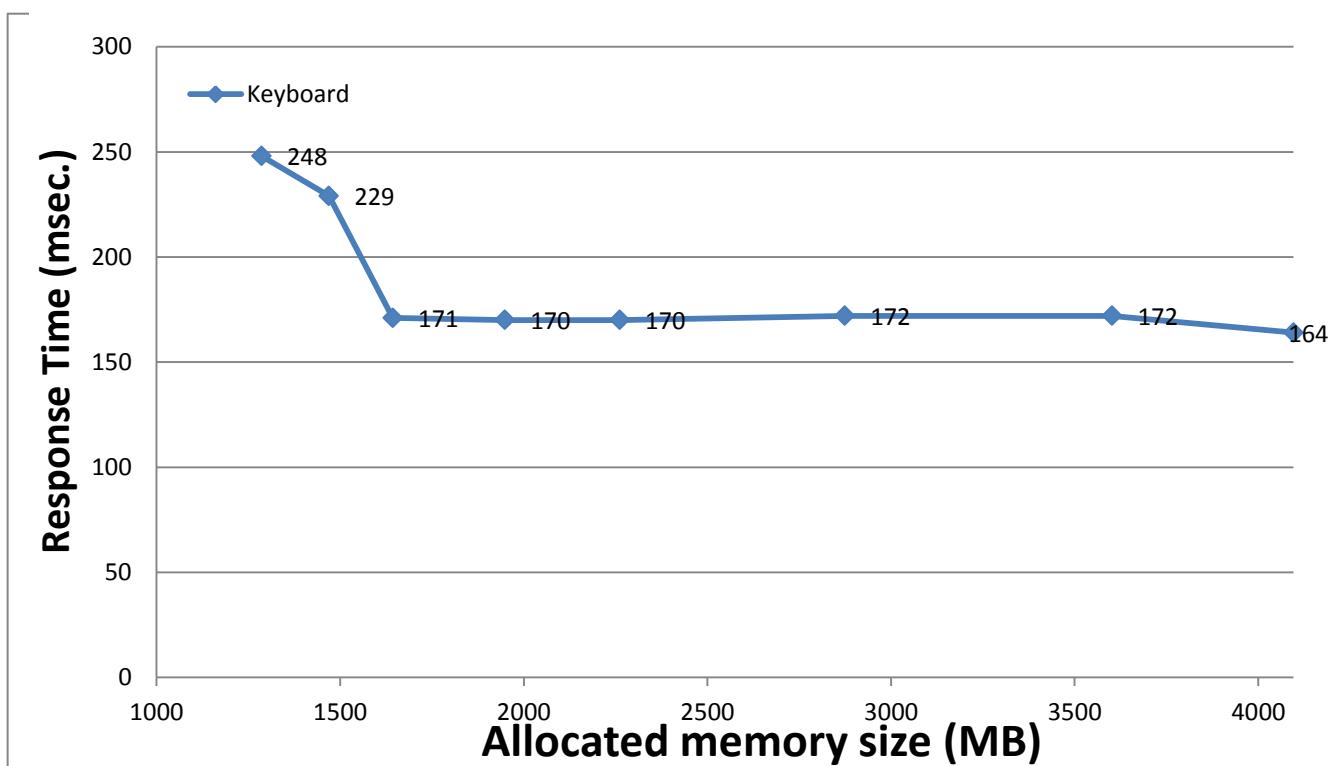


圖 16、實驗設計 B，伺服器記憶體資源

圖 17，橫軸為開啟虛擬機器數量並執行消耗磁碟讀寫的動作，縱軸為使用者執行鍵盤輸入動作後畫面將輸入結果正確顯示的時間，此實驗結果與變化處理器資源類似，我們可以看出，磁碟讀寫的忙碌情形，與鍵盤輸入的動作並無太大關聯，雖然圖形看似不太規則但畫面時間最長與最短只差了 10 毫秒，所以此實驗數據也可提供 VDI 服務提供業一個參考，磁碟讀寫的效能和鍵盤輸入之間並無大關係，所以可以根據 VDI 使用者的使用行為，在伺服器配置上進行調整。

圖 18，橫軸為伺服器與接收裝置之間的線路延遲，縱軸為使用者執行鍵盤輸入動作後畫面將輸入結果正確顯示的時間，此實驗的結果一樣是顯而易見的，隨著線路延遲的增加，畫面時間也成等比例的增加，可以看出線路延遲對於鍵盤輸入操作是有著重大的影響，因為鍵盤輸入動作，畫面回應時間比起其他動作時間是短的多，所以如果線路延遲時間增加，使用者將很直接的感受到延遲的現象，所以如果在評估是否適合使用 VDI 平台時，伺服器與接收裝置間的線路延遲將是一大考量，因為每個鍵盤或是滑鼠操作，若處於線路延遲惡劣的環境下，使用者可能都必須等到兩到三倍以上的時間才能得到畫面回覆。我們同樣藉由這個實驗結果來檢視評估工具所量測畫面回應時間的正確性，以線路延遲 1 毫秒為基準，在 100、200、300、400、500 毫秒線路延遲時，畫面回應時間分別增加了，195、365、572、778、978 毫秒，由此結果可以看出我們所量測的線路延遲也是相當準確的。

圖 19，橫軸為伺服器與接收裝置間的封包遺失率，縱軸為使用者執行鍵盤輸入動作後畫面將輸入結果正確顯示的時間，可以看出隨著封包遺失率的增加，畫面回應時間也跟著增加，而在封包遺失率大於 10% 時，畫面回應時間增加的幅度更為劇烈，與之前討論相同，與傳輸協定是使用 TCP 有很大的關聯，而在封包遺失率太高的狀況下，可以看出畫面增加的時間比伺服器端處理鍵盤輸入動作要多上好幾倍，所以在評估是否適合使用 VDI 平台時，也要將封包遺失率列入重要考量。

