

科技文化

# 從計算到思維？

## ——談並行電腦的發展與前景

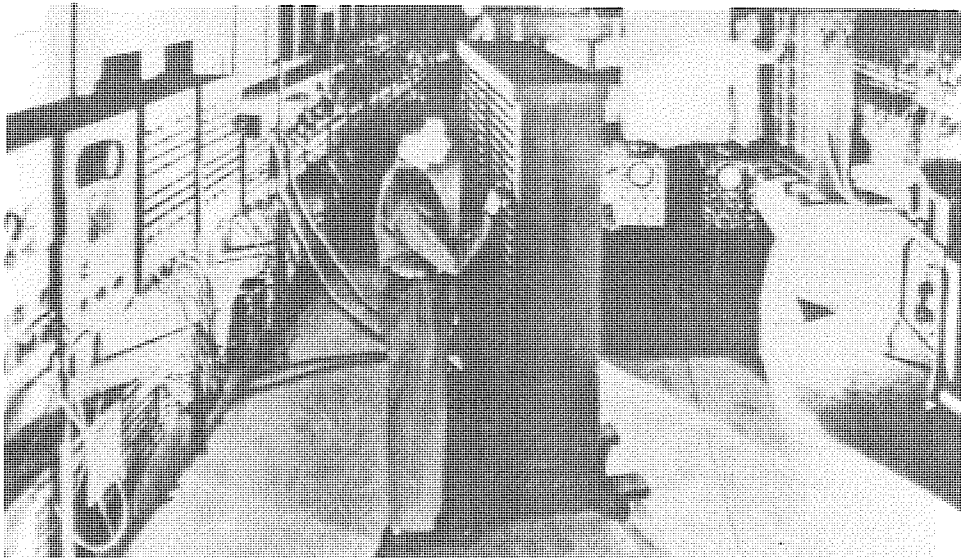
● 趙挺偉

電子計算機已成為我們生活中這麼密切的一部分，以致我們忘記了它的結構可以說是完全由一個人對計算程序組織的看法而決定的。

I. Aleksander and H. Morton<sup>①</sup>

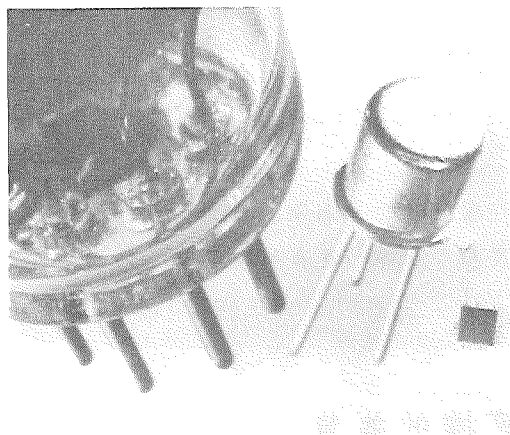
### 一 計算機的蛻變

自從 40 年代初賓夕凡尼亞大學的 ENIAC (Electronic Numerical Integrater And Computer) 出現以來(圖一)，電子計算機至今已經有將近半世紀歷史了。在這期間，計算機的改進、蛻變之神速超過任何人的夢想：它從價值數百萬美元，體積佔據整幢房子那麼大的龐然巨物，變成了只值一千幾百元，可以放進公事包裹的小匣子，從科學家、工程師、原子彈設計者、彈道專家的稀有和獨特儀器變成了辦公室、銀行、乃至商場中不可或缺的普通工具：



圖一 第一部電子數碼計算機ENIAC，是二次大戰期間美國陸軍部用以計算炮彈彈道的。

**字節 byte** 這是計算機記憶體容量的基本單位，每一字節普通有32個或64個單元bits(每單元代表二進數的一位，即是0或1)，也就是說可以表達任何二進制中的32位或64位數。MB (megabyte)指 $10^6$ 字節，KB (kilobyte)指 $10^3$ 字節。

