

Lezioni di Calcolo Parallelo

Prof. Almerico Murli ¹

Università degli Studi di Napoli
“Federico II”

¹Già direttore del Centro di Ricerca per il Calcolo Parallelo e i Supercalcolatori del Consiglio Nazionale delle Ricerche (CPS-CNR), ora direttore della Sezione di Napoli dell'Istituto di Calcolo e Reti ad Alte Prestazioni del CNR (ICAR).

Premessa

In questo volume sono contenute alcune delle lezioni di Calcolo Parallelo tenute a partire dal '90, agli studenti del Corso di Laurea in Matematica, e più recentemente, agli studenti del Corso di Laurea in Informatica dell'Università degli Studi di Napoli "Federico II" e dell'Università degli studi "Parthenope" nonché agli studenti del corso di Laurea in Matematica della seconda Università di Napoli. Si suppone che lo studente abbia già risolto qualche problema con un calcolatore monoprocesso (dalla progettazione e sviluppo dell'algoritmo alla sua descrizione con un linguaggio di programmazione fino alla sua implementazione in uno specifico ambiente di calcolo che può essere anche un usuale PC).

L'obiettivo principale è quello di introdurre all'uso delle metodologie e degli strumenti di calcolo che consentono di raggiungere alte prestazioni. Il calcolo parallelo è infatti una delle metodologie principali del Calcolo Scientifico e più in generale della "Computational Science" per ottenere prestazioni elevate e nella parte più strettamente numerica può essere considerato come un capitolo dell'Analisi Numerica.

È opportuno fissare la data nella quale è stata ultimata la redazione di queste lezioni, data la rapida obsolescenza della tecnologia: **"la tecnologia avanzata di oggi è la tecnologia superata di domani."**

Napoli, Luglio 2004

Indice

Introduzione	1
1 Calcolo parallelo: l'idea di base	17
1.1 L'operazione di somma	34
2 Le architetture parallele	49
2.1 Classificazione delle architetture parallele	49
2.2 Reti di interconnessione per sistemi a memoria distri- buita	60
3 Il paradigma dello scambio di messaggi (message pas- sing)	67
3.1 L'operazione matrice per vettore	68
3.2 Somma di $p = 2^m$ vettori	101
3.3 Lo standard MPI	110
3.4 Il prodotto di due matrici	117
4 I parametri di valutazione del software parallelo	147
5 Alcune librerie per il calcolo matriciale numerico	181
5.1 Rivisitazione dell'algoritmo di eliminazione di Gauss: da BLAS1 a BLAS3	184
5.2 Da LINPACK a ScaLAPACK	209
5.2.1 BLACS	210

5.2.2	ScaLAPACK	213
5.3	La fattorizzazione LU su architetture MIMD a memoria distribuita	215
5.3.1	La versione Block-partitioned di LAPACK	215
5.3.2	Distribuzione ciclica a blocchi nella Fattorizzazione LU	226
5.3.3	Strategia di parallelizzazione della Fattorizzazione LU	231
5.3.4	Analisi dell'efficienza	236
5.4	Cenni ad ATLAS	243
6	Cenni sull'evoluzione dei calcolatori presenti sul mercato	247
6.1	Le prime macchine vettoriali e la Cray Research	250
6.2	La Thinking Machine Corporation	256
6.3	Sistemi INTEL	259
6.4	La TOP500	261
6.5	I supercalcolatori dell'ultimo decennio presenti nella TOP500	263
6.5.1	IBM SP	264
6.5.2	CRAY T3D	265
6.5.3	L'HP Exemplar X-Class	267
6.5.4	Il programma ASCI	268
6.5.5	L'Earth Simulator	273
6.5.6	NUMA-Q	273
6.5.7	SGI-ORIGIN2000	275
6.5.8	BLU MOUNTAIN	276
6.5.9	BEOWULF	277

*"... As soon as Analytical Engine exists, it will
necessarily guide the future course of science.
Whenever any result is sought by its aid,
the question will then arise -
by what course of calculation can these results
be arrived at by machine in the shortest time? ..."*
Babbage, 1864

*"...Non appena la Macchina Analitica esisterà,
essa necessariamente guiderà il futuro della scienza.
Tutte le volte che un risultato sarà ottenuto con il suo aiuto,
una domanda sorgerà spontanea -
Con questa macchina attraverso quale tipo di calcoli
si può ottenere lo stesso risultato
nel minimo tempo possibile ? ..."*

Introduzione

Uno dei primi riferimenti in letteratura all'utilizzo di uno strumento di calcolo in grado di *“ottenere più risultati allo stesso tempo”* è probabilmente quello comparso nella pubblicazione della Biblioteca Universale di Ginevra dell'ottobre 1842 intitolata *“Sketch of the Analytical Engine Invented by Charles Babbage”* di J.F. Menabrea. Nell'elencare le capacità della macchina analitica, egli scrive [31]:

“[...] In secondo luogo, risparmiare tempo: per convincerci di questo basti pensare che la moltiplicazione di due numeri, costituiti entrambi da venti cifre, richiede almeno tre minuti di calcolo. Parimenti, quando deve essere eseguita una lunga serie di calcoli identici, come quelli richiesti per realizzare una tabella numerica, la macchina può essere impiegata in modo tale da fornire più risultati allo stesso tempo, cosa che ridurrebbe significativamente il numero di processi. [...] ”

Non è chiaro se il *“parallelismo”* fosse effettivamente implementato dalla macchina analitica, ma, in ogni caso, fu evidente già a quei tempi l'impatto dello strumento di calcolo sulla metodologia di risoluzione di un problema scientifico.

