

EVOLUZIONE E PROSPETTIVE NELL'HIGH PERFORMANCE COMPUTING

Attività quali le previsioni meteorologiche o la simulazione di processi nucleari sono state per molto tempo effettuate con supercomputer costruiti con una tecnologia denominata *high performance computing* (HPC). Tuttavia, i grandi progressi ottenuti nella costruzione e nell'aggregazione in *cluster* dei microprocessori hanno consentito di costruire nuovi tipi di supercomputer. In questo articolo, si esamina come l'HPC si sia evoluto e quali siano le prospettive tecnologiche e applicative che si possono intravedere per il futuro.

1. PREMESSA

Si immagini di aver costruito un modello computerizzato del pianeta terra e di entrare come in un videogioco nella realtà virtuale di questo modello per esplorare paesaggi, città, strade, case e uffici. Il computer potrebbe consentire di controllare, in ogni istante, la densità della popolazione, la distribuzione dei prodotti agricoli, la disponibilità dei beni di consumo, le congestioni del traffico, e tutto ciò su qualunque scala.

In linea di principio, ingrandendo per gradi l'immagine, si potrebbe entrare in ogni ufficio o in ogni negozio per osservarne le più minute attività. Si esaminerebbero in dettaglio, e in tempo reale, ogni transazione effettuata in una specifica agenzia bancaria così come l'acquisto di ogni singola bottiglia in un negozio di vini. Si potrebbe così osservare l'attività dell'economia di una nazione, o di una città, identificandone eventuali inefficienze e possibili miglioramenti.

Rick Stevens, direttore del dipartimento di matematica e *computer science* del *National Argonne Laboratory* di Chicago, ha denominato

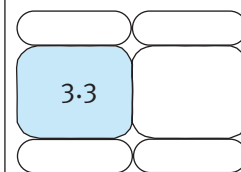
questo modello computerizzato il *Mirror World*, ossia il mondo riflesso in uno specchio. Dal punto di vista concettuale, un simile modello non ricade in quelle classi di problemi che per la loro natura sembrano, come si dice in linguaggio matematico, "intrattabili", ma richiede, evidentemente, un'estrema capillarità di dispositivi periferici e una prodigiosa capacità di calcolo.

Intuitivamente, si comprende che per attuarlo è necessario poter eseguire enormi quantità di istruzioni in breve tempo. Ma, volendo essere più concreti, quante? Questa domanda introduce, quasi naturalmente, al tema dei computer più potenti, ovvero dei *supercomputer*. Cosa sono i supercomputer e in cosa si distinguono dai grandi computer utilizzati, per esempio, per la gestione della prenotazione posti di una compagnia aerea o dei conti correnti di una grande banca? Per capire le diverse esigenze di un grande computer e di un supercomputer si cercherà di fare un raffronto, ancorché semplificato, tra due possibili scenari.

Si prenda, innanzitutto, il caso di una grande



Ernesto Hofmann



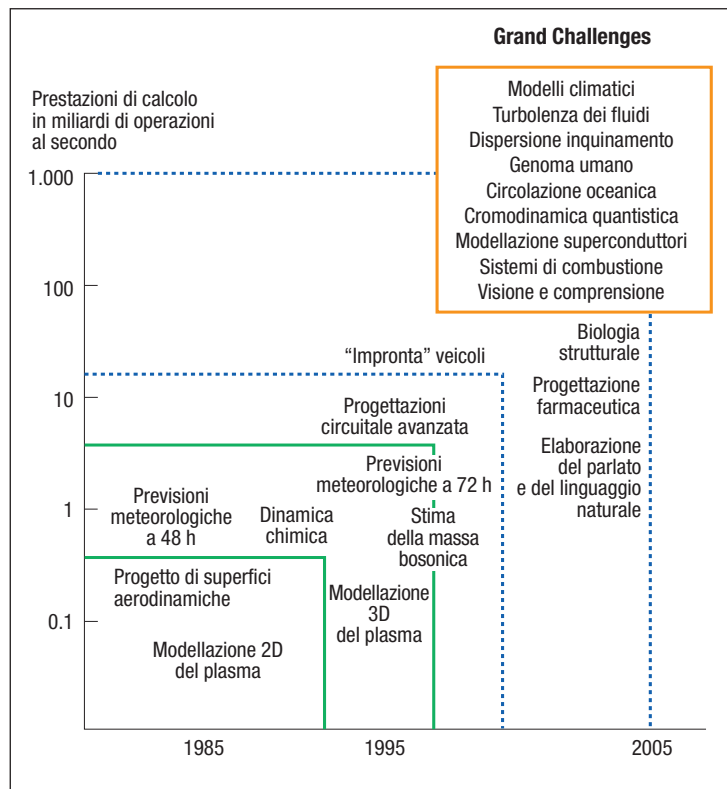
cento miliardi di fotogrammi sono, quindi, centomila miliardi di miliardi di istruzioni da dover eseguire in un decimillesimo di secondo. Si immagini, infine, di avere a disposizione un supercomputer che esegua un milione di miliardi di istruzioni al secondo: esso dovrebbe lavorare per cento milioni di secondi, ossia circa tre anni. E oggi i supercomputer più potenti sono almeno cento volte meno potenti di tale supercomputer.

Ma quello del ripiegamento delle proteine è un caso limite. Ci sono, infatti, problemi meno complessi, ancorché tuttora difficili da trattare, come le nuove prospezioni petrolifere, il comportamento delle galassie, i servizi finanziari avanzati, l'*on-line trading*, la simulazione dell'urto di due buchi neri, la progettazione di strutture complesse di edifici, automobili e aerei, la simulazione di esplosioni nucleari, l'evoluzione del clima, l'andamento delle correnti marine, la previsione dei terremoti, e altro ancora (Figura 1).

I supercomputer più potenti oggi disponibili, come il *NEC Earth Simulator* presso l'*Institute for Earth Science* di Yokohama, gli *ASCI/Q* della *HP-Compaq* installati presso il *Los Alamos National Laboratory*, l'*IBM ASCI White* al *Lawrence Livermore Laboratory*, o ancora l'*HP-Compaq Terascale Computing System* al centro di supercalcolo di Pittsburgh, dispongono oggi di capacità di calcolo che sono dell'ordine delle migliaia di miliardi di istruzioni (in virgola mobile) al secondo, ovvero teraflop/s. Quali sono allora le caratteristiche costruttive di questi computer e come si potranno evolvere per poter affrontare problemi ancora più complessi come quelli posti, appunto, dal ripiegamento delle proteine?

2. L'AVVENTO DEI SUPERCOMPUTER

Un'istruzione, per essere eseguita dal computer, utilizza, mediamente, alcune decine di circuiti microelettronici. Più rapidi sono questi ultimi più veloce è l'esecuzione dell'istruzione. Ma un'istruzione, per essere eseguita, ha spesso bisogno di utilizzare *operandi* che potrebbero non essere immediatamente disponibili ai circuiti utilizzati dall'istruzione stessa. È, quindi, necessaria una struttura di memoria che consenta ai dati, più frequente-



mente utilizzati, di essere immediatamente disponibili. Dalla metà degli anni '60 era già evidente che per superare i vari "colli di bottiglia" che si potevano presentare nell'esecuzione di un'istruzione, occorreva una complessa organizzazione circuitale.

Si era compreso che, per alcune classi di problemi, occorrevano computer che si discostassero dall'originale architettura di Von Neumann la cui caratteristica più saliente è la sequenzialità delle operazioni sia nelle unità aritmetico-logiche sia nella gestione della memoria.

