An overview of the BLACS

R. Clint Whaley
University of Tennessee
rwhaley@cs.utk.edu
http://www.cs.utk.edu/
~rwhaley/Index.html

Basic Linear Algebra Communication Subprograms

A linear algebra oriented message passing layer designed for use in ScaLAPACK.

Motivation

- Encapsulate platform dependent code to small communication kernel
  - Keep small for ease of use and porting
  - NOTE: MPI not around at this time (more on this later)
- Have level of symbolism similar to computation kernel, the BLAS
- Have efficient port of ScaLAPACK synonymous with efficient platform-specific BLAS and BLACS

I. FEATURES OF THE BLACS

1. Standard interface across parallel platforms
   - Available on: CMMD (CM-5), NX (Intel family), MPL (IBM SP series), PVM, and MPI.
2. 2D array-based communication
   - General Rectangular arrays
   - Trapezoidal arrays
3. ID-less communication
4. 2-D grid/mesh of processes
   - SCOPE : 'Row', 'Column', or 'All'
5. Context (corresponding to MPI's communicator) allows the user to:
   - create arbitrary groups of processes,
   - create multiple overlapping and/or disjoint grids,
   - isolate each process grid so that grids do not interfere with each other.

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

Figure 1: 8 processes mapped to a 2 x 4 process grid.

II. FOUR TYPES OF ROUTINES IN THE BLACS

1. Support Routines
2. Point to Point (send and receive)
3. Broadcast (along row, column or all of grid)
4. Combines (along row, column or all of grid)
   - element-wise summation
   - element-wise maximization
   - element-wise minimization

IIa. SUPPORT ROUTINES

There are a number of routines which do not deal directly with communication that are nonetheless required for programming in a parallel environment. The BLACS label these routines as *support* routines. Examples include routines to form contexts (`BLACS_GRIDINIT` and `BLACS_GRIDMAP`), routines to return information about a grid (`BLACS_GRIDINFO`), etc. All standard support routines are prefixed by `BLACS_`.

IIb. COMMUNICATION ROUTINES

Point to point Send and Receive

Send (sub)matrix from one process to another:

```
□xxSD2D( ICONTXT, [UPLO, DIAG,] M, N, A, LDA, RDEST, CDEST )
□xxRV2D( ICONTXT, [UPLO, DIAG,] M, N, A, LDA, RSRC, CSRC )
```

Broadcast/send Broadcast/recv

Send (sub)matrix to all processes or subsection of processes in `SCOPE`, using various distribution patterns (`TOP`):

```
□xxBS2D( ICONTXT, SCOPE, TOP, [UPLO, DIAG,] M, N, A, LDA )

□xxBR2D( ICONTXT, SCOPE, TOP, [UPLO, DIAG,] M, N, A, LDA,
         RSRC, CSRC )
```

SCOPE	TOP
'Row'	' ' (default)
'Column'	'Increasing Ring'
'All'	'1-tree' ...

□ (Data type)	xx (Matrix type)
I: INTEGER, S: REAL, D: DOUBLE PRECISION, C: COMPLEX, Z: DOUBLE COMPLEX.	GE: GEneral rectangular matrix, TR: TRapezoidal matrix.

IIc. COMBINE OPERATIONS

1. Global combine operations

Perform element-wise SUM, |MAX|, |MIN| operations, on rectangular matrices:

```
□GSUM2D( ICONTXT, SCOPE, TOP, M, N, A, LDA,
         RDEST, CDEST )

□GAMX2D( ICONTXT, SCOPE, TOP, M, N, A, LDA,
         RA, CA, RCFLAG, RDEST, CDEST )

□GAMN2D( ICONTXT, SCOPE, TOP, M, N, A, LDA,
         RA, CA, RCFLAG, RDEST, CDEST )
```

- `RDEST = -1` indicates that the result of the operation should be left on all processes selected by `SCOPE`,
- For |MAX|, |MIN|, when `RCFLAG = -1`, `RA` and `CA` are not referenced; otherwise, these arrays are set on output with the coordinates of the process owning the corresponding maximum (resp. minimum) element in absolute value of `A`. Both arrays must be at least of size $M \times N$ and their leading dimension is specified by `RCFLAG`.

III. Why use BLACS instead of direct MPI calls

- BLACS provide correct level of symbolism for ScaLAPACK
  - Direct MPI calls add many more lines of code
  - Would wind up with wrappers anyway
- MPIBLACS may be viewed as ScaLAPACK's set of wrappers around MPI
  - Get roughly same performance, because MPIBLACS simply call the appropriate MPI routines
  - Can get a MPI communicator from a BLACS context – allows user to switch to direct MPI calls if needed

BLACS call:

```
CALL DGERV2D( ICONTXT, M, N, A, LDA, RSRC, CSRC )
```

Direct MPI calls:

```
CALL MPI_TYPE_VECTOR( N, M, LDA, MPI_DOUBLE_PRECISION,
                      GEMAT, IERR )
CALL MPI_TYPE_COMMIT( GEMAT, IERR )
CALL MPI_RECV( A, 1, GEMAT, MAP1D(RSRC,CSRC), MSGID,
               COMM, STAT, IERR )
CALL MPI_TYPE_FREE( GEMAT, IERR )
```

IV. OUTSTANDING ISSUES IN THE MPIBLACS

- Correct BLACS/MPI interface
- Globally blocking context formation
  - MPI blocks on entire group used to form a sub-group
  - Until now, BLACS blocked only on sub-group
- Inter-language operations
  1. MPI_INIT
     - C version takes argc and argv: not generally accessible from library level
     - Interface of MPI_INIT must match language of main program
     - No standard ruling on how MPI_INIT effects different interfaces
  2. No standard way to translate communicators from one interface to another
     - At present, use several workarounds
     - MPI committee examining this issue
  3. Integer communicator handles
     - C interface to MPI may have non-integer context handles
     - BLACS assume integer system handles
     - Presently, we have simple mapping routines

V. DOWNLOADING THE PACKAGE

- BLACS software and documentation can be obtained via:
  - WWW: `http://www.netlib.org/blacs`,
  - (anonymous) ftp `ftp.netlib.org`: `cd blacs; get index`
  - email `netlib@www.netlib.org` with the message:

    `send index from blacs`

- Comments and questions can be addressed to

  `blacs@cs.utk.edu`

VI. PAPERS OF INTEREST

BLACS

- "A User's Guide to the BLACS v1.0".
  `http://www.netlib.org/blacs/lawn94.ps`
- "Installing and Testing the BLACS".
  `http://www.netlib.org/blacs/blacs_install.ps`

BLACS and MPI (drafts)

- "Outstanding Issues in the MPIBLACS".
  `http://www.netlib.org/blacs/mpiblacs_issues.ps`
- "Some Plebian Extensions to MPI".
  `http://www.netlib.org/blacs/mpi_prop.ps`
- "Using BLACS and MPI in ScaLAPACK".
  `http://www.cs.utk.edu/~rwhaley/BLACS/DOWNLOAD/BlacsMpiInSL.ps`