Dai tempi della *“analytical engine”* di Babbage ad oggi è sempre stata forte l'influenza tra l'avanzamento della tecnologia e la

necessità e/o possibilità di risolvere problemi nuovi e sempre più complessi [45].

Attualmente le infrastrutture per il calcolo e la comunicazione sono uno strumento indispensabile per l'avanzamento della conoscenza in tutte le scienze, dalla fisica alla biologia, all'economia. La disponibilità di calcolatori e reti veloci, tuttavia, non è necessariamente garanzia di risoluzione di problemi complessi senza, a volte, una modifica sostanziale delle metodologie, degli algoritmi e del relativo software¹.

In questa ottica, da tempo, si parla di una terza metodologia per l'indagine scientifica che si basa sull'approccio computazionale, affiancando gli approcci classici, cioè quello teorico e quello sperimentale (*Computational Science and Engineering* (CSE) Fig. 1).

La simulazione computazionale consente la risoluzione di problemi prima ritenuti inaccessibili o comunque di risoluzione difficile. Essa, infatti, consente un'analisi qualitativa e quantitativa di fenomeni troppo complessi da studiare analiticamente (gallerie del vento, evoluzione del clima, evoluzione delle galassie,...) o di problemi troppo pericolosi per essere replicati in laboratorio (sicurezza delle armi nucleari, efficacia dei farmaci,...).

¹*"The fundamental law of computer science: As machines become more powerful, the efficiency of algorithms grows more important, not less."* L. Trefethen - "MAXIMS ABOUT NUMERICAL MATHEMATICS, SCIENCE, COMPUTERS, AND LIFE ON EARTH", SIAM News, Jan/Feb 1998.

