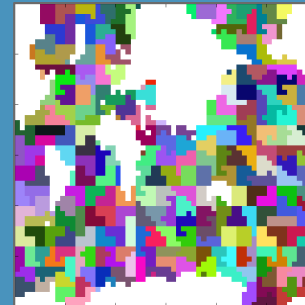
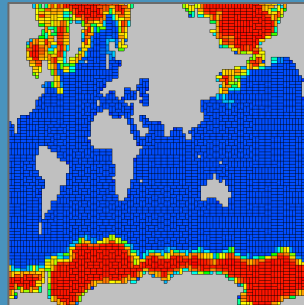
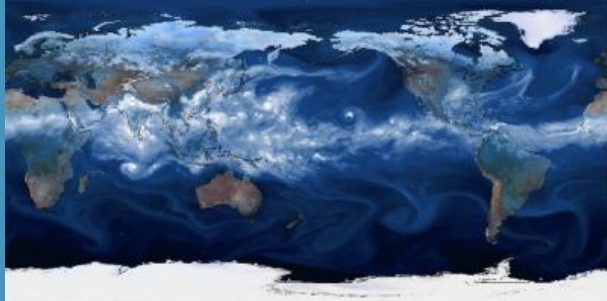




German Research School
for Simulation Sciences

Load-Balancing Simulator

Workshop on Extreme-Scale Programming Tools

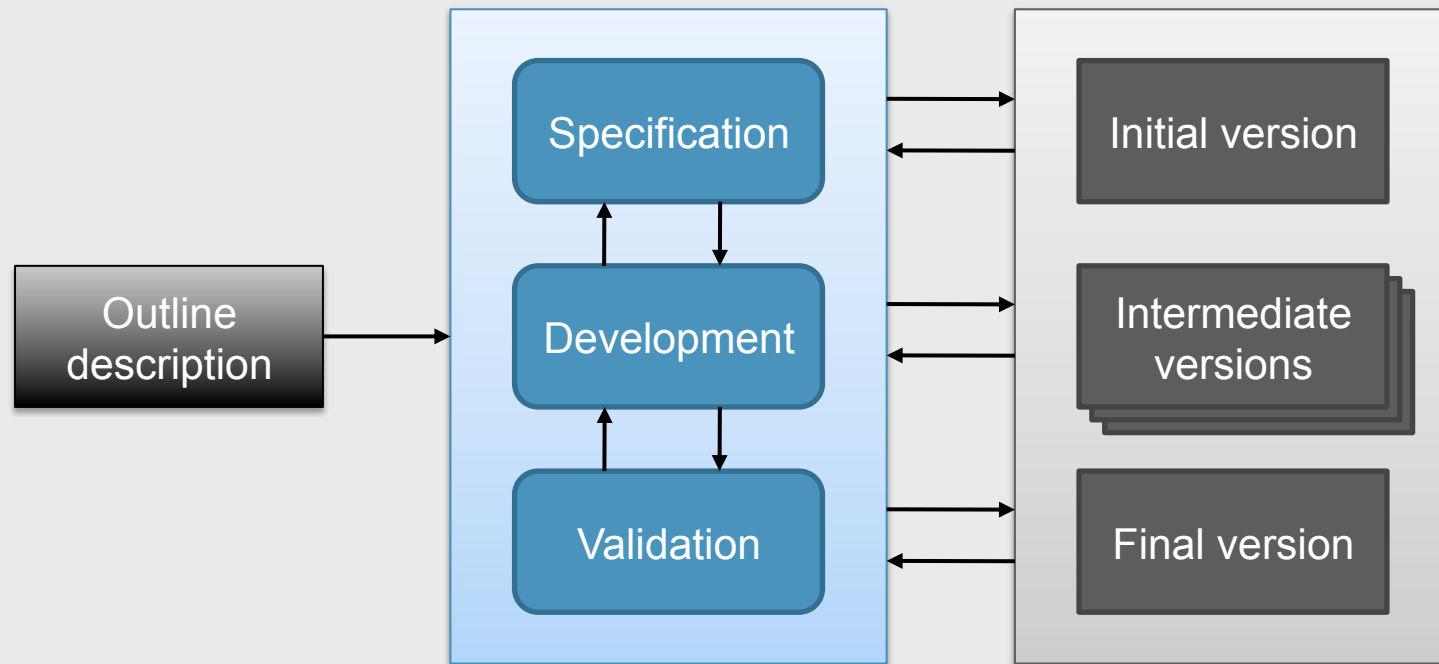


Supercomputing 2013, Denver, November 18

Monika Lücke, German Research School for Simulation Sciences

Software engineering in HPC

Incremental development



Source: Ian Somerville, Software Engineering, Pearson, 9th edition, 2011

Load balancing

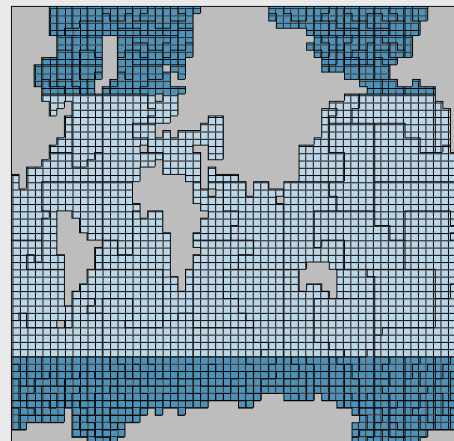
Assignment of work to processors

