國立交通大學

資訊學院 資訊學程

碩 士 論 文

實作一個在 UNIX 環境上的資源監控系統

Implementing a resource monitoring system on UNIX environment

研　　究　　生：王 鵬 翔

指 導 教 授：王 豐 堅　教授

中 華 民 國 九十九 年 四 月

實作一個在 UNIX 環境上的資源監控系統

Implementing a resource monitoring system on UNIX environment

研 究 生：王鵬翔　　Student：Peng-Hsiang Wang

指導教授：王豐堅　　Advisor：Dr. Feng-Jian Wang

國 立 交 通 大 學

資訊學院　資訊學程

碩 士 論 文

A Thesis

Submitted to College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

Apr 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年四月

實作一個在 UNIX 環境上的資源監控系統

學生：王鵬翔　　指導教授：王豐堅

國立交通大學　資訊學院　資訊學程碩士班

摘　要

在 UNIX 環境中，同時使用多台的工作站，若不能有效的分配負載，不僅造成工作站使用率低，還會導致服務品質不佳。為了提高使用率，許多使用者會採用 Load Sharing 系統來改善。然而，市面上的 Load Sharing 系統皆只針對 CPU 的 utilization 來做管理，使用對象侷限在 designer 的 batch job，對其它工作站資源（例如：Memory）及使用者的便利則考慮不多。

在本篇論文中，我們提供了一套資源監控系統給所有使用者。它可以掌握所有工作站資源的使用狀態，讓使用者輕鬆地挑選可利用的工作站來作業，減少因多人共用一台工作站導致其它工作站閒置的情況發生。這套資源監控系統以 perl/Tk 來撰寫，架構為 client-server，它提供直覺且友善的 GUI 介面給使用者。同時它採用 socket 方式來傳遞資訊，並每 60 秒更新資訊一次。除了上述功能的提供外，我們也根據軟體的使用性、容錯能力、效能及重複使用性分析，確保它的品質。

關鍵字：負載共用，電子設計自動化，資源監控

Implementing a resource monitoring system on UNIX environment

Student：Peng-Hsiang Wang　　Advisor：Dr. Feng-Jian Wang

Degree Program of Computer Science
National Chiao Tung University

ABSTRACT

In an UNIX environment, if the loads cannot be distriputed effectively, the execution in multiple workstations might have several defects, e.g., low usage rate in workstations, poor service quality, etc. Currently, many users apply load sharing system to enhance the usage rate. However, commercial systems of resource sharing managements are based on CPU utilization and their functions work on the management of designer's batch jobs only. To user, the usability and resource accessing of other workstations are less concerned.

In this thesis, we describe a resource monitoring system, RMS, to solve above problems. RMS is designed to monitor the usage of all the workstation resources. By providing an intuitive and friendly interface, RMS can help reduce the overall loading in the whole system since the available workstation(s) can be easily selected to work. The second part in RMS is to update related information in each workstation (e.g., CPU status, memory) every 60 seconds. This part can clear the garbage process/memory and help increase utilization too. RMS is developed with perl/Tk and based on client-server architecture, where socket method is used for data exchage. Finally, we also analyze usability, fault tolerance, efficiency and reusability to ensure its quality.

Keywords： Load sharing，EDA，Resource monitor

誌　　謝

首先感謝我的指導教授 王豐堅老師在學生就學期間耐心地給予指導與彈性的研討時間。再者感謝口試委員 吳毅成教授、梁德容教授及黃俊龍教授願意擔任學生的口試委員，並且在口試時，提供學生在研究上可以再改善的寶貴意見，使本論文得以修改得更完善。

對在職進修的研究生而言，同時有工作上和課業上的雙重壓力，所以必須感謝公司主管張念祖在學生就學期間工作上的安排，以及溫坤龍同仁、石惠如同仁的協助，使得學生得以順利完成學業。

最後，要感謝我的家人一直給我無限的支持，讓我能夠專心學習，而無後顧之憂。

目　　錄

表　　目　　錄

圖　　目　　錄

第一章、緒論

1.1 動機

在探討一個系統效率的時候，資源的使用並不只有 CPU，還包含 Memory、Disk 等，甚至作業系統的版本也是；在過去已有許多演算法被提出（[4], [5], [6], [7], [8], [10], [12], [14], [15], [16], [17], [18]），雖然這些方法各有其特色、分析及論證，但 end-user 要的卻不是這樣。在 UNIX-like 的環境下，我們觀察到：1）designer 常需要 simulation，所以需要強大的運算主機資源 2）layout 人員要做繪圖，圖中包含大量的 component，所以需要較大的記憶體資源 3）對使用 EDA（Electronic Design Automation）人員而言，常發生某些 vendor 因併購或淘汰，不再支援某些產品，tool 無法更新，或者當初購買時，license 綁定在某主機上，故只能綁定在某些 platform、版本、主機上執行，也有最新的 EDA 只支援到某些 platform、版本情況發生 4）當在趕計劃的尖峰時刻，資源使用率高，資源搶佔情形嚴重，end-user 常反應資源使用不足，並質疑 IT 人員讓許多主機閒置而未釋出給使用者。

雖然在 UNIX-like 的環境下，市面上已有許多 Load Sharing 的工具 [3], [13], [25], [26], [27]，但大都是針對需要 CPU 運算的 Batch Job 作處理，對非 Batch Job 則沒有幫助，只能解決上述第 1 點，再則，隨著科技的進步，現在的主機大多已非單一 CPU 的主機，雙 CPU，甚至更多 CPU 的主機已漸漸普及，而多 CPU 主機則帶來一些執行上的差異，一般來說，有的 Load Sharing 管理系統，在分配 CPU 資源時，皆是以 CPU 平均值來決定分配的，然而，以此方法略顯不足。舉例來說，有二台主機，甲台主機為雙 CPU 主機，而乙台為 8 CPU 主機；當甲主機其中一顆 CPU 被使用，其平均值為 50%；而乙主機其中 6 顆 CPU 被使用時，其平均值為 75%；若依現有 Load Sharing 管理系統，下一個工作會被分配到甲主機，而事實上乙主機比甲主機的資源更為寬裕。更進一步說，在企業內大都還留有舊型主機，食之無味，棄之可惜，如果一個需要大量運算的 job 被分配到舊主機，它需要花較多的時間來執行。所以，若能提供一個簡單直觀的工具，讓 end-user 能掌握當下所有主機與詳細的資源使用情形，包含硬體 Spec. 及每顆 CPU 的使用情形，以方便其尋找可利用之主機，一則可提高負載平衡，二則可讓 end-user 解除資源利用不足的疑慮或不公平？

1.2 目的

對於以上的不便與限制，我們提供了一個動態的資源監控系統（Resource Monitor System, RMS）來解除 Batch Job 的限制。該資源監控系統隨時確認主機存活狀態，有效掌握所有主機資源的使用狀態，並且依作業平台（Solaris、Linux、HP）及記憶體大小（24GB 以下、32GB、64GB 以上）來分類，讓使用者可快速地找到可利用的資源。而此資源監控系統是一個獨立系統，對於之前已建立的 Load Sharing 管理系統沒有任何衝突，仍可繼續使用，不會浪費已有建置成本。

1.3 章節簡介

本篇論文的內容大致如下

第二章將對背景做些介紹，包含現有 Load Sharing 系統工具的比較及使用上的優缺點。

第三章則詳細說明本資源監控系統的設計架構及運作方式。

第四章會說明此資源監控系統的實作結果及性能分析。

第五章將會作些討論並提到此系統未來的可改善目標。

第二章、背景

在 UNIX 環境下，目前市面上已有許多 Load Sharing 系統 [13]，其優點包括可平行處理 job、動態負載平衡、GUI 模組化、容錯性佳、彈性排程、授權機制等，功能相當強大；但亦有不足的地方，比如 source code 不開放、價錢昂貴、限定執行的作業平台、只能處理 batch job 等。

2.1 UNIX 環境之主流 Load Sharing 系統比較

表 2.1 對目前市場上二套主流的 Load Sharing 管理軟體作比較 [26], [27]，第一套為 LSF（Load Share Facility），第二套為 SGE（SUN Grid Engine）系統。