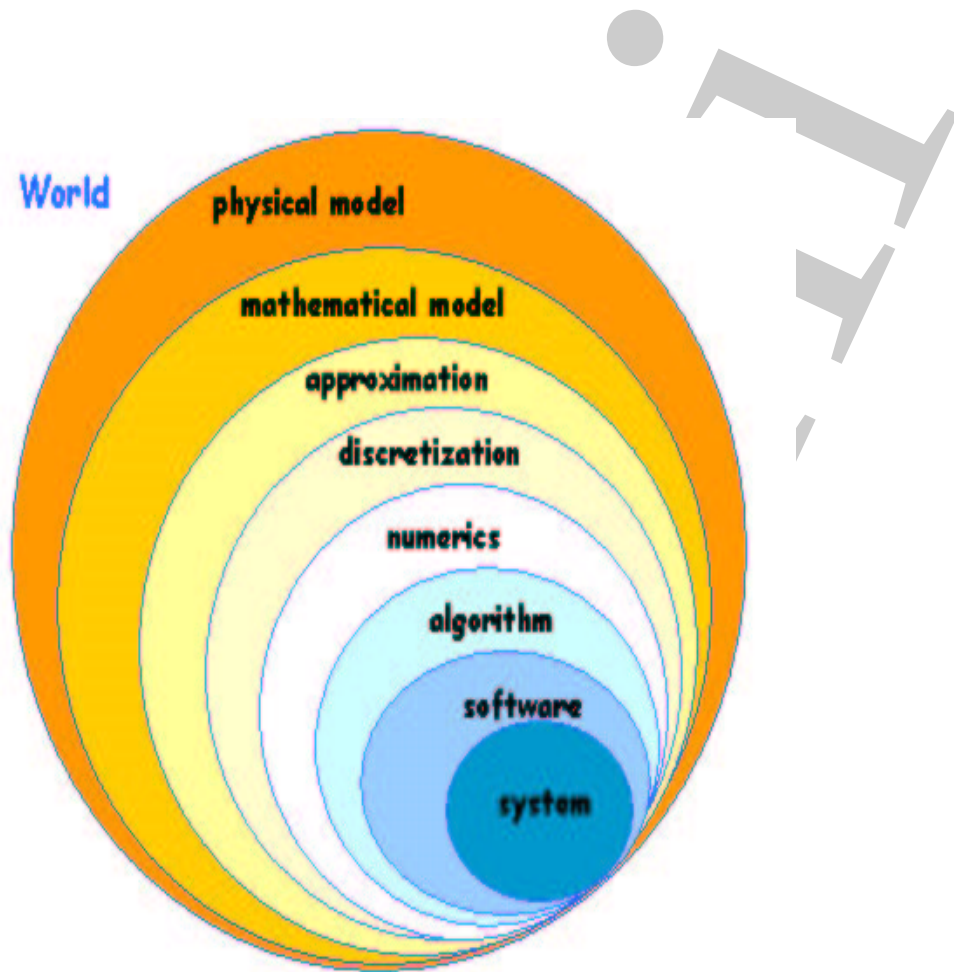


Figura 1: Il modello computazionale del Problem Solving della CSE può essere rappresentato come un insieme di cerchi concentrici. Il mondo reale è situato all'esterno dei cerchi. I livelli successivi rappresentano le diverse fasi del processo di risoluzione computazionale del problema: la costruzione del modello fisico, la formulazione del modello matematico, l'approssimazione e discretizzazione di quest'ultimo, lo sviluppo di algoritmi, l'implementazione di questi ultimi in uno specifico ambiente di calcolo (software) e la realizzazione del sistema che costituisce lo strumento finale a disposizione dell'utente.

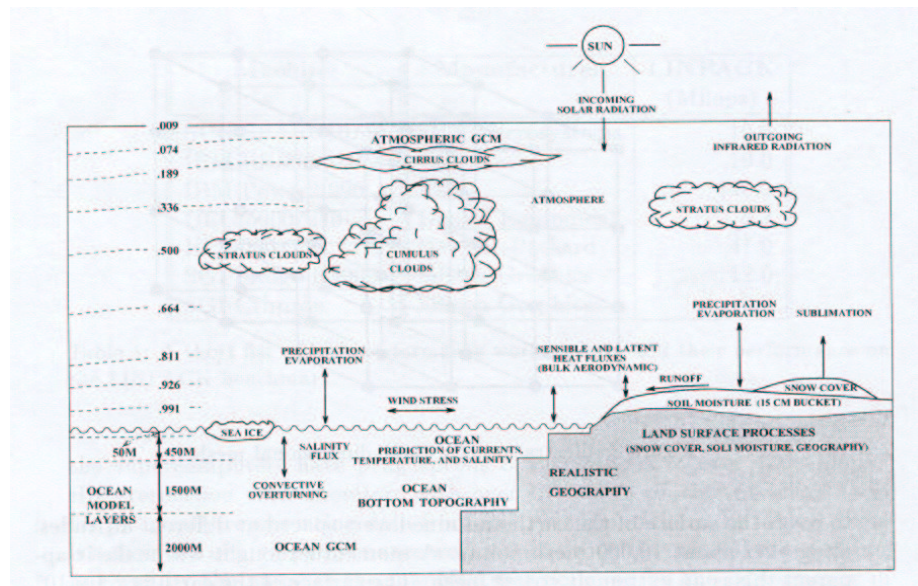


Figura 2: Schema del processo fisico alla base del modello GCM

♣ Esempio 1 [21]

Il surriscaldamento del globo è oggetto di un'attenzione sempre maggiore da parte della comunità scientifica. Lo studio di questo problema serve a comprendere come la variazione di concentrazione di diossido di carbonio nell'atmosfera contribuisca al surriscaldamento del pianeta per effetto serra. Uno studio di questo tipo richiede la modellizzazione del clima in un certo periodo di tempo. Sono stati effettuati studi al *National Center for Atmospheric Research*. Un modello climatico noto come General Circulation Model (GCM) viene usato per studiare il surriscaldamento che si genererebbe raddoppiando la concentrazione del diossido di carbonio in un periodo di 20 anni. Gli effetti che il modello GCM tenta di descrivere sono mostrati in Fig. 2.

L'atmosfera è un fluido, quindi il modello è descritto da equazioni differenziali alle derivate parziali. Le soluzioni numeriche di queste equazioni sono ottenute mediante algoritmi in cui le derivate rispetto alle coordinate spaziali e al tempo vengono approssimate da formule alle differenze nello spazio e nel tempo. Viene quindi generato un reticolo (griglia) a tre dimensioni nello spazio.

Per determinare la soluzione del problema vengono fissate alcune condizioni iniziali che assegnano alle variabili dei valori nei punti della griglia.

Se si usa un reticolo tridimensionale con circa 2000 punti, che coprono la superficie terrestre e nove piani posti a diverse altitudini, in totale vengono utilizzati circa 18000 punti per approssimare la soluzione. Questa discretizzazione risulta però abbastanza

A. Murli, *Lezioni di Calcolo Parallelo*

Versione provvisoria solo per uso personale, soggetta ad errori.

