

Matlab 上使用 GPU

吳俊育¹、蔡至清²、陳焜燦²

¹ 國家高速網路與計算中心

² 國立中興大學應用數學所

adherelinux@hotmail.com

jet.tsai98004@gmail.com

ktchen@amath.nchu.edu.tw

摘要

本篇論文主要是使用 Matlab 軟體利用 Nvidia Tesla c1060 顯示卡做運算。使用連結套件軟體如 GPUmat、Jacket 兩種做效能的測試。舉例一個 $N \times N$ 的矩陣相乘討論尺度的大小所花的計算時間，並討論哪軟體的效能比較優越。

關鍵詞： NVIDIA、GPUmat、Jacket。

Abstract

This paper use Matlab software for Nvidia Tesla c1060 on computing. We test performance on two toolbox. The subject is n by n of matrix multiply. We considered different scales for execution time. We observed more performance between two toolbox.

Keywords: NVIDIA、GPUmat、Jacket.

1. 前言

Matlab 是由 Math Works 公司於 1984 年所開發出來的一套軟體，主要是一個矩陣運算的軟體，它不但擅長矩陣數值運算，也適合其他的科學運算，它支援多種作業系統 Linux、Windows。然而目前 Matlab 提供很多的工具可以方便使用，目前已經許多人投入使用中。但 Matlab 是個直譯式程式語言在進行大量計算的時候，常常要花費很多時間做運算以至於利用許多軟體連結其他設備做運算。例如：使用顯示卡做運算。

GPGPU(General Purpose Graphic Unit): 主要是以繪圖晶片進行通用的運算，由於新型的繪圖晶片開始允許可程式性，讓原本被鎖定用於繪圖運算的繪圖晶片，開始可執行通用運算。一般而言，中央處理器負責大部分的運算工作，也是所謂的通用運算，而顯示卡晶片專門處理影像，彼此的分工明確。GPU 多核心的架構，比起傳統 CPU 架構帶來更強的處理能力，人類對於運算的需求越來越大，如核能試爆、氣候預測與病毒防治，都需要大規模的運算能力，但傳統的中央處理器架構，以英特爾的 Xeon 處理器為例，最高只有 8 個核心，然而一

顆 Nvidia Geforce 9800 GTX 繪圖晶片，卻擁有 128 個處理核心，由於多核心的架構，更適合超級電腦的平行運算運用。以華碩推出的 GPU 伺服器，內含 Nvidia TeslaC1060 繪圖晶片，就擁有 240 個運算核心。

在 2001 微軟推出 DirectX9，讓繪圖處理器也能用浮點進行數學運算，讓繪圖晶片可程式化，開始做為通用運算之用，再加上，繪圖晶片擁有數量龐大的核心數，針對平行運算應用，帶來更為強大的浮點運算能力，讓 GPGPU 就因此而誕生，利用繪圖核心來處理原本中央處理器負責的通用運算任務。

2006 年 Nvidia 公司編寫一套編譯器 NVCC 得函式庫，NVCC 是處理器與編譯器的混合體。當遇到 CUDA 程式碼，編譯器會自動編譯為 GPU 的代碼，也就是會轉換為 CUDA Driver 的代碼。若是 Host Host C++ 代碼，則用 C++ 編譯器進行編譯。

CUDA 軟體架構，我們舉例 Nvidia 公司的產品 Nvidia 9-Series Architecture (Geforce 9800 GT) 為範例。

硬體架構主要區分每個 multiprocessors，有 8 個 thread processors。而 9800GT 的產品，共有 16 multiprocessors，所以總共 128 thread processors。圖一說明硬體架構。

