Added technical aspects of p4est: Alternative quadrant representation and MPI-3 shared memory

Mikhail Kirilin, Carsten Burstedde
University of Bonn, Germany

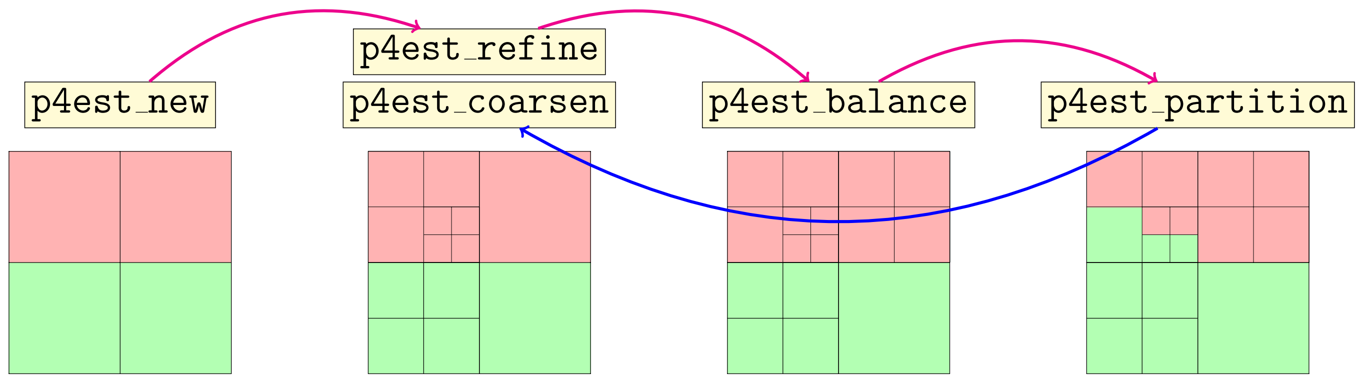
Deutscher Akademischer Austauschdienst
German Academic Exchange Service

BONN INTERNATIONAL GRADUATE SCHOOL OF MATHEMATICS

Introduction

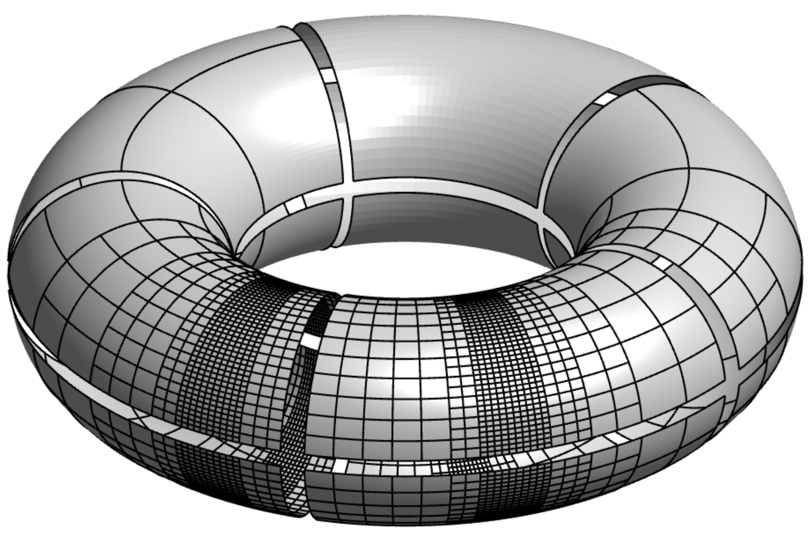
Many numerical simulations requires a mesh of computational cells covering the domain of interest. The solution is approximated by functions associated with a set of cells.

- Implementation: the **p4est** software library
- Dynamic management of adaptive octrees
- Discretization of a computational domain
- Efficiently manages large-scale parallel tasks

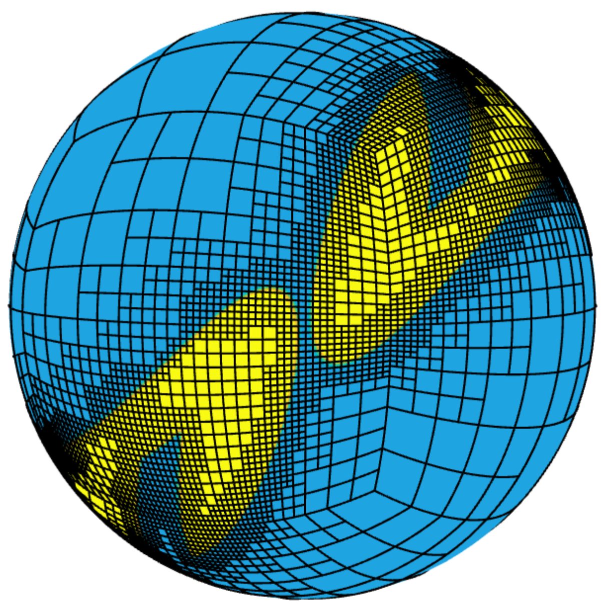


Typical workflow of the `p4est` software library.

