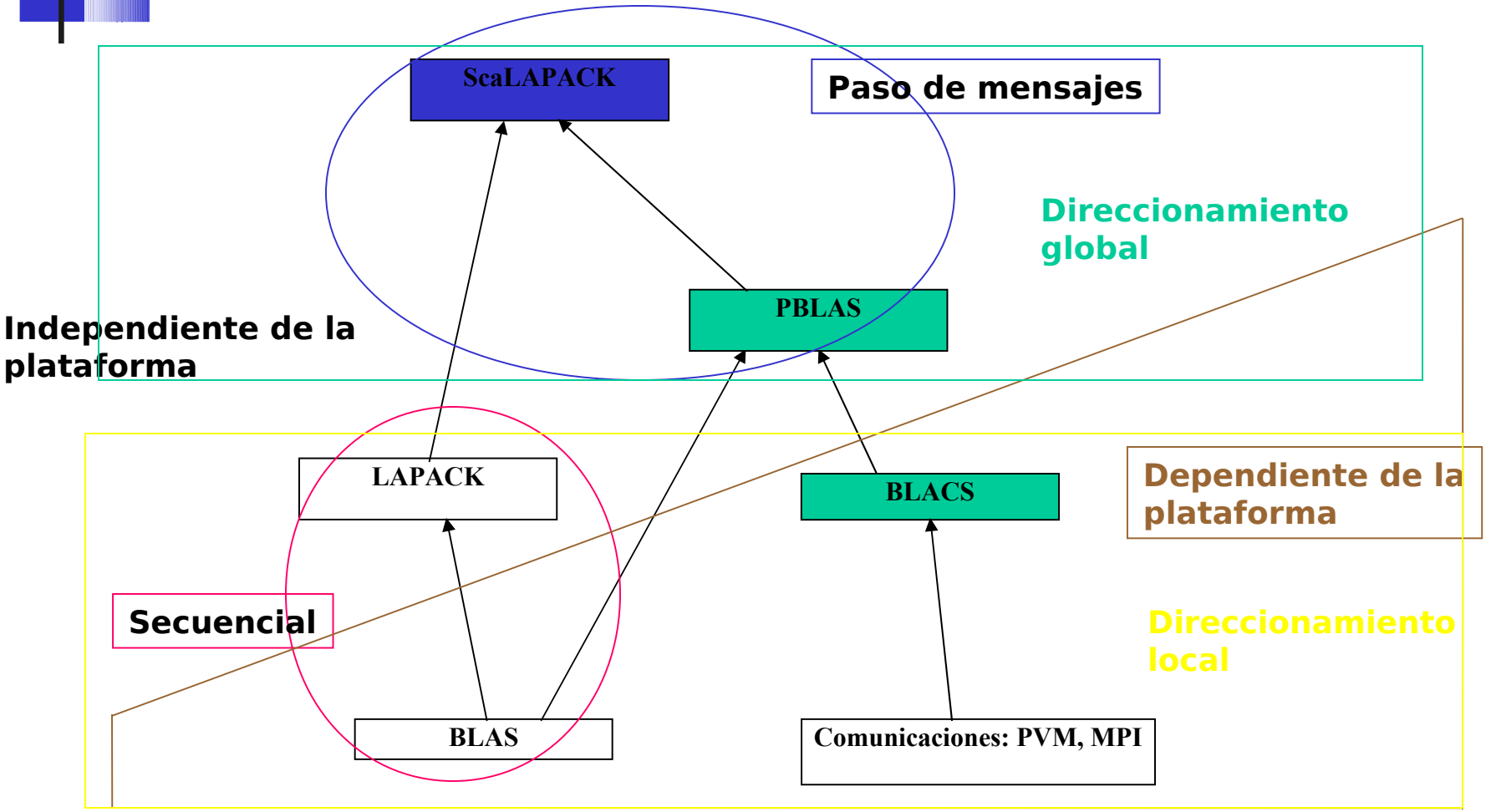
Librerías paralelas
BLACS, PBLAS, ScaLAPACK

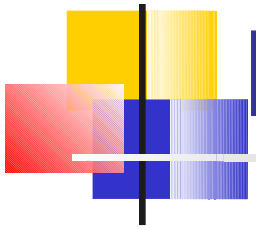
Domingo Giménez
Universidad de Murcia
http://dis.um.es/~domingo

Contenido

- Introducción
- Librería BLACS
- Librería PBLAS
- Librería ScaLAPACK

Introducción





BLACS

- Rutinas básicas de comunicación de matrices.
- Portabilidad para rutinas de comunicación de álgebra lineal.
- Se usa en ScaLAPACK para llevar a cabo las comunicaciones:
  - Desarrollo de rutinas paralelas de álgebra lineal con llamadas a BLAS para computación y a BLACS para comunicaciones.
- Se pueden usar para hacer comunicaciones en programas de álgebra lineal con paso de mensajes.

BLACS

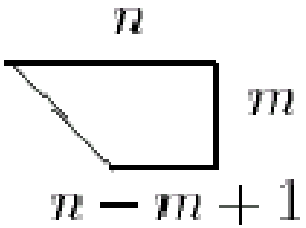
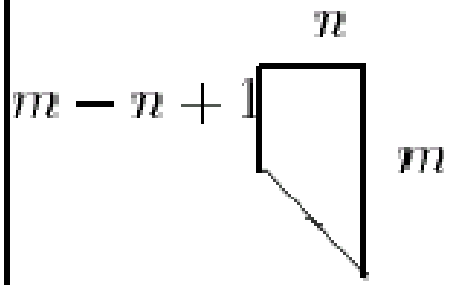
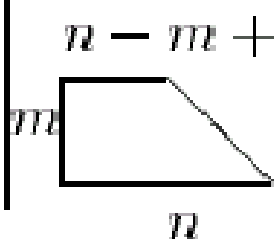
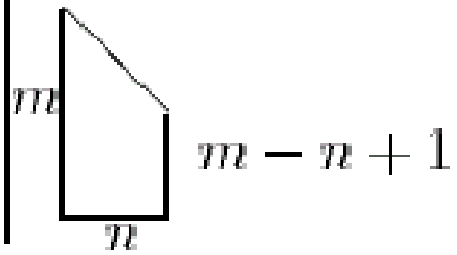
- Sobre paquetes generales de comunicación (PVM, MPI, ...)
- Facilitar el uso de rutinas de comunicación en problemas de álgebra lineal:
  - Al ser rutinas específicas se simplifica su uso.

