

# 虛擬環境中自動化資源分配機制之設計原理與 績效分析

## (Design Issues and Performance Analysis of Automatic Resource Allocation Scheme on Virtualization Environment)

楊欣哲\*<sup>1</sup> 陳立群\*<sup>2</sup> 曾錫輝\*<sup>1</sup> 鍾惠光\*<sup>2</sup> 林旭陽\*<sup>1</sup>

\*<sup>1</sup>: 東吳大學資訊管理學系

\*<sup>2</sup>: 資策會創新應用服務研究所

Email: [sjyang@csim.scu.edu.tw](mailto:sjyang@csim.scu.edu.tw); [seth@iii.org.tw](mailto:seth@iii.org.tw); [97756020@scu.edu.tw](mailto:97756020@scu.edu.tw); [minolu@iii.org.tw](mailto:minolu@iii.org.tw); [jl@csim.scu.edu.tw](mailto:jl@csim.scu.edu.tw)

**摘要**—近年來，快速發展的虛擬化技術就是將電腦系統的資源，如 CPU 運算能力、記憶體空間以及應用程式抽離出來，虛擬化(Virtualization)其目的是要讓資源的使用方式更具效率，更靈活，利用此架構來取得更多的資源。因此，在一個實體主機上可同時執行多個虛擬機器(VM, Virtual Machine)，彼此之間的系統資源可動態地分配共享。

依據 RedHat Virtual Machine Manager 系統，本論文將提出並設計虛擬環境中自動化資源分配機制，稱之為 ARAS。因此，ARAS 主要的功能將包含：(1) 執行自動化虛擬機器的資源如 CPU、記憶體、磁碟空間之分配功能；(2) 對實體主機的資源利用率進行判斷與分析作為執行虛擬機器的線上遷移之依據；(3) 執行虛擬機器的線上遷移功能，以達到自動化虛擬機器的負載平衡。此外，將驗證 ARAS 機制之效益，藉由模擬實驗進行於 VM 下使用 ARAS 之績效評估與結果分析，以提高伺服器的資源利用率並提昇系統的處理能量。

**關鍵詞**：虛擬機器，自動化資源分配機制(ARAS)，自動化的負載平衡，線上遷移

**Abstract**—Recently, virtualization is designed to isolate the system resource such like computing power, memory space, and application. Also, virtualization technique can make the resource usage more efficient and dynamically allocate for utilization. Hence, the virtualization can make virtual machine (VM)

un-interrupt to be executed via direct migration between Physical servers and also make hardware maintenance and load balancing much better. Virtual architecture is to implement virtual layer on the host operating system, therefore the server can efficiently perform multiple VMs and also dynamically share the system resource between these VMs.

Based on the RedHat Virtual Machine Manager system, we propose and design new automatic resource allocation technique called ARAS (Automatic Resource Allocation Scheme). Hence, the primary features of ARAS model will cover: (1) to perform the automatic allocation of resource such as CPU, Memory, and disk on the VM; and (2) to indentify and analyze the resource availability on physical servers for migration task on the VM; (3) to perform live migration methods of VM for automatic balancing. This paper will validate the efficiency and effect of the ARAS, and also perform simulations for its performance analysis under using ARAS on the VM. The final simulation results indicate that increase the resource utilization and enhance the system throughput on the VM with ARAS.

**Keywords**: VM, ARAS, Automatic Load Balancing, Live Migration

## 一、前言

### 1.1 虛擬化介紹

傳統上，一台主機搭配一個作業系統以及執行一個應用程式，這樣的架構底下，往往一台主機的資源使用率都非常低，況且當前的主機運算效能大幅提高，讓這些資源閒置下來，實為可惜，所以最近幾年伺服器虛擬化的做法便應運而生。

在虛擬的 IT 環境中，最重要的工作莫過於資源的有效管理，以因應不斷變化的應用程式需求，這當中包含了系統資源的調度與監控，如動態調整資源和遷移虛擬機器。基本上，系統資源包含 CPU、記憶體、網路、磁碟儲存空間，一個虛擬機器上的應用程式有 CPU-intensive, Memory-intensive, I/O-intensive 這幾種類型，所以虛擬化資源管理系統須能做到即時(real-time)的偵測資源使用率及動態調整資源；並且當虛擬機器所在的實體主機資源不足時，能不停機執行線上遷移，將虛擬機器移到一個資源充足的實體主機，以提供虛擬機器正常運作所需的資源。