Objectives

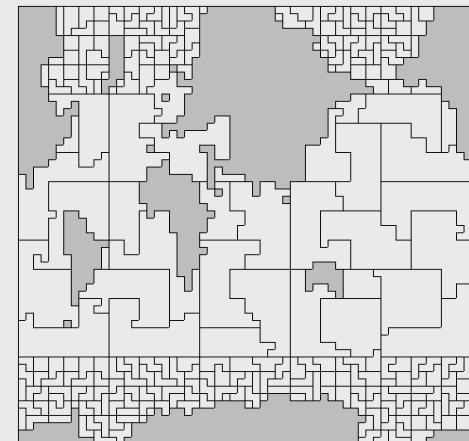
1. Balance computational load
2. Minimize communication
3. Balance communication

Classification

- static
- dynamic



Computational load



Partitioning
(comp. load balanced)

Load balancing (2)

Essential for performance and scalability

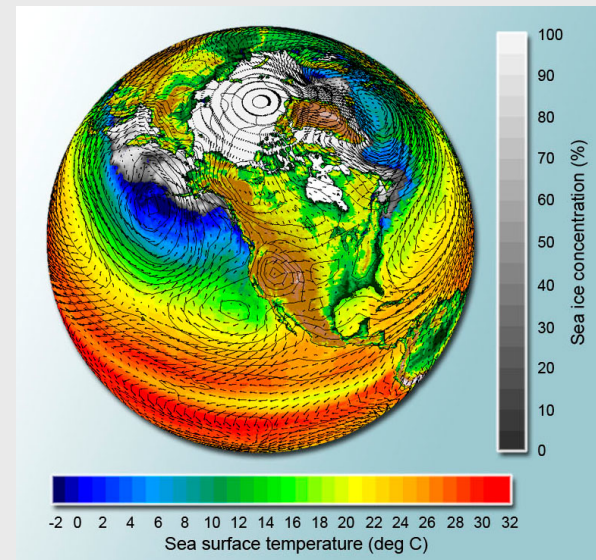
Typically added during a later stage of the development process

- Focus of the initial design: correctness & simplicity

Example

- Sea ice module of CESM
- Developed at NCAR

Problem: re-engineering a grown code base is expensive



How to evaluate a partitioning?

