

行政院國家科學委員會專題研究計畫 期中進度報告

應用於嵌入式異質多核心平台之系統軟體關鍵技術與開發
工具--總計畫(1/2)
期中進度報告(精簡版)

計畫類別：整合型
計畫編號：NSC 98-2220-E-009-050-
執行期間：98年08月01日至99年07月31日
執行單位：國立交通大學資訊工程學系(所)

計畫主持人：楊武
共同主持人：單智君、雍忠

處理方式：本計畫可公開查詢

中華民國 99年06月28日

應用於嵌入式異質多核心平台之系統軟體關鍵技術與開發工具一

總計畫(1/2)

A Java Virtual Machine for Software key technologies and development tools (1/2)

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 98-2220-E-009-050

執行期間：98年8月1日至99年7月31日

計畫主持人：楊武

共同主持人：

計畫參與人員：沈柏擘、孫信慶、邱筱惠、溫致絹

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫
及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立交通大學

中華民國 99 年 6 月 14 日

摘要

本計劃探討實做一個完整支援嵌入式異質多核心系統的JVM 整合方案相關的各項重要工作及研究議題，分成以下三個階段：

1. 移植及針對系統中各個處理器優化JVM 及JIT 編譯器。
2. 探討JVM 中的並行性(concurrency)與平行性(parallelism)。
3. 整合自動工作分割機制及平行化垃圾收集演算法。

Abstract

In the duration of this plan, our project focus on the research design and implementation of a complete and integrated JVM for embedded heterogeneous multi-core systems. Our project is divided into the following three stages:

1. Porting and optimizing JVM (including the JIT compiler inside the JVM) for heterogeneous multi-core embedded systems.
2. Exploiting concurrency and parallelism in the multi-core systems.
3. Integrating an automatic task partitioning mechanism and parallel garbage collection algorithms.

前言

為了滿足對計算機系統日益增加的效能需求，目前處理器的架構主要藉由增加處理器核心(processor core)的方式來達到這個目的，這樣的架構通稱為多核心處理器(multi-core processor)。由於嵌入式(embedded)系統在功率消耗上的限制，因此嵌入式系統上的多核心處理器通常採用異質多核心(heterogeneous multi-core)的架構。其中著名的代表有TI的OMAP與IBM/SONY的Cell。使用異質多核心作嵌入式系統硬體平台的架構有越來越普遍的趨勢。

研究目的

有鑒於目前高階的嵌入式系統多採用異質多核心處理器(heterogeneous multiprocessors)，為縮短應用程式的開發人員在異質多核心平台上開發程式的時程，本計畫希望開發出一套適用於異質多核心平台的系統軟體關鍵技術與開發工具，讓應用程式的開發人員無須關心底層硬體平台所提供的資源與特色，便可以有效利用嵌入式異質多核心平台所提供的資源。

文獻探討

在 Lee et al. 的論文中[1]，描述且實作了一個轉換 OpenMP 原始碼到 CUDA 原始碼的編譯器框架，作者使用了很多種轉換策略，包括發現核心區域（kernel regions）的演算法。這個編譯器框架主要提供使用者簡單易用的程式撰寫模型來撰寫 GPU（Graphic Processing Unit）程式。藉由使用 OpenMP 當前端程式撰寫模型，該轉換器適當地轉化 OpenMP 的迴圈平行性為 CUDA 的資料平行性，使得 OpenMP 適合用來撰寫 GPU 程式。作者也使用了很多轉變技術來達到有效的存取全域記憶體（global memory），如：parallel loop-swap、針對有規則性應用程式的 matrix traspose、以及針對無規則性應用程式的 loop collapsing。實驗顯示，這些技術可使轉換出來的 CUDA 程式在效能上幾乎與由人撰寫的 CUDA 程式相同。

這篇[2]主要在描述如何透過 Polyhedral Model 來達到 C 原始碼到 OpenMP 原始碼的轉換框架，並增進程式的平行度及資料區域性來提升程式執行的效能。藉由 Polyhedral Model 的輔助，可以避免執行不同轉換時所產生的副作用，而使得產出的程式碼有較好的結果。實驗結果顯示，大部分程式可增進二至五倍效能，有些程式甚至可達十倍之多。