然而，由於虛擬環境的獨特性，資源的管理以全新的概念顛覆傳統的做法，一部虛擬機器要增加 CPU 及記憶體不再需要關機拆螺絲，只要實體主機還有閒置的資源就可以直接線上增加 CPU 和記憶體資源給虛擬機器使用，又或者是利用線上遷移虛擬機器來達到工作負載平衡的目的，本計劃的目標即是在現有的虛擬基礎架構上發展出一套最佳化的資源管理機制，當虛擬機器工作負載變大的狀況持續發生時能有效的調配系統資源提供給虛擬機器使用，並利用自動遷移虛擬機器來達到充份運用系統資源及平衡主機工作負載的重要任務。

### 1.2 研究動機及目的

伺服器虛擬化之後，在一台實體主機上可以同時運作多台的 Guest OS，如此一來可以節省企業很多的硬體採購費用支出及維護成本，所以很多企業組織對於引進此一新興的技術，都有著相當濃厚的興趣，當前較為成熟的產品有 VMware ESX server, Xen 及 Microsoft HyperV，為了能深入的探討及學習虛擬化，並了解各家所長，所以我們選擇了虛擬環境下的資源管理為研究重點，希望能提出一個能有效增進應用程式績效及提高資源使用率的方法，並達到以下二個目的：

#### ● 簡化虛擬環境資源管理

由於虛擬環境的特殊性，傳統的資源管理方式已無法完全套用在此一新興發展的技術，隨著 CPU 運算能力的不斷提昇，在此架構下，我們重視的是資源的分配與管理，並可衡量各項績效資料，包括 CPU、記憶體、網路和儲存空間使用率等方面數據，並根據這些資料，以自動化的方式動態供應虛擬機器所需的資源。並且在主機資源不足時也提供管理者快速簡單地遷移虛擬機器，將之移動到另一部資源充足的實體主機，以獲得更多的可用資源。

#### ● 提高虛擬架構在 IT 營運系統的可用性

為了提高虛擬架構在資料中心的營運系統的可用性，除了考量系統資源的有效管理之外，還必須考慮到當實體主機遭遇計劃性停機如例行性硬體維護檢測，或非計劃性停機如突然發生硬體故障時，對於運作在上面的虛擬機器能做到保護其不受這些因素影響而發生中斷服務的情形，建構資料中心虛擬化即可輕易的解決上述的問題，大大地提高 IT 營運系統的可用性。

欲達到上述之目的，本論文將提出並設計虛擬環境中自動化資源分配機制，稱之為 ARAS (Automatic Resource Allocation Scheme)。此外，ARAS 機制亦提供線上遷移 (Live Migration) 虛擬機器的功能。當實體主機

需要進行維護時，可以由管理者下達遷移的指令，以不停機的方式把一台虛擬機器遷移到另一部運作中的實體主機，在維護工作完成系統恢復運作後，再以同樣的方式將虛擬機器遷移回來。

本論文章節內容結構如下：第二章為文獻探討並提出與本文相關的研究；第三章提出並說明 ARAS 之系統架構及設計；第四章說明模擬環境及模擬程序，並依據模擬實驗結果進行績效評估與分析；第五章為結論並說明未來研究方向。

## 二、文獻探討及相關研究

目前在虛擬化方面，有如雨後春筍般的一些相關研究提出來，首先在資源管理方面[3, 7]，資源的管理有廣義上有兩種方法，一種為非約束型，另一個為約束型的；在非約束型的方法中，一部虛擬機器的資源使用量不會被限制，它可以使用不超過這台實體主機的所有資源，這個方法很容易建構，但有些不切實際，因為這樣會造成其他重要的虛擬機器得不到任何資源，而造成寡佔資源的問題，因為它沒有資源使用優先權順序的觀念。第二種方法為約束型的，這個方法有三個優點，第一項優點是每一部虛擬機器的資源使用量都有一定的限制，不會無限制的取得主機資源，管理者可依據虛擬機器的重要程度不同而有不同的使用量設定，第二項優點是設定使用量的上限後，我們將可很容易的估算系統資源的使用量及剩下多少可運用的資源，第三項優點是對於資源的運用是可預測的(Predictable)。

研究和探討 Xen 的虛擬化，我們不應該忽略 VMM (Virtual Machine Monitor) 排程器[4, 9, 11]，它對應用程式 I/O 績效的影響甚巨，在傳統的做法中，VMM 排程器重視的是資源分配的公平性，而且在運算密集型的環境有不錯的

效果，但是如把頻寬密集型和延遲敏感型的應用程式也考量進來的話，這樣的做法很有可能會造成不可預期的應用程式績效低落問題，藉由對 VMM 進行研究[1, 4]，得到一個結論就是如果把運算密集型和延遲敏感型的應用程式分別執行在不同的虛擬機器上，就會得到較佳的運作績效。

