

以網格高效能運算為目的之平行計算叢集建置

張傑生¹ 周秉誼^{1,2} 陳更欣³ 李秀惠¹ 賴飛鵬¹
台灣大學計算機及資訊網路中心¹ 臺灣大學農藝系² 台北護理學院電算中心³
{jsc, b91601049, seventeen, hhlee, flai}@ntu.edu.tw

摘要

科學計算往往伴隨著非常大量的運算資源，為了滿足研究學者的需求，利用叢集系統(Cluster)配合平行運算，可說是近年來十分流行的高效能運算系統(High Performance Computing)建置方法。其中又以Linux與MPI程式庫所組成的運算叢集系統最廣為接受。除了open source所帶來的彈性與便利外，成本低廉、效能出眾的優點，更令大家趨之若鶩。在多使用者的環境之下，配合適當的工作排程管理，可以讓整體系統資源獲得最佳化的調配。定期進行系統分析，並加以調校，可確保系統效能隨時處於顛峰狀態。更進一步地，配合網格技術(Grid)，透過網路串連多組Cluster系統，更能夠有效率地分享閒置計算資源，進而達成資源共享的目的。

本篇文章將著重於網格運算叢集的建置經驗分享，希望能藉此推廣open source軟體與高效能計算平台，進而促成各大學積極分享計算資源，降低資源閒置的機會，以收拋磚引玉之效。

關鍵詞：網格運算、叢集系統、高效能運算、平行計算

1. 前言

常見的科學研究例如大氣模擬、物理及天文科學等，大多需要龐大的數值計算。過去這種工作往往曠日廢時，耗費大量人力。隨著計算機的發明與普及，機器所帶來的強大運算能力，造福這些研究得以快速獲得實驗結果。這種現象也解釋了二十世紀科學發展的突飛猛進。

以往，想要擁有迅捷的運算速度與充足的儲存空間，必須求助於超級電腦(Super Computer)或大型主機(Mainframe)。然計算機相關科技演進至今，舊式體積龐大的大型主機日漸淘汰，過去高成本、封閉的作業環境，已為輕便又便宜的個人電腦所取代。個人電腦的微處理器進步飛快，價格低廉，運算速度卻不斷提升，平均以每一年半就加倍的速度成長著，也使運算時脈屢創巔峰。不只處理器方面的躍升，記憶體存取速度、儲存容量和價格，也因為DRAM製造技術的進步，有著相當大的成長[2]。近幾年來，因應記憶體容量提升，記憶體定址長度需要做相對應的擴充，於是六十四位元處理器

隨之而生。由於六十四位元定址解決了科學運算對於大量記憶體操作的需求，相信在不久的將來，這種處理器將會大量普及。

當前的高效能運算大多是以平行處理的概念為主流。其運算原理主要是將原本循序執行的程式，切分為數個小部份同時執行，以增進程式執行的效率[3]，這就是所謂的平行化。平行處理受到重視的原因，在於這種運算方式能夠突破單一處理器在硬體方面所受的侷限。當單一處理器需要大幅提昇運算能力時，相當容易遇到硬體設計的瓶頸，包括處理器時脈不可能無限制提升、相關週邊I/O匯流排的頻寬無法跟上處理器速度等等。為了提升運算能力，因此產生多重處理器(Symmetric Multiple Processor)的架構。在這種環境下，程式必須做相對應的改變，也就是需要切成小部分，分散在不同處理器上平行執行。這種多重處理器的架構，現今廣為應用。許多為大量計算而設計的主機系統，就是由很多個人電腦的微處理器所組成。在Top500的超級電腦排行榜中，幾乎所有的系統都是採用多重處理器的架構，其中有超過一半的主機系統是採用個人電腦的微處理器。而目前全球最快的計算機系統，就是以多重處理器構成的系統，運算能力高達136.8 Tera Flops[17]。