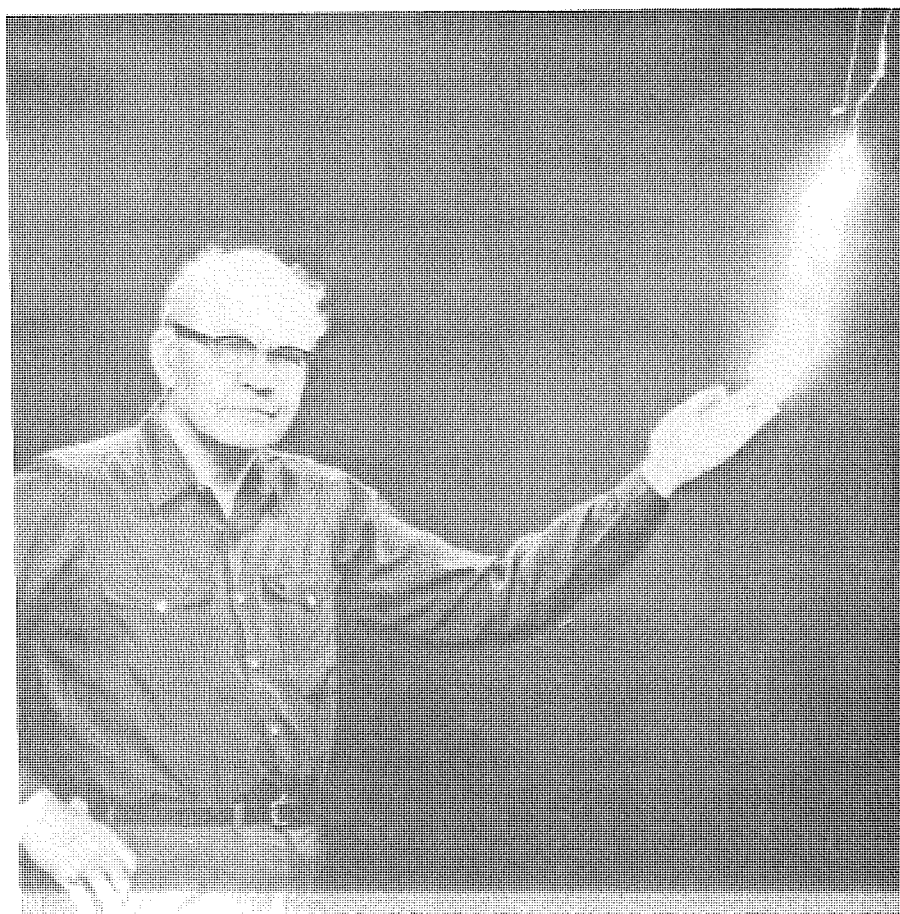


圖二 真空管(左)、晶體管(中)和微積集電路硅片(右)

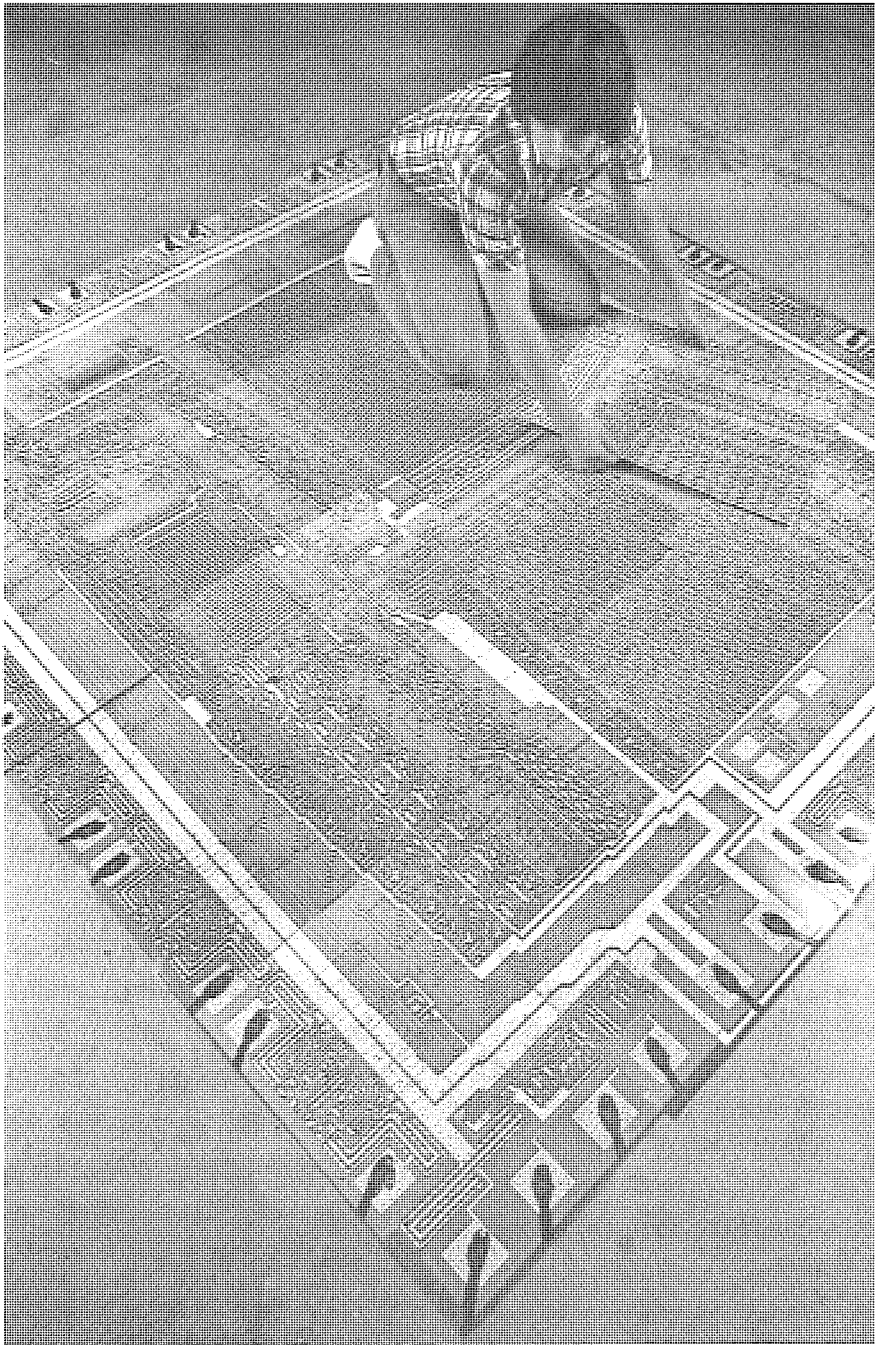
它的速度、容量也以成千上萬的倍數提高：今天坐在動輒有上「百萬字節」(MB)記憶元素的個人計算機前的秘書小姐，大概難以想像三十年前幾十甚至上百位科學家共用一部只有64KB記憶的所謂「主計算機」(mainframe computer，譬如IBM709型機)是怎麼一回事吧！