有一些學者提出了適用性的資源控制系統[6, 7]，可動態地調整應用程式資源的使用比率 (即 share 值)，針對應用程式設定它可使用的資源的比率，來達到應用程式的 QoS (Quality of Service) 要求，並且是第一個提出對於 Multi-tier 的應用程式環境來做探討的論文。作者之系統中有兩個主要的控制器，一個是使用率控制器，可用來為個別應用程式設定使用比例的控制器，另一個是仲裁控制器，在應用程式發生資源競爭時，系統會依據各個應用程式的 QoS 差異性設定來仲裁資源的分配。但此系統的管理，都必須由管理者手動來調整及設定。

對於資源自動控制的方法，為了要達到服務等級目標(SLOs, Service Level Objectives)，應用程式的資源分配必須要能做到自動化[2, 8]。它包含四個方面，第一是績效保證，在系統資源充足的情形下，每一個應用程式都應該得到足夠的資源，反之，如果在資源飽和的狀況下，就要依據各別應用程式的優先權來分配資源，以保證高優先權的應用程式能獲得足夠的資源，以保障績效；第二是自動化，自動化控制的功能可以手動啟用或被關閉，但資源的分配則必須要自動化；第三是適用性，自動化控制的機制要可以適用於各種不同的負載和系統；第四是延展性，自動化控制能夠延伸到管理多個節點及多個應用程式。

其實除了實體主機本身的資源管理之外，我們還必須考量到資源的使用不是取之不盡用之不竭的，當一台實體主機的資源耗盡，

再也無法分出一點資源給應用程式時，這時我們就必須要利用虛擬化的另一個重要功能——線上遷移虛擬機器。有學者提出一個有效運用資源的自主式學習的虛擬機器動態遷移架構 [5,10]，當實體主機的 CPU 和記憶體使用率到達某一個門檻值時，虛擬機器會自動執行虛擬機器線上遷移的動作，來平衡每部主機上的工作負載，並依據歷史數據來記錄遷移前後對虛擬機器的績效影響為正向或負向，再從這些數據中學習到較佳的資源分配方法。所以門檻值的設定就非常的重要，但這個研究有一個問題是當我們把負載過大的機器遷移到別台主機上時，那這台主機會不會也是有整體負載過大的問題，而又再次發生遷移，如此便會造成系統的不穩定，即所謂的乒乓球問題。

綜合上述的文獻探討，我們了解很多有關虛擬環境中的資源分配的方法及觀念，每一位作者所提出的作法，都值得我們效法，亦皆其理論基礎。然而有些研究大多著重於資源的分配而缺少自動化，有些則運用線上遷移虛擬機器來達到負載平衡，但忽略了自動化的資源分配。有鑑於此，我們提出自動化資源分配機制：ARAS，亦即為達到自動化資源分配及負載平衡的目的。

### 三、ARAS 之架構及設計

#### 3.1 虛擬化架構之挑戰

當某一個虛擬機器上的應用程式在某一段時間，需用到大量的系統資源時，一個靈活的自動化資源管理系統，須能做到及時動態的調整資源的使用，並且當伺服器上的系統資源所剩不多，工作負載也很重時亦能不停機的遷移虛擬機器到另一部資源充足的實體主機，以有效且平衡地運用其資源。基本上，有關虛擬環境中的資源分配與管理之議題將衍生的一些挑戰與問題如下：

- 確認系統可用資源，在我們能管理資源之前，首先，我們必須要知道有多少的資源可以被支配使用，這當中包括全部的資源，當前已使用的資源，未來可能被使用的資源（預先保留給某系統）以及剩餘資源，這是個基本的課題，但也最常被忽略。
- 依時間因素而動態改變的資源需求，我們發現大部份的系統應用程式會在某些時間點發生大量的工作負載，有些有規律性，有些則否，所以我們要運用何種工具來偵測這些事件的發生，及如何正確的執行資源的分配，則是一大挑戰。
- 如何滿足多樣的服務滿意度指標，哪些虛擬機器的績效指標需要被滿足？我們可以動態的分配資源給需要的虛擬機器使用，但一旦資源緊張時，發生資源競爭時要如何協調優先權，便是一個很重要的工作。

欲解決上述這些挑戰，提出了 ARAS 機制，對於 CPU-intensive, Memory-intensive, I/O-intensive 的虛擬機器，在發生資源不足的情形時，可以線上提供一個資源充足的運作環境，讓事件發生對企業營運系統的績效影響降到最低，為企業提供一個可靠穩定的虛擬化基礎建設。