A parità di tecnologia circuitale – la cosiddetta tecnologia di semiconduttore – si potevano ottenere più elevate prestazioni se venivano utilizzate, contemporaneamente, molteplici unità di elaborazione in grado di eseguire, in parallelo, gruppi differenti, e opportunamente selezionati, di istruzioni.

Negli stessi anni, si evolveva così un insieme di computer progettati per ottenere prestazioni molto più elevate rispetto ai cosiddetti *general purpose computer*. L'obiettivo di tali supercomputer era, in generale, quello di risolvere problemi caratterizzati dalla grande quantità di dati numerici sui quali occorreva eseguire ripetutamente sequenze di istruzio-

FIGURA 1

Prestazioni di calcolo per problemi "Grand Challenge"

ni abbastanza simili, piuttosto che la casualità delle sequenze di istruzioni tipiche degli ambienti commerciali.

Ciò che distingueva i supercomputer erano essenzialmente tre elementi: l'utilizzo delle più avanzate tecnologie circuitali, l'uso di molteplici unità di elaborazione e il disegno di opportuni algoritmi.

Alcuni sistemi, come l'ILLIAC IV (costruito per l'Università dell'Illinois dalla Burroughs) e il PEPE (*Parallel Ensemble of Processing Elements dell'U.S. Army Advanced Ballistic Missile Defense*), invece di risolvere i problemi attraverso una serie di operazioni successive eseguivano, simultaneamente, un elevato numero di operazioni.

Un tipico problema che l'ILLIAC IV poteva risolvere, molto più rapidamente rispetto a un computer convenzionale, era la soluzione dell'equazione alle derivate parziali di Laplace che descrive la distribuzione della temperatura sulla superficie di una piastra.

Nell'ILLIAC IV duecentocinquantesi *processing element* erano raggruppati in quattro quadranti di 8×8 . Ogni *processing element* aveva una sua memoria ed era in grado di comunicare con i quattro *processing element* adiacenti.

Altri computer, come lo STARAN (della *Goodyear Aerospace Corporation*), utilizzavano il principio delle memorie associative. Nello STARAN una memoria associativa, con un accesso in parallelo a 256 voci di 256 *bit*, era gestita attraverso un aggregato di unità di elaborazioni indipendenti (una per ciascuna voce di memoria). Lo STARAN veniva utilizzato per applicazioni che permettevano un elevato livello di parallelismo intrinseco quali analisi di tracce *radar* o calcolo matriciale. Intorno alla metà degli anni '60, furono soprattutto due costruttori, la *Control Data Corporation* (CDC) e la IBM, ad avviare una straordinaria sfida industriale per la costruzione di una nuova classe di computer che fossero almeno un ordine di grandezza più potenti rispetto ai più potenti computer per uso commerciale.

Fa ormai parte della leggenda l'aneddoto secondo cui Thomas Watson Jr., presidente della IBM, fosse molto contrariato nell'apprendere poco dopo l'annuncio, nell'aprile del 1964, del sistema IBM 360 - che la CDC era pronta a costruire anch'essa un computer, il CDC 6600, più veloce del più potente computer della se-

rie 360, ossia quel 360/75 che avrebbe comunque costituito uno degli elementi chiave del successo del progetto Apollo.

L'IBM avviò, immediatamente, la progettazione e costruzione del 360/91 che, come il CDC 6600, avrebbe introdotto nelle architetture dei computer il principio della *pipeline*.

La sfida sarebbe poi proseguita, negli anni successivi, con l'introduzione di due computer ancora più potenti: il CDC 7600 (1969) e il sistema IBM 360/370 195 (1970).

Con la tecnica della pipeline, che è possibile immaginare come un'estensione del principio dell'*overlap* delle istruzioni, ci si proponeva di far diventare il computer una vera e propria catena di montaggio di istruzioni che passavano via via attraverso unità funzionali specializzate.

Il termine pipeline nasceva dall'analogia con la condotta di petrolio nella quale non interessa la velocità della singola molecola bensì la quantità di petrolio che sgorga al secondo dalla condotta stessa.

Analogamente, nei computer di tipo pipeline non interessava tanto il tempo necessario a eseguire una singola istruzione quanto piuttosto la quantità di istruzioni eseguite al secondo. In quegli stessi anni, inoltre, si cominciavano a comprendere sempre meglio i meccanismi, non solo fisici, ma anche logici sui cui si fonda un'effettiva esecuzione, in parallelo, di molteplici attività. Si immagini di avere 52 carte di uno stesso mazzo distribuite in modo casuale e di volerle mettere in ordine. Se ci fossero quattro giocatori a volersi dividere le attività secondo i colori (cuori, quadri, fiori, picche) si farebbe certamente prima che se si dovesse svolgere la stessa attività da soli. Ma fossero 52 i giocatori, tutti intorno allo stesso tavolo, a voler ordinare il mazzo si farebbe probabilmente solo una gran confusione.

Questo banale esempio può far comprendere come esista un limite al di là del quale non conviene spingere sul parallelismo degli agenti ma, piuttosto, sulla velocità del singolo.

Gene Amdahl, uno dei maggiori architetti della recente storia dei computer, enunciò all'inizio degli anni '70 un'ipotesi che venne poi denominata legge di Amdahl: un perfetto parallelismo nelle attività dei computer non è possibile perché esisteranno sempre delle sequenze ineliminabili di *software* intrinsecamente seriale [1]. Basti pensare, per esempio, a una serie di nu-



meri di Fibonacci: 1,1,2,3,5,8,13,21,... generata dal semplice algoritmo $F(n+2) = F(n+1) + F(n)$. Non è possibile eseguirla in parallelo perché ogni successiva operazione dipende dalle precedenti due¹.

Esistono però anche attività che mostrano un intrinseco parallelismo. Tornando all'esempio iniziale del ripiegamento della proteina si può, per un momento, riflettere sul fatto che la conformazione finale della proteina nello spazio tridimensionale sarà quella con la minore energia potenziale. Si potrebbero, allora, calcolare indipendentemente, e in parallelo, migliaia di conformazioni indipendenti della proteina stessa. Calcolate le diverse conformazioni sarà facile trovare quella con la minima energia, che è proprio quanto è stato fatto negli ultimi anni, per molte proteine, con l'ausilio di alcuni tra i migliori supercomputer disponibili. Ma è anche intuibile che il livello di parallelismo del computer pone un limite al numero di possibili conformazioni che è possibile calcolare in parallelo che, nel caso della proteina, possono essere, anche, infinite. Occorre, infine, ricordare che in quegli anni si attuava anche un enorme progresso nella comprensione dei meccanismi di sincronizzazione di sempre più complessi aggregati microcircuitali che comprendevano ormai memorie di transito ad alta velocità (*cache*), per istruzioni e per dati (*I-cache*, *D-cache*), gerarchie di memorie (L1, L2, L3,...), registri interni di vario tipo e, soprattutto, una crescente molteplicità di unità di caricamento (*pre-fetch* e *fetch*) di istruzioni, di decodifica (*instruction unit*) e di esecuzione (ALU, *Arithmetic Logical Unit*).