在這四十五年間，計算機的進步基本上是元件的進步。40年代最早期的計算機是憑機械性的電磁開關和熱離子性的真空管運作的，它們不但佔據大量空間，耗費大量能量，而且運作速度大概只有每秒數千次左右。到50年代，晶體管和磁芯(magnetic core)記憶元素的出現(圖二)大大降低了計算機的體積、電流和能耗，同時，基本上由晶體

管開關時間(switching time)決定的運作速度提高到每秒百萬次(MOPS)的範圍，這就是1952-63年間出現的第二代計算機②。



圖三 堪薩斯州木訥寡言的 Jack Kilby：他的發明有如雷電，震撼了電子工業界。



圖四 只有1平方厘米的積集電路顯微照片放大二、三百倍後充分顯示出它的複雜性來。

1958年德薩斯儀器公司(Texas Instrument)的基爾比(Jack Kilby)(圖三)和費爾差(Fairchild)公司的奈司(Robert Noyce)發明了在同一塊硅片上製造不同元件——晶體管、電阻、電容等等——並且把它們連結成電路，即所謂「集成電路」(integrated circuit)(圖四)，掀起了1962-75年間大量利用小型和中型集成電路板的第三代計算機革命。本來，改進電腦功能有一個基本矛盾：若要它解決大型、複雜的問題，那麼必須增加元件數目；若要它快速運算，那麼訊號必須能迅速傳遞，也就是說元件必須互相接近。但大小固定的元件多了就不能

互相接近，而且太接近時所產生的熱能無法耗散。集成電路的特點就是隨着技術進步，元件的面積不斷減小，因此，在同一塊硅片上，元件的數目不斷增加。這樣，就解決了上述矛盾，達到同時增加計算機容量和速度這兩個目的；並且，由於大規模生產技術的進步，也造成了價格的大幅下跌。

到了70和80年代，由於精密光蝕刻(photolithographic)技術的開發，大型(LSI)和超大型集成(VLSI)電路出現，在1平方厘米的硅片上元件數目竟可以超過百萬，甚至達到4百萬(即每一元件只有幾個微米 $10^{-6}$ m寬度)，運算速度達到數百MOPS的「超級計算機」(supercomputer)陸續面世，計算機的蛻變也就進入第四代了。

## 二 一個人的構想

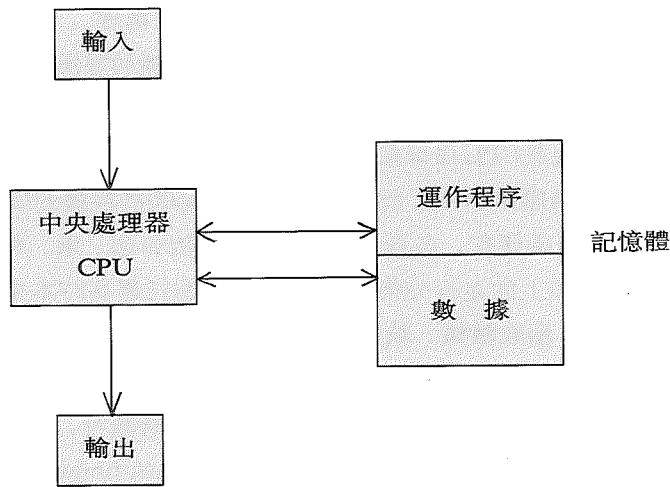
上文說：在過去將近半世紀，計算機的進步基本上是元件的進步；這也就是說：它的結構(architecture)未曾有基本改變。

MOPS, MIPS, MFLOPS 都是計算機運算速度的指標，分別指 $10^6$ 基本運作(operations) / 秒， $10^6$ 指令(instructions) / 秒和 $10^6$ 浮點運算(floating-point operations) / 秒。由於不同計算機的指令長度和複雜性不同，用以處理浮點算術運作的硬件也不一樣，所以這三者所標示的性能是有差別的。而且，運算速度和所用的測試程序有密切關係，因此指標只能作為粗略比較的依據。此外，也還有GFLOPS和TFLOPS(分別指 $10^9$  / 秒和 $10^{12}$  / 秒)的用法。

在1944年，普林斯頓高等學術研究所天才橫溢的匈牙利籍猶太數學家范奈曼(John von Neumann, 1903–57)因為偶然機緣參加了ENIAC的發展工作，其後在他領導下很快就奠定了現代電子計算機的具體構想(事實上同時的英國數學家都靈[Alan Turing, 1912–53]貢獻也很大)③，例如它內部以二進制表示所有數目，以數目表示所有數據、文字以及計算指令，「計算程序」(computational programme)是儲存在電子計算機記憶系統之內並以數目代表的一系列指令，程序可以任意設計和更換等等，當時都已經預見了。

這構想中所謂「循序處理」(sequential processing)原則是有決定重要性的。根據這原則，無論計算多麼繁複——例如從風力和流體力學原則計算炮彈軌道，或者模擬大洋中海流的變動，我們都要先把它分拆成依循先後次序(sequence)排列的基本運算步驟，這就是所謂「運算程序」(computational programme)。計算機的核心是「中央處理器」(central processing unit 或 CPU)，它的功能就是從「記憶體」(memory)中循序抽取程序中的指令，根據指令和記憶體其他部分供給的數據進行基本運算和報告結果，然後再抽取下一項指令，

根據「循序處理」原則，無論計算多麼繁複，我們都要先把它分拆成依循先後次序排列的基本運算步驟。



圖五 循序計算機主要功能單位示意圖

以迄程序完畢為止(圖五)。Aleksander和Morton在文首引言所指的，就是這樣一個在CPU控制下的「循序計算機」(sequential computer)的構想。

