A Proposal for a Set of Parallel Basic Linear Algebra Subprograms

(PBLAS)

Jaeyoung Choi
Soongsil University, South Korea

Jack J. Dongarra
University of Tennessee, Knoxville, and
Oak Ridge National Laboratory, USA

Susan Ostrouchov
University of Tennessee, Knoxville, USA

Antoine Petitet
University of Tennessee, Knoxville, USA

David Walker
Oak Ridge National Laboratory, USA

R. Clint Whaley
University of Tennessee, Knoxville, USA

`http://www.netlib.org/scalapack`

presented at the BLAS Workshop, Knoxville, TN, USA
November 13, 1995

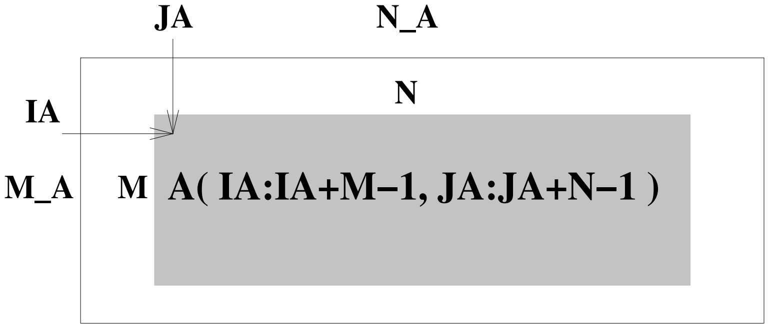
PBLAS – MOTIVATIONS

Parallel Basic Linear Algebra Subprograms
for distributed memory MIMD computers.

- **Similar functionality** as the BLAS: distributed vector-vector, matrix-vector and matrix-matrix operations,
- **Software re-use** of dense linear algebra codes: especially when implemented on top of the BLAS,
- **Clarity:** code is shorter and easier to read,
- **Modularity:** gives programmer larger building blocks,
- **Program portability:** machine dependencies are confined to the PBLAS (BLAS and BLACS).

PBLAS – OPERAND SPECIFICATIONS

- **MATRICES (Input/Output):** Ideally:



  More realistically: `M, N, A, IA, JA, DESCA`

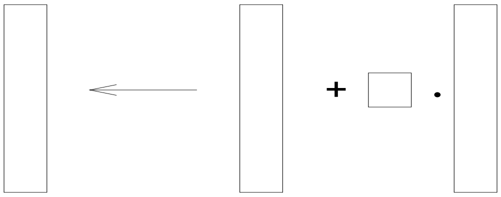
- **VECTORS (Input/Output):** particular matrices, re-use the BLAS increment to distinguish a row from a column vector:

  `N, X, IX, JX, DESCX, INCX`

- **SCALARS:**
  - **(Input):** SPMD $\Rightarrow$ replicated value
  - **(Output):** Only in Level-1, for efficiency:
    - Avoid synchronization, interruption of computation communication flow,
    - Correct only in vector operand scope.

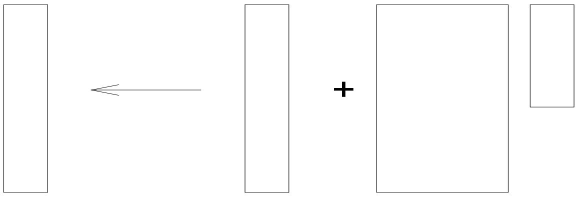
PBLAS – OPERATION SEMANTICS

Level 1 - Vector-vector operations



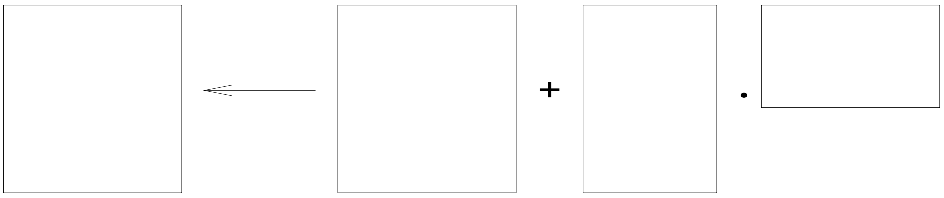
$$y \leftarrow y + \alpha x,\ dot \leftarrow x^T y,\ x \leftarrow \alpha x,\ \ldots$$

Level 2 - Matrix-vector operations



$y \leftarrow y + Ax$, also triangular solve, rank-1 update

Level 3 - Matrix-matrix operations



$C \leftarrow C + AB$, block triangular solve, rank-k update

- No vector rotations, no dedicated subprograms for banded or packed matrices,
- Matrix transposition available.

PBLAS – NAMING CONVENTIONS

The PBLAS naming scheme follows the conventions of the BLAS, with necessary adaptations:

P□XXYYY

□ (Data type)	XX (Type of matrix operands)
S: REAL,	GE: All matrix operands are GEneral rectangular,
D: DOUBLE PRECISION,	HE: One of the matrix operands is HErmitian,
C: COMPLEX,	SY: One of the matrix operands is SYmmetric,
Z: DOUBLE COMPLEX.	TR: One of the matrix operands is TRiangular.