表 2.1 LSF 及 SGE 比較表

	LSF	SGE
異質平台支援	支援 SUN、Apple MacOS、HP、IBM AIX、Linux、Windows、NEC SX-8、SGI Irix	支援 SUN、Apple MacOS、HP、IBM AIX、Linux、Windows
產品支援	Platform 專門 24x7 支援	提供技術資料庫及使用 web 支援
可靠度和容錯	有 check-pointing，job failure 會重新執行，fail-over 時保證 0 downtime，可靠度和容錯度高	有容錯能力且可靠度佳
規模大小	目前已超過 2000 個客戶使用，包含 100 以上 cluster，200,000 顆 CPU，500,000 active jobs	規模最大可到 63,000 顆 CPU
報表	使用 LSF 本身提供的 accounting 來收集資料，精確度較高	使用 UNIX 提供的 accounting 來收集資料
價格	價格昂貴，以 CPU 顆數來計算	有免費版及 Enterprise 版本，

		Enterprise 版的價格相比 LSF 也較低

在表 2.1 中，我們從下述六個角度出發去判斷彼此的優劣

1. 異質平台支援

LSF 支援的作業平台包含支援 SUN、Apple MacOS、HP、IBM AIX、LINUX、Windows（2000、XP、Vista、2003、2008）、NEC SX-8、SGI Irix 等。

SGE 可執行主機的作業平台包含 SUN、Apple MacOS、HP、IBM AIX、Linux、Windows(2000、XP)，但 Master 主機必須要為 SUN 或 Linux。

2. 產品支援

Platform 公司的 LSF 產品，有專門人員作每天 24 小時的即時技術服務，也有 web portal 提供服務。

SGE 則提供技術資料庫及使用 web 服務，但沒有每天 24 小時即時服務，所以不一定可立即解決使用者的問題。

3. 可靠度和容錯

LSF 有 check-pointing，若 job failure，將自動重新執行，且若主機 fail-over 時，保證 0 downtime ，job 不會因軟硬體錯誤而遺失，可靠度和容錯度高。

SGE 也有不錯的容錯能力及可靠度，但偶爾會有 job 執行失敗情況發生。

4. 規模大小

LSF 目前已超過 2000 個客戶使用，包含 100 以上 cluster，200,000 顆 CPU，500,000 active jobs。

SGE 其規模最大可管理到 63,000 顆 CPU。

5. 報表

在收集 job 的 running 和 complete 資料中，LSF 使用本身提供的 accounting 系統，精確度較高，且報表能力強，可依使用者、主機、EDA license 的使用等條件，製作出華麗的報表。

SGE 則使用 UNIX 提供的 accounting 來收集資料，其報表能力比較陽春。

6. 價格

LSF 的價格是以 CPU 顆數來計算，而 UNIX 的 CPU 價格又是 Linux

CPU 的二倍，所以整體建置費用昂貴。

SGE 則分免費版及 Enterprise 版，即使是 Enterprise 版，其建置費用也比 LSF 低。

由表 2.1 的比較表可知 LSF 比 SGE 在許多方面有優勢，功能性較 SGE 強大，但並不意味著 LSF 就可以完全取代 SGE。何況，除了功能性外，SGE 也有相對優點，比如「價格」：LSF 須花錢購買，且價格不斐，對數量眾多的主機，整體建置費用較為昂貴，而 SGE 有免費版及 Enterprise 版本，只要些許花費與客製化，就能達到類似效能，僅管它有較多限制。而且隨著 Linux 的廣泛使用、功能的提昇、報表功能的加強、EDA tool 的 support 等多項功能提升，近年來 SGE 的市佔率已有相當的提升，「價格」永遠是企業的主要考量之一。

2.2 UNIX 環境之 Load Sharing 系統的缺點

除前一節所說 UNIX 環境之缺點外，我們亦從 end-user 與 system administrator 這兩個角度，分析其使用上的缺陷

1. End-user 方面
   （1） 現有 Load Sharing [13, 26, 27] 系統中，要 submit 的 job 必須為 batch job，若不是，則無法使用此 Load Sharing 系統處理。
   （2） 現有 Load Sharing 系統會限制在某些作業平台上才有支援。
   （3） 使用者只能掌握部分的主機狀態，無法全面掌握目前所有主機的狀態。
   （4） 需要資源監控系統提供立即的，且更為詳盡的主機資源使用狀態，包含每顆 CPU 的使用情形、軟體版本，及硬體 Spec. 等，以協助使用者開發。

2. System administrator 方面
   （1）必須用其 Load Sharing 管理系統的方法或指令來 submit job，若有使用者不遵照規則，則可能出現資源搶佔情形。
   （2） 建置費用高昂，企業可能先只利用些許數量的 computing nodes 建立此 Load Sharing 系統。
   （3） source code 不開放或有限制，客製化能力有限。

在 UNIX 環境中，現有的 Load Sharing 系統皆針對 CPU 的

utilization 做管理，對於電腦主機其它資源（如：Memory）則無能為力，市面上也看不到能提高其它資源使用率的軟體，鑒於以上的種種缺點，我們將提供一個工具來解決上述的問題。這一工具除了保障現有 Load Sharing 運行架構正常外，它也讓所有使用者能容易監控所有主機資源，以達到互補之效。換句話說，這一工具將讓使用者輕鬆地尋找可利用之主機來執行，以提高工作效率。

第三章、設計與架構

3.1 設計概念與目標

本資源監控系統不但要與原有的 load sharing 系統達到互補之效，而且最好還能給所有人員使用，並非侷限 load sharing 使用人員，因此，考慮多方需求，我們對其使用者展開意見調查，希望資源監控系統能提供哪些功能及目標，以下為調查結果，整理如下

1. System Administrator 方面
   （1）確認資源監控系統運作時不會有 dead lock 現象。
   （2）確認資源監系統運作時不會當機。
   （3）需有良好的使用性，容易學習及使用。
   （4）需有良好的效率，能快速的提供資訊給使用者。
   （5）精簡空間的使用，不會因持續執行而讓使用空間成長。
   （6）錯誤發生後的回復能力，亦即若系統有非預期的因素而中斷，需能快速回復服務。
   （7）可再次使用性，亦即容易移植，相容性高，且維護容易。
   （8）若新增主機時，可自動抓取新主機之所有 Hardware spec.，並更新至主機資訊檔。
   （9）因主機保留或 benchmark，為避免不相關人員誤選，提供可依群組來設定主機觀看權限，讓相關之群組人員才可看到該主機。

2. End-User 方面
   （1）提供所有主機之各項資源使用資訊（CPU、memory、swap、users…）及軟硬體資訊（OS、kernel version、memory size、CPU Qty…）。
   （2）提供系統資訊即時更新及定時更新功能，定時更新時間為每 60 秒更新一次。
   （3）為方便使用者尋找主機，各項狀態區的欄位（如主機名稱、memory、swap…等）可依升/降冪排列。
   （4）因所提供的主機資訊眾多，使用者可依個人喜好，來決定哪些資訊要顯示。
   （5）為避免畫面呈現過寬或過長，CPU 欄位可依個人喜好，來設定每列要顯示之 CPU 顆數。
   （6）所有主機狀態區的欄位可依個人喜好，來設定其顯示的順序。
   （7）除可使用 GUI 使用外，亦提供此系統之 command-line mode，

可將資源監控系統的資訊，以文字方式，提供給有需要的使用者自行利用，並提供 help 說明。

（8）為簡化使用者操作程序，使用者在決定好要使用的主機後，可直接點畫面中選該主機名稱，將連至該主機並開啟 xterm，即可開始作業。

（9）因主機隨時會保留給其它計劃使用，故提供保留主機提示功能，以減少其它使用者搶占主機資源。

3.2 資源監控系統架構

1. client-server model

在網路上使用應用程式時有一個共同的機制，就是 client-server model。因為計算機以及網路的速度比人類的反應速度快的多，在二台機器間如果想要建立聯繫，在沒有一方處於等待的情況下，單純的靠剛好雙方一起連結的機率進乎於零，所以必需有一方在啟動系統時便處於等待狀態，等待其它主機發出需求，並在接到需求後給予回應，這便是 client-server model，而本資源監控系統便是採用此架構，如圖 3.1 所示。

