國立交通大學

理學院科技與數位學習學程

碩士論文

COOR: 基於 RBAC 架構的企鵝龍線上系統

Clonezilla Online Of Role-based access control

研究生：蘇仕文

指導教授：蔡文能　教授

中華民國一百年五月

COOR：基於RBAC架構的企鵝龍線上系統
Clonezilla Online Of Role-based access control

研 究 生：蘇仕文　　Student: Shin-Wen Su
指導教授：蔡文能　　Advisor: Wen-Nung Tsai

國 立 交 通 大 學
理學院網路學習學程
碩 士 論 文

A Thesis
Submitted to Degree Program of E-Learning
College of Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Degree Program of E-Learning

May 2011

Hsinchu, Taiwan, Republic of China
中華民國一百年

COOR:基於 RBAC 架構的企鵝龍線上系統

學生：蘇仕文　　指導教授：蔡文能教授

國立交通大學理學院科技與數位學習學程

摘　　要

目前多數電腦維護的工具並無法提供維護大量 linux 電腦環境完整功能，且在校園的環境中，均需由資訊組長或系統維護廠商才能進行處理，一般使用者當電腦出狀況時只能等待處理。本研究採用了屬於自由軟體的 Clonezilla 作為核心，加入 Role-based access control 的概念提升系統安全性，建置一套網頁式系統維護的機制，除了提供資訊組長透網頁進行即時或非即時大量電腦管理外，也提供一般的使用者管理自己所屬的電腦系統。當使用者的電腦系統發生問題時，在權限內立即自行還原自己的電腦，不需要等資訊組長或系統維護廠商來處理電腦狀況，以提升電腦管理的效能。

本研究預期達到下列目標：

1.節省經費支出。

2.簡化校內電腦維護流程。

3.資訊組長能有更多元的方式管理校內所有電腦。

4.有權限的老師可以更自主的使用及管理自己的電腦。

5.資訊組長可減輕負擔。

關鍵字：自由軟體、開放源碼、Clonezilla、RBAC、還原系統

Clonezilla Online Of Role-based access control

student: Shin-Wen Su Advisors: Dr. Wen-Nung Tsai
Degree Program of E-Learning
College of Science
National Chiao Tung University

ABSTRACT

At present, most computer maintenance tools can not provide full function for a large number of Linux's computer environment. In the campus environment, problems can only be solved by the IT leader or system vendor. Normal users can't do anything but wait for their help. In this research, we based on an open source software called Clonezilla, and Role-based access control concept to build a Web-based system called COOR (Clonezilla Online Of Role-based access control). COOR not only allows IT leader to manage a large number of computers, but also general users to manage their own computer system. When a system problem occurs, users can immediately restore their own computer within the permission, instead of waiting for help. This mechanism can enhance the performance of computer management.

The research is expected to achieve the following objectives:

1. Reduce expenditure.
2. Simplify the school computer maintenance process.
3. Give IT leaders various ways to manage all the computers on campus.
4. Allow teachers with permission to use and to manage their own computers.
5. Lighten IT leaders' load.

Keywords: freeware, open source, Clonezilla, RBAC, recovery system

誌　　謝

本論文能順利完成，首先感謝蔡文能教授在研究方向與論文的內容的指導，也感謝林正中教授與曾秋蓉教授提出寶貴的建議與修正，使本論文能夠更加完善，在此特別感謝三位教授的協助。

當然，也要感謝我的家人，因為有了你們的支持與協助，才能讓我可以無後顧之憂的完成我的論文。最後，僅以本論文獻給在推動自由軟體的伙伴，本著自由軟體樂於分享的精神與大家分享，希望能在自由軟體的推動上盡一己綿薄之力。

目　　錄

表目錄

圖目錄

第一章 緒論

隨著病毒及駭客問題越來越嚴重，個人電腦作業系統的穩定度也備受考驗，往往因為個人疏忽未修補系統漏洞，加上安裝有問題的盜版軟體、插入有問題的隨身碟或連上有問題的網站，造成系統效能越來越低、處理動作越來越慢，最後導致整個系統無法開機而延誤個人事務的進行。

有鑑於此，許多軟體商開發了系統備份還原軟體，例如：ghost、true image、clonezilla...等。其中clonezilla[20]是由國家高速網路中心[23]所開發的opensource solution，除了可以透過網路進行備份還原，亦跨平台的處理MS-Windows系列及Unix-like系列的作業系統，且處理速度快速，以8G大小的作業系統為例，透過網路進行備份需8分鐘左右，單機還原只需要5分鐘左右的時間，而36台電腦採unicast 模式，也只需要30分鐘左右的時間，在效能及功能上均有不錯的表現。

但是clonezilla的安裝、設定及使用上，均採用文字介面模式處理，對於一般老師來說文字模式造成使用上很高的門檻，所以對學校單位來說，系統備份還原向來都是落在資訊組長身上。但近年來資訊組長業務越來越龐大，加上作業系統更新及頻繁的漏洞修補，作業系統的維護儼然成為一個沈重的負擔。本研究以open source的clonezilla為基礎，整合雲端運算及RBAC的概念，建置一套方便資訊組長管理及一般老師簡易使用的網頁還原系統，一方面減輕資訊組長的負擔，另一方面也讓一般老師可以輕易的維護自己所用的電腦，形成雙贏的局面。

1.1 研究背景與動機

當環境中存在多種不同的機器的類型時，對於管理大量電腦的系統維護者來說，常會面臨到許多管理上的問題；特別是當採用微軟的作業系統時，會產生更多管理上的問題，因為該系統對於不同的主機板無法採用同一份image檔案來進行維護，所以即使在同一區域環境的電腦，當主機板不同時，就得從頭準備該機器的還原檔，無形中增加了管理人員的負擔，也間接的增加維護的時間成本。

在學校的環境中，通常同一場域的電腦數量最多的就非電腦教室莫屬了，動輒三四十台電腦，管理上當然就不是一件容易的事；加上有時因應研習的需要得更換不同的作業系統環境時，對於管理人員亦是一個沈重的負擔。不過，雖然電腦教室的電腦數多，其使用環境比起行政電腦與教室

內教師使用的電腦比較起來相對的單純很多，電腦教室內電腦維護時考量點只要時間允許即可。但是對於行政電腦及教室電腦的使用者，同樣的方式並不可行，因每個人對於使用電腦的概念不同，有些人習慣將檔案放置在系統槽中，所以當需要同時進行大量電腦維護時，就需要等到所有人都準備就緒、資料備份完成後才能開始進行系統的維護，因此同步處理對此區域的電腦來說，並不見得是最有效率的方式。

再者，教育經費緊縮，雖然市面上販售的備份還原軟體可以解決微軟系統環境的問題，但卻需付出大量的經費來支應，這對於日益拮据的經費來說無疑是一個沈重的負擔，而且就目前市售的軟體，並無法對所有的作業系統平台（如：GNU/linux、MAC）提供完整的功能。

基於以上的問題，本研究基於自由軟體建置了一套酷鵝系統(COOR)，提供管理者一套 Web 介面的備份還原系統，可以簡單、直覺、快速的處理大量電腦系統維護的問題；另外，搭配 RBAC 的規則，提供一般使用者可以有權限自行維護該使用的電腦。

1.2 研究目的與範圍

本研究範圍限定於校內區域網路環境(LAN)，對於區域網路環境中所有電腦使用者提供線上維護電腦的服務。研究目的在於提供一般使用者使用電腦的自由度，讓使用者可以依不同的需求選擇所要使用的映象檔來進行系統的維護，且當系統出現問題時，使用者可以簡單、快速的進行電腦維護，不需等待系統維護人員或維護廠商排定時程進行處理。如此不僅可以大幅的縮短系統維護流程，另一方面也可以大幅的節省系統維護的時間、金錢與人力。

另外，亦提供系統管理人員維護大量電腦的功能，系統管理人員不僅同時間可以處理多個維護工作，也可以進行即時或非即時的維護，讓系統管理人員可以更輕鬆的管理大量電腦。

1.3 論文架構

第一章緒論介紹研究背景與動機、研究目的與範圍。第二章介紹背景知識涵蓋自由軟體、備份還原模式、Clonezilla、角色為基礎的存取控制(RBAC)、雜湊演算法(MD5 與 SHA-1)。第三章介紹相關的研究包含備份還原之研究與存取控制之研究。第四章由系統架構設計、角色運算處理及系

統模組規劃三方面介紹COOR酷鵝系統。第五章系統建置與分析依序介紹資料庫實作、系統實作、實驗建置與實作、系統分析比較進行介紹。第六章介紹結論與未來工作。

第二章 背景知識

本章於第 1 節的部份介紹自由軟體的定義、法律上的協議及世界各國採用自由軟體的狀況，第 2 節討論不同的軟體的備份還原模式，第 3 節介紹國家高速網路中心開發的企鵝龍系統及其運作的模式，第 4 節介紹角色為基礎的存取控制(RBAC)，第 5 節介紹雜湊演算法(MD5、SHA-1)的運算方式。

2.1 自由軟體

自由軟體的定義是由自由軟體基金會（Free Software Foundation）的創辦人Richard M. Stallman所提出的。任何軟體只要符合下列的四項自由，則該軟體就可以稱之為「自由軟體[5]」。

自由 0：任何使用者，不論其目的為何，皆有使用該軟體的自由。
自由 1：任何使用者，皆有研究該軟體並改寫該軟體的自由，以符合自身的需要。（前提為可取得該軟體之原始碼）
自由 2：任何人均有不受限制散佈該軟體的自由。
自由 3：任何人均改良再利用該軟體的自由。（前提為可取得該軟體之原始碼）

自由軟體擁有上述的四項自由，讓所有人可以無限制的使用自由軟體，亦可取得其原始碼，但自由軟體的開發者仍保有其版權。所以雖然自由軟體的取得通常是免費的，但若將自由軟體用於商業用途，只要其方式符合自由軟體的四大自由亦是被認可的。換個角度來看，自由軟體本身雖不收費，但其服務（使用者特別的需求委請他人協助收集、協助改良、協助客制化...等）是可以收取費用的。

而自由軟體在法律上，通常採用的協議為GNU(GNU's Not Unix) GPL(General Public Licence)[22]。GNU GPL協議的精神在於：既然使用者可以無限制的取得原始碼，所以基於公平互惠的原則下，其修改的成果也應該公佈給其他人，讓知識可以因為分享而偉大，進而達到自由、分享、互惠的目標。

基於自由軟體的四大自由及其低成本付出，全世界有意識到自由軟體價值的國家也紛紛做出因應。茲將網路新聞中各國採用自由軟體的政策整理如下：

1.法國國會決定採用 Linux(2006.11)。
2.印度教育最普及的省份 Kerala 開始全面實施 Linux 教學課程(2006.8)
3.西班牙的 Zaragoza 市決定全面改用 Linux 桌面系統(2008.1)。
4.印度喀拉拉學校只使用 Linux(2008.3)。
5.日內瓦學校將全面換用 Linux(2008.4)。
6.三所位於芬蘭 Kauniainen 市的學校轉移到開放原始碼(2008.4)
7.俄羅斯郵政切換到開放原始碼桌面(2008.7)。
8.德國柏林藝術大學轉為 Linux 平台(2009.6)。
9.2009 年底將會有 6 萬名俄羅斯老師學習 Linux(2009.9)。
10.2011 年七月起，日本山形縣政府將大量導入 OpenOffice.org。
11.2011 年起倫敦證交所（LSE）用 Linux 打造新一代交易平臺 。
（以上資料整理自：教育部校園自由軟體應用諮詢中心）

所以從自由軟體四大自由的價值與世界推廣自由軟體的潮流來看，採用自由軟體的系統或是應用程式，無論是成本上的考量或者是軟體使用的未來性均是一個利大於弊的選擇（本研究者任職學校在 94 年開始全面換用 linux 作業系統、自由軟體相關的桌面及應用程式，而且將軟體製成 live-CD 提供全校學生及老師使用，以上的作為不僅得到教育部電算中心『TANET 傑出人員貢獻獎-縮短數位落差類』的獎項，也接受了資策會的採訪[21]）。另外在軟體開發方面，為求軟體更長遠、完整的發展及顧及到公共的利益，開放源碼亦不失為一個好的選擇，因為如此才能避免壟斷及技術失傳的情勢發生。有鑑於此，本研究酷鵝系統(COOR)除了基於自由軟體發展外，當發展完整後，亦以開放源碼的方式進行分享。

2.2 備份還原模式

系統備份還原的模式種類繁多，且近年來機器及網路的速度愈來愈快，所以各類的備份還原機制在速度跟效能上均有長足的進步，大致上可分成：從存取權限管理、利用其他儲存媒體進行壓縮備份還原及暫存區進行存取三大類的範圍中，以下就從此三類的備份還原方式進行探討：

1.從存取權限管理

在 Unix-Like 的作業系統中，由於使用者權限架構清楚，所以可

以輕易的針對重要的系統設定檔設定進行管理，分作下列兩種方式：

(1)將重要系統檔設成唯讀，所以無論使用者在該次的作業中，如何更動系統設定，均無法回存設定。所以當使用者重新開啟作業階段時，就會載入系統的原始設定值，讓該次作業可以正常執行。但這種方式仍有風險，當使用者自行去將設定檔改成可讀寫，此防護措施即失效。

(2)將使用者的重要系統檔的所有者改成 root，在此種方法下，仍保有設成唯讀的功效，而且使用者無法更動，所以在防護上可以收到更好的效果。

以上的方法在 Unix-Like 的作業系統中，均可以得到不錯的成果，但是在 Windows 系列的作業系統中，雖然後期的作業系統慢慢的在仿造 Unix-Like 的權限設定，但是由於設定仍不夠周全，所以利用相同方式管理時，效果並不好。

2.利用其他儲存媒體進行壓縮備份還原

此種方式很直覺，將目前的作業系統的整個磁區複製，當系統有問題時，就將正常的磁區覆蓋掉有問題的磁區，即可達到還原的效果。以儲存的方式來可分成：硬碟對硬碟及硬碟對檔案兩種方式。

(1)硬碟對硬碟就是將設定好的硬碟者個備份到另一顆同樣大小的硬碟中，當系統出現問題時就直接將備份的硬碟覆蓋掉原來的硬碟，以達到還原的效果，但是此種模式兩個硬碟需要大小相同，否則無法正常運作。

(2)硬碟對檔案就是將硬碟裡要備份的磁區複製並壓縮成一個檔案，當系統出現問題時就將該檔案解壓縮後覆蓋到有問題的磁區，以達成還原的效果，此種方式不限硬碟大小，只要磁區規劃大小相等即可運作，所以彈性較大。

另外對於檔案存放的位置可分為：放在同一個硬碟的其他磁區、另一顆硬碟、光碟片、USB 隨身碟或網路上的儲存空間等五種方式。以表 1 進行比較。

表 1：還原檔案存放位置比較表

存放位置	空間大小	傳輸方式	可同時還原數
其他磁區	視硬碟剩下空間	IDE、SATA	單機
其他硬碟	視硬碟空間，目前最大可到 2TB	IDE、SATA、USB	視該主機可接硬碟數
單面 DVD 光碟片	4.7GB	IDE	視該主機可接硬碟數
USB 拇指碟	目前最大 64GB	USB	視該主機可接硬碟數
網路上的儲存空間	無限制	網路	一個 Class C 可到 254 台主機

所以從管理者角度來看，當需要管理大量電腦時，採用硬碟對檔案的儲存方式搭配網路儲存空間能達到最佳的管理效果。

3.以暫存區進行存取

在硬碟中切割出來一個隱藏磁區進行控制，此種方式，通常需要外接硬體來處理。目前市售的還原卡大多採用此種方式。其控制方法主要分為下列兩種：

(1)將隱藏磁區切割與系統磁區同樣大小，並將系統磁區整個備份到隱藏磁區，當系統發生問題時，將隱藏磁區內容整個覆蓋回有問題的系統磁區，以達還原效果。

(2)將該次作業所需儲存的設定先暫時放在隱藏磁區中，關機後即清空隱藏磁區的資料，所以每次開機都會是預設的系統設定環境。

上述三種方法並無絕對的優劣，需依照設備設置的環境與架構進行選擇與配置，方能使整體管理設備的效能達到最大。

2.3 企鵝龍(Clonezilla)

Clonezilla[20]是由國家高速網路中心[23]所開發的備份還原系統，架構在GNU/Linux的作業系統上，搭配open source的相關工具程式(partimage,ntfsclone,partclone,dd...等)，來針對分割區或整個硬碟來進行備份或還原，其主要發行的版本有兩種。

1.Clonezilla Live(單機免安裝版本)

Clonezilla Live版本可以放在隨身碟或光碟片等可開機媒體上，使用該媒體開機後，隨即就會進入Clonezilla的系統中，此時就可以針對分割區或硬碟進行還原或是備份的處理。此種模式，因受限於媒體的硬體限制(IDE插槽數、SATA插槽數、USB插槽數)，所以儘適合用於單機狀況的處理，不適合處理大量的機器。