#### 3.2 設計原理

本論文之 ARAS 是基於 RedHat Virtual Machine Manager 架構而加以設計，其主要之功能在於管理及監控虛擬機器。然而，對於系統資源之分配方面則相對欠缺，所以我們於此基礎上提出一套結合此 VM 管理工具之自動化資源分配機制，以提高管理之便利性及實用性。

在前面的相關研究中，我們了解到 Xen 虛擬機器在新建立的時候，其資源的設定值中有一項 limit 值，它是屬於約束型的資源分配方法，為的就要有效的控管資源使用量以及可以

很輕易的估算總體資源的使用情形，這對於 ARAS 機制的運作是很重要的一項依據。因此，ARAS 才能做到：(1) 執行自動化虛擬機器的資源如 CPU、記憶體、磁碟空間之分配功能；(2) 對實體主機的資源利用率進行判斷與分析作為執行虛擬機器的線上遷移之依據；(3) 執行虛擬機器的線上遷移功能，亦即進行全自動，半自動或手動等遷移方式，以達到自動化虛擬機器的負載平衡。

在我們的環境中，將利用 Red Hat Virautl Machine Manager，並加之修改以符合實驗所需的管理要求，我們稱之為 DynaVirtual 管理員，DynaVirtual 提供了使用者一個圖形化的介面來管理運行於伺服器上的虛擬機器。透過本程式，管理者將可以開啟、關閉、暫停、新增、移除、編輯、監控、儲存並回復虛擬機器。同時，DynaVirtual 也提供了獨一無二的警示功能，我們稱為「服務水準協議(SLA, Service Level Agreement)」。一旦設定好警示條件，程式將可幫你監控虛擬機器。並會在警報發生時，寄 email 通知管理者和顯示在介面裡並記錄起來如圖 1 所示。

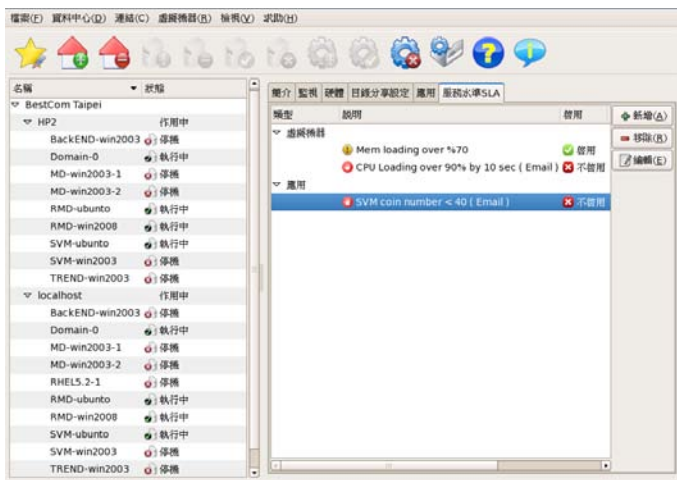


圖 1、服務水準協議(SLA)

SLA 警示條件的設定的是一個門檻值，當警示條件滿足持續一段時間之後，該警示才會被發佈，如此可避免重覆發送警示的情形發

生。

此外，為了能夠獲得較準確的數據，我們必須於虛擬機器上安裝監控軟體。一旦 DynaVirtual 和監控軟體連結之後，管理者就可在監視頁面裡清楚地看到虛擬機器上作業系統的 CPU、記憶體和磁碟的使用量，如圖 2 所示。

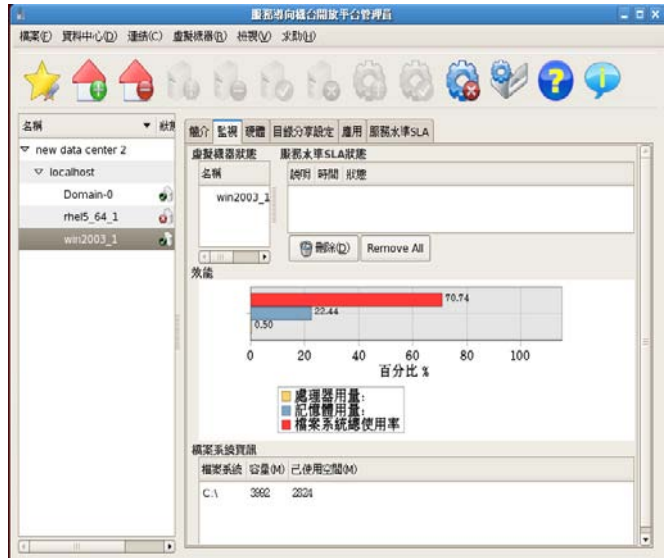


圖 2、虛擬機器的資源使用量

我們有了監控資源使用量的工具之後，就可使往後的研究工作更順利地進行，因為要分配資源，前提是要能即時的偵測在資料中心 (Data Center) 的虛擬化環境裡的資源使用情形及可支配的剩餘資源，再搭配 ARAS 機制，預期將可達到自動化資源分配的目的，其運作原理如圖 3 所示。

