HLST Report: POMS project

T.M. Tran [1] A. Ratnani [2]

[1]SPC/SB-EPFL

[2]IPP/Garching

11th October 2017

SWISS PLASMA CENTER

Outline

- POMS: *Parallel and Optimal Multigrid for B-Splines*
- The 2D parallelization of the MG solver: problems and solutions:
  1. Partition of *vectors* for the solution and RHS.
  2. Partition of the *matrix* for linear operators.
  3. Partition of MPI processes using the 2D MPI Cartesian topology.
  4. Ghost cell exchanges.
  5. Getting *global* solution, frequently needed for applications and to avoid fully parallelize all complex procedures.
  6. Using the parallel MUMPS direct solver for the first fine grid (*code verification*) and the last coarse grid (as *direct solver*),
- Conclusions

The 2D local vectors

- The $N_i + p_i$ Splines on each dimension are partitioned by defining its starting and ending index ($s_i, e_i,\ i = 1, 2$).
- The size of the local 2D array holding the Splines coefficients should be *augmented* with *ghost cells* of widths g_i at both ends of the vectors and can be implemented using the following Fortran **`derived type`**

```
TYPE gvector_2d
INTEGER, DIMENSION(2)     :: s, e, g
REAL(rkind), ALLOCATABLE :: val(:,:)
END TYPE gvector_2d
```

- For matrix-vector product, $g_i = p_i$ is sufficient.
- At some grid levels, g_i should be increased due to the structure of the inter-grid operators!
- The local vectors are *not contiguous*: use the **`MPI_Type_vector`** for the send buffer type.

SWISS PLASMA CENTER
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

The local matrix structure

- The *quadratic bilinear operator* array $A_{i_1,i_2,i_1',i_2'} = A_{i_1,i_2,i_1'-i_1,i_2'-i_2}$ can be implemented as a Fortran *compact* **`derived type`** (here called *stencil data structure*) as

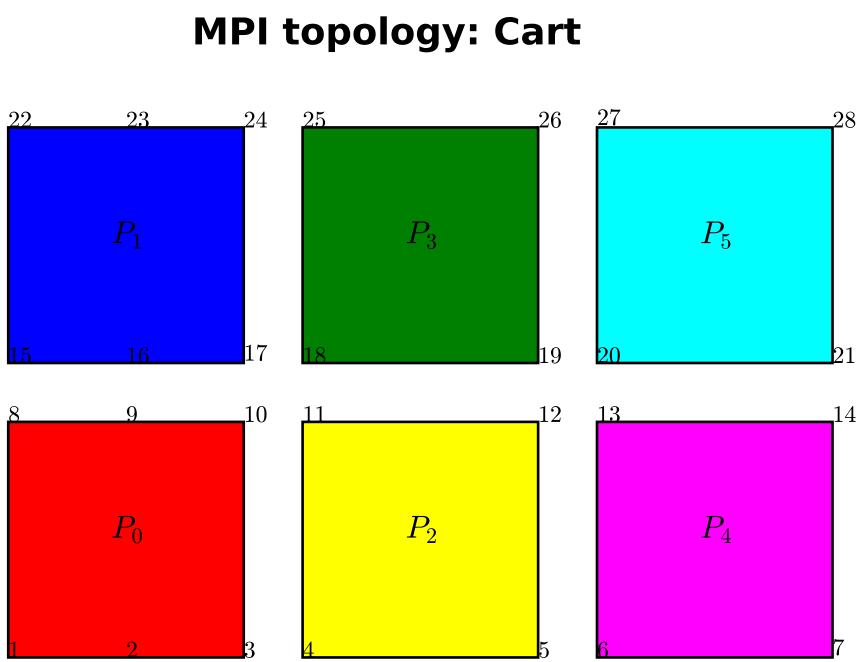
```
TYPE stencil_2d
   INTEGER, DIMENSION(2) :: ldim, gdim
   INTEGER, DIMENSION(2) :: s, e, nd
   REAL(rkind), ALLOCATABLE :: val(:,:,:,:)
END TYPE stencil_2d
```

- Here, **`nd`** are just equal to the Splines orders p_i.
- The *local* matrix-vector product is thus defined simply by

$$v_{i_1,i_2} = \sum_{d_1=-nd_1}^{nd_1} \sum_{d_2=-nd_2}^{nd_2} A_{i_1,i_2,d_1,d_2}\, u_{i_1+d_1,i_2+d_2}, \quad s_1 \le i_1 \le e_1,\ s_2 \le i_2 \le e_2,$$

