A user-controlled GGDML code translation technique for Performance Portability of Earth System Models

Nabeeh Jum'ah, Julian Kunkel

Scientific Computing
Department of Informatics
University of Hamburg

ISC HPC Research Poster Presentation
2018-06-26

This work was supported in part by the German Research Foundation (DFG) through the Priority Programme 1648 "Software for Exascale Computing" (SPPEXA) (GZ: LU 1353/11-1)

GGDML and Higher-Level Code

Goals

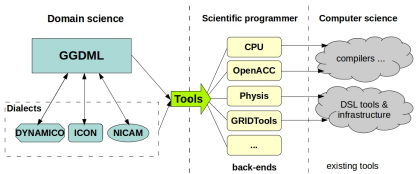
- Improve code quality, scientists productivity, code maintainability
- Provide better performance-portability of code

GGDML

- **GGDML**:***G**eneral **G**rid **D**efinition and **M**anipulation **L**anguage*
- Hides memory access details
- Abstracts higher concepts of grids, hiding connectivity details
- Constructs for the abstraction of grids
  - Grid and field definition
  - Field data access/update
  - Stencil operations

Strategy

- Foster separation of concerns
  - Scientists develop the problem logic in the source code
  - Scientific programmers provide translation configurations
- Modeling with GGDML
  - Allows using the modeling language, e.g. C
  - Provides language extensions
- GGDML features
  - Coding in terms of scientific concepts
  - No machine concepts
- A source-to