



1

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

Basic Linear Algebra Subprograms

Anni '60 - inizio '70: algoritmi "naïve" (sequenze di "for")
- es. Libreria EISPACK (calcolo di autovalori $Ax=Ix$)

1973: sviluppo di BLAS (1)

- Libreria di 15 operazioni su vettori
- "AXPY" ($y = \alpha \cdot x + y$), prod. scalare, scale ($x = \alpha \cdot x$), etc
- 4 versioni (S/D/C/Z), 46 routines, 3300 linee di codice
- obiettivi
 - Accesso uniforme alle routine, leggibilità, documentazione
 - Robustezza, attenta scrittura del codice, attenzione all'aritmetica
 - Portabilità ed efficienza attraverso implementazioni specifiche
- Calcolo di q per BLAS 1 : $O(N^1)$ comm su $O(N^1)$ flop $\rightarrow q \sim O(1)$
- Usata in librerie come LINPACK (sistemi lineari densi)

2

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

BLAS-2

Limiti di BLAS-1

- Valore di q troppo grande ($q \sim 1$ richiede $t_{mem} \sim t_{flop}$ per avere $1/2$ della Perf*)
- Utile per calcolatori vettoriali come il CRAY X-MP/12 (SIMD) ma poco utile per i supercalcolatori degli anni '80

1984: sviluppo di BLAS-2

- Libreria di 25 operazioni quasi tutte tipo matrici/vettore
 - "GEMV": $y = \alpha \cdot A \cdot x + \beta \cdot y$, "GER": $A = A + \alpha \cdot x \cdot y^T$, "TRSV": $y = T^{-1} \cdot x$
 - 4 versioni (S/D/C/Z), 66 routines, 18K linee di codice
- Calcolo di q per BLAS 2 : $O(N^2)$ comm su $O(N^2)$ flop $\rightarrow q \sim O(1)$
 - Ancora poco efficiente per macchine con cache memory

3

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

BLAS-3

1987: BLAS-3

- Libreria di 9 operazioni quasi tutte tipo matrice/matrice
 - "GEMM": $C = A \cdot B + C$, "SYRK": $C = \alpha \cdot A \cdot A^T + \beta \cdot C$, "TRSM": $C = T^{-1} \cdot B$
 - 4 versioni (S/D/C/Z), 30 routines, 10K linee di codice
- Calcolo di q per BLAS-3: $O(N^2)$ comm su $O(N^3)$ flop
- $q \sim (4N^2)/(2N^3) \sim 1/N$
 - Possibilità di ottimizzare l'uso delle cache

BLAS1/2/3 disponibile su www.netlib.org/blas

- Solo implementazioni di riferimento non ottimizzate
 - Es: 3 loop innestati per GEMM

4

1

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

riassumendo

BLAS1	BLAS2	BLAS3
$q=O(1)$	$q=O(1)$	$q=O(1/N)$

↓

OBIETTIVO

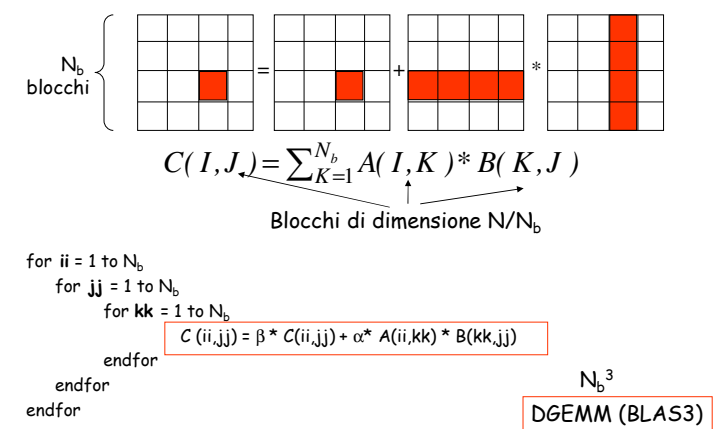
Nella riorganizzazione degli algoritmi
e' opportuno, ove possibile, **utilizzare moduli BLAS3**

5

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

esempio: prodotto di matrici

Soluzione
Suddividere le matrici a blocchi



$C(I,J) = \sum_{K=1}^{N_b} A(I,K) * B(K,J)$

Blocchi di dimensione N/N_b

```

for ii = 1 to N_b
  for jj = 1 to N_b
    for kk = 1 to N_b
      C(ii,jj) = beta * C(ii,jj) + alpha * A(ii,kk) * B(kk,jj)
    endfor
  endfor
endfor
    
```

N_b^3
DGEMM (BLAS3)

6

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

analisi dell'algoritmo

- Ricordiamo che per il prodotto di matrici il valore di $q \sim (4N^2)/(2N^3) \sim O(1/N)$
- Le prestazioni crescono con la dimensione di N
MA
- Limite: tre blocchi di A,B,C devono entrare nella cache, cosi' N non puo' essere arbitrariamente grande
- Poiche $3b^2 \leq M_{cache}$, si ha $b \leq (M_{cache}/3)^{1/2}$
- di fatto il valore di q e' limitato da $q > O(1/b) > O((M_{cache}/3)^{-1/2})$
- Teorema (Hong & Kung, 1981): Qualunque riorganizzazione di questo algoritmo e' limitata a $q = O((M_{cache})^{-1/2})$

7

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

Performance tuning

Un problema essenziale nell'utilizzo di BLAS e' definire i valori dei parametri legati alle dimensioni dei livelli di memoria

Come definire b (ordine dei blocchi)?

