

行政院國家科學委員會專題研究計畫(第三年)成果報告

以 WWW 為介面的高效能平行計算編譯環境

A Web-Based Compiling Environment for High-Performance Parallel Computing

計畫編號：NSC 89-2218-E-009-094

執行期限：89年8月1日至90年7月31日

主持人：曾煜棋 教授 交通大學資訊工程系

一、中文摘要

本計畫是為期三年的整合型計畫，目的為設計一跨平台的高效能計算編譯環境，希望能經由此環境之發展，提供程式設計師或學術研究者一個友善（User-Friendly）且高效率的平行程式發展環境，以期減輕開發平行程式所需花費的代價，並開發高效能的平行程式。此高效能計算編譯環境的另一項特色是具有跨平台（Platform-Independent）的特性，可提供更廣泛的使用者使用此高效能計算編譯環境，進而提昇各界人士的工作效率。

使用者可經由瀏覽器，透過建構在全球資訊網路（WWW）下的使用者介面，將所撰寫的循序程式（FORTRAN 程式）送到一平行編譯伺服器（Parallel Compiling Server）上去編譯。經由 SUIF Compiler System 之前置處理（Preprocessing）後所產生的語法樹（Syntax Tree），我們對輸入程式做靜態（Static）及動態（Dynamic）的資料分配分析，找出最適當的資料分配方式。基於這樣的資料分配方式，找出所需的通訊集合（Communication Sets），產生一 SPMD（Single Program Multiple Data）模式的程式。為了使 SPMD 程式在各處理器上執行的更有效率，我們再進一步做地域性（Locality）的最佳化處理，以期增加快取記憶體（Cache Memory）與區域記憶體（Local Memory）的存取效率。然後根據實體處理器（Physical Processors）的拓樸架構（Topology）及處理器間之通訊比重（Communication Weight），將虛擬處理器（Virtual Processors）對映到實體處理器上，同時做通訊最佳化處理（Communication Optimization）。最後配合所設計的通訊函式庫（Communication Library），並根據使用者所指定的平行工作平台是同質式（Homogeneous）或異質式（Heterogeneous），分別產生對應的 C + PVM / MPI 或 Java + PVM / MPI 的程式，結束編譯。當程式編譯完成後，代理者伺服器（Agent Server）會將編譯後的平行程式在使用者所指定的平行工作平台上執行，並將執行結果送回。在程式於編譯及執行的過程中所產生的各種訊息會以視覺化（Visualization）的方式呈現，並可經由使用者介面讓使用者瞭解程式的特性、執行效率及系統資源的利用情形，使用者並可透過使用

者介面微調編譯程式的各項參數，以達到最高的執行效能。

關鍵詞：代理者管理計算資源、通訊程式庫、通訊集合、動態資料分配、MPI、平行化編譯器、PVM、靜態資料分配、SPMD。

Abstract

This report proposed an integrated project lasting for three years. The purpose of this integrated project is to design and implement a web-based compiling environment for high-performance parallel computing. In order to alleviate the overhead of writing parallel programs and develop high-performance parallel programs, we hope the environment can provide programmers and/or researchers a user-friendly and high-performance parallel programming environment. One of the advantages of the environment is platform-independent. Due to this advantage, the environment can provide wide-ranging users to use this environment and further improve their work performance.

This environment allows a sequential program (FORTRAN programs) as the input. By the user-interface constructed under WWW, users can use a browser to input the sequential program. The input program will be sent to a parallel compiling server and be compiled. SUIF compiler system will preprocess the input program and generate an abstract syntax tree. According to the abstract syntax tree, we will figure out an appropriate data distribution for the input program after static and dynamic data distribution analyses. Based on the data distribution, we will first calculate the necessary communication sets and then generate the SPMD program. To improve the performance of the SPMD program, we further optimize the data locality to increase the access of cache memory and local memory. After that, the virtual processors will be mapped to physical processors according to the topology of the physical processors and communication weights among physical processors. Machine-dependent communication optimization is also done in this phase. Finally, we link the communication library to generate the C+PVM/MPI or Java+PVM/MPI code according to the parallel

platform is homogeneous or heterogeneous, respectively. While the compilation phase is done, agent server will send the execution code to the computation server. The computation server then sends the results back after the execution code is executed. The collected information from compilation and execution phases will be demonstrated in visualization. The users can observe the execution performance and resource utilization of the program via user-interface. With the observation, users are able to fine-tune, through the user-interface, the parameters of the parallel program to improve the performance of parallelized programs.

