Loadbalancing in a Parallel Adaptive Grid Generator

Jörn Behrens
Munich University of Technology, Germany

Jens Zimmermann
Ludw.-Max.-University Munich, Germany

1. Introduction

The simulation of large multi-scaled problems (like ocean and atmosphere circulation) requires parallel and adaptive methods. Parallel adaptive grid control needs dynamic load balancing. Appropriate data structures supporting communication of grid items have to be found.

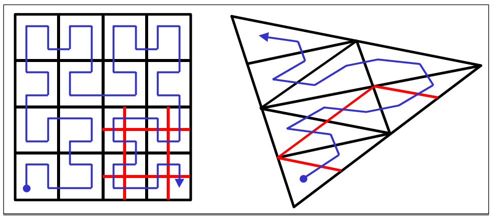

Fig. 1: Adaption of a space-filling Hilbert curve to triangular grid

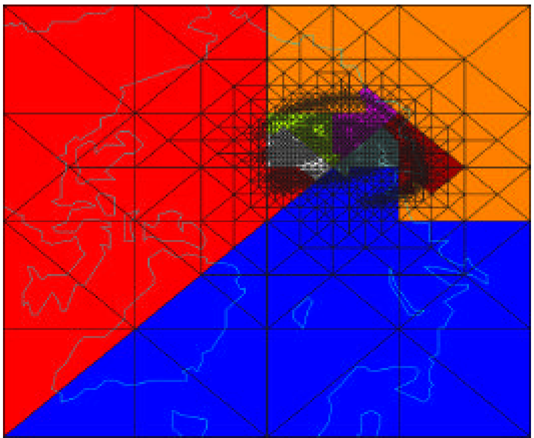

Fig. 2: Adaptively refined grid resulting from an atmospheric trace gas simulation, distributed to 8 processors.

2. Load Balancing

Interpret the grid as a graph G:

$$G = (I, E);\ |I| = n$$

I node set (elements), E edge set. Then, the load balancing problem consists of finding a partition $I_1, I_2, \ldots, I_p$ such that:

(1) $$I_i \cap I_j = \emptyset, (i \neq j);\quad |I_i| = n/p;\quad \bigcup_i I_i = I.$$

Additionally the subset of edges in E that belong to different processors is to be minimized.

Originally, space-filling curves (SFCs) were introduced by Peano and Hilbert resp. [3]. Application of SFCs to dynamical load balancing of sparse grids has been proposed by Griebel and Zumbusch [2]. Our new approach modifies the space-filling curve for adaptive triangular grids (cf. Fig. 1, Algorithm. 1).

The recursive calculation of element indices by the SFC requires $O(n \log n)$ operations. All steps of the algorithm are parallelizable.

Algorithm 1 (Partitioning)

1. $\forall$ elements: calculate index by recursive SFC algorithm
2. Sort global index set I in ascending order
3. Partition I according to formula (1)
4. Assign each processor one I_j and distribute the corresponding elements accordingly

3. Parallel Software `pamatos`

The load balancing strategy, introduced in 2., has been implemented in a parallel self-adaptive grid generator [1]. The parallelization has been realized by several software layers, as given in figure 3. Utilizing these layers, a reduction of complexity and an improvement of portability can be achieved.

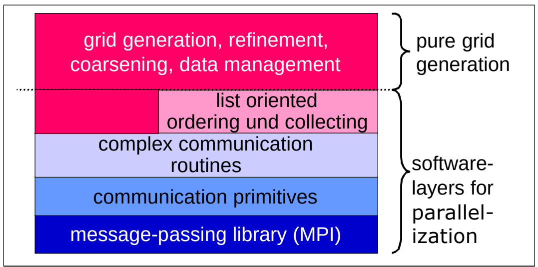

Fig. 3: Software layers in `pamatos`

4. Features

Adaptive grid generation for atmospheric and oceanic simulations
Parallelization for message-passing architectures
Dynamic load balancing by space-filling curves
Parallel load balancing algorithm
Portable grid generation software

References

1. **J. Behrens und J. Zimmermann** (2000): Parallelizing an unstructured grid generator with a space-filling curve approach, Report TUM-M0002, TU München.
2. **M. Griebel und G. Zumbusch** (1998): Hash-storage techniques for adaptive multilevel solvers and their domain decomposition parallelization, *Contemporary Math.*, 218, 279-286.
3. **D. Hilbert** (1891): Über die stetige Abbildung einer Linie auf ein Flächenstück, *Math. Ann.*, 38, 459-460.

Contact: Dr. Jörn Behrens, TU München, Zentrum Mathematik, D-80290 München, behrens@ma.tum.de, www-m3.ma.tum.de/m3/behrens/