BLACS

- Almacenamiento por columnas (estilo Fortran)
- Tipos de matrices:
  - Rectangulares generales
    - M filas, N columnas, LDA leading dimension
  - Trapezoidales
    - M, N, LDA
    - UPLO : trapezoidal superior o inferior
    - DIAG : se incluye o no la diagonal

BLACS

- Matrices trapezoidales

UPLO	M ≤ N	M > N
"U"		
"L"		

BLACS

- Se utiliza malla bidimensional de procesos: más escalabilidad que unidimensional
  - P=R*C procesos se mapean en una malla RxC

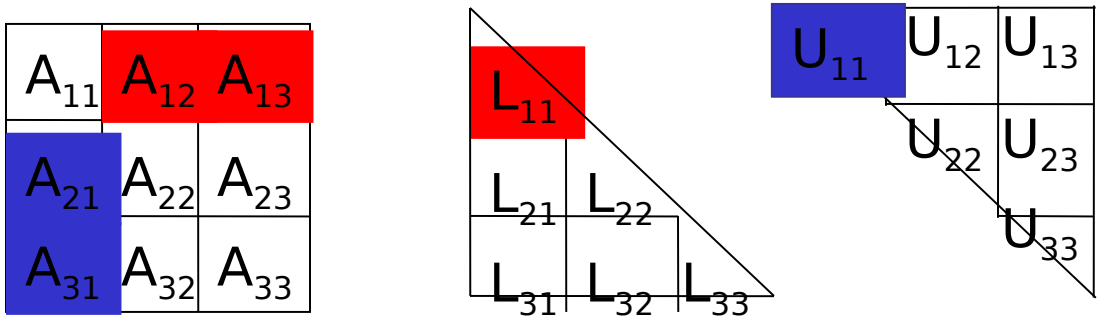
	0	1	2	3
0	0	1	2	3
1	4	5	6	7

BLACS

- Comunicaciones punto a punto
- Comunicaciones colectivas
  - Ámbito (scope) de una comunicación colectiva en una malla 2D
    - Por filas: participan todos los procesos en la fila
    - Por columnas: todos los procesos en la columna
    - Todos: todos los procesos en la malla

BLACS

- Factorización LU



Paso 1: $A_{11}=L_{11}U_{11}$ secuencial

Paso 2: $[A_{12},A_{13}]=L_{11}[U_{12},U_{13}]$ necesita comunicación en fila

Paso 3: $[A_{21}^T,A_{31}^T]=U_{11}^T[L_{21}^T,L_{31}^T]$ necesita comunicación en columna