2.Clonezilla Server Edition(適合大量佈署的伺服器版)

Clonezilla SE(Server Edition)伺服器版本，內建於DRBL(Diskless Remote Boot in Linux)[19]伺服器中，適合用來佈署大量機器的環境(如：電腦教室)。但需要搭配連線正常的網路環境及可網路開機的Client端電腦，雖然環境建置的要求較多，但是在大量機器的環境中，此版本的Clonezilla SE仍為首選。

此伺服器提供DHCP(Dynamic Host Configuration Protocal)[18]、TFTP(Trivial File Transfer Protocol)[17]、NFS(Network File System)[16]、YP/NIS(Network information Services)[15]。當Cliet端電腦從PXE網卡開機，透過DHCP取得IP及TFTP取得開機所需的設定檔設置好網路環境，接著從NFS取得掛載的根目錄，最後利用NIS完成登錄動作。登入後使用Client端機器的CPU與記憶體來執行程式。以下就Clonezilla SE所提供的儲存模式及還原模式進行介紹。

(1)儲存模式：

Clonezilla SE在Client端使用者設定檔與資料的處理方式

有兩種，如下列的表格：

表 2：Clonezilla Full 與 Clonezilla Box 的比較表

	Full Clonezilla	Clonezilla Box
Client 端的/etc、/var 目錄	NFS-based	tmpfs-based
Client 端的/etc、/var 目錄下修改過的檔案重開機後	保留	消失
增加一個 Client 所需空間	~50 MB	0 MB

資料來源：國家高速網路與計算中心

Clonezilla Box 模式中，是利用 client 端電腦的記憶體來降低 Server 的負載及網路頻寬，此種模式非常適合用在大量電腦的佈署環境中，一般只要 client 端電腦記憶體有到 512MB，使用此種模式是沒有問題的。而 Clonezilla Full 的模式，相對來說就需要較高等級的伺服器來作為因應，但好處在於可以針對較舊電腦(記憶體低於 512MB)的環境提供服務。在 Clonezilla Se 中，切換兩種模式的畫面如下：(選擇 0 使用完全再生龍模式即為 Clonezilla Full，選擇 1 使用再生龍盒模式即為 Clonezilla Box)

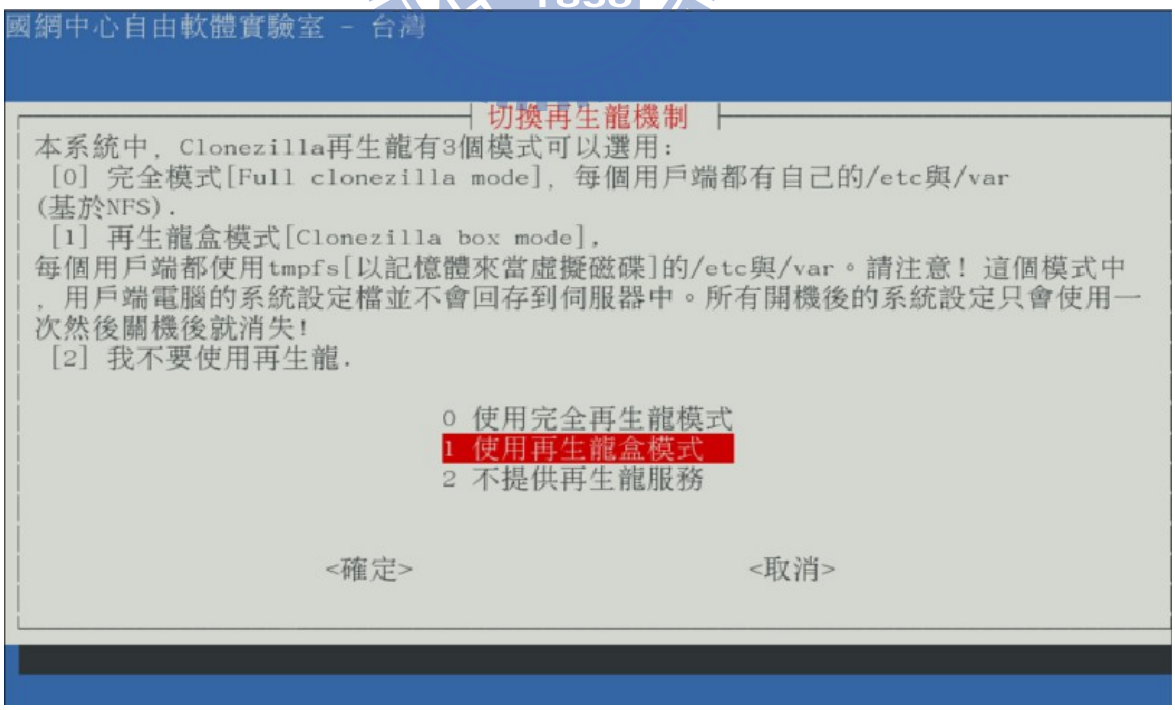

圖 1：Clonezilla SE 模式切換

兩種模式的比較：Full Clonezilla 模式中每個使用者都可以保留其設定及資料，但伺服器就需負擔較多的系統資源及空間。Clonezilla Box 模式則剛好相反，系統不需負擔額外的空間及資源，但是缺點就是所有使用者的設定與資料無法保留，兩者無所謂好壞，端看系統環境建置者依其所規劃的環境自行選用。

(2)還原模式：

對於還原的方式，Clonezilla Se 提供下列三種模式：

- unicast：利用 NFS 取得 image 檔案中的資料。此種模式 server 的 CPU 及網路的負載會隨著 Client 端機器數量增加而上升。
- multicast：利用 NFS 取得 image 檔案中的資料，透過 udpcast[14]來接收分割區的印象檔。此種模式的封包是透過網路交換器來複製的，且只會把封包丟給有加入 multicast 模式的機器，所以 server 的 CPU 及網路負載不會隨著 Client 端機器數量增加而上升，。
- broadcast：利用 NFS 取得 image 檔案中的資料，透過 udpcast 來接收分割區的印象檔，另外在 udpcast 多加一個參數'—broadcast'。此種模式與 muticast 的差別在於，不管 client 端電腦是否有加入 multicast 模式，封包都會繼續傳遞，server 的 CPU 及網路負載也不會隨著 Client 端機器數量增加而上升。

此三種模式，可以使用者可以就自己的伺服器等級及網路環境進行選擇做適合該環境的模式。

國家高速網路中心所開發維護的 Clonezilla 不管在 live 模式或是 Server 模式，在實務上已經可以有不錯的效能，但是在使用上仍需使用文字介面進行控制，且需要較高的資訊能力才能掌控整個軟體。而本研究就基於 Clonezilla 目前的核心，針對易用性及實用性進行改良。

2.4 以角色為基礎的存取控制(RBAC)

傳統的存取控制僅在使用者與權限中進行規範，針對不同的使用者給予不同的權限進行存取，觀念上很直覺，但是在組織裡應用時，會產生問

題，當某一位使用者職務進行調整時（如：部門間調動或職位的升遷），就需將該使用者的所有權限進行重新的設定，如此不但費力耗時，而且當不小心某一權限設定錯誤時（如：把較高權限給予低職位的使用者），可能會造成內部機密外洩，或者是資料被不當存取的危險，於是以角色為基礎的存取控制(Role-Based Access Control[2])的觀念就在 1996 年被 Dr. Sanhu 提出且開始應用。

以角色為基礎的存取控制在原本的使用者與權限中加入了角色的層面，每個使用者被賦予不同的角色，每個使用者允許被同時賦予不同的角色（如：公司裡的職員可以同時被賦予文書作業員及專案管理人員的角色）加上每個角色所可執行的權限是被設定好的，所以當有新進人員或人員調動的時候，就只需要針對該使用者進行角色的設定即可，不需進行細部的權限設定，如此就可讓整個權限管理的工作更為簡單、富有彈性及貼近組織運作的需求。

在安全性的考量上，採用 RBAC 時，需要注意到三個基本的安全原則：

1.使用最少的權限(Last Privilege)

因為使用者角色時，我們允許同一位使用者被同時賦予不同的角色，所以當我們在設定角色可以使用的權限時，就需要進可能讓單一角色的權限是最少的，這樣才可以避免權限遭到不當的使用甚至發生衝突。

2.責任需要被區隔(Separation)

在眾多不同角色中，有些角色是有互斥性的，同一位使用者不可以同時被賦予具有互斥性的角色(如：會計與出納不可以同時由同一個人擔任，以避免發生監守自盜的情事)。

3.資料的抽象化(Data Abstraction)

在傳統的存取控制只定義了基本的操作，如：讀取、寫入及執行等。但在 RBAC 中允許以較為抽象的方式來定義權限，如：申請專案、專案分配。

在 RBAC 發展的歷史中，原本只有定義核心 RBAC(Core RBAC)，其中包含了三個主要的元件，使用者(Users)、角色(Roles)及權限

(Permissions)，其中權限又細分為操作(Operations)及物件(Objects)。

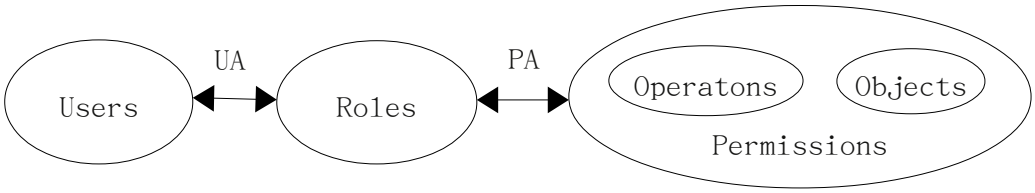

圖 2：Core RBAC 架構圖

接著在發展的過程中，Core RBAC 只加入了層級(hierarchies)讓角色可以相互繼承形成 RBAC1。Core RBAC 只加入的限制(constrains)作用在於限制角色所需條件及將適當的權限設定給適當的角色，以避免不當的指派權限而形成 RBAC2。若將 Core RBAC、RBAC1 及 RBAC2 整合在一起，就形成了 RBAC3。

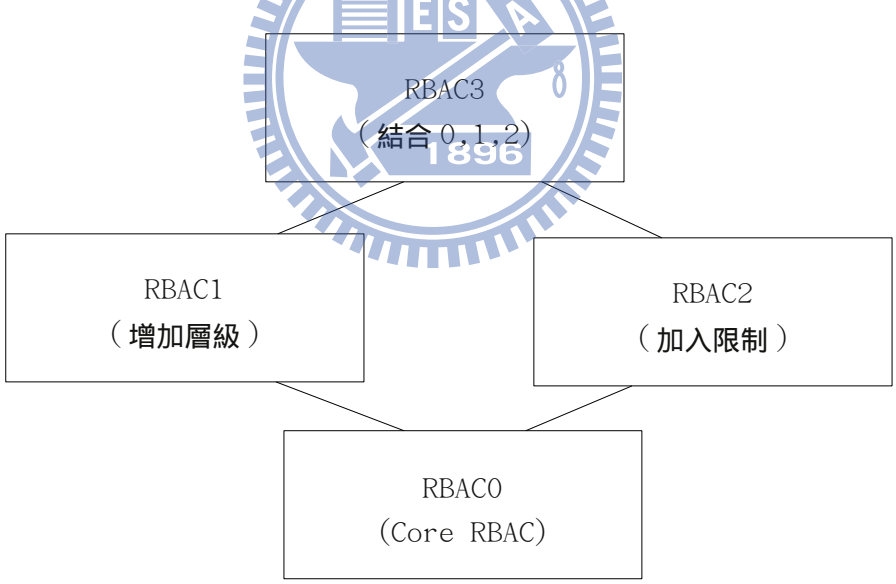

圖 3：RBAC 關係圖

RBAC 發展至 2001 年時，由 Farraiolo 與 Sandu 修改並制定了美國國家技術標準局(National Institute of Standards and Technology, NIST)[13]的 RBAC 標準，將 RBAC 的基本元件分成七個，並將元件間的關係做更進一步的定義，而成為目前較為完整的架構。

表 3：NIST 版本的 RBAC 基本元件

元件	功能
使用者(Users)	使用系統的人員。
角色(Roles)	RBAC 中被定義的角色，將設定給使用者。
權限(Permissions)	對於操作及物件所設定的權限。
操作(Operations)	對於物件可以操作功能，包括讀取、寫入，也可以是抽象的操作功能，如申請專案、專案分配。
物件(Objects)	可以存取的物件，如：檔案、文件。
會期(Sessions)	使用者對應到角色的過程。
限制(Constrains)	用來限制角色間的關係，分為靜態責任區隔關係(Static Separatoin of Duty Relations, SSD)及動態責任區隔關係(Dynamic Separation of Duty Relations, DSD)，前者屬強互斥關係不可指派給同一使用者，後者屬弱互斥可指派給同一使用者。

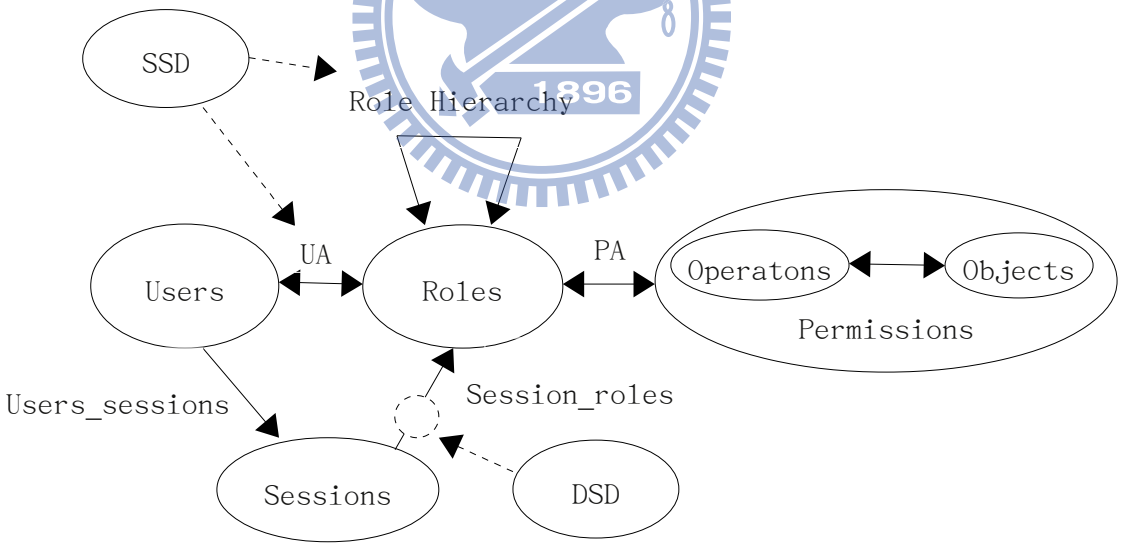

圖 4：NIST 版本的 RBAC 架構圖

本研究中將加入 Role-Based Access Control 的原則進行考量及設計，依 RBAC 的設計原則讓系統中的角色可以融入 RBAC 的優點兼顧彈性及安全性的應用。

2.5 雜湊演算法(MD5、SHA-1)

在系統使用者密碼處理的部份，一般而言都會進行加密或是透過金鑰的模式來處理，以提高系統的安全度。在本研究中，因為系統環境限制在學校的區域網路中，且對外有架設防火牆，所以密碼的處理就選用有一定的安全度及普遍性的 MD5 及 SHA-1 作為考量，以下就針對 MD5 與 SHA-1 的運算方式進行探討(本節數值來源：[6])。

1.MD5

MD5 訊息摘要(message digest)演算法是 Ron Rivest 在 MIT 所發展出來的，其演算的步驟如下：

(1)在訊息後加上附加位元(padding bits)

此步驟目的在於讓整個訊息的長度（以位元為單位）在經過 mod 512 後等於 448。也就是說經過附加位元後，會使整個訊息的長度成為 512 位元的倍數少 64 個位元(512-448=64)，而且就算原本的訊息剛好是 448 位元，我們仍需為此訊息加上附加位元（此時附加的位元會剛好是 512 位元）。附加位元的方式是先加一個 1，之候補 0 到我們所要求的長度（附加位元的長度可以從 1 到 51）。

(2)補足長度

我們將原始訊息的長度，以 64 位元的資料來表示，若原始訊息的長度超過 64 位元，我們就只取訊息長度的最低 64 位元，再將這 64 位元附加在步驟一產生出來訊息的後端，也就是說將整個訊息的長度補足成為 512 位元的倍數。

(3)設定 MD 暫存區(MD buffer)的初始值

我們使用暫存區的長度為 128 位元，用來儲存運算中的暫存數值及最後的結果。以 4 個 32 位元的暫存區(A、B、C、D)來表示，其初始值如下：（以 16 進位來表示）