Keywords: Agent-based Computing, Communication Library, Communication Sets, Dynamic Data Distribution, MPI, Parallelizing Compiler, PVM, Static Data Distribution, SPMD.

二、緣由與目的

本整合型計畫設計一具跨平台的高效能計算編譯環境，經由此環境之發展，提供程式設計師或學術研究者一個友善 (User-Friendly) 且高效率的平行程式發展環境，以期減輕開發平行程式所需花費的代價，並開發高效能的平行程式。此高效能計算編譯環境的另一項特色是具有跨平台 (Platform-Independent) 的特性，可提供更廣泛的使用者使用此高效能計算編譯環境，進而提昇各界人士的工作效率。

分散式記憶體多處理機系統及分散式計算系統均沒有一共用的記憶體，各處理器有各自的區域記憶體 (Local Memory)；因此在設計程式時，須考慮如何將一件工作及工作所需的資料分配到各處理器上；除此之外，還須自己控制資料傳輸的工作、處理器間同步的問題等耗時又易出錯的工作。若再要求執行效率的話，則還須將各處理器的工作負載量、資料相依性等因素考慮進去，設計起來更形複雜。當程式除錯時，程式設計師更是捉襟見肘，不僅沒有良好的除錯工具可用，自己還須模擬平行程式的運作方式找出錯誤的位置，真是苦不堪言。就另一方面而言，目前現存大量的循序程式，若要一一改寫為平行程式，不僅工程浩大且耗時耗力，只怕所費不貲且事倍功半，得不償失。因此，本整合型計畫希望能經由此環境之發展，提供程式設計師或學術研究者一個友善 (User-Friendly) 且高效率的平行程式發展環境，以期減輕開發平行程式所需花費的代價，並開發高效能的平行程式。

由於全球資訊網路 (WWW) 的盛行，建立在網際網路 (Internet) 下的平行執行環境及介面，提供更廣泛的使用者來使用高效能計算資源，進而提昇各界人士的工作效率，是目前很重要的一項服務。另外，Java 語言因其高度的可攜性、安全性及透通性，提供了嶄新的平行及分散式計算環境。利用 Java 之跨平台能力，經由網路之伸展，建立具可延展性、高效能之 Java 計算環境，亦是刻不

容緩的。因此，本整合型計畫希望能經由此環境之發展，提供更廣泛的使用者使用此高效能計算編譯環境，進而提昇各界人士的工作效率。

目前國外有相當多的研究學者投注於資料平行程式環境的研究。Fortran D Compiler 計畫[1,2] 為 K. Kennedy 所領導的，其目的是為資料平行的應用程式提供一個與機器無關的程式設計環境，並將與機器相關的部分由編譯器負責。H. P. Zima 所領導的 SUPERB 計畫[3,4]則是一個半自動的平行化系統，架構在 SIMD 及 MIMD 模式的 SUPRENUM 多處理機上。此系統是專為數值應用程式設計的，是一個交談式的 source-to-source 轉換的系統。PARADIGM 計畫[5]是由 P. Banerjee 所領導的，此編譯器可以完全自動地將循序程式轉換成可在分散式記憶體多處理機上有效率執行的平行程式。M. S. Lam 所領導的 SUIF 編譯器系統[6]則針對共享式記憶體多處理機自動地產生資料及計算的分配方式。此系統可以解決較廣泛的問題而且可以在平行度與資料區域性間找到較佳的處理方式，同時可提供更多的機會處理通訊最佳化的問題。