考量一般單位有限的經費，大而集中式的多處理器主機價格過高，遠超過可負擔的能力。退而求其次，可以考慮利用多台廉價電腦所組成的運算叢集系統(Cluster)，以達到媲美大型主機運算環境及運算能力的高效能運算(High Performance Computing)。Cluster是現在最常見到的高效能計算系統，因為同時具有擴充性佳、價格低廉、效能優異等優點，所以被許多研究單位用來處理高能物理計算、高解析影像處理、蛋白質等巨型分子結構模擬、大氣模擬、和基因序列搜尋等問題。許多人開玩笑地說，Cluster是窮人的超級電腦。

為了達到高效能運算的目的，除了利用Cluster之外，尚可採用網格運算(Grid)的概念。Grid在技術上可視為Cluster的延伸，目的希望達成虛擬化的運算服務。Grid可以透過中介軟體，將分散各地的運算能量加以整合，對使用者而言，就像面對一組超大型但更分散的Cluster系統。許多利用Grid概念的大型計算計畫正在網路上風行，例如SETI@Home[16]利用全球自願者個人電腦的閒置運算資源，分析宇宙中的電波，找尋其他可能的高

等文明。其他的大型應用包括 LCG[10]、EDG[5] 等等。

本文將嘗試分享台灣大學建置網格式叢集運算系統的經驗。文章架構如下：第二章介紹 Cluster 相關背景知識，第三章說明架設與規劃的重要考量，第四章則簡述如何使用 Cluster 系統，第五章描述將 Cluster 透過 Grid 與其他 Cluster 串連之相關議題，第六章則是效能分析，第七章為結論。

2. 背景

為了以數台個人電腦，達到分散式平行計算的目的，必須藉由 Cluster 的架構，並配合特殊的平行化程式庫。以下將針對幾項基本要素加以說明。

2.1 Cluster 架構

基本的 Cluster 系統是由數個部份所組成，包括 Master、Computing Nodes、資料交換介面、以及 Data Center[1]。

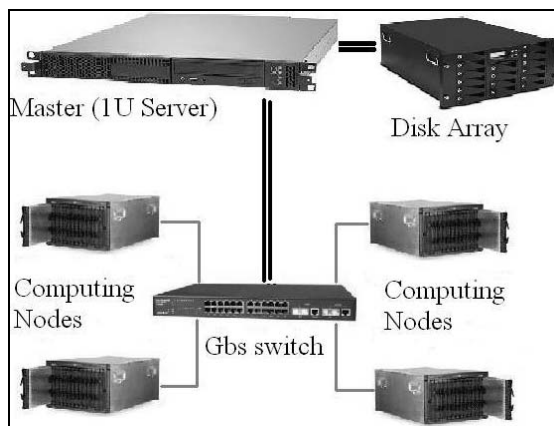


圖 1 Cluster 系統架構圖

以下針對這些名詞作簡單介紹。

- 主控台 (Master) 通常由一台獨立的伺服器擔任。除了必須面對使用者，接受使用者提交的工作之外，更重要的功能在於管理整套 Cluster 的資源運用，及分配各個工作在不同的 Computing Nodes 上執行。同時它也必須連接資料儲存媒體，將使用者的資料和計算結果儲存在大容量、高效能的資料中心。

- 計算節點 (Computing Nodes) 是真正執行運算工作的機器。通常機器的數量代表著 Cluster 的規模。為了提高機房空間的利用率，可以考慮採用超薄型或刀鋒型伺服器。當 Master 接收到使用者提交的工作後，會依一定的原則和工作的性質，將工作分配給數個 Computing Nodes 做計算，Computing Nodes 計算完成後必須將結果送回 Master 記錄。

- 資料交換介面即為高速網路。考量經濟成本可以選用廉價的 Gigabit Ethernet；若需要追求效能，則不妨嘗試 Myrinet、InfiniBand 或 Dolphin。當平行化程式執行時，各個 Computing Node 需要頻繁

地交換資料與同步進度，也因此 Cluster 的整體運算效能將取決於網路傳輸效率。

- Data Center 通常由磁碟陣列等大容量儲存媒體所組成，再利用 NFS 或 GFS 分享給每個 Computing Node。大部分的科學計算工作都需要大量的儲存空間，唯有穩定可靠的儲存媒體才能確保辛苦得來的數據安全無虞。