A：0x01234567

B: 0x89ABCDEF

C: 0xFEDCBA98

D: 0x76543210

(4)處理512位元的區段

此演算法是由四個處理回合所組成，每一個回合都有16個步驟且都有其不同的邏輯函數，表示成F、G、H、I，其各自的邏輯函數如下：

F(B,C,D)=(B AND C)OR((NOT B) and D)

G(B,C,D)=(B AND C)OR(C and (NOT D))

H(B,C,D)= B XOR C XOR D

I(B,C,D)= C XOR (B OR (NOT D))

運算的過程可以歸納如下：

CV0 = IV

Cvq+1 = SUM32

(CVq,RFI[Yq,RFH[Yq,RFG[Yq,RFF[Yq,CVq]]]])

MD = CVL

其中：IV = A、B、C、D暫存區的初使值

Yq = 訊息裡第q個512位元區段

L = 訊息裡512位元區段的個數

CVq = 第q個區段處理後的暫存值

RFF、RFG、RFH 、RFI = 四個邏輯運算函數

MD = 最後的結果

SUM32 = 將輸入的兩個區段字元分別相加後 mod 32的結果。

(5)得到結果

將四個暫存區A、B、C、D經過步驟四的運算後，接在一起回傳。我們就能得到一個128位元的結果。

2.SHA-1

SHA-1 安全雜湊演算法(secure hash algorithm)是由國家標準與技術協會(NIST)所發展出來的，其原理如下：

(1)、(2)與MD5演算法相同。

(3)設定 MD 暫存區(MD buffer)的初始值

我們使用暫存區的長度為 160 位元，用來儲存運算中的暫存數值及最後的結果。以 5 個 32 位元的暫存區(A、B、C、D、E)來表示，其初始值如下：（以 16 進位來表示）

A：0x67452301

B: 0xEFCDAB89

C: 0x98BADCFE

D: 0x10325476

E：0xC3D2E1F0

(4)處理 512 位元的區段

此演算法是由四個處理回合所組成，每一個回合有 20 個步驟，每一回合都有其不同的邏輯函數及其加入運算的 Kt，表示成 f1、f2、f3、f4，其各自的邏輯函數如下：

表 4：SHA-1 函數表

步驟數	函數名稱及值	常數 Kt
0<=t<=19	f1=f(t,B,C,D)=(B AND C)OR((NOT B) and D)	5A827999
20<=t<=39	f2=f(t,B,C,D)= B XOR C XOR D	6ED9EBA1
40<=t<=59	f3=f(t,B,C,D)=(B AND C)OR(B AND D)OR(C AND D)	8F1BBCDC
60<=t<=79	f4=f(t,B,C,D)= B XOR C XOR D	CA62C1D6

運算的過程可以歸納如下：

CV0 = IV， Cvq+1 = SUM32 (CVq,ABCDEq)

MD = CVL

其中： IV = A、B、C、D 暫存區的初使值

ABCDEq = 訊息裡第 q 個區段最後一回合的輸出

L = 訊息裡 512 位元區段的個數

MD = 最後的結果

SUM32 = 將輸入的兩個區段字元分別相加後 mod 32 的結果

(5)得到結果

將四個暫存區 A、B、C、D 經過步驟四的運算後，接在一起

回傳。我們就能得到一個 160 位元的結果。

比較 MD5 與 SHA-1 兩種常用的雜湊演算法。在安全上考量，基本上兩種演算法都有一定的安全度，但就長度來考量，MD5 長度為 128 位元，SHA-1 的長度為 160 位元，在長度上 SHA-1 就比 MD5 多出 32 位元，因此 SHA-1 安全度就高於 MD5。但對於運算速度來說，MD5 執行過一次需要運算 64 個步驟，SHA-1 執行過一次卻需要運算 80 步驟，且 SHA-1 的長度比 MD5 多出 32 位元需要運算，所以執行速度上 MD5 的速度會比 SHA-1 來得快。

在本研究中，由於環境限制在學校的區域網路內，所以在選擇上會偏向選擇執行速度較快的演算法；加上校內已經建置的一些線上系統，其採用的編碼方式即為 MD5，為了讓老師們直接使用原來的帳號密碼來使用本系統，本研究中使用者密碼的編碼方式就採用了 MD5 的演算法。日後若有更高的安全性需求，亦可將編碼方式改用安全性較高的 SHA-1 編碼方式。

第三章 相關研究

本章於第 1 節討論備份還原的研究，第 2 節討論存取控制的研究。

3.1 備份還原之研究

備份還原的模式已於第二章探討。本節將針對系統備份還原相關的研究進行探討，茲分為還原卡、排程還原及無碟系統三部份：

1.還原卡

廖瑞民[9]在 2002 年設計一套完全獨立的介面裝置，其裝置要解決的是舊式的還原卡的問題。

舊式還原卡主要架構於 Windows 的 Device，所以在進行資料 I/O 處理時，是藉由處理每一個 Windows 上層的 I/O 要求，來達到系統保護的效果。此模式在普及率提高時，可能出現新的電腦病毒系統能對硬碟做 Direct I/O acces，此時整個復元的系統就可能會輕易的被瓦解。

而廖瑞民設計的完全獨立介面裝置，將該裝置串接在硬碟與主機板的中間，針對硬碟 I/O 的最底層進行攔截轉向，進而達到系統備份還原的功能。其還原卡系統架構圖如圖 5 所示、還原卡系統工作流程圖如圖 6 所示、還原卡裝置俯視圖如圖 7 所示。

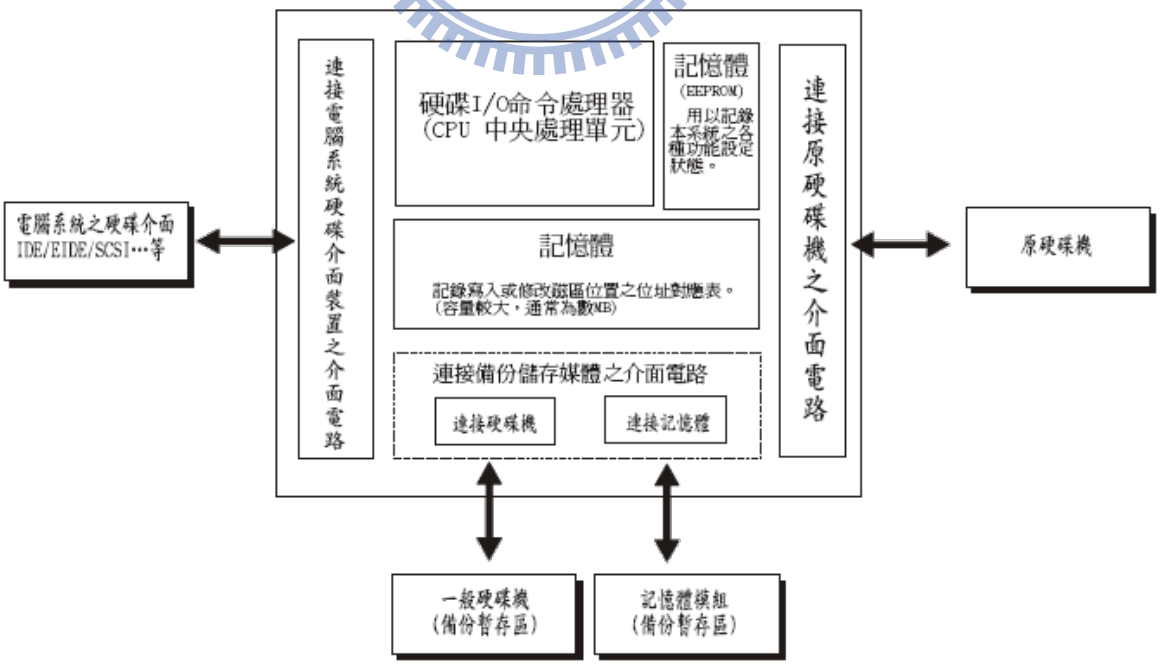

圖 5：還原卡系統內部架構圖

資料來源：[9]

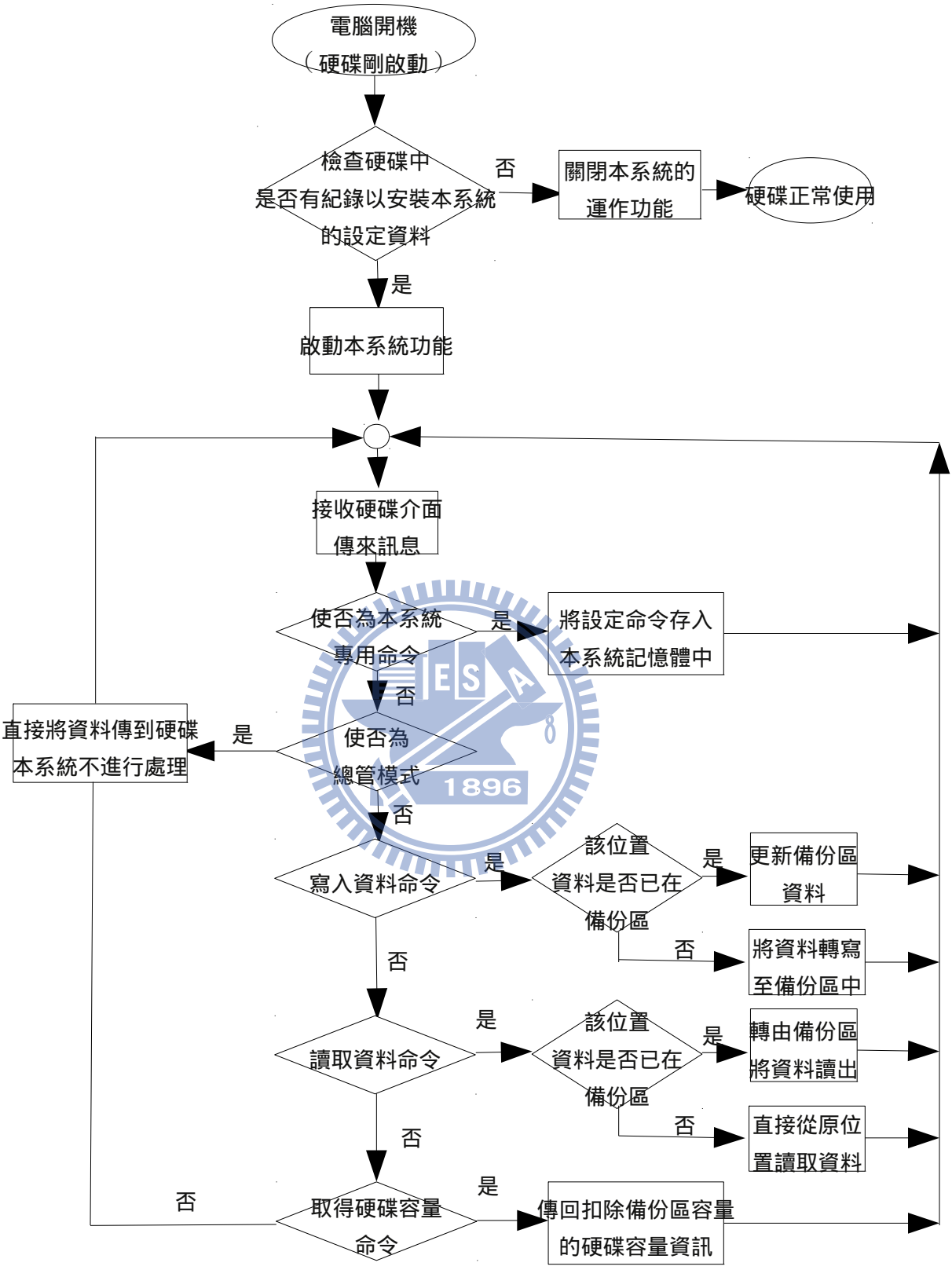

圖 7：還原卡系統工作流程圖

資料來源：[9]

圖 8：還原卡系統裝置俯視圖
資料來源：[9]

該研究提出的解決方法，在 Windows 98 了系統下成功的解決舊式還原卡的問題，但是幾個問題需要被處理：

(1)當要大量複製時，會面臨到複製的方法及複製的效能的問題，這問題對於系統管理人來說是一項重要考量的指標，但該研究中並未提及此問題的處理方式及實驗的數據。

(2)該研究的實驗環境採用的是 Windows98 的系列產品，在當時作業系統加上常用的應用程式整體的空間鮮少超過 1GB 的大小，所以該研究只需要 500MB 大小的備份空間就綽綽有餘。但是對於目前系統安裝隨便就要數 GB 的空間的情況來說，該研究可能在建置上就會遭遇到問題，另外使用的效能上，也得重新評估。

(3)Windows NT；Windows 2000；Windows Xp 的版本發行之後，其對硬碟的存取改為直接的 I/O，此種狀況要使用廖瑞民所設計的裝置，就又需在系統中多設計一個 Driver，如此就與作業系統產生了相依性，無法成為獨立的系統。

(4)該裝置所費不貲，以一間電腦教室 36 台電腦來說，可能需要將近十萬的經費，對小學的經費來說無疑是一沈重的負擔。

2.排程還原

王德源[8]在2002年提出一套網路排程還原機制 。其研究架構的環境是由Web Server + MTS(Microsoft Transaction Server) + Database Server +自行設計的Schedule Manager以及多台File Server建置而成的復原環境，其排程系統完整示意圖如下：

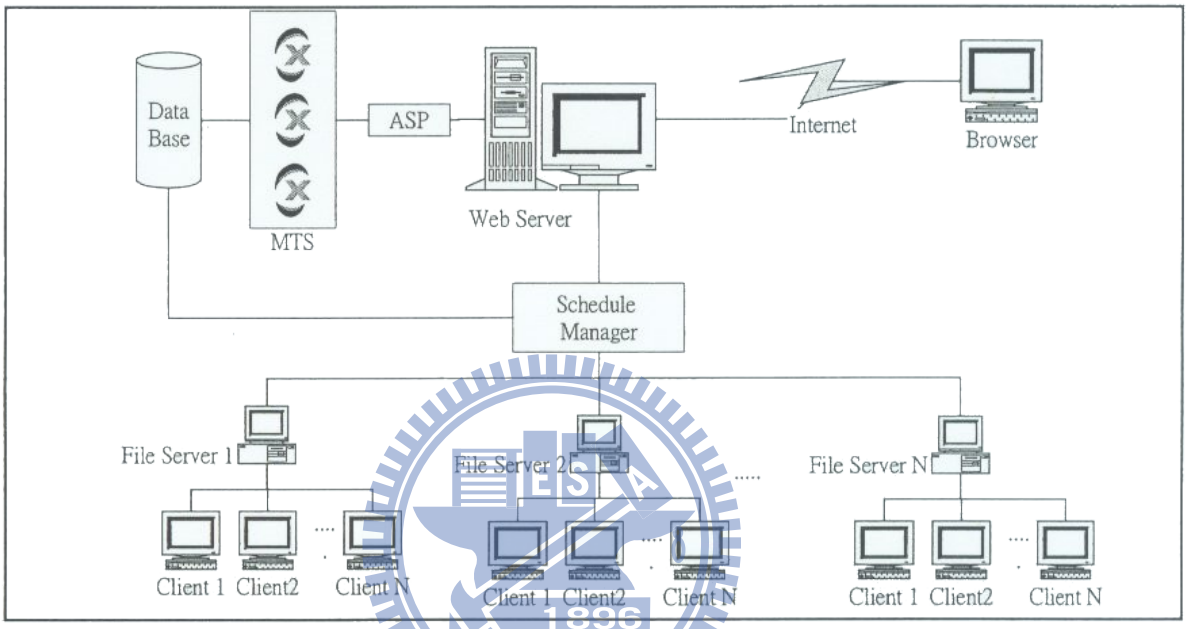

圖9：排程系統架構圖
資料來源：[8]

系統架構上大致可分為三部份：「排程表單設定」、「排程程序掃描」、「用戶端系統重建」。當管理者要進行系統重建時，就從瀏覽器登入系統設定好排程表單，之後就由排程程序接手執行排程表單從遠端進行喚醒用戶端電腦後，透過用戶端腦裡的Ghost與File Server的連繫進行系統重建。

排程系統之硬體建置：

Web Server一台規格如表 5所示、File Server數台規格如表6所示、Client電腦規格如表7所示。

表 5：排程系統 Web Servere 規格表

CPU	Pentium III 600
RAM	256MB
BIOS	Phoenix Bios 4.0 Release6.0
硬碟	20GB
作業系統	Windows NT 4.0
應用軟體	Delphi 5.X
資料庫	MS Access 2000

資料來源：[8]

表 6：排程系統 File Server 規格表

CPU	Pentium III 600
RAM	256MB
BIOS	Phoenix Bios 4.0 Release6.0
硬碟	20GB
作業系統	Windows NT 4.0
應用軟體	Delphi 5.X
資料庫	MS Access 2000