Implementazione
sviluppata dai
produttori hardware




Ottimizzazione
automatica del
codice

8

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

implementazione dei produttori

I maggiori produttori di CPU, sulla base della conoscenza specifica dell'hardware, hanno sviluppato:

- Libreria Intel Math Kernel Library (MKL)**
 - Ottimizzata per processori Intel
 - Contiene BLAS1, 2 e 3, FFT, Sparse Solver, LAPACK, ...
- Libreria AMD Core Mathematical Library (ACML)**
 - Ottimizzata per processori AMD
 - Contiene BLAS1, 2 e 3, FFT, LAPACK, ...
- Libreria CUDA Basic Linear Algebra Subprograms (CUBLAS)**
 - Ottimizzata per GPU NVIDIA
 - Contiene BLAS1, 2 e 3

9

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

ottimizzazione di un codice

E' una impresa dispendiosa in termini di tempo e di risorse uomo

- Differenze tra le CPU in termini di
 - clock,
 - dimensioni e gestione delle cache,
 - numero e profondita' delle pipeline di calcolo
- necessaria una conoscenza dettagliata e profonda dell'ambiente
- Inoltre per ogni processore e' necessario ripetere il processo di ottimizzazione.

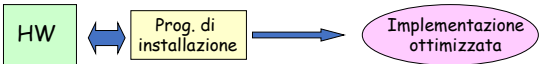
- IDEA: farlo fare alla libreria stessa

10

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

Automatic Performance Tuning

- obiettivo: far determinare al programma di installazione le caratteristiche del sistema
 - Numerosi test e benchmark in fase di installazione
 - Eseguiti una sola volta off-line e sempre valida sulla stessa architettura
 - Impiega circa un'ora per definire il codice ottimizzato
 - In generale approccio utile per algoritmi con esecuzione prevedibile e "ben strutturata" che non dipende dai dati (tolleranza, grado di sparsita',...)
 - Es. **Automatic Tuning Linear Algebra Subprograms (ATLAS)**



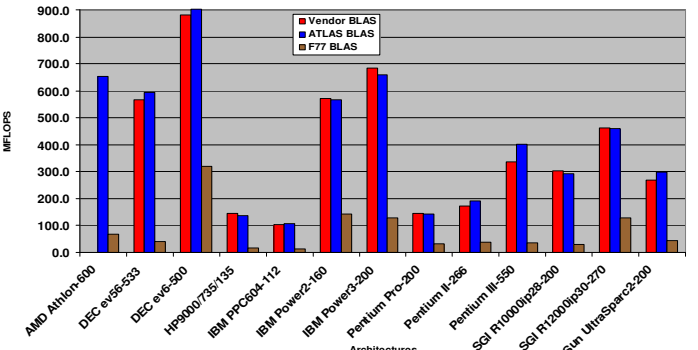
```

    graph LR
      HW[HW] --> Prog[Prog. di installazione]
      Prog --> Imp[Implementazione ottimizzata]
    
```

11

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

ATLAS: C=C+AB N=500



Architecture	Vendor BLAS (MFLOPS)	ATLAS BLAS (MFLOPS)	FFT BLAS (MFLOPS)
AMD Athlon-600	650	650	50
DEC ev6-553	580	600	50
DEC ev6-500	900	900	320
HP9000/725/125	150	150	20
IBM PPC604-112	120	120	20
IBM Power2-160	580	580	150
IBM Power3-200	680	680	150
Pentium Pro-200	150	150	20
Pentium II-266	180	180	20
Pentium III-350	320	380	20
SGI R1000/128-200	300	300	20
SGI R1200/160-270	450	450	150
Sun UltraSparc2-200	280	300	20

- ATLAS e' piu' veloce di ogni implementazione portabile "scritta a mano" ed ha prestazioni paragonabili a quelle specifiche per la macchina fornite dai vendors
- ATLAS e' scritta in C (www.netlib.org/atlas)

12


Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

Una macchina virtuale per l'Algebra Lineare

La disponibilita' di routine efficienti per le operazioni di base ci permette di riorganizzare gli algoritmi in questo contesto

Macchina Virtuale per l'Algebra Lineare

BLAS
(Basic Linear Algebra Subprograms)



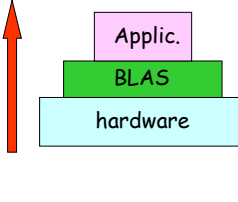
BLAS e' un esempio di tecnologia software che ha l'obiettivo di rendere piu' efficienti le applicazioni sovrastanti

