台灣二〇〇二年國際科學展覽會

科　　別：電腦科學

作品名稱：Parallelize it! 運算分享與系統自我校調

得獎獎項：電腦科學科第一名
　　紐西蘭二〇〇二年科技展覽會正選代表

學　　校：臺北市立建國高級中學

作　　者：莊偉赳　張津愷

English Abstract

The research is about the optimal on parallel processing. Through boot disk – which will automatically finish booting configuration, .it is efficient and quick to build high performance PC clusters.

The advantage of parallel computing could be applied to massive image processing. By sharing processing and breaking huge processing load into lots of pieces, we could get more efficient result. It is also possible to optimal parallel system through some special means such as dynamic configuration.

Through the means, the system could distribute work loading itself. It could also adjust itself to get the highest performance and the most stable environment.

中文摘要

本研究之目的在於探討平行處理中的計算資源的最佳化，透過自動完成開機設定的 Boot Disk 來有效快速建製出高效率的 PC Clusters 環境，並透過動態配置與類神經網路的校調，使整體叢集的運算能自動調整至最佳化。

平行處理優勢，可以應用在耗費極大量的運算資源的影像處理上。透過運算資源分享，可以以很高的效率將極為龐大的運算工作分散成許多較小的程序，使影像處理速度加快。經由平行演算法及實際應用的調整，可對已成形之平行系統作效能上的加強。

使用類神經網路的方式訓練，使其系統能夠自我分配運算工作量，且隨著各平行化程式與各節點的不同，能自我校調至最佳化，達到高效率且穩定的運算環境。

本研究透過高效率且能自我調校的運算環境，可用於優化其本身結構，以達到演化出更進一步系統，具有相當大的發展潛力。

一、前言

隨著近代電腦科技的迅速演進，運算上的速度越來越快；而超大型積體電路的製作技術快速發展，使得晶片體積越作越小而運算速度卻大幅提升。目前的個人電腦已經能有效應付於處理一般簡單的運算和基本的資料處理工作，但對於需要極龐大計算工作，例如高能物理、氣象分析、3D 光跡追蹤、流體力學、天文計算和近來當紅的基因解碼等工作方面，單微處理器就顯得十分不足，也因此各種不同的電腦型態例如：分散式系統，SMP(Symmetric Multi-Processors)和 MPP（Massive Parallel Processors）以及 Cluster 被發展出來使用在同時處理大量資料的電腦架構上。又由於 PC Cluster 具有極高的擴充性，極佳的價格性能比，在科學領域中已廣泛使用。

另外一方面，這種運算上的優勢可以應用在一般的影像處理上。影像處理必須耗費極大量的運算資源，透過運算資源分享，可以以很高的效率將極為龐大的運算工作分散成許多較小的程序，使影像處理速度加快。經由平行演算法及實際應用的調整，可以對已經成形的平行系統作效能上的加強。

本研究之目的在於透過自動完成開機設定的 Boot Disk 能有效快速建製出高效率的 PC Clusters 環境，並透過動態配置與類神經網路的校調，使整體叢集的運算能自動調整至最佳化，以達到運算資源分享的目的。

二、研究方法及過程

(一) 建構 PC Clusters 環境

1. Automatic Configure Boot Disk

因建構一個叢集環境需要花相當多的時間，包括每一個叢集節點的系統設定，網路設定等等。為解決此問題，設法製作出能在開機時自動連結到主伺服器並且完成自動設定的 Boot Disk 系統。

2. Network

網路是一般 Cluster 間傳遞訊息的主要方法，因此在研究叢集系統的同時，首先對網路機制作一番了解。

(a) 網路傳輸速度與封包大小之關係：

利用 netperf 程式來測定網路速度，比較傳輸速度與封包大小之關係。

(b) 傳輸模式與資料接收的影響：

以訊息傳送函式庫(Parallel Virtual Machine，PVM)的非阻攔式接收 pvm_nrecv 修改一個已經平行化的程式，測試傳輸模式與資料接收的影響。

3. PVM

平行化 PC Cluster 程式時必須配合 PVM 撰寫。使用 PVM 將原先為單一 CPU 系統設計的程式和資料做適當的切割和分配，即所謂之程式平行化，為可達到運算資源分享的方法之一。

(二) 平行處理

1.平行機制之探討

(a) Amdahl's Law：

在一個固定大小的待處理資料中，若欲透過平行處理加速，一般最簡單的方法就是透過增加計算節點，但在增加節點時，有很多是須要注意的，增加的節點數不一定與增加的效益比成正比，由於在 *Amdahl's Law* 中方程式

$$Sn = W / (\alpha W + (1 - \alpha)(W / n) = n / (1 + (n-1) \alpha) \rightarrow 1/\alpha$$