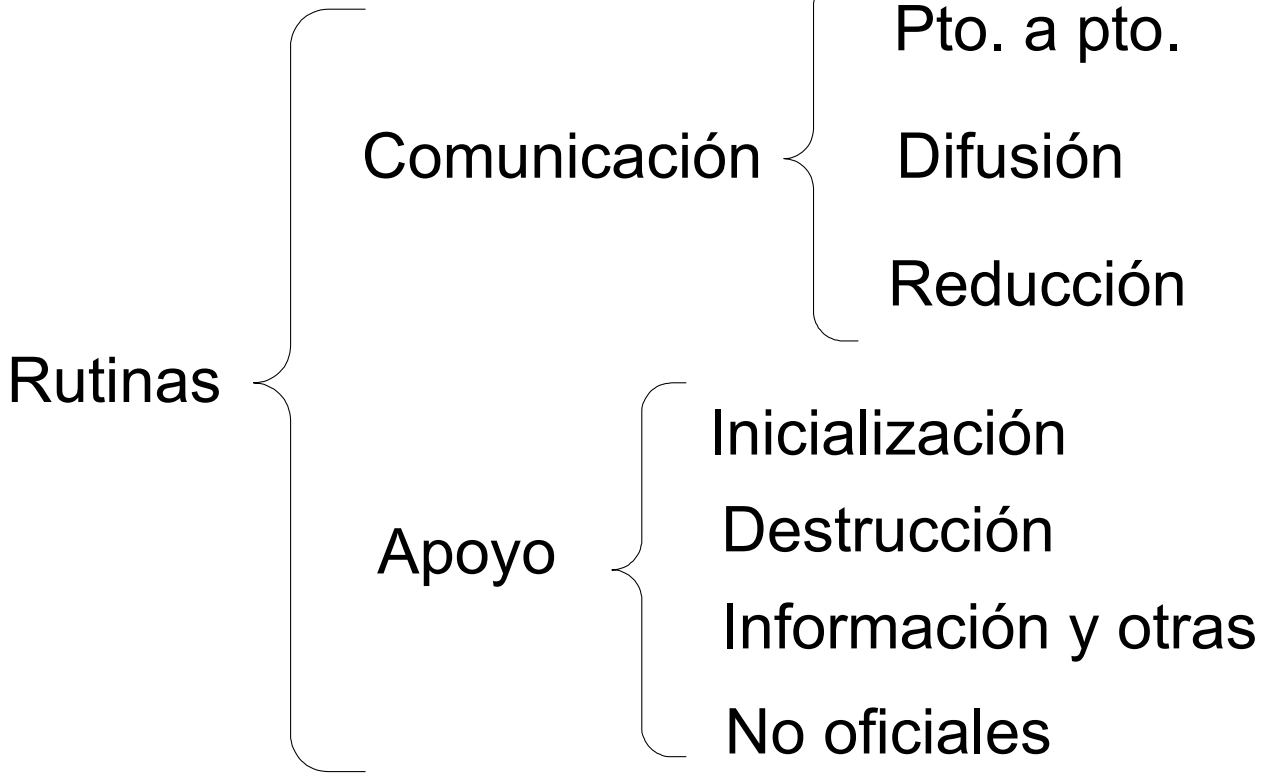
BLACS

- Contextos
  - A cada malla lógica de procesos se asigna un contexto BLACS.
  - Permiten:
    - Crear grupos de procesos.
    - Crear varias submallas solapadas o disjuntas.
    - Las comunicaciones de procesos en una submalla no interfieren con las de otra submalla.

BLACS

- Características de las comunicaciones:
  - BLACS genera identificador de la comunicación:
    - Simplifica la programación
    - Facilita el trabajo de varios programadores
  - Niveles de bloqueo:
    - En la mayoría de las comunicaciones hay bloqueo global: cuando se acaba la operación el buffer de comunicaciones está listo para ser usado
    - En el envío punto a punto el bloqueo es local: el que envía puede seguir trabajando aunque el que recibe no haya recibido

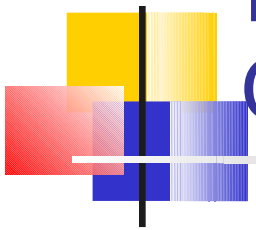
BLACS: Rutinas



BLACS: Rutinas.
Comunicación Pto a pto

- Nomenclatura de las rutinas punto a punto y difusión

vXXYY2D	
Tipo de datos: v	I (Integer), S, D, C, Z
Forma de la matriz: XX	GE TR
Tipo Comunicación: YY	SD, RV (Send, Recieve) BS, BR (Bcast send, recv)



BLACS: Rutinas.
Comunicación Pto a pto

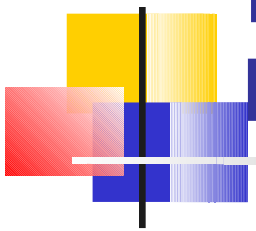
- Envío
  - vGESD2D(ICONTXT,M,N,A,LDA,RDEST,CDEST)
  - vTRSD2D(ICONTXT,UPLO,DIAG,M,N,A,LDA,RDEST,CDEST)
- Recepción
  - vGERV2D(ICONTX,M,N,A,LDA,RSRC,CSRC)
  - vTRRV2D(ICONTX,UPLO,DIAG,M,N,A,LDA,RSRC,CSRC)

BLACS: Punto a punto

```
CALL BLACS_GRIDINFO(ICONTXT,NPROW,NPCOL,MYROW,MYCOL)

IF (MYROW.EQ.0 .AND. MYCOL.EQ.0) THEN
    CALL DGESD2D(ICONTXT,5,1,X,5,1,0)
ELSE IF (MYROW.EQ.1 .AND. MYCOL.EQ.0) THEN
    CALL DGERV2D(ICONTXT,5,1,Y,5,0,0)
END IF
```

- DGESD2D(contexto , m , n , datos , lda , fila pro. , columna pro.)
- DGERV2D(contexto , m , n , datos , lda , fila pro. , columna pro.)

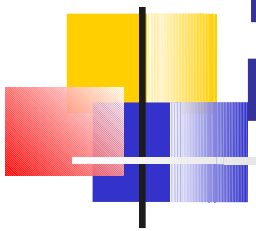


BLACS: Rutinas.
Difusión

- Envío
  - vXXBS2D(ICONTX,SCOPE,TOP[,UPLO,DIAG],M,N,A, LDA)
- Recepción
  - vXXBR2D(ICONTX,SCOPE,TOP[,UPLO,DIAG],M,N,A, LDA,RSRC,CSRC)

BLACS: Topologías

- Se indica topología lógica en que se hacen las comunicaciones:
  - ´ ´, dependiente del sistema
  - ´I´, anillo ascendente
  - ´D´, anillo descendente
  - ´H´, hipercubo
  - ´F´, completamente conectado
  - ´2´, árbol de broadcast con 2 ramas
  - ... (ver la guía de referencia)



BLACS
Difusión - Topologías

- Basadas en anillo (pipelining)
  - Anillo unidireccional
  - Anillo dividido
  - Multianillo
- Basadas en árboles (non-pipelining)
  - Hipercubo
  - Árbol general

BLACS: Difusión

```
CALL BLACS_GRIDINFO(ICONTXT,NPROW,NPCOL,MYROW,MYCOL)

IF (MYROW.EQ.0) THEN
    IF (MYCOL.EQ.0) THEN
      CALL DGEBS2D(ICONTXT,'Row',' ',5,1,A,5)
    ELSE
      CALL DGEBR2D(ICONTXT,'Row',' ',5,1,A,5,0,0)
END IF
```

- DGEBS2D(contexto , scope, top, m , n , datos , lda)
- DGEBR2D(contexto , scope, top, m , n , datos , lda , fila pro. , columna pro.)

scope: 'Row', 'Column', 'all'

top: topología

BLACS: Combinación

- Operaciones con todos los elementos de una matriz:
  - Máximo (en valor absoluto)

  xGAMX2D(ICONTX,SCOPE,TOP,M,N,A,LDA,RA,CA,RCFLAG,RDEST,CDEST)
  - Mínimo (en valor absoluto)