資料來源：[8]

表 7：排程系統 Client 電腦規格表

CPU	Pentium III 600
RAM	256MB
BIOS	Phoenix Bios 4.0 Release6.0
硬碟	20GB
作業系統	Windows 9X
應用軟體	Ghost 6.5

資料來源：[8]

其設計的實驗採用 Ghost 還原軟體，利用兩種方法（方法 A 採用網路硬碟提供 IMAGE 檔案，方法 B 採用該實驗所設計的還原系統），使用 500MB 大小的 IMAGE 檔案，對於 1~32 台電腦進行還原並紀錄其還原時間，實驗數據如下：

表 8：還原所需時間表（秒）

電腦數	1 台	2 台	4 台	7 台	14 台	20 台	32 台
方法 A	158	170	220	381	821	997	1131
方法 B	223	220	219	221	225	223	218

資料來源：[8]

從實驗數據中可以得知該系統比原來使用網路硬碟節省時間，且便於管理。但該系統的研究限制如下：

(1)雖然採用 multicasting 來節省時間，但卻無法同時進行不同的排程。
(2)系統所需的軟體均需授權，所以需要一筆所費不貲的經費支應。
(3)採用的 Ghost 還原軟體，僅對於微軟系統支援完整，對於其他的 作業系統（如：linux）支援度並不完整。
(4)系統操作環境限制在電腦教室中。
(5)整個系統的操作仍需管理者負責管理，無法讓一般使用者藉由所屬的權限進行。

以上的研究所遭遇到的問題，所思考到的解決方式如下，並應用到本論文的研究中：

(1)採用 unicasting 所以可以同時進行不同的排程。
(2)採用自由軟體，無需授權費並且可以自由散佈。
(3)採用的 clonezilla 對於大部分的檔案格式均支援，尤其對於 open 的檔案格式支援度更高。
(4)設計針對校內不同的網段提供支援。
(5)設計讓不同的使用者依其權限登入操作。

3.無碟系統

王光山[7]規劃的電腦教室環境使採用LTSP(Linux Terminal Server Project)[12]無碟系統伺服器搭配老舊電腦當作Client建置電腦教室環境，並對於LTSP設計一套軟體還原(ROYSWNG[7])，以維護該台Server正常運作。其運作原理是Client端從軟碟或是從網路開機，連上LTSP Server提出開機要求，LTSP Server從其設定中判斷是否是設定中的電腦，若是則從LTSP Server的系統資源切出一部份供Client端使用，而Client端就利用伺服器中的資源將Client端使用者的操作運算後，由Client端的螢幕提供給使用者。其所規劃的系統架構圖如圖10所示、ROYSWNG復原概念圖如圖11所示。

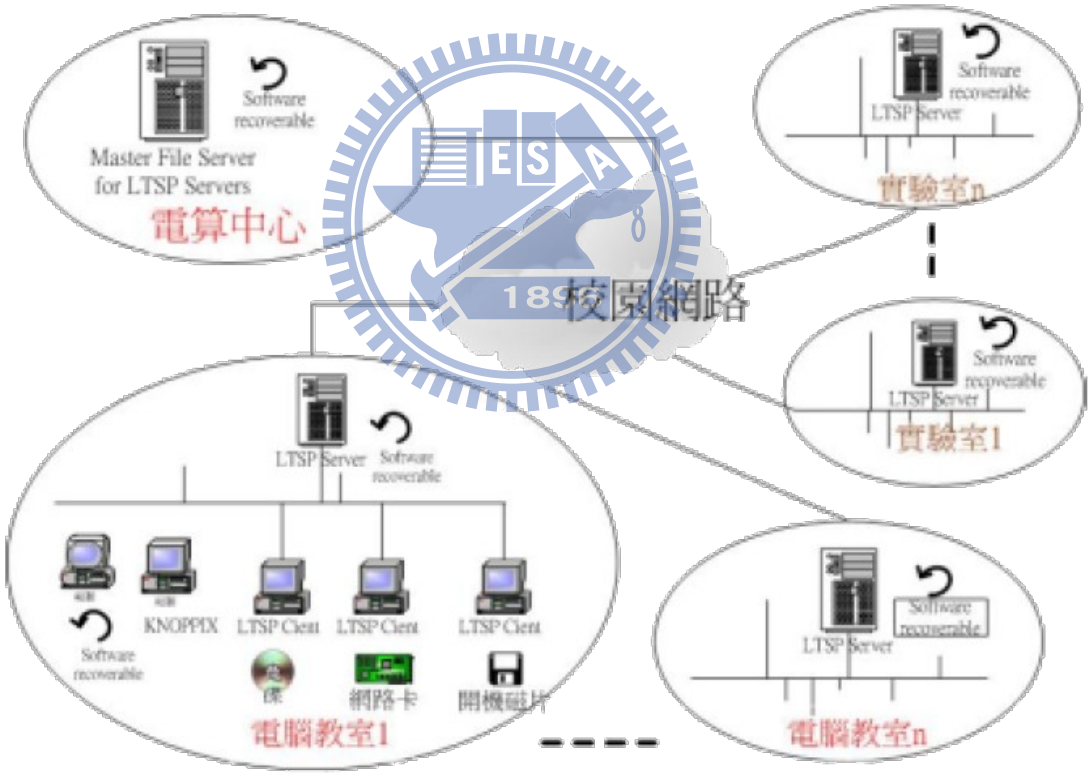

圖10：無碟系統規劃圖
資料來源：[7]

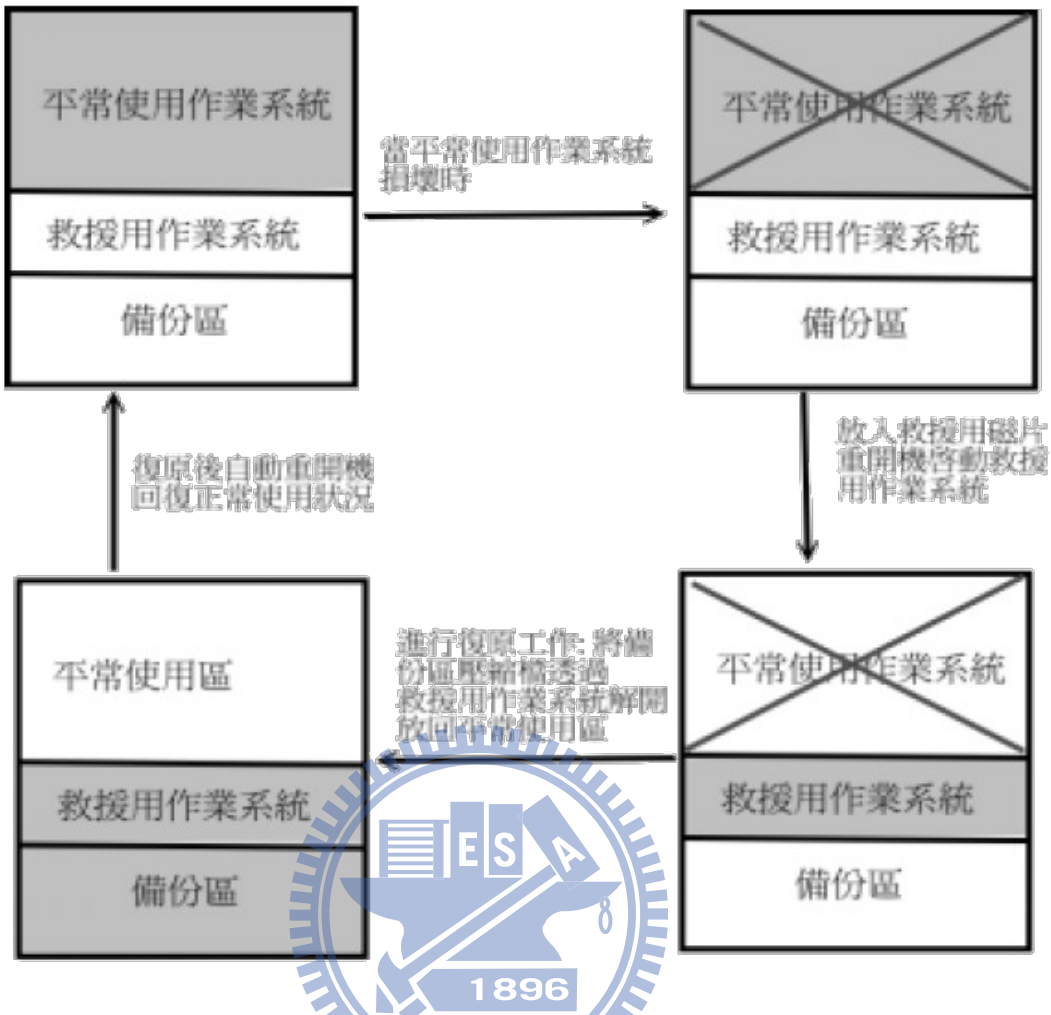

圖 11：無碟系統規劃圖

資料來源：[7]

依上述規劃的電腦教室，Client 端並不需要維護，所以維護的重點就放在 LTSP Server 上，該研究採用軟體復原(GNU/Linux 版時光回溯器：RecOverY—SoftWareNG，ROYSWNG[7])來對 LSTP Server 進行復原。ROYSWNG 原理是將硬碟切割成三部份：第一部份是平常的作業系統，第二部份是救援的系統，第三部份是擺放完整系統備份的壓縮檔。當系統產生問題時，管理者使用 floppy、CD、Hard Disk 或 Network 開機啟動第二部份的救援系統，救援系統啟動後會將第三部份的備份檔解壓縮放回第一部份覆蓋掉出現問題的系統，藉此可以將第一部份的系統還原成預設的狀況，重開機後系統就可以恢復正常運作。

王光山的規劃是為了要解決偏遠地區電腦教室管理的問題：以 LTSP 解決電腦硬體設備等級不夠的問題；以 GNU/Linux 取代 Windows

來解決軟體版權的問題；以 GNU/Linux 版時光回溯器(ROYSWNG)來解決電腦教室維護的問題。

但在近幾年，由於電腦教室電腦不堪使用，各縣市府也逐年規劃更新，因此電腦教室的電腦等級均已逐步提升，所以以目前的電腦硬體狀況，若使用 LTSP 來建置電腦教室的環境，就覺得稍嫌浪費了 Client 端電腦的硬體效能；但若 Client 端採單機運作，輔以 GNU/Linux 版時光回溯器(ROYSWNG)來復原 Client 端的電腦，可能會因為學生不當的操作而讓復原頻繁，一來浪費時間，二來可能造成硬碟壽命減短且系統也可能因此造成不穩定的狀況。

3.2 存取控制之研究

林孟勳[10]中，提出依套 single sign on 的機制，在原本 intranet 中所有的系統前，架構一套基於 RBAC 的登入系統。該系統先將所有的伺服器及所有的角色編碼，所以每一個人對於每一伺服器都可以得到一組編碼後的座標，再將所有座標代入義大利數學家 Joseph Louis Lagrange 提出的 Lagrange 內差多項式[11]的公式，就可以得到該使用者的 Lagrange 多項式。Lagrange 內差多項式的數學表達式如下：

$$f(x)=\sum_{i=1}^{n} y_i \prod_{j=1;j\neq i}^{n} \cdot \frac{x-x_j}{x_i-x_j}$$

$$= y_1 \cdot \frac{(x-x_2)(x-x_3)\ldots(x-x_n)}{(x_1-x_2)(x_1-x_3)\ldots(x_1-x_n)}$$

$$+ y_2 \cdot \frac{(x-x_1)(x-x_3)\ldots(x-x_n)}{(x_2-x_1)(x_2-x_3)\ldots(x_2-x_n)} + \ldots + y_n \cdot \frac{(x-x_1)(x-x_2)\ldots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\ldots(x_n-x_{n-1})}$$

$$= a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\ldots+a_1x^1+b$$

所以利用此公式所得到的使用者 Lagrange 多項式，並將此公式放入 RBAC 伺服器中。當我們想要得知某使用者在 A 伺服器所擁有的角色代碼時，只需要先找出該使用者的 Lagrange 多項式，代入該伺服器的代碼，即可得到該使用者的角色值，進而判斷該使用者所擁有的權限，其系統流程圖如圖 12 所示。

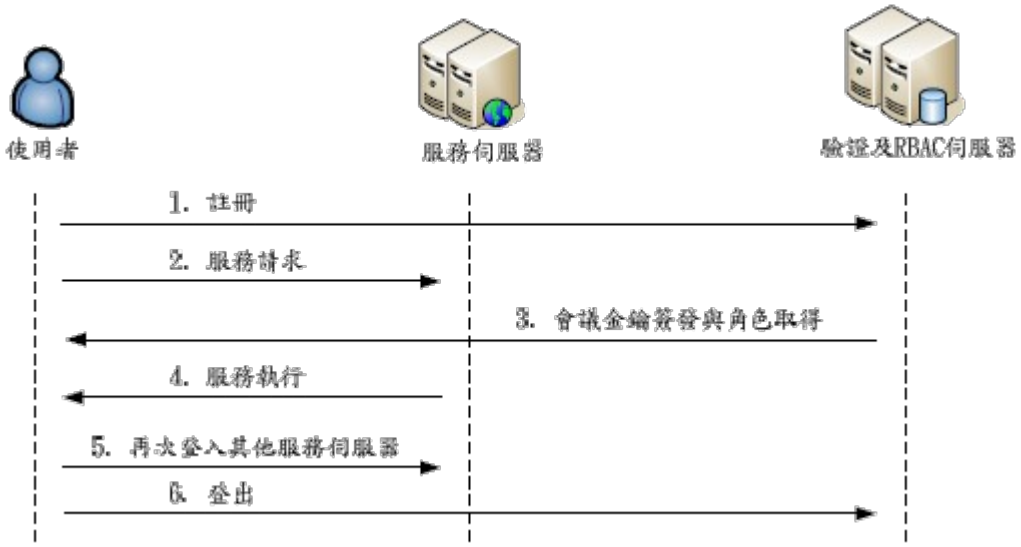

圖 12：存取控制系統流程圖
資料來源：[10]

另外由於資料庫中並無紀錄使用者在各伺服器中的角色，所以當人事主任調整職務的時候就需要規劃運算流程來處理，如圖 13 所示。

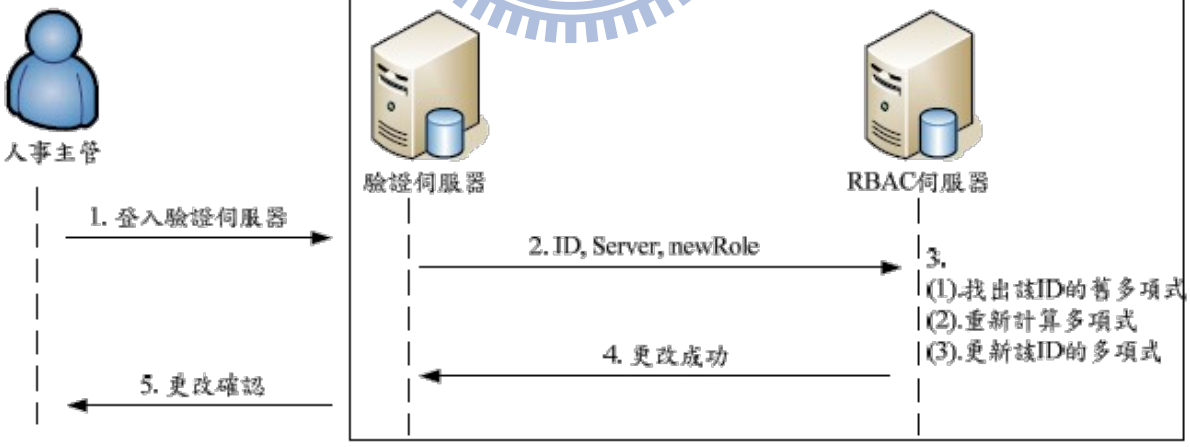

圖 13：存取控制系統流程圖
資料來源：[10]

使用此種方法，主要是安全性的考量，不在資料庫中直接紀錄使用者在所有伺服器的身份，伺服器與角色的代碼表也不對外公佈，所以使用者自己也不知道角色與角色代碼的對應，所以也無從計算出代表身份的多項式，所以角色值可以得到較好的保護。

基於以上的安全性考量，本研究中亦將引用 Lagrange 多項式，但因本研究只有唯一的伺服器，所以將要帶入的伺服器值及角色值，改用角色值及允許值來作為替換，系統管理者可以自行設定角色值及允許值來導出每個使用的 Lagrange 多項式，以提高系統的安全度。

第四章 COOR酷鵝系統

本章於第1節的部份介紹COOR酷鵝系統架構，第2節討論COOR酷鵝系統角色運算處理，第3節介紹COOR酷鵝系統運作流程，第4節介紹介紹COOR酷鵝系統模組的規劃。