- The Level 1 routines use meaningful abbreviations, for example PSCOPY, PSDOT ...
- The matrix transposition routine is called P□TRAN□.

YYY	(Operation)
MM	Matrix-matrix product;
MV	Matrix-vector product;
R	Rank-1 update of a matrix;
R2	Rank-2 update of a matrix;
RK	Rank-k update of a symmetric or Hermitian matrix;
R2K	Rank-$2k$ update of a symmetric or Hermitian matrix;
SM	Solves a system of linear equations for a matrix of right-hand sides;
SV	Solves a system of linear equations for a right-hand side vector.

PBLAS – ARGUMENT CONVENTIONS

- **Code reusability**, interface very close to BLAS:

```
 CALL  DGEMM( 'No Transpose', 'No 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 )
```

$$\downarrow$$

```
 CALL PDGEMM( 'No Transpose', 'No Transpose',
$    M-J-JB+1, N-J-JB+1, JB, -ONE, A, J+JB,
$    J, DESCA, A, J, J+JB, DESCA, ONE, A,
$    J+JB, J+JB, DESCA )
```

- In the PBLAS, the increment specified for vectors is always global. So far only `INCX=1` and `INCX=DESCX(1)` are supported.

BLAS	PBLAS
`INTEGER LDA` `INTEGER INCX`	`INTEGER IA, JA, DESCA( 8 )` `INTEGER INCX, IX, JX, DESCX( 8 )`
`A, LDA` `X, INCX`	`A, IA, JA, DESCA` `X, IX, JX, DESCX, INCX`

- Level 1 BLAS functions have become PBLAS subroutines. Output scalar correct in operand scope.

PBLAS – STORAGE CONVENTIONS

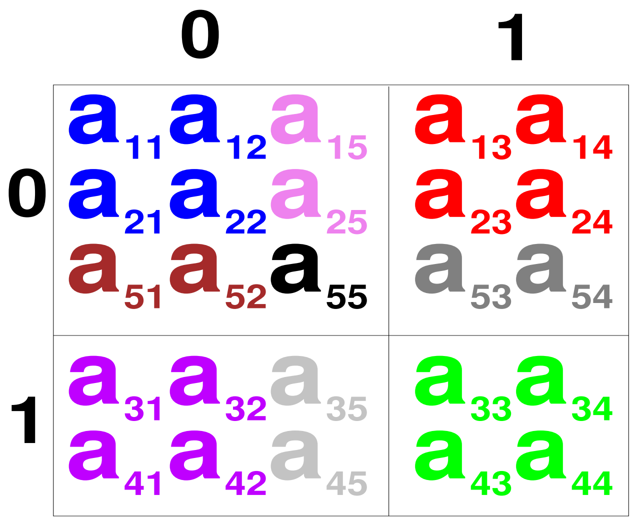
- An `M_`-by-`N_` **matrix is block partitioned** and these `MB_`-by-`NB_` blocks are distributed according to the

  2-dimensional block-cyclic scheme
  $\Rightarrow$ **load balanced computations, scalability**

- Locally, the scattered **columns are stored contiguously** (FORTRAN "Column-major")

  $\Rightarrow$ **re-use of the BLAS** (leading dimension `LLD_`).

5 x 5 matrix partitioned in 2 x 2 blocks



2 x 2 process grid point of view

Descriptor `DESC_`**:** 8-Integer array describing the matrix layout, containing `M_`, `N_`, `MB_`, `NB_`, `RSRC_`, `CSRC_`, `CTXT_` and `LLD_`, where (`RSRC_`, `CSRC_`) are the coordinates of the process owning the first matrix entry in the grid specified by `CTXT_`.

Ex: `M_=N_=5`, `MB_=NB_=5`, `RSRC_=CSRC_= 0`, `LLD_` $\geq$ `3` (in process row 0), and `2` (in process row 1).

PBLAS – RATIONALE

- **Software reusability**: BLAS and PBLAS interfaces similar,
- Level 1 PBLAS: Output scalar is only correct in operand scope,
- Level 3 PBLAS uses **“shaped adaptive” procedure** to perform the computations,
- The PBLAS routines follow an **SPMD** programming model (every process in the grid must call the routine).
- The BLACS context provides the ability to have separate “universes” of message passing:

  Safe co-existence with other parallel libraries,
- The PBLAS do not perform “inter-context” operations.

PBLAS – REFERENCES

- PBLAS software and documentation can be obtained via:
  - WWW: `http://www.netlib.org/scalapack`,
  - (anonymous) ftp `ftp.netlib.org`:

    `cd scalapack; get index`
  - email `netlib@www.netlib.org` with the message:

    `send index from scalapack`
- Comments and questions can be addressed to

  `scalapack@cs.utk.edu`

- J. Choi, J. Dongarra, S. Ostrouchov, A. Petitet, D. Walker and R. C. Whaley, *A Proposal for a Set of Parallel Basic Linear Algebra Subprograms*, Technical Report UT CS-95-292, LAPACK Working Note #100, University of Tennessee, 1995.