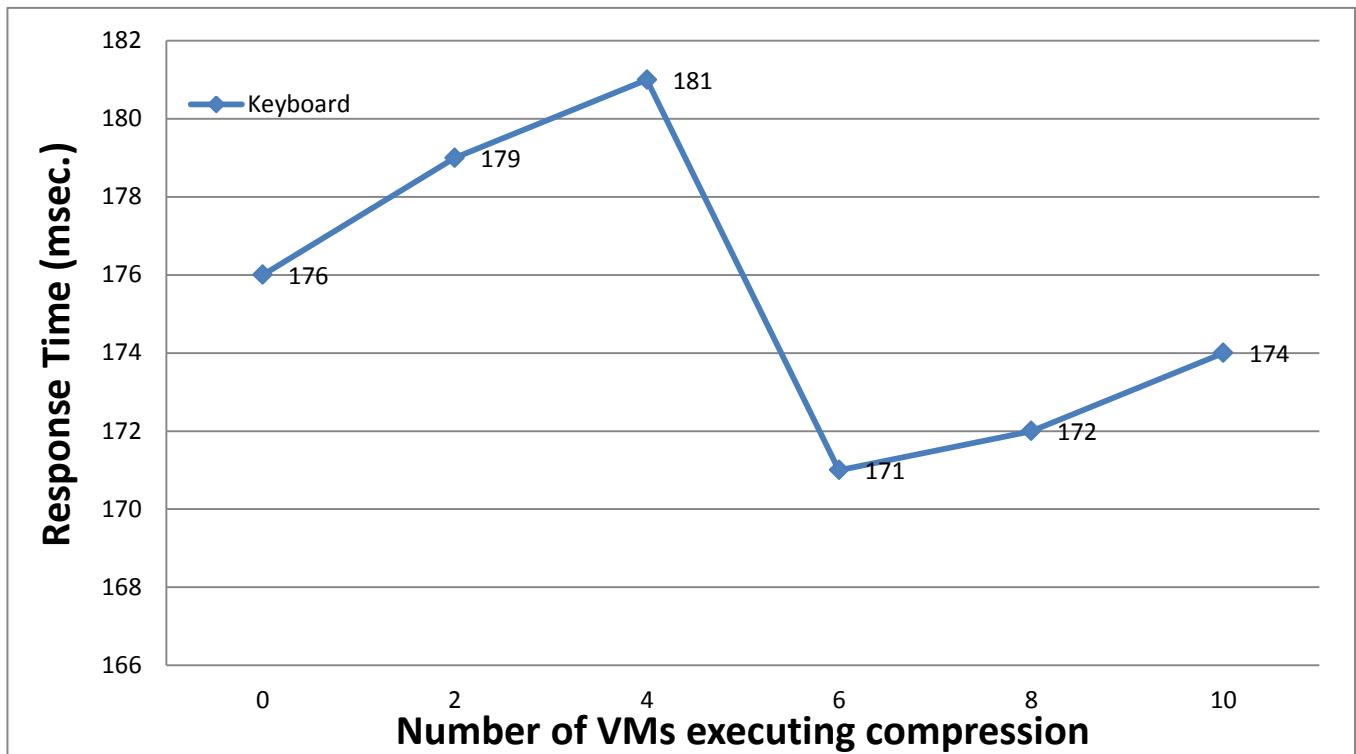


圖 17、實驗設計 B，伺服器磁碟資源

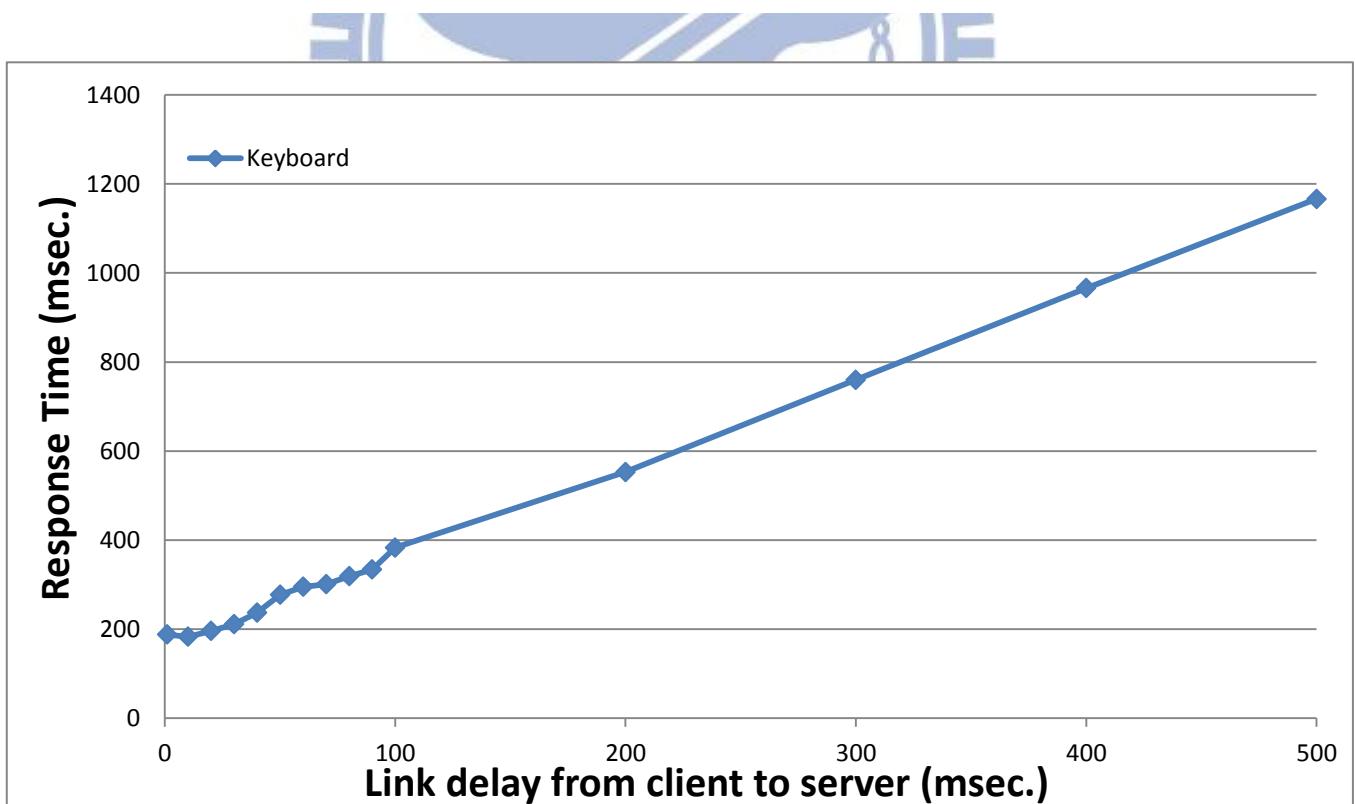


圖 18、實驗設計 B，線路延遲

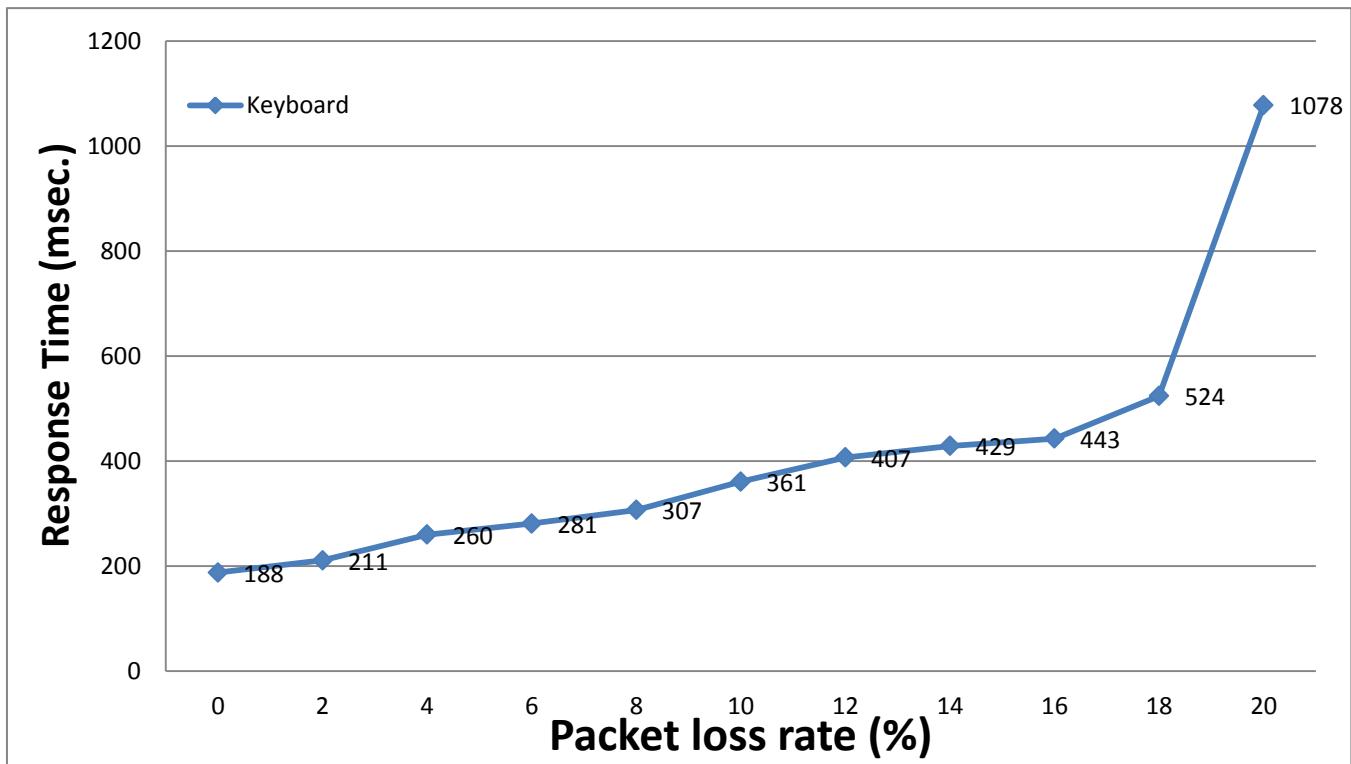


圖 19、實驗設計 B，封包遺失率



5.4.3 評估實驗設計 C 之結果

實驗設計 C 的設計目的是為了看出使用者在進行磁碟讀寫動作時，隨著伺服器的負擔不同，會有著什麼樣的影響，和實驗設計 A 與實驗設計 B 不同，我們只變化了伺服器上處理器、記憶體、磁碟讀取的資源。

圖 20~22 是我們評估工具在不同伺服器資源所得到的畫面回應時間，實驗是將一個約 600MB 的檔案壓縮，並觀察其壓縮完成後畫面的回應時間，共重複執行五次並將結果取平均值。

以下我們將說明我們所觀察到的情形，這些數據也可以給 VDI 服務提供業者參考，知道伺服器配置與使用者執行磁碟讀寫有著什麼樣的關聯。

圖 20，橫軸為開啟虛擬機器數量並執行我們設定消耗處理器資源的程序，縱軸為使用者壓縮檔案後畫面顯示檔案壓縮完成的時間，由此圖可以看出壓縮檔案由於過程中需要計算壓縮的資料，所以隨著處理器資源忙碌時，壓縮檔案所需的時間也上升，可以觀察到，當開啟的虛擬機器數量約 20 台時，由於 CPU 使用率約 100%所以在虛擬機器數量達到 20 時，壓縮時間也呈現較明顯的上升。