Non è autorizzata la diffusione. Tutti i diritti riservati.

larga. La superficie terrestre è infatti di circa $5.103 \times 10^5 \text{ km}^2$, quindi ogni punto del reticolo rappresenta 28.350 km^2 della superficie. Ad esempio, tenuto conto che l'Italia ha una superficie di circa 301.401 km^2 , solo 10 puni del reticolo rappresentano il nostro Paese. È ovvio quindi che occorre una accuratezza maggiore: se raddoppiamo il numero dei punti in ognuna delle tre direzioni aumentiamo il numero totale di un fattore $2^3 = 8$ anche se abbiamo solo 80 punti che rappresentano la superficie dell'Italia.

△

In tale contesto, il risultato del processo di risoluzione di un problema è un software funzionante in uno specifico ambiente di calcolo. L'intero processo, noto come ***Computational Problem Solving Methodology*** consiste nella costruzione del modello fisico, del modello matematico, nell'approssimazione e discretizzazione di tale modello, nell'applicazione di un metodo numerico, nel progetto di un algoritmo ed infine, nell'implementazione di un software per uno specifico sistema di calcolo.

Ed è proprio in questo contesto che si inserisce il ***Calcolo Parallelo***, caratterizzato da quell'insieme di principi e metodologie, che intervengono in tutte le aree del calcolo scientifico in un ambiente ad alte prestazioni e che rendono possibile la risoluzione computazionale di un problema.

Nella parte più strettamente numerica, il Calcolo Parallelo può essere considerato come un nuovo Capitolo dell'Analisi Numerica ed ormai oggi è la metodologia fondamentale per il calcolo ad alte prestazioni.

♣ Esempio 2 [5]

Si vuole stimare il tempo di risposta necessario per effettuare una ricerca sulla rete Internet tramite "motore di ricerca".

È evidente che il tempo di risposta dipende soprattutto dalla velocità del collegamento che si ha a disposizione e dal motore di ricerca. Per quanto riguarda il collegamento, se questo viene effettuato ad esempio da casa, il tempo necessario ad inviare la richiesta è uguale ad 1 secondo (modem a 56 Kbs). Se il collegamento viene effettuato

A. Murli, *Lezioni di Calcolo Parallelo*

Versione provvisoria solo per uso personale, soggetta ad errori.

Non è autorizzata la diffusione. Tutti i diritti riservati.

da un centro universitario come ad esempio l'Università degli Studi di Napoli Federico II, al momento in cui si scrive il tempo per inviare la richiesta è circa $\frac{1}{10^6}$ di secondo $(1\text{G-GE})^2$. Una volta inviata la richiesta nasce il problema di fornire una risposta nel più breve tempo possibile.

La ricerca di informazioni tramite un motore di ricerca è realizzata mediante parole chiave: l'utente fornisce al motore di ricerca le parole più significative e questo restituisce un elenco di tutte le pagine web contenenti tali parole. La ricerca di documenti da parte dell'utente è basata sul significato concettuale di ciò di cui ha bisogno. Il problema è che singole parole non evidenziano il contenuto concettuale di un documento. Ci sono infatti molti modi per esprimere un concetto (*synonymy*), quindi i termini letterali forniti in una richiesta (*query*) possono non essere quelli presenti nel documento. Inoltre, molte parole possono avere significati diversi in contesti diversi o se usati da persone diverse (*polysemy*).

È stato sviluppato un metodo per l'indicizzazione e la ricerca di documenti. Esso è noto come LSI (**L**atent **S**emantic **I**ndexing) [2, 5] e si basa sull'utilizzo della Decomposizione in Valori Singolari (**S**ingular **V**alue **D**ecomposition)³ [27]. Gli elementi della matrice sono l'occorrenza di ogni parola in un documento

$$A = [a_{ij}]$$

con a_{ij} frequenza con cui il termine i occorre nel documento j . Poiché ogni parola normalmente non appare in tutti i documenti, la matrice A generalmente è sparsa. Ogni coefficiente a_{ij} della matrice A è ottenuto moltiplicando il peso $L(i, j)$ locale del termine i , nel documento j , e il peso $G(i)$ globale dei termini i . Ovvero ogni elemento

²Il simbolo G indica 1 **G**igabit al secondo, mentre GE indica **G**igabit **E**thernet. A regime si avrà una banda di 2,5 gigabit.

Nel sito www.garr.net è possibile accedere a grafici che mostrano, in tempo reale, la velocità di collegamento alla rete dall'Università e dai principali nodi italiani.

³Diamo un breve richiamo della SVD. Data una matrice A di dimensione $m \times n$ con $m \geq n$ e $\rho(A) = r$ (ρ denota il rango della matrice), la SVD di A , denotata da $\text{SVD}(A)$, è definita da:

$$A = USV^T$$

dove $U^T U = V^T V = I_n$ e $S = \text{diag}(\sigma_1, \dots, \sigma_n)$, $\sigma_i > 0$ per $1 \leq i \leq r$, $\sigma_j = 0$ per $j \geq r + 1$. Le prime r colonne delle matrici ortogonali U e V definiscono gli autovettori ortonormali associati agli r autovalori non nulli di AA^T e $A^T A$, rispettivamente. Le colonne di U e V rappresentano i vettori singolari sinistri e destri rispettivamente, e i valori singolari di A sono definiti dagli elementi diagonali di S , che sono le radici quadrate degli autovalori di AA^T . Si dimostra [27] che se della matrice S si considerano solo i k valori singolari più grandi e le rispettive colonne delle matrici U e V , la matrice risultante A_k è la matrice di rango k che meglio approssima nel senso dei minimi quadrati la matrice A .

