Una interfaccia GA ad alcune routines di ScaLAPACK

Giuseppe Vitillaro

Dipartimento di Chimica

Universita' degli Studi di Perugia

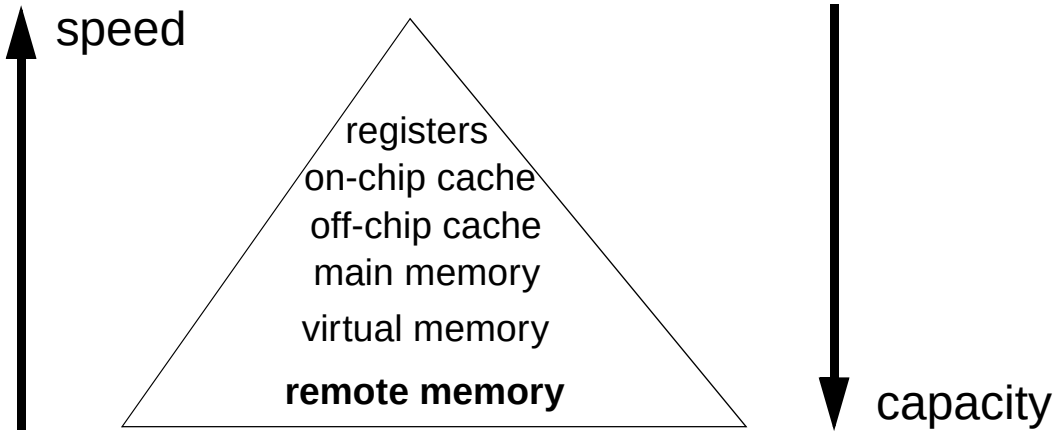
e-mail: <peppe@unipg.it>

Perugia, 30 Maggio 1996

Global Array (GA) toolkit

- Il Global Array (GA) toolkit implementa un modello di programmazione che permette un uso semplice di gerarchie di memoria con tempi di accesso non uniforme (NUMA: Non-Uniform Memory Access).

- Il costo dell'accesso a differenti tipi di memoria ha un impatto notevole sulle perfomance di calcolatori di architettura «sequenziale» come PC e Workstations:



NUMA architecture

Global Array (GA) toolkit

- I calcolatori paralleli con architetture MIMD (Multiple Instruction Multiple Data), a memoria distribuita, aggiungono un altro livello alla gerarchia: «**remote memory**», la memoria remota.

- La memoria remota viene acceduta attraverso lo scambio di messaggi «**message passing**» tra processori indipendenti interconnessi mediante un sottosistema di comunicazione (rete, switch, etc.).

- La memoria remota ha tempi di accesso e latenze (il tempo necessario per lo scambio del messaggio piu' piccolo) generalmente maggiori della memoria locale.

Global Array (GA) toolkit

- Nei calcolatori paralleli a memoria condivisa «**shared memory**» piu' processori condividono la stessa memoria centrale.

- Il modello di programmazione a memoria condivisa e', nella maggior parte dei casi, piu' semplice del modello «message passing».

- Il Global Array toolkit combina le caratteristiche del paradigma shared memory e di quello message passing in un'unica interfaccia di programmazione.

- Il concetto chiave del GA e' fornire un modo semplice ed efficiente di accedere matrici distribuite in blocchi logici nella memoria locale dei processori di una macchina parallela.

Global Array (GA) toolkit

- Il GA toolkit e' portabile su macchine parallele di differente architettura a memoria condivisa o distribuita. Le architetture correntemente supportate sono:

  Message Passing

  - IBM SP1 e SP2
  - Intel IPSC, Delta, PARAGON
  - Cray-T3D

  Shared Memory

  - KSR-2

  Networks of workstations

  - SUN
  - SGI
  - IBM

Global Array (GA) toolkit

- Caratteristiche fondamentali:
  - I processi possono comunicare creando ed accedendo matrici distribuite e scambiando messaggi;
  - Le matrici vengono suddivisi in blocchi e distribuite tra i vari processi;
  - Ogni processo puo' accedere in modo indipendente ed asincrono qualsiasi porzione, «patch», di una GA distribuita;
  - Ad ogni processo e' garantito accesso «veloce» a qualche porzione della matrice distribuita ed accesso «lento» per la parte rimanente. Questa differenza di velocita' «definisce» i dati come «locali» o «remoti».
  - Ogni processo puo' determinare quale porzione di una qualsiasi matrice distribuita e' memorizzata

Global Array (GA) toolkit

«localmente». Ogni elemento di una GA e' locale ad uno ed un solo processore.

