MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University

Lecture 41.5:

Parallelization on a cluster of distributed memory machines

Part 3: Distributed computing in deal.II

deal.II with MPI

Examples and documentation can be found here:

- Step-40 (the canonical example)
- Step-32
- Step-42
- Step-48
- Older styles: step-17, step-18

- Documentation module on “Parallel computing with multiple processors using distributed memory”

- Bangerth, Burstedde, Heister, Kronbichler: “Algorithms and data structures for massively parallel generic finite element codes”, ACM TOMS, 2011.

deal.II with MPI

Philosophy:

- *Global objects* require *O(N)* memory (*N*=# of cells)
- *Every* global data structure needs to be distributed:
  - Triangulation
  - DoFHandler
  - Hanging node constraints
  - Matrix
  - Solution and right hand side vectors
  - Postprocessed data (DataOut)
- No processor may hold all data for a global object

- Processors hold *O(N/P)* "locally owned" data
- Processors may also hold *O(εN/P)* "ghost elements"

deal.II with MPI

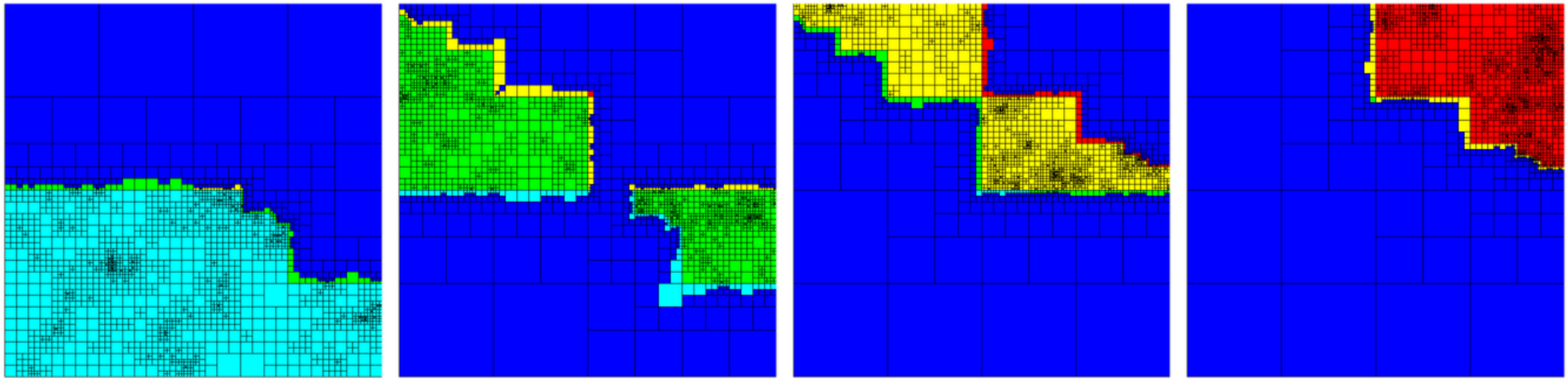
Philosophy:

- Every processor may only work on locally owned data (possibly using ghost data as necessary)
- deal.II carefully communicates data that may be necessary early on, tries to avoid further communication
- Use PETSc/Trilinos for linear algebra
- (Almost) No handwritten MPI necessary in user code

deal.II with MPI

Example:

- There is an “abstract”, global triangulation
- Each processor has a parallel::distributed::Triangulation object that stores “locally owned”, “ghost” and “artificial” cells (and that's all it knows):



P=0 *P=1* *P=2* *P=3*

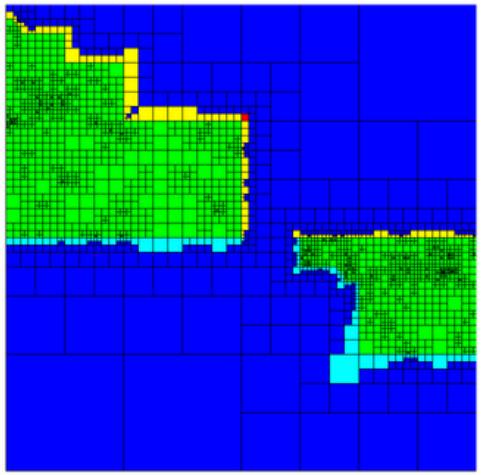
(magenta, green, yellow, red: cells owned by processors 0, 1, 2, 3; blue: artificial cells)

deal.II with MPI

Example:

- There is an “abstract”, global DoFHandler
- Each processor has a DoFHandler object built on it

- Each processor “owns” part of the DoFs on the cells it owns
- Knows global numbers of all DoFs on locally owned and ghost cells:
  - locally owned DoFs
  - locally active DoFs
  - locally relevant DoFs



step-40

Step-40:

- The parallel version of step-6
- Has been shown to scale well to 16k processors

Differences to step-6:

- Need to let
  – system matrix, vectors
  – hanging node constraints
  know about what is locally owned, locally relevant
- Need to restrict assembly to locally owned cells
- Need to create one output file per processor

- Everything else happens under the hood

step-40

Read through the commented program at

http://www.dealii.org/8.1.0/doxygen/deal.II/step_40.html

Then play with the program:

```
cd examples/step-40
cmake -DDEAL_II_DIR=/a/b/c . ; make
mpirun -np 4 ./step-40
```

This will run the program and generate output files:

```
ls -l
```

Then visualize the solutions.

MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University