2.2 平行化程式

平行計算的基本概念是將程式分為數個小部份，分別在不同的微處理器上運行，以達到同時計算、提高效能為目的，所以程式的可平行程度和各種平行演算法對效能有著決定性的影響。我們可以利用程式平行化 (parallel) 與序列化 (serial) 執行時的效能相比，進行平行化的效益評估，稱為 Speedup。

程式的平行化可以利用訊息交換函式庫呼叫、或使用專為平行計算設計的程式語言及編譯器來達成。一般來說，利用訊息交換函式庫呼叫的方式較有彈性，也可以在現有程式語言平台上工作，可以直接改寫原有程式達到平行計算。現在常見的訊息交換函式庫有 MPI 和 PVM。

MPI (Message Passing Interface) 是一種為平行計算而設計的訊息交換介面的標準，它讓分散在不同微處理器執行的程式，可以透過網路等媒介交換資料[12]，一般是利用 fortran 和 C/C++ 語言撰寫相關程式。程式設計時，以函式呼叫的方式，將訊息交換寫在程式碼中，編譯後和實作 MPI 的函式庫連結，就可以在數個微處理器上平行計算了。

MPI 有許多在不同環境運行的實作，在 Linux 下有一公開原始碼的程式庫實作 MPICH 可供使用。MPICH 利用 mpirun 程式，透過 SSH 連線到不同 Node 上執行 MPI 程式，再由網路交換訊息[13]。

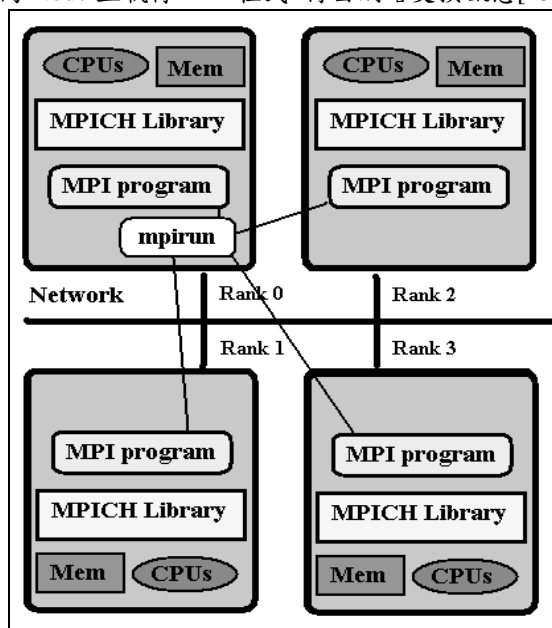


圖 2 MPI 平行程式運作架構圖

2.3 工作排程

在多使用者的環境，使用者會互相競爭運算資源，透過適當的工作排程管理，才可以讓整個 Cluster 系統的資源得到最大且最公平的運用。工作排程可利用佇列管理軟體 (Queue System) 來達成，如 PBS、Condor[4]、Cmeta 等。

PBS 有一開放原始碼的版本 OpenPBS，可以設置數個不同性質的佇列，將不同性質的工作放入各種佇列，且不同佇列可設定不同優先權和資源使用權，利用數個不同的佇列設定，就可以將資源妥善的分配[14]。PBS 系統程式主要分為三部份，pbs_server 用來配置各種佇列且做為使用者和節點間的介面。pbs_sched 依設定的權限，將佇列中的工作分配給 Computing Nodes 執行。pbs_mom 運行在 Computing Nodes 上，用來執行工作。

以上介紹的就是 Cluster 的架構。組成 Cluster 系統後，配合平行化程式庫及工作排程軟體，加上編譯程式及計算時會用到的程式庫，即可完成簡單的 Cluster 系統。

3. Cluster 規劃與架設

本中心秉持的精神，即是以 open source 軟體建置平台，並透過該平台，發展更多 open source 軟體，以回饋網路社群。因此所安裝的程式庫和軟體，皆為開放原始碼 (Open Source) 或自由軟體 (Free Software)，所有人都可以用最低的成本自由運用。