圖 21，橫軸為接收裝置使用的虛擬桌面所配置到的記憶體大小，縱軸為使用者壓縮檔案後畫面顯示檔案壓縮完成的時間，由此圖可以看出，記憶體對於壓縮檔案動作也有一定的影響，由於在壓縮時也需使用到記憶體資源，所以當記憶體不足時，必須一直做 Memory Swap 的動作，導致壓縮的時間也跟著拉長，所以綜合實驗設計 A、B、C 可以看出當記憶體不足時，使用者所有操作都會受到不小的影響。

圖 22，橫軸為開啟虛擬機器數量並執行消耗磁碟讀寫的動作，縱軸為使用者壓縮檔案後畫面顯示檔案壓縮完成的時間，如圖所示隨著競爭磁碟讀寫的數量變多，使用者壓縮檔案的時間也跟著提高，但當競爭數目變多時，時間並未跟著上升許多，但是我們有發現到，其它執行消耗讀寫動作的虛擬機器，所得到磁碟讀寫的效能是降低許多的，所以可以看出，但磁碟讀寫競爭激烈時。由於實驗設計 A 和 B 我們並未觀察到磁碟讀寫對於使用者的影響，所以藉由這個實驗可以看出磁碟對使用者操作的影響，可以提供診斷工具一個診斷的依據，即如果觀察到使用者在執行壓縮動作時間增加許多，就代表有其它虛擬機器也正在執行磁碟讀寫的可能。