假設，當虛擬機器的 vCPU (Virtual CPU) 或記憶體使用率超過使用者預訂的門檻值時，就會啟動事件控制器(Event Controller)，並依據設定會觸發一個動作，像是發送電子郵件或 SNMP trap 通知管理者，由管理者自行手動調整資源的使用，或者是呼叫 ARAS 的資源分配器(Resource Allocate Controller)來自動調配資源給此 VM，資源分配器在收到警報通知後，會主動偵測該實體主機的剩餘資源是否充

足，如果已經沒多餘的資源可供調配，這時本機制便會自動執行線上虛擬機器遷移的動作，以獲取足夠的資源；另一方面，如果本機的資源是充足的，那就會依據警報的內容對於所需的資源如 CPU、記憶體自動地增加。

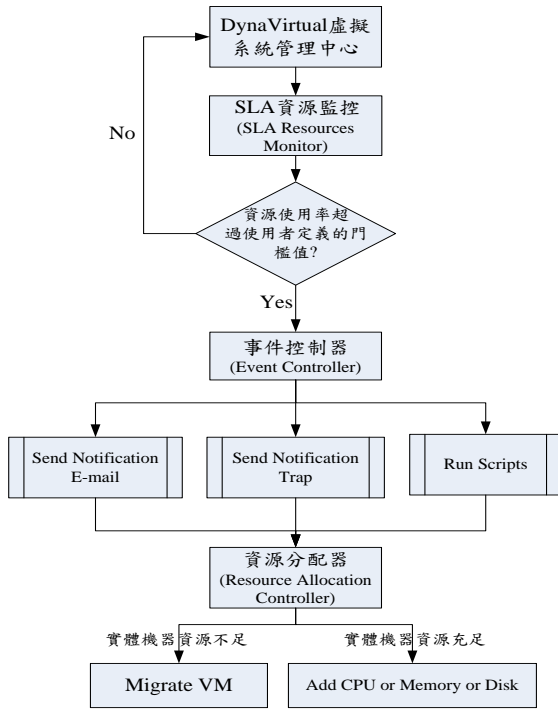


圖 3、ARAS 運作原理

依據上述之運作原理以設計 ARAS 機制，其演算法之虛擬碼(Pseudo code)如下：

**Algorithm ARAS**

**BEGIN**

**Input:**

//User have to pre-define resource usage rate.

UserDef.CPU.UsageRate as integer

UserDef.Memory.UsageRate as integer

UserDef.Low, UserDef.Medium, UserDef.High as integer

**Output:**

VM will obtain more CPU or memory resources to perform Live Migration

**Method:**

Host.CPU.UsageRate as integer

Host.Memory.Unused as integer

Host.Disk.Usage as integer

Get UserDef.CPU.UsageRate

Get UserDef.Memory.UsageRate

Get UserDef.Low, UserDef.Medium, UserDef.High

while

// Execute SLA Resources Monitor in DynaVirtual Virtual System Management Center

if VM.CPU.UsageRate >= UserDef.CPU.UsageRate or VM.Memory.UsageRate >= UserDef.Memory.UsageRate

// Event Controller (ARAS-EC)

switch action.type

case "1": send notification email

case "2": send notification SNMP trap

case "3": run script

case "4": call resource allocate controller

end switch

// Resource Allocation Controller (ARAS-RAC)