La cosiddetta prima legge di Moore cominciava a far sentire i suoi effetti. Ogni diciotto mesi raddoppiava la quantità di *transistor* e, quindi, di circuiti disponibili per *chip*. Nell'IBM, per esempio, dalla tecnologia SMS (*Standard Modular System*) dell'inizio degli anni '60 si passava alla SLT (*Solid Logic Technology*), alla ASLT

(*Advanced SLT*), alla MST (*Monolithic System Technology*) via via fino alla LSI (*Large Scale Integration*) del 1978 e, infine, alla VLSI (*Very Large Scale Integration*) della metà degli anni '80. La quantità di circuiti disponibile su di un chip passava, nel volgere di un paio di decenni, da un *transistor* a migliaia di circuiti, per la gioia dei progettisti che potevano così aggregare sempre più funzioni in uno stesso spazio senza doversi preoccupare dei ritardi di segnale.

Se gli attuali microprocessori costruiti con la tecnologia CMOS (*Complementary Metal-Oxide-Silicon*) possono oggi essere così complessi lo si deve anche al fatto che tra gli anni '60 e '70 i grandi computer costruiti con tecnologia bipolare (prevalentemente ECL, *Emitter Coupled Logic*) hanno esplorato innumerevoli possibilità di aggregare i microcircuiti per ottenere sempre più elevati livelli di parallelismo nell'esecuzione delle istruzioni.

3. I COMPUTER VETTORIALI

Negli anni '60, due tra i maggiori protagonisti nella progettazione dei grandi computer erano stati indiscutibilmente Gene Amdahl e Seymour Cray. Ambedue, poi, avrebbero lasciato le aziende per le quali lavoravano (rispettivamente, l'IBM e la CDC) per avviare proprie aziende. Amdahl fondava la *Amdahl Corporation* che avrebbe costruito alcuni dei più evoluti mainframe a cavallo tra gli anni '70 e '80. Seymour Cray avviava, invece, la *Cray Research Inc.* Nel 1975, veniva installato il primo Amdahl 470/V6 e un anno dopo presso il Los Alamos Scientific Laboratory veniva installato il primo computer vettoriale Cray 1.

Il 1976 può a ragione essere considerata la data di inizio del *supercomputing*. Prima del Cray 1, infatti, computer come il CDC 7600, che pure venivano definiti supercomputer, non differivano, realmente, dai computer più convenzionali. Ciò che li caratterizzava, a parte le più elevate prestazioni, come è stato già detto, era, essenzialmente, l'elevato livello di parallelismo interno. Occorre anche aggiungere che la CDC aveva, in realtà, realizzato un computer vettoriale, il CDC Star 100, già nel 1972; ma questa macchina non aveva avuto successo, soprattutto perché le sue prestazioni, in taluni casi, erano persino inferiori a quelle dei più potenti sistemi scalari.

¹ È anche vero che esiste da oltre un secolo una formula, quella del matematico Binet, che consente di individuare direttamente l'ennesimo numero di Fibonacci. La formula è però di difficile semplificazione per numeri di Fibonacci molto grandi per i quali si può ricorrere invece al calcolo di potenze della radice quadrata di cinque.

Con il Cray 1 nasceva un'architettura innovativa che utilizzava istruzioni vettoriali e registri vettoriali per risolvere alcune classi di problemi essenzialmente matematici.

Un processore vettoriale è un processore che può operare con una singola istruzione su di un intero vettore. Si consideri, per esempio, la seguente istruzione di somma:

$$C = A + B$$

Sia nei processori scalari sia in quelli vettoriali l'istruzione significa "aggiungere il contenuto di A al contenuto di B e mettere il risultato in C". Mentre in un processore scalare gli operandi sono numeri in un processore vettoriale l'istruzione dice al processore di sommare, a due a due, tutti gli elementi di due vettori. Un particolare registro di macchina, denominato *vector length register*, dice al processore quante somme individuali devono essere eseguite per sommare i due vettori.

Un compilatore vettoriale è, a sua volta, un compilatore che cercherà di riconoscere, per esempio, quali *loop* sequenziali possono essere trasformati in una singola operazione vettoriale. Il seguente loop può essere trasformato in una sola istruzione:

```
DO 10 I = 1, N
C(I) = A(I) + B(I)
10 CONTINUE
```

La sequenza di istruzioni del loop viene tradotta in un'istruzione che imposta a N la lunghezza del *length register* seguita da un'istruzione che esegue l'operazione di somma vettoriale. L'uso delle istruzioni vettoriali consente migliori prestazioni essenzialmente per due motivi. Innanzitutto, il processore deve decodificare un minor numero di istruzioni e, quindi, il carico di lavoro della *I-unit* è fortemente ridotto e il *bandwidth* di memoria viene, di conseguenza, ridotto anch'esso. In secondo luogo, l'istruzione fornisce alle unità esecutive (le ALU vettoriali) un flusso costante di dati. Quando inizia l'esecuzione dell'istruzione vettoriale il processore sa che deve caricare coppie di operandi che sono disposte in memoria in modo regolare. Così il processore può chiamare il sottosistema di memoria per alimentare in maniera costante le ALU con le coppie successive.

Con una memoria dotata di un buon livello di parallelismo di accesso (il cosiddetto *interleaving*) le coppie possono essere alimentate al ritmo di una per ciclo ed essere dirette consecutivamente, come in una catena di montaggio (la pipeline, appunto) per essere via via sommate. Occorre, tuttavia, anche un cambio di mentalità. Programmatori abituati a ragionare in modo sequenziale dovevano acquisire una nuova mentalità e, anche se parte della conversione da software applicativo sequenziale a parallelo poteva essere eseguita in maniera automatica, restava pur sempre da svolgere un'ulteriore attività di "parallelizzazione" manuale che poteva richiedere la ristrutturazione di algoritmi che erano stati pensati in modo essenzialmente seriale.

4. I MASSIVE PARALLEL SYSTEM (MPP)

Nella seconda metà degli anni '80, iniziò a farsi strada una nuova classe di sistemi: i computer paralleli a memoria distribuita. Questi sistemi differivano dagli SMP (*Shared-Memory Multiprocessor*), nati all'inizio degli anni '70, poiché la memoria non veniva condivisa da un unico sistema operativo, ma ogni processore gestiva la propria memoria e il proprio software applicativo.

L'architettura degli MPP non consentiva una sincronizzazione così stretta delle attività applicative come negli SMP e richiedeva, come si può intuire, che le diverse applicazioni fossero abbastanza indipendenti l'una dall'altra. È anche evidente che nell'esecuzione di una singola applicazione, solo se la sequenza delle istruzioni di quest'ultima può essere efficacemente suddivisa (parallelizzata) in sezioni eseguibili indipendentemente sui vari processori, l'MPP può dare effettivi vantaggi.

Dal punto di vista della programmazione applicativa, i sistemi SMP erano più semplici da utilizzare. Con un'organizzazione di memoria distribuita il programmatore deve, infatti, tenere conto di chi detiene che cosa. Gli SMP, però, funzionano bene solo se i processori non sono tanti, altrimenti nascono troppi conflitti per specifiche posizioni di memoria. I primi modelli di simili *massive parallel system* apparvero sul mercato nel 1985 in maniera abbastanza eterogenea. La *Thinking*



Machine Corporation (TMC) presentava la *Connection Machine-1* (CM-1), mentre la Intel annunciava l'iPSC/1 che utilizzava microprocessori Intel 80826 interconnessi da una rete *Ethernet* in un cosiddetto ipercubo.

In virtù della relativa facilità di aggregazione di simili sistemi, vennero rapidamente a costituirsi molte nuove aziende costruttrici di sistemi MPP.