L'idea è che questa matrice, contenendo solo le prime k componenti di A linearmente indipendenti, evidenzia le associazioni nella matrice ed elimina i documenti con cui non ci sono associazioni.

Label	Titles
B1	A Course on <u>Integral Equations</u>
B2	Attractors to Semigroups and Evolution <u>Equations</u>
B3	Automatic Differentiation of <u>Algorithms</u> : <u>Theory</u> <u>Implementation</u> and <u>Application</u>
B4	Geometrical Aspects of <u>Partial Differential Equations</u>
B5	Ideals, varieties, and <u>Algorithms</u> - An <u>Introduction</u> to Computational Algebraic Geometry and Commutative Algebra
B6	<u>Introduction</u> to Hamiltonian Dynamical <u>Systems</u> and the N-body <u>Problem</u>
B7	Knapsack <u>Problems</u> : <u>Algorithms</u> and Computer <u>Implementations</u>
B8	<u>Methods</u> of Solving Singular <u>System</u> of <u>Ordinary Differential Equations</u>
B9	<u>Nonlinear System</u>
B10	<u>Ordinary Differential Equations</u>
B11	<u>Oscillation Theory</u> for Neutral <u>Differential Equations</u> with <u>Delay</u>
B12	<u>Oscillation Theory</u> of <u>Delay Differential Equations</u>
B13	Pseudodifferential Operators and <u>Nonlinear Partial Differential Equations</u>
B14	Sine <u>Methods</u> for Quadrature and <u>Differential Equations</u>
B15	Stability of Stochastic <u>Differential Equations</u> with Respect-to Semi-Martingales
B16	The Boundary <u>Integral</u> Approach to Static and Dynamic Contact <u>Problems</u>
B17	The Double Mellin-Barnes Type <u>Integral</u> and Their <u>Applications</u> to Convolution <u>Theory</u>

Tabella 1: *Titoli di libri estratti dalla SIAM Review. Le parole sottolineate compaiono in piú di un titolo.*

a_{ij} è del tipo:

$$a_{ij} = L(i, j) \times G(i)$$

Si consideri ad esempio un piccolo database di titoli di libri. Nella tabella 1 sono riportati 17 titoli di libri estratti dal *SIAM Review*, Volume 54, Number 4. Le parole sottolineate sono parole chiave usate come referenti al titolo del libro. Il metodo di analisi utilizzato richiede che le parole chiave appaiano in piú di un titolo di libro.

A partire dalla tabella viene costruita una matrice termini-documenti di dimensione 16×17 mostrata in Fig. 3.

Mediante la SVD, la matrice A viene fattorizzata nel prodotto di tre matrici. La SVD fornisce il modello della struttura semantica latente a partire dalle matrici U , V e S . Queste matrici rappresentano una separazione delle relazioni originali in vettori

A. Murli, *Lezioni di Calcolo Parallelo*

Versione provvisoria solo per uso personale, soggetta ad errori.

Non è autorizzata la diffusione. Tutti i diritti riservati.

Terms	Documents														
<i>algorithms</i>	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
<i>application</i>	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
<i>delay</i>	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
<i>differential</i>	0	0	0	1	0	0	0	1	0	1	1	1	1	1	0
<i>equations</i>	1	1	0	1	0	0	0	1	0	1	1	1	1	1	0
<i>implementation</i>	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
<i>integral</i>	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
<i>introduction</i>	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
<i>methods</i>	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
<i>nonlinear</i>	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
<i>ordinary</i>	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
<i>oscillation</i>	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
<i>partial</i>	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
<i>problem</i>	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
<i>system</i>	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
<i>theory</i>	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1

Figura 3: Matrice termini-documenti corrispondente ai titoli di libri riportati nella tabella 1. Ogni elemento della matrice rappresenta la frequenza con cui il termine i occorre nel documento j .

linearmente indipendenti. L'uso di k componenti è equivalente ad approssimare la matrice A con A_k .

Calcoliamo la SVD troncata con $k = 2$ per ottenere l'approssimazione di rango 2, A_2 .