3.1 規劃 Cluster 架構

開始規劃 Cluster 之前，要先考量設置的目的，以便進行規劃配置。節點 (Nodes) 數量通常代表 Cluster 規模。一般來說，可依 Computing Nodes 的硬體規格及網路配置，分為八到十二個為一組的數個小群。將不同小群以不同代號命名，並依群組名將各節點命名編號，方便日後指定運算節點。另外，還需規劃佇列，設立兩到三個佇列，各列有不同資源權限和資源使用量上限，方便不同性質的工作執行。也需要規劃各節點的硬碟配置，設立大容量磁碟分割給需要大型暫存檔的程式使用。

表 1 佇列設定參考

Name	Running	User Run	Nodes	Time (hr)
large	2	1	16	168
normal	8	2	8	72
mini	12	4	4	24

表 2 節點硬碟分割參考

Mount Point	Size (Mb)	Desc.
/	1000	基本系統
swap	2000	swap
/tmp	2000	供系統程式暫存使用
/usr	10000	放置套件及程式
/scratch	20000	供計算程式暫存使用

3.2 安裝 Master 及建立基本服務

先安裝好 Master 的作業系統 Debian GNU/Linux，完成系統更新，再利用 apt-get 套件管理程式，架設以下服務及套件。

- NIS：讓整個 Cluster 的每個 Node 有相同的使用者帳號及密碼，方便使用者管理。
- NFS：讓 Master 的檔案和資料中心的資料可以被所有節點分享，基本上就是將使用者的家目錄分享給每一個 Node。
- DHCP：以 Computing Nodes 的網路卡卡號 (MAC)，分配 IP 位址。同時為透過網路自動化大量安裝程序所需。
- NAT：讓內部虛擬網路可以對外連線。在安裝 Computing Nodes 時會下載套件，需要對外連線到 apt 服務主機。
- 編譯及直譯程式：gcc、g++、g77、perl、R、octave、python、java 等 (以科學計算和生物資訊為主)。這些是使用者編寫程式和運行需要的。
- 程式庫：mpich、blas、atlas、libgmp、libgsl、lapack、blast 等 (以科學計算和生物資訊為主)。

3.3 大量安裝作業系統 Debian GNU/Linux

一般來說，Cluster 通常至少具備 16 組，甚至 64/128 組以上的 Nodes。若無法利用自動化的方法安裝系統，對於管理成本來說，將是一項沈重的負擔。我們建議採用網路開機 (PXE) 方式，進行自動化安裝，首先依規劃時的群組，紀錄網路卡號碼，以 DHCP 分配網路位址。在 Master 上安裝 FAI server (Fully Automatic Installation)[6]，依據節點硬體和 Cluster 配置修改 FAI 的 class 設定，加入各種計算需要的套件及相關設定檔，設定硬碟分割，讓 FAI 程序自動安裝。以具 PXE 功能之網路卡啟動 Master 上的核心 (kernel)，進行 Computing Nodes 的安裝流程，完成後重新開機 Computing Nodes 即自動上線。如安裝過程出現核心錯誤，可能需要自行編譯可支援 NFS 為根目錄的核心。

3.4 安裝工作排程軟體 PBS

開放原始碼的 OpenPBS 可在 PBS 網站上取得，也可以使用延伸自 OpenPBS 的 TORQUE。安裝完成後，將各個 Node 的名稱、分組和處理器個數寫入設定，在 Master 啟動 pbs_server 和 pbs_sched，在 Computing Nodes 啟動 pbs_mom。且依安裝前規劃，利用 qmgr 指令設定不同性質佇列，供使用者選用。

另外可安裝幾個監控 PBS 系統的小工具。
 pestat 用來顯示每一個節點的系統負載 (load)、記憶體及運行工作的編號。pbs_acct 可以從 PBS 的 log 裡，統整出各使用者工作提交情形和工作花費時間。pbsnodect 也是從 log 裡統整出每個 node 的使用情形，處理器忙碌的程度等。