The 2D MPI Cartesian Topology

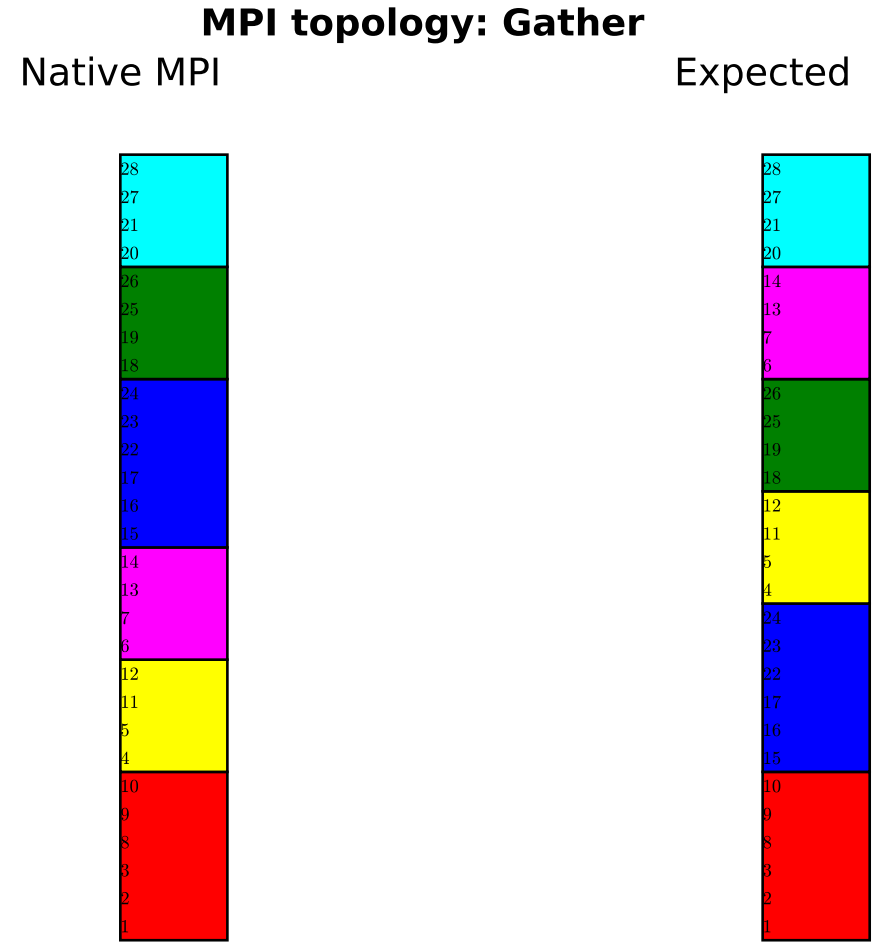
- Help to improve the *scalability of ghost cell exchanges*
- Problem of MPI standard `C` ordering of the MPI ranks: example of a 3×2 process grid, with a 7×4 Splines:



SWISS PLASMA CENTER

Results of `MPI_Allgatherv`

The results are *out of order* for a 7×4 Splines 1D numbering



SWISS PLASMA CENTER

Ghost cell exchanges

- Exchange along one dimension should be *completed before* the exchange on the next dimension to include contributions of the *corner* processes
- For coarse grid levels with small local numbers of local *larger order* Splines, update of ghost cells from *remote neighbor* processes might be needed: the exchange becomes thus *non-concurrent*.

Local to global transformation of vectors

- Facilitate the mapping between the 1D data partition used by MUMPS and the 2D partition of the MG solver.
- Applications often require the *global* solution (and RHS) of the solver.
- Avoid to fully parallelize all (especially some complex or not used frequently) procedures such as the computation of the Spline expansion or the L_2 norms.
- Methodology: use the **`MPI_Allgatherv`** using the non-contiguous **`MPI_Type_vector`**, followed by a *permutation*.
- The inverse *global to local* operation is straightforwards:

$$u^{\text{loc}}(s_1-g_1 : e_1+g_1, s_2-g_2 : e_2+g_2) = u^{\text{glob}}(s_1-g_1 : e_1+g_1, s_2-g_2 : e_2+g_2)$$

SWISS PLASMA CENTER
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Conclusions

- List of implemented and tested components:
  - Ghost cell exchange.
  - Parallel MUMPS at the first and last grid level.
  - Local matrix-vector product.
  - Computation of residues.
  - Inter-grid prolongation and restriction.
  - Parallel Jacobi relaxation.
- Missing part: The *Preconditioned Conjugate Gradient* required for the GLT *post-smoother*.
- Close to the *fully* assembled 2D parallel MG solver!