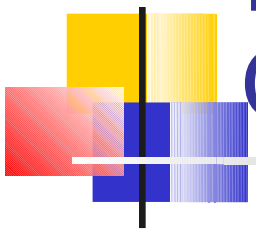
  xGAMN2D(ICONTXT,SCOPE,TOP,M,N,A,LDA,RA,CA,RCFLAG,RDEST,CDEST)
  - Suma

  xGSUM2D(ICONTXT,SCOPE,TOP,M,N,A,LDA,RDEST,CDEST)

RA y CA almacenan las coordenadas del proceso que contiene el máximo o mínimo

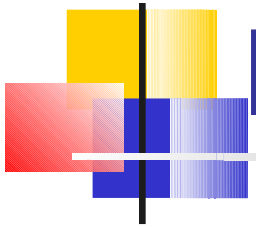
RCFLAGS: leading dimension de RA y CA, si –1 no se referencian

Si RDEST=-1 el resultado se deja en todos los procesos



BLACS
Combinación - Topologías

- General Tree Gather
- Intercambio bidireccional



BLACS: Rutinas de soporte

- Inicialización
  - BLACS_PINFO(mypnum,nprocs)
  - BLACS_SETUP(mypnum,npros). Sólo en PVM, para añadir procesos
  - BLACS_GRIDINIT(context,order,nprow,npcol). Mapea los procesos y devuelve identificador de contexto
  - BLACS_GRIDMAP(context,pmap,ldpmap,nprow,npcol). Indica cómo mapear los procesos respecto a la numeración en la máquina

BLACS: Rutinas de soporte

- Destrucción
  - BLACS_FREEBUF(context,wait). Libera los buffers usado por BLACS. wait indica si hay que esperar para liberar los buffers en operaciones no bloqueantes.
  - BLACS_GRIDEXIT(context). Libera el contexto.
  - BLACS_ABORT(context,errornum). Para abortar un proceso.
  - BLACS_EXIT(continue). Se debe llamar en un proceso cuando termina de usar BLACS. Se puede indicar que se seguirá usando el sistema de comunicaciones.

BLACS: Rutinas de soporte

- De propósito general:
  - BLACS_GET(context,what,val). Devuelve información interna. what indica qué información devolver en el array val.
  - BLACS_SET (context,what,val). Establece valores internos de BLACS.

BLACS: Rutinas de soporte

- De información:
  - BLACS_GRIDINFO(context,nprow,npcol,myprow,mypcol). Devuelve información de número de procesos por fila y columna e identificación de proceso.
  - int BLACS_PNUM(context,prow,pcol). Devuelve número de proceso dada la fila y columna.
  - BLACS_PCOORD(context,pnum,prow,pcol). Devuelve coordenadas del proceso dado su número.