當 n 趨近於無限大

W 是所有工作量

Sn 是加速量

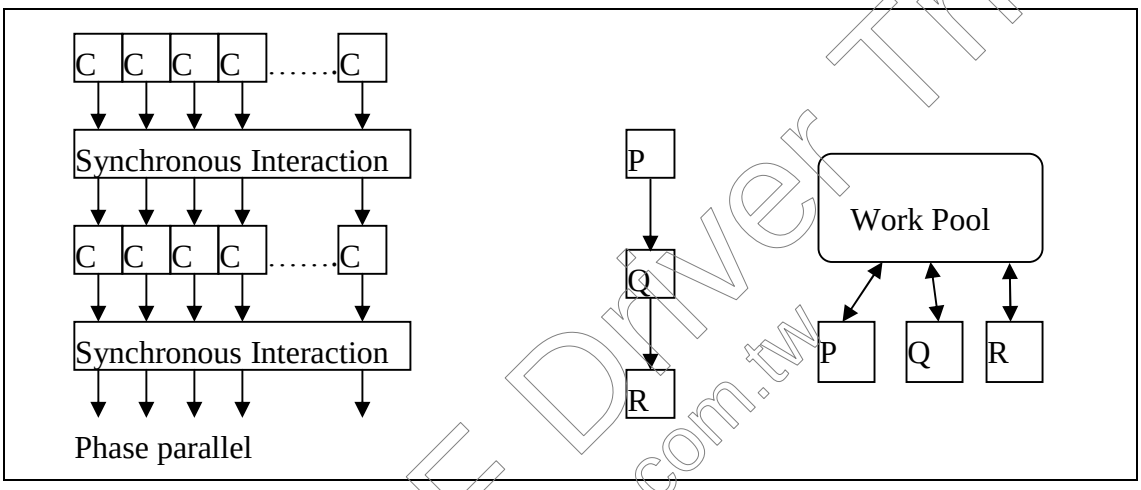
N 是計算節點

α 是不可平行化部分的百分比

在此可以看出，必需在程式撰寫部份下功夫，使得程式中能被平行化的部份增加，進而使得效率提升。在本研究中，我們的焦點是在利用資料的獨立性做平行化。這種模式下，資料中每一部份的相關性並不是很大，所以針對這種特性做處理，將資料分散給計算節點。

(b) 函式處理：

實際運作上，可發現在平行處理裝置上，所得的效能並不能與 CPU 個數的成長量成正比，通常是因為環境和平行化程式本身的問題（例如 CPU 間的訊息傳遞耗費時間以及 I/O 等等），而會有相當的差值產生。假設 cpu 時脈為 p_x，運算資料為 n，則提升效率可視為 $T(p_1,n)/T(p_x,n)=S(p_x,n)$。 圖一為 PC Cluster 的幾何架構。



圖一　PC Cluster 的幾何架構

2. POV (Persistence Of Vision)

POV 是一種需要複雜運算光跡追蹤程式，而其運算與顯示卡無關，完全靠 CPU 來進行運算處理，因此適合用以測試每台電腦的運算校能。

(三) 圖形處理之實際應用

1. 碎形

碎形是一種分數型態維度的圖形，內部結構複雜，不可微分，並且具有放大後產生與原圖類似的特殊性質，例如最常見的例子：孟德博集(Mandelbrat set)，我們用此例子做為下面說明範例。產生孟德博集的公式如下：

$$\left\{ \begin{array}{c} Z_0 = c \\ Z_n = (Z_{n-1})^2 + c \end{array} \right\}$$

將 Z 疊代至 n（n 為一特定數字），並設定一邊界值 b，當 Z 隨著 n 成長而增加時，判斷 Z 是否會在疊代至 n 前超過邊界值，依照其結果在複數平面上產生圖形。

以一個碎形產生器而言，如要將其平行化，首先須了解其基本架構。對於每個叢集中的電腦而言，每一圖素產生的規則都一樣，差異僅僅在於處理的數據不一樣，也就是可以將之視為一個 SIMD 程式。在這裡，可以看出碎形的每一點的值都是相互獨立的，各點的值並不會受到相臨點的影響，於是由此著手，第一階段中，我們的構想是類似排隊的做法，在節點總數 g 中編號 q 的節點會負責處理 q, g+q , 2*g+q…等等的部份，也就是每隔數個固定的數量作處理