The `p4est` library is actively used worldwide: linked e.g. by solver libraries deal.ii, PETSc, ForestClaw. Some possible applications: continuum mechanics and particle simulation.



A representation of a refined mesh built by `p4est` within ForestClaw[1] on a torus.



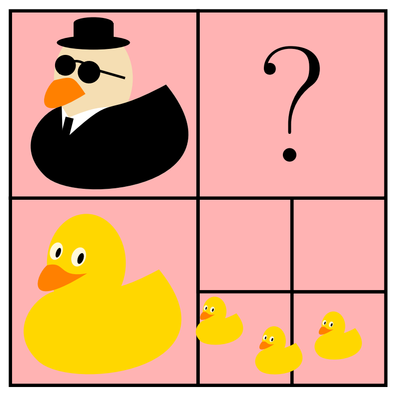
Modelled advection problem performed by ForestClaw[1] solver.

Alternative quadrant representations

Cells as quadrants

`p4est` represents cells with 2D squares (3D cubes) which we call **quadrants (octants)**. Their characterization:

- Defined by the coordinates of a corner and a level.
- It is allowed for them to be of various sizes.
- Store user's information depending on application.



Per-quadrant operations are listed in the original paper on `p4est` [1] and the source code.

```
*r = _mm_sub_epi32 (
        _mm_and_si128 (*q, _mm_set_epi32
                        (~QUAD_LEN (level)
                       , ~QUAD_LEN (level)
                       , ~QUAD_LEN (level)
                       , 0xFFFFFFFF))
   , _mm_set_epi32 (0, 0, 0, 1));
```

Implementation of `Parent` algorithm, constructing parent `r` of the 128-bit quadrant `q`. Written with use AVX/SSE.

Some quadrant properties:

- Form a disjoint union of all leaves in a forest.
- Partitioned btw. MPI processes by space filling curve (SFC) order.
- SFC is aka Morton or Z-curve [2, 3].
- Quadrants can be set by ℓ and either (x,y,z) or Morton index *id*.

Since an octant is defined by x, y, z and ℓ, we consider four-way **SIMD** (Single Instruction Multiple Data) for accelerated processing. We base new quadrant representation on the Advanced Vector Extensions/Streaming SIMD Extensions (**AVX/SSE**).

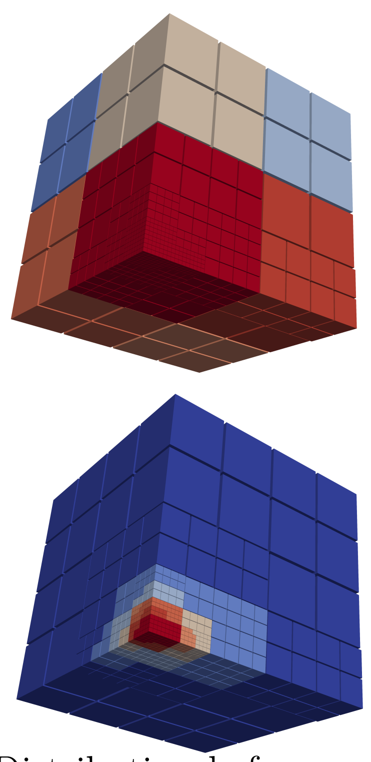
$$\begin{cases} x & = 0\; x_0\; x_1\; x_2 \ldots x_{30} \\ y & = 0\; y_0\; y_1\; y_2 \ldots y_{30} \\ z & = 0\; z_0\; z_1\; z_2 \ldots z_{30} \\ \text{level} & = 0 \ldots 0\; l_0\; l_1 \ldots l_7 \end{cases}$$

l_7	l_6	$\cdots$	0	z_{30}	$\cdots$	z_0	0	y_{30}	$\cdots$	y_0	0	x_{30}	$\cdots$	x_0	0