另外 PBS 尚有網頁介面，由 php 和 javascript 驅動，方便對 linux 指令系統不熟悉的使用者使用 [15]。

node	state	load	pmem	ncpu	availmem	usr	jobs	jobids
t03	free	0.90*	1011	4	37	1/1*	1	7576
t05	free	0.97*	1011	4	168	1/1*	1	7576
t07	free	0.98*	1011	4	78	1/1*	1	7576
t08	free	0.93*	1011	4	88	1/1*	1	7576
t09	free	0.97*	1011	4	95	1/1*	1	7576
t0a	free	0.94*	1011	2	53	1/1*	1	7576
cg0	free	0.91*	1011	4	59	1/1*	1	7576
cg4	free	0.02	1011	4	861	0/0	0	
cg5	free	0.93*	1011	4	41	1/1*	1	7576
cg6	free	0.97*	1011	4	270	1/1*	1	7576
cg7	free	0.86*	1011	4	286	1/1*	1	7576
cg8	free	0.98*	1011	4	162	1/1*	1	7576
cg9	free	0.99*	1011	4	163	1/1*	1	7576
ag0	free	0.99*	1011	4	11	1/1*	1	7576
ag1	free	0.97*	1011	2	285	1/1*	1	7576
ag2	free	0.90*	1011	2	25	1/1*	1	7576
ag3	free	0.99*	1011	4	117	1/1*	1	7576
ag4	free	0.00	1011	2	624	0/0	0	
ag5	free	0.00	1011	4	476	0/0	0	
ag6	free	0.02	1011	4	448	0/0	0	
ag7	free	0.00	1011	4	483	0/0	0	
ag8	free	0.00	1011	2	483	0/0	0	

圖 3 pestat 監看各節點狀況圖

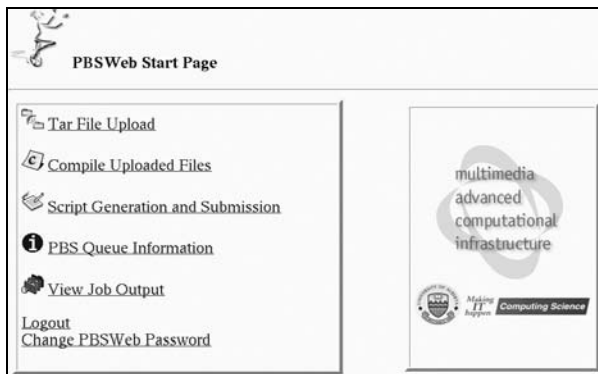


圖 4 PBSWeb 介面

3.5 管理、監控及效能測試工具

當 Cluster 系統的節點數量龐大時，系統管理將成為極大的人力負擔。此時我們必須藉助適當的工具來幫助我們管理系統、監控系統，甚至進行效能測試評比。

- rgang：根據事先定義的節點群組集體管理，可以讓每一個節點執行同樣的工作。
- ganglia：專為 Cluster 系統設計的監控程式。可以將每一個節點的處理器、記憶體和網路等的使用量和運作情況，統整收集[7]。並具備圖形化的網站介面，可透過 rrdtool 將收集到的資訊繪製成易於掌握的圖表。
- llcbench (Low-Level Characterization Benchmarks)：由三個套件 (MPBench、CacheBench、BLASBench) 組成的效能測試組，用以測試系統的低階結構特性，如 MPI 網路效能、微處理器快取效能、及浮點數計算能力，在平行程式設計時可做為參考[11]。
- HPL (High-performance Linpack Benchmark)：

透過 MPI，解隨機產生的大型非稀疏線性方程組，用以評量 Cluster 系統的浮點數運算能力 (flop)，也可藉此了解系統效能提升比例 (Speedup)[9]。