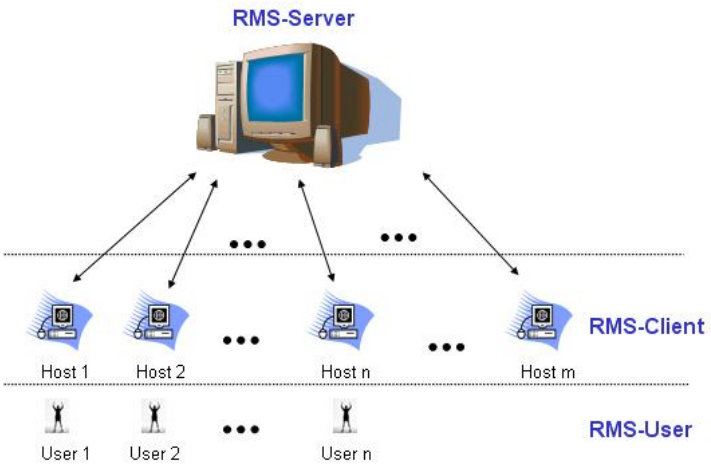


圖 3.1 資源監控系統架構圖

提出需求的主機為 client 端，亦即在本系統裡，所有主機皆為 RMS-client，數量為 m，而回應需求的主機則為 RMS-server，只有一台，而有使用者（不論幾人）執行本資源監控系統 GUI 程式的主機，稱為 RMS-user，數量為 n，則 n <= m。

2. 建置環境

在執行 EDA 軟體過程中，需長時間執行，甚至有些少數 job 執行時間可達數星期之久，執行途中若遇當機，則會影響設計週期，故業界在執行 EDA 軟體時，多採用在較穩定的 UNIX 系統環境上作業。本建置環境也以 UNIX 環境為主。

本資源監控系統網路環境如圖 3.2，所有主機都透過 ethernet 網路連接，中間並無防火牆阻擋，所有主機皆為 UNIX 作業系統。

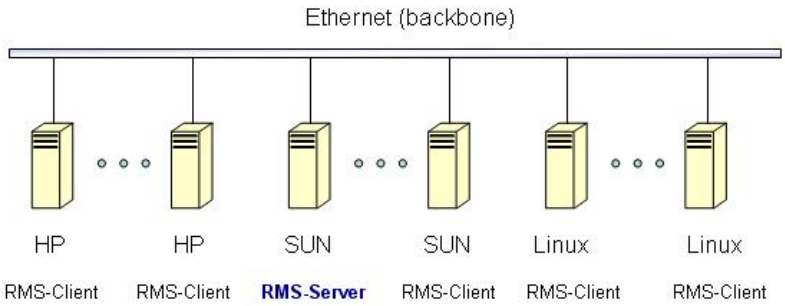


圖 3.2 網路環境結構圖

3. 程式結構

本資源監控系統程式建構於 UNIX 環境之上，其結構圖如圖 3.3，程式語言採用 perl 搭配 TK 撰寫而成，程式執行中，我們只讓主機資訊檔（registry table）更新儲存，其餘所有資訊皆存於 RMS-server 的記憶體中。這麼做是因為本資源監控系統定義 spec. 時，其中更新主機資源使用資訊須在 60 秒內。為了達成目標，在資料儲存時，要儘量避免開關檔的動作。如此一來，即使 RMS-client 數量愈來愈多，也不會使儲存設備的 loading 增加，導致執行速度降低。另外，目前傳遞資訊以 SNMP 或 socket 方式為主，經過測試，在安裝 SNMP package 之後，有些作業系統會有問題，甚至造成無法正常開機，所以我們採用 socket 方式 [11]來傳遞資訊。

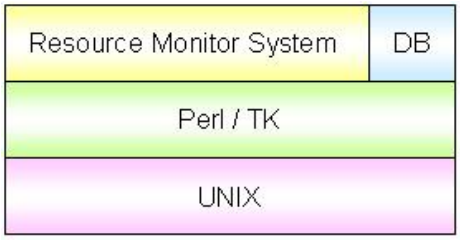


圖 3.3 資源監控系統程式結構圖

3.3 使用者介面

為了簡化 end-user 使用上的複雜度，達到容易掌控所有主機系統資源狀態，操作介面的設計上尤為重要。我們的工具介面具有下述特點

1. 提供一個友善的管理介面，讓使用者一目瞭然，容易上手。
2. 盡量減少或消除一些重複的步驟，以簡化管理。
3. 讓管理介面具彈性能力，可依個人喜好調整及設定顯示結果。
4. 管理系統可隨時使用，並可長時間提供服務，不需另花時間管理。

這個工具介面由一個二維的目錄（群組，工具）來決定要顯示的訊息。

在群組這一部分，它包含六個群組（dms、down、mis、queue、rd、reserve），每一個主群組會包括一系列基本次群組，依作業系統種類（Solaris、HP、redhat）及 memory 大小（24G 以下、32G、64G 以上）來區分。

而工具方面，本系統包括一組各自獨立的工具（訊息欄、現在時間、View、adminfo、refer、reload、資訊更新倒數時間、Exit）。

我們在螢幕顯示的訊息如圖 3.4 所示

圖 3.4 資源監控系統管理介面圖

1. 工具區：置於第一列，提供一些資訊及設定，總共八個項目，說明如下：

（1）訊息欄：當游標點選群組或移至主機狀態欄之欄頭時，會顯示簡短提示。
（2）現在時間。
（3）View：可設定主機的哪些資訊要顯示，目前default顯示6個欄位值，包含主機名稱、各CPU使用狀態、目前使用者人數、每5, 10, 15分鐘之load average值、memory之使用狀態（used/available)、保留主機給計劃使用（project name/deadline)。
（4）adminfo：顯示各project的開始日期及結束日期。
（5）refer：包含四項，第一項為linux作業系統版本及EDA軟體之間，搭配的測試結果，第二項為linux作業系統版本及公司開發的應用程式，其搭配的測試結果，第三項為本系統功能及操作說明，第四項為mark的簡寫符號說明。
（6）reload：立即更新主機狀態資訊。
（7）資訊更新倒數時間：資源監控系統每60秒鐘主動更新資訊一次，這裡顯示更新資訊之倒數秒數。
（8）exit：離開程式。

2. 群組區：置於左邊的方塊（欄）中。我們資源監控系統有主群組及次群組，以所有主機功能及規格來分類。分類如下
（1）主群組：以功能性分組，可分為
① dms：給非RD人員使用，例如layout人員。
② down：已當機或無法提供服務的主機。
③ mis：給mis人員使用，可benchmark主機或提供某些專門服務。
④ queue：專門跑load sharing的主機，例如LSF或SGE。
⑤ rd：給rd人員使用。
⑥ reserve：保留給特殊需求的人員使用，例如趕tape-out。

（2）次群組：以作業系統、Memory或特殊用途來分類，可分為
① 作業系統：HP、Solaris、Linux（redhat)。
② memory：24GB以下、32GB、64GB及以上。
③ 特殊用途：專門給某一個project或提供某種服務之用，例如版本控制服務。

當主群組被點到之後，其次群組即被展開，如圖3.3所示，dms被點開之後，其內部8個次群組顯示出來。當使用者點到一次群組時，該次群組的label就變成黃色，同時右邊主機狀態區方塊會顯

示此次群組的訊息。

3. 主機狀態區：右邊的方塊（紅色線包圍）區，顯示根據 view 選擇的主機資源使用狀態。

圖 3.4 的例子是我們在 view 設定的 default 選項。本區第一列主要顯示資訊的名稱。圖 3.4 顯示(HOST、CPU、USERS、LOAD AVERAGE、MEM、RESERVE)，而第二列以後，則依欄位顯示在該次群組之各主機的狀態值。

這個顯示資訊有三個較特別的地方

（1）為了方便使用者尋找想要的主機，我們提供了讓所有欄位都可進行排序的功能。其模式如下，在第一列的欄位點選時，資源監控系統將以此欄位為第一 key 值，而以上次所選的欄位為第二 key 值進行排序，重複點選時，則反覆進行升/降冪的排序。

（2）在使用者找到可用主機後，為簡化使用者的操作程序，有提供自動連線功能；亦即在資源監控系統 HOST 欄位中，使用者點選欄位下方之主機名稱後，可連線至此該主機並開啟 xterm，即可開始作業。

（3）因現在主機的 CPU 顆數不一，由 1 顆 CPU 到 8 顆 CPU 皆有，為了讓顯示畫面不會因 CPU 欄位顯示過寬或過長，使用者在第一列的 CPU 欄位點選“滑鼠右鍵”時，螢幕會出現一小視窗以調整要顯示之 CPU 列數，亦即多少顆為一列顯示，超過則為下一列。