4.1 系統架構設計

系統架構是在現有的Clonezilla server的系統架構上，建構一層網頁式的管理系統，透過RBAC的原則，將角色分為：系統管理者、進階使用者、一般使用者三類。功能分為：進行還原(基本功能)、角色管理、使用者管理、權限管理、電腦管理、備份管理、還原管理、image管理、刪除管理等九種模組。

功能繼承部份：進階使用者可以繼承一般使用者的功能。

功能分配部份：

一般使用者：進行還原。

進階使用者：進行還原、備份管理。

系統管理者：所有管理模組。

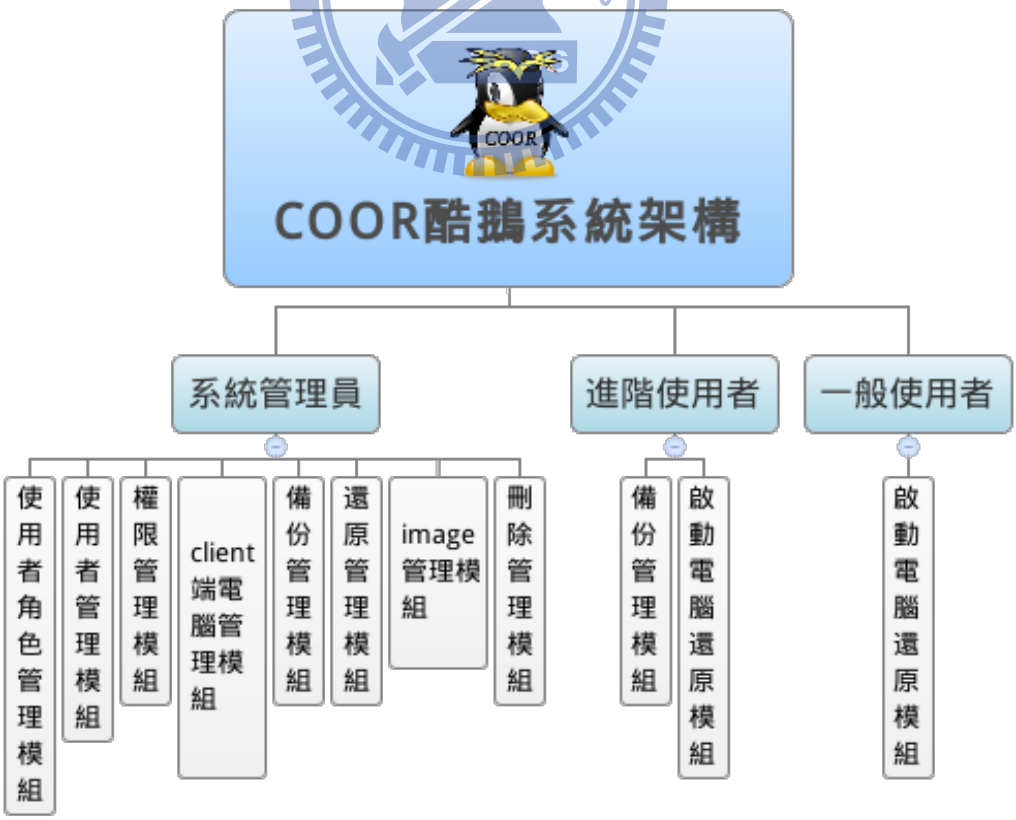

圖14：COOR酷鵝系統架構

為了進行角色的運作，特別設計 rbac 的 table，且 user 的 table 中也加入了四個 role 欄位(userRoleA、userRoleB、userRoleC、userRoleD)來搭配使用，其角色的運算方式，將於下一節進行探討。其他的 table 的部份：role 的 table 處理角色所能使用的模組；user 的 table 除了處理帳號密碼外還可處理可儲存的備份檔個數；computer 的 talbe 處理電腦所使群組、使用者及該電腦所用的備份檔；cgroup 的 table 處理電腦所屬的不同群組、則因應系統架構進行設計；image 的 table 處理備份檔相關的資訊；manage 的 table 處理模組相關資訊。資料庫的規劃如圖 15：

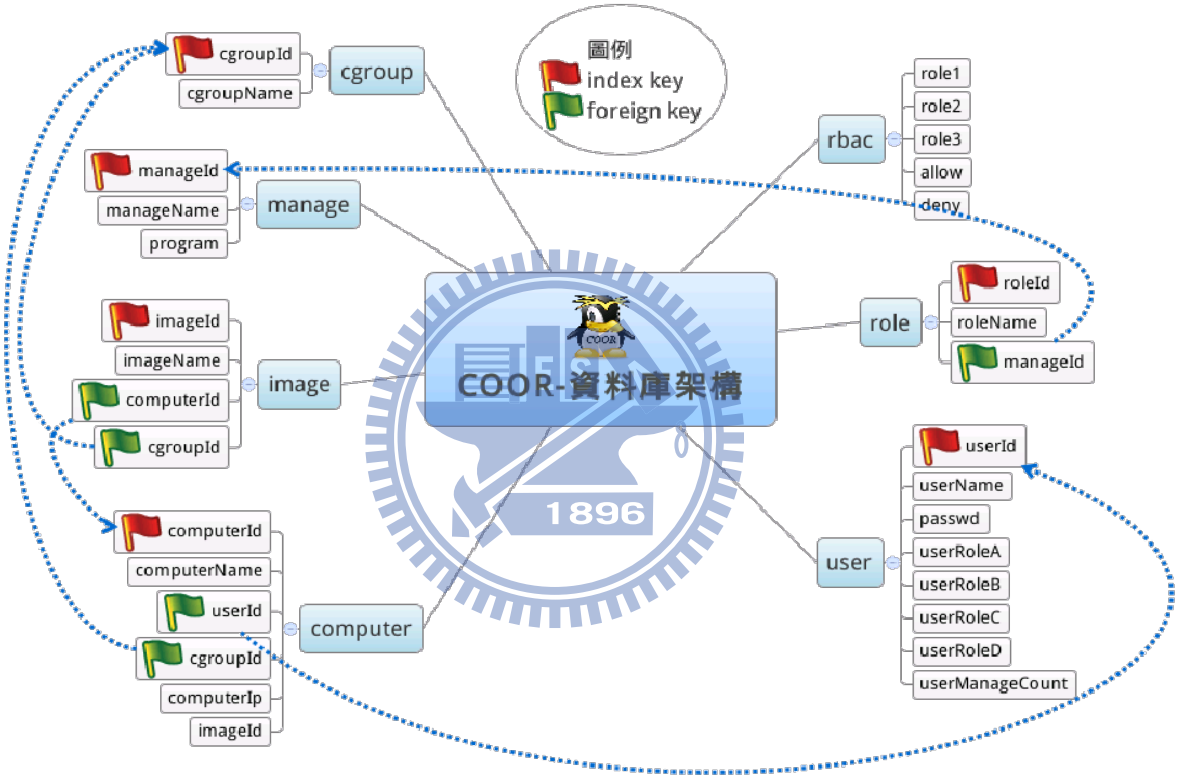

圖 15：COOR 酷鵝系統資料庫設計

4.2 角色運算處理

在本系統中，為提高系統安全性，所以在使用者 user 的 table 中並沒有設置紀錄身份的欄位，是依據 Lagrange 多項式[11]來進行運算進而得到身份的認定。lagrange 多項式在代數上，是藉由已知的點反推得到 f(x)多項式，其幾何上的意義是多點可以決定一條線，而經過線的點有無窮多個，所以用此方法來保護使用者角色，可提高系統的安全度。以下就

使用者角色所代表的方程式處理及如何判定使用者角色進行探討。

1.計算使用者角色所代表的方程式：

本研究利用三個角色值(放置在 x 座標)、一個允許值及一個不允許值(放置在 y 座標)進行搭配，得到該使用者所屬角色的 3 點座標，假設為(a1,b1)、(a2,b2)、(a3,b3)，將此 3 點座標帶入 Lagrange 多項式，可得到：

$$f(x)=b_1.\frac{(x-a_2)(x-a_3)}{(a_1-a_2)(a_1-a_3)}+b_2.\frac{(x-a_1)(x-a_3)}{(a_2-a_1)(a_2-a_3)}+b_3.\frac{(x-a_2)(x-a_3)}{(a_3-a_2)(a_3-a_1)}$$

由上述方程式中可以判斷出可能會有出現係數不為整數的狀況，在程式做運算時會造成誤差而影響角色判斷的正確性。因此，在處理係數上將分子與分母分開運算。為配合上述狀況，使用者 user 的 table 中則以 userRoleA 紀錄 x 平方項係數、userRoleB 紀錄 x 項係數、userRoleC 紀錄常數項係數及 userRoleD 通分後分母的數值來作為因應。所以經通分整理後的方程式如下：

$$f(x)=b_1.\frac{(x-a_2)(x-a_3)(a_2-a_3)}{(a_1-a_2)(a_1-a_3)(a_2-a_3)}-b_2.\frac{(x-a_1)(x-a_3)(a_1-a_3)}{(a_1-a_2)(a_2-a_3)(a_1-a_3)}$$
$$+b_3.\frac{(x-a_2)(x-a_3)(a_1-a_2)}{(a_2-a_3)(a_1-a_3)(a_1-a_2)}$$

整理 x 平方項係數可得： $b_1(a_2-a_3)-b_2(a_1-a_3)+b_3(a_1-a_2)$

整理 x 項係數可得：

$$-b_1(a_2-a_3)(a_2+a_3)+b_2(a_1-a_3)(a_1+a_3)-b_3(a_1-a_2)(a_1+a_3)$$

整理常數項可得：

$$b_1a_2a_3(a_2-a_3)-b_2a_1a_3(a_1-a_3)+b_3a_1a_2(a_1-a_2)$$

經過以上整理，可以直接由使用者所屬角色的 3 點座標，計算出該使用者所代表的方程式，並由分子分母分開處理以解決非整數係數

帶來的誤差問題。

2.判定使用者的角色：

經推算已知使用者的角色方程式為：

$$f(x)=\frac{(b_1(a_2-a_3)-b_2(a_1-a_3)+b_3(a_1-a_2))}{(a_1-a_2)(a_1-a_3)(a_2-a_3)}x^2$$

$$\frac{-(b_1(a_2-a_3)(a_2+a_3)-b_2(a_1-a_3)(a_1+a_3)+b_3(a_1-a_2)(a_1+a_3))}{(a_1-a_2)(a_1-a_3)(a_2-a_3)}x$$

$$\frac{+(b_1a_2a_3(a_2-a_3)-b_2a_1a_3(a_1-a_3)+b_3a_1a_2(a_1-a_2))}{(a_1-a_2)(a_1-a_3)(a_2-a_3)}$$

當需要運算使用者的角色時，只需要將資料庫中的角色值，代入使用者的方程式，若得出來的解為允許值則該使用者即具有該角色的身份，反之則不具身份。

4.3 系統模組規劃

本節就系統架構中的角色管理模組、使用者管理模組、權限管理模組、電腦管理模組、備份管理模組、還原管理模組、image 管理模組、刪除管理模組及進行還原模組等九大模組進行介紹。

1.角色管理模組:

由於所有使用者所屬角色代表的方程式，都是利用系統中的角色值、允許值與不允許值運算後所求出來的，所以當系統管理者更動五個數值中任意一個數值，就會讓所有使用者在資料庫中紀錄的方程式係數全部無效，因此當系統管理者為了提高系統安全性，在一段固定

的時間更改數值時，系統就需要下列的流程，進行系統更新，如此才能保證所有角色運算的正確性，規劃其流程如圖 16。

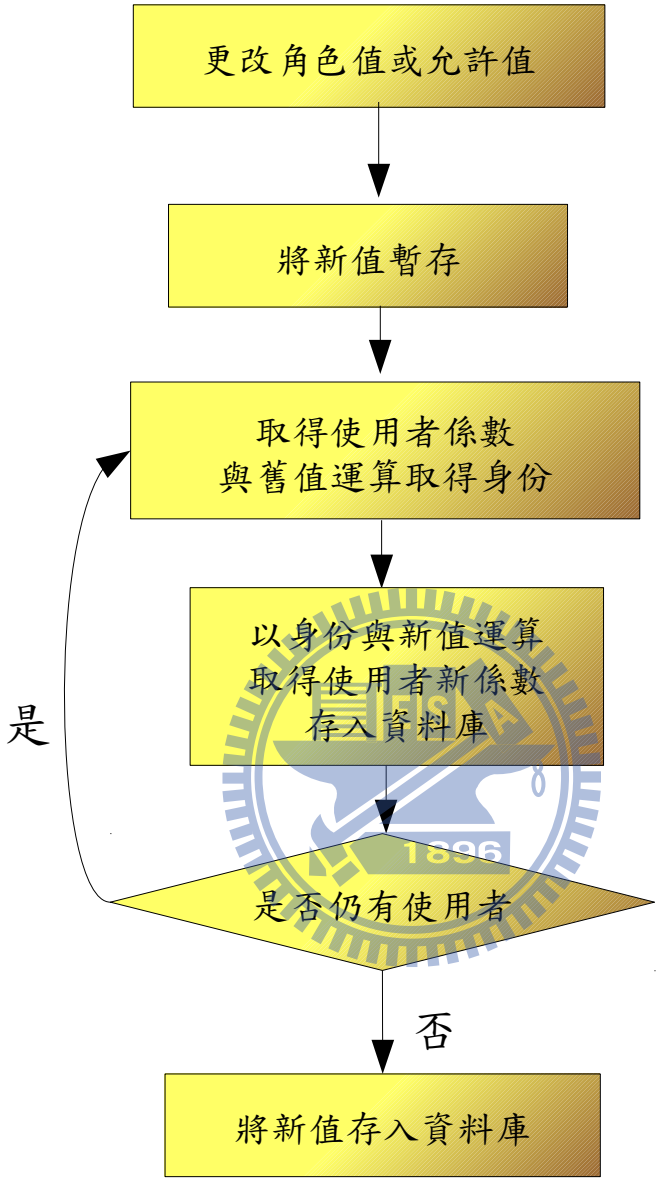

圖 16:角色值及允許值更改流程

2.使用者管理模組：

在使用者管理的模組中規劃了兩大部份：新增使用者與使用者更改。此兩部份的功能類似，差別在於新增使用者會新增新的帳號，而使用者更改則是將系統內已有的帳號進行密碼或角色的更動。兩者相同的部份：在密碼的處理會將設定的密碼進行 MD5 編碼後才存入資料

庫；在角色處理的部份，則會依照角色運算的流程進行處理。規劃其流程如圖 17。

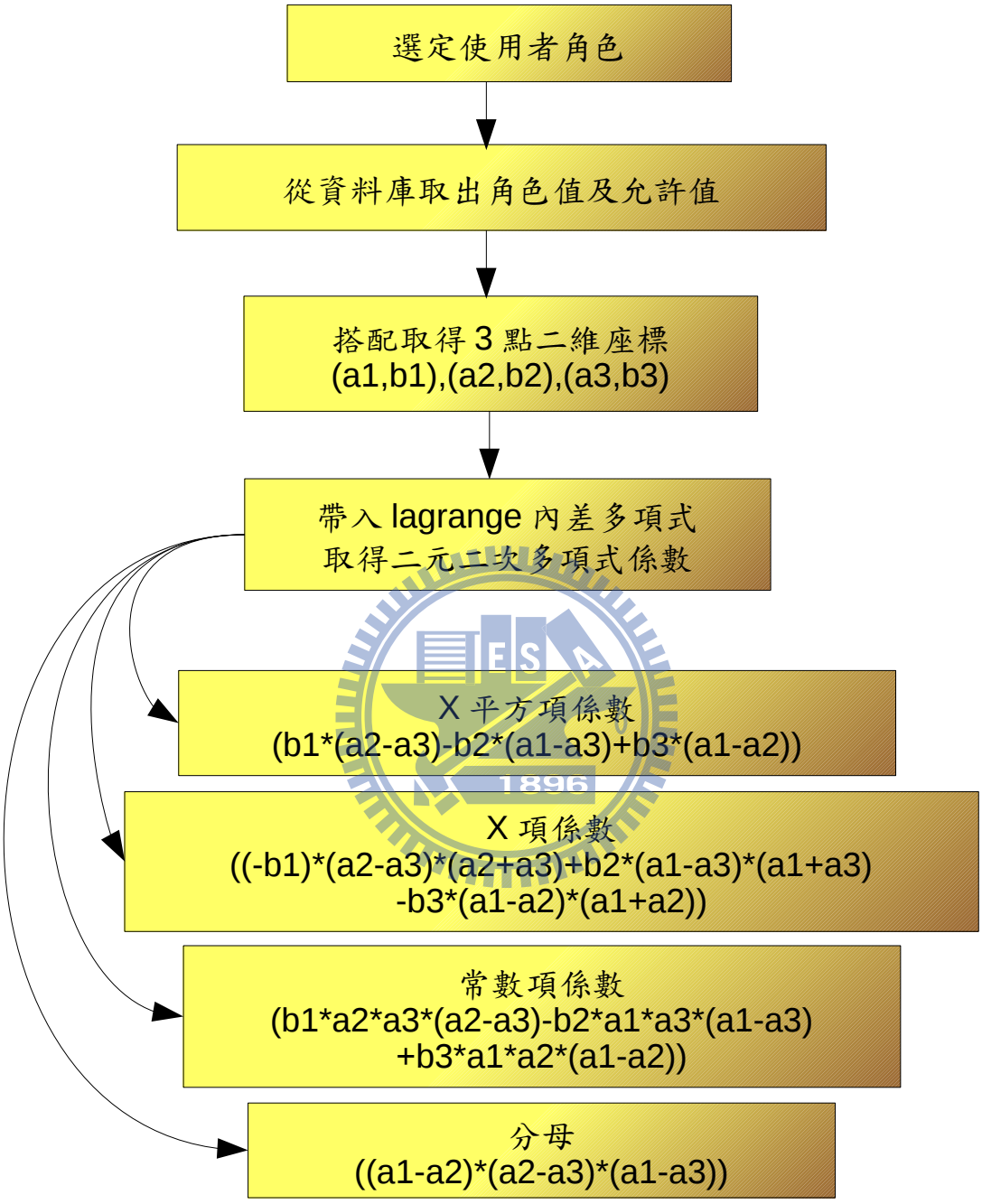

圖 17：角色運算流程

3.權限管理模組：

本系統共有三個角色：系統管理者、進階管理者及一般使用者。進階管理者的系統規劃是為了要協助系統管理者可協助管理，讓進階使用者可以登入後，進行協助管理。一般使用者基於安全考量僅提供自行還原該使用電腦的權限。因此規劃權限管理模組功能可提供系統使用者設定進階使用者可以協助管理的權限。規劃當進行設定後，使用者進行系統登入的流程如圖 18：