switch event.type

case CPU-intensive

if Host.CPU.UsageRate < UserDef.High

Add CPU to VM

else

Migrate VM to other Host

endif

case Memory-intensive

if Host.Memory.Unused > UserDef.Medium

Add Memory to VM

else

Migrate VM to other Host

endif

case I/O-intensive

if Host.Disk.Usage is UserDef.Low

Add another Disk to VM

else

Migrate VM to other Host

endif

end switch

```
endif
loop
End ARAS.
```

## 四、模擬實驗及績效評估

### 4.1 模擬環境與執程序序

本論文所實做的虛擬平台是基於 Xen 的架構，Xen 虛擬化是最快和最安全的虛擬化基礎架構解決方案，支援廣泛的虛擬機器作業系統，包括 Windows, Linux, Solaris 以及各種版本的 BSD。透過 Xen 虛擬化，Xen Hypervisor 是一個在伺服器硬體和虛擬機器作業系統之間的虛擬層。它負責 CPU 排程和分配記憶體給所有虛擬機器使用，這個虛擬層的建構，使得每台實體主機上可以執行一個或多個虛擬機器。本實驗環境中有三部實體主機，其中二部主機安裝了 Xen Server 虛擬平台，第三部主機則設定成 iSCSI target Server，做為存放虛擬機器檔案之儲存空間，每一部 Xen Server 主機上各建立 3 台虛擬機器，每一台虛擬機器上分別安裝 Windows Server 2003、CPU burn-in 燒機程式、PC wizard 2007 以及 iometer，如圖 4 所示。

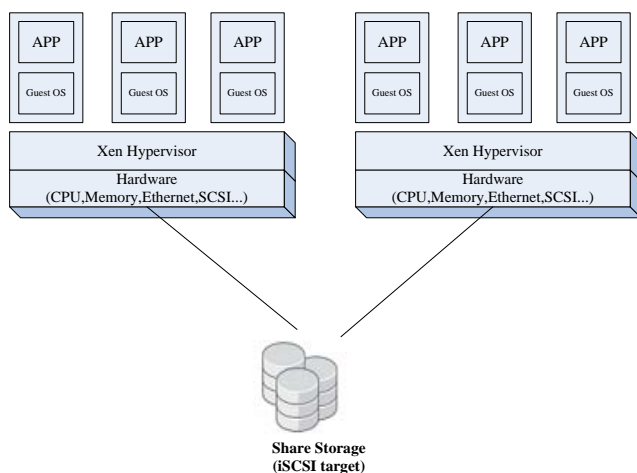


圖 4、實驗環境系統架構

本實驗之執程序序依據資源績效量測之

不同而區分為 CPU 資源實驗、記憶體資源實驗及 Disk I/O 實驗，分別敘述如下；

### (一)、CPU 資源實驗

在 CPU 的績效監控方面，為了使實驗環境單純化，減少干擾，我們只有一個 CPU 變數，其餘的硬體元件資源皆為固定，實驗步驟如下；

步驟 1. 選擇一部主機，在上面的 VM 裡執行 CPU Burn-in 程式，以產生 CPU workload。

步驟 2. 啟動 DynaVirtual 監控程式，設定當 VM 的 CPU 使用率超過使用者預訂的門檻值時，便發送 E-mail 通知管理者。

步驟 3. 管理者增加 CPU 給 VM。

步驟 4. 利用 PC wizard 2007 測試 VM 增加 CPU 後的 benchmark。

### (二)、記憶體資源實驗

在記憶體的資源分配方面，我們假設 VM 系統中有 Memory-intensive 的應用程式。實驗步驟如下；

步驟 1. 啟動 DynaVirtual 監控程式，設定當 VM 的記憶體使用率超過使用者預訂的門檻值時，便發送 E-mail 通知管理者。

步驟 2. 管理者增加記憶體給 VM。

步驟 3. 監測 VM 記憶體數據。

### (三)、Disk I/O 實驗

在磁碟資源方面，我們的實驗為監控磁碟讀寫使用頻率為主，為了方便監測磁碟使用，每一部 VM 裡除了系統磁區(C 碟)之外，還必須要有一顆存放資料硬碟(D 碟)，以做為實驗磁碟，其步驟如下；

步驟 1. 在每個 VM 裡新增一顆硬碟。

步驟 2. 在第 1 台 VM，利用 iometer 對此新硬碟產生持續性的讀寫工作，此時不要啟動第 2 台 VM 的 iometer。

步驟 3. 使用 iometer 監測各項磁碟 I/O 數據。

步驟 4. 約十分鐘後，啟動第 2 台 VM 的 iometer，做為產生 Disk I/O 競爭之目的。

步驟 5. 再次記錄第 1 台 VM 裡 iometer 的各項磁碟 I/O 數據。

#### 4.2 實驗結果與績效分析

從上述實驗中，我們發現套用 ARAS 機制的虛擬機器，在需要大量資源的時間段時，可以即時的得到資源，以滿足運算之所需，在 CPU 的方面，我們得到的實驗數據分別如圖 5 及圖 6 所示。一開始我們只有單顆 CPU 在運作，當 VM 的 CPU 運算資源緊張時，我們再即時增加一顆 CPU 以提高系統運算效能，實驗結果其數據顯示，虛擬機器在獲得 2 顆 CPU 之後的處理速度從 13478Mips 變成 26814Mips 增加了約一倍，且 CPU 的反應時間也從 17.89 秒變成 8.938 秒足足快了一倍。