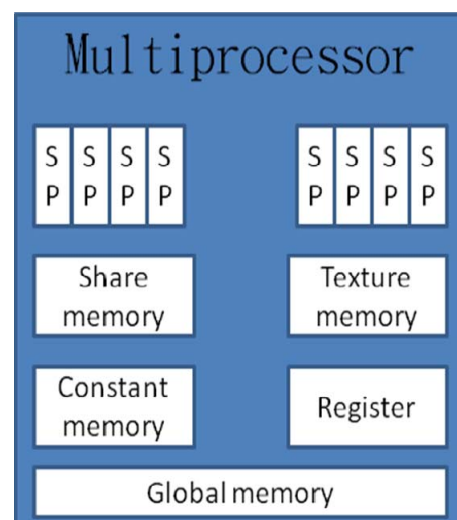


圖 1 硬體架構

然而 Nvidia 公司利用在軟體方面區分為 Grid、Thread Block、Thread 三類。Grid 對應 Device，Thread Block 對應 Multiprocessor，Thread 對應 Thread Processor。圖二說明軟硬體配置。

Software	Hardware
Thread	Thread Processor
Thread Block	Multiprocessor
Grid	Device

圖 2 軟硬體的配置

2. 圖形處理器的工具

目前有幾套軟體可以讓 Matlab 執行 GPU 作運算，其中有 GPUmat、Jacket。

2.1 Jacket

AccelerEyes 公司 2008 年 6 月提出 Jacket 1.1 版本。Jacket 為 MATLAB 在 GPU(繪圖處理器)運算的橋樑，圖三說明 Matlab 與 GPU 如何連結。主要 Jacket 1.1 版本有支援雙精算(Double-precision)。雖然目前 Jacket 1.1 還有許多 Matlab 的語法尚未支援，但未來會陸陸續續的改善。

Jacket 適合很多的應用，例如：醫學影像、金融業、汽車製造業等等。此外 Jacket 的圖形工具包含許多做視覺計算的功能。

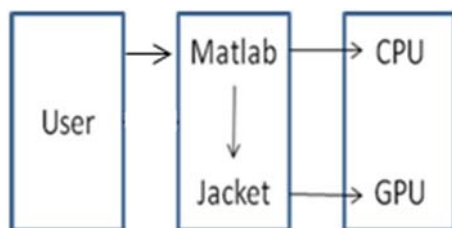


圖 3 Matlab 與 GPU 連結

2.2 GPUmat

GPUmat 軟體是由 GP-you 團隊所發展的一套工具軟體，使得我們可以很容易、方便利用 GPU 做計算。近幾年在嵌入式系統、數值模擬的軟體開發上有所進展。GPU 屬於開放性軟體。

3. 測試環境

硬體：

CPU：Intel (R) Xeon (R) Quad Core 2 E5520 2.27 GHz

RAM：12G DDR3

GPU：Tesla C1060

作業系統：Ubuntu 9.04 64 bit

CUDA 版本：CUDA Tool Kit 2.3

軟體：Matlab R2010a、Jacket 1.3(試用版)、GPUmat 1.1

4. 效能的測試結果

$$\text{矩陣產生: Hilbert Matrix } a_{ij} = \frac{1}{i+j-1}$$

時間由 tic toc 計算，包括從 CPU host memory 寫入 GPU global memory，相乘，寫回 host memory 的時間。無平行化 C 程式語言在 $N = 2048$ 約 160 s， $N = 4096$ 約 1518 s。

4.1 Matlab 的效能

表 1，N: 矩陣大小，利用雙精確浮點數做運算所花的時間。

表 1 Matlab 效能

N*N	Matlab (s)
2048*2048	0.487
4096*4096	3.829
6144*6144	12.837
8192*8192	30.582
10240*10240	59.266
12288*12288	104.274
14336*14336	164.255
16384*16384	246.129

4.2 Matlab + Jacket

表 1，N: 矩陣大小，利用單倍精確浮點數做運算所花的時間。

表 2 Matlab + Jacket 單倍精確浮點數的效能

N*N	Jacket(s)
2048*2048	0.069
4096*4096	0.470
6144*6144	1.487
8192*8192	3.342
10240*10240	6.350
12288*12288	11.786
14336*14336	17.044
16384*16384	25.088