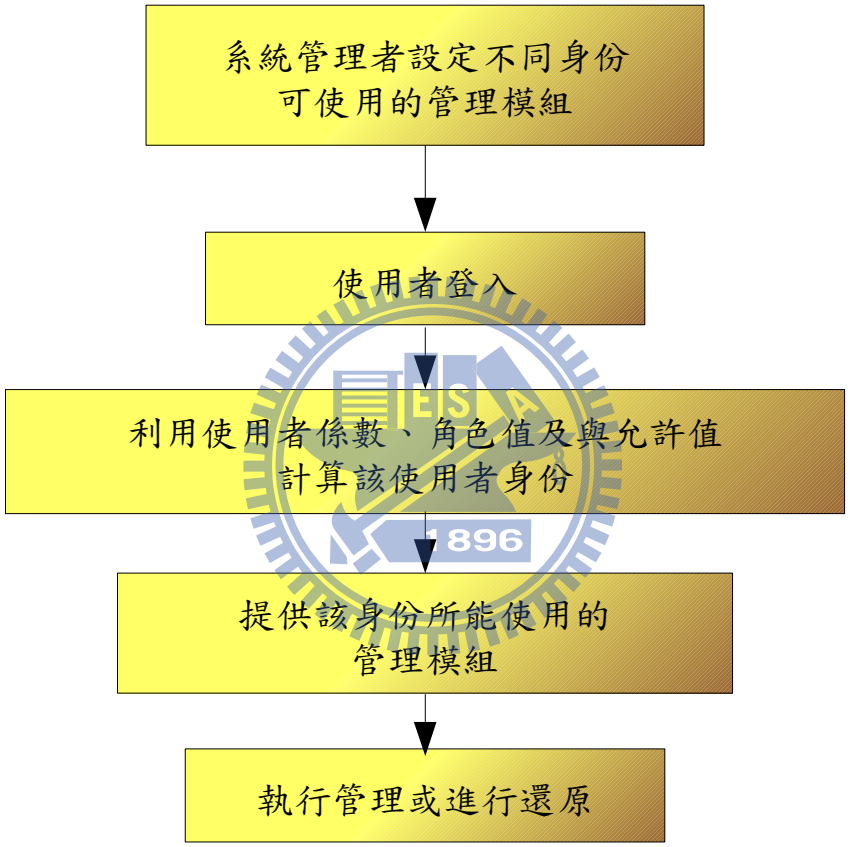

圖 18：系統登入流程

4.電腦管理模組：

在設定新電腦到系統時，電腦需要有唯一的識別方式，才能進行管理。從使用的觀點進行考量每一台電腦都有使用者在使用，所以每台電腦都該對應到一位使用者(公用電腦的使用者則設定為系統管理者)。而每台電腦都會屬於某一個使用群組(如：電腦教室)，此群組內系統的設定應該都是一樣的，如此進行大量還原管理時，即可由群

組進行處理，而不需要逐台電腦進行設定。另外，考量電腦數量多時，應該提供批次處理的功能。

基於以上的考量，電腦管理模組將提供下列功能：

(1)可產生電腦唯一的識別方式。

(2)可設定電腦所屬使用者。

(3)可設定電腦所屬群組。

(4)可批次處理大量電腦的設定。

5.備份管理模組：

此模組中提供的功能在對於已經設定好的系統環境進行備份，以提供此台電腦或同群組的電腦可以進行還原或升級。其規劃的系統流程如圖 19。

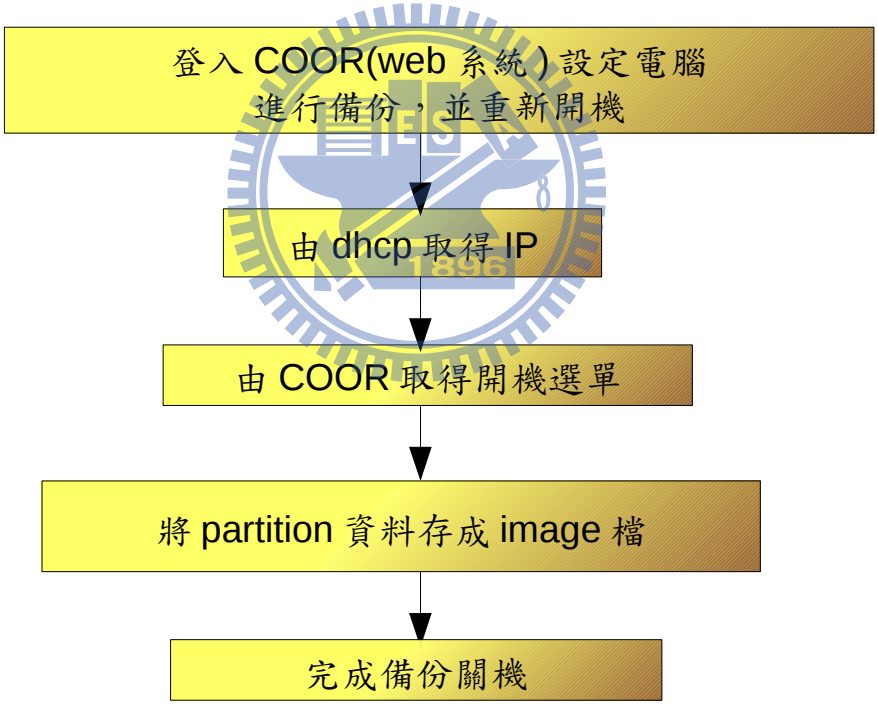

圖 19：備份流程

6.還原管理模組：

對所有的電腦進行還原時，考量到的狀況有：單台電腦出問題需重灌系統、整個群組的電腦進行更新還原、緊急時需立即還原、位節省電費及不影響上班時間運作，應於電費較低的時段進行還原。

基於以上的考量，還原管理應該提供下列功能，並規劃還原進行的流程如圖 20 及可使用 image 檔案判定的流程如圖 21：

(1)可進行單台電腦還原更新。

(2)可進行群組電腦還原更新。

(3)可進行立即還原更新。

(4)可依選擇時間進行還原更新選擇。

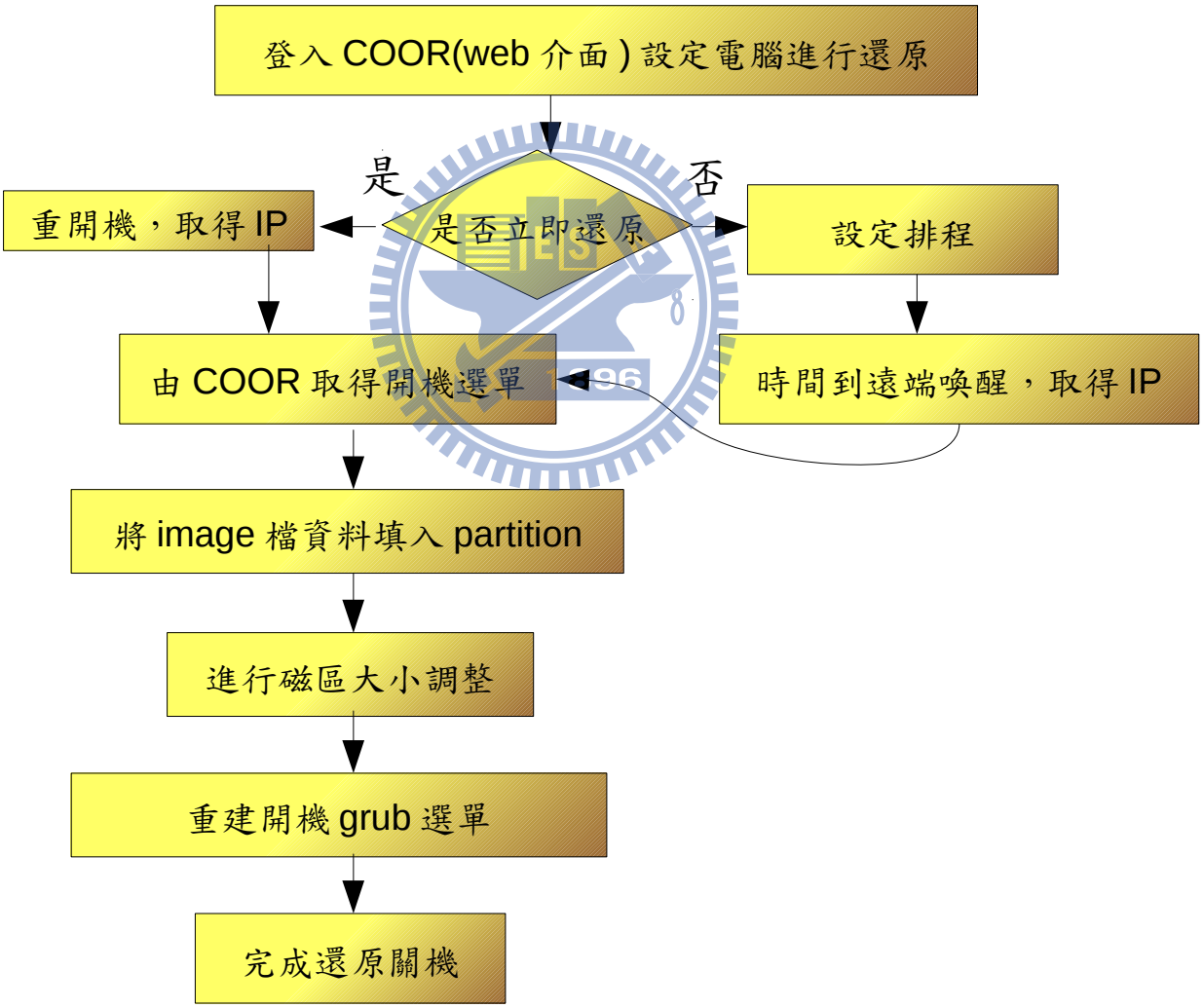

圖 20：還原流程

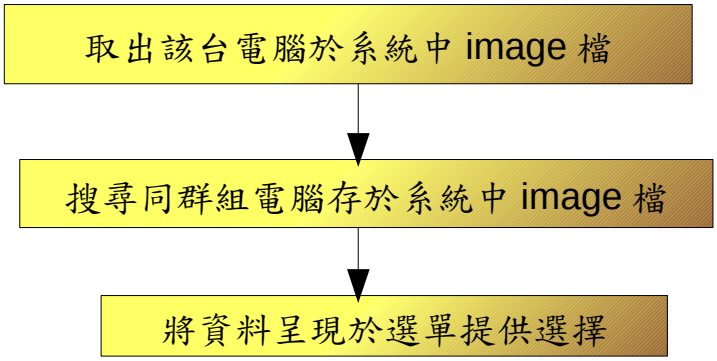

圖 21：可使用 image 檔產生流程

7.image 管理模組：

此模組在規範進階使用者可以儲存於系統中的 image 檔個數，考量的狀況有：同一台電腦產生過多的 image 檔案造成系統空間的浪費、進階使用者可能帶多台筆電到校內進行備份。

基於以上的考量，image 管理應該提供設定進階管理者可於系統中儲存的份數。

8.刪除管理模組：

由於酷鵝系統(COOR)可將所有的管理模組設定給進階使用者協助管理，擔心一旦使用者，不小心將設定刪除可能造成無法挽回的局面，因此將系統中有提供新增的模組裡刪除的功能，集中於此模組中進行管理，並且於刪除前得判斷其他關聯的 table 中是否仍需要該項資料，若無才能進行刪除，如此即可降低系統風險。

基於以上的考量，刪除管理應該提供下列資料刪除的功能：

(1)可進行電腦資料刪除。

(2)可進行使用者資料刪除。

(3)可進行群組資料刪除。

(4)可進行 image 檔案刪除。

9.進行還原模組:

此模組為基本模組為一般使用者基本的配置，而進階使用者因繼承一般使用者的關係，也具備有此項功能。基於考量一般使用者系統概念及系統操作能力有限，故僅規劃提供選則其所使用的電腦行還原及還原時間設定的功能。此模組規劃的系統流程同於還原模組。

第五章 系統建置與分析

本章於第 1 節的部份介紹 COOR 的資料庫實作，第 2 節討論 COOR 的系統實作，第 3 節介紹 COOR 實驗的進行，第 4 節介紹介紹 COOR 與其他系統的比較。

5.1 資料庫實作

本節將 4.1 節中所規劃的資料庫於 Mysql 中進行實作，共計有 rbac、role、user、manage、computer、cgroup 及 image 等七個 table。

1.rbac table:

此 table 設計紀錄三個角色值及兩個允許值供運算角色使用，由於數值將拿來運算，故每個欄位設定為 int 的型態，且僅會有一筆資料，故無設計 index 的欄位，如圖 22 所示:

	欄位	型態	屬性	Null	預設值	附加	執行
☐	role1	int(11)		否	0		
☐	role2	int(11)		否	0		
☐	role3	int(11)		否	0		
☐	allow	int(11)		否	0		
☐	deny	int(11)		否	0		

圖 22：design of 'rbac' table

2.role table:

此 table 設計不同的角色所可以使用的模組，其中 manageId 欄位因會紀錄多個 manageId，所以採用 varchar 的型態，資料以逗號分隔，存取時再由程式進行處理，如圖 23 所示:

	欄位	型態	屬性	Null	預設值	附加	執行
☐	roleId	int(11)		否		auto_increment	
☐	roleName	varchar(60)		否			
☐	manageId	varchar(100)		是	*NULL*		

圖 23：design of 'role' table

3.user table:

此 table 在 passwd 欄位以 char(32)的型態處理，紀錄以 md5 處理過的密碼長度。userRoleA、userRoleB、userRoleC 及 userRoleD 所紀錄的是該使用者角色代表方程式的係數，依序為 x 平方項係數、x 項係數、常數項係數及分母數值。UserManageCount 紀錄該使用者可以存於 COOR 中 image 的個數，預設為 1，細部設計如圖 24 所示:

	欄位	型態	屬性	Null	預設值	附加	執行
☐	userId	int(11)		否		auto_increment	
☐	userName	char(40)		否			
☐	passwd	char(32)		否	0		
☐	userRoleA	int(11)		否	0		
☐	userRoleB	int(11)		否	0		
☐	userRoleC	int(11)		否	0		
☐	userRoleD	int(11)		否	0		
☐	userManageCount	int(11)		否	1		

圖 24：design of 'user' table

4.manage table:

此 table 在處理 COOR 所提供的模組，及該模組所對應的程式位置，設計如圖 25 所示:

	欄位	型態	屬性	Null	預設值	附加	執行
☐	manageId	int(11)		否		auto_increment	
☐	manageName	char(60)		否			
☐	program	char(60)		否			

圖 25：design of 'manage' table

5.computer table:

此 table 在紀錄電腦的使用者及所屬群組，其中 computerIp 的欄位是為了紀錄該台電腦所設定的 IP，以確保電腦在網路上身份唯一性。imageId 的欄位會紀錄該台電腦目前使用的 image 檔案。設計如圖 26 所示:

	欄位	型態	屬性	Null	預設值	附加	執行
☐	computerId	int(11)		否		auto_increment	
☐	computerName	char(60)		否			
☐	userId	int(11)		否	0		
☐	cgroupId	int(11)		否	0		
☐	computerIp	char(60)		否			
☐	imageId	int(11)		是	0		

圖 26：design of 'computer' table

6.cgroup table:

此 table 在紀錄所有的群組，採用 cgroup 的名稱，有兩個目的：一為代表 computer 的群組，另一個為避免與 SQL 中的'group by'語法產生衝突，設計如圖 27 所示:

	欄位	型態	屬性	Null	預設值	附加	執行
☐	cgroupId	int(11)		否		auto_increment	
☐	cgroupName	char(60)		否			

圖 27：design of 'cgroup' table

6.image table:

此 table 在紀錄該 image 所適用電腦及群組，設計如圖 28 所示:

	欄位	型態	屬性	Null	預設值	附加	執行
☐	imageId	int(11)		否		auto_increment	
☐	imageName	char(60)		否			
☐	computerId	int(11)		否	0		
☐	cgroupId	int(11)		否	0		

圖 28：design of 'image' table

5.2 系統實作

本節將介紹於 4.3 節中設計的角色管理模組、使用者管理模組、權限管理模組、電腦管理模組、備份管理模組、還原管理模組、image 管理模組、刪除管理模組及進行還原模組等九大模組，於 Apache server 上以 php 語言的實作；以下依不同角色的操作介面進行介紹。

1.系統管理者-角色管理介面：

此模組在提供管理者進行數值的更動，當使用者更動任一數值時，除了更新'rbac' table 外，也會將所有使用者的角色方程式所代表的所有係數欄位進行更新，以確保系統角色運算的正確性。

圖 29：系統管理者-角色管理介面

2.系統管理者-使用者管理介面：

此模組在提供管理者進行新增使用者或更改使用者密碼、角色的介面，如圖 30 所示。

圖 30：系統管理者-使用者管理介面

3.系統管理者-權限管理介面：

此模組在提供管理者設定使用者可使用的模組，如圖 31 所示。

圖 31：系統管理者-權限管理介面

4.系統管理者-電腦管理介面:

此模組提供新增電腦、電腦所屬使用者與群組更改及由 CSV 檔案進行批次更改的功能。其中新增電腦的選項中，若下拉式選單沒有適合的群組，則由下方的空格直接輸入新的群組名稱，當按下確定時，即會新增該群組，如圖 32 所示。

圖 32：系統管理者-電腦管理介面

5.系統管理者-備份管理介面：

此模組當輸入 image 名稱後，按下確定即會將所選電腦備份。

圖 33：系統管理者-電腦管理介面

6.系統管理者-還原管理介面：

此介面提供管理者進行單一電腦或群組電腦的還原，程式會判定該電腦或群組是否適用所選的 image 檔案，另提供立即還原或選定其他時間還原的功能，如圖 34 所示。

圖 34：系統管理者-還原管理介面

7.系統管理者-image 管理介面：

此介面提供管理者設定使用者可存 image 檔的個數，如圖 35 所示。

圖 35：系統管理者-image 管理介面

8.系統管理者-刪除管理介面：

為避免進階使用者誤刪資料，故將所有刪除的功能其中到此介面，以提高系統的穩定性，如圖 36 所示。

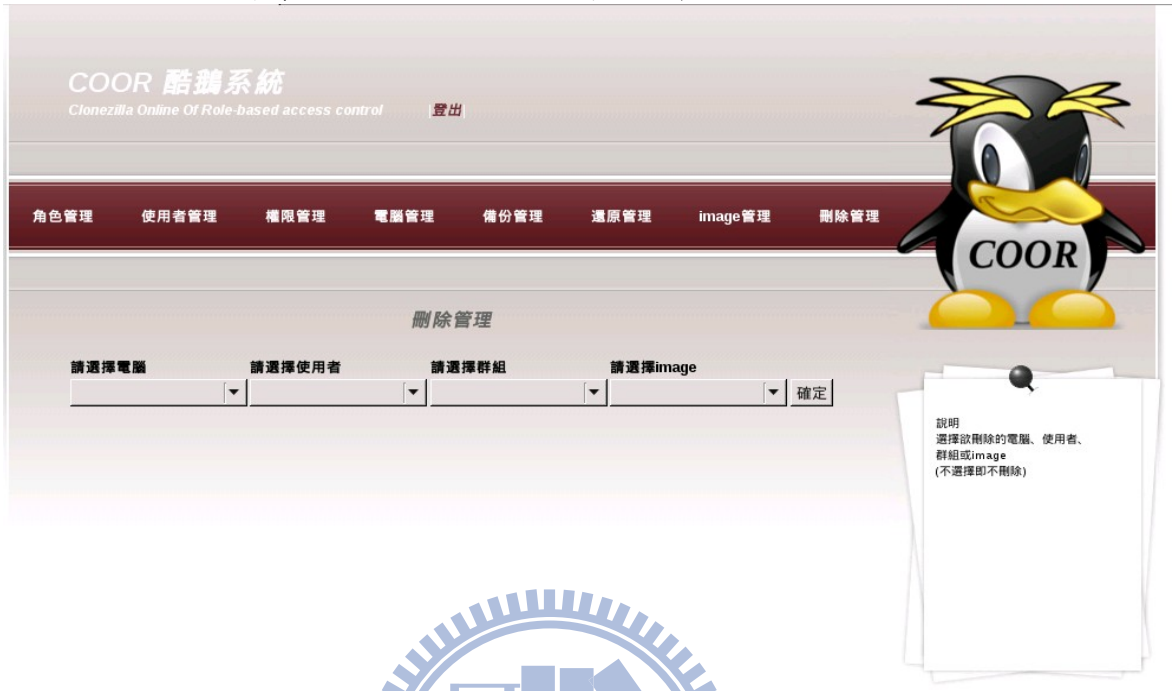

圖 36：系統管理者-刪除管理介面

9.進階使用者-進行還原介面：

此介面繼承自一般使用者，可對所使用電腦進行立即或排定其他時間的還原動作，如圖 37 所示。

圖 37：進階使用者-進行還原介面

10.進階使用者-備份管理介面：

此介面為系統剛建立時，進階使用者預設可以使用的管理介面以協助系統管理者進行管理，若要提供其他模組的使用，則可由系統管理者-權限管理介面進行設定，如圖 38 所示。

圖 38：進階使用者-備份管理介面

11.一般使用者-進行還原介面：

此介面為使用者最基本的功能，可對該使用電腦進行維護，如圖 39 所示。

圖 39：一般使用者-進行還原介面

5.3 實驗建置實作

本節將依硬體建置、軟體建置、網路配置及實驗步驟四個部份進行介紹。

1.硬體建置：

網路設備採用 4 台無網管功能之超高速乙太網路交換器 24 埠 10/100/1000Base-T(SMC 8024L)，server 及 client 端硬體規格如表 9 表 10 所呈現。

表 9：Server 硬體規格

型號	asus TS-300
CPU	Intel Xeon 3050 (2.13GHz / 1066MHz FSB /2M L2 cache)
RAM	2GB DDRII 667MHz ECC
硬碟	250GB SATAII 7200 rpm
RAID	0,1,0+1
網卡	二組 Broadcom® BCM5721 Gigabit Ethernet Controller

表 10：Client 電腦硬體規格

型號	acer FT100
CPU	Intel Celcron 2.8GHz (533MHz FSB)
RAM	DDR400 256MB+512MB=768MB
晶片組	SiS 661 FX +964L
硬碟	80GB IDE 7200 rpm
網卡	10/100Mbps(含 PXE 功能)

2.軟體建置：

client 端電腦於實驗中安裝的作業系統為 B2D linux，此系統是由台南縣網由 Knoppix 進行改良，除了整合 KDE 與 Gnome 兩種桌面系統及對於中文部份進行優化外，在軟體部份也針對使用者常用及好用的相關應用程式進行預先安裝以方便使用者使用，整個系統安裝完畢約佔用 9GB 硬碟空間。

Server 的系統部份，欲安裝酷鵝系統其系統的軟體需求如表 11 所示：

表 11：Server 系統需求

OS	Ubuntu 10.04
Web service	apache server (2.2.12-1ubuntu2.4) php (5.2.10.dfsg.1-2ubuntu6.5)
Database service	mysql server (5.1.37-1ubuntu5.5)
Other service	clonezilla server(1.2.6-40) 、dhcp server(3.1.3-2ubuntu6)

3.網路配置：

由於 COOR 提供的服務僅限校園的 intranet，所以為了提高 COOR 安全性，關閉 COOR 的對外連線且將 NAT 及 DHCP 的功能轉由防火牆來提供，其網路配置如圖 40 所示：