[5]算是最早在異質多核心處理器上探討有關於程式執行能源使用效率的可能性。文獻中利用多個 Alpha 處理器來模擬異質多核心處理器架構，每顆處理器有不同的效能規格。針對 applu benchmark 執行期間，不同的執行效率(instructions per sec)，給予轉換執行核心來達到能源使用的效率。

這篇[3]主要在描述如何透過Dynamic Voltage Frequency Scaling，在一般的多核心處理器上做不同頻率的設定以形成核心效能皆不同的異質多核心處理器。文獻中還探討了在此種多核心處理器上，程式的執行期間應該要如何選擇核心。針對這樣的問題，文獻中提出了For IPC和For Cache Miss的兩種演算法來作為核心選取的依據。

[4]總結了在執行效能皆不同的異質多核心處理器上，達到程式能源使用效率最大化的形式。主要分成兩種。一種稱為在Thread Level Parallelism上達到使用效率，程式執行可分為序列化部份和可平行化部份，將序列化的部分放在較快的核心上執行，而可平行化的部分就交給多個較慢的核心執行。而另一種型式則是在Microarchitecture上取得效率，程式可分為CPU-bound的程式和Memory-bound的程式，CPU-bound的程式使用CPU效率較高所以應該要放在效能較高的核心上執行，另一種程式則相反。

研究方法

本計畫為應用於嵌入式異質多核心平台之系統軟體關鍵技術與開發工具之總計畫，其中包含多個子計畫，以下分別描述各子計畫內容及整合。

本計畫主要分成三個子計畫，分別敘述如下：

- (1) 嵌入式異質多核心平台爪哇虛擬機器(Embedded Heterogeneous Multiprocessor Platform Java Virtual Machine, EHMPJVM)

以異質多核心處理器為平台，依照嵌入式系統中低功耗和記憶體容量受限的需求，設

計並實作一個支援動態優化能源消耗，並兼顧記憶體容量的使用以及執行效率的爪哇虛擬機器，此計畫以修改為主軸，因此只描述修改之項目架構，子項目分別為作業系統核心支援(OSKS)、JVM 動態執行核心選取(DECS)以及 JVM 功耗效能測寫(JPPR)，子項目架構如圖 1 至圖 3。

