高性能矩陣計算淺介

莊　重

由於過去 25 年電腦向量和平行處理的進展，因此數值分析家目前的重要挑戰之一是設計程式和處理計算使得軟體的設計能配合硬體的進展。傳統上在評估程式的表現最主要的指標是計算的次數（ the amount of arithmetic ）。然而對向量化和平行化的電腦來說，計算的次數雖然仍是重要的，但是資料的流通(the flow of data)的情形至少是被一樣重視的。所謂高性能矩陣計算，即是對資料流通付出至少一樣的關心。事實上，有些最好的 sequential 程式從向量化和平行化的觀點來看是不令人滿意，應該被修正或是淘汰。反之，有些程式從 sequential 的角度來看不是頂好的，但却具有自然平行的特性。又比方說處理高斯消去法（ Gaussian elimination ）時，若對memory traffic 小心安排將是十倍於不細心安排的速度。也因此很自然的，時下我們對矩陣計算的性能（ performance ）的要求是很高的。本文主要目的是舉些例子，使得讀者在唸矩陣時能夠對資料的流通，也即性能的表現，更加關注。當然本文只是拋磚引玉的工作，資料流通也絕不是下面的例子可完全表達，有興趣的讀者可參考〔1～4〕。

第一個例子表達的是相同結果的程式，不同的Do loops 安排方式，却有截然不同的性能表現。

```
(A1)   Do   100   i=1,N
         Do   100   j=1,N
100           B(i)=B(i)+A(i,j)
(A2)   Do   100   j=1,N
         Do   100   i=1,N
100           B(i)=B(i)+A(i,j)
```

(A1)和(A2)從 sequential 的觀點來看是相同的，但從向量計算來看是不同的。(A1)的 loops 得到的只是 scalar 的加法。(A2)可以得到一個向量化的向量加法指令。用一個簡單的比喻來說明：(A1)和(A2)的任務(程式結果)是裝配車(1)，…車(100)，共一百輛車。(A1)的策略是令所有的工人去裝配車(1)，裝完後，所有的工人再去裝車(2)，……，然後車(100)。(A2)的作法是有一條裝配線，車(1)，……，車(100)可以同時進行裝配。如此(A1)和(A2)的好壞眞是不言可喻了。

第二個例子是矩陣(A)乘向量(x)的二種不同 loops 的寫法：

(B1)：列計算

$z(1:m)=0$

(initialing z to be the $m\times 1$ zero vector)

Do 100 $i=1:n$

Do 100 $j=1:m$

100 $z(i)=z(i)+A(i,j)x(j)$

(B2)：行計算

$z(1:m)=0$

Do 100 $j=1:n$

Do 100 $i=1:m$

100 $z(i)=z(i)+x(j)A(i,j)$

同樣的，B(1)和B(2) 若以結果和計算次數來看是完全一樣的。但是 arrays 在 Fortran 上是以行的順序來儲存的。舉例來說如果一個矩陣 A 儲存在一個 100×50 的 array，那麼 $A(4,2)$ 和 $A(5,2)$ 是鄰居，可是 $A(4,2)$ 和 $A(4,1)$ 是離 100 個記憶單位。打個比方：假使系上檔案都以號碼按順序建檔。對一位秘書來說，要拿出相鄰的 100 個檔案比要拿出 2, 12, 22, ……, 982, 992 號的檔案容易多了，想必電腦也應有同感。細心的你或妳是否可以判斷出(B1)和(B2)分別指那一種情形呢？從而指出那一個程式在 Fortran 上是較 efficient 的。在矩陣的計算中了解程式的結構改變(如上述兩個例子 loops 的次序交換)是相對於原先矩陣的什麼改變是重要的。

我們希望這二個簡單的例子，能讓有興趣的讀者了解資料的流通，確實因爲向量化和平行化電腦的興起，而成爲衡量程式好壞的重要指標。目前，這方面的研究是非常熱門的。

參考書目

1. C. F. Van Loan, A Survey of Matrix Computations, Cornell Theory Center, 1990.
2. G. H. Golub and C. F. Van Loan, Matrix Computation, 2nd ed., The John Hopkins University Press, Baltimore, 1989.
3. R. H. Hockney and C. Jesshope, Parallel Computers: Architecture, Programming and Algorithms. Adam Hilger, Bristol, 1981.
4. J. M. Ortega and R. G. Voigt, Solution of Partial Differential Equations on Vector and Parallel Computers. SIAM Review, Vol.27, No.2, 1985, 149～240.

——本文作者任教於交通大學應數系、所——