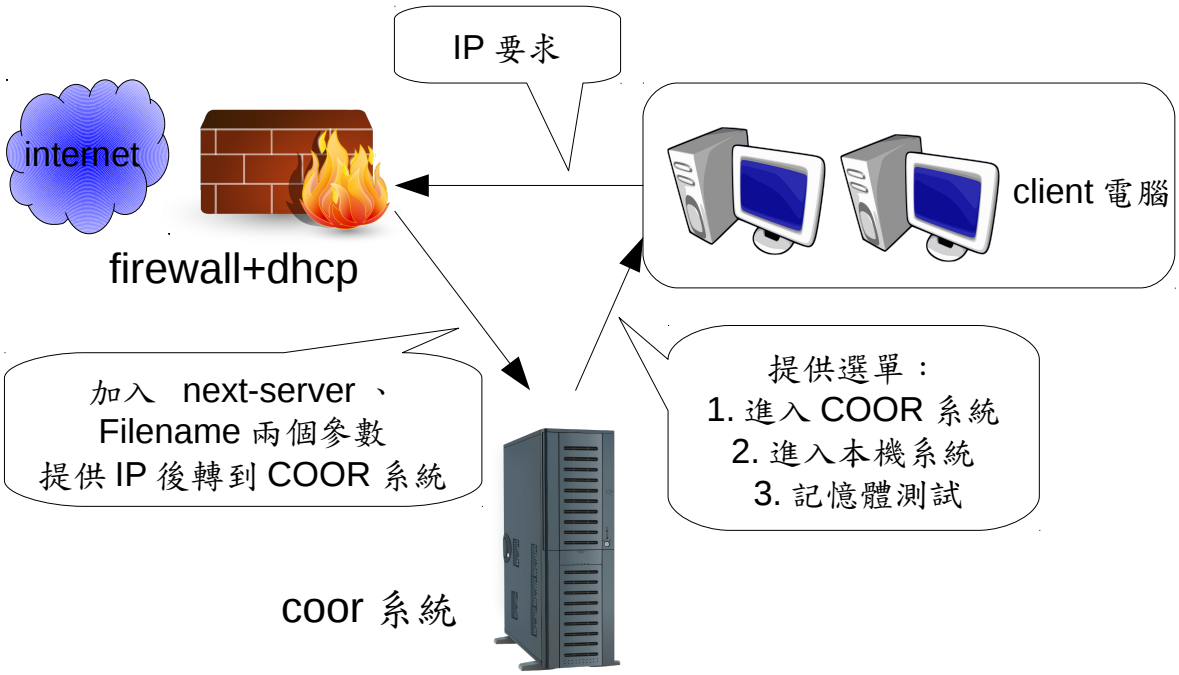

圖 40：網路配置

4.實驗進行：

實驗場地在本研究者任職學校的電腦教室進行，分成使用者自行維護、系統管理者進行備份及系統管理者大量維護三部份進行。

(1)使用者自行維護：

首先使用者先登入 COOR 進行還原設定，完成設定後電腦重新開機，操作畫面如下：

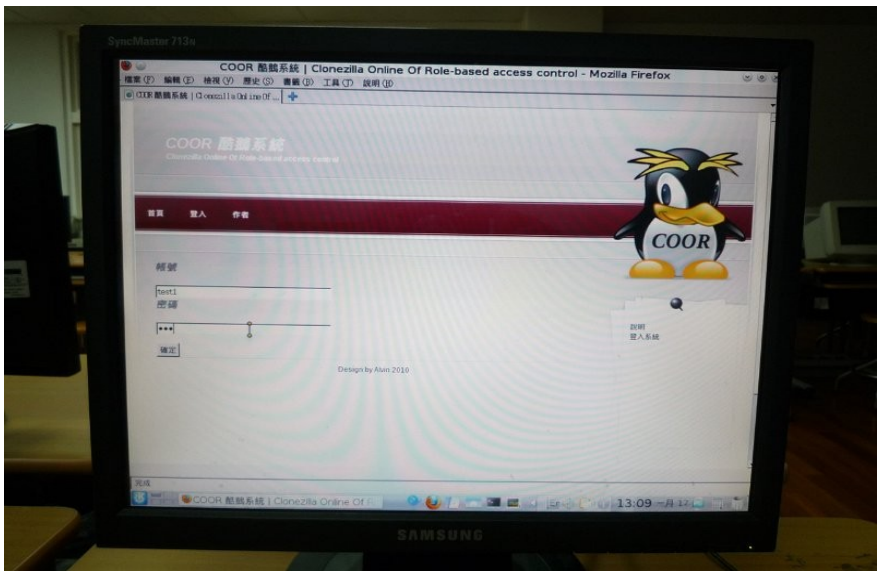

圖 41：一般使用者登入

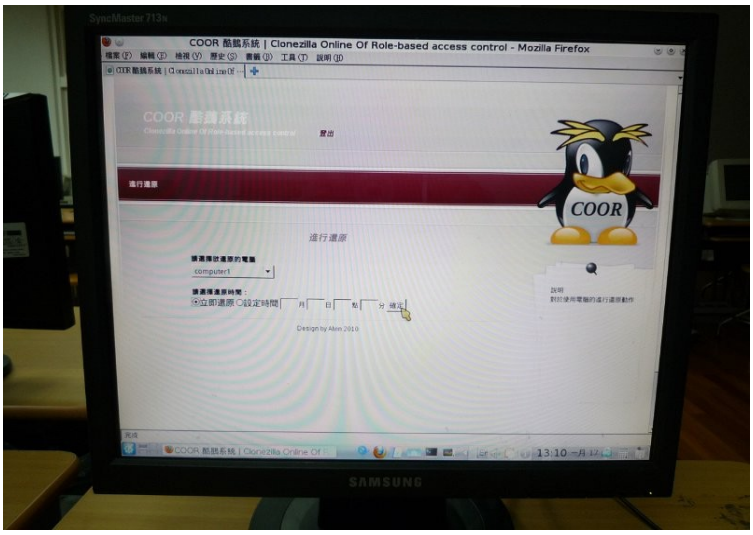

圖 42：一般使用者設定還原動作

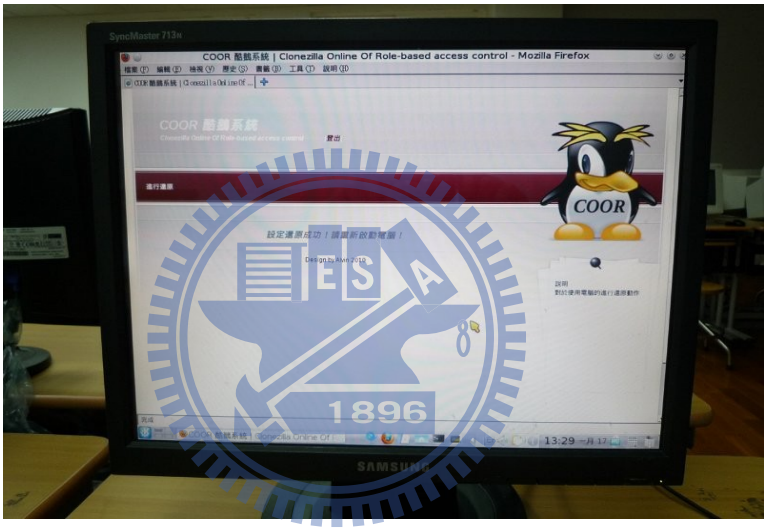

圖 43：一般使用者設定設定成功準備重新開機

圖 44：取得開機選單

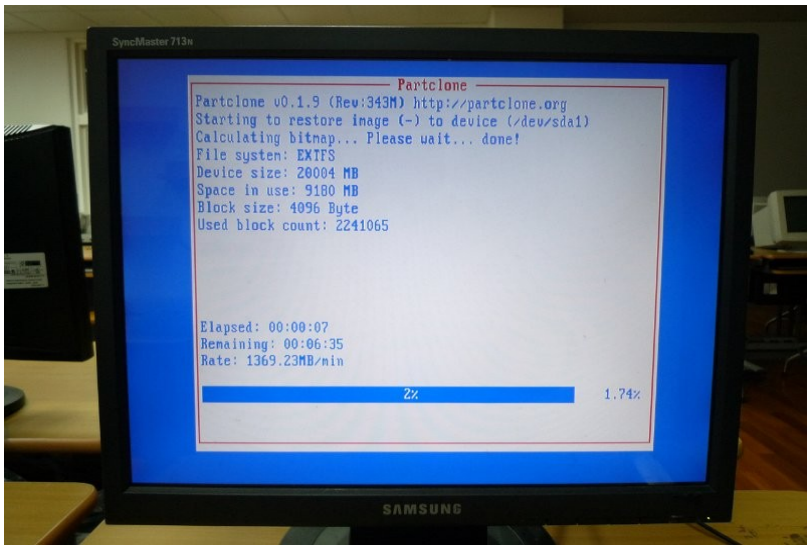

圖 45：一般使用者電腦進行還原

由實驗得到單機還原所需時間為 6 分 42 秒，rate 1369.23MB/min。

(1)管理者進行備份：

管理者先登入後，選擇欲備份的電腦，並設定 image 檔名，將該台電腦重開機，及進入備份模式，操作畫面如下：

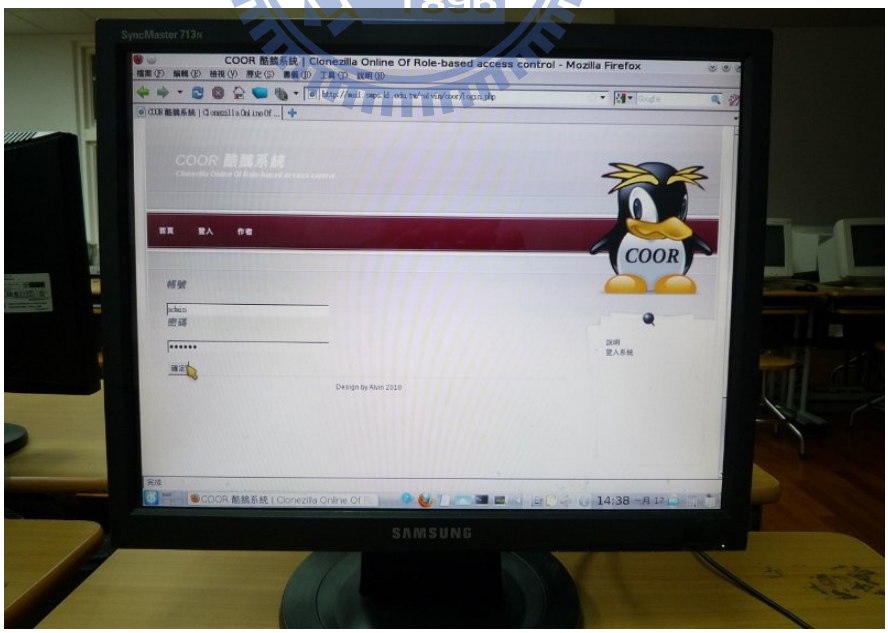

圖 46：系統管理者登入

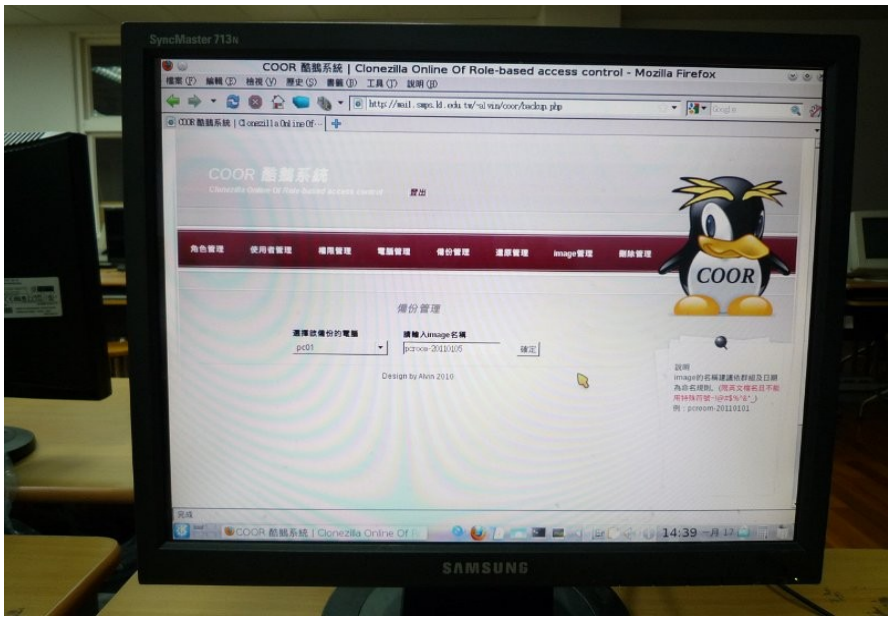

圖 46：系統管理者設定備份程序

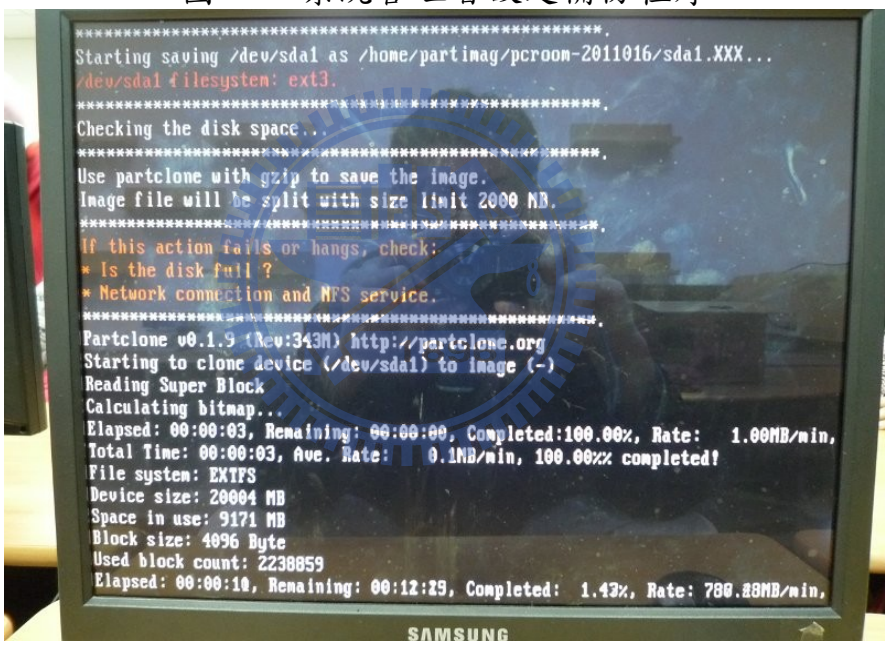

圖 47：備份畫面

由實驗得到備份實際資料 9171MB 大小的磁區，所需時間為 12 分 40 秒，rate 789.38MB/min。

(3) 系統管理者大量維護：

系統管理者登入後，設定群組與還原時間，確認後系統會從遠端開啟與還原群組的所有電腦，操作畫面如下：

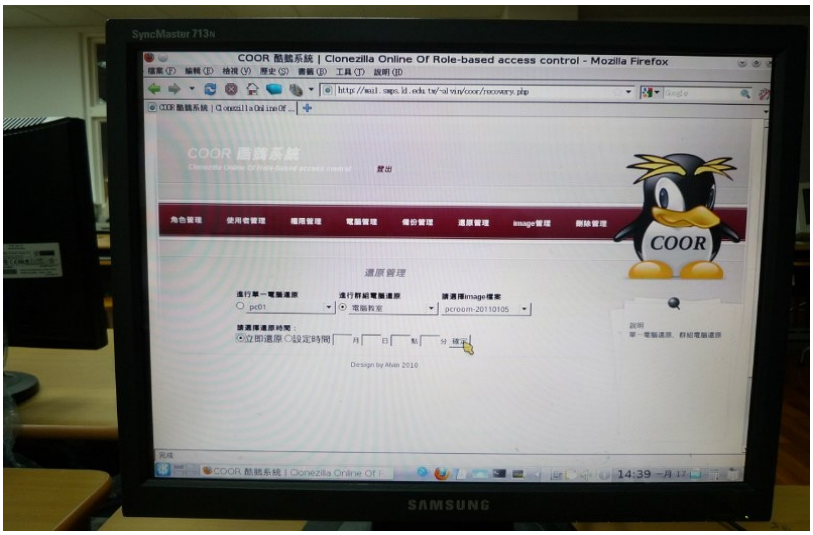

圖 48：系統管理者設定群組電腦還原

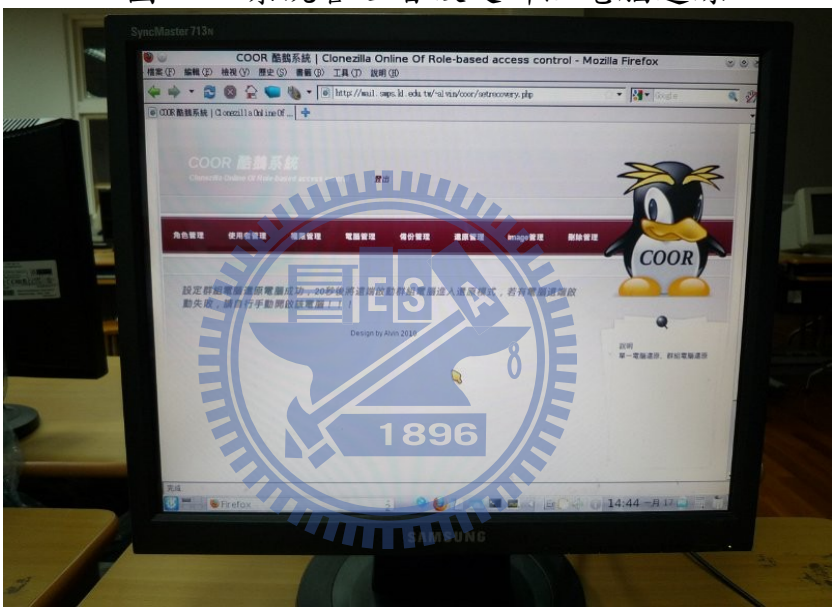

圖 49：系統管理者設定群組電腦還原成功畫面

圖 50：72 台群組電腦進行還原畫面

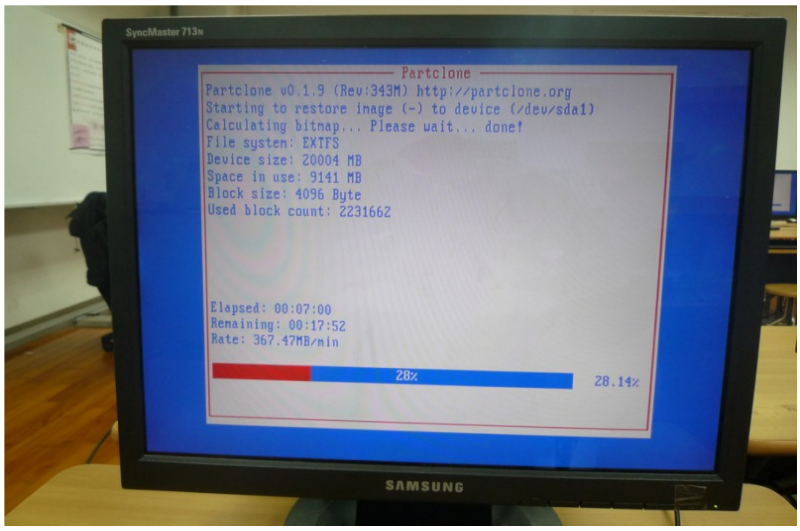

圖 51：群組電腦進行還原單機畫面

由實驗得到還原實際資料 9171MB 大小的磁區，在 72 台電腦同時運作的狀態下，所需時間為 24 分 52 秒，rate 367.47MB/min。

5.4 系統分析比較

茲將 COOR 與其他三項研究，就管理與操作進行比較整理如表 12。以系統的管理、系統易用性及降低系統管理人員的負擔來說，COOR 是優於其他三項的研究。

表 12：酷鵝系統與其他研究比較表：

系統名稱	酷鵝系統	還原卡	排程還原	無碟系統
網頁式管理介面	有	無	有	無
一般使用者可否自行操作	可	不可	不可	不可
管理者可否同時進行不同的還原工作	可（批次處理）	可（逐台操作）	不可	不需要
系統不穩時，單機可否正常操作	可	可	可	不可
系統操作難易度	易	中	難	難

第六章 結論與未來工作

6.1結論

本研究 COOR 為了達到預設目標，採用以下的方法，讓系統能更完整、更安全且更方便的使用。

1.COOR 所採用的所有軟體均為自由軟體，以達到節省經費支出的目的。

2.加入 RBAC 概念及 MD5 編碼提系統安全性，且藉由多角色的處理提供一般老師自我管理電腦的功能，藉以降低資訊組長的負擔。

3.提供網頁式介面，降低使用難度讓一般老師可以無痛上手。

4.提供網頁式管理介面，提供資訊組長更彈性且方便的操作。

6.2未來工作

本研究 COOR 限定於校園的區域網路中，未來若將 COOR 運行於 internet 的環境時，可朝下列幾點方向進行更深入的探討。

1.需要針對安全性做更深入的探討。

2.在校園的網路環境中，頻寬影響不大，但 internet 上 client 頻寬將影響系統效能。

3.在 internet 的網路環境中，因應 client 端在系統進行還原運作時斷線，需考慮備援機制。

4.目前 COOR 設定的環境是校園環境，所以並不進行使用者付費考量，但若將 COOR 運作於商業模式中，則可依據使用者所需要儲存的 image 數量進行收費，並另外設置出納、業務...等其他角色，來搭配使用。

參考文獻

[1]D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn and R. Chandramouli, “Proposed NIST Standard for Role-Based Access Control”, ACM Transactions on Information and System Security, Vol. 4, No. 3, pp. 224-274, 2001.

[2]Ravi S. Sandhu , Edward J. Coyne , Hal L. Feinstein , Charles E. Youman, “Role-Based Access Control Models”, Computer, v.29, n.2, p.38-37, February 1996

[3]R. S. Sandhu, E. J. Coyne and C. E. Youman, “Role-Based Access Control Models”, IEEE Computer , pp. 38-47, 1996.

[4]R. S. Sandhu, D. Ferraiolo and R. Kuhn, “The NIST Model for Role-Based Access Control：Towards a Unified Standard”, Proceedings of the 5th ACM Workshop on Role-Based Access Control, pp. 26-27, 2000.

[5]Peter Wayney 著，開放原始碼，蔡憶懷譯，商周出版 ，台北市，2000

[6]Stallings William 著，密碼學與網路安全原理與實務，曾志光，巫坤品譯，碁峰出版社 ，台北市，2001

[7]王光山，「以自由軟體協助電腦教室管理」，朝陽科技大學資訊管理系碩士班碩士論文，2002

[8]王德源，「網路系統備份與復原」，國立中興大學電機工程學系碩士論文，2002

[9]廖瑞民，「完全獨立之硬碟資料瞬間復原架構」，靜宜大學資訊管理學系研究所碩士論文，2002

[10]林孟勳，「結合 RBAC 授權之網站單一簽入機制研究」，世新大學資訊管理學系碩士論文，2005

[11]Lagrange 內差多項式，取自 http://zh.wikipedia.org/zh-tw/拉格朗日插值法 ，2010 年 12 月 15 日

[12]LTSP，取自 http://en.wikipedia.org/wiki/Linux_Terminal_Server_Project ，2010 年 12 月 15 日

[13]NIST，取自 http://www.nist.gov/index.html ，2010 年 12 月 15 日

[14]Udpcast，取自 http://udpcast.linux.lu/ ，2010 年 12 月 15 日

[15]NIS，取自 http://en.wikipedia.org/wiki/Network_Information_Service ，2010 年 12 月 15 日

[16]NFS，取自 http://en.wikipedia.org/wiki/Network_File_System_%28protocol%29 ，2010 年 12 月 15 日
[17]TFTP，取自 http://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol ，2010 年 12 月 15 日
[18]DHCP，取自 http://zh.wikipedia.org/zh-tw/DHCP ，2010 年 12 月 15 日
[19]drbl，取自 http://drbl.nchc.org.tw/ ，2010 年 12 月 15 日
[20]clonezilla，取自 http://clonezilla.nchc.org.tw/ ，2010 年 12 月 15 日
[21]自由軟體分析-深美國小，取自 http://www.oss.org.tw/?q=node/37 ，2010 年 12 月 15 日
[22]GNU General Public License，取自 http://www.gnu.org/licenses/gpl.html ，2010 年 12 月 15 日
[23]國家高速網路中心，取自 http://www.nchc.org.tw/tw/ ，2010 年 12 月 15 日