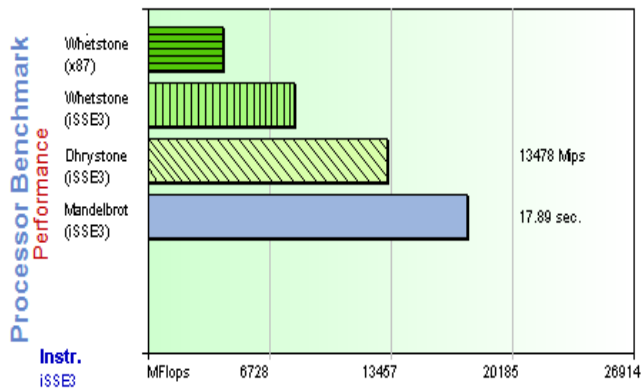


圖 5、單顆 vCPU Benchmark

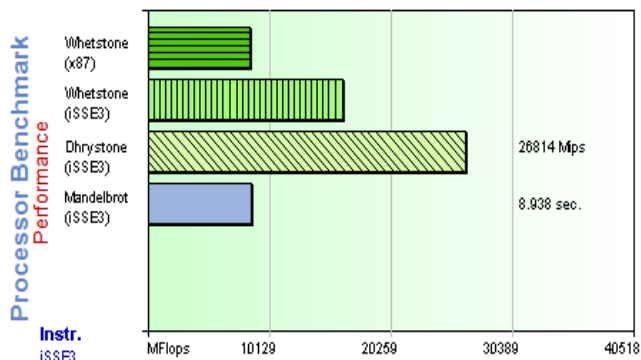


圖 6、2 顆 vCPU Benchmark

在記憶體方面，由於績效檢測工具程式，僅針對硬體做測試，在我們的實驗中我們是增加記憶體，像是 memory size 從 512MB 增加到

1024MB，而非更換實體主機的記憶體模組，所以跑出來的 benchmark 是一樣的，如圖 7 所示，但我們都知道增加記憶體對於 Memory-intensive 的應用程式的執行績效是絕對幫助的，所以我們計劃加強在應用程式方面的績效測試，以證明 ARAS 之效益。

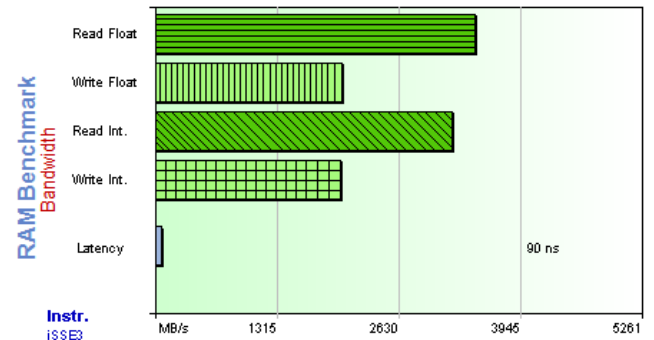


圖 7、記憶體 Benchmark

在磁碟讀寫的測試方面，依據上述所提的實驗步驟，我們發現當第 2 台 VM 也啟動 iometer 時，第 1 台 VM 的 Disk 使用率很明顯的降低了(如圖 8)，這是因為有其他的 VM 和它在搶 Disk 的 I/O 資源，所以我們的 ARAS 以此為依據，當監測到有多台 I/O-intensive 的 VM 同時在一部實體主機運作時，我們便會採取行動，像是線上遷移虛擬機器到另一部主機上，或是讓這些 VM 走不同路徑來存取磁碟資源，以提高 I/O 的效能。