13

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

andando verso l'alto

Avendo a disposizione una libreria di base efficiente per l'algebra lineare possiamo pensare di risolvere problemi piu' complessi (LU, LLT, $Ax=\lambda x, \dots$)



E' necessario riscrivere le applicazioni in termini di operazioni di base

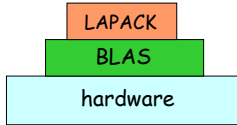
Possibilita' di sviluppo di Algoritmi a Blocchi

14

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

LAPACK

- LAPACK (1989) "Linear Algebra PACKage" - usa BLAS-3
 - Contenuti di LAPACK
 - Algoritmi che possono essere riscritti quasi per intero con BLAS3
 - Sistemi Lineari: calcola x tale che $Ax=b$
 - Minimi quadrati: calcola x che minimizza $\|r\|_2 \equiv \sqrt{\sum r_i^2}$ dove $r=Ax-b$
 - Algoritmi che possono essere riscritti per il 50% con BLAS3
 - "autovalori": calcola λ e x tale che $Ax = \lambda x$
 - Singular Value Decomposition (SVD): $A^T A x = \sigma^2 x$
 - Numerose varianti (A matrice a banda, $A=A^T$, etc)
 - 1582 routines, 490K linee di codice,
 - Ultima versione nel 2008, scritta per le nuove architetture
- www.netlib.org/lapack



15

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

ScaLAPACK (1995)

Rappresenta lo stato dell'arte per quanto riguarda le librerie parallele per l'algebra lineare densa

Ambienti paralleli a memoria distribuita

Modello di programmazione SPMD

Ultima versione 2007

Algoritmi di

- Fattorizzazione LU, LLT, QR, ...
- Risoluzione sistemi di equazioni
- Calcolo di autovalori, minimi quadrati, SVD

www.netlib.org/scalapack

16

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

obiettivi

Scalabilita' e prestazioni

- Moduli locali ottimizzati relativamente al rapporto comunicazioni/calcolo (BLAS)
- Bilanciamento del carico (distribuzione ciclica a blocchi 2D)

Portabilita' e modularita'

- Strumenti consolidati: BLAS, PBLAS, BLACS, LAPACK

Semplicita' di utilizzo

- Interfaccia quanto piu' simile possibile a LAPACK

17

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

organizzazione dei dati

Distribuzione ciclica a blocchi 2D

Matrice A

Suddivisione di A su una griglia 2x3 di processori

- assicura un buon bilanciamento del carico
- Diminuisce le comunicazioni
- buon rapporto comm/calc
- Poco intuitivo

18

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

ScaLAPACK = parallelizzazione di LAPACK

ScaLAPACK si ottiene da LAPACK sostituendo le chiamate di **BLAS** con opportune chiamate a **PBLAS**, una libreria parallela per le operazioni di base dell'algebra lineare

Necessita' di strumenti per implementare facilmente gli algoritmi per le operazioni di base dell'algebra lineare (es. SUMMA per il prodotto matrice matrice)

19

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

BLACS

Basic Linear Algebra Communication Subprograms

Una libreria di comunicazione sviluppata per diverse piattaforme (MPI, PVM, NX, ...)

Uno strumento per facilitare lo sviluppo di codici paralleli a memoria distribuita per l'algebra lineare

Orientato alla comunicazione di oggetti 2D (blocchi di matrici)

- Miglioramento della leggibilita'

implementazioni ottimizzate per varie piattaforme

- Miglioramento della portabilita' e dell'efficienza

Interfaccia Fortran / C

<http://www.netlib.org/blacs>

20

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

BLACS

I processi sono organizzati secondo una griglia 2D per facilitare il mapping dei blocchi di una matrice ai processi

- Processi identificati da una coppia di indici (myrow,mycol)

Per facilitare la distribuzione dei blocchi, le operazioni di comunicazione prevedono tre "ambiti" o "scope", in cui partecipano tutti i processi dell'ambito stesso

Per una griglia 2D ci sono 3 ambiti naturali

- Riga, colonna, tutti

Es. Griglia di Processi 2x3

	0	1	2	3	colonna
0	0	1	2	3	
1	4	5	6	7	riga
2	8	9	10	11	

21

Marco Lapegna
Parallel High Performance Computing
8 - arch. sw ScaLapack

PBLAS

Parallel BLAS

Libreria parallela a memoria distribuita per le operazioni di base dell'algebra lineare

Sviluppata utilizzando

- BLAS (per il calcolo)
- BLACS (per la comunicazione)

Obiettivi:

- Leggibilita' (il codice e' piu' corto)
- Efficienza (sfrutta l'efficienza di BLAS)
- Modularita' (fornisce grandi building blocks per lo sviluppo di applicazioni)
- Portabilita' (la dipendenza dalla macchina e' confinata in PBLAS)

<http://www.netlib.org/scalapack>

22