Il sistema MPP più interessante di quegli anni sarebbe rimasto il CM-2 disegnato da Danny Hillis per la TMC. Costituito da 64 microprocessori interconnessi in un ipercubo, insieme a 2048 *Witek floating point unit*, il sistema poteva operare su di una singola applicazione

sotto il controllo di un solo sistema di *front-end*. Il CM-2 per alcune categorie applicative riuscì persino a competere con i migliori sistemi vettoriali MP dell'epoca, come il Cray Y-MP. Fu proprio il crescente successo di questi nuovi sistemi a determinare la necessità non solo di una rigorosa tassonomia dei vari supercomputer ma anche una classificazione relativa secondo un'opportuna metrica, quella del LINPACK, che prevede la soluzione di un cosiddetto sistema denso di equazioni lineari. Le prestazioni vengono oggi misurate in teraflop/s, ossia in migliaia di miliardi di operazioni in virgola mobile eseguite al secondo.

La lista TOP500 (Tabella 1), dei 500 più poten-

TABELLA 1

La TOP500 List

del Novembre 2002

Rank	Manufacturer Computer/Procs	R _{max} R _{peak}	Installation Site Country/Year
1	NEC Earth-Simulator/5120	35860.00 40960.00	Earth Simulator Center Japan/2002
2	Hewlett-Packard ASCI Q - Alpha Server SC ES45/1.25 GHz/4096	7727.00 10240.00	Los Alamos National Laboratory USA/2002
3	Hewlett-Packard ASCI Q - Alpha Server SC ES 45/1.25 GHz/4096	7727.00 10240.00	Los Alamos National Laboratory USA/2002
4	IBM ASCI White, SP Power3 375 MHz/8192	7226.00 12288.00	Lawrence Livermore National Laboratory USA/2000
5	Linux NetworX MCR Linux Cluster Xeon 2.4 GHz - Quadrics/2304	5694.00 11060.00	Lawrence Livermore National Laboratory USA/2002
6	Hewlett-Packard Alpha Server SC ES 45/1 GHz/3016	4463.00 6032.00	Pittsburgh Supercomputing Center USA/2001
7	Hewlett-Packard Alpha Server SC ES 45/1 GHz/2560	3980.00 5120.00	Commissariat a L'Energie Atomique (CEA) France/2001
8	HPTI Aspen Systems, Dual Xeon 2.2	3337.00 6158.00	Forecast Systems Laboratory - NOAA USA/2002
9	IBM pSeries 690 Turbo 1.3 GHz/1280	3241.00 6656.00	HPCx UK/2002
10	IBM pSeries 690 Turbo 1.3 GHz/1216	3164.00	NCAR (National Center for Atmospheric Research) - USA/2002
11	IBM pSeries 690 Turbo 1.3 GHz/1184	3160.00 6156.00	Naval Oceanographic Office (NAVOCEANO) USA/2002
12	IBM SP Power3 375 MHz 16 way/6656	3052.00 9984.00	NERSC/LBNL USA/2002
13	IBM pSeries 690 Turbo 1.3 GHz/960	2560.00 4990.00	ECMWF UK/2002
14	IBM pSeries 690 Turbo 1.3 GHz/960	2560.00 4990.00	ECMWF UK/2002

ti supercomputer installati nel mondo, viene pubblicata semestralmente dalle Università di Mannheim e del Tennessee e dall'U.S. Department of Energy's National Energy Research Scientific Computing Center.

La maggior parte dei sistemi MPP era, sostanzialmente, costituita di *cluster* più o meno integrati di microprocessori preesistenti. Ciò faceva sì che la distinzione tra sistemi convenzionali e supercomputer divenisse molto più sfumata. Un sistema convenzionale poteva, opportunamente "clusterizzato", diventare di fatto un supercomputer.

Inoltre, a livello di tecnologia di semiconduttore la crescente densità dei chip consentiva alla tecnologia CMOS di rimpiazzare, soprattutto in termini di rapporto prezzo/prestazioni e di consumi e dissipazione di calore, la tecnologia ECL.

Questo fatto, poichè la capacità industriale di produrre e integrare in un unico disegno sempre più evoluti microprocessori CMOS si restringeva via via a poche aziende, riduceva il numero di costruttori di supercomputer. Alle soglie del 2000, delle 14 maggiori aziende costruttrici di supercomputer ne rimanevano solo quattro costituite dai tre maggiori costruttori giapponesi (Fujitsu, Hitachi e NEC) e dall'IBM, cui venivano ad aggiungersi la Silicon Graphics, la Sun, la Hewlett-Packard e la Compaq. Queste ultime due si sarebbero poi consociate.

5. IL PROGRAMMA ASCI

Attualmente, ci sono negli USA diverse iniziative che cercano di consolidare attraverso l'*High Performance Computing* le capacità computazionali di università, laboratori di ricerca e organizzazioni militari. Il più interessante di questi programmi è, probabilmente, l'*Accelerated Strategic Computing Initiative* (ASCI) [2] il cui compito è quello di creare le capacità computazionali e di simulazione d'eccellenza necessarie per mantenere la sicurezza l'affidabilità e le prestazioni dello *stockpile* nucleare USA e ridurre il rischio nucleare. Nell'ambito di questo programma sono stati realizzati diversi sistemi.

Nella TOPLIST del Novembre 2002 al secondo, terzo e quarto posto si trovano proprio tre sistemi ASCI. Nell'ordine, al Los Alamos Natio-

nal Laboratory sono operativi dal 2002 due ASCI Q – Alpha Server SC della HP-Compaq costituiti da 4096 processori ES45 da 1.25GHz con una capacità di calcolo di 7,727 Tflop/s. Al Lawrence Livermore Laboratory è operativo, dal 2000, l'IBM ASCI White costituito di 8192 processori SP POWER 3 da 375 MHz per una potenza complessiva di 7,226 Tflop/s.

L'ASCI Red, prodotto dalla Intel e installato presso il Sandia National Laboratory, è costituito da 9472 microprocessori Intel Pentium Xeon ed è stato, con 2.1 Tflop/s, il primo sistema a superare la barriera di un teraflop/s e a porsi, per un certo periodo, al vertice della TOP500.

L'ASCI Blue Mountain installato al Los Alamos Laboratory è stato costruito dalla SGI ed è costituito da 6144 processori e reagisce 1.6 Tflop/s.

Ci sono poi altri sistemi molto potenti. Il governo giapponese ha sovvenzionato lo sviluppo di un sistema per simulare e prevedere l'ambiente globale. Il NEC Earth Simulator è costituito di 640 nodi per un totale di 5120 unità di elaborazione, occupa uno spazio di quattro campi da tennis e utilizza 2800 km di cavo per le interconnessioni. Raggiunge una capacità di calcolo di 40 Tflop/s e appare come il supercomputer più potente secondo la classifica TOP500 di Novembre 2002.

Nell'ambito del programma ASCI, il 19 Novembre 2002 è stato annunciato che l'IBM avrebbe avviato la costruzione di due nuovi supercomputer, ASCI Purple e Blue Gene/L, che insieme dovrebbero fornire 500 Tflop/s, ossia una volta e mezzo la potenza di tutti i supercomputer della più recente TOP500. ASCI Purple, con una capacità di calcolo di 100 Tflop/s, sarà costituito da 12608 microprocessori IBM POWER5 e consumerà 4 MW (l'equivalente dell'energia elettrica necessaria per circa 5000 abitazioni). I microprocessori saranno contenuti in 196 computer, con un bandwidth di memoria di 156000 Gbyte, e interconnessi da un *data-highway* con un bandwidth di 12500 Gbyte. ASCI Purple conterrà anche 50 Tbyte di memoria e 50 petabyte di memoria su disco, che è all'incirca l'equivalente di un miliardo di libri di media lunghezza.