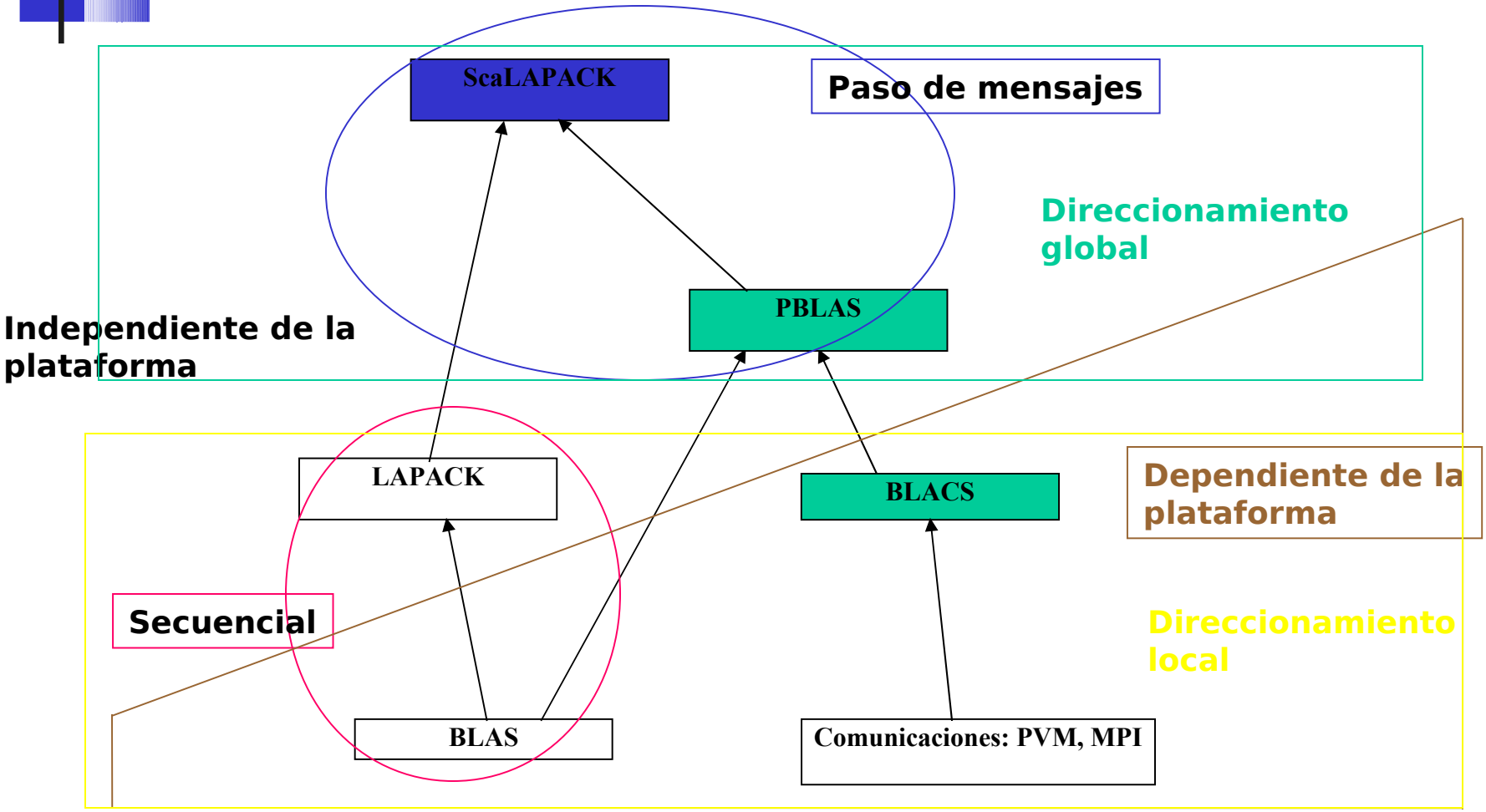
BLACS: otras rutinas

- BLACS_BARRIER(context,scope). Scope indica si la fila, columna, o la malla.
- Rutinas no oficiales:
  - No forman parte del BLACS estándar.
  - Pueden no estar en todas las implementaciones.
  - Ejemplos:
    - De tiempo: DCPUTIME, DWALLTIME
    - De identificadores: KSENDID, KRECVID, KBSID,KBRID

BLACS: Referencias

- BLACS:
  - Página principal: http://www.netlib.org/blacs/
  - LAPACK Working Note 94, "A User's Guide to the BLACS v1.1", Jack J. Dongarra, R. Clint Whaley, May 5, 1997

Librería PBLAS
(Parallel Basic Linear Algebra Subprograms)



PBLAS

- Funcionalidad similar a la de BLAS, en memoria distribuida. Que sea estándar para memoria distribuida como BLAS lo es para secuenciales.
- Simplifica la paralelización de código de álgebra lineal, especialmente si se ha desarrollado sobre BLAS. Código parecido al de LAPACK.
- Claridad: el código es más corto y fácil de leer.
- Modularidad, al programarse usando bloques de programas.
- Portabilidad: las dependencias de la máquina se confían a PBLAS.

PBLAS

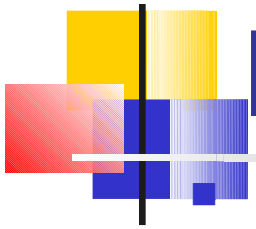
- Los mismos niveles que BLAS.
  - Nivel 1: Operaciones vector-vector. O(n)
  - Nivel 2: Operaciones matriz-vector. $O(n^2)$
  - Nivel 3: Operaciones matriz-matriz. $O(n^3)$
- No contiene rotaciones de vectores ni rutinas para matrices banda o empaquetadas.
- Contiene transposición de matrices.

PBLAS

- Nombre de las rutinas: PXYYZZZ (igual que BLAS con P delante)
  - En nivel 1: PDCOPY, PDDOT, ...
  - Transposición: P**X**TRAN**Y**
  - X, tipo de dato.
  - YY, tipo de matriz
    - GE, todos los operandos general rectangular
    - HE, un operando es Hermitiano
    - SY, un operando simétrico
    - TR, un operando triangular

PBLAS

- ZZZ, tipo de operación
  - MM, producto matriz matriz
  - MV, producto matriz vector
  - R, actualización de rango 1
  - R2, actualización de rango 2
  - RK, actualización de rango K de matriz simétrica o Hermítica
  - R2K, actualización de rango 2K de matriz simétrica o Hermítica
  - SM, sistema de ecuaciones múltiple
  - SV, sistema de ecuaciones simple



PBLAS

Distribución de las matrices

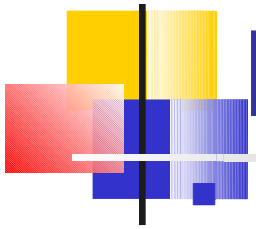
- Una matriz MxN se divide en bloques MBxNB
- Los bloques se asignan a los procesos con esquema cíclico por bloques bidimensional, para balancear la carga y conseguir escalabilidad