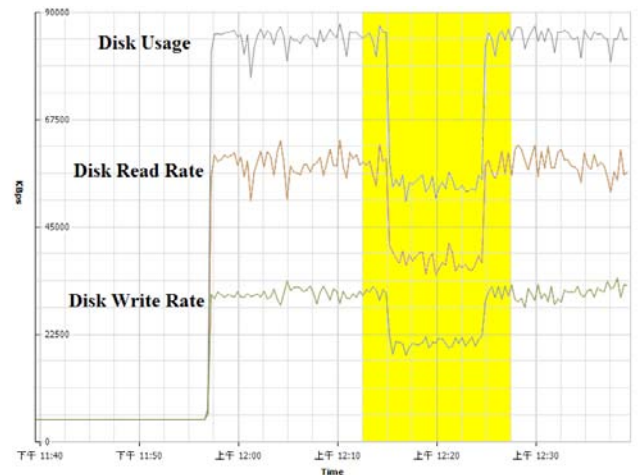


圖 8、磁碟使用率



由圖 8 中的心電圖數據由上而下為依序為 Disk usage、Read rate、Write rate，黃色區域的部份是當我們啟動第 2 台 iometer 時，第 1 台 VM 的各項 I/O 數據馬上就降下來，因為此時有多台的 VM 在存取共同的 I/O 資源，之後我們採取將第 2 台 VM 遷移到另一部主機的方式來化解此資源之競爭，此時我們觀察到第 1 台的磁碟使用情形，又上升恢復到最初的狀況，這證明我們 ARAS 所採用的線上遷移 VM，以減少資源競爭的方法是有效的。

## 五、結論

今日的伺服器不管是處理器效能、記憶體容量、網路速度以及儲存空間都遠遠超過使用者的需求，所以為了有效地運用這些過多的資源，大多數使用者優先考慮使用伺服器虛擬化，但伺服器虛擬化之後身為管理者更應做好系統資源的分配管理，才能保證這些虛擬機器的運作績效。

本論文以現有之 Xen 虛擬化資源管理基礎上，把原本需要使用者手動介入的部份減到最小，並期望做到自動化的資源分配機制，讓虛擬化在未來的應用上更加普及。在自動化的資源分配機制上，對於我們所提到的重要系統資源如 CPU、記憶體、磁碟儲存空間，我們提出的 ARAS 機制確實已可做到這些資源自動化分配，但在實驗中發現，雖然系統資源可線上直接增加，但 Guest OS 本身須具有 Hot Add 的功能配合以偵測並更新其系統資源，目前 Windows Server 2003 Enterprise 以上的版本或 Redhat Enterprise Linux Server 均具有此功能。

另外，我們已能自動的增加 VM 的 CPU 及記憶體可用的資源，也經由實驗證明，增加資源的確有助於提升虛擬機器的績效。未來，我們希望利用情境模式導入來提高應用

程式的效能，如 Web 或資料庫伺服器的交易處理能力，以證明 ARAS 不僅在學理上是可行的，在實務的應用上更能符合當前 IT 實際環境的需求。

誌謝：本研究依經濟部補助財團法人資訊工業策進會「98年度服務導向機台開放平台軟體發展計畫(1/4)」辦理，計畫編號：98-EC-17-A-02-01-0841。

## 六、參考文獻

- [1] Aravind Menon, Jose Renato Santos, Yoshio Turner, G. (John) Janakiraman, Willy Zwaenepoel, “Diagnosing performance overheads in the xen virtual machine environment”, *Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*.
- [2] Diego Ongaro, Alan L. Cox, Scott Rixner, “Scheduling I/O in virtual machine monitors”, *Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*.
- [3] Evangelia Kalyvianaki, Themistoklis Charalambous, Steven Hand, “Self-adaptive and self-configured CPU resource provisioning for virtualized servers using Kalman filters”, *Proceedings of the 6<sup>th</sup> International Conference on Autonomic computing*.
- [4] Hwanju Kim, Hyeontaek Lim, Jinkyu Jeong, Heeseung Jo, Joonwon Lee, “Task-aware virtual machine scheduling for I/O performance”, *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*.

- [5] Hyung Won Choi, Hukeun Kwak, Andrew Sohn, Kyusik Chung, “Autonomous learning for efficient resource utilization of dynamic VM migration”, *Proceedings of the 22nd annual international conference on Supercomputing*.
- [6] Michael R. Hines, Kartik Gopalan, “MemX: supporting large memory workloads in Xen virtual machines”, *Proceedings of the 3rd international workshop on Virtualization technology in distributed computing*.
- [7] Pradeep Padala, Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant, Kenneth Salem, “Adaptive control of virtualized resources in utility computing environments”, *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*.
- [8] Pradeep Padala, Kai-Yuan Hou, Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant, “Automated control of multiple virtualized resources”, *Proceedings of the fourth ACM european conference on Computer systems*.
- [9] Virtual Machine Manager, A Red Hat Emerging Technology Project,  
<http://virt-manager.et.redhat.com/index.html>
- [10] VMware vCenter and ESX server,  
<http://www.vmware.com/>
- [11] Xen Virtual Machine,  
<http://www.xen.org/support/>