Analytical modeling

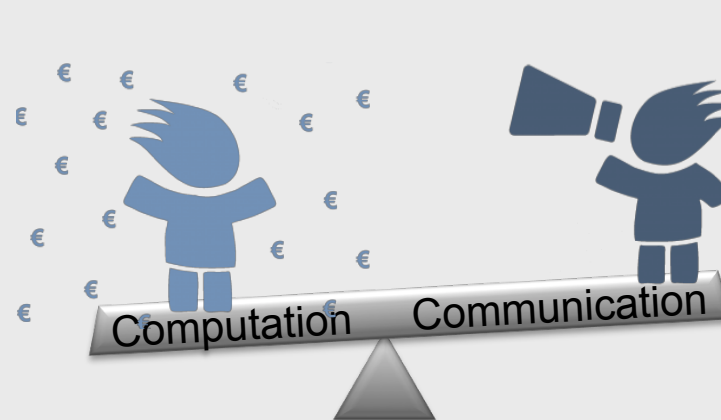
- + Saves computing time
- Requires expert knowledge in performance modeling
- Error-prone

Too laborious for the complex nature of simulation codes

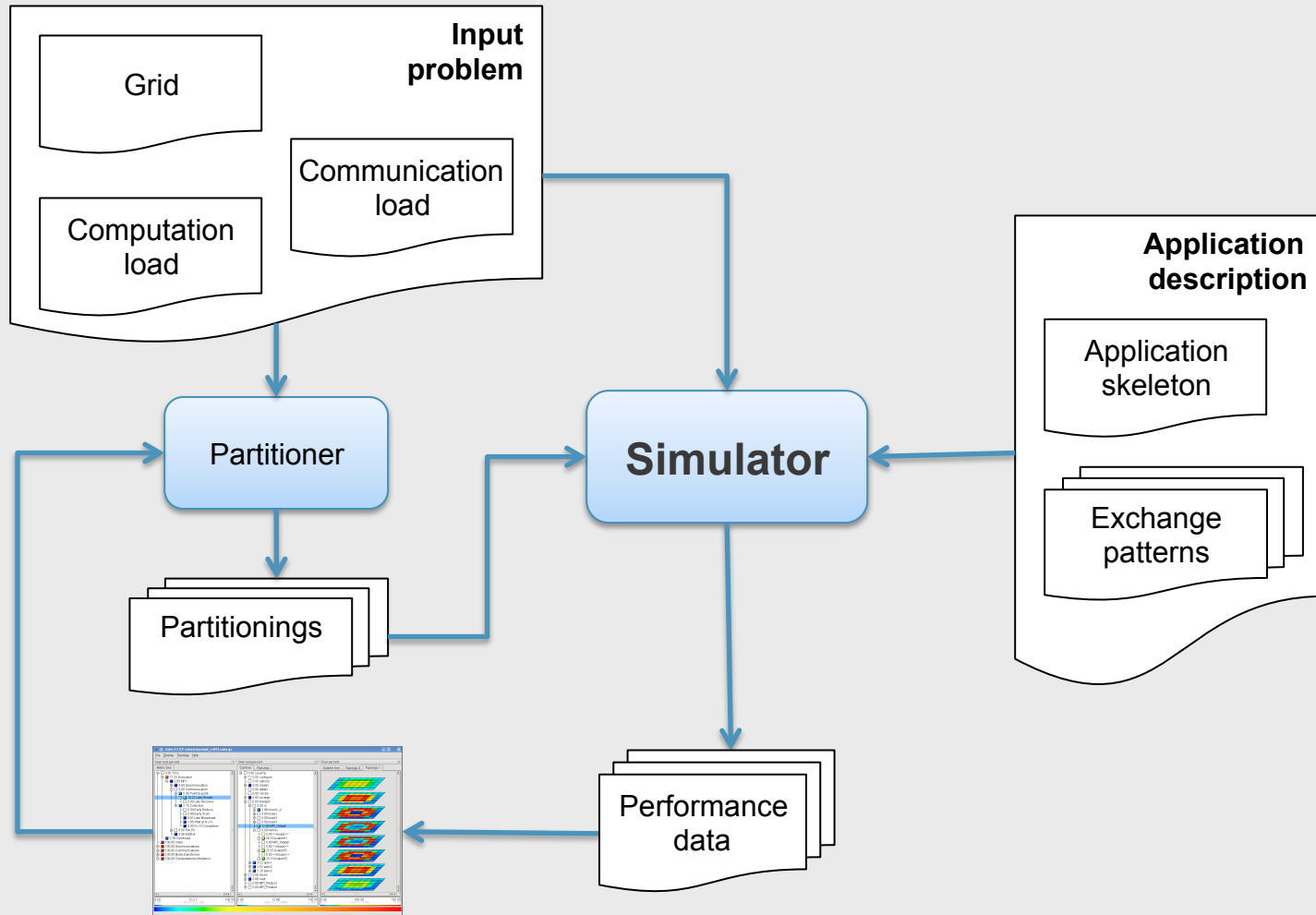
Test

- + Reliable
- Requires computing time
- Requires prior implementation