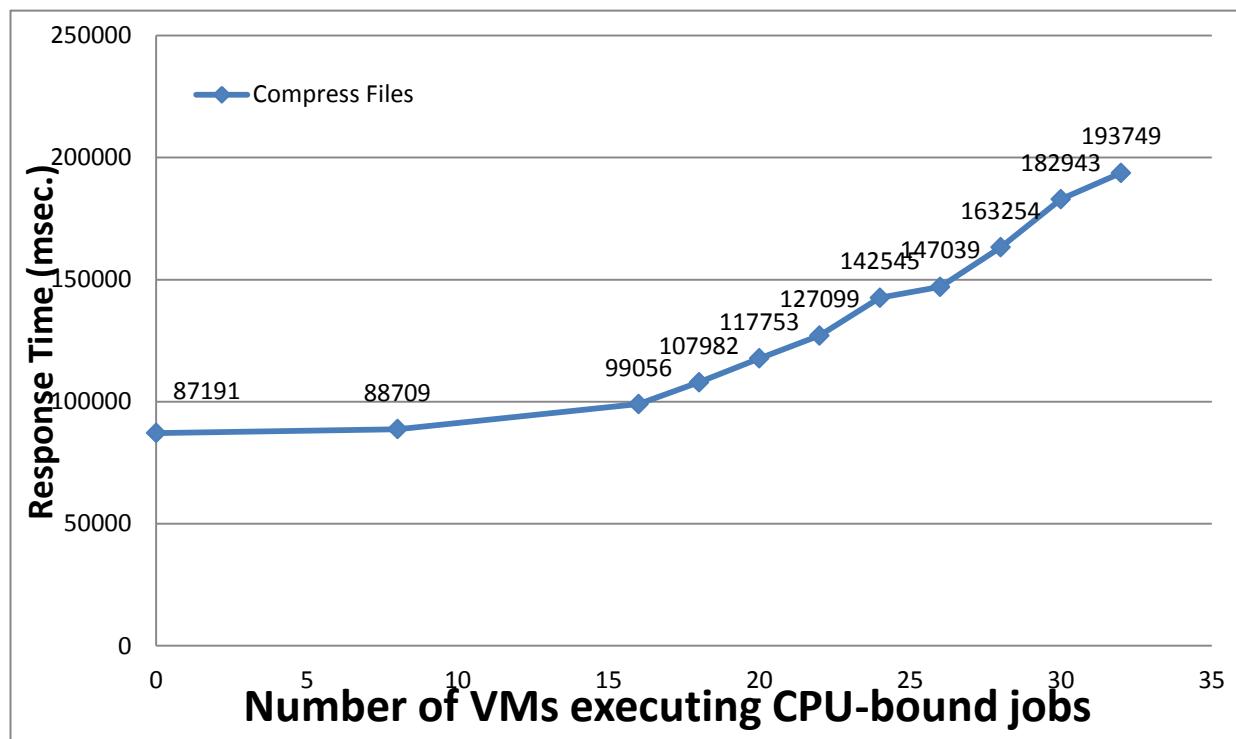


圖 21、實驗設計 C，伺服器處理器資源

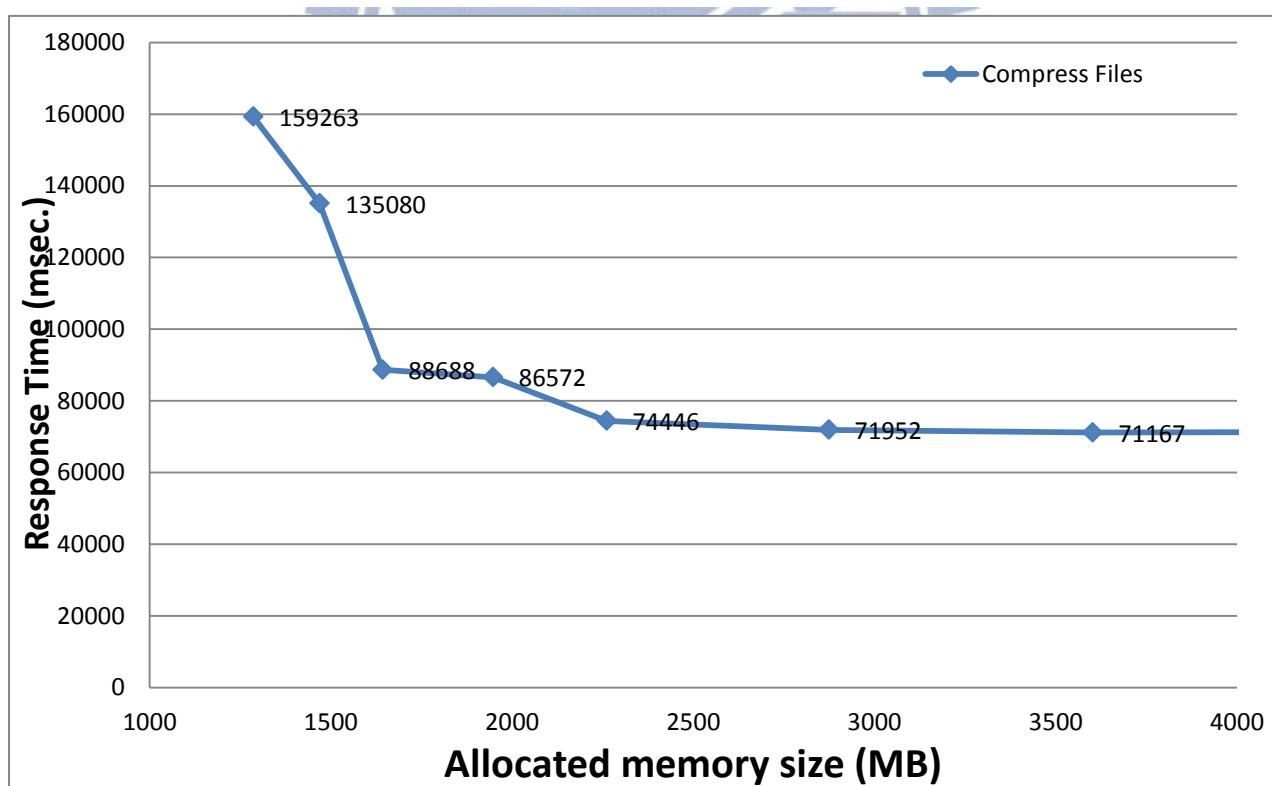


圖 20、實驗設計 C，伺服器記憶體資源

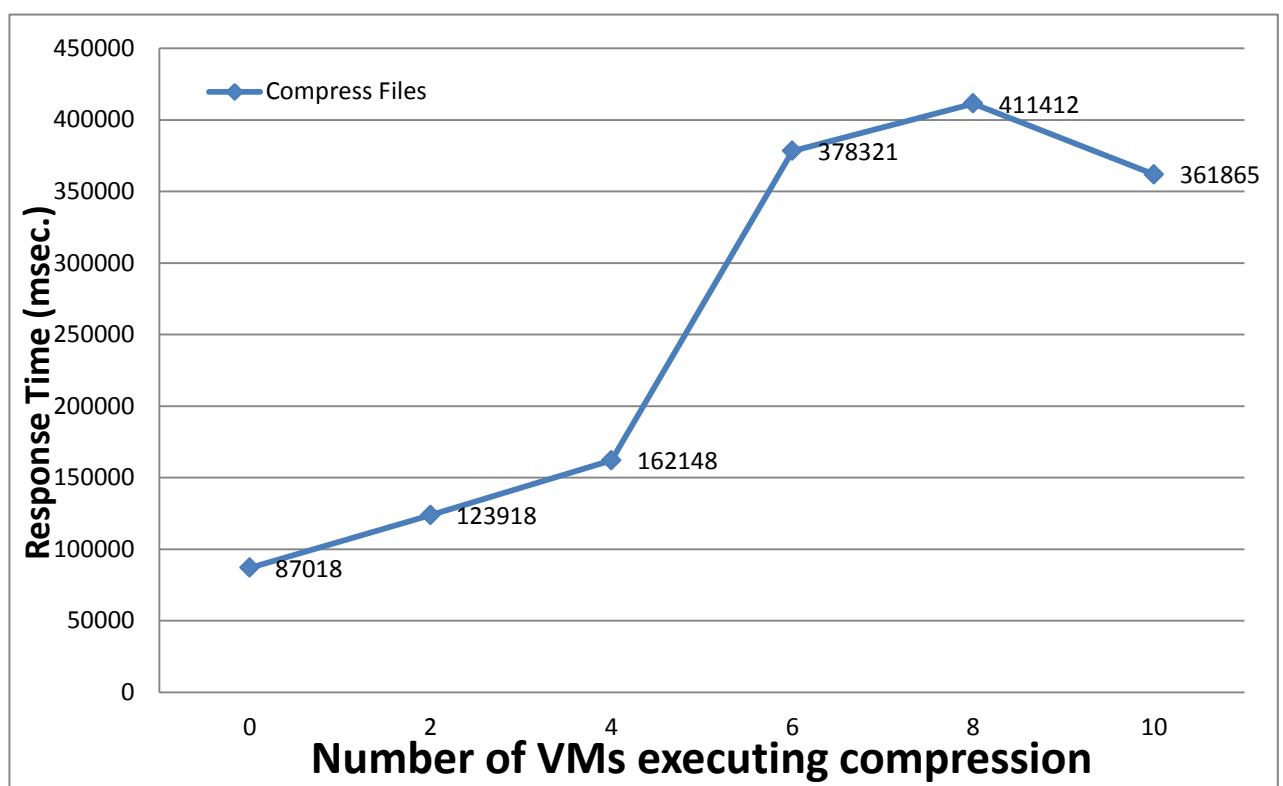


圖 22、實驗設計 C，伺服器磁碟資源



六、診斷工具實驗環境與實驗結果

6.1 研究與實驗方式

在本論文的研究中，我們主要想藉由我們設計的診斷工具，能在當使用者覺得因延遲而操作不順，或是感覺效能不佳的時候，讓使用者知道是何種原因造成的，以便他們能夠知道向誰反應以便尋求改善，再者也可能讓服務提供業者在伺服器端，得知使用者診斷的結果，知道伺服器上目前的使用者是否有效能不佳的情況，長時間的資料統計也能幫助他們更有效率的配置伺服器資源。

我們的實驗參數與評估工具所使用的類似，都是對於伺服器資源或是網路環境產生變化，我們以下將設計幾個實驗，將先產生單一的伺服器資源缺乏或是網路環境惡劣的情形，來測試我們的診斷工具是否能診斷出是何種因素造成使用者感受效能低落，