4.3 Matlab + Jacket

表 3, N: 矩陣大小, 雙精確浮點數做運算所花的時間。

表 3 Matlab + Jacket 雙精確浮點數的效能

N*N	Jacket(s)
2048*2048	0.267
4096*4096	1.927
6144*6144	6.388
8192*8192	14.983
10240*10240	29.484
12288*12288	50.241
14336*14336	80.568

4.4 Matlab +GPUmat

表 4, N: 矩陣大小, 利用單倍精確浮點數做運算所花的時間。

表 4 Matlab +GPUmat 單倍精確浮點數的效能

N*N	GPUmat(s)
2048*2048	0.087
4096*4096	0.467
6144*6144	1.399
8192*8192	3.215
10240*10240	6.158
12288*12288	10.546
14336*14336	16.668
16384*16384	24.646

4.5 Matlab +GPUmat

表 5, N: 矩陣大小, 利用雙精確浮點數做運算所花的時間。其中 X 代表記憶體不足。

表 5 Matlab +GPUmat 雙精確浮點數的效能

N*N	GPUmat(s)
2048*2048	0.239
4096*4096	1.853
6144*6144	6.212
8192*8192	14.706
10240*10240	28.632
12288*12288	49.408
14336*14336	78.422

4.6 CUDA 平行程式

撰寫 CUDA 平行程式, 使用單倍精確浮點數做運算, 並且利用 shared memory 進行加速。表 6, N: 矩陣大小。

表 6 CUDA 平行程式單倍精確浮點數的效能

N*N	GPUmat(s)
2048*2048	0.17

4096*4096	0.63
6144*6144	1.88
8192*8192	4.43
10240*10240	8.96
12288*12288	16.99
14336*14336	30.09
16384*16384	49.91

5. 結論

Matlab 對於工程上使用起來是非常方便的, 但是由於一些大型問題不適合在 Matlab 做運算。在這裏用相同的矩陣大小且除了 Matlab 本身為雙精確浮點數, 其餘為單倍精確浮點數的效能做為比較, 可見圖 4。我們比較 Jacket 與 GPUmat 其實效能差異性並不大。目前 Jacket 比 GPUmat 的函數庫還多, 也比較成熟。撰寫 CUDA 平行程式效能雖然沒有其他兩個還要好。但是懂得 C 語言就可以對任意的問題做平行, 至於 Jacket 或者 GPUmat 了解內建指令即可撰寫。我們可以看出 GPUmat 速度上效能較好, 而且非常適合初學者使用。

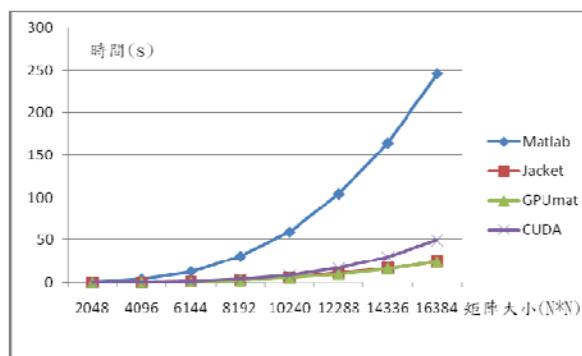


圖 4 效能比較

參考文獻

- [1] Nvidia CUDA Programming Guide 2.3, August, 2009, Nvidia company.
- [2] Nvidia CUDA Programming Best Practices Guide 2.3, August, 2009, Nvidia company.
- [3] Getting Started Guide, 2010, AccelerEyes company.
- [4] Jacket Function Reference, AccelerEyes company, 2010, AccelerEyes company.
- [5] GPUmat User Guide Version 0.25, GP-you, April 2010.
- [6] GPUmat User Modules Version 0.15, GP-you, April 2010.
- [7] GPUmat User Modules Reference Version 0.15, GP-you, April 2010.
- [8] MATLAB 程式設計與應用, 張智星.