因此，資源監控系統操作介面非常直覺，容易使用，即使是新手也沒有適應問題，另外，我們還提供 help 功能，裡面有更詳細的使用說明給使用者。

3.4 資源監控系統設計方式

為了讓資源監控系統能從所有主機上收集資訊，並將這些資訊傳遞給使用者，我們根據（server/client/user）分別設定了其相對應的角色來負責各自對應的任務，下面依序說明其功能

1. RMS-Server：此主機會執行三支獨立運作的程式

（1）RMS-Server Process：

在資源監控系統開始時，先執行 RMS-Server Process，此 process 會呼叫另外二支 process（Client_Invoke Process 和 Registry_Update Process），並且負責功能有二

① 接收RMS-Client所傳來的現在主機資源使用資訊。

② 若有 RMS-User 發出需求，則此 process 會把所有主機資源使用資訊傳給 RMS-User，如圖 3.5 和 3.6 所示。

（2）Client_Invoke process：

Client_Invoke Process 被啟動後，負責的功能有二

① 會根據主機資訊檔（Registry Table）的資料，在設定的所有主機上啟動 RMS-Client Process。

② 定時向 RMS-Server 要現在正常的主機資源使用資訊，並比對主機資訊檔的所有主機，以找出哪些主機未啟動 RMS-Client Process，並將它重啟。

（3）Registry_Update Process：

Registry_Update Process 被啟動後，負責定時檢查主機資訊檔（Registry Table）是否有更新，若有新增主機，則會半自動抓取新主機之 Hardware Spec.，並更新至主機資訊檔。

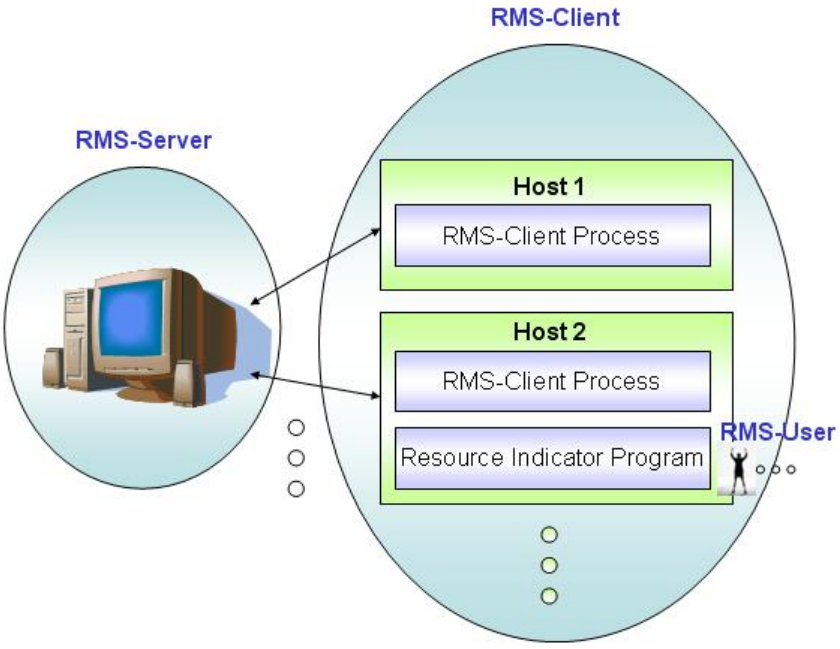


圖 3.5 資源監控系統執行流程圖 1

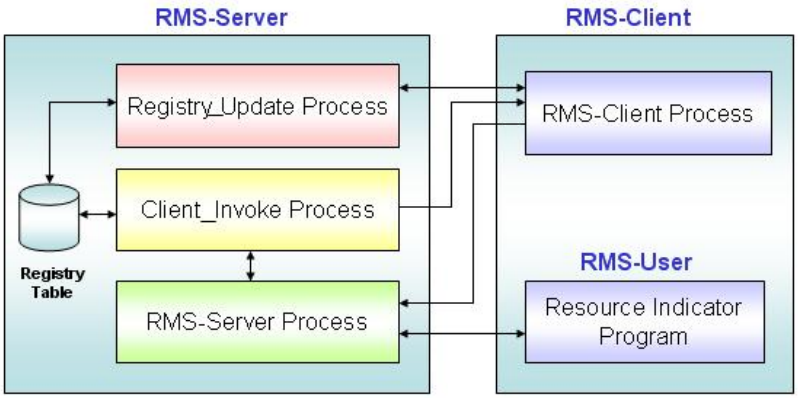


圖 3.6 資源監控系統執行流程圖 2

2. RMS-Client

RMS-Client Process 在被 Client_Invoke Process 喚起後，主要負責定時將該主機的資源使用狀態回報給 RMS-Server，所以每一台主機都會有這樣一支固定的 RMS-Client Process 來負責此事，換句話說，所有的主機皆為 RMS-Client。

3. RMS-User

基本上，所有主機皆為 RMS-Client，而有使用者在某一主機上執行 Resource Indicator Program（RIP）時，則該執行主機則稱為 RMS-User。其負責的功能是將最新的主機資源使用狀態，依使用者的設定，呈現於管理介面上；並定時向 RMS-Server 要求最新的資源使用資訊，更新於它的使用者畫面上，如圖 3.4。

3.5 程式流程圖

1. RMS-Server 將包含三支 process，（1）RMS-Server Process （2）Registry_Update Process （3）Client_Invoke Process，而要啟動整個資源監控系統的源頭 RMS-Server Process，我們在 RMS-Server 主機上設定一個 cron job，並且每 5 分鐘執行一次 RMS-Server Process，以下依序說明其流程圖

（1）RMS-Server Process 流程圖：負責處理 RMS-Client / RMS-User 的需求

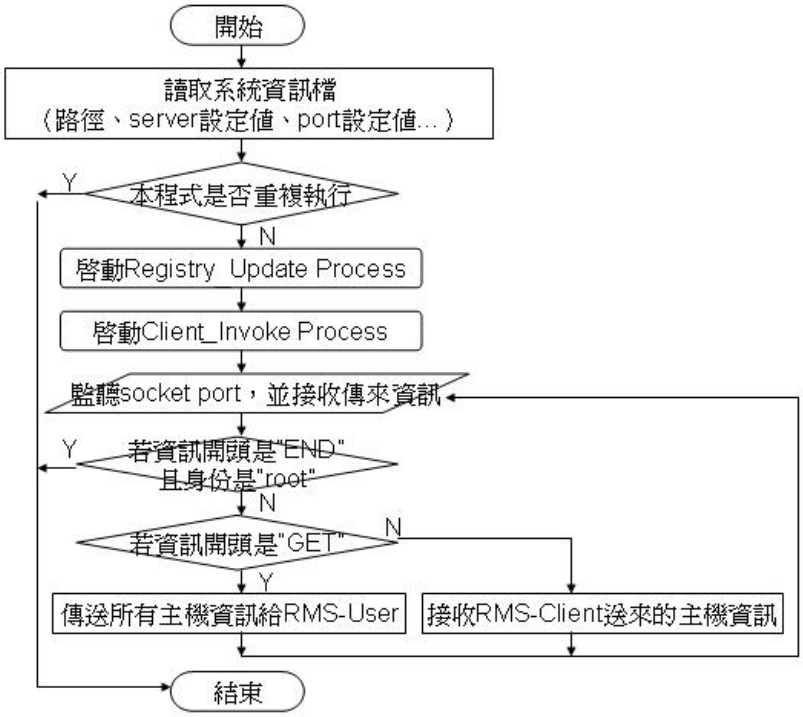


圖 3.7 RMS-Server Process 流程圖

（2）Registry_Update Process 流程圖：負責抓取新增主機之 H/W Spec.