這構想有簡單、合理和易行的優點：單線計算程序不但設計容易，而且偵錯、改正也容易。從40到80年代，計算機不斷蛻變，它的結構卻始終未曾擺脫「循序計算」這基本原則。一直到80年代中期，情況才漸漸改變，「並行」(parallel)這醞釀已久的革命性新思想才發芽滋長。

### 三 「並行處理」的革命

集成電路的不斷改進使計算機記憶體的容量爆炸性地增長——事實上從70年代初開始大約每三年就翻兩番，即增為4倍，因此「中央處理器」的整體運作速度必須相應增長，以免成為在記憶體中大量數據的「瓶頸」(bottleneck)<sup>④</sup>。同時，計算機的廣泛應用，使它面對的問題從繁複但數據不一定很多的數學轉到可能很簡單但包含大量數據的日常事物，例如上百架同時在機場四周盤旋的飛機航線，或者包含了數千萬畫面元素(pixel)的衛星圖象等等。很明顯地，將所有這些「同時性」的數據強為排列先後，順序處理，那是沒有效率和不合理的辦法。而且，所需的速度也往往超出單個CPU的能力。

所以，60和70年代的微型化解決了記憶容量「更多」的要求，同時也凸顯了相應的計算速度「更快」的要求。

要「更快」可以循兩條完全不同的途徑。第一條途徑，也可說是最基本的辦法，是除去緩慢的根源——即電子和它的惰性質量。也就是說，不再用電量代表數目(基本的0和1)或以電流傳遞訊息，改用速度更快，並且不會彼此干擾的光子作為計算機中的活動媒體(active medium)。事實上，浮點運算速度估計可以達到TFLOPS  $10^{12}$ 次/秒的「光子計算機」(optical computer)已經研究了多年，並且原則上證實可行。但它的實際出現，恐怕還得等待下一世紀。

60和70年代的微型化解決了記憶容量「更多」的要求，同時也凸顯了相應的計算速度「更快」的要求。

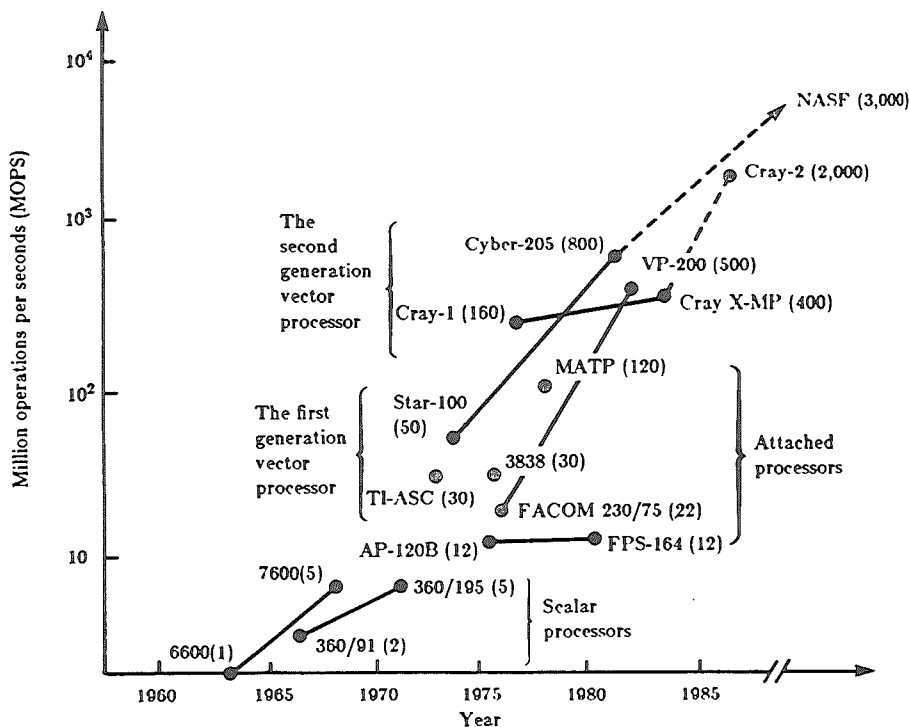
要「更快」，最基本的辦法，是除去緩慢的根源——即電子和它的惰性質量。

第二條途徑就是放棄「循序處理」原則，改用「並行處理」，換而言之，就是用幾個甚至成千上萬個處理器同時運作，以增加計算機的整體有效運算速度。

其實，「並行處理」原則在60年代就已經被應用了，但當時它並不抵觸CPU的觀念，而表現於CPU內部結構的多元化、複雜化。例如CPU原本只有單獨一個算術和邏輯運作單位，但1964年的CDC6600計算機的CPU內就有十個功能單位，它們同時運作，各自專司一種計算(例如加、乘、移位，等等)，這是處理過程在空間上的並行分佈。又例如CPU進行運算時本來必須依次執行抽取指令、釋碼、抽取數據、執行這四個不同步驟，但1966年的IBM360/91型計算機的CPU內，則有不同的特殊單位以工廠流水作業形式，同時分別處理這些步驟，也就是說，多個指令的不同部分可以同時執行。而且，同樣原理還用以處理本來極為緩慢的浮點運算(floating-point arithmetic)和輸入／輸出等其他運作。這樣就大大加快了CPU整體運作的速度。這種稱為「管道化」(pipelining)的策略，是處理過程在時間上的並行分佈。70年代出現的「超級計算機」(supercomputer)例如 Cray-1, Cyber 205 等，其驚人的運算速度基本上就來自大量利用多功能化和管道化(這包括設計能實施「向量式，vectorised」計算的處理器)這兩個原則不斷改進的CPU。