6.2 實驗情境參數

由於我們在診斷工具所使用的實驗環境和實驗參數與評估工具是一樣的，實驗參數一樣分別為處理器資源、記憶體資源、磁碟讀寫資源、線路延遲和封包遺失率，但不同的是我們在診斷工具的實驗中，有些實驗一次會設定兩個實驗參數，主要是觀察我們的診斷工具是否能診斷出當有多種因素同時造成使用者效能感受不佳。

另外我們將藉由圖 23 來說明診斷結果是如何呈現的，我們診斷結果主要分成四大部分，由上至下分別為伺服器資源使用狀況、使用者虛擬桌面資源使用狀況、接受裝置與伺服器間的網路狀況和我們診斷是何種因素造成功能低落的結果。伺服器資源又分為處理器使用率、目前記憶體分配情形和伺服器整體磁碟讀寫效率，可以讓使用者了解目前伺服器的情形。而虛擬桌面資源一樣分成處理器使用率、目前記憶體分配情形和磁碟讀寫效率，可以讓使用者了解被配置的虛擬桌面目前的狀況。而網路狀況則讓使用者得知伺服器之間封包的 RTT (Round-Trip Time)和封包遺失率。而最下面則是我們工具診斷的結果，我們將可能的因素分為五項，分別為處理器、記憶體、磁碟、線路延遲和封包遺失率，後面的 X 越多，代表此因素越有可能是造成功能不好的原因，一共從兩個 X 到五個 X，沒有 X 代表狀況良好，而五個 X 是代表此資源嚴重不足或是環境惡劣。

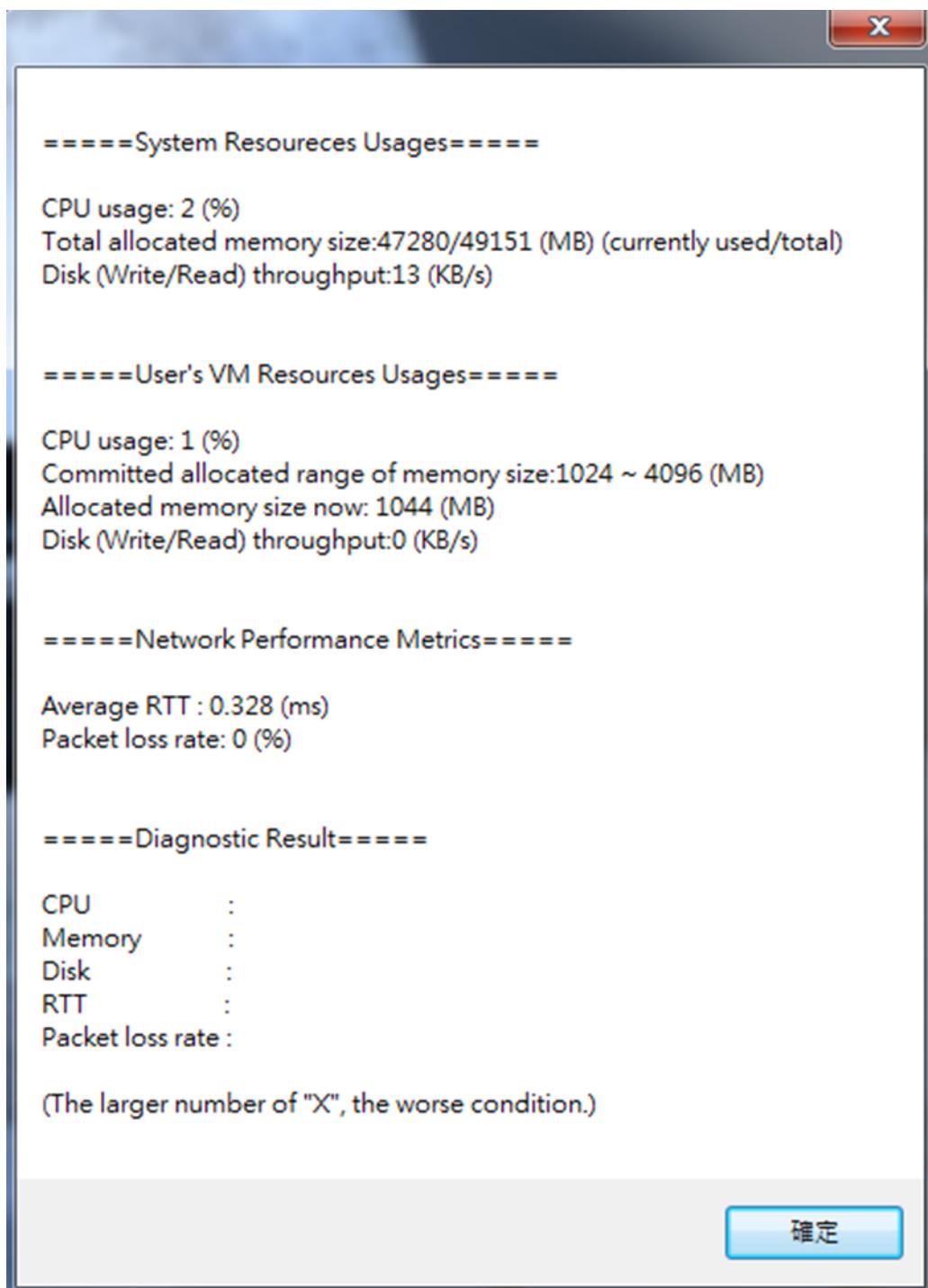


圖 23、診斷結果說明

6.3 診斷實驗設計

在本論文的診斷實驗規劃中，我們設計了幾個不同資源缺乏或是網路狀況惡劣的情境，期望我們的診斷工具能正確指出造成使用者效能的因素，各種診斷實驗案例的設計目的、說明、以及該設計中使用的實驗情境參數種類分別說明如下：

6.3.1 診斷實驗設計 A

診斷實驗設計 A，是針對如果目前有一項因素造成使用者感受效能不佳，我們所設計的診斷工具是否能正確的判斷出來，這種情形是雖然是最簡單診斷但也是最常發生的，因為使用者通常不會等到有兩項因素才察覺到目前虛擬桌面正遭遇一個效能不好的情形，所以只要能在有一項因素造成使用者感受不佳時就能正確診斷，就能避免等到多項因素惡劣時讓使用者感受更延遲的情形，診斷實驗設計 A 的設計目的以及說明請見表 8。

6.3.2 診斷實驗設計 B

診斷實驗設計 B，是針對如果目前有兩項因素造成使用者感受效能不佳，我們所設計的診斷工具是否能正確的判斷出來，如果同時有多項因素造成使用者效能不佳，對我們的診斷工具來說是相對困難的，因為多項因素會造成我們在效能量測上，很難判斷因為疊加所造成效能惡劣的情形會是如何，但我們在診斷流程上是有考慮如果有兩項因素的情形，因此我們想藉由診斷實驗設計 B，來測試我們診斷的結果是否正確，於診斷實驗設計 B 的設計目的以及說明請見表 9。