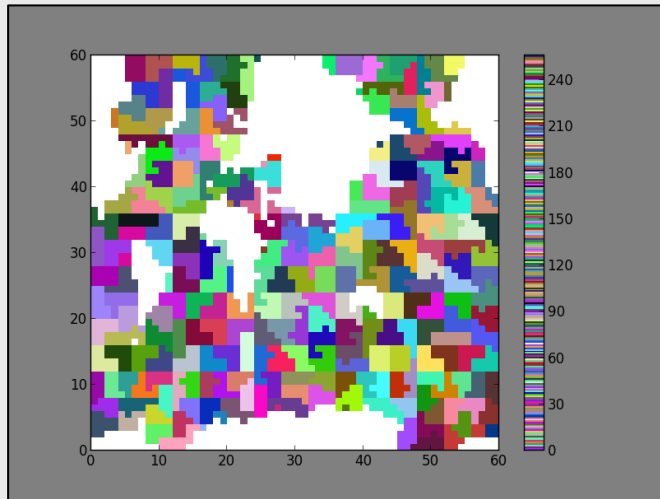
Too expensive to re-engineer a grown code base



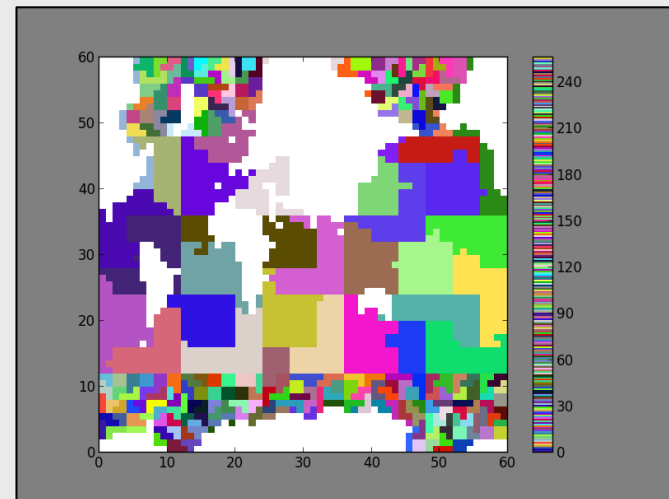
Tools needed to easily find a suitable partitioning



Partitioning

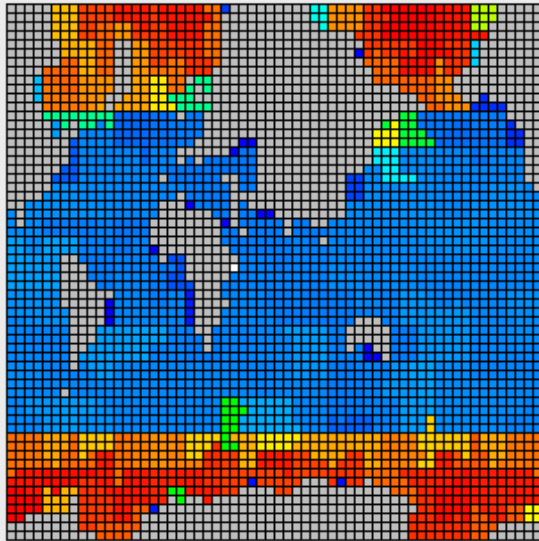


- Space-filling curve
- Weighted by probability of sea ice
- #elements per process restricted