在1966-86的二十年間，CPU的內部改進使計算機的整體速度提高數百倍(圖六)，因此CPU本身逐漸發展成龐大複雜的結構。到80年代中期，沿同樣方向繼續改進的經濟效率已開始降低；況且，這時超大型集成技術已經非常普遍，所以「並行化」就從CPU內部轉向計算機整體，促成「並行計算機」的出現了。

圖六 1960-85年間  
超級計算機速度的增長



## 四 並行計算機的發展

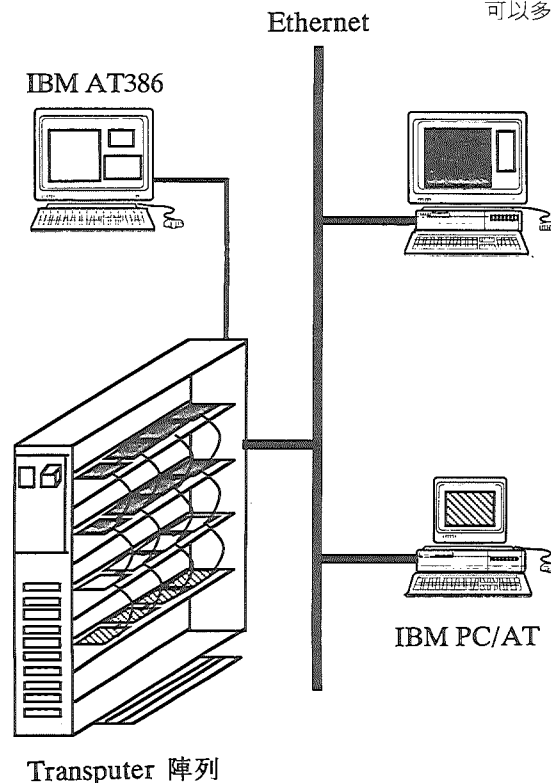
甚麼是並行計算機？「傳統」計算機基本上是「個體戶」：它的CPU要兼顧「統籌」（包括輸入／輸出、抽取指令、解碼、取存數據等）和「運算」這兩大類工作。相比之下，並行計算機則是個「集體」：它有專司「統籌」的「控制單位」（control unit 或 CU），這通常就是一部小型傳統電腦；實際進行運算，則是在CU控制和協調下的計算網絡中進行。網絡由多個「節點」（node），即附有記憶體的處理單元（processing element 或 PE）互相聯絡組成，所以又稱「處理器陣列」。並行計算系統的控制單位一般固定，但PE陣列的節點數和節點連結方式則可以因應需要而改變。

其實，第一部並行計算機是二十年前出現的Illiacc-IV，它具有64個PE，但程序設計非常困難，所以並不成功。十年前，美國太空總署為處理衛星圖象，又特地建造了一部有16,384個PE的「MPP計算機」，但它只是一部特殊功能處理器，而並非通用計算機。並行計算機的發展和普及，還是過去十年間的事。

說來奇怪，在這十年間，推動並行計算機發展的一股動力卻是高能理論物理學家。他們面對的一個長期困擾是：描述粒子基本作用的「場論」（field theory）除了在特殊情況（例如只有電磁作用的情況）外，無法求解析（analytic）解。為了突破這個數學障礙，威爾遜（K. Wilson）在1974年提出「點格場論」（lattice field theory）。他把連續時空體上具有無限自由度的粒子場，簡化為點格上只具有有限自由度的點格場，並且證明在特定條件下兩者有相同性質。從80年代起，點格場數值解的研究就蓬勃起來，並造成兩個意想不到的後果：第一，高能物理學者，特別是威爾遜本人的奔走呼籲，引起美國政府和各大計算機公司對超高速計算系統發展的關注，其後全美超級計算機網絡的建立與此不無關係。第二，由於解決點格場問題所需的計算能力，其實超越了傳統超級計算機的速度，而且後者又委實太昂貴，引致高能物理學者着手利用當時已經極端普遍的微處理器（microprocessor），來自己動手設計和建造物美價廉的並行計算機。這「自己動手製」（do-it-yourself）的並行機大半是為特殊用途而製，但速度竟可以達到5 GFLOPS（例如哥倫比亞大學N. Chris所用的），已

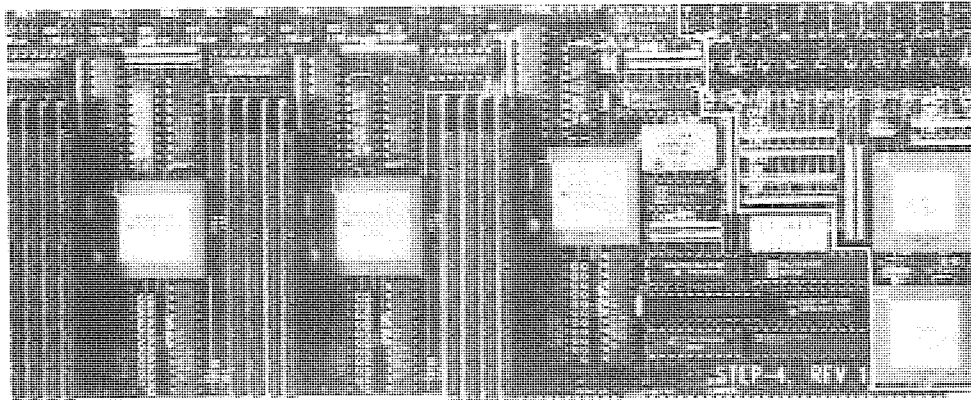
「傳統」計算機基本上是「個體戶」；並行計算機則是個「集體」。

圖七 一個簡單的並行計算系統：左上是作為「控制單位」的IBM AT386個人計算機，左下是它控制的由16個分置4塊聯結板上的Transputer所組成的處理器陣列。通過通訊網絡（Ethernet），這系統可以多人共用。