ASCI Purple, che verrà installato entro il 2004 presso il Lawrence Livermore National Laboratory, dov'è già installato ASCI White, servirà come il principale supercomputer del-



l'U.S. Department of Energy nell'ambito del *National Security Administration* (NNSA) dove sarà utilizzato nel *Stockpile Stewardship Program* per simulare l'invecchiamento e l'operatività delle testate nucleari americane senza dover ricorrere a test sotterranei.

6. LA RESURREZIONE DEL VETTORIALE

Pur producendo sistemi MPP di successo, come i TRD/T3E la Cray ha continuato a sviluppare supercomputer vettoriali anche dopo l'accordo di cooperazione con la SGI. In particolare, la Cray nel 1998 ha costruito il supercomputer SV1, poi perfezionato nel 2000.

La Cray ha poi ritenuto possibile utilizzare il meglio delle due architetture, quella vettoriale e quella MPP, per costruire un nuovo supercomputer, denominato durante la fase di progetto SV2.

Nel novembre 2002, finalmente la Cray ha annunciato il Cray X1, già disponibile, che, in sostanza, riunisce il meglio dei sistemi vettoriali SV1 e T90 e dei sistemi MPP T3D/E.

Nel sistema SV1, la Cray aveva introdotto i cosiddetti processori *multistreaming* (MSP) per sfruttare il vantaggio di avere, invece di un'unica profonda pipeline vettoriale, molteplici pipeline più brevi. Da un punto di vista puramente intuitivo, è possibile immaginare che molteplici piccole pipeline possano essere utilizzate per gestire in parallelo, più efficientemente, loop abbastanza brevi, ma frequenti nell'ambito di un programma.

Nel Cray X1 il blocco costruttivo fondamentale è un processore di tipo MSP costituito di quattro unità scalari ciascuna dotata di un paio di *vector pipe*. Ogni unità scalare può eseguire, non in sequenza su una delle due unità esecutive, due istruzioni per ciclo. Le quattro unità scalari operano a 400 MHz e forniscono una capacità di calcolo totale di 3.2 Gflop/s.

Come accennato in precedenza, i compilatori per i sistemi vettoriali sostanzialmente "aprono" i loop in sequenze di istruzioni ottimizzate per l'*hardware* vettoriale. Per loop più grandi la sequenza di istruzioni di un unico loop può essere distribuita su tutte le 8 pipeline di un processore.

Le 8 pipeline vettoriali operando a 800 MHz forniscono una capacità di calcolo complessi-

siva di 12.8 Gflop/s per operazioni a 64 bit. Poiché nel disegno complessivo del Cray X1 sono previsti fino a 4000 MSP, il Cray X1 stesso può fornire una capacità di calcolo di 52.4 Tflop/s.

7. LE PROSPETTIVE DEI SUPERCOMPUTER: VERSO IL PETASCALE COMPUTING

Negli Stati Uniti i supercomputer sono stati sempre considerati un elemento strategico fondamentale per mantenere la *leadership* non solo tecnologica.

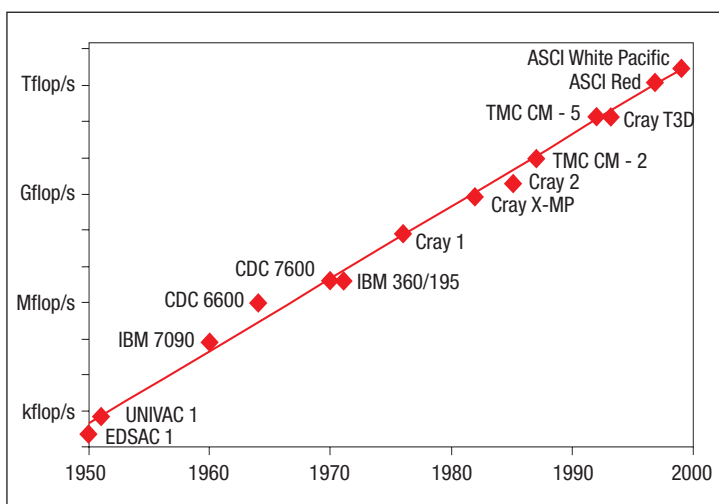
Il rapporto PITAC del 1999 (*Report from the President's Information Technology Advisory Committee*) raccomanda un programma federale di ricerca che consenta di " ... ottenere prestazioni da petaflop/s su applicazioni reali entro il 2010 attraverso un opportuno bilanciamento di strategie hardware e software" [5]. Il rapporto identifica molte applicazioni che potranno trarre considerevoli vantaggi dal petascale computing:

- gestione degli ordigni nucleari;
- crittologia;
- modellazione ambientale e climatica;
- ricostruzione tridimensionale delle proteine;
- previsione di uragani;
- disegno avanzato di aerei;
- nanotecnologie molecolari;
- gestione in tempo reale di immagini per la medicina.

Se si traccia su di una scala logaritmica una curva che rappresenti l'evoluzione dei supercomputer si può vedere come la prima legge di Moore sia stata in gran parte rispettata (Figura 2).

FIGURA 2

L'evoluzione dei supercomputer



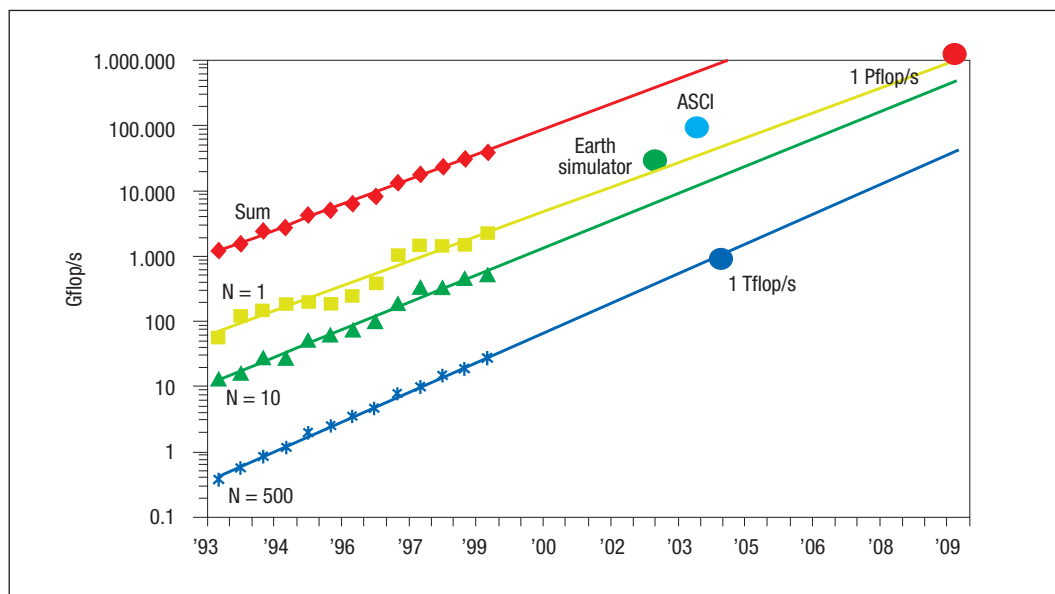


FIGURA 3
Verso il petascale computing

Si potrebbe fare un esercizio anche più interessante esaminando l'evoluzione delle prestazioni nella TOP500, prendendo di volta in volta l'evoluzione del sistema più potente, del meno potente (ossia, il sistema classificato al posto 500) e, quindi, del sistema classificato al decimo posto (Figura 3).