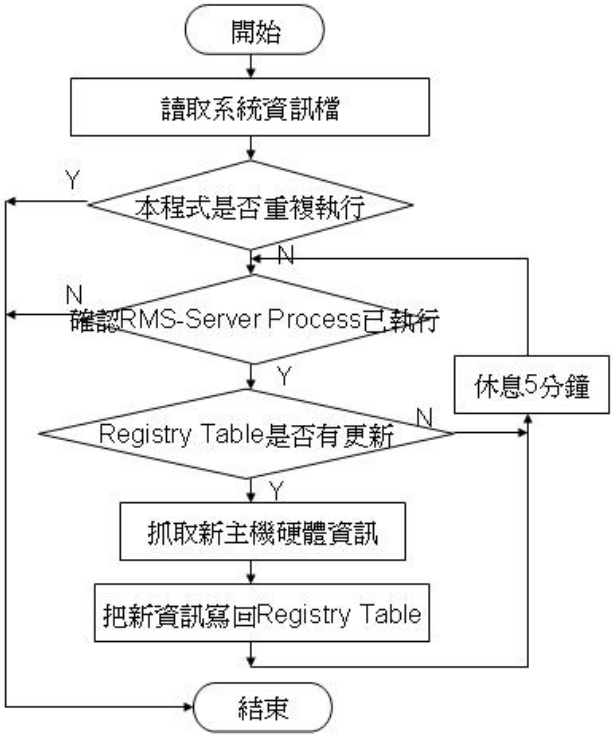


圖 3.8 Registry_Update Process 流程圖

（3） Client_Invoke Process 流程圖：負責喚起 RMS-Client Process

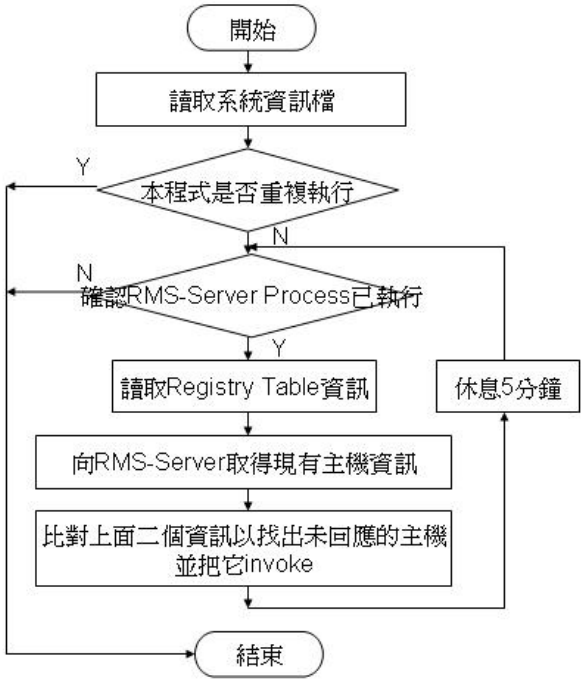


圖 3.9 Client_Invoke Process 流程圖

2. RMS-Client 只有一支 RMS-Client Process，主要負責把主機資源使用狀態定時回報給 RMS-Server，其流程圖如下

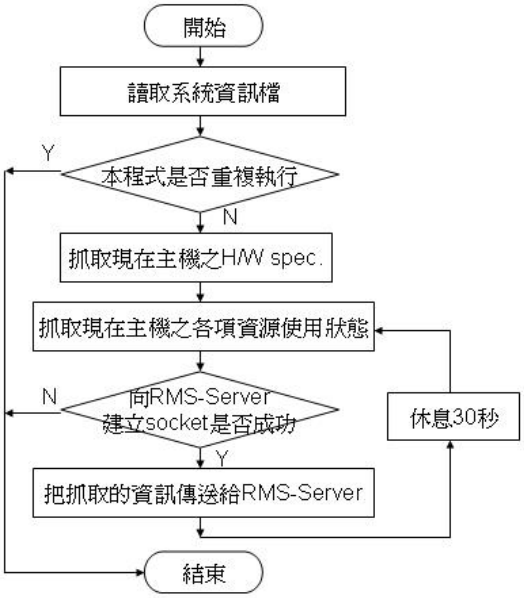


圖 3.10 RMS-Client Process 流程圖

3. RMS-User 則有一支 Resource_Indicator Program（RIP），功能為以 GUI 或 command-line mode 方式，將資訊呈現給使用者，其流程圖如下：其中以引數來判斷的呈現方式是以 command-line mode 或是 GUI 方式

圖 3.11 RIP 流程圖

第四章、實作

4.1 開發環境需求

1. 作業系統：CentOS 4.7
2. 軟體：perl 5.8.8
3. 安裝工具：

   Tk-804.028

   Tk-MK-0.17

   rxvt-2.6.4

4.2 實作結果

根據第三章所定義的規格，我們成功實作出本資源監控系統，RMS-Server 啟動 Registry_Update Process 及 Client_Invoke Process，並處理 RMS-Client 和 RMS-User 的需求；RMS-Client 則把主機現在的資源使用狀態，定時回報給 RMS-Server；最後，RMS-User 則可執行 Resource Indicator Program（RIP）來檢視所有主機的資源使用狀態；定義的各角色都按照其所設定的功能來運作。

1. 群組區

   資源監控系統群組區包含主群組及次群組，主群組以功能性來分組，分別為 dms、down、mis、queue、rd、reserve 等；次群組以作業系統、Memory（24GB 以下、32GB、64GB 及以上）或特殊用途來分類，結果如圖 4.1 所示

圖 4.1 資源監控系統群組圖

2. 工具區

（1）View 功能

如圖 4.2 上方黃色區塊所示，點選此「View」按鈕可設定「主機狀態區」的欄位，當按下（選上）「View」按鈕時，其設定視窗如圖 4.3 所示，左邊藍色區域為所有可顯示的項目，右邊紅色區域則是已選定的項目，並且可調整顯示的順序。以圖 4.3 為例，其結果如圖 4.2。

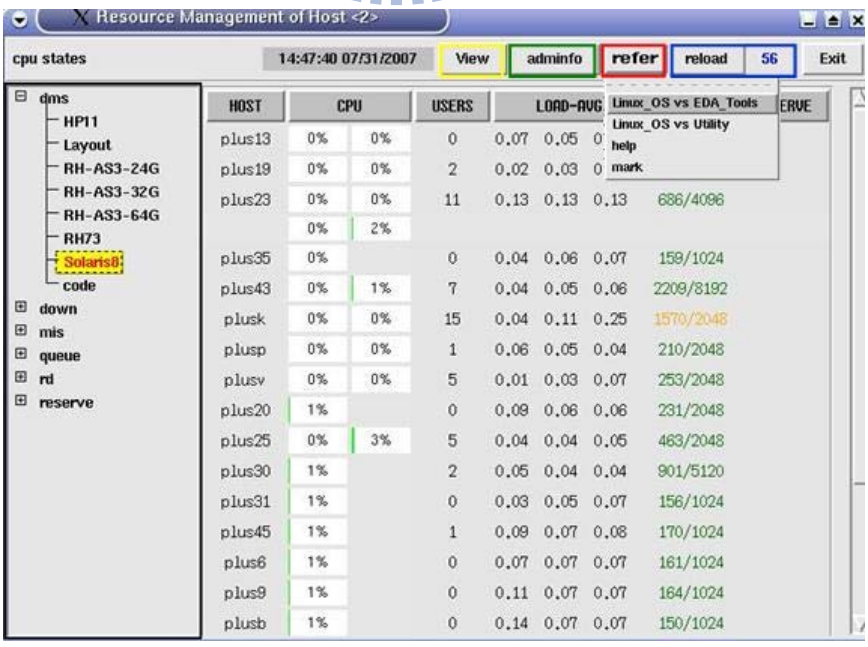


圖 4.2 資源監控系統工具區圖

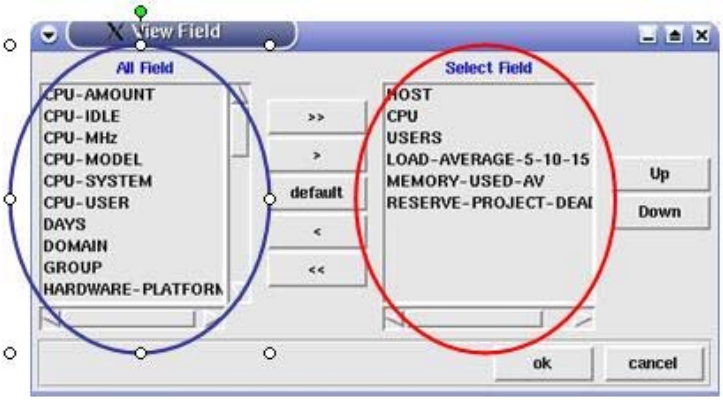


圖 4.3 資源監控系統 View 設定圖

（2）adminfo 功能

應 end-user 的要求，我們提供 adminfo 功能。如圖 4.2 之綠色區塊所示之「adminfo」按鈕，當 end-user 按下「adminfo」按鈕時，此功能可顯示現在執行中的計劃有哪些，其開始和結束的日期等，如圖 4.4 所示