4. Cluster 之使用方式與流程

在多使用者的環境下使用 Cluster 系統進行計算工作，需透過 PBS 工作排程系統提交工作，以利資源分配。

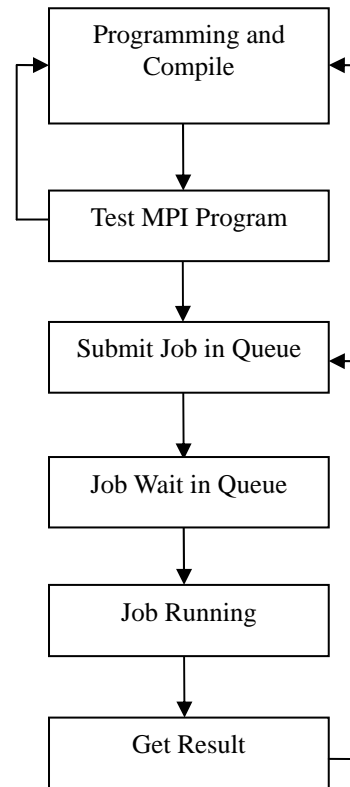


圖 5 Cluster 系統使用流程圖

以下是簡單的運算程序。

- 使用者在 Master 撰寫程式，並完成編譯。編譯可以用 mpicc/mpicc 等編譯器，或直接使用 gcc/g++ 編譯，再連接到 mpich lib。在編譯程式時，應善加利用編譯器提供的各種最優化參數 (Optima option)。
- 編譯完成的程式先在小規模的測試環境下運行、測試及偵錯。當結果符合預期時，再提交到 Cluster 的 PBS 佇列，以減低資源浪費。由於工作有最大執行時間的限制、Computing Nodes 的記憶體也有限，在測試時也要估算程式所耗費時間和記憶體用量，以適當切割計算為數個工作。
- 測試完成後，依 PBS 的格式，撰寫一腳本 (script) 指定資源需求，並以 qsub 指令提交工作。
- 工作被提交後，會在佇列裡等待所要求資源可供使用。當系統使用人數較多、工作較大時，就要花費很多的時間等待，管理者應由 pbsacct 指令，定期監控等待情況，而更改佇列設定，使系統資源適當分配。
- 當獲得資源分配時，工作即會開始執行。工作

執行期間，可透過 `pestat` 觀察程式運作，處理器使用量和記憶體使用，以便計算效能的提高。

- 工作完成後，會依腳本中的設定，將結果儲存在特定檔案。使用者可依此結果，修改程式除去錯誤，或提交不同設定之工作。

使用者可以利用幾個簡單的指令來了解程式執行和 PBS 系統的狀況，也可管理已提交的工作。

- `qstat`：可顯示佇列中的工作列表，工作所要求的資源，及每個佇列的狀態。
- `qsub`：用來提交工作，可以利用參數設定不同的資源需求，但一般多利用工作腳本 (PBS Script) 來設定資源。
- `qdel`：當工作發生錯誤時，可將已提交的工作刪除。

5. 網格運算應用

在 Cluster 的使用過程中，常常發生以下兩種問題。首先是資源閒置，當無人使用的時候，Cluster 仍舊開機運轉，繼續消耗電力資源。再來則是「大問題」的限制，當使用者的工作需求大量 CPU nodes，而目前 Cluster 內閒置的 CPU node 卻無法達到要求，那麼該工作將無限期等待，造成使用者的焦急與無奈。因應以上問題，Grid (網格運算) 的概念油然而生。Grid 的精神在於資源分享，也就是要善用運算資源。將自己 Cluster 的閒置 CPU nodes 提供予其他 Cluster，進而未來需要的時候，也可以向其他 Cluster 借用閒置 CPU nodes，以達到資源分享的目的。

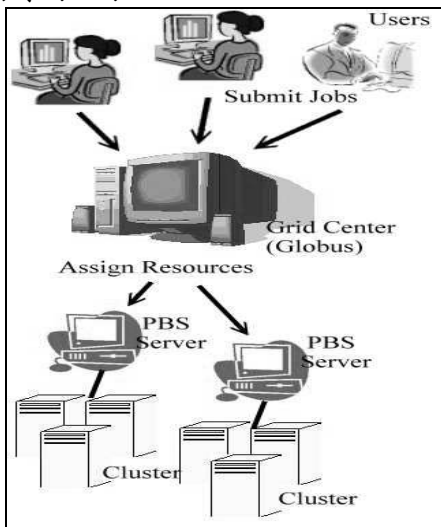


圖 6 Grid 架構圖

目前的 Grid 大都由 Globus 工具組提供的協定所延伸。Globus 工具組主要分為三個部份，Resource Management 用於資源的管理分配、Information Service 收集 Grid 上的資訊、Data Management 進行資料的散布和管理，以上服務皆建構在 Security Infrastructure 上，以確保 Grid 安全 [8]。想要使多個 Cluster 串連，達到 Grid 的效果，需利用 Globus 來管理數個 PBS Server 及其資源，