對於一個幫助程式設計師發展平行程式的環境而言，用圖形的方式來表現系統的行為是非常重要的。事實上，這種觀念早已融入許多的平行程式設計環境中，如 ParaGraph[7]配合 PICTL[8]、架在 PVM 上的 HeNCE[9]、配合 Poker[10] 的 Voyeur[11]，及設計一個 Web-Based 的分散式 3D 的動畫環境[12] 等系統均是。另一方面，在國內外，Java 的平行及分散式編譯計算的相關研究有，Indiana University 所提出的 Javar and Javab-Java Restructuring Compiler[13,14]。是利用 Java 本身語言具有的多工緒 (Multithread) 的功能，將循序的程式化成 Java 的多工緒的執行。此種編譯的結果，適合於單機多處理器的機器將會比單機單處理器的機器，有明顯的效益的增進，針對多機多處理器的編譯系統，是 Javar and Javab 系統所不足。近來，Ken Kennedy[15] 設計一個編譯 Java 的高效率計算環境，可見 Java 高效率計算環境相關研究的重要性。另外，有相關於 Java 版本的 PVM 及 MPI 系統的研製，如 Java PVM 系統[16]，JPVM 系統[17]、Java-MPI Package [18]等系統之研發。目前這些系統功能仍相當簡單，與本整合型計畫所規劃之功能相較，仍有許多不足之處。

目前在國內亦有多所大學及研究單位從事此一方面的研究，如台大、清華、交大、雲林技術學院、中山大學、逢甲、成大及中研院等。中央大學從事此一領域的研究已有多年之經驗，且亦有多篇研究成果發表在國際著名之期刊及會議上[19, 20, 21, 22, 23]。本整合型計畫希望根據過去的研究成果及經驗繼續發展，提供程式設計師及學術研究者一個友善 (User-Friendly) 且高效率的平行程式發展環境，以減輕開發平行程式所需花費的代價，開發高效能的平行程式。

三、研究方法

在本整合型計畫中，子計畫一負責靜態 (Static) 與動態 (Dynamic) 的資料分配分析。子計畫二負責通訊集合的產生及資料地域性的最佳化。子計畫三負責支援 SPMD 程式編譯過程中，相關通訊指令之產生及設計各種通訊延遲最小化的技術。子計畫四負責使用者與編譯器間的介面設計、視覺化資料顯示設計、資料庫設計及公用程式設計。圖一為本整合型計畫的架構圖，各計畫的工作及計畫間的關聯性可由此圖得知。

以下將各子計畫在本年度的工作做一摘要性的介紹。

1. 編譯系統的設計與製作—通訊最佳化的靜態與動態的資料分配

我們主要的研究範疇專注於靜態的資料分配暨資料對齊，以及動態的資料分配暨資料對齊，希望藉由歪斜的資料轉換及對齊的研究，可以存在更多的機會使得一迴圈程式執行時的資料通訊量達到最小。該歪斜資料分配與對齊的方法經驗證與之前的研究學者所提出的方法比較，確實有很明顯的改善程式執行效率。

2. 編譯系統的設計與製作—通訊集合產生與資料地域性最佳化

在第二年的計畫中，我們已成功找出一個程式中可以平行執行的部份，並將其依平行機器的結構來分配這些可平行執行的碼到各個處理機中，雖然這種分配程式碼及資料的方式將可保證程式在不同處理機間可平行執行以加速程式完成的時間，然而，當每個處理器在執行其所分配到的程式碼及資料時，影響其執行效率重要因素之一為陣列資料的存取與處理機多階層記憶體間的搭配是否恰當。在第三年的計畫中，我們主要的目標乃放在資料地域性的最佳化，針對安排在每個處理器的程式分析其陣列存取狀況及程式行為，並嘗試著減少階層式記憶體間資料搬動的次數，我們針對多階層記憶體來研究資料的存放位置及程式的存取行為，首先，我們將多階層的記憶體依照程式的 working set 切割，並在宣告陣列間插入適當的 Padding array，使陣列能分別依照我們所規劃的方式對應到分割後的多階層記憶體中，我們亦搭配 tiling 的技巧來同時開發程式中的 spatial locality 及 temporal locality。為顯示我們所研發的多階層記憶體管理技巧的確能增進程式執行時記憶體的存取效率，我們以 Atom 為工具來實作一模擬器以模擬多階層記憶體存取的模式，實驗顯示我們所研發的多階層記憶體及陣列的管理技術可增進資料的存取效率，程式的執行時間因而大量縮短。

