Gabriel: a modern Fortran-library for simple, fast and verified exchange of halo regions

John Donners (john.donners@surfsara.nl)

ABSTRACT

The Fortran language and the MPI standard are used by a large portion of computational fluid dynamics and climate models. Fortran has recently been expanded with features like object-oriented programming and assumed-rank arrays, while the MPI library received a major overhaul in MPI-3.0 with features like neighbor communicators and more derived types. Both developments can have a great impact in terms of programmability and performance of earth system models.

The Gabriel library combines these recent developments into a library that improves the use of MPI for applications written (largely) in Fortran. Gabriel offers a high performance, verification and ease of use, especially for models on regular grids.

The library is released into the public domain and can be found at
https://github.com/jdonners/gabriel

Use

Just imagine that we would like to use some finite difference scheme to solve a differential equation. We choose a regular 3x3 2D decomposition, where rank 4 has 8 neighboring ranks.

```
.-------.-------.-------.
|       |       |       |
|   0   |   1   |   2   |
|      X|XXXXXXX|X      |
.-------.-------.-------.
|      X|OOOOOOO|X      |
|   3  X|OOO4OOO|X  5   |
|      X|OOOOOOO|X      |
.-------.-------.-------.
|      X|XXXXXXX|X      |
|   6   |   7   |   8   |
|       |       |       |
.-------.-------.-------.
```

```
[0-8] - rank
O - local domain for rank 4
X - halo regions for rank 4
```

Rank 4 calculates the central part of the array (indicated by 'O'), while the values at the edges (indicated by'X'), called halos, are calculated at other ranks and need to be communicated.
Gabriel sets up all communication to update the halo regions of a variable in a single MPI exchange.

Advantages

- use the actual indices of your arrays in subroutine calls.
- arrays of any dimension and rank are supported,
- the elements located on diagonal ranks are communicated.
- periodic boundary conditions are supported (e.g. rank 2 and 6 exchange data for periodicity in both directions)
- decomposition can be irregular as well.
- missing domains are not a problem (e.g. land regions in an ocean model).

Requirements

1. a recent Fortran compiler supporting Fortran 2003 and TS 29113:
   Intel 16.0.0 or newer
   GNU gfortran 4.9 or newer
   recent Cray Fortran
   recent IBM XL Fortran, can anyone test this?

2. a recent MPI library supporting the MPI-3.0 standard:
   Intel MPI 5.0.0 or newer
   MPICH2 3.0.4 or newer (and maybe older)
   recent Cray MPI
   recent IBM POE, can anyone test this?

Code example

The routine autocreate sets up all communication to update all halo regions of a variable. Note that this includes the elements located on diagonal ranks. The gabriel library support arrays of any dimension and rank.

The input arguments are the variable to be updated, the lower and upper indices of the array that are calculated locally and the communicator of all processes that will calculate part of the full domain. E.g. for the simple case of a 1D array, it would be something like:

```
MPI rank                 0          1
Indices of array A   0123456    3456789
Locally computed     .....        .....
Halo region               ..    ..
```

So rank 0 calculates indices 0-4 and rank 1 calculates indices 5-9. Both ranks need some data from the neighbor in the halo region to be able to calculate its local indices. This would look something like:

```
program one_two
  use mpi
  use gabriel

  real,dimension(:),allocatable :: a
  type(decomposition) :: dec
  integer ierr,rank

  call MPI_Init(ierr)
  call MPI_Comm_rank(MPI_COMM_WORLD,rank,ierr)

  if (rank.eq.0) then
    allocate(a(0:6))
    a=0.0
    call dec%autocreate(a,(/0/),(/4),MPI_COMM_WORLD)
  endif
  if (rank.eq.1) then
    allocate(a(3:9))
    a=1.0
    call dec%autocreate(a,(/5/),(/9),MPI_COMM_WORLD)
  endif
  call dec%update(a,a)
  write(*,'(a,i2,a,5f5.1)')'Rank=',rank,'   My
values:=',a
end program
```

and the output of the program will look like:

```
Rank= 0 My values:=  0.0  0.0  0.0  0.0 0.0  1.0 1.0
Rank= 1 My values:=  0.0  0.0  1.0  1.0 1.0  1.0 1.0
```

And this works analogously for multi-dimensional arrays. The routine has an optional argument 'offset' to set the offset for each rank. E.g. if your arrays are defined to start at index 1 on each process, the offset will be added to the indices of the array to determine its position in the global domain across all processes. The optional argument 'periodic' can be used to indicate periodic boundary conditions with a simple boolean.

CONCLUSIONS

Advanced MPI-3.0 features can give great performance and memory benefits. The Gabriel library hides the complexity of MPI-3.0, without losing its benefits.

Gabriel also uses Fortran features of the latest TS29113 specification. You can simply use the array indices as you defined them in calls to Gabriel.

This allows you to spend more time on the global view of your model.

E INFO@SURFSARA.NL
T 0031 800 1300
W SURFSARA.NL

SURFSARA
PO BOX 94613
1090 GP AMSTERDAM
THE NETHERLANDS

SURF is the collaborative ICT organisation for Dutch research and higher education. SURF offers scientists, students and lecturers in the Netherlands access to the best possible network, data storage and compute resources and technical advice and support.

More information about SURF: WWW.SURF.NL

E INFO@SURF.NL
T 0031 88 78 73 000

SURF
PO BOX 19035
3501 DA UTRECHT
THE NETHERLANDS