División de la matriz

A11 A12 A13 A14 A15
A21 A22 A23 A24 A25
A31 A32 A33 A34 A34
A41 A42 A43 A44 A45
A51 A52 A53 A54 A55

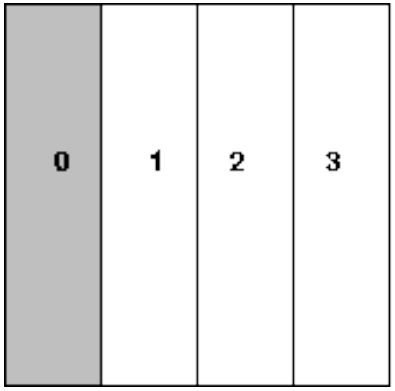
Asignación al sistema

	0	1
0	A11 A12 A15 A21 A22 A25 A51 A52 A55	A13 A14 A23 A24 A53 A54
1	A31 A32 A34 A41 A42 A45	A33 A34 A43 A44

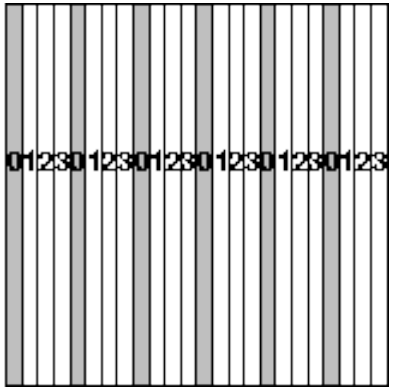


PBLAS

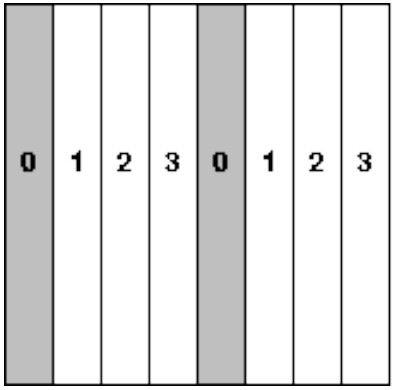
Distribuciones típicas



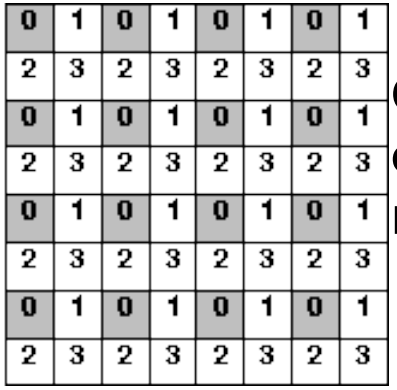
Column wise



Column wise cyclic



Column wise cyclic by blocks



Checkerboard cyclic by blocks: más escalable

PBLAS

- Se usa descriptor DESC para describir la distribución de la matriz. Se consigue ocultar el acceso a los índices. Es array de 9 enteros:
  - Descriptor de tipo de matriz, 1 para matrices densas
  - CTXT, malla de trabajo
  - M, filas de la matriz global
  - N, columnas de la matriz global
  - MB, tamaño de bloques por filas
  - NB, tamaño de bloques por columnas
  - RSRC, fila del proceso con primer dato de la matriz
  - CSRC, columna de proceso con primer dato
  - LLD, leading dimension (local), puede ser distinto en cada procesador

PBLAS

- El descriptor de arrays encapsula la información necesaria para describir una matriz distribuida.

- El descriptor se puede inicializar llamando a la rutina `descinit`

BLAS y PBLAS

- Reusabilidad de código:
  - DGEXXX(... , M , N , A(IA , JA) , LDA , ...)

  Se trabaja con matriz cuyo primer elemento es A(IA,JA) y que tiene leading dimension LDA (dentro de otra matriz de dimensión LDA)
  - PDGEXXX(... , M , N , A , IA , JA , DESCA , ...)

  Se trabaja con matriz A cuyas dimensiones y leading dimension locales se dan en DESCA. Con IA y JA se indica la posición del primer elemento en la matriz global (no hay que hacer cálculo de posiciones)

BLAS y PBLAS

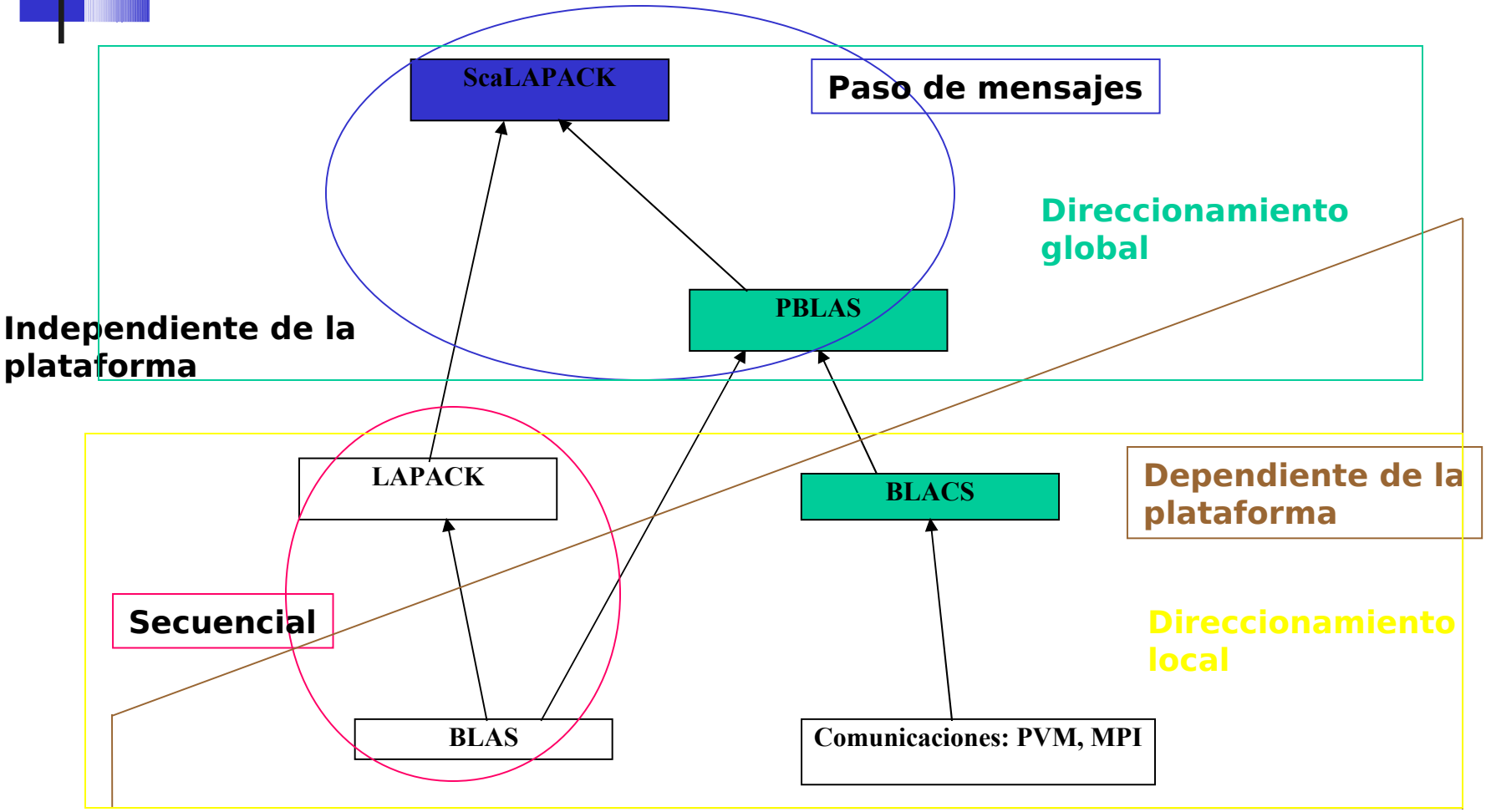
```
CALL DGEMM(
'No transpose', 'No transpose',
    transpose‘,
M-J-JB+1, N-J-JB+1, JB,
-ONE,
A( J+JB, J ),
LDA,
A( J, J+JB ),
LDA,
ONE,
A( J+JB, J+JB ),
LDA )
```

```
CALL PDGEMM(
'No transpose', 'No
M-J-JB+JA, N-J-JB+JA, JB,
-ONE,
A, J+JB, J,
DESCA,
A, J, J+JB,
DESCA,
ONE,
A, J+JB, J+JB,
DESCA )
```