2. 影像處理

一般常見的影像處理程式，均需耗費極大量的處理器資源與記憶體資源在特效處理上。在此，我們試著使用分散式儲存以及分散式計算的方法處理圖像的特效。

(四) 利用類神經網路來校調動態配置資料

在進行資料的動態配置部分，隨著每台機器效能不同，所分配到的處理資料量也有所不同。

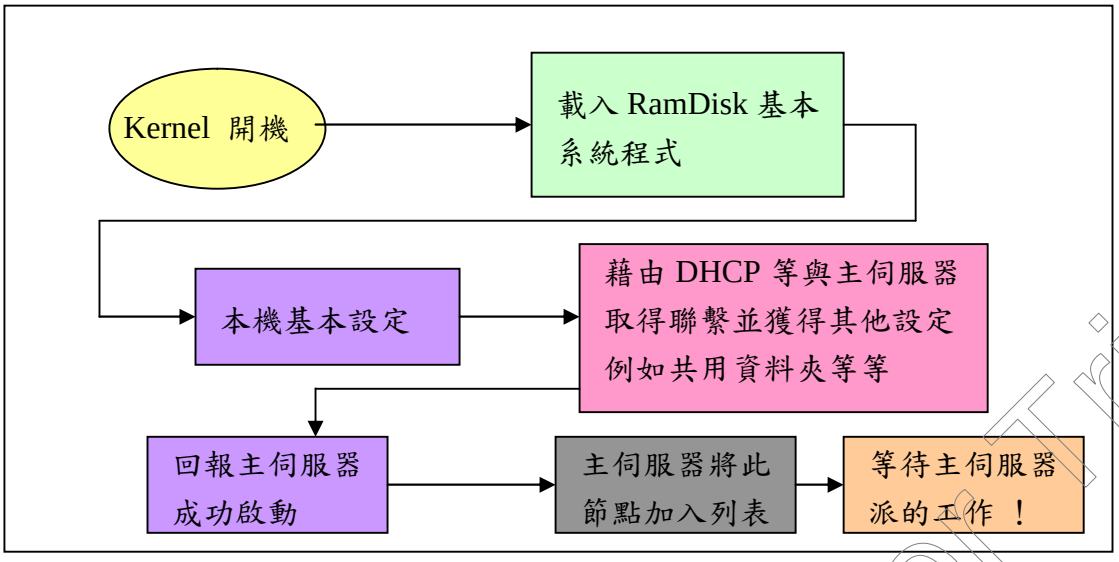
本研究中所利用的動態配置使利用平行化程式中，將一個極為龐大的算式(以微積分計算 π 值，並調整精準度的指標值至十億)分割成許多小的 process，針對不同的主機依據他們的運算速度來進行加權分配 process。

三、研究結果與討論

(一) 建構 PC Clusters 環境

1. Automatic Configure Boot Disk

在自動完成設定磁片系統中，使用小型的開機系統核心並使用 NFS 等程式做到程式及檔案交換，並在系統上裝設平行處理系統必須的程式（例如訊息傳遞函式庫）圖二為自動完成設定磁片的整體開機流程。由於在製作時已將一些系統中無用的設定等移除，所以整體系統效能在一開始即有效提升。

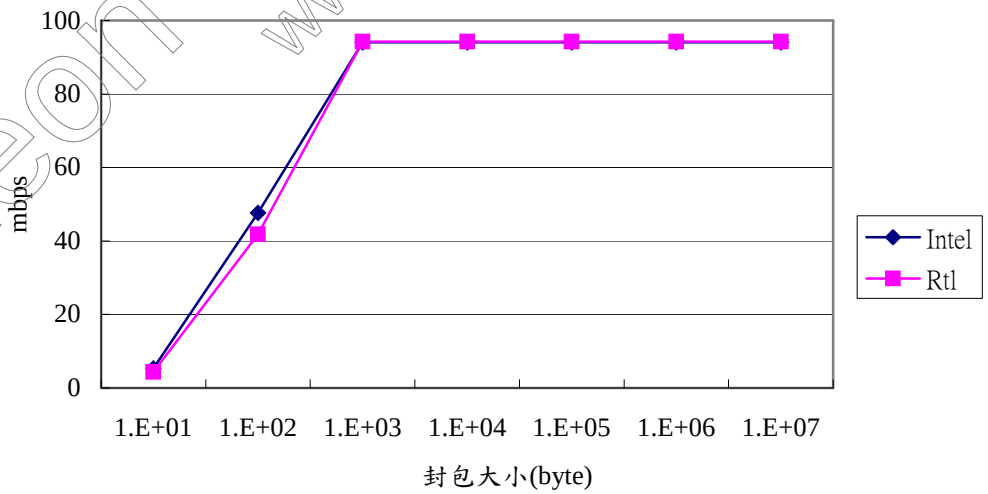


圖二 自動完成設定磁片的整體開機流程

2. Network

(a) 網路傳輸速度與封包大小之關係：

在 TCP/IP 網路中，可以發現 TCP/IP 封包的大小會影響傳輸的速度，也就是說，傳送相同的資料量所需的傳輸時間會隨著封包大小改變而變化，如**圖三**所示。