使用者直接對 Globus 提交工作，讓 Globus 透過網路在數個分散各地的 Cluster 中，找到最適合的資源執行工作。

台灣大學於 2004 年與台北醫學大學合作，透過 Grid 網格技術串連雙方 Cluster，藉以實驗可行性與效能分析。雙方成功合作發展「醫療影像擷取分析系統」，將複雜的醫療影像分析工作，透過網路同時交由兩組 Cluster 分析，以即時獲得分析資料，降地醫師等候時間。

6. 使用狀況與效能分析

Cluster 架設完成後，即可進行系統效能測試和開放使用者使用。本中心目前架設了兩套 Cluster，用以進行高效能平行計算及 Grid 運算的實驗，並開放資源予學術研究之用。以下從使用情況和系統效能來表達 Cluster 系統成效。

6.1 使用情況

目前本校使用者進行的計算，以物理、化學與生物領域的科學計算為主，以下是使用情形的相關統計。

表 3 Cluster 系統使用統計表

	No. of Jobs	CPU Days	Avg. Nodes per Job	Avg. Day in Que
Jan-05	347	1297.88	5.43	0.02
Feb-05	644	1581.11	3.17	0.09
Mar-05	1543	1154.28	4.72	0.04
Apr-05	1146	1240.47	7.62	0.09
May-05	198	625.52	4.52	0.25
Jun-05	259	534.80	4.09	0.07
Total	4137	6434.06	4.93	0.09

6.2 系統效能

效能測試的目的在於瞭解系統，進而調校系統。根據各種不同的測試結果，可以找出系統的瓶頸，嘗試解決該瓶頸，以獲取最大效能。

- `l1bench`：系統架設在 Gigabits Ethernet switch 上，所以 MPI 網路的吞吐量 (throughput)，可達 100 MB 以上，在大量資料交換上有不錯的成績。