表 8、診斷實驗設計 A 之設計目的與說明

模擬設計編號	A
設計目的	為了能先測試診斷工具能診斷出最簡單的情況，也就是只有單一因素影響了使用者效能不佳的情形，這也可能是大多數使用者會遭遇的狀況，是由於一項因素資源缺乏所造成使用虛擬桌面時突然發生延遲。
實驗設計說明	在診斷實驗設計 A 中，我們設計了五種狀況，分別是處理器、記憶體、磁碟資源不足，以及線路延遲時間過長和在封包遺失率較高的網路環境下，測試我們的診斷工具是否能正確指出我們所設定的是哪一種狀況。

表 9、診斷實驗設計 B 之設計目的與說明

模擬設計編號	B
設計目的	為了能先測試診斷工具能診斷出較複雜情況，也就是有兩項不同因素同時影響了使用者效能不佳的情形，雖然這種情形可能較不常發生，但我們還是期望我們的診斷工具能判斷出，使用者由於多項資源不足或是網路環境惡劣，所以產生效能不佳的情形。
實驗設計說明	在診斷實驗設計 B 中，我們設計了三種狀況，分別是處理器資源不足加上線路延遲時間過長、磁碟資源不足加上高封包遺失率、記憶體資源不足加上線路延遲時間過長的情境，測試我們的診斷工具是否能正確指出我們所設定的是哪一種狀況。

6.4 診斷實驗結果與分析

6.4.1 診斷實驗設計 A 之結果

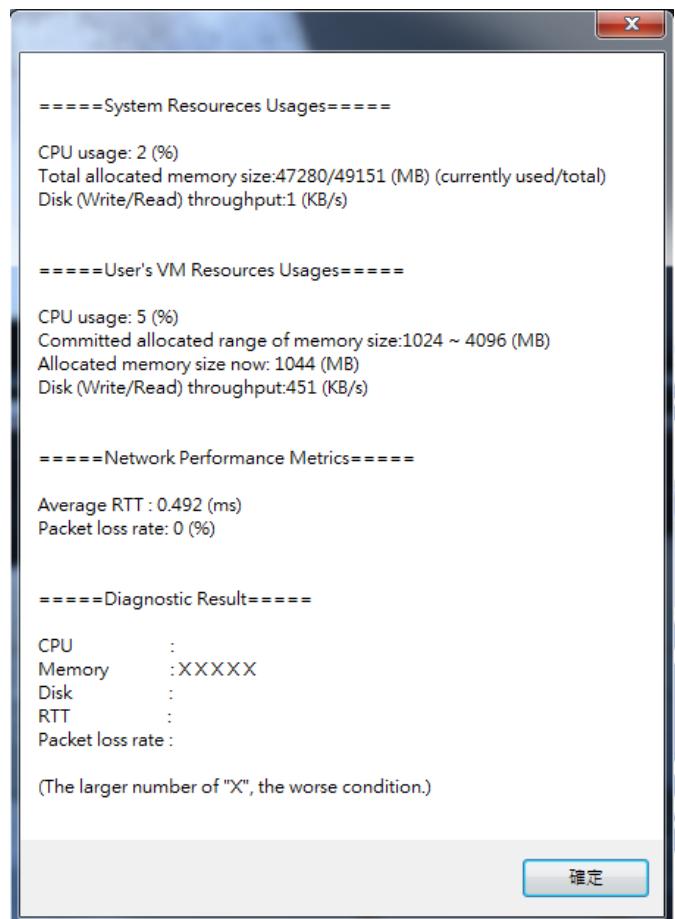


圖 24，我們將機器全數開啟，並在使用者的虛擬機器上執行佔用記憶體資源的應用程式，我們看到我們量測到開 Microsoft Word 的時間約比正常情況多了一倍，以及虛擬桌面磁碟讀寫忙碌，所以診斷結果顯示記憶體資源嚴重不足，為五個 X。



圖 24、診斷實驗設計 A、記憶體資源不足

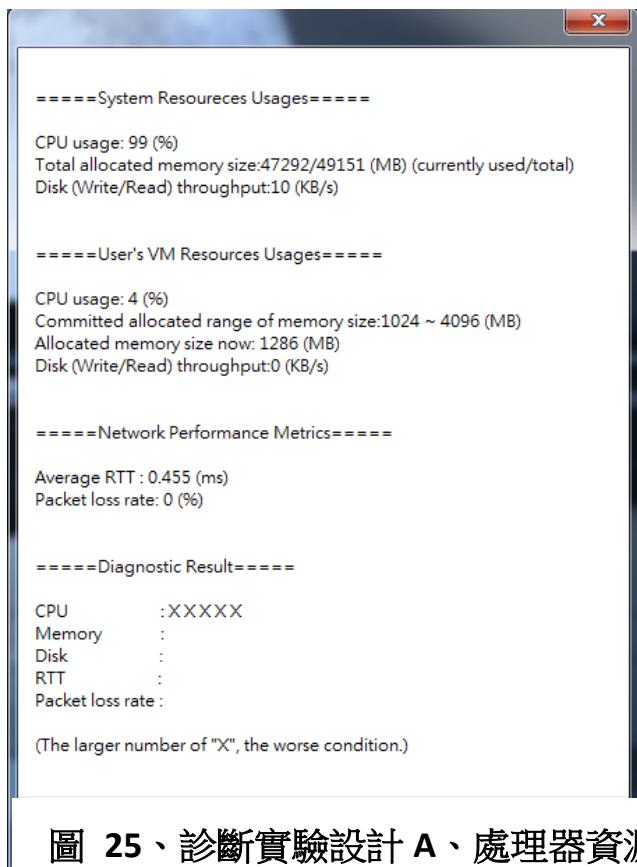


圖 25、診斷實驗設計 A、處理器資源不足

圖 25，我們在三十二台虛擬機器中，做消耗處理器的動作，並開始診斷工作，所以從圖中可以看出處於 99% 忙碌狀態，加上我們量測到開啟 Microsoft Word 的時間約多了 80%，所以診斷結果顯是處理器資源嚴重不足，為五個 X。



圖 26、診斷實驗設計 A、磁碟資源不足

圖 26，我們在十台虛擬機器中，做壓縮檔案的動作，並開始診斷工作，所以從圖中可以看出伺服器磁碟處於忙碌狀態，加上我們量測到壓縮檔案的時間比正常情況多了約四倍，所以診斷結果顯是磁碟資源嚴重不足，為五個 X。

圖 26、診斷實驗設計 A、磁碟資源不足

```

=====System Resources Usages=====

CPU usage: 2 (%)
Total allocated memory size:46032/49151 (MB) (currently used/total)
Disk (Write/Read) throughput:4 (KB/s)

=====User's VM Resources Usages=====

CPU usage: 1 (%)
Committed allocated range of memory size:1024 ~ 4096 (MB)
Allocated memory size now: 1286 (MB)
Disk (Write/Read) throughput:0 (KB/s)

=====Network Performance Metrics=====

Average RTT : 802 (ms)
Packet loss rate: 0 (0%)

=====Diagnostic Result=====

CPU      :
Memory   :
Disk     :
RTT      : XXXXX
Packet loss rate:

(The larger number of "X", the worse condition.)