Utilizzando per la coordinata x la prima colonna di U_2 moltiplicata per il primo valore singolare σ_1 e per la coordinata y la seconda colonna di U_2 moltiplicata per il secondo valore singolare σ_2 , i termini possono essere rappresentati su di un piano cartesiano. Analogamente la prima colonna di V_2 moltiplicata per il primo valore singolare σ_1 fornisce la coordinata x e la seconda colonna di V_2 moltiplicata per il secondo valore singolare σ_2 fornisce la coordinata y dei documenti. In Fig. 4 sono rappresentati i termini e i documenti della matrice termini-documenti. Si osservi che i documenti e i termini che riguardano l'argomento “*differential equations*” sono posizionati tutti intorno all'asse x e i termini e i documenti riguardanti l'argomento “*algorithms and applications*” sono posizionati tutti intorno all'asse y . Questo suggerisce che l'insieme $\{B2, B4, B8, B9, B10, B13, B14, B15\}$ contiene titoli simili nel significato.

Per il recupero di informazioni, l'utente deve formulare una richiesta. Tale richiesta può essere rappresentata da un vettore nello spazio di dimensione k che deve essere confrontato con i documenti. Una richiesta è un insieme di parole e può essere rappresentata da:

$$\bar{q} = q^T U_k S_k^{-1} \quad (1)$$

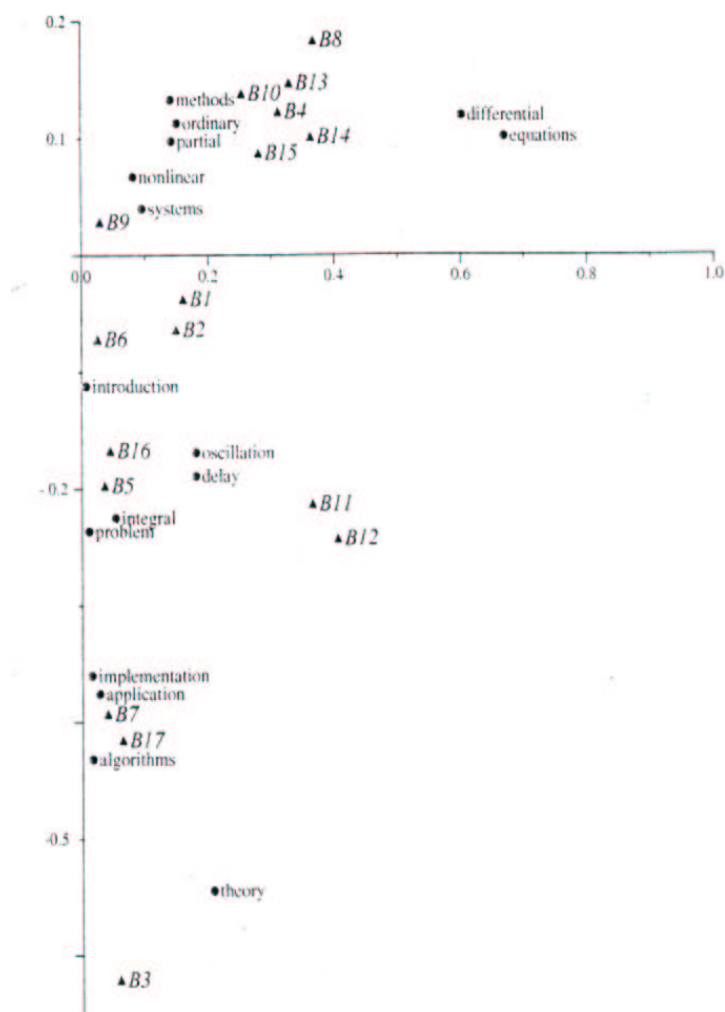


Figura 4: Rappresentazione nel piano cartesiano dei termini e dei documenti.

$$(0.0511 \ -0.3337) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} 0.0159 & -0.4317 \\ 0.0266 & -0.3756 \\ 0.1785 & -0.1692 \\ 0.6014 & 0.1187 \\ 0.6691 & 0.1209 \\ 0.0148 & -0.3603 \\ 0.0520 & -0.2248 \\ 0.0066 & -0.1120 \\ 0.1503 & 0.1127 \\ 0.0813 & 0.0672 \\ 0.1503 & 0.1127 \\ 0.1785 & -0.1692 \\ 0.1415 & 0.0974 \\ 0.0105 & -0.2363 \\ 0.0952 & 0.0399 \\ 0.2051 & -0.5448 \end{pmatrix} \begin{pmatrix} 4.5314 & 0 \\ 0 & 2.7582 \end{pmatrix}^{-1}$$

Figura 5: Coordinate della richiesta “application theory”. Il primo vettore a secondo membro rappresenta la nostra richiesta. Tenendo presente l'ordine in cui sono inserite le parole chiave nel database, nel vettore richiesta vi è 1 in corrispondenza di una parola richiesta e 0 in corrispondenza di una parola non richiesta. Il secondo termine a secondo membro è una matrice di dimensione 16×2 e rappresenta la matrice U_2 ottenuta mediante SVD troncato con $k = 2$. Il terzo termine a secondo membro è una matrice di dimensione 2×2 e rappresenta la matrice Σ ottenuta mediante SVD.

dove $\bar{q}$ è il vettore delle parole della richiesta dell'utente moltiplicato per gli appropriati pesi del termine: $q^T U_k$ rappresenta la somma di vettori di dimensione k e la moltiplicazione per S_k^{-1} pesa in modo diverso la dimensione.