PBLAS: Referencias

- Online quick reference guide: http://www.netlib.org/scalapack/html/pblas_qref.html

Librería ScaLAPACK (Scalable LAPACK)

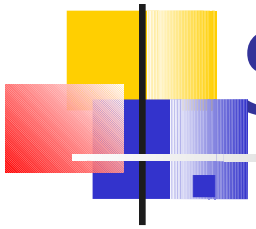


ScaLAPACK

- Finalidad: portar el paquete LAPACK a entornos de memoria distribuida.
- Buenas prestaciones: algoritmos por bloques, basados en operaciones de nivel 3. Usa librerías de menor nivel optimizadas.
- Escalabilidad: utiliza malla bidimensional.
- Portabilidad: basado en BLAS y BLACS.
- Usabilidad: llamadas similares a las de LAPACK.
- Modularidad: BLAS, BLACS, PBLAS.

ScaLAPACK

- Usa paradigma SPMD.
- Paso de mensajes para comunicación entre procesos.
- Sobre PVM y MPI.



ScaLAPACK

Distribución de las matrices como en BLACS

División de la matriz

A11 A12 A13 A14 A15
A21 A22 A23 A24 A25
A31 A32 A33 A34 A34
A41 A42 A43 A44 A45
A51 A52 A53 A54 A55

Asignación al sistema

	0	1
0	A11 A12 A15 A21 A22 A25 A51 A52 A55	A13 A14 A23 A24 A53 A54
1	A31 A32 A34 A41 A42 A45	A33 A34 A43 A44

- Balanceo de la carga ⇒buenas prestaciones y escalabilidad
- Necesarias rutinas de redistribución

ScaLAPACK

- Problemas que resuelve:
  - Como LAPACK pero algunos algoritmos menos
- Tipos de rutinas, como en LAPACK:
  - Driver routines
  - Computational routines
  - Auxiliary routines

ScaLAPACK

- Tipos de matrices, como LAPACK:
  - Densas.
  - Banda.
  - Reales y complejas.

    … no escasas
- Tipos de sistemas:
  - Paso de mensajes:
    - Multicomputadores de memoria distribuida
    - Máquinas de memoria virtual compartida (con paso de mensajes)
    - Redes de ordenadores

ScaLAPACK

- Formato de rutinas conductoras y computacionales: PXYYZZZ

X: Tipo de datos:

S : REAL

D : DOUBLE PRECISION

C : COMPLEX

Z : DOUBLE COMPLEX

YY: Tipo de matriz

ZZZ: Operación:

SV: sistemas de ecuaciones

EV: valores propios ...

ScaLAPACK

Tipos de matrices:

```
DB general band (diagonally dominant)
DT general tridiagonal (diagonally dominant)
GB general band
GE general (i.e., unsymmetric, in some cases rectangular)
GG general matrices, generalized problem
HE (complex) Hermitian
OR (real) orthogonal
PB symmetric or Hermitian positive definite band
PO symmetric or Hermitian positive definite
PT symmetric or Hermitian positive definite tridiagonal
ST (real) symmetric tridiagonal
SY symmetric
TR triangular (or in some cases quasi-triangular)
TZ trapezoidal
UN (complex) unitary
```

ScaLAPACK

- Ecuaciones lineales: AX = B
  - Rutina simple: PxyySV
  - Rutina experta: PxyySVX. Puede llevar a cabo otras funciones:
    - $A^TX=B$ o $A^HX=B$
    - Número de condición, singularidad, ...
    - Refina la solución y hace análisis de error.
    - Equilibrado del sistema.