```

圖 27、診斷實驗 A、線路延遲過長

圖 27 我們利用 EstiNet 設置了單向線路延遲 400 毫秒，並開始診斷工作，所以從圖中可以看出連線延遲約 800 毫秒，加上我們量測輸入特定字串等待畫面回應的時間也比正常時間高出四倍，所以診斷結果顯示是連線延遲過長，為五個 X。

```

=====System Resources Usages=====

CPU usage: 3 (%)
Total allocated memory size:46032/49151 (MB) (currently used/total)
Disk (Write/Read) throughput:7 (KB/s)

=====User's VM Resources Usages=====

CPU usage: 1 (%)
Committed allocated range of memory size:1024 ~ 4096 (MB)
Allocated memory size now: 1286 (MB)
Disk (Write/Read) throughput:0 (KB/s)

=====Network Performance Metrics=====

Average RTT : 3 (ms)
Packet loss rate: 25 (0%)

=====Diagnostic Result=====

CPU      :
Memory   :
Disk     :
RTT      : XXXXX
Packet loss rate: XXXXX

(The larger number of "X", the worse condition.)

```

圖 28、診斷實驗 A、封包遺失率過高

圖 28 我們利用 EstiNet 設置單向封包遺失率 15%，並開始診斷工作，所以從圖中可以看出封包遺失率約 25%，加上我們量測輸入特定字串等待畫面回應的時間也比正常時間高出一倍，所以診斷結果顯示是封包遺失率過長，為五個 X。

6.4.2 診斷實驗設計 B 之結果

此部分我們將介紹診斷實驗設計 B 之結果，實驗環境為多項因素造成使用者結果感受效能不佳，以下將個別說明。

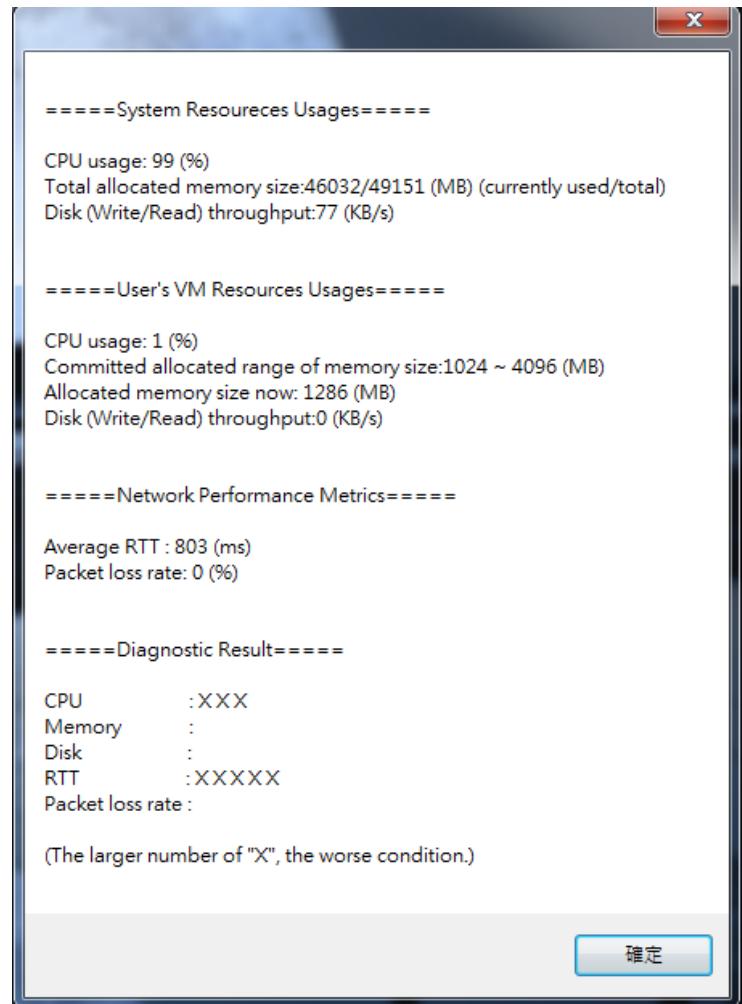


圖 29，我們在伺服器上二十六台虛擬機器中執行消耗記憶體的程式，並利 EstiNet 設置單向 400 毫秒的線路延遲，輸入特定字串等待畫面回應的時間比平常高且配合量測到的連線延遲為 800 毫秒，因此推論有嚴重連線延遲問題，接著扣掉因為線路延遲而造成開啟 Office 時間增加的部分，我們量測到開啟 Office 時間約多了 600 毫秒，因此推論處理器處於中度忙碌的狀態為三個 X。

圖 29、診斷實驗 B、處理器資源不足與線路延遲過長

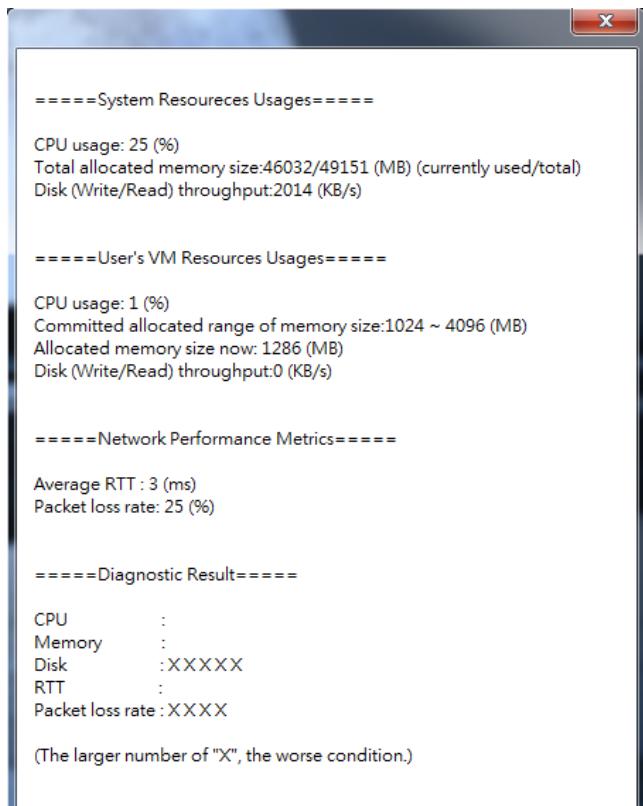


圖 30、診斷實驗 B、磁碟資源不足與封包遺失率過高

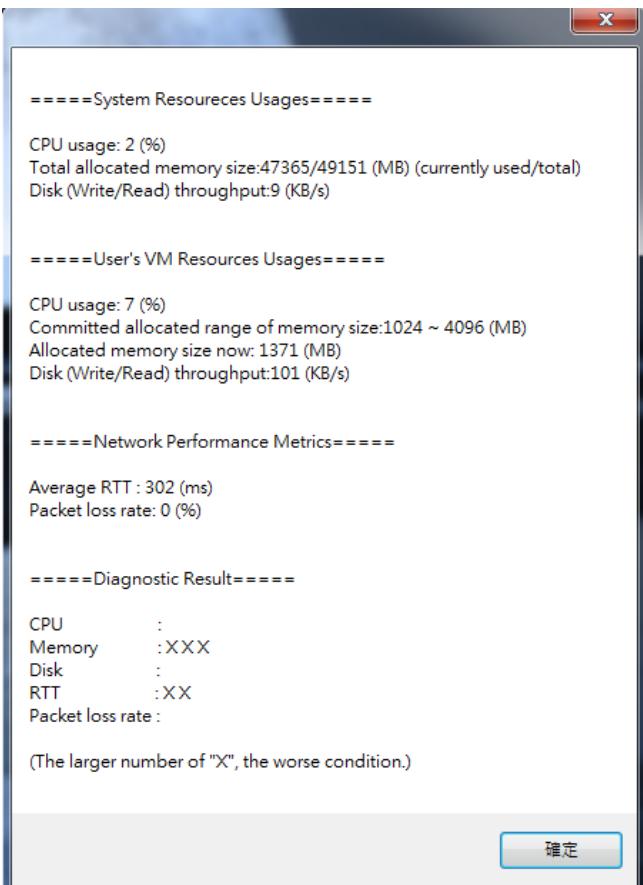


圖 31、診斷實驗 B、記憶體資源不足與線路延遲過高

圖 30，我們在十台虛擬機器中，及利用 EstiNet 設置單向封包遺失率 15%，我們由做壓縮檔案的動作時間太長以及伺服器整體讀寫忙碌推論有磁碟資源的問題，而量測到封包遺失率約 25%，加上我們量測輸入特定字串等待畫面回應的時間也比正常時間高出一倍多，所以診斷結果也顯示封包遺失率過長，為四個 X。