經遠超過傳統超級計算機。然而，這還未能滿足晶格場論的需要：現在仍然有好幾個高能理論組在努力建造更優越的並行機，希望達到 TFLOPS 的領域。他們的信念是：任何能有助於了解物理的事物，不論是先進電子技術抑或抽象數學，他們都能學習、創新、改革和應用。

圖七所示，就是一個「自己動手製」的並行計算系統：它以一部個人電腦(例如IBM AT386)為控制單位，計算工作分配到由 16個 Transputer(英國Inmos公司製的微處理器)組成的PE網絡進行。每個Transputer附有 4MB 記憶體，以四個為一組裝在特殊設計的聯結板(圖八)上。這樣一個系統具有 64MB「分立」記憶體容量，整體運算速度理論上可以達到 160 MIPS 和 16 MFLOPS，比超級計算機例如CRAY X-MP(這其實也是一部並行機，但只有四個處理器)只差一個數量級，但它的費用(數萬美元)則便宜了至少兩、三個數量級。



圖八 EEC公司的step 4型處理器聯結板。左邊4個正方形積集電路是Transputer，它們左方各有4條垂直線，那是附屬的記憶體，每條容量1MB，共16MB。

另一方面，從90年代開始，商業並行計算系統也在市場上陸續出現。這其中最突出的無疑是美國麻省Thinking Machine公司的CM-2型並行機：它最多竟然可以用到 65,536個PE，整體記憶有 2GB，理論速度可以達到 40 GFLOPS，然而價錢(US\$10M)與傳統超級電腦卻也相去不遠。比較有代表性的系統，也許要推加州MasPar公司(現在已經被Digital Equipment公司收購)的MP-1 1200型並行機。它PE數目在 1,000 至 16,384 之間，理論速度達 1.5 GFLOPS，價錢(US\$1M)則還保持在中型電腦範圍內。但要認真比較市場上這許多產品(表一)的性能，我們還須談談並行系統的結構和特性。

## 五 並行計算系統：一個集體

並行計算系統的優越性(或該說策略)在於以數量換取質量：即是以多個同時工作的PE，產生按PE數目N倍增加的整體運算速度。它的基本困難(這和所有「集體」都一樣)則在於整體工作的分配和個體間的協調難以盡善，因此整體



表一 常見並行計算機性能的比較(其中CRAY X-MP是介乎傳統超級機與並行機之間的產物)

名稱	運算模式	節點大小	處理器數目	每節點的記憶(MB)	計算功能極限(GFLOPS)	價格US\$M
個人計算機 加 Transputer	MIMD	中	64	4	0.064	0.1
INTEL IPSC2	MIMD	中	128	8	10.2	3.0
N CUBE	MIMD	中	8,192	16	24.6	?
MasPar	SIMD	小	16,384	0.016	1.5	1.0
Thinking Machine CM-2	SIMD	小	65,536	0.032	40.0	10.0
CRAY-XMP	MIMD	大	4	(32) (共用)	0.8	10.0 (?)

「速率增加」(speedup)  $S$  往往遠低於  $N$  倍，也就是說效率  $\varepsilon = S / N$  不能達到 100%。例如，由於估計錯誤，在運作過程中 PE 陣列某些部分會「超負荷」(overload)，因此導致整體延誤；又或者某些類別計算(例如計算電子在複雜電磁場中的軌迹)有高度內在循序性，難以「分解」成許多同時進行的運算等等。並行系統有多種不同設計，為的就是要因應問題特性，以求維持高效率。

所以，PE 陣列有「粗粒」(coarse-grained) 和「細粒」(fine-grained) 之分。前者個別 PE 的功能較強，也就是說記憶體較大，速度較快，可以同時各自執行不同指令和處理不同數據，即以 MIMD (multiple-instruction stream and multiple-data stream) 模式運作。至於後者，則節點上的 PE 較簡單，附屬記憶體較小，它們多數(但也有例外)只能執行由控制單位「廣播」的同樣命令，即以 SIMD (single-instruction stream and multiple-data stream) 模式運作；此外還有「中粒」陣列，也是屬 MIMD 模式的。「粗粒」、「細粒」的選擇，一般要由問題本身特性決定：PE 數目較少的「粗」、「中」粒陣列宜用於牽涉長串複雜計算，而且不太容易高度分解的科學、工程問題；它另外一個優點則是由於節點各自長期獨立運作，因此統計上傾向於消除局部的超負荷。至於 PE 數目較多的「細粒」機，則宜用於牽涉大量「同時性」數據的圖象、訊號、和文字處理工作，因為它們大都可以自然地分解成許多互不相涉的小問題。在商業並行系統中，節點數目多數有很大彈性，可以由使用者選定。

整體工作的分配，也影響到單元之間的連結形式(connection topology)。整體計算可以看成將運算過程(algorithm)施於一個「數據空間」(data space)，並行機所做的就是把這空間「分割」(decompose)成  $N$  份，分別交由「節點」各自處理。問題是：經過一段工作後，節點就會需要交換數據，但結構上每一個節點只可以與有限幾個相鄰節點直接聯繫，作間接數據傳送的話，就要耗費大量時間，成為損害整體效率的另一個重要因素。因此，如何使數據流通的「樣式」