ScaLAPACK

- Problema de mínimos cuadrados
  - Problema:

$$\underset{x}{\text{minimize}} \; \|b - Ax\|_2$$

  - Rutina: PxGELS

ScaLAPACK

- Problema simétrico de valores propios
  - Problema:

$$Az = \lambda z, \quad A = A^T, \text{ where } A \text{ is real.}$$

  - Rutinas:
    - Conductora simple: xyyEV
    - Conductora experta: xyyEVX

ScaLAPACK

- Descomposición en valores singulares
  - Rutinas: PSGESVD, PDGESVD
- Problema generalizado simétrico definido positivo de valores propios
  - Rutinas: PSSYGVX, PCHEGVX, PDSYGVX, PZHEGVX

ScaLAPACK

- Rutinas computacionales para sistemas lineales:
  - PxyyTRF, factorización LU
  - PxyyTRS, aplica la factorización para resolver sistema por sustitución progresiva o regresiva
  - PxyyTRI, aplica la factorización para obtener la inversa
  - PxyyCON, calcula el recíproco del número de condición
  - PxyyRFS, calcula cotas de error en la solución y la refina
  - PxyyEQU, calcula factores de escalado para equilibrar la matriz

ScaLAPACK

- Rutinas computacionales de factorización:
  - PxGEQRF
  - PxGELQF
  - PxGEQPF, QR con pivotaje
  - PxGEQLF
  - PxGERQF
  - PxGGQRF, QR generalizada
  - PxGGRQF, RQ generalizada

ScaLAPACK

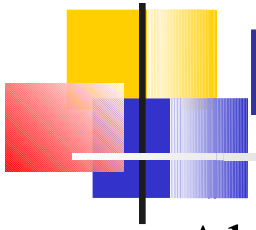
- Rutinas computacionales de problemas simétricos de valores propios:
  - PxSYTRD, reducción tridiagonal
  - PxORMTR, multiplicación de matrices tras reducción
  - xSTEQR2, nuevo algoritmo de rutina LAPACK, problema tridiagonal simétrico
  - PxSTEBZ, valores propios de tridiagonal
  - PxSTEIN, vectores propios de tridiagonal

ScaLAPACK

- Rutinas computacionales de problema no simétrico de valores propios:
  - PxGEHRD, reducción a Hessenberg
  - PxORMHR, multiplicación de matrices tras reducción
  - PxLAHQR, valores propios y forma de Schur

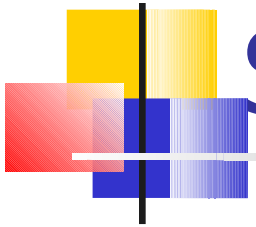
ScaLAPACK

- Rutinas computacionales de descomposición en valores singulares:
  - PxGEBRD, reducción a bidiagonal
  - PxORMBR, multiplicación de matrices tras reducción
- Rutinas computacionales de problema de valores propios simétrico generalizado:
  - PxSYGTS, reducción



LAPACK ≠ ScaLAPACK

- Algunos problemas que se resuelven en LAPACK no se resuelven en ScaLAPACK
- En algunos casos se usan algoritmos distintos
- Las rutinas de ScaLAPACK llevan argumentos para zonas de trabajo temporales
- En ScaLAPACK algunas rutinas tienen restricciones de alineamiento



ScaLAPACK: Práctica

- Hacer cambios en scala_01, 02 y 03 y comprobar su funcionamiento
- Comparar los tiempos de ejecución de la factorización LU en LAPACK y ScaLAPACK
- Comprobar la variación en el tiempo de ejecución de la factorización LU y QR al variar el número de procesadores, tamaño de problema y tamaño de bloque.

ScaLAPACK

- Para obtener buenas prestaciones en multicomputadores:
  - Use the right number of processors.
    - Rule of thumb: p=MxN/1000000 for a MxN matrix. This provides a local matrix of size approximately 1000 by 1000.
    - Do not try to solve a small problem on too many processors.
    - Do not exceed physical memory.
  - Use an efficient data distribution.
    - Block size (MB,NB=64)
    - Square processor grid
  - Use efficient machine-specific BLAS (not the Fortran 77 reference implementation BLAS) and BLACS

ScaLAPACK

- Para obtener buenas prestaciones en redes de ordenadores:
  - The bandwidth per node, in Mbytes/second/node, should be no less than one tenth of the peak floating-point rate, in Mflops/second/node.
  - The underlying network must allow simultaneous messages (not standard ethernet)
  - Message latency should be no more than 500 microseconds.
  - All processors should be similar in architecture and performance. ScaLAPACK will be limited by the slowest processor. Data format conversion significantly reduces communication performance.
  - No other jobs should be allowed to execute on the processors that are being used. If the processors are gang scheduled and there is enough physical memory for all jobs on all processors, this requirement may be relaxed, but we do not recommend doing so without careful study.
  - No more than one process should be executed per processor.

ScaLAPACK

- Para mejorar las prestaciones:
  - Use the best BLAS and BLACS libraries available.
  - Start with a standard data distribution.
    - A square processor grid (r=c=sqrt(p)) if p>9
    - A one dimensional processor grid (r=1,c=p) if p<9
    - Block size = 64
- Determine whether reasonable performance is being achieved.
- Identify the performance bottleneck(s), if any,
- Tune the distribution or routine parameters to improve performance further.

ScaLAPACK: Proposal 2004

- Incluir más rutinas de LAPACK
- Incluir mejores algoritmos numéricos
- Extender su funcionalidad
- Mejorar la facilidad de uso
- Tuning de prestaciones:
  - Autotuning tipo ATLAS. ILANEV devuelve un tamaño de bloque según tamaño de problema y algoritmo. No se ha analizado su variación.
  - En heterogéneos
  - Cuando la carga es variable
- Mejorar fiabilidad y soporte

ScaLAPACK: Referencias

- Página principal: http://www.netlib.org/scalapack/scalapack_home.html

- User's guide v1.7: http://www.netlib.org/scalapack/slug/index.html

Agradecimientos y material utilizado

- Ferran Mas, trasparencias de la exposición de ScaLAPACK del 2003/2004, ejemplos
- Javier Cuenca y Luis Pedro García, algunos códigos de ejemplo y trasparencias
- Tutorial de ScaLAPACK (Jack Dongarra y Antoine Petitet) en PARA 1995.
- CD con ScaLAPACK Users' Guide, editado por SIAM en 1997
- Demmel, Dongarra: ScaLAPACK proposal 2004
- Material en la web