圖 31 我們將機器大多數開啟，並在使用者的虛擬機器上執行佔用記憶體資源的應用程式，以及利用 EstiNet 設定單向線路延遲 150 毫秒，我們量測到輸入特定字串等待畫面回應的時間約比正常時間多了 400 毫秒，加上測量到的連線延遲 302 毫秒，推論約有輕微的線路延遲問題，接著將量測到開啟 Microsoft Word 的時間扣掉 302 毫秒，我們得到開啟 Microsoft Word 的時間約比狀況良好時多了百分之五十，以及虛擬桌面磁碟讀寫有些忙碌，所以診斷結果顯是記憶體資源有些不足，為三個 X。

七、結論

隨著現今網路速度越來越快，雲端相關的服務也越來越熱門，而虛擬遠端桌面環境也是其中之一，已經有越來有多研究機構開始研究，如何能讓使用者更快速及順暢的使用遠端桌面，另一方面也有許多企業開始捨棄傳統電腦轉為使用這種架構。

在本論文中，我們實作了評估工具以及診斷工具，並針對在虛擬遠端桌面環境中各種伺服器資源或是網路環境，做不同的資源變化及使用者操作，再進行效能量測以及診斷工作，如此一來可以看出在虛擬遠端桌面環境下，各種資源或網路環境對伺服器造成的影响，而我們更進一步的利用效能量測的結果，作為診斷工具的參考資料，實作出能診斷多項因素且將其分級，並具有準確性的診斷工具。

我們將本論文中經由實驗、觀察所得到的結果整理成以下結論：

1. 處理器資源對於開啟、關閉和執行應用程式有著很大的關聯。
2. 記憶體資源對於開啟、關閉和執行應用程式也有著很大的關聯，但記憶體資源嚴重缺乏時，虛擬桌面各種操作也會遭受到極大的延遲。
3. 磁碟資源只影響需要磁碟相關之操作，如壓縮、解壓縮、檔案搬移等等..。
4. 網路環境影響每項使用者操作的回應時間，對於需要較快回應畫面的動作，網路環境影響的比例更加劇烈，如鍵盤輸入或是滑鼠操作等等..。
5. 本論文實作的評估工具及診斷工具可以協助虛擬遠端桌面服務提供業者，更直接掌握伺服器的狀況及使用者的效能感受，以便進行更即時的伺服器配置，並提供更佳的使用品質且容納更多的使用者。

八、未來工作

在本論文中，我們藉由變化不同伺服器資源以及網路環境對 Citrix XenDesktop 平台做了效能量測，但除了我們設定的項目外，還是有一些情形會導致使用者的感受不佳，而我們的診斷工具主要是依靠畫面比對相同，來確認由使用者送出的動作對應的畫面已正確呈現，所以無法在使用者操作時便進行診斷，有一些議題是可以再繼續深入研究的，以下詳述之。

1. 除了我們在本篇論文當中討論的幾個因素，處理器、記憶體、磁碟讀寫、線路延遲和封包遺失率外，可能還有其他會造成使用者效能感受不佳的因素，例如因為頻寬不足因素，導致使用者產生接收畫面延遲的現象，所以若能將更多因素列入診斷工具裡，就能讓使用者更放心的使用各種 VDI 平台。
2. 我們量測回應時間主要是依據，現在呈現的畫面與預期的畫面相同，所以我們必須得知，在做出操作時，畫面顯示的結果為何，此做法雖然有效但卻失去了很多彈性，例如若畫面突然出現與預期不符的結果，例如更新訊息或是執行的動作與預期有些許偏差，都有可能導致診斷過程中斷，另一方面也因為我們這個原因，所以我們並無法在使用者正常使用時便進行診斷，在診斷過程中使用者也無法進行操作。如果能在畫面比較上改良，例如如果有百分之八十以上相似就判斷為相同，可以使診斷更加順利，也可能可以達成在正常使用時就順便做診斷的動作，如此一來便增加了許多彈性，也能讓更加感受到診斷工具的方便性。

參考文獻

- [1] J. Rhee, A. Kochut, K. Beaty, “DeskBench: Flexible Virtual Desktop Benchmarking Toolkit”, *Proc. of Integrated Management (IM)*, 2009.
- [2] A. Berryman, P. Calyam, M. Honigford, A. M. Lai, “VDBench : A Benchmarking Toolkit for Thin client based Virtual Desktop Environments”, IEEE CloudCom, Indianapolis, IN December 2nd, 2010
- [3] J. Nieh, S. Yang, N. Novik, “Measuring thin-client performance using Slow-motion benchmarking”, ACM Transactions on Computer Systems, Vol. 21, No. 1, Pages 87-115, 2003.
- [4] S.Y. Wang, C.L. Chou, C.H. Huang, C.C. Hwang, Z.M. Yang, C.C. Chiou, and C.C. Lin, “The Design and Implementation of the NCTUms 1.0 Network Simulator,” Computer Network, vol. 42, no. 2, pp. 175-197, June 2003.
- [5] S.Y. Wang, C.L. Chou, and C.C. Lin, “The GUI User Manual for the NCTUms 6.0 Network Simulator and Emulator” is available at
<http://ns110.csie.nctu.edu.tw/support/documentation/GUIManual.pdf>
- [6] S.Y. Wang, C.L. Chou, C.C. Lin, and C.H. Huang, “The Protocol Developer Manual for the NCTUms 6.0 Network Simulator and Emulator” is available at
<http://ns110.csie.nctu.edu.tw/support/documentation/DeveloperManual.pdf>
- [7] S.Y. Wang and H.T. Kung, “A Simple Methodology for Constructing Extensible and High-Fidelity TCP/IP Network Simulator,” IEEE INFOCOM’99, March 21-25, 1999, New York, USA
- [8] N. Zeldovich, R. Chandra, “Interactive Performance Measurement with VNCplay”, Proc. of USENIX Annual Technical Conference, 2005.
- [9] H. Van, F. Tran, J. Menaud, “Autonomic Virtual Resource Management for Service Hosting Platforms”, Proc. of ICSE Workshop on Software Engineering Challenges of Cloud Computing, 2009.

[10] A. Lai, J. Nieh, “On The Performance Of Wide-Area Thin-Client Computing”, ACM Transactions on Computer Systems, Vol. 24, No. 2, Pages 175-209, 2006.

[11] Citrix Systems. Independent Computing Architecture Protocol.
<http://www.citrix.com/>.

[12] Citrix Corporation. Citrix Application Delivery Infrastructure.
<http://www.citrix.com>.