3. 多處理機網路上通訊指令的編譯及最佳化

在此報告中，我們針對一個以蟲洞繞徑的二維 torus/mesh 中，考慮多點群播的問題；這裡所謂的多點群播指的是：任意的起始點個數，每一個起始點都有任意數目的終點欲做傳播。為了解決因繞徑所產生的競爭與壅塞的問題，我們提出了切割網路成子網路的方法，如此可平衡各網路鏈結的流量負載。我們討論了幾種切割網路的方法。同時，實驗模擬的結果也顯示，這樣的作法比現有在二維 torus/mesh 中的做法要好，可獲得大大的改善。

4. 在 WWW 上設計製作一個平行編譯計算環境及使用者介面

由於全球資訊網路 WWW 的盛行，建立在網際網路 Internet 下的平行執行環境及介面，提供更廣泛的使用者來使用高效能計算資源，進而推動各界人士的工作效率，是一個很重要的研究主題。本子計畫負責在網際網路下設計與製作跨平台的平行計算平台、使用者介面、及視覺化效能分析、及資料庫的維護與管理的工作。使得經整合型計劃所設計的平行編譯器所產生之平行化執行碼，可以藉由全球資訊網路下的使用者介面，在整合的平行計算平台上執行，並且及時地作視覺化的效能分析，讓使用者可以輕易地使用高效能計算資源，並且獲得計算結果與分析資料。圖二、圖三及圖四為此平行編譯計算環境的使用者操作介面。圖二為 Client 端的使用者操作介面，圖三為可動態連結任意機器的展示，圖四為效能分析圖。

由於空間限制之故，無法詳載各計畫之詳細進行步驟，欲知詳情，請參閱各子計畫之成果報告。

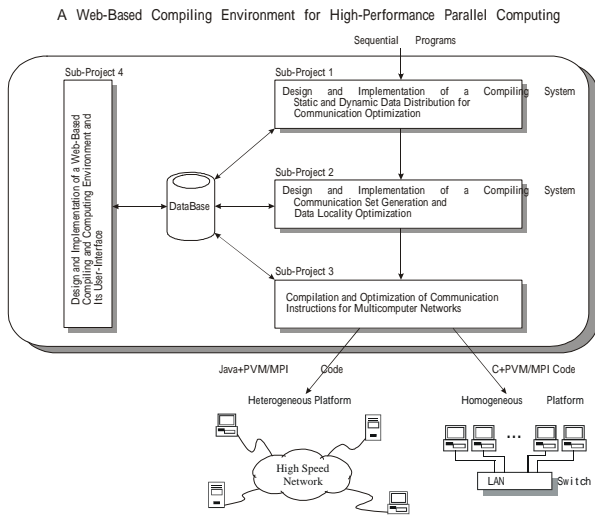
四、結論與討論

本整合型計畫主要的目的是設計並製作一具跨平台的高效能計算編譯環境。經由此環境之發展，可提供程式設計師或學術研究者一個友善且高效率的平行程式發展環境，減輕開發平行程式所需花費的代價，並開發高效能的平行程式。此一環境包含有平行化編譯器系統，資料通訊系統的產生及有效的處理機間通訊環境的製作，視覺化的使用者介面及跨平台的平行執行環境。以上幾個系統彼此互相緊密的組合並合成一個完整的平行化編譯器執行環境。