La curva dei supercomputer alla posizione 500 mostra un miglioramento annuale di un fattore medio di 1.8. Le altre curve mostrano un miglioramento intorno a un valore annuale pari a 2. Estrapolando si può vedere che i primi sistemi da 100 Tflop/s dovrebbero essere disponibili intorno al 2005. Nel 2005, nella TOP500 non ci dovrebbe essere più alcun supercomputer con potenza di calcolo minore di un teraflop/s.

Estrapolando ulteriormente ci si può spingere fino a un petaflop/s e vedere che questa potenza dovrebbe essere raggiunta intorno al 2009. In effetti, come si vedrà più avanti, ci sono due sistemi, l'IBM Blue Gene e l'HTMT che promettono di fornire una simile potenza intorno a quella data.

La CRAY sostiene che il sistema Cray X1 è un primo passo verso il petascale computing che potrebbe essere realizzato, attraverso un'opportuna evoluzione tecnologica e di disegno, entro il 2010.

Il petascale computing sembrerebbe, quindi, raggiungibile in tempi abbastanza brevi, ma occorrono ancora notevoli progressi tecnologici e anche progetti alternativi

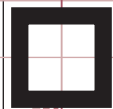
perché non è detto che lungo la strada qualche progetto non si dimostri, di fatto, irrealizzabile.

Si ritiene, oggi, che vi siano almeno quattro possibili alternative tecnologiche: l'evoluzione della tecnologia convenzionale, il disegno cosiddetto *processor in memory* (PIM), le tecnologie superconduttive e, infine, il *Grid Computing*.

7.1. Le tecnologie convenzionali

Il futuro del supercalcolo sarà condizionato sia dall'evoluzione della tecnologia circuitale sia da quella del disegno di sistema che, a sua volta, non potrà non essere influenzato dalla tecnologia disponibile.

Allo stato attuale, i microprocessori più densi hanno quasi 200 milioni di transistor: nel volgere di pochi anni, saranno disponibili chip con oltre un miliardo di transistor. La *Semiconductor Industry Association* (SIA) nel suo ultimo documento [6] ha delineato la possibile evoluzione della tecnologia di semiconduttore fino al 2014. Le sue proiezioni indicano, in estrema sintesi, che nel 2008 le memorie dinamiche (DRAM) avranno 5 miliardi di bit per cmq, mentre le memorie statiche (SRAM) avranno 500 milioni di transistor per cmq e, infine, i chip di circuitazione logica ad alte prestazioni avranno 350 milioni di transistor per cmq. Queste densità verranno quasi triplicate entro il 2011, mentre le frequenze per lo



stesso periodo arriveranno quasi a una decina di GigaHertz contro il singolo GigaHertz attuale.

Queste proiezioni sono impressionanti perché è presumibile che le dimensioni dei chip saranno tra 400 e 800 mm², e perciò, entro pochi anni, saranno disponibili chip con miliardi di transistor. Ciò avrà profonde conseguenze sul disegno interno dei computer e quindi sulle loro prestazioni e sulle loro possibilità applicative

Si immagina, infatti, di poter disporre di un chip di 400 mm² con una densità di transistor quale quella prevista da SIA.

Un processore, di disegno sufficientemente evoluto, occuperebbe 5 mm² e, quindi, 16 processori da 6 GHz impegnerebbero 80 mm², lasciando perciò 320 mm² disponibili per le intercomunicazioni tra i processori e per la memoria. Utilizzando 50 mm² per le comunicazioni restano 270 mm² per la memoria che, alle densità previste da SIA, fornirebbe 64 Mbyt di SRAM oppure 11 Gbyt di DRAM. Nel 2011 i 16 processori lavorerebbero a 10 GHz e la memoria DRAM diventerebbe di 35 Gbyt.

Raccogliendo su di una superficie (*board*), di 20 × 20 cm, 64 chip si può arrivare a interconnettere oltre 1000 processori per ottenere una capacità complessiva di 20 Tflop/s, che è quella oggi fornita dai massimi supercomputer disponibili.

Simili strutture, che sembrano quasi ricordare quella di un alveare, prendono il nome di sistemi cellulari e già oggi, con la tecnologia attualmente disponibile, sono in fase di progettazione. Il supercomputer IBM Blue Gene [3] il cui progetto è stato annunciato nel dicembre del 1999, ha già queste caratteristiche.

Come accennato in precedenza, nel novembre del 2002, l'IBM ha annunciato l'avviamento di un altro progetto da completarsi in tempi più brevi rispetto all'originario Blue/Gene: Blue Gene/L.

Blue Gene/L è costituito da 64 *cabinet*, ciascuno dei quali contiene 128 *board*, ognuna delle quali alloggia 8 chip, su ciascuno dei quali prendono posto 2 processori. Blue Gene/L è costituito quindi da circa 130.000 processori di una potenza di circa 2.8 Gflop/s, operanti in Linux, per una potenza

complessiva che può arrivare a 360 Tflop/s. L'obiettivo di Blue Gene/L, oltre alla simulazione del folding delle proteine, è quello di consentire la modellazione tridimensionale delle stelle per verificare perché le stelle binarie, con un elevato rapporto di masse, diventino instabili e si fondono. Sarà anche possibile utilizzare tecniche avanzate di modellazione della meccanica quantistica per studiare con grande accuratezza le proprietà della materia. Un'area di particolare interesse sarà la possibilità di studiare come riparare sequenze di DNA danneggiate dalla radiazione. Infine, la possibilità di simulare con grande precisione la propagazione di onde sismiche e acustiche potrebbe essere utilizzata per la previsione di terremoti, per le prospezioni petrolifere, per creare sempre più accurate immagini a uso medico.

7.2. Il Processing in memory (PIM)

Durante l'evoluzione dell'High Performance Computing, precedentemente descritta, è emerso un fattore tecnologico che, poco alla volta, è diventato uno dei maggiori vincoli alla crescita dell'effettiva capacità di calcolo dei supercomputer.

Questo vincolo è rappresentato dal progressivo sbilanciamento tra la tecnologia dei processori e quella della memoria dei supercomputer: la velocità di esecuzione delle istruzioni è aumentata molto più rapidamente del tempo di accesso alla memoria centrale.

Il *gap* tra il miglioramento nell'esecuzione delle istruzioni e nell'accesso alla memoria sta aumentando di circa il 50% annuo (Figura 4).