- Il GA toolkit e' stato sviluppato presso l'Environmental Molecular Sciences Laboratory (**EMSL**) del Pacific Northwest National Laboratory (**PNL**), Richland, Washington («**http://www.emsl.pnl.gov:2080/**»).

- Riferimenti sul GA toolkit (descrizione dell'interfaccia, piattaforme supportate, sources, dati sulle perfomances) sono accedibili mediante il WWW all'indirizzo: **«http://www.emsl.pnl.gov:2080/docs/global/ga.html».**

- L'autore Jarek Nieplocha puo' essere contattato all'indirizzo «j_nieplocha@pnl.gov».

Message Passing

- Il calcolo parallelo puo' essere basato su diversi modelli concettuali.

- Il modello «**message passing**» si basa sull'idea di un «insieme di processori», ognuno dei quali accede direttamente solo la propria memoria «locale», ma e' in grado di comunicare con gli altri processori inviando o ricevendo «messaggi».

  Il contenuto della memoria locale ad un processore puo' essere acceduto da un altro «solamente» mediante una serie di operazioni, che coinvolgono entrambi i processori, in grado di trasferire fisicamente i dati dalla memoria locale di un processore a quella dell'altro.

Message Passing

- Il modello astratto del «message passing» e' oggi utilizzabile in differenti implementazioni software proprietarie e di pubblico dominio.

- Alcune «librerie» di pubblico dominio come:

  - PVM (Parallel Virtual Machine System)
  - TCGMSG

    (Theoretical Chemistry Group Message Passing System)
  - P4 (Portable Programs for Parallel Processors)

  forniscono implementazioni di interfacce di programmazione basate sul modello message passing.

  Sono «portabili» (a livello applicativo) su differenti piattaforme hardware (anche a memoria condivisa).

MPI

- La richiesta di standardizzazione e portabilita' ha portato ricercatori e case commerciali ad iniziare un processo di standardizzazione del modello «message passing».

- Questo processo si' e' concretizzato nella definizione e nell'implementazione di un modello «standard» definito **MPI, «Message Passing Interface»** («http://www.mcs.anl.gov/mpi/»).

- MPI e' una «libreria» (non un linguaggio) che si propone come scopo la definizione sintattica e semantica di una serie di subroutines e funzioni (C e F77).

MPI

- Il modello di programmazione «message passing» si basa su due operazioni fondamentali:
  - send per spedire un messaggio
  - receive per ricevere un messaggio

  Nel caso MPI le primitive che implementano queste operazioni hanno la forma:

```
MPI_Send(buf, count, datatype, dest, tag, comm)

MPI_Recv(buf, count, datatype, source, tag, comm, status)
```

  Lo scambio di messaggi avviene tra un processo «sorgente» ed un «destinatario» in uno specifico «gruppo» di processi definiti dall'argomento «communicator».

(buf, count, datatype) definiscono il «tipo» ed il «numero» di messaggi che vengono scambiati nella singola operazione.

MPI

- Lo standard MPI (versione 1.1 Giugno ‘95) definisce 125 primitive, ma la maggior parte delle funzionalita’ possono essere definite in 6 funzioni fondamentali:
  - MPI_INIT — Inizializza MPI
  - MPI_COMM_SIZE — Numero di processori
  - MPI_COMM_RANK — Processore corrente
  - MPI_SEND — Spedisce un messaggio
  - MPI_RECV — Riceve un messaggio
  - MPI_FINALIZE — Termina MPI
- E’ disponibile un package di dominio pubblico che implementa lo standard: **MPI_CH** («http://www.mcs.anl.gov/Projects/mpi/mpich/index.html»)