並行計算系統的優越性在於以數量換取質量；它的基本困難(這和所有「集體」都一樣)則在於整體工作的分配和個體間的協調難以盡善。

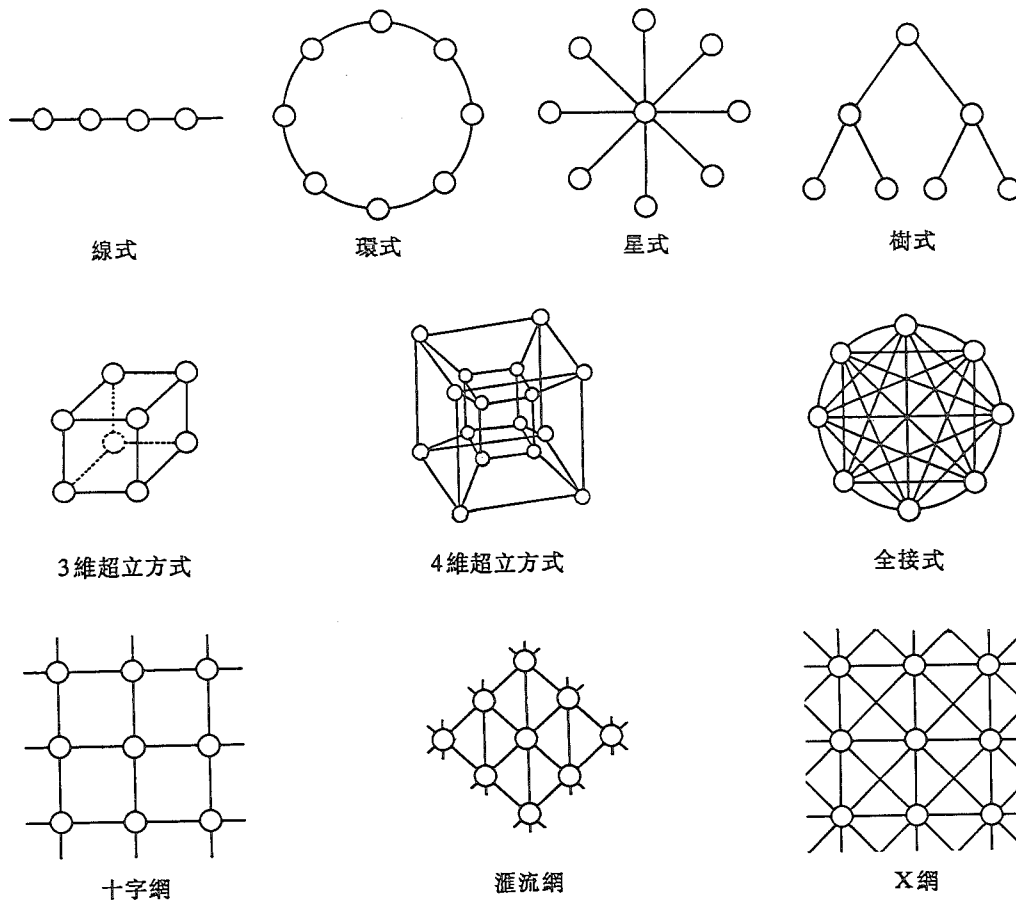
(pattern)與節點的「連接形式」配合，以減低間接數據傳送的需要，是極重要的。在目前，常見的「粗粒」或「中粒」機連接形式有線式、環式、樹式、星式、超立方(hypercube)式，全接式等等，至於「細粒機」則以各種網式為最普遍(圖九)。像MasPar那樣比較特別的「細粒機」，還包括了三種連接方式：CU與所有PE的連線；X-網連線；以及可以為1,024對任意節點同時建立直接連線的「整體通訊網」(global router)系統。因此，它有高度靈活性，可以應付多種數據流通樣式的需要。

「並行革命」的真正意義，可能是在另外一個和我們平常所謂「計算」(computing)迥然不同的領域。

目前並行機還只是在初生階段，但它的出現，和半導體或集成電路不同：它不單解決了一個老問題，而且開拓了一個新領域。現在並行計算已經從訊號、圖象和低層次文字處理這類較簡單而且「自然適合」的工作，迅速進入流體力學、氣象和海流研究、電磁波研究、高速動體設計等複雜領域，甚至已經被華爾街的投资公司用作短期股票預測和決定投資策略的工具了。誠然，它的網絡結構、程序「語言」(目前都是從傳統語言如FORTRAN, C, Pascal等借用的)、程式設計技巧等等，都還在摸索中，但可以預見，它將來在計算、預測、控制各方面的發展潛力，幾乎是無限巨大的。

但「並行革命」的真正意義，卻可能是在另外一個領域——一個和我們平常所謂「計算」(computing)迥然不同的領域。

圖九 微處理器陣列的各種聯結方式



## 六 從計算到思維？

計算機的運作和人腦的活動是有極大差異的。計算機在極簡陋的階段，已經能以千百倍於任何人的速度作完全準確的邏輯和算術運算；同樣，對一個不識字的三歲小孩來說極自然、容易的事，例如辨別聲音、認人、認路、運用語言等等，至今最先進的超級計算機還未完全模仿得來。這其中的關鍵到底何在？