圖三 網路傳輸速度與封包大小關係圖

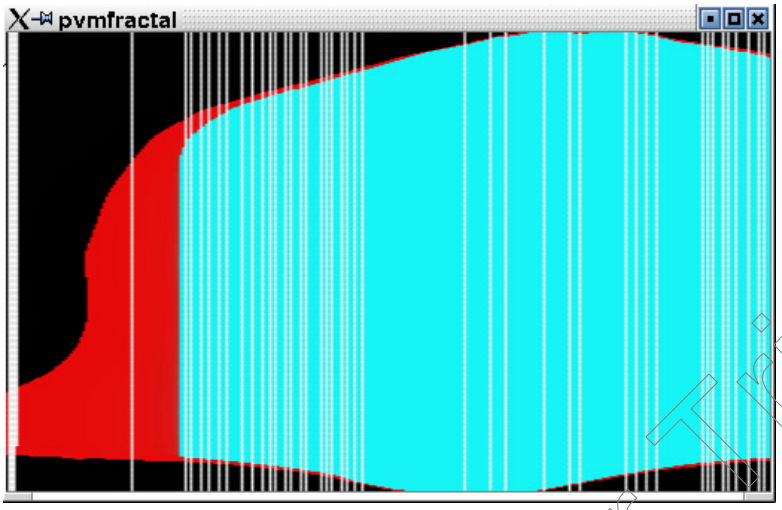
為比較不同網路卡之間的差異，測試中採用了兩廠牌的網路卡(Intel 網路卡和廉價的螃蟹卡)，結果發現，網路的實際傳輸速度與網路卡本身的品質關係不大，但封包大小的改變對於傳輸速度的影響卻很大。由此可見，在網路傳輸時中必須盡可能將一次傳輸的量控制到最大，以使效率應用提升，在**圖三**中所示，在 100mbps 網路環境中，使傳輸封包大小逐步從 10^0 提升為 10^7，網路卡的傳輸效能也隨之成長，但當封包大小成長至約 10^3, (= 1000bytes)時會達到一個接近極限的值，網卡效率的提升至此不再明顯，與 10/100 網路卡的理想最高傳輸量已相當接近。由於以上的原因，再加上考慮高速傳輸資料時同時可能伴隨而生的碰撞與傳輸問題，我們選擇 1000 bytes 的封包作為我們以下研究中對於叢集間傳輸量的重要參考值，我們以下的平行化程式皆採用接近 1000bytes 長度的封包，以提升整體效能。

不僅如此，我們還對 Cluster 的環境做了模擬，對於之後研究中可能會遇到的通信環境，作出測試。在這種模型中，Cluster 中對作為分配資料和整合資料的 Master Server 會產生極大的流量，而 Master 對於各 Slave 的回傳流量則小的不成比例，在這種情形下，封包大小對於流量的限制更加明顯。

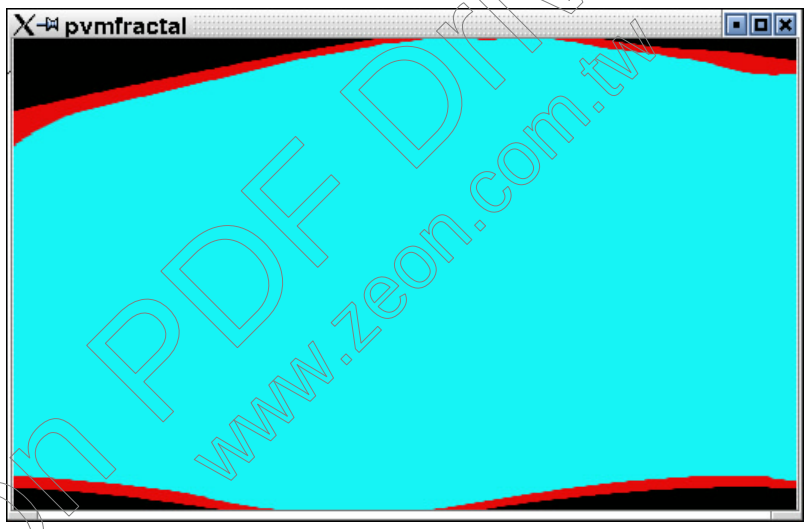
(b) 傳輸模式與資料接收的影響：

以ＰＶＭ的非阻攔式接收 pvm_nrecv 修改一個已經平行化的碎形程式，結果發現 Master 在非阻攔模式接收下會發生很嚴重的問題：

如**圖四**所示，可以發現原本完整的圖像缺了幾條直線，也就是系統無法收到傳回的資料。經檢查結果，發現原因是由於主程式在第一次檢查緩衝記憶體時沒有任何資料傳回，而在下一次檢查時，兩筆（或兩筆以上）的傳回資料同時在主程式返回前到達，由於緩衝記憶體有限，於是後來者會蓋掉前面的，於是就會形成**圖四**的問題造成某些資料的消失。為了克服這個問題，我們必須設計容錯系統，針對因種種網路品質或者不明原因消失的資料，重新讀取一份，調校之後的程式如**圖五**，已不再有此困擾。