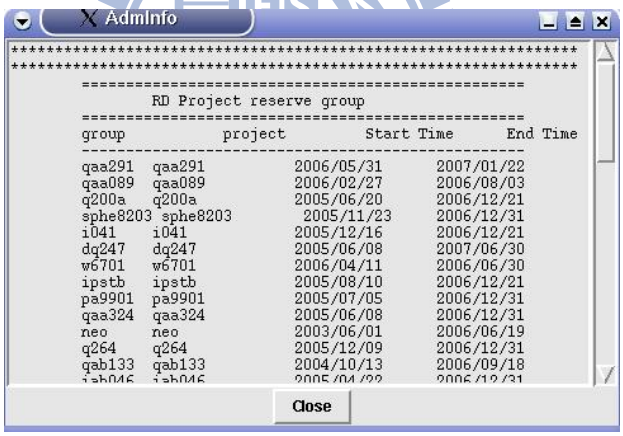


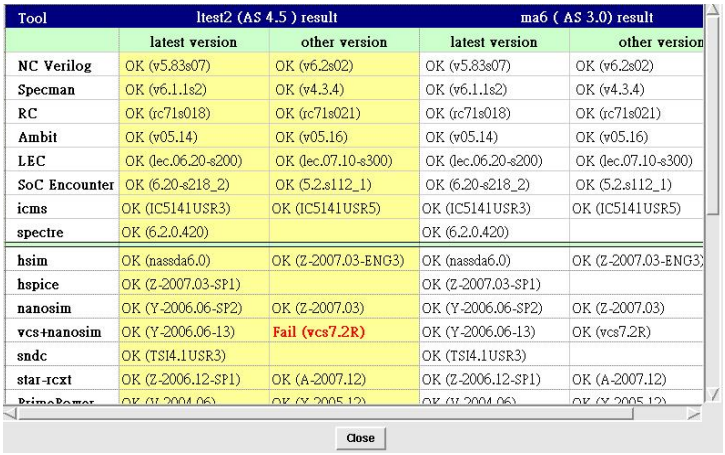
圖 4.4 資源監控系統 adminfo 功能圖

（3）refer 功能

如圖 4.2 紅色區塊所示之「refer」按鈕，當按下「refer」按鈕之後，會出現四個選項，第一項為 linux 作業系統版本及 EDA 軟體間，搭配的測試結果，如圖 4.5，第二項為 linux 作業系統版本及公司開發的應用程式，其搭配的測試結果，如圖 4.6，第三項為本系統功能及操作說明，如圖 4.7，第四項為 mark 的簡寫符號說明，如圖 4.8。

其中第一、二項對 end-user 來說最為重要，因作業環境中有許多種的 O.S，而執行的 EDA 工具不但種類繁多，而且時常更

新，故有許多種版本，所以 O.S 和 EDA 的搭配很重要，若版本不符合，可能執行失敗，徒然浪費開發時間。



Tool	ltest2 (AS 4.5) result		ma6 (AS 3.0) result	
	latest version	other version	latest version	other version
NC Verilog	OK (v5.83s07)	OK (v6.2s02)	OK (v5.83s07)	OK (v6.2s02)
Specman	OK (v6.1.1s2)	OK (v4.3.4)	OK (v6.1.1s2)	OK (v4.3.4)
RC	OK (rc71s018)	OK (rc71s021)	OK (rc71s018)	OK (rc71s021)
Ambit	OK (v05.14)	OK (v05.16)	OK (v05.14)	OK (v05.16)
LEC	OK (lec.06.20-s200)	OK (lec.07.10-s300)	OK (lec.06.20-s200)	OK (lec.07.10-s300)
SoC Encounter	OK (6.20-s218_2)	OK (5.2.s112_1)	OK (6.20-s218_2)	OK (5.2.s112_1)
icms	OK (IC5141USR3)	OK (IC5141USR5)	OK (IC5141USR3)	OK (IC5141USR5)
spectre	OK (6.2.0.420)		OK (6.2.0.420)	
hsim	OK (nassda6.0)	OK (Z-2007.03-ENG3)	OK (nassda6.0)	OK (Z-2007.03-ENG3)
hspice	OK (Z-2007.03-SP1)		OK (Z-2007.03-SP1)	
nanosim	OK (Y-2006.06-SP2)	OK (Z-2007.03)	OK (Y-2006.06-SP2)	OK (Z-2007.03)
vcs+nanosim	OK (Y-2006.06-13)	Fail (vcs7.2R)	OK (Y-2006.06-13)	OK (vcs7.2R)
sndc	OK (TSI4.1USR3)		OK (TSI4.1USR3)	
star-rcxt	OK (Z-2006.12-SP1)	OK (A-2007.12)	OK (Z-2006.12-SP1)	OK (A-2007.12)
PrimePower	OK (Y-2004.06)	OK (Y-2005.12)	OK (Y-2004.06)	OK (Y-2005.12)

Close

圖 4.5 OS 及 EDA 測試結果圖

Program	AS4.5 (ltest2)	AS3.0 (ma6)
vcheck	OK	OK
initapr	Fail	
	(Can't load module Tk::Event, dynamic loading not available in	OK
buildlib	Fail	
	(Can't load module Tk::Event, dynamic loading not available in	OK
dvgui	Fail	
	(Can't load module Tk::Event, dynamic loading not available in	OK
dpiparser	OK	OK

圖 4.6 OS 及 Utility 測試結果圖

○ 功能說明：
1、畫面功能說明：畫面由三大區塊組成
上方：工具區
左方：群組區
右方：主機狀態區
1.1、工具區：由左至右，依序包含訊息欄、現在時間、View、adminfo、help、reload、倒數時間、exit
1.1.1、訊息欄：當游標移至主機狀態欄之欄位時，會顯示簡短說明
1.1.2、現在時間：現在時間
1.1.3、View：可以設定主機的哪些資訊要顯示
A、All Field：可選擇之主機資訊
B、Select Field：目前已選擇之主機資訊
C、" >> " 按鈕：表示All Field的所有選項，全部選擇
D、" > " 按鈕：將All Field中選擇的項目，加入到Select Field
E、" default "：回復成初始設定
F、" < " 按鈕：將Select Field中選擇的項目，取消放回All Field
G、" << " 按鈕：將Select Field中所有項目取消，放至All Field中
H、" Up " 按鈕：將Select Field中選擇的項目，顯示順序往上（前）
I、" Down " 按鈕：將Select Field中選擇的項目，顯示順序往下（後）
1.1.4、adminfo：顯示各計劃的start time、end time
1.1.5、help：功能說明
1.1.6、reload：更新主機狀態資訊
1.1.7、倒數時間：自動更新主機狀態資訊之倒數秒數
1.1.8、exit：離開程式
1.2、群組區：包含主群組及次群組
1.2.1、點這主群組：可顯示所有次群組之主機狀態
1.2.2、點這次群組：可顯示此次群組之主機狀態
1.3、主機狀態區：依照所選擇的項目來顯示資訊
1.3.1、欄位：
A、點這欄位時，將以此項欄為鍵值進行排序，而以上次欄位為排序的第二鍵值
B、若欄位重複點選時，則進行升（降）冪的排序
C、CPU欄位：點這"滑鼠右鍵"，可以調整要顯示之CPU行數
1.3.2、主機資訊列：
A、點這Host時，會連至此主機，並開啓xterm
B、其它欄位則顯示此主機相關資訊
2、command-line mode
2.1、指令說明：請執行hrm -h，會有指令使用說明
2.2、執行方式：hrm [primary-group] [sub-group]
例：hrm dms Layout
=> 將會顯示dms之Layout次群組之主機狀態
例：hrm dms
=> 會顯示dms之所有次群組之主機狀態
3、個人設定檔
3.1、個人設定檔位置：個人帳號下之.hrm_帳號.ini
例：/home/seanwang/.hrm_seanwang.ini
3.2、設定檔內容：包含CPU欄位數，顯示的欄位，HP、Linux、SUN之xterm的設定，排序的鍵值

圖 4.7 資源監控系統 help 說明圖

圖 4.8 資源監控系統 mark 說明圖

（4）即時和定時更新

如圖 4.2 藍色區域所示，資源監控系統提供即時和定時的資訊更新，左邊的「reload」鈕為即時更新，點選時，資源監控系統會立即和 RMS-Server 要最新的主機資源使用資訊，並顯示於螢幕上，而右邊的秒數倒數顯示，則為定時更新，資源監控系統會定時每 60 秒更新一次。