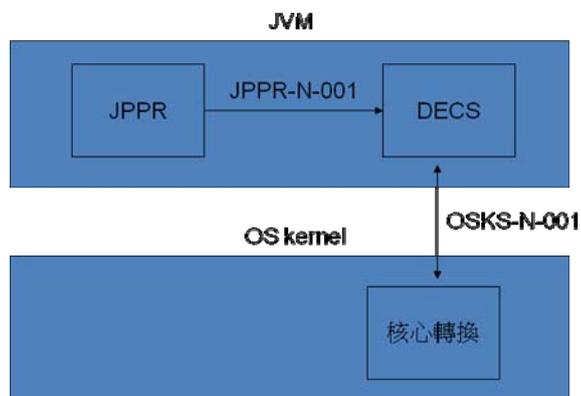


圖1：OSKS項目架構圖

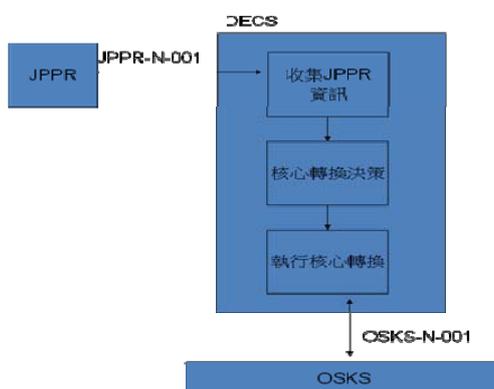


圖2：DECS項目架構圖

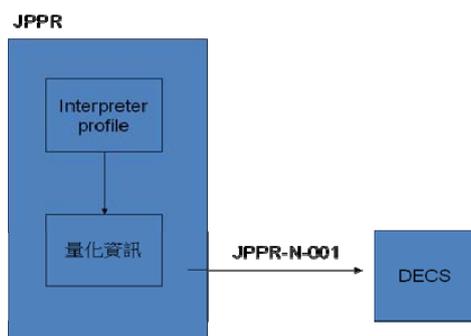


圖3：JPPR項目架構圖

(2) 異質多核心處理器靜態程式碼最佳化與轉換系統 (Heterogeneous Multi-Core Static code Optimization and Translation system, HMCSOT)

HMCSOT 將使用 OpenMP 撰寫的程式碼自動轉成 OpenCL 的程式碼，並對需要進行轉換的核心函式進行記憶體存取優化。藉由本計畫所開發的工具，程式開發者可以使用 OpenMP 來開發 heterogeneous multi-core 程式，藉以縮短程式開發者在發展 heterogeneous multi-core 程式時的時程，子計畫架構如圖 4。

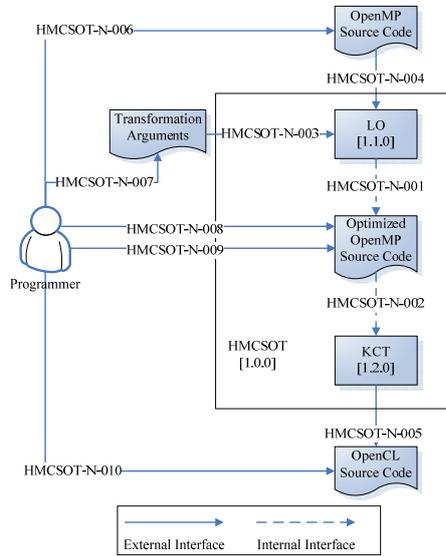


圖 4：HMCSOT 系統架構圖

(3) 垃圾收集平台 (The Garbage Collecting Platform, GARCOL)

當物件不再被使用時，它所佔用的記憶體空間便需歸還給Java執行環境。Java程式語言提供了自動的記憶體回收管理機制，即「垃圾收集」(garbage collector)。程式開發者不需再親自處理記憶體的控管，可專注於開發應用程式，也可減低系統在執行期間的出錯機率，子計畫架構如圖5。