圖四 非阻攔模式接收產生的碎形圖形



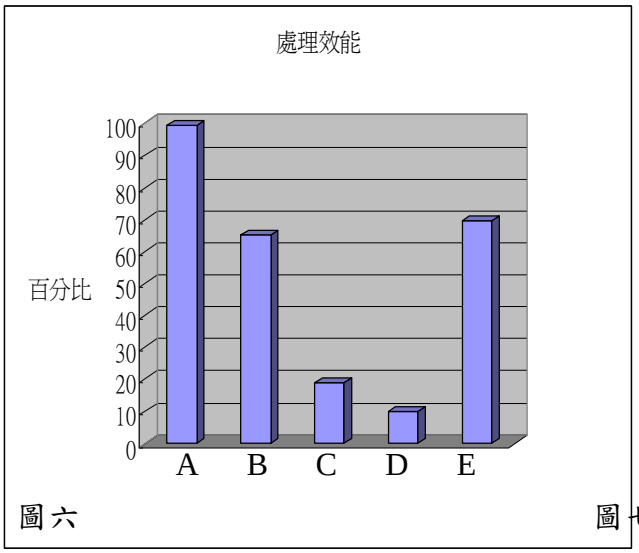
圖五 調校之後產生的碎形圖形

3. PVM

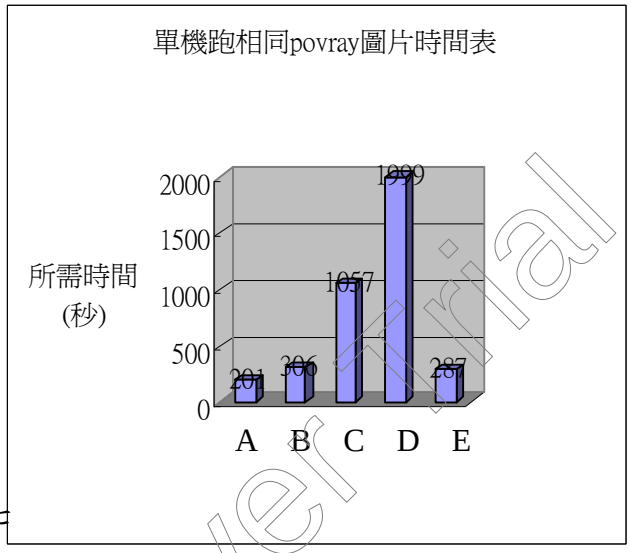
由於 PVM 是使用 TCP/IP 網路作為傳遞訊息的媒介，即使網路速度再怎麼快，仍和 CPU 與匯流排之間的速度有相當大的差異。所以利用 PVM 為 PC Cluster 撰寫平行處理程式時，最大的課題就是在如何盡量減少傳輸次數以及有效利用 TCP 網路。

(二) 平行處理

由 POV 檢測每一台電腦的效能，其結果如圖六和圖七所示。



圖六

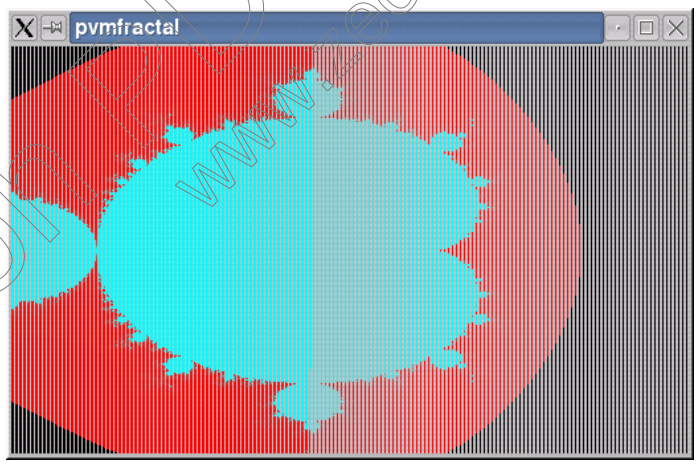


圖七

(三) 圖形處理之實際應用

1. 碎形

在平行化模式之下，所得出來的結果在一個 PC 環境較不單純的情形下，會因為各節點所能負載的計算量不同造成部份節點有些快或者有些慢，所以顯現出這種方式並不好，如圖八所示。



較慢的節點繪圖

此一節點已完成工作

圖八 使用三個節點計算碎形圖形之結果

2. 繪圖系統

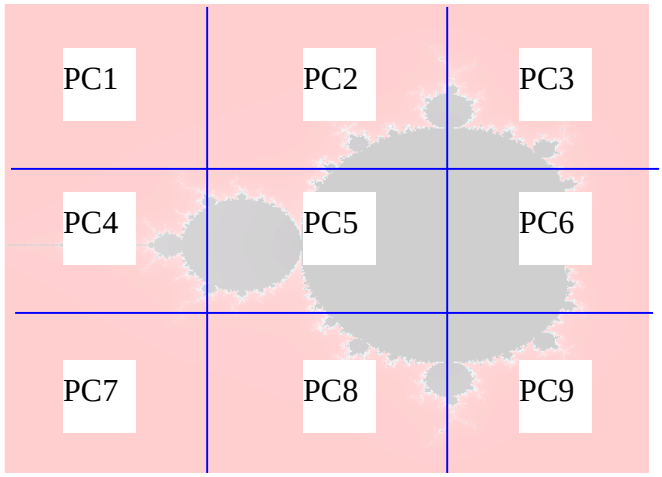
對於整體 Cluster 而言，可以將圖形化為一個二維的的陣列，將此陣列做切割，使得每個節點中各負責一部分的圖像，至於分配切割的問題而言，方法有很多，一般得視特效的特性做不一樣的設定，如下面的幾種常見的特效來說