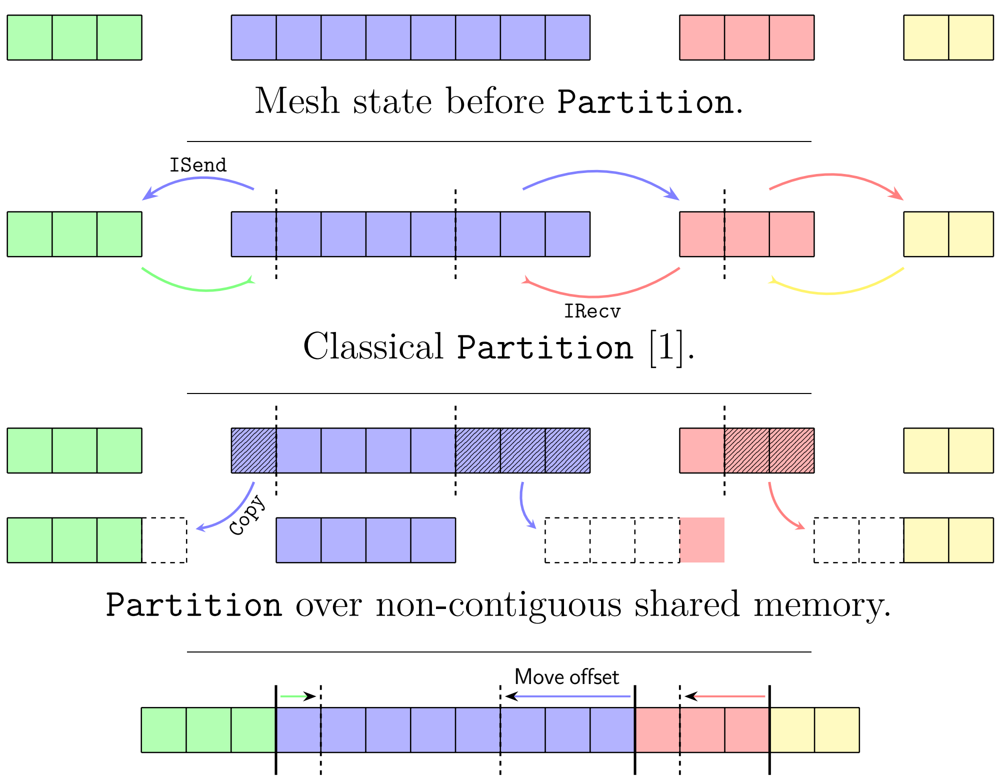
These intrinsics operate on extended processor registers. Specifically, we chose the special SSE2 type `__m128i`, that stores 128 bits of data interpreted as signed integers.

Shared memory algorithm: Partition

The `Partition` algorithm for redistributing work-load guarantees similar amounts of work for all computation nodes. Current version of the `Partition`, as published in [1] and [4], asynchronously sends and receives the data. We aim to eliminate the redundancy of the sends within each node with **shared memory (SM)** windows introduced in MPI 3.0.



Distribution before and after `Partition`.



Mesh state before `Partition`.

Classical `Partition` [1].

`Partition` over non-contiguous shared memory.

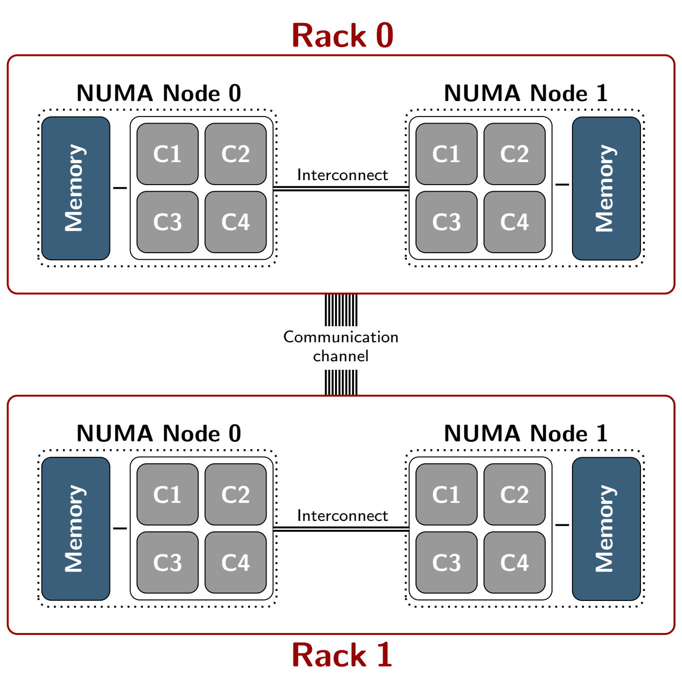
`Partition` over contiguous shared memory.