Supponiamo di essere interessati ai documenti che riguardano l'argomento “*applications and theory*”.

Dal punto di vista matematico, le coordinate cartesiane della richiesta (vedi Fig. 5) sono determinate dall'equazione (1) e la richiesta è rappresentata dal punto marcato QUERY mostrato in Fig. 6.

Il vettore richiesta viene confrontato con i vettori dei documenti e i documenti vengono classificati in base alla loro similitudine alla richiesta. Una misura della similitudine è data dall'angolo delimitato dal vettore richiesta e dal vettore documento. I documenti il cui coseno supera un certo valore limite vengono restituiti all'utente.

La risposta alla nostra richiesta sarà ad esempio costituita dai documenti il cui vettore forma con il vettore richiesta un angolo il cui coseno è maggiore di 0.9. Tali documenti sono situati nella Fig. 6 tra l'asse negativo delle y e il vettore più grande. Analizziamo la complessità di tempo di questo algoritmo.

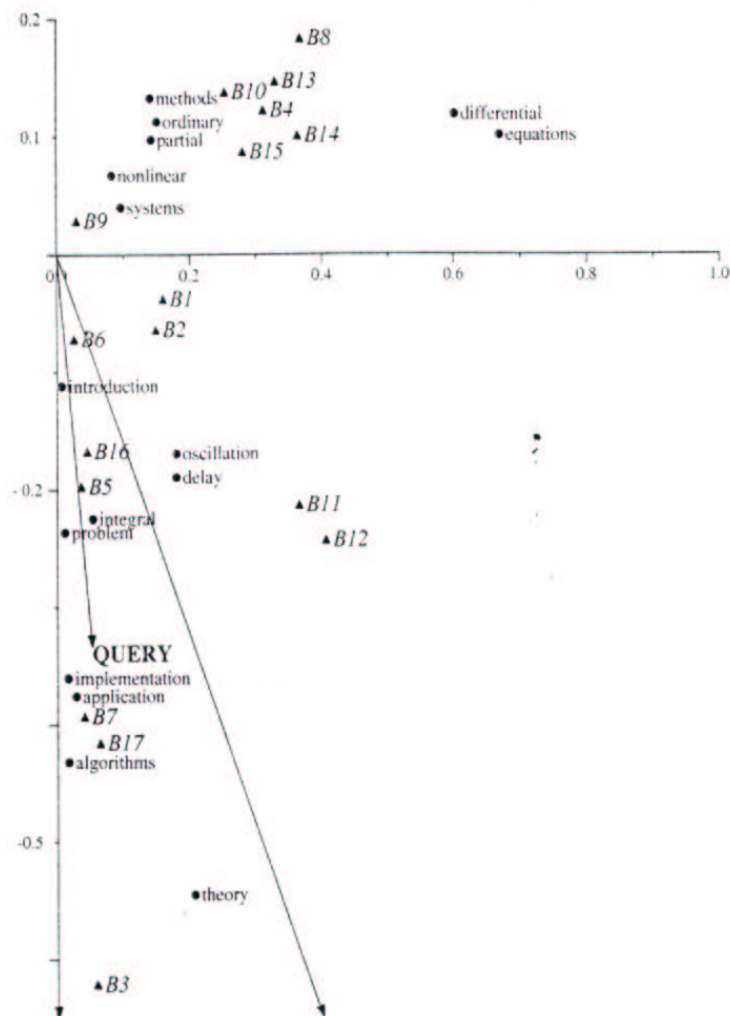


Figura 6: Rappresentazione nel piano cartesiano della richiesta denotata in figura con *QUERY*.

La complessità di tempo è data da $T(n) = kN^2$ con k numero di dimensioni considerate. Supponiamo che il motore di ricerca contenga 20 milioni di documenti (tale numero raddoppia ogni 5 mesi) e che risulti $k = 1000$. Questo vuol dire che $T(N) = kN^2 = 10^3 \cdot 2 \cdot 10^7 = 2 \cdot 10^{10}$, sono richiesti cioè 20 miliardi di operazioni per la risoluzione del problema.

△

L'esempio del motore di ricerca in Internet descrive una delle applicazioni di calcolo ad alte prestazioni più diffuse e più vicine alle nostre attività quotidiane, cioè la ricerca di una informazione sulla rete del web.

♣ Esempio 3

Quanto si è disposti ad aspettare il risultato di una ricerca sulla rete internet?

Prendendo spunto da questa "naturale" esigenza vediamo quali caratteristiche dovrebbe avere un calcolatore perché sia in grado di elaborare una nostra richiesta sulla rete internet in 1 sec (secondo).