1. Walsh transform (馬賽克特效)
2. Motion Blur (移動模糊)

Walsh transform 可能較適用於使用方格型的切割方式，因為此種特效一個單位的資料可以儘可能的擺置在同一顆 CPU 上，減少資料上的傳輸時間；而就 motion blur 來說，分析了它的結構與演算法後，可發現每一個圖點會參考由其同一行的圖點（橫向移動），因此我們可能較常使橫條式的切割，使其每一圖點的產生能更快速的產生。

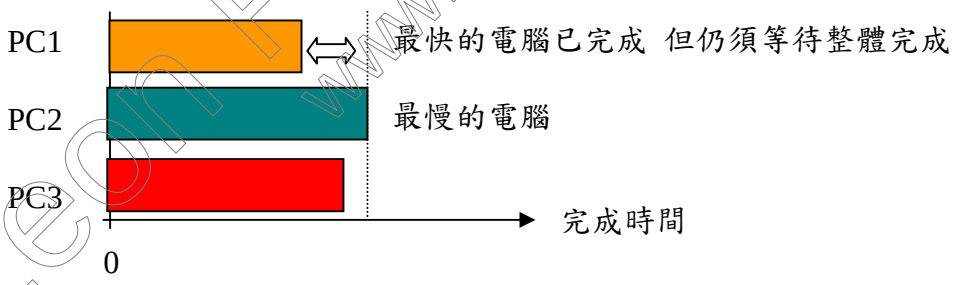
由於 PC Cluster 的特性是網路速度遠遠不及 CPU 與匯流排之間的處理速度，適合使用在 PC cluster 上的一些圖形特效並不能完全能獲得一樣的效率比，通常獨立性越高的，使用計算效能上會越快。

在本研究中，主要先觀察這兩種不同的特效，在系統中，將一張圖分割丟給每個節點 1/n 大小，存放在記憶體中（如**圖九**，各個節點各負責實際上的一部份圖像）而理想中，每個節點處理圖像的範圍就是以存放在其記憶體內的圖素為主，馬賽克特效就是屬於典型適合 PC Cluster 的運算，因為其計算出來的結果就是以一個範圍間的圖素值平均為結果，整塊範圍內所需要傳回的值僅僅只有一個，對於網路傳輸的要求很小，對於傳輸上所需的時間花費最少。



圖九 節點與圖像之分配關係圖

PC Cluster 通常是指若干台相同系統的個人電腦利用高速網路連結在一起，在本研究的環境之下，很難找到配備完全相同的電腦，例如所使用的設備從 Pentium4 到 Pentium MMX，差異極大，結果各電腦的效能就如之前 POV 測試所示，此時，如果要加強效率的應用，必須得對每個節點的流量，即負責計算的量作控制。圖十為固定計算量所花的時間。



圖十 固定計算量所花的時間

不平衡的環境會使得效能的使用率下降，甚至影響整體計算時間，在一開始的研究中，先試著採用較簡易的無流量分配的做法，同時運作的數個節點所花費的時間不一樣，產生此種無可預期的結果，且無論其中一個節點多早完成，整體所花的時間仍是無法縮短。

為了解決這個問題，我們嘗試著使用動態配置的觀念，增加節點間與 Master 的交際，不再以固定的大小丟給節點處理，而是讓節點自己向 Master 提出需求，再由 Master 決定須要的內容，使其間透過更多的訊息交換達到負載平衡。