3. 主機狀態區

（1）排序功能

如圖 4.9 藍色區域所示，當 end-user 點選此區域任一欄頭時，則會依所選擇的欄位為 key 值，進行排序；若重覆點選時，則反覆進行升/降冪排序。

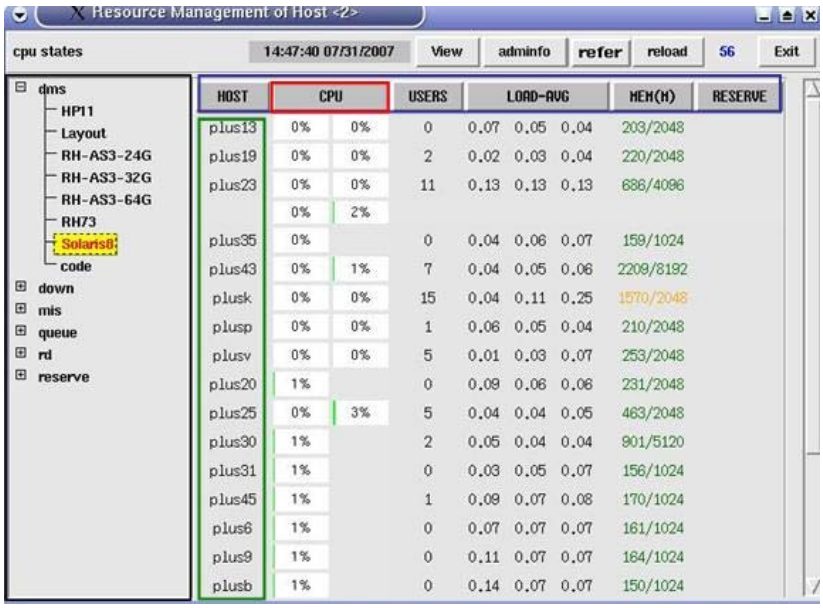


圖 4.9 資源監控系統主機狀態區圖

（2）CPU 顆數顯示設定

如圖 4.9 紅色區域所示，當 end-user 在 CPU 欄頭按滑鼠右鍵時，其設定視窗如圖 4.10，此功能可依個人喜好設定每列要顯示的 CPU 顆數，以避免畫面顯示過寬或過長。

圖 4.10 資源監控系統 CPU 顆數設定圖

（3）自動登入並開啟視窗

如圖 4.9 綠色區域所示，當 end-user 在選定的「主機名稱」點選時，可連至該主機並開啟 xtem 視窗來進行作業，此功能為簡化使用者操作程序。

4.3 資源監控系統的品質

1. Usability

一般而言 usability 是根據 end-user 的使用經驗來決定好壞。這裡我們根據 1）Easy learning 2）Memorability 3）level of automations 來說明本系統較高的 usability。

（1）Easy learning

本資源監控系統非常直覺且友善的，所以不需要特別經過訓練，也不需要非常有經驗的使用者才可執行，即使是新手，從未使用過類似的軟體，也能立即上手。

另外，資源監控系統還提供 help 使用說明，若要詳細了解所有功能則可點選此鈕，從開始到完全了解本系統，只需五分鐘即可。

（2）Memorability

因使用者需掌握所有主機狀態，以方便作業，在執行此資源監控系統時，只需一個指令，操作簡單，容易記，重複使用率非常高，不需重新學習；使用此資源監控系統的 end-user，習

慣固定開啟它，並放置桌面，不再關閉。

(3) level of automations

在 system administrator 方面，此資源監控系統只需執行一個指令即可啟動，之後便可提供服務，完全自動化收集並更新資訊。而在新增主機時，也只需提供基本名稱及群組設定，即可自動抓取其它資訊，管理上非常方便。

在 end-user 方面，只需一個指令即可開啟，主機資訊將會自動更新，完全自動化，除了在設定個人喜好外，其餘均不需動手，操作上非常簡單，不需進行教育訓練。

2. 容錯

此項功能在系統遇到非預期中執行失敗的時候，能儘快的或不中斷的回復並提供服務，以下將對錯誤回復能力加以說明

(1) RMS-Server

為避免此程式被誤砍而沒有提供服務，故在 RMS-Server 的主機上，會設定一個 cron job，此 job 每五分鐘會執行一次 RMS-Server 的程式，而執行此程式時，會先檢查是否重複執行，若未執行，則會啟動它，反之，則跳過。

(2) RMS-Client

在 RMS-Client 檢查方面，則由 Client_Invoke Process 來負責檢查 RMS-Client Process 是否已執行；而 Client_Invoke Process 在 RMS-Server Process 啟動時，便立即被喚起，以專門檢查所有 RMS-Client 的 process 是否已執行，它每五分鐘執行一次檢查， 因此，若 RMS-Client 的 process 被誤砍，它會 invoke RMS-Client 的 process。

(3) RMS-User

RMS-User 有一支獨立的應用程式 (RIP)，功能主要為監控所有主機的資源使用狀態，若程式呈僵屍狀態，並不影響他人或整個資源監控系統的運作，只需重新下個指令再開啟即可，而新開啟的程式一開始便會檢查舊有程式是否存在，若有，則會一併刪除，避免多重開啟程式。

(4) dead lock

在資源監控系統所用到的資源方面，共用的資源只有儲存設備；因原先在設計時，就刻意避免開關檔的動作，只有主機資訊檔需要儲存，而此部分也只有RMS-Server會用到，所以並無資源搶占情形，也不會有dead lock現象。

（5）當機

資源監控系統在運作時，會執行一些程式（例RMS-Server Process、RMS-Client Process、Client_Invoke Process、Registry_Update Process…等），而這些process都有各自的檢查機制，若程式被誤砍，會自動重啟，所以不會影響程式運作。

此外，這些程式執行時所需的memory很小，不會隨時間而增加memory使用量，可長時間服務。因係獨立系統，所以不會妨礙機器正常運作。而在停機維護過後，重新啟動資源監控系統時，也非常簡單，只需執行一個指令即可恢復。

3. Efficiency

一個系統的 efficiency 評估應該含有時間和空間，亦即包含系統執行速度和使用的空間，以下依序說明

（1）processing time

執行時間主要 dependent on 主機速度，經過測試，在第一次開啟資源監控系統時，在約 200 台 linux 主機上，CPU 速度在 2~3GHz 的測試結果，約花 1.8 秒的時間。而在以後的定時資訊更新時間裡，處理的時間更少，約花 0.5 秒即可，主要在資訊傳遞及圖形重繪，其 CPU loading 非常輕，故整體執行效能非常好。

（2）Space

對於本資源監控系統的空間需求說明如下

① 資源監控系統程式

本資源監控系統的所有程式及相關設定檔其佔用的空間在 1MB 內，只要設定 auto-mount，讓所有主機皆能讀取執行此程式即可。

② log 儲存

因採用 socket 的方式，讓資源監控系統不存放任何 log，所有資訊均儲存於記憶體，所以不需硬體儲存空間，

而執行時所需的 memory，也只有幾 mb 而已。

4. Reusability

對於資源監控系統的 resuability 包含以下說明

（1）Portability

資源監控系統非常容易移植，只需將系統相關程式複製到新 Unix/Linux 系統上，並設定主機資訊檔及主機間的執行權限，即可開始運作，因資源監控系統與其它程式間並無相依性問題，可避免程式相互干擾，所以方便管理及維護。

（2）Compatibility

資源監控系統可安裝在任何 SUN、HP、Linux 主機上，相容性高，但並不適用於 windows、MacOs 等其它作業系統。使用上，建議 SUN 使用 solaris 5.8 以上，HP 則為 11.x 以上，而 Linux 則使用 kernel 2.6 以上較佳。

（3）Extensibility

因本資源監控系統之程式主體架構明確，程式內容均模組化，並且精簡。若欲新增功能，或者修改使用介面都非常容易，具彈性，本資源監控系統維護上不複雜，即使交由其它管理者維護也不困難。

第五章、結論及未來方向

5.1 結論

在本文中，我們已詳細說明並且圖示此資源監控系統的設計架構及運作方法，透過角色的定義，讓各個 process 負責各自的功能，來掌控所有運算系統的資源，再把資訊提供給使用者。依照規格定義，我們實作出一功能良好，且效能佳的資源監控系統，RMS-Client 可定時的回報主機資源使用狀態給 RMS-Server，而良好的使用者介面，可定時/即時地從 RMS-Server 取得所有主機資源使用狀態，讓即使是職場新手也能快速上手，一覽無遺的掌控所有資源，也可根據工作所需，快速的找到可利用的主機來作業，善用主機資源，使用上沒有限制。