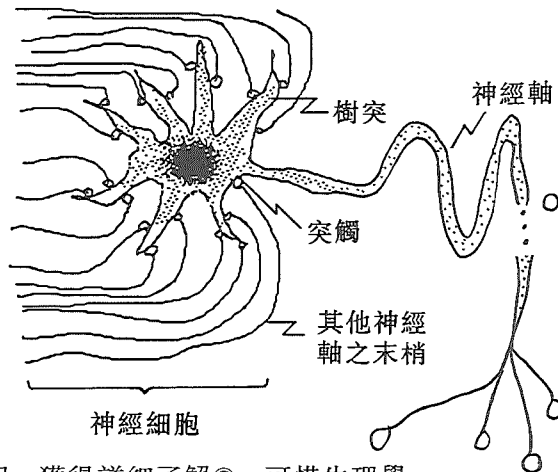
最簡單的答案自然是人腦比計算機來得「複雜」。這是不錯的：目前，一部先進計算機的主體記憶容量大約有  $10^8$  字節，這比人腦神經細胞(neuron)的數目( $10^{11}$ )差了大約兩、三個數量級。而且，隨着記憶體容量的增加，某些人腦功能(例如圖象識別)也的確逐漸變成可以被計算機模仿。

但人腦還有一個更根本，更重要的複雜性：那就是它絕非由單一的CPU指揮運作，而是從有極其複雜連接模式的神經網絡(neural network)的整體活動中產生思維的——在這過程中，大部分神經細胞都是活躍參與的。近數十年來，神經解剖學和生理學的進步，使我們對個別神經細胞通過神經軸(axon)末梢與其他5,000–15,000個同樣細胞上的突觸(synapse)聯結，以及它們以神

經軸中的電脈衝互相激發或抑制的個別過程，獲得詳細了解<sup>⑤</sup>：可惜生理學家、物理學家根據這些了解去建立數學模型和計算機模型，以研究神經網絡的整體活動，雖然也已經有多時，但具體成果則似乎還沒有多少。

然而，有一點不必懷疑，那就是神經網絡的作用，和每一步驟、每一細節都要事先按一定次序完全安排好，不容有絲毫含糊的「循序計算」運作方式，是絕不相同的。這恐怕也就是兩者功能迥異的根源：大腦不是一架完全服從CPU的邏輯推理機，而更接近於一個龐大的並行處理系統——一個有  $10^{11}$  PE 節點，但並沒有單獨「控制中心」的系統。在最近的將來(譬如說今後十年內)，要建造這樣一個龐大複雜的系統自然還不可能。然而，不可否認，「並行處理」和「大量並行化」(massive parallelism)的思想，正是從邏輯性、決定性的「計算」，跨向整體性(holistic)、聯想性(associative)甚至帶有若干隨機性和意外性的「思維」的橋樑。過去幾年間，並行機的出現和普及，說明這座橋樑不再是空中樓閣，而已經開始動工了。倘若這個看法不無道理，那麼真正的「並行革命」——也就是「電子計算機」蛻變為「電腦」的革命，可能也不會太遙遠了。

圖 並行處理單元的未來？這是一個腦神經細胞的示意圖。



大腦不是一架完全服從CPU的邏輯推理機，而更接近於一個龐大的並行處理系統。

### 參考資料

John L. Hennessy and David A. Patterson, *Computer Architecture, A Quantitative Approach* (San Mateo, CA: Morgan Kaufmann, 1990).

Kai Hwang and Fayé A. Briggs, *Computer Architecture and Parallel Processing* (New York: McGraw-Hill, 1985).

Igor Aleksander and Helen Morton, *An Introduction to Neural Computing* (London: Chapman & Hall, 1990).

T.R. Reid, *The Chip* (New York: Simon & Schuster, 1985).

### 註釋

① Aleksander & Morton, p.viii.

② 計算機每一「代」的起迄是有爭議的。這裏所採用的是以某一類型計算機最早出現以及開始流行的時期為準，接近 Hwang & Briggs 的用法。

③ ENIAC的發明者是賓州大學的Presper Eckert與John Mauchly。范奈曼的貢獻主要是在1944年提出建造根據存儲程式運作的計算機EDVAC的建議(但Eckert, Mauchly以及Goldstein等人都可能曾參與討論)，以及在1946年和Goldstein與Burks三人聯名發表一個全面、有決定性的報告，詳盡列出「循序計算機」的構想。有不少人認為范奈曼在計算機發展史上獨享大名，許多其他開拓者被忽略了，是不公平的，但以數字代表程式，和以存儲程式控制運作這關鍵構想源自范奈曼，則是肯定的。

④ 其實在80年代CPU速度的增長相當快，1985以後幾乎可以與記憶容量的增長率並駕齊驅。問題在於記憶的存取(access)速度增長每年平均只有7%，因此嚴重影響CPU的整體運作速度，這是計算機當前發展的重要問題。見Hennessy & Patterson, pp. 426-27。

⑤ Aleksander & Morton, Ch. 2有簡單介紹。

## 後記

本文由趙挺偉教授供稿，本刊編輯室協助編撰。本刊對下列三位中大同事之大力協助表示衷心感謝：黎建宇博士提供補充資料；陳天機教授校閱文稿，並提出寶貴意見；楊振寧教授提供有關范奈曼事蹟的資料。但文中如有錯誤，仍應由本刊編輯室負責。

**趙挺偉** 1951年出生，現任台灣大學物理系教授，從事高能理論及量子場論的研究。曾任加州理工學院物理系客座教授；在1985至1987年間完成建造4個節點的並行計算硬體；自1987年起使用多種不同大型並行電子計算機進行晶格場論研究；1987年曾提出隨機方塊晶格場論。