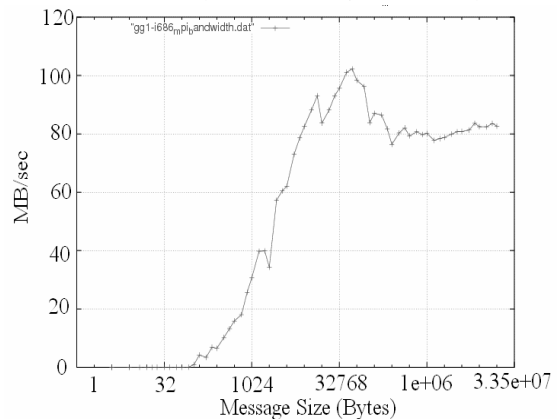


圖 7 由 l1bench 測得之 MPI 網路頻寬

- HPL：當 32 個處理器同時執行時，可達到 26.19Gflops 的運算能力。當同時使用的處理器個數多到一個程度，系統的 Speedup 就越差。

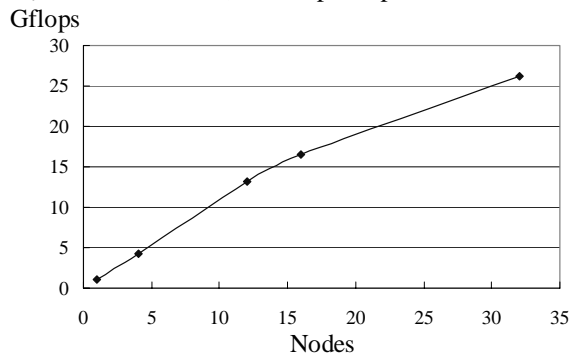


圖 8 由 HPL 測得的浮點數運算能力

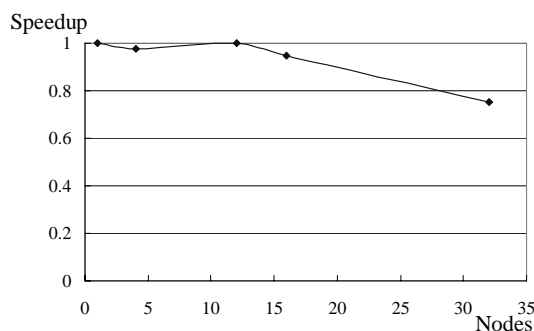


圖 9 不同數量處理器的 Speedup

6.3 平行化效能討論

為了瞭解處理器數目與平行化效能之相關性，我們分別使用 1、4、16、32 組處理器測試實驗。透過 HPL 效能分析結果顯示，當處理器的數量大於 16 個，系統的 Speedup 開始迅速遞減。推測由於使用的 node 數目增加，MPI 程式需要建立的連線也增多，雖然仍在網路硬體設備的負擔範圍內，但網路的碰撞 (Collision) 卻使整體傳輸效率大減。此外，使用 node 數目越多，就需要處理越大的問題，然單一 node 的記憶體有限，當問題大於記憶體能儲存，就要用到速度很慢的硬碟去做 SWAP。所以我們在佇列的設定中，單一工作使用的 node 數目極限為 16 個。為了達到最佳的運算效果，除了管理者必須了解上述的 MPI 平行程式特性，我們也得持續透過教育訓練，灌輸使用者正確的觀念。

7. 結論

台灣大學目前利用價格較為低廉的 Intel x86 架構電腦，配合 open source 軟體，包括：Linux 與 MPI 等，成功建置平行化高效能運算叢集系統。以目前 210 組計算節點的規模，換算成單一處理器的運算時間，早已超過 12,000 個運算日，相當於單

一電腦連續工作 40 年以上。我們提供了充足的資源，可以讓物理、化學、生物、資訊領域等研究團隊前來利用，不但滿足他們對於龐大科學計算能量的需求，也有效加速他們在專業領域上的實驗腳步，進而產出各式研究成果。

在 Cluster 本身的管理技術方面，我們著重於效能調校，透過持續監控系統紀錄，配合各式效能分析軟體，調整系統設定，觀察效能改變，以取得最佳化的設定值。希望藉此壓榨出電腦的極限，用有限的預算成本，提供最高的效能予使用者。

未來我們希望能夠藉由推廣 Cluster 的建置經驗，鼓勵其他學術單位參與建置。除了可以提供該單位內研究團隊所需的計算資源外，更進一步地，可以藉由 Grid 技術，透過高速網路連結，將許多 Cluster 連線，共享寶貴資源，將 Cluster 閒置時的 CPU 時間，分享予其他需要的單位，以提高整體利用率的方式，避免資源浪費。

8. 參考文獻

- [1] Lucke, R. W., Building Clustered Linux Systems., New Jersey: Prentice Hall PTR. 2005
- [2] Patterson, D. A., and J. L. Hennessy. Computer Organization Design. The Hardware/Software Interface., San Francisco: Elsevier Inc. 2005
- [3] Quinn, M. J., Parallel programming in C with MPI and OpenMP., Boston: McGraw-Hill Higher Education. 2004
- [4] Condor
<http://www.cs.wisc.edu/condor/>
- [5] European DataGrid Project
<http://eu-datagrid.web.cern.ch/eu-datagrid/>
- [6] Fully Automatic Installation
<http://www.informatik.uni-koeln.de/fai/>
- [7] ganglia
<http://ganglia.sourceforge.net/>
- [8] Globus Toolkit
<http://www.globus.org/toolkit/>
- [9] HPL
<http://www.netlib.org/benchmark/hpl/>
- [10] LHC Computing Grid Project
<http://lcg.web.cern.ch/LCG/>
- [11] llbench
<http://icl.cs.utk.edu/projects/llbench/index.html>
- [12] MPI
<http://www-unix.mcs.anl.gov/mpi/index.htm>
- [13] MPICH
<http://www-unix.mcs.anl.gov/mpi/mpich/>
- [14] OpenPBS
<http://www-unix.mcs.anl.gov/openpbs/>
- [15] PBS Web
<http://www.cs.ualberta.ca/~pinchak/PBSWeb/>
- [16] SETI@HOME
<http://setiathome.ssl.berkeley.edu/>
- [17] Top500, <http://top500.org>