在本次實作中，為了讓所有 end-user 都能使用本系統，我們也和各方面的使用者進行討論，包含 designer、layout 人員和其它使用人員（如 mis、客戶），功能上盡可能達到各方所需，故群組的規劃和資訊結果的呈現等，都是討論後所得的結果，所以，在系統完成上線後，確實解決使用者尋找主機的問題，提昇主機使用率，最終得到大家的認同及贊賞。

此系統為獨立運作，若再搭配原有之 LSF 或 SUN Grid Engine 系統，不但不會浪費之前的建置費用，還可讓主機使用率提昇，達到互補的作用。

5.2 未來方向

目前提供的資源監控系統可清楚顯示所有主機的資源使用狀態，有效地提昇主機的使用率，但隨著時間的過去，作業環境及使用習慣的改變，有以下五點可進行改善，使用介面更為友善

1. 使用 multicast 方法

當初在設計建置初期，考慮當時的使用者習慣，因 SUN 工作站性能較穩定，大部分使用者只在 SUN 主機登入，故會在此主機上執行此資源監控系統，而其它主機執行的可能性及次數較少，因此資源監控系統採用 socket 方式來做資訊的交換；但時過境遷，隨著 Linux 系統的速度及穩定性大幅增加，及資源監控系統的大量使用，現在

主機 loading 已大為分散了，使用者也不在侷限在某些特定系統上登入，所以期望在下一版的管理系統可改用 multicast 的方式來改進，可提高效能。

2. 使用流量分析工具

可考量系統可搭配流量分析工具，如：MRTG，來觀看所有主機之 CPU、Memory 的歷史使用狀態，由此可分析主機使用的 loading，進而評估因業務成長，主機資源何時進入高峰期，以便事前添購運算主機或 memory 較大的主機。然而，要注意的是，因需記錄 log 資訊，會產生開關檔的動作，更可能大幅度的拖慢執行速度，若再加上設計方法不良，甚至可能會來不及在 60 秒內更新的情況發生，且 log 資訊會愈來愈多，則會影響現有執行效率及使用空間，故需加以評估。

3. 新增 filter 功能

目前此資源監控系統已有排序功能，但可再新增 filter 功能，讓使用者可針對想要的條件來尋找主機，例如查詢 CPU 速度在多少 Hz 以上？ Memory 在多少 GB 以上？ kernel 的版本多少？ 要執行某個 EDA tool 時，建議的主機有哪些？ 等諸如此類條件的設定，以快速尋找符合條件之主機。

4. Failover 功能

雖然此資源監控系統在 migration 時，設定相當容易，但仍不俱有 failover 功能，未來可新增 failover 功能，讓資源監控系統在 server 有問題時，可自動的尋找適當的主機來接手，以正常提供服務，讓自動化更為完善。

5. 其它程式語言設計

目前資源監控系統採用 perl 來撰寫，未來可考慮以 Java、web 或 flash 方式來設計，讓使用者介面更方便使用，唯要評估其執行效能是否符合預期。

參 考 文 獻

1. NetApplications inc., 〝Top Operating System Share Trend, 〞 NetApplications, Market Share Report, August 2009.
2. M.J. Litzkow, M. Livny and M. W. Mutka, “Codor – A Hunter of Idle Machines, 〞 in *proceedings of 8th International Conference on Distributed Computing Systems*, pp. 104-111, 1988.
3. Dave Field, Deron Johnson, Don Mize, Robert Stober, “Scheduling to Overcome the Multi-Core Memory Bandwidth Bottleneck, 〞 *HP & Platform Whitepaper*, November 2007.
4. Yongwei Wu, Yulai Yuan, Guangwen Yang, Weimin Zheng, “Load prediction using hybrid model for computational grid, 〞 *Proceedings of the 8th IEEE/ACM International Conference on Grid Computing*, September 2007.
5. Kinshuk Govil, Dan Teodosiu, Yongqiang Huang, Mendel Rosenblum, “Cellular disco: resource management using virtual clusters on shared-memory multiprocessors, 〞 *Transactions on Computer Systems (TOCS) , Volume 18 Issue 3*, August 2000.
6. Matthieu Lemerre, Vincent David, Guy Vidal-Naquet, “A communication mechanism for resource isolation, 〞 *Proceedings of the Second Workshop on Isolation and Integration in Embedded Systems*, March 2009.
7. Umit Rencuzogullari, Sandhya Dwardadas, “Dynamic adaptation to available resources for parallel computing in an autonomous network of workstations, 〞 *Proceedings of the eighth ACM SIGPLAN symposium on Principles and practices of parallel programming*, July 2001.
8. Eitan Frachtenberg, Fabrizio Petrini, Juan Fernandez, Scott Pakin, Salvador Coll, “STORM: lightning-fast resource management, 〞 *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, November 2002.
9. 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*, June 2005.
10. Timothy Wood, Ludmila Cherkasova, Kivanc Ozonat, Prashant Shenoy, “Profiling and modeling resource usage of virtualized applications, 〞 *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, December 2008.
11. Hideki Eiraku, Yasushi Shinjo, Calton Pu, Younggyun Koh, Kazuhiko Kato, “Fast networking with socket-outsourcing in hosted virtual machine environments, 〞 *Proceedings of the 2009 ACM symposium on Applied Computing*, March 2009.

12. Fernando Rodríguez, Felix Freitag, Leandro Navarro, “On the use of intelligent local resource management for improved virtualized resource provision: challenges, required features, and an approach,〞 *Proceedings of the 2nd workshop on System-level virtualization for high performance computing*, March 2008.
13. Alexandru V Staicu, Jacek R. Radzikowski, and Nguyen Nguyen , Suboh A. Suboh, “CONCEPTUAL COMPARATIVE STUDY OF JOB MANAGEMENT SYSTEMS, 〞 *A Report for the NSA LUCITE Task Order Productive Use of Distributed Reconfigurable Computing*, February 21, 2001.
14. Kevin J. Barker, Nikos P. Chrisochoides, “An Evaluation of a Framework for the Dynamic Load Balancing of Highly Adaptive and Irregular Parallel Applications, 〞 *Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, November 2003.
15. Erik Putrycz, “Design and implementation of a portable and adaptable load balancing framework, 〞 *Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research*, October 2003.
16. Weiguang Shi, M. H. MacGregor, Pawel Gburzynski, “Load balancing for parallel forwarding, 〞 *Transactions on Networking (TON) , Volume 13 Issue 4*, August 2005.
17. David R. Karger, Matthias Ruhl, “Simple efficient load balancing algorithms for peer-to-peer systems, 〞 *Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures*, June 2004.
18. Junwei Cao, “Self-Organizing Agents for Grid Load Balancing, 〞 *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, November 2004.
19. M.H. Jansen-Vullers, M. Netjes and H.A. Reijers,“Innovation, management & strategy：Business process redesign for effective e-commerce, 〞 Proceedings of the 6th international conference on Electronic commerce, 2004.
20. ANDRZEJ CICHOCKI and MAREK RUSINKIEWICZ, “Providing Transactional Properties for Migrating Workflows, 〞 Mobile Networks and Applications, Volume 9 Issue 5, 2004.
21. Bastin Tony Roy Savarimuthu, Maryam Purvis, Martin Fleurke, “Monitoring and controlling of a multi – agent based workflow system, 〞 Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32, 2004.

22. UNIX International Waterview Corporte Center, “Performance management activities within UNIX international, 〞 SIGMETRICS Performance Evaluation Review , Volume 21 Issue 2, December 1993
23. Donna Spencer, “usability & information architecture 〞, Step Two DESIGNS, 2004/11.
24. Marc Clifton, “What Is A Framework 〞 *THE CODE PROJECT, Design and Architecture*, 3 Nov 2003.
25. Sun microsystems, “Grid Engine Portal, 〞 *Sun microsystems, Whitepaper Grid Engine Portal*, 21 Jul 2007.
26. Sun Microsystems. Sun Grid Engine. http://www.sun.com/software/sge/.
27. Platform Computing. Platform Load Sharing Facility（LSF）. http://www.platform.com/products/platform-lsf/.