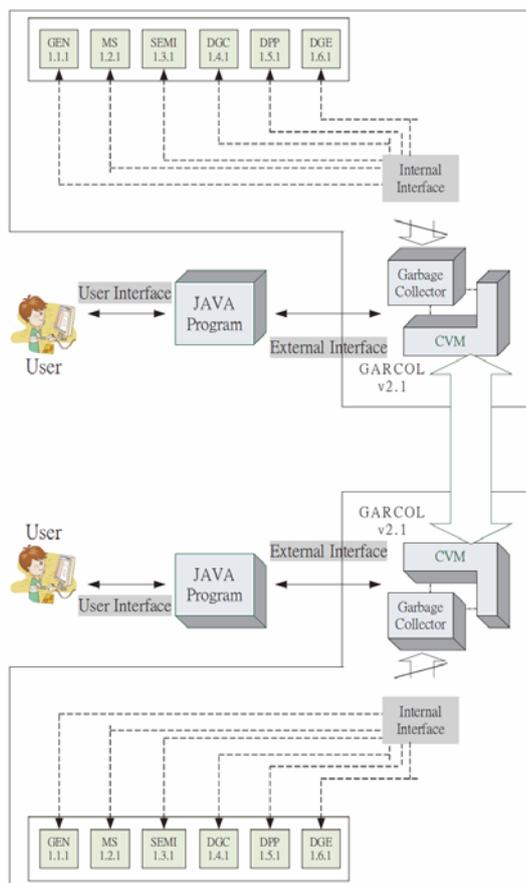


圖 5：GARCOL 系統架構圖

各子系統整合後之互動，用以下之 Activity Diagram 來描述 SKTDTJVM 因外界觸發，而造成各個子系統的反應。

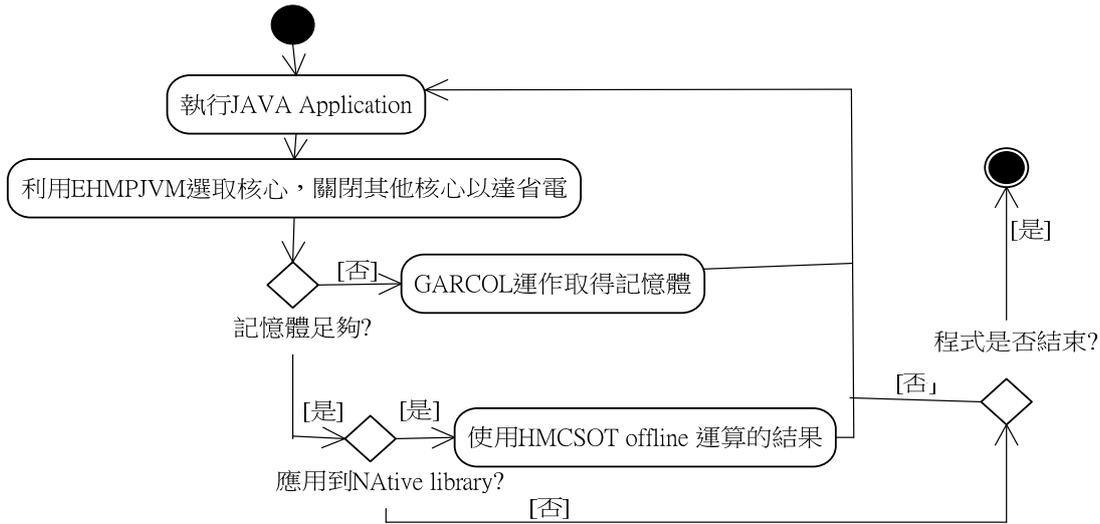


圖 6： Activity Diagram

結果與討論

OpenMP 程式容易撰寫而 OpenCL 程式不好撰寫，勢必要有一些有用的工具，轉換 OpenMP 原始碼到 OpenCL 原始碼，來減輕開發異質多處理器程式的難度。轉換過程中的最佳化首重記憶體存取，不適當的記憶體存取方式將會嚴重影響效能。

今年度的進度將會完成這個系統，而明年度的進度將會進一步使用與修改這個系統，來轉換更複雜的 OpenMP 程式，如：OpenMP-based H.264 decoder。

透過自動化的選擇執行核心，在程式不需要大量運算資源供應執行時，轉換執行核心到較低效能的核心上，本系統將使程式更有效率的使用資源。

今年度的進度將會完成這個系統的雛型，而明年度的進度將會進一步使用與修改這個系統，來實作更複雜核心選取機制，以期望能達到更有效率的程式執行。

參考文獻

- [1] Seyong Lee, Seung-Jai Min, and Rudolf Eigenmann. OpenMP to GPGPU: A Compiler Framework for Automatic Translation and Optimization. In PPOPP'09, 2009.
- [2] Uday Bondhugula, A. Hartono, J. Ramanujan, P. Sadayappan. A Practical Automatic Polyhedral Parallelizer and Locality Optimizer. In ACM SIGPLAN PLDI, 2008.
- [3] Rangan, Krishna K. and Wei, Gu-Yeon and Brooks, David . Thread motion: fine-grained power management for multi-core systems, SIGARCH Comput. Archit. News, 2009
- [4] Fedorova, Alexandra and Saez, Juan Carlos and Shelepov, Daniel and Prieto, Manuel. Maximizing power efficiency with asymmetric multicore systems Commun. ACM ,2009
- [5] R. Kumar, D. M. Tullsen, P. Ranganathan, N. P. Jouppi, and K. I. Farkas. Single-ISA heterogeneous multi-core architectures: the potential for processor power reduction. In Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, pages 81–92, Dec. 2003