BLACS

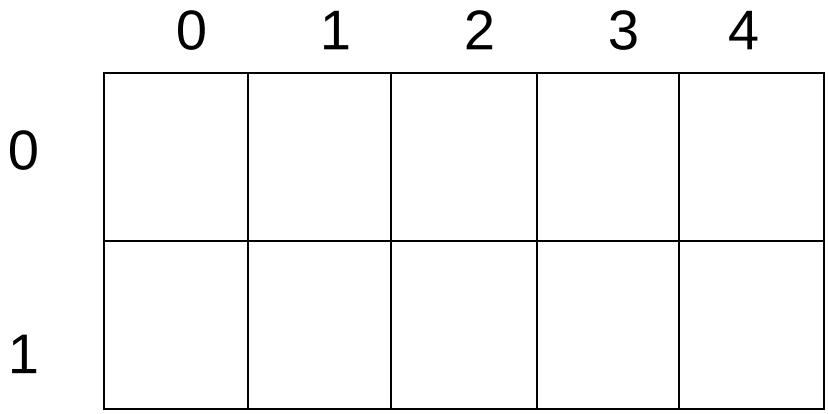
- La libreria **BLACS** «Basic Linear Algebra Communication Subprograms» e' un insieme di subroutines progettate con lo scopo di semplificare l'uso del modello di programmazione message passing **(«http://www.netlib.org/blacs/index.html»).**

- Fornisce le primitive di comunicazione necessarie alla implementazione di una libreria portabile, generalizzata, per le operazioni fondamentali dell'algebra lineare, su macchine parallele a memoria distribuita.

- Il modello astratto message passing e' troppo generale per poter essere utilizzato con efficacia a questo scopo.

BLACS

- Le subroutines BLACS operano su matrici a due dimensioni distribuite su una griglia bidimensionale di processori.



10 processori in una grglia di 2 righe per 5 colonne

- Le BLACS sono portabili su diverse librerie message passing ed architetture hardware: PVM, MPI, SP1/2 MPL, Intel NX, TMC CM-5 CMMD.

PBLAS

- La libreria **PBLAS** «Parallel Basic Linear Algebra Subprograms» e' una collezione di routines, per macchine parallele a memoria distribuita, che forniscono le funzioni fondamentali dell'algebra lineare.

- Il loro obiettivo e' quello di fornire al programmatore l'interfaccia per le operazioni fondamentali tra vettore-vettore, matrice-matrice e matrice-vettore (matrici a due dimensioni).

- Le PBLAS sono modellate sullo stesso schema della libreria **BLAS** «Basic Linear Algebra Subprograms», che risolve gli stessi problemi su macchine sequenziali.

PBLAS

- Le PBLAS, parte del progetto ScaLAPACK, utilizzano la libreria BLACS per il message passing e la libreria BLAS per le operazioni fondamentali dell'algebra lineare.

- L'efficienza dell'implementazione BLAS e' fondamentale per le perfomances computazionali: su RISC/6000 le PBLAS possono utilizzare la libreria IBM ESSL (nella versione POWER2 sui nodi SP2) o le blas IBM.

- L'efficienza del message passing dipende dalla libreria di base e dalla implementazione BLACS: su SP2 le librerie MPL (3.2) ed IBM MPI (4.1) in HPS user space mode (switch).

PBLAS

- Il modello corrente delle PBLAS assume che le matrici siano distribuite sui processori in uno schema a blocco ciclico (**SBS** «Square Block Scattered» decomposition):

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}
a_{51}	a_{52}	a_{53}	a_{54}	a_{55}

matrice 5x5 decomposta in blocchi 2x2

	0		1	
0	a_{11}	a_{12}	a_{15}	a_{13}	a_{14}
	a_{21}	a_{22}	a_{25}	a_{23}	a_{24}
	a_{51}	a_{52}	a_{55}	a_{53}	a_{54}
1	a_{31}	a_{32}	a_{35}	a_{33}	a_{34}
	a_{41}	a_{42}	a_{45}	a_{43}	a_{44}

punto di vista di una griglia di processi 2x2

ScaLAPACK

- **ScaLAPACK** «A Scalable Linear Algebra Package for Distributed Memory Concurrent Computers». Include subroutines F77 per la:
  - fattorizzazione LU, Cholesky, ortogonale
  - soluzione di sistemi di equazioni lineari
  - inversione di matrici
  - riduzione di matrici nella forma di Hessenberg, bidiagonale e tridiagonale
  - risoluzione di problemi di autovalore ed autovettore per matrici non-simmetriche, simmetriche ed hermitiane

  Il software e' disponibile per problemi in singola o doppia precisione reale o complessa, su macchine parallele a memoria distribuita

  («http://www.netlib.org/scalapack/index.html»).