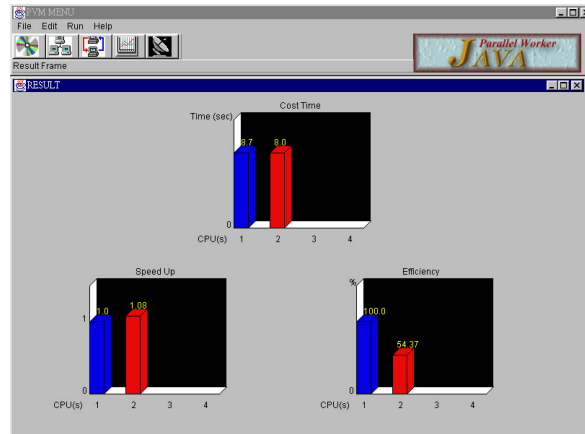
五、參考文獻

- [1] S. Hiranandani, K. Kennedy, and C. W. Tseng, "Compiling Fortran D for MIMD Distributed-Memory Machines," *Communications of the ACM*, Vol. 35, No. 8, pp. 66-80, Aug. 1992.
- [2] S. Hiranandani, K. Kennedy, and C. W. Tseng, "Evaluating Compiler Optimizations for Fortran D," *Journal of Parallel and Distributed Computing*, Vol. 21, pp. 27-45, 1994.