(四) 利用類神經網路來校調動態配置資料

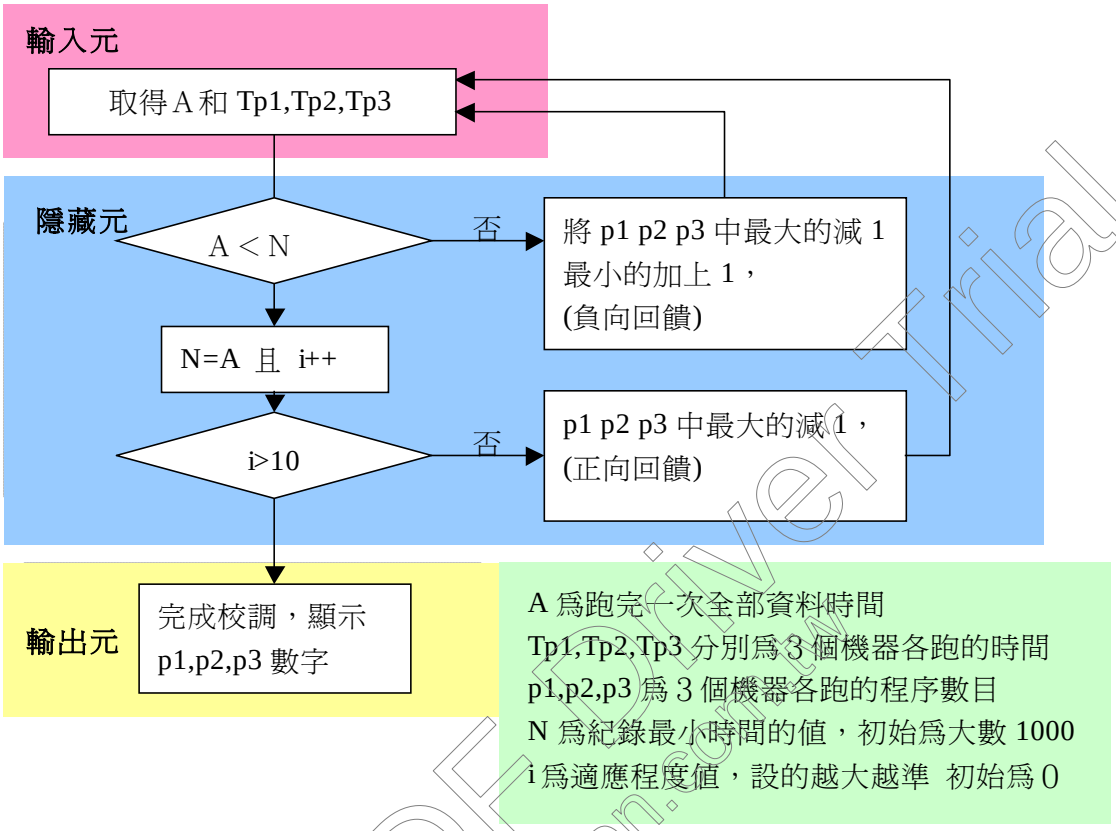
以微積分計算 π 值，調整精準度的指標值至十億，在 3 個節點時 3 台電腦 CPU 速度分別為 1600、700 和 200mhz 進行加權分配 process。在運算速度如此懸殊的情況下，希望能夠在最短時間完成運算，而同時在每台如果都能無閒置狀態的話效能可以提升數十甚至數百倍，其結果如**表一**。

表一 在精確度指標 10^{10}(十億)下，不同 process 分配與花費的時間

CPU	1.6Ghz	700Mhz	200Mhz	總花費時間（秒）
各電腦分配的 process 數量	1	1	1	117.107
	12	10	1	19.172
	12	9	1	16.104
	24	18	2	32.156
	6	5	1	71.131

在沒有使用動態配置前需要 117.107 秒來完成，效率遠低於使用最高速單機(1.6Ghz)來跑的速度 32.724 秒，因此手動將 process 分配給三部電腦。由研究結果發現，將 process 分割成 24 份，依序分配成 12 , 9 ,1 時有極佳的效果 16.104 秒，效能為未分配的 727 %。由此可見，動態分配資料是影響平行運算速度相當重要的一環。

在動態分配計算工作時，可以發現不同速度的 CPU 間即使在 10^{10} 時已經完成動態分配工作的校調，使其差異在 2 秒以內，可是若將計算精確度指標調成 10^{11} 時整體運算速度卻差值可高達 8 百多秒，可見動態分配仍是有問題存在，推測其原因可能因為資料量瞬間流量過於龐大，即使分配成 25 個 process 還是不夠，此時在系統間的傳輸速度遠高於網路的傳輸速度，使得原先分配的模式失去意義。因此，進而想再深入一步的研究動態分配的方法，利用類神經網路來校調使其擁有最佳化的分配。



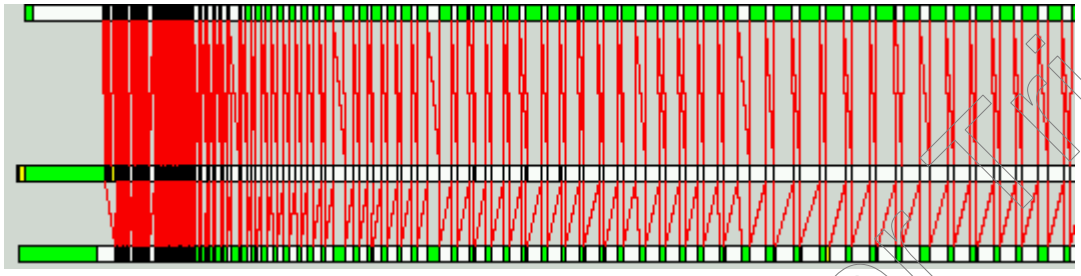
圖十一 利用類神經網路來校調平行運算流程圖

如圖十一，假設運算一個方程式所需的總體時間為 A，3 台機器分別完成工作的時間分別為 Tp1,Tp2,Tp3，各機器跑的 process 數目分別為 p1,p2,p3，i 為適應程度值。

平行運算的重要特點就是速度，所以 A 值決定了最重要的關鍵，為類神經網路的第一判斷條件，而其次就是 3 台機器分別完成工作的時間為適應函數，而影響 3 台機器分別完成工作的時間則是該機器所分配到的 process 數目。因此利用分配的 process 數目來作為校調的神經元，當 A 值小於最小值 N 時，則將 N 指定為目前的 A，且對該神經元作正向回饋，既對原先工作量最多的機器減少 process，因為分散越多的 process 雖然可以增加動態分配的精確度，可是分散越多也意味著在開始部份需要更多時間來溝通，反而會浪費更多時間如圖十二所示。