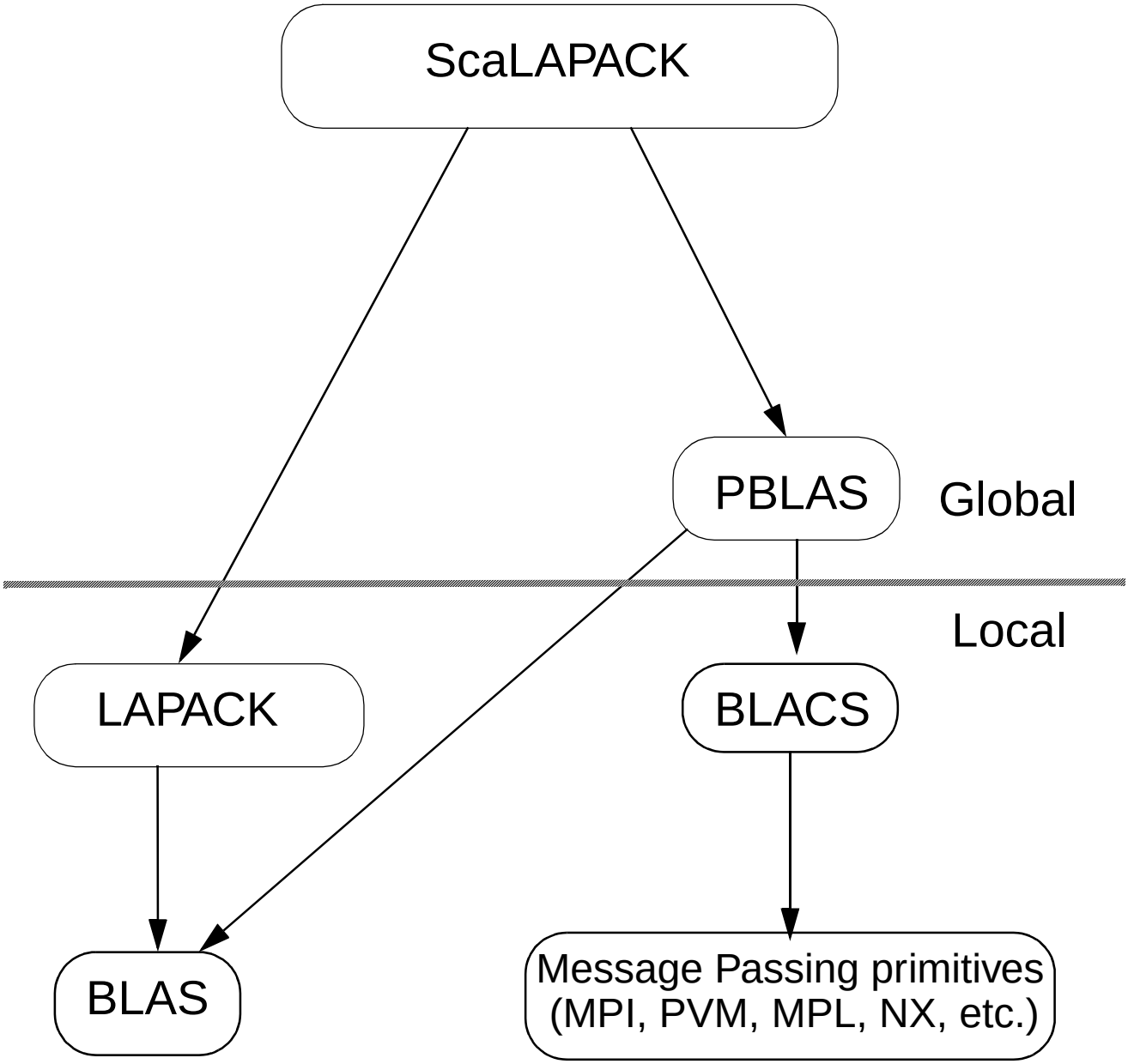
ScaLAPACK

- La libreria ScaLAPACK usa le routines della libreria PBLAS.

  E' utilizzabile su diverse piattaforme hardware: IBM SP1 ed SP2, Cray T3, INTEL, Thinking Machine CM-5, cluster di workstations e su tutti i sistemi nei quali siano disponibile le interfacce message passing PVM o MPI.

- ScaLAPACK e' una estensione della libreria LAPACK «Linear Algebra Package» («http://www.netlib.org/lapack/index.html).

ScaLAPACK


ScaLAPACK
PBLAS
Global
Local
LAPACK
BLACS
BLAS
Message Passing primitives
(MPI, PVM, MPL, NX, etc.)

