**Simulation Techniques and Scientific Computing**
Prof. Dr.-Ing. Sabine Roller

UNIVERSITÄT SIEGEN

# Load Balancing in HPC Applications:
## Quantification of LB related performance optimizations

Monika Harlacher

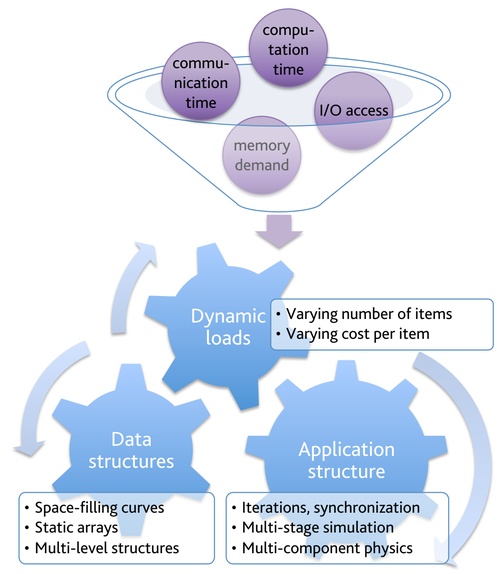
## Motivation

- Application pattern and data structures often limit load balancing quality in MPI applications.
- Impact of communication cost is hard to model.
- Implementation and testing of alternatives is usually not possible without major code intervention.
- Software redesign is expensive and actual performance gain uncertain in advance.
- Developers most often choose a design by experience rather than by cost assessment.
- **Means are missing to quantify performance differences in advance.**

- Study the correlation of


## Load balancing difficulties

- Balancing criteria is usually a combination of multiple objectives.
- Iterative nature of scientific simulations accumulate waiting time.
- **Multi-component** and **multi-stage** models generate additional dependencies.
- Workload may evolve over time.
- **Data structures** constrain partitioning layout.



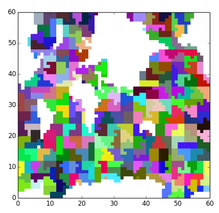

## Use cases: Impact of communication

### 1) Partitioning quality

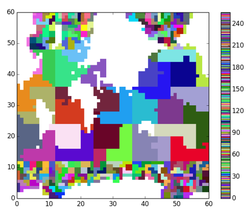
#### CESM static arrays limit partition sizes

Computational load in the CESM sea ice model is almost exclusively located at pole regions. The implementation uses fixed-size arrays, that bound memory demand and avoid allocation overhead. While optimal partitioning by computational weights is straightforward, it translates into different communication pattern.

- How do we model communication cost and performance gain?



CESM sea ice partitioning

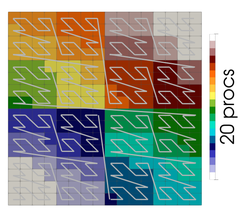


Optimal partitioning by computational weight

#### APES z-curve produces fragmented partitions

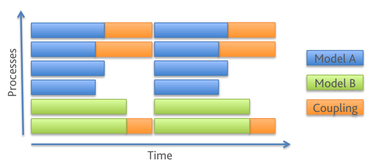
Mapping multi-dimensional domains to 1D curves preserves locality only partially. Fragmented partitions introduce additional communication.

- What performance gain can we expect when using smarter partitions?




Balanced partitioning on z-curve

### 2) Multi-stage applications: Coupling in APESmate




Domain is split into several physical subdomains which run on distinct sets of processors. Only interface elements need data exchange between the different models.

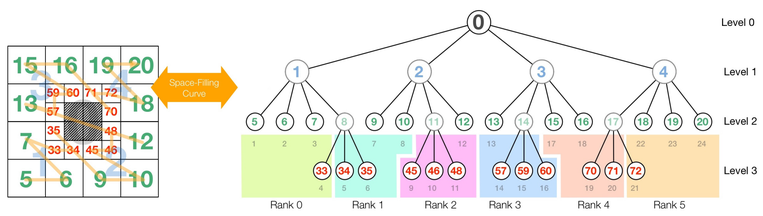
- How do we find an optimal configuration of processor sets and distribution of elements?

### 3) Multi-level implementation

In a multilevel domain elements depend on their neighbors, whether on the same level or next coarser and finer level. They frequently exchange information to progress simulation.

In the current Discontinuous-Galerkin implementation, processes first propagate information from coarser to finer elements. Compute flux between elements and communicate this information backwards. An element can advance the time step only with flux information from all its neighbors. Partitioning the octree data structure along a z-curve results in poor level-wise balance. Processes with unbalanced number of elements per level experience wait time.

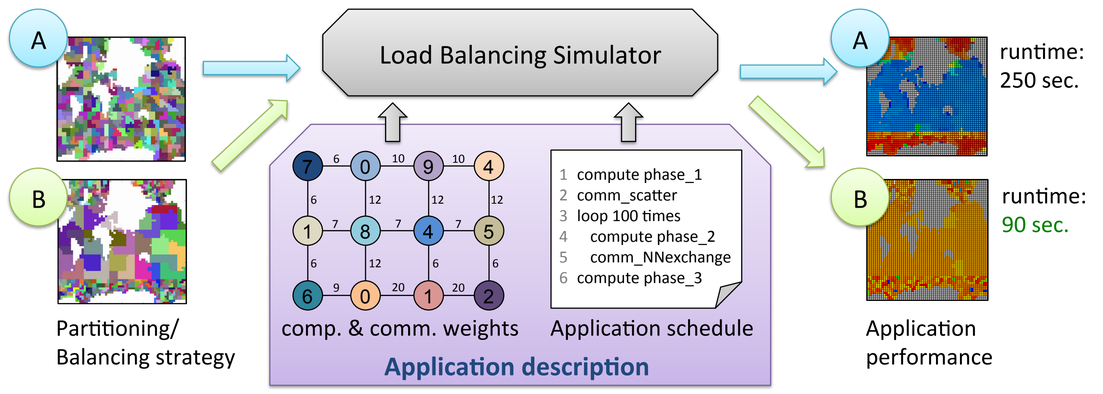
- What are alternative execution patterns and how can we model their effects on runtime?



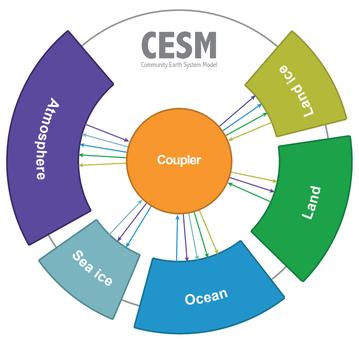

## Methodology: Load balancing simulator

Software engineering tool

- Run an abstract application model to study application behavior
- Facilitate comparison of different load balancing strategies
- Test exemplary implementation of alternative data and application structures
- Unnecessitate time-consuming code modifications to original application
- **Guide decisions for expensive re-engineering**




## Target applications




**CESM** :: Community Earth System Model, is a fully-coupled, community, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states.
Webpage: http://www.cesm.ucar.edu

**APES** :: Adaptive Poly-Engineering Simulator, provides an efficient simulation stack for engineering simulations. The framework relies on a common octree-based library and encompasses individual numerical solvers, which can also be coupled.
Webpage: https://www.mb.uni-siegen.de/sts


Simulation Techniques and Scientific Computing
Adolf-Reichwein-Straße 2, 57068 Siegen, Germany
http://www.mb.uni-siegen.de/sts