Il tempo di esecuzione della richiesta, τ , è proporzionale alla complessità di tempo dell'algoritmo utilizzato, T , e al tempo di esecuzione di 1 operazione, μ :

$$\tau \cong T \cdot \mu$$

Dall'esempio 2, si è visto che la complessità computazionale dell'algoritmo LSI, nel caso di un motore di ricerca che deve elaborare 20 milioni di documenti, è:

$$T = 2 \cdot 10^{10}$$

Quindi, imponendo che $\tau = 1 \text{ sec}$ si ha:

$$1 \text{ sec} = 20 \text{ Gflop} \cdot \mu$$

cioè

$$\mu = \frac{1}{2 \cdot 10^{10}} \text{ sec} = 0.5 \times 10^{-10} \text{ sec}$$

Indicata con d la distanza tra la CPU e la memoria e con c la velocità della luce, affinché l'attesa sia di 1; sec deve risultare

$$d = c \cdot \tau \leq 3 \cdot 10^{11} \text{ mm/sec} \cdot \frac{1}{2 \cdot 10^{10}} \text{ sec} \approx 15 \text{ mm}$$

A. Murli, *Lezioni di Calcolo Parallelo*

Versione provvisoria solo per uso personale, soggetta ad errori.

Non è autorizzata la diffusione. Tutti i diritti riservati.

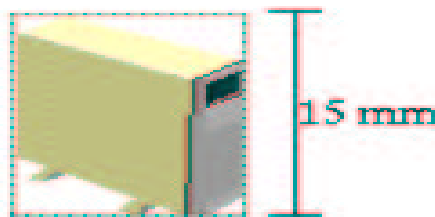


Figura 7: Affiché un motore di ricerca possa rispondere ad una richiesta in circa 1 sec le dimensioni del calcolatore devono essere di circa 15 mm.

In definitiva per avere la risposta in 1 sec occorrerebbe utilizzare un calcolatore (CPU + Memoria) piccolo più o meno quanto quello rappresentato in Fig. 7 in scala reale.

△

Da quanto fino ad ora illustrato, anche se con grande semplificazione, risulta evidente che con i calcolatori mono processore non è possibile ottenere “*alte prestazioni*”. **L'introduzione del parallelismo, ovvero del calcolo parallelo, cioè la decomposizione del problema in sottoproblemi risolti contemporaneamente su più calcolatori**, ha reso possibile il superamento di limiti tecnologici.

Oggetto di discussione dei capitoli seguenti saranno appunto alcuni aspetti metodologici alla base del Calcolo Parallelo. In particolare, data una applicazione verranno discusse alcune problematiche preliminari con l'intento di fornire una, seppur semplificata, risposta alle domande seguenti:

- Come può, e deve, essere *parallelizzata* una data applicazione?⁴

⁴Facendo riferimento al processo di risoluzione computazionale di un problema, l'introduzione del parallelismo può avvenire a vari livelli; ad esempio a livello della definizione del problema fisico - modelli disaccoppiati - a livello della discretizzazione - decomposizione del dominio - a livello dell'algoritmo - formulazione dell'algoritmo a blocchi - e infine, a livello del software - distribuzione delle operazioni nei nuclei computazionalmente più onerosi.

- Qual è il sistema più adatto?
- Quali sono gli algoritmi più adeguati?
- Quali strumenti software usare? (linguaggio, compilatore, programma di comunicazione)

In altre parole ciò richiede:

- a) Identificare le componenti del calcolo che possono “girare” in parallelo.
- b) Adottare una strategia opportuna per decomporre il problema (algoritmo, programma) in componenti parallele.
- c) Scegliere il modello di programmazione.
- d) Utilizzare gli strumenti software che dipendano dall'applicazione e dal modello di programmazione scelto.

Verrà data particolare attenzione al concetto di **decomposizione** ovvero alla suddivisione di un problema in più sottoproblemi, alla suddivisione dei dati e dei calcoli in più processi (o *tasks*); la decomposizione rappresenta l'idea di base del Calcolo Parallelo⁵. In particolare, nel **primo capitolo** verranno introdotte alcune idee di base del Calcolo Parallelo attraverso un semplice esempio: la somma di n numeri.

Nel **secondo capitolo** verrà introdotto lo schema funzionale di alcuni tipi di architetture parallele e illustrate le caratteristiche fondamentali dei calcolatori paralleli.

Nel **terzo capitolo** mediante alcuni esempi verrà spiegato uno dei paradigmi più utilizzati per la comunicazione tra i processi: il message passing.

Nel **quarto capitolo** verranno definiti i parametri di valutazione per il software in ambiente parallelo.

⁵ Altri concetti importanti alla base del Calcolo Parallelo sono la *concorrenza* e la *scalabilità*.

Nel **quinto capitolo** verranno discussi i principi alla base della progettazione di software parallelo per il calcolo matriciale.

Infine, nel **sesto capitolo** sarà presentata una panoramica delle principali architetture parallele che dagli anni '80 fino ad oggi (Luglio 2004) sono state presenti sul mercato.