We use the following notation:

- N global and N_p local to process p number of quadrants in a forest.
- Element offsets O_p: $O_{p+1} - O_p = N_p$.

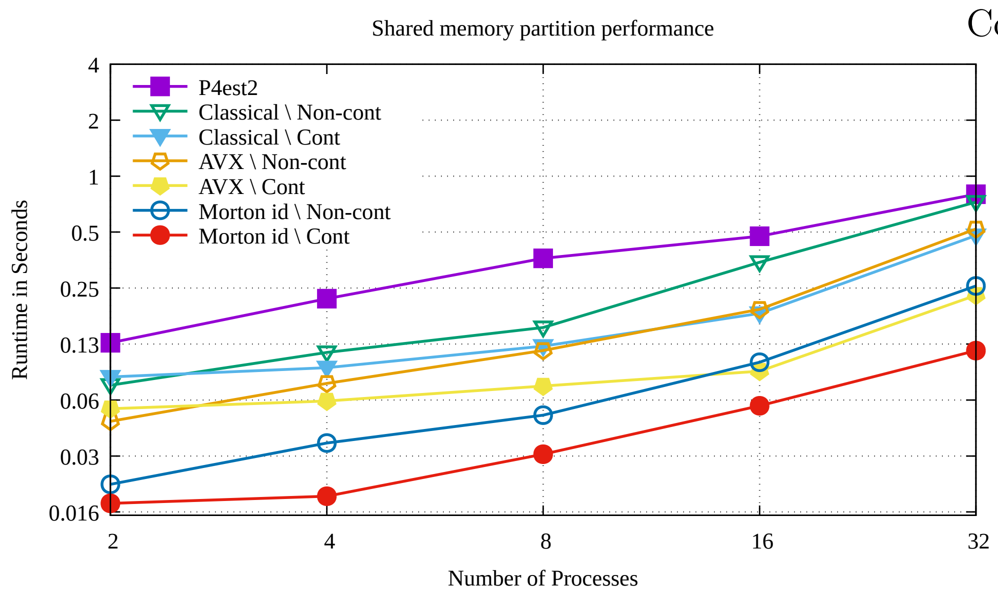
Highlights of approaches for `Partition`:

- In each case we calculate new offsets O'_p to define new boundaries on SFC.
- Classical `Partition` relies on asynchronous `ISend` and `IRecv` calls.
- With MPI-3 we allocate new SM and perform simply copying into it.
- Contiguous shared memory allows reassigning new element offsets O'_p.



Partition: numerical results

We run mesh partition test for various combinations of algorithm and quadrant implementations. The performance scalability was tested on a desktop PC with up to 36 physical cores split between two sockets. We modeled an unbalanced mesh with approximately 3 million quadrants per core shipping 80% of them, which is equal to sending 2.5 GB of data.



Conclusions and highlights:

- Quadrants with Morton index *id* demonstrate the fastest results over others.
- BUT contribute to changes of quadrant bit operations.
- Contiguous shared memory shows better performance over non-contigious.
- We consider the pair of AVX/contiguous as a major user's selection.

References

[1] Carsten Burstedde, Lucas C. Wilcox, and Omar Ghattas. p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011.

[2] G. M. Morton. A computer oriented geodetic data base; and a new technique in file sequencing. Technical report, IBM Ltd., 1966.

[3] Herbert Tropf and H. Herzog. Multidimensional range search in dynamically balanced trees. *Angewandte Informatik*, 2:71–77, 1981.

[4] Hari Sundar, George Biros, Carsten Burstedde, Johann Rudi, Omar Ghattas, and Georg Stadler. Parallel geometric-algebraic multigrid on unstructured forests of octrees. In *SC12: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2012.

Mikhail Kirilin
Doctoral Student in Mathematics at University Bonn

2023 University of Bonn, Hausdorff Center for Mathematics, Institute for Numerical Simulation.
This work is supported by a scholarship of the German Academic Exchange Service (DAAD).
kirilin@ins.uni-bonn.de

[1] Pictures courtesy Donna Calhoun, Boise State University.