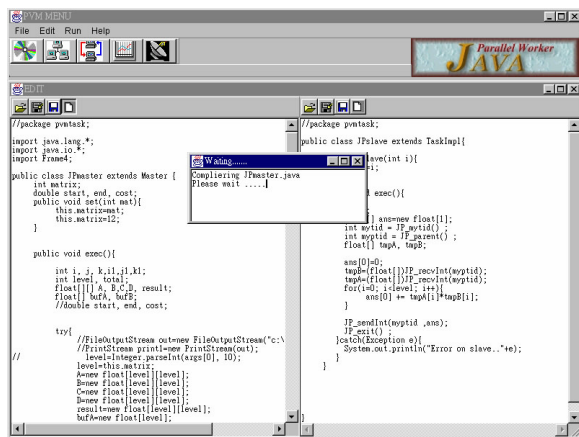
- [3] H. P. Zima, H.-J. Bast, and M. Gerndt, "SUPERB: A Tool for Semi-Automatic MIMD/SIMD Parallelization," *Paral. Comput.*, Vol. 6, No. 1, pp. 1-18, Jan. 1988.
- [4] H. P. Zima, P. Brezany, and B. M. Chapman, "SUPERB and Vienna Fortran," *Parallel Computing*, Vol. 20, pp. 1487-1517, 1994.
- [5] E. Su, A. Lain, S. Ramaswamy, D. J. Palermo, E. W. Hodges IV, and P. Banerjee, "Advanced Compilation Techniques in the PARADIGM Compiler for Distributed-Memory Multi-computers," *Proceedings of International Conference on Supercomputing '95*, pp.424-433, Jul. 1995.
- [6] S. P. Amarasinghe and M. S. Lam, "Communication Optimization and Code Generation for Distributed Memory Machines," *Proceedings of the ACM SIGPLAN'93 Conference on Programming Language Design and Implementation*, pp. 126-138, Jun. 1993.
- [7] M. T. Heath and J. A. Etheridge, "Visualization the Performance of Parallel Programs," *IEEE Software*, Vol. 8, No. 5, pp. 29-39, Sep. 1991.
- [8] G. A. Geist, M. T. Heath, B. W. Peyton, and P. H. Worley, "A User's Guide to PICL, A Portable Instrumented Communication Library," ORNL/TM-11616, Dec. 1992.
- [9] A. Beguelin, J. J. Dongarra, G. Geist, R. Manchek, and V. Sunderam, "Graphical Development Tools For Network-Based Concurrent Supercomputing," *Proceedings of Supercomputing '91*, Albuquerque, New Mexico, pp. 435-444, Nov. 1991.
- [10] L. Snyder, "Parallel Programming and the Poker Environment," *IEEE Comput.*, Vol. 17, No. 7, pp. 27-36, Jul. 1984.
- [11] D. Socha, M. L. Bailey and D. Notkin, "Voyeur: Graphical Views of Parallel Programs," *SIGPLAN Notices*, Vol. 24, No. 1, pp. 206-215, Jan. 1989.
- [12] Tain-chi Li, Chung-wen Chiang, Chungnan Lee, Tong-Yee Lee, "A Web-Based Distributed and Collaborative 3D Animation Environment", *ACM 1997 Workshop on Java for Science and Engineering Computation*, June 21-97, PPOPP, Las Vegas.
- [13] Aart J. C. Bik and Dennis B. Gannon, "Javar – a Prototype Java Restructuring Compiler," *Technical Report of Computer Science Dept., Indiana University*, July 1997.
- [14] Aart J. C. Bik and Dennis B. Gannon, "Javab – A Prototype Bytecode Parallelization Tool" *Technical Report of Computer Science Dept., Indiana University*, July 1997.
- [15] Ken Kennedy (Center for Research on Parallel Computation, Rice University) "An Environment for Compiling Java for High Performance on Servers," *ACM 1997 Workshop on Java for Science and Engineering Computation*, June 21-97, PPOPP, Las Vegas.
- [16] "JavaPVM: The Java to PVM Interface", URL <http://www.isye.gatech.edu/chmst/JavaPVM/overview.html>.
- [17] "JPVM: The Java Parallel Virtual Machine", URL <http://www.cs.virginia.edu/~ajf2j/jpvm.html>.
- [18] "Prototype Java-MPI Package", URL http://cizr.anu.edu.au/~sam/java/java_mpi_prototype.html.
- [19] T. Chen, Y.-C. Tseng, & J.-P. Sheu, "Balanced Spanning Trees in Complete and Incomplete Star Graphs", *IEEE Trans. on Parallel and Distributed Systems*, Vol. 7, No. 7, pp.717-723, Jul. 1996.
- [20] J. P. Sheu and C. Y. Chang, "Synthesizing Nested Loop Algorithms Using Nonlinear Transformation Method," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 2, No. 3, pp. 304-317, Jul. 1991.
- [21] J. P. Sheu and T. H. Tai, "Partitioning and Mapping Nested Loops on Multiprocessor Systems," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 2, No. 4, pp. 430-439, Oct. 1991.
- [22] T.-S. Chen and J.-P. Sheu, "Communication-free Data Allocation Techniques for Parallelizing Compilers on Multicomputers," *IEEE Trans. on Paral. and Distrib. Sys.*, Vol. 5, No. 9, pp. 924-938, Sep. 1994.
- [23] Y.-C. Tseng & J.-P. Sheu, "Toward Optimal Broadcast in a Star Graph Using Multiple Spanning Trees", *IEEE Trans. on Computers*, Vol. 46, No. 5, pp. 593-599, May 1997.



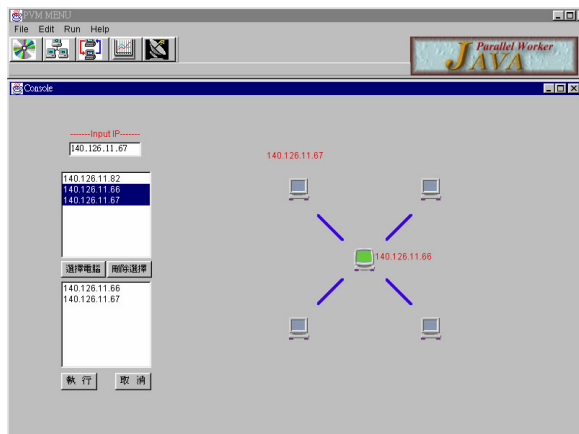
圖一：以 www 為介面的高性能平行計算編譯環境 - 架構圖



圖四：效能分析圖



圖二：Client 端的使用者操作介面



圖三：可動態連結任意機器