Già negli anni '60-'70, non era facile mantenere caricata correttamente la pipeline in ogni momento ed erano state progettate molteplici tecniche, come il pre-fetch delle istruzioni, e come i *buffer* di transito ad alta velocità (le cosiddette, cache) che erano successivamente stati distinti in I-cache e D-cache (per le istruzioni e i dati). Si era anche esaminato il problema della cosiddetta "località dei dati", ossia la distanza media che c'era in memoria tra le istruzioni e i dati su cui queste dovevano operare. Tale località varia a seconda della natura delle

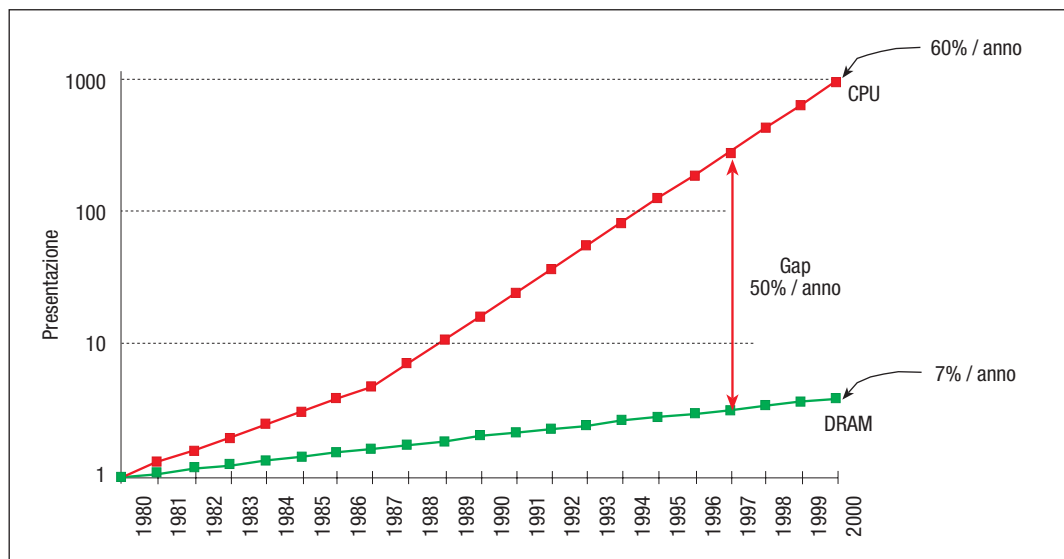


FIGURA 4
Il latency gap delle DRAM

applicazioni e può richiedere approcci sostanzialmente diversi.

Tutte queste tecniche avevano portato alla costruzione di complesse gerarchie di memoria, soprattutto nei computer più potenti e, in particolare, nei supercomputer.

Si è visto nella sezione 6.1, che la densità fotolitografia aumenterà nei prossimi anni in misura tale da consentire di alloggiare nello stesso chip memorie SRAM ovvero DRAM e molteplici microprocessori.

Numerose tecniche sono, oggi, allo studio per utilizzare al meglio le possibilità che la tecnologia sarà, quindi, in grado di fornire. Già da alcuni anni, si stanno evolvendo nuove tecniche di memoria come la PIM [8] o la IRAM (*Intelligent RAM*) che prevedono di integrare sullo stesso chip logica e memoria.

7.3. Le tecnologie superconduttive

I petaflop/s computer sembrano, comunque, destinati a consumare enormi quantità di corrente. Si è già visto che un sistema da un decimo di petaflop/s, come ASCI Purple, richiederà almeno 4 MW. Un sistema da un petaflop/s costruito con tecnologia convenzionale potrebbe arrivare a dover dissipare 100 MW di potenza e richiederebbe un impianto di raffreddamento grande come un edificio di tre piani e al buio apparirebbe dotato di una certa luminosità.

L'idea dei sistemi cellulari, come Blue Gene, è allora quella di utilizzare un numero molto

grande, fino a decine di migliaia di chip con un disegno sufficientemente semplice da non richiedere un'elevata potenza elettrica individuale.

Sembra, però, possibile anche un'altra strada che è quella di utilizzare una tecnologia superconduttiva denominata *Rapid Single-Flux-Quantum Logic* (RSFQ), inventata da Konstantin Likharev della *State University* di New York [7].

L'elemento chiave di questa tecnologia sono microscopici anelli che possono quasi istantaneamente commutare dallo stato magnetizzato a quello non-magnetizzato (0 e 1) senza, praticamente, consumare potenza. Gli anelli possono rimpiazzare gli attuali transistor e consentire di costruire microprocessori fino a mille volte più rapidi di quelli attuali.

La tecnologia RSFQ viene oggi utilizzata nell'ambito di un ambizioso progetto denominato *Hybrid Technology Multi-Threaded* (HTMT) supportato dal *Defense Advanced Research Projects Agency* (DARPA), dalla *National Security Agency* (NSA), e dalla *National Aeronautics and Space Administration* (NASA). Del progetto fanno parte prestigiosi ricercatori il cui obiettivo è quello di realizzare il petascale computing entro il 2010.

7.4. Il Grid Computing

La potenza di calcolo complessiva del Web è certamente enorme ma è facilmente intuibi-



le, anche per quanto è stato detto finora, che a causa della *latency* di memoria e dell'eterogeneità del bandwidth tale potenza non possa facilmente essere utilizzata per eseguire, in modo unitario, applicazioni che richiedano grande capacità di calcolo.

Esistono, comunque, classi di applicazioni, le cosiddette *loosely coupled application*, che, pur richiedendo elevata capacità di calcolo, non hanno vincoli di esecuzione particolarmente stringenti e che, inoltre, si prestano a essere suddivise per essere eseguite su molteplici sistemi.

In questo caso, la definizione stessa di applicazione tende a diventare abbastanza indefinita perché, una volta che differenti sequenze di istruzioni vengano eseguite quasi indipendentemente per risolvere un problema, non è chiaro se esse possano complessivamente venire a costituire un'unica applicazione in senso classico.

Un programma di suddivisione del software applicativo sui sistemi del Web è, comunque, in atto da parte di organizzazioni e costruttori diversi nell'ambito di una strategia che prende complessivamente il nome di Grid Computing.

Secondo questo modello, non importa dove specifiche sequenze di istruzioni vengano eseguite, quanto piuttosto che la loro sincronizzazione complessiva, e quindi il raggiungimento del risultato, vengano raggiunti.

Il termine *grid* deriva dall'analogia con infrastrutture già esistenti, come quella elettrica o quella dell'acqua. Nell'aprire un rubinetto o nell'accendere una lampada non ci si domanda da dove provenga l'acqua o l'elettricità: l'importante è che queste siano rese disponibili. Analogamente, nel Grid Computing, quando una sequenza di istruzioni deve essere eseguita, non importa quale processore se ne farà carico: il sistema complessivo dovrà individuarlo nell'ambito di quelli disponibili in rete e ivi verranno eseguite le istruzioni.

Con una simile strategia è intuibile che il modello di sviluppo applicativo venga ulteriormente modificato. Già nel passaggio dall'informatica centralizzata degli anni '60-'70 a quella sempre più distribuita (soprattutto *client-server*) degli anni '80-'90, si era

assistito all'evolversi delle tecniche di modularizzazione del software applicativo.

Dalle semplici chiamate *call-return* nell'ambito di un solo sistema si era passati ad *application programming interface* che dovevano coordinare logicamente (e fisicamente) l'esecuzione di sequenze di istruzioni che potevano risiedere su sistemi diversi e anche ubicati remotamente. Poiché i vari sistemi potevano avere anche un'architettura diversa, oltre ai diversi sistemi operativi attivi sui rispettivi computer, diventava necessaria l'aggiunta di una specie di "colla" software che consentisse di collegare tali sistemi in una visione logica complessiva, ma in maniera il più indolore possibile, soprattutto, per i programmatori applicativi. Tale colla prese il nome di *middleware*, ossia il software che si poneva in mezzo.

In una strategia Grid Computing il *middleware*, che qualcuno già scherzosamente chiama *griddleware*, diventerà elemento di fondamentale importanza e il ruolo di organizzazioni come CORBA, Legion, Globus [10, 11] è proprio quella di ottimizzare e standardizzare quanto più possibile tale *middleware*.