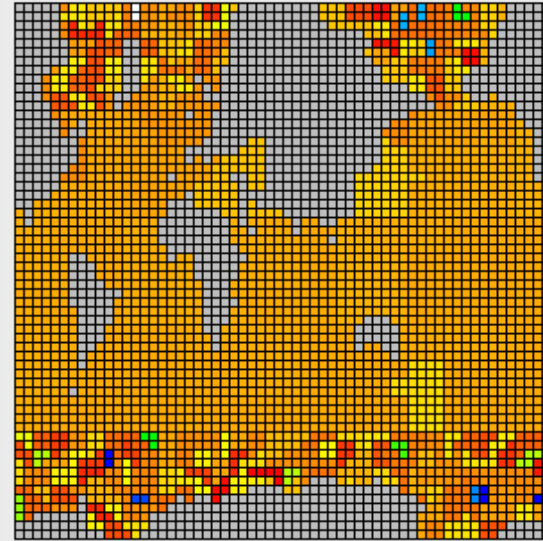


- Same space-filling curve
- Weighted by measured timings
- #elements unrestricted

Computation

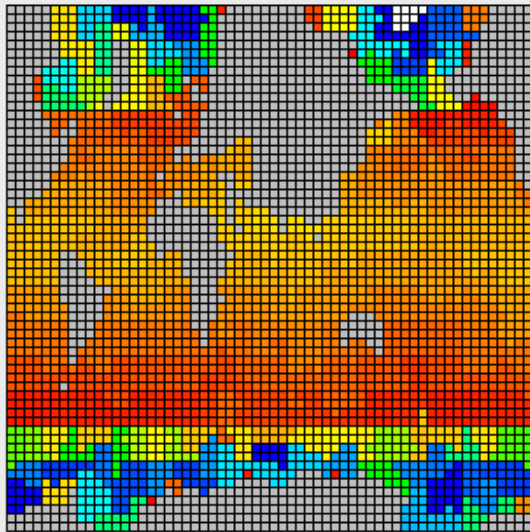


- $\Sigma = 19,800$ s
- min time = 2 s
- max time = 254 s
- $\Delta = 252$ s

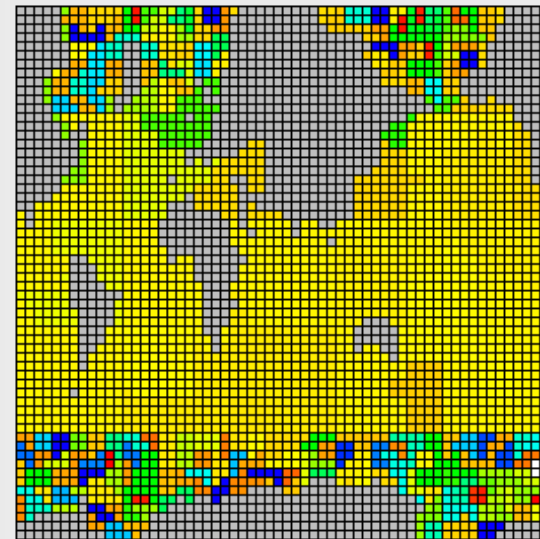


- $\Sigma = 19,800$ s
- min time = 54 s
- max time = 93 s
- $\Delta = 49$ s

Communication (incl. wait states)



- $\Sigma = 34,200$ s
- min time = 0 s
- max time = 247 s
- $\Delta = 247$ s



- $\Sigma = 3,200$ s
- min time = 0 s
- max time = 36 s
- $\Delta = 36$ s

Summary & outlook

Load-balancing simulator:

- Informed choice of a load-balancing strategy with little effort
- Test bed for alternative communication patterns
- Justification of expensive re-engineering decisions

Planned:

- Interfaces to most common partitioning libraries
- Further application case studies
- Score-P extension to automatically measure computation weights
- Communication pattern libraries

Acknowledgement

- John Dennis, National Center for Atmospheric Research
- Felix Wolf, German Research School for Simulation Sciences

ECS - Enable Climate Simulation at Extreme Scale

The G8 Research Councils Initiative on Multilateral Research Funding

Interdisciplinary Program on Application Software towards Exascale Computing for Global Scale Issues



Advertisement

Using Automated Performance Modeling to Find Scalability Bugs in
Complex Codes (Alexandru Calotoiu)

Wed 11/20/13 4:30 - 5pm room 405-407