Sistemi di equazioni lineari

- Dato il problema: $AX = B$

  dove A,B sono matrici quadrate di ordine n ed X e' un vettore di dimensione n, la matrice A puo' essere fattorizzata nella forma:

  - LU nel caso di una matrice A generale:

    $A = P\,LU$

    dove L e' una matrice sottotriangolare, U una matrice sopratriangolare e P e' una matrice di permutazione;

  - Cholesky nel caso di una matrice A simmetrica definita positiva:

    $A = LL^T$ oppure $A = U^TU$

    dove L e' una matrice sottotriangolare ed U una matrice sopratriangolare.

Sistemi di equazioni lineari

- La fattorizzazione della matrice A viene utilizzata per la risoluzione del sistema di equazioni lineari e per l'inversione.

- ScaLAPACK include routines per:
  - fattorizzazione LU e Cholesky
  - la risoluzione del sistema lineare (forward o backward substitution)
  - l'inversione di matrice.

- Fattorizzazione LU (doppia precisione reale):
  - PDGETRF fattorizzazione LU di A
  - PDGETRS risoluzione di AX=B
  - PDGETRI inversione di A

Sistemi di equazioni lineari

- Fattorizzazione di Cholesky:
  - PDPOTRF — fattorizzazione LLT di A
  - PDPOTRS — risoluzione di AX = B
  - PDPOTRI — inversione di A
- L'interfaccia di PDPOTRF:

```
SUBROUTINE PDPOTRF( UPLO, N, A, IA, JA, DESCA, INFO )
```

dove:

UPLO	forma 'U' Upper, 'L' Lower
N	ordine del problema
A	matrice distribuita
IA,JA	indice di riga/colonna globali di A
DESCA	descriptor della matrice A
INFO	codice di errore

Il decriptor della matrice DESCA contiene le informazioni relative alla distribuzione della matrice A.

Decomposizione SBS

```
In entrata A contiene la matrice da fattorizzare, in
uscita il fattore di Cholesky U o L.

Tutti i processi chiamano PDPOTRF con identici
parametri, tranne per la matrice A=sub(A) che identifica
la parte locale della matrice A decomposta in forma SBS.
```

- L’interfaccia PBLAS e quindi lo ScaLAPACK prevedono che la matrice globale A sia distribuita sui vari processori nel formato SBS (Square Block Scattered).

- L’interfaccia GA puo’ essere utilizzata per manipolare matrici distribuite e trasformarle da una decomposizione a blocchi (formato interno del GA) ad una decomposizione SBS.

- E’ quindi possibile utilizzare il modello di programmazione del GA toolkit e chiamare le

routine di ScaLAPACK per risolvere sistemi di equazioni lineari.

GA toolkit e ScaLAPACK

- L'interfaccia e' costituita da una coppia di routines che convertono il formato a blocchi GA da e nel formato SBS ScaLAPACK e da una serie di routines che interfacciano le routines PDGETRx, PDPOTRx (solo per il caso in doppia precisione reale)

- Le routines di interfaccia:
  - convertono le matrici in input nel formato SBS
  - costruiscono gli argomenti ScaLAPACK
  - chiamano le routines ScaLAPACK
  - convertono le matrici di output nel format GA.

- Lo stesso template logico puo' essere utilizzato per interfacciare altre routines ScaLAPACK.

GA toolkit e ScaLAPACK

- Routine GA per la fattorizzazione LLT di Cholesky:

```
ga_create(MT_DBL, n, n, 'A', 1, 1, g_A)
...
ga_llt_f('L', g_A, -1)
```

- L'interfaccia GA e':
  - typeless
  - di semplice uso
  - compatibile con il modello GA
- Sono disponibili i drivers per il test ed il benchmarking dell'interfaccia.

- I tests sono stati eseguiti su RISC/6000 e su SP2 utilizzando il package MPI_CH (1.0.12).

GA toolkit e ScaLAPACK

- E’ stato da poco annunciata la versione 1.2 di ScaLAPACK che contiene routines per il calcolo di tutti gli autovalori ed autovettori di una matrice simmetrica reale o complessa hermitiana.

- E’ da pochi giorni disponibile la User Guide della libreria ScaLAPACK.

- Al GA toolkit verra’ aggiunto il supporto per il tipo dati complesso e cio’ permettera’ di scrivere una interfaccia anche per questo insieme di routines ScaLAPACK.