8. ALGORITMI E MODELLI APPLICATIVI

Quanto detto fin qui, ha soprattutto evidenziato l'evoluzione della tecnologia di base, del disegno delle macchine e in parte dei sistemi operativi. Resta, però, da considerare l'importantissimo ruolo che svolgerà il software applicativo per conseguire realmente il petascale computing. È necessario, innanzitutto, considerare l'importanza degli algoritmi. Quest'ultimo è un aspetto spesso sottovalutato. Infatti, per utilizzare al meglio un computer che esegua contemporaneamente su molteplici complessi circuiti svariati gruppi di istruzioni occorre risolvere due problemi. È necessario, innanzitutto, sviluppare in software un algoritmo che si presti a essere suddiviso in più parti, ossia in diverse sequenze di istruzioni, da eseguire in parallelo. In secondo luogo, occorre disporre di un linguaggio, quello che in gergo informatico si chiama un *compilatore*, che sappia ottimizzare nelle giuste sequenze da

0

1

0

1

0

1

0

64

distribuire in parallelo le istruzioni scritte dal programmatore.

La progettazione di un algoritmo efficiente è, spesso, più efficace di un hardware sofisticato. Basti pensare a un algoritmo tra i più banali: quello della moltiplicazione. Si potrebbe, ad esempio, sommare 57 volte il numero 23 su se stesso, ma questo procedimento, anche per un uomo, sarebbe molto più lungo del notissimo algoritmo che consente di moltiplicare i due numeri tra loro con pochi passaggi aritmetici. È intuibile che si possano sviluppare algoritmi molto raffinati ed è questo un fondamentale terreno di ricerca nel disegno dei supercomputer.

Purtroppo, non è facile trovare la soluzione giusta. Con la disponibilità del calcolo parallelo, il ricercatore che debba sviluppare una nuova applicazione potrebbe essere inghiottito a cercare fin dall'inizio algoritmi intrinsecamente parallelizzabili. Più difficile è, evidentemente, il compito di chi debba riadattare software applicativo già esistente alle nuove possibilità tecnologiche.

Il paragone con il gioco degli scacchi può essere utile, non solo per comprendere l'accanimento con il quale l'industria del computer ha cercato, negli ultimi anni, di sconfiggere di volta in volta il campione del mondo in carica, ma anche per capire meglio il tema della complessità algoritmica.

L'*input* per un programma che gioca a scacchi è la specifica disposizione dei pezzi, ossia la configurazione di una scacchiera, che in un'analogia biologica può corrispondere alla sequenza di aminoacidi di una proteina. L'algoritmo scacchistico cerca, in una libreria elettronica di partite giocate dai grandi maestri, se vi sia una configurazione quasi identica. Se questa viene trovata, la mossa successiva sarà uguale a quella effettuata nella partita omologa individuata nella libreria.

In modo analogo, il modello tridimensionale della nuova proteina può essere costruito usando come stampo la struttura della proteina di riferimento.

Fino al 30% delle proteine relative a genomi, pienamente sequenziali, mostra una rassomiglianza alla sequenza genomica di proteine la cui struttura è già nota, cosicché anche

per le prime può esserne costruito un buon modello tridimensionale.

Ma se non c'è una sufficiente rassomiglianza ossia se, nell'analogia, la configurazione della scacchiera non è immediatamente riconoscibile tra quelle delle partite memorizzate elettronicamente, che fare?

Nel gioco degli scacchi occorre un algoritmo di riconoscimento delle possibili rassomiglianze tra le configurazioni dei pezzi molto più raffinato che nella tecnica cosiddetta dell'omologia. In quest'ultima, come già detto, la configurazione della partita in gioco e quella della partita memorizzata sono pressoché identiche. Ma nella tecnica del riconoscimento di forma delle proteine (*fold recognition*) le rassomiglianze, riconosciute come tali, devono essere catturate in maniera più sottile.

A questo punto, può veramente entrare in gioco la potenza di calcolo del computer che, come si è visto, sembra poter ancora crescere in maniera considerevole. Le prospettive tecnologiche sono, quindi, molto incoraggianti e indicano che la tecnologia di semiconduttore, il disegno dei sistemi cellulari e la progettazione di sofisticati algoritmi saranno sinergicamente in grado di fornire nel prossimo decennio una crescita nella capacità di elaborazione che va persino al di là della cosiddetta prima legge di Moore. Si potranno, allora, affrontare con maggior ottimismo le sfide dei prossimi anni e si schiuderà, probabilmente, un'epoca nella quale sarà possibile migliorare in maniera sorprendente la qualità della vita. Forse si potranno predire terremoti e maremoti, si potrà sapere in anticipo come saranno le stagioni, comprendere l'andamento dei mercati finanziari, progettare nuovi materiali, coltivare meglio i terreni, ma, soprattutto, venire a capo di alcuni flagelli, come le malattie degenerative, che rappresentano, tuttora, un incubo per la maggior parte dell'umanità.

Bibliografia

- [1] Amdahl's law: www.sc.ameslab.gov/publications/AmdahlsLaw/Amdahls.pd
- [2] Advanced Simulation and Computing at Lawrence Livermore National Laboratory: www.llnl.gov/asci



- [3] Blue Gene: A vision for protein science using a petaflop supercomputer: <http://www.research.ibm.com/journal/sj/402/allen.html>
- [4] Hybrid Technology Multi-threaded (HTMT) Architecture Project: http://ess.jpl.nasa.gov/subpages/reports/ooreport/lc_oo.htm
- [5] Information Technology Research: Investing in Our Future, President's Information Technology Advisory Committee Report to the President: www.ccic.gov/ac/report
- [6] International Technology Roadmap for Semiconductors, SIA, Austin, Texas, 1999: http://public.itrs.net/files/1999_SIA_Roadmap/Home.htm
- [7] Likharev K: *Rapid Single Flux Quantum (RSFQ) Logic*. In Encyclopedia of Material Science and Technology RSFQ: <http://gamayun.physics.sunysb.edu/RSFQ/publications.html>
- [8] Mitros P: Computing in Memory: <http://www.ai.mit.edu/projects/aries/course/notes/pim.html>
- [9] Overview of HTMT: <http://www.cacr.caltech.edu/powr98/presentations/htmt/sldoo1.htm>
- [10] Foster I, Kesselman C: The Globus Project: A Status Report: <http://citeseer.nj.nec.com/foster96globu.html>
- [11] Grimshaw Andrew S, Wulf Wm A: The Legion vision of a worldwide virtual computer: <http://portal.acm.org/citation.cfm?id=242867&coll=portal&dl=ACM&ret=1>
- [12] van der Steen AJ, Dongarra JJ: Overview of recent supercomputers. <http://www.phys.uu.nl/~steen/webo2/overviewo2.html>
- [13] TOP500 SUPERCOMPUTER SITES: www.TOP500.org

ERNESTO HOFMANN, laureato in fisica presso l'Università di Roma. È entrato in IBM nel 1968 nel Servizio di Calcolo Scientifico. Nel 1973 è diventato manager del Servizio di Supporto Tecnico del Centro di Calcolo dell'IBM di Roma. Dal 1984 è Senior Consultant. È autore di molteplici pubblicazioni sia di carattere tecnico sia divulgative, nonché di svariati articoli e interviste. Dal 2001 collabora con l'Università Bocconi nell'ambito di un progetto comune Bocconi-IBM.
Ernesto_Hofmann@it.ibm.com