當 A 值大於最小值 N 時，則對該神經元作負向回饋，既對原先工作量最少的機器增加 process，而且同時也對作量最多的機器減少 process，來使得該基因程式能產生足

夠的歧異度，避免產生無窮迴圈。而利用 i 值來控制基因適應度，此值為校調中第 i 個的時間最小值，數字越大，代表所需校調出來的 A 值需要越小時間而所需訓練時間越久。



圖十二 節點運算之資料傳遞圖

(在開頭的部分紅色的訊息傳遞相當密集)

(四) 遇到的問題：

1. 整體環境的不完全成熟：

目前的平行編譯器仍處於發展中，即使有成品出來，往往所能提供的效能也不是很好，在必須自行發展平行環境之下，開發速度並不高。特別是在多處理器系統上，除錯是一件非常困難的事，希望未來能改善此瓶頸。另外，平行處理的模型有很多種，不像一般的非平行電腦的唯一模型－馮紐曼，每種發展又各不相同，造成研究上的複雜。

2.平行演算法的應用

在研究過程中，我們發現，如果在該問題不是非常複雜的情況下，若利用平行演算法反而會增加程式本身的複雜度，而可以利用其他程式設計的技巧來達到平行化的目的，且在節點傳輸上，也毋須考慮到平行演算法上邏輯或實體鏈結的方式。

因此，在使用平行演算法時，必須對該問題作各方面的評估，才能有效利用平行演算法的方式達到更高效率的運算。

四、結論與未來展望

1. 透過自動完成開機設定的 Boot Disk 能有效的快速建製出高效率的 PC Clusters 環境，可以大量用於現有的區域網路，例如學校中的電腦教室或一個小型的辦公環境等，達到運算資源分享的目的。
2. 使用 PVM 撰寫平行化程式時，需針對網路傳輸流量與程式性質與其最佳化。
3. 利用計算碎形圖形所需的大量運算和其方程式相當的獨立性，適合使用在 PC Clusters 進行運算，但其若未對該平行化程式進行配置最佳化者，效能將大受影響。
4. 在進行繪圖計算時，整體圖形間的相依性越低則使用分散式計算的效能越高。
5. 在 PC Cluster 環境中，若各節點間的運算速度不同時，動態配置各點間的工作量是相當重要的。
6. 透過類神經網路的方式訓練，使其系統能夠自我分配運算工作量，且隨著各平行化程式與各節點的不同，均可自我校調至最佳化，達到高效率且穩定的運算環境。
7. 在本研究中類神經網路的調校方式是適合用於需要以同一程式重複循環計算，對於不同條件的運算模式則須以不同的方式調校。
8. 有鑒於目前建構 PC Clusters 環境仍需要相當的技術，利用自動完設定的 BootDisk 可以廣泛且快速在目前的網路環境中建構環境，減輕了研究時的負擔，未來可使需要用於大量運算的人員進行高效率的研究。
9. 在對於提高平行運算的效能方面，負載平衡是一個相當大的關鍵，利用類神經網路來處理將有著極大的發展空間。

五、參考文獻

1. J.R.Parker , Algorithms for image processing and computer vision ,New York,Wiley Computer Publishing , 1997.
2. David Hm Spector , Building Linux Clusters , USA, O'Reilly & Associates, Inc., July 2000.
3. Yukiya Aoyama & Jun Nakano , RS/6000 SP: Practical MPI Programming , USA,IBM International Technical Support Organization, August 1999.
4. F. Thomson Leighton, Introduction to Parallel Algorithms and Architectures , America , Morgan Kaufmann Publishers , Inc , 1992.
5. W. Richard Stevens ,TCP / IP Illustrated Volume 1,USA , Addison-Wesley,September, 2000.
6. Kai Hwang & Zhiwei Xu , Scalable Parallel Computing , Singapore , McGraw-Hill Companies, Inc , 1998.
7. Richard Stones & Neil Matthew , Beginning Linux Programming 2nd Edition, USA , Wrox Press Ltd. , 1999, 12-2~13-34.
8. Al Geist & Adam Beguelin & Jack Dongarra & Weicheng Jiang & Robert Manchek & Vaidy Sundream , PVM A User's Guide and Tutorial for Networked Parallel Computing ,USA , Massachusetts Institute of Technology , 1994.
9. Marc Snir & Steve Otto & Steven Huss-Lederman & David Walker & Jack Dongarra ,MPI: The Complete Reference , USA ,Massachusetts Institute of Technology , 1996.
10. Ta-YuTsai , Apr 2001 ,Magic Ray-Tracing , Linuxer , No.17 ,P59~64.
11. Hung-Shou Chang & Chao-Tung Yang , Configure and Application Parallel Cluster Computing System on Linux , Linuxer , No. 12 , P84~103.

評　　語

（1）本作品透過可自動設定的起動硬碟建造高效率的 PC 叢集環境，並經由類神經網路動態校調，使運算效能最佳化。

（2）整體研究過程完整，結果頗具實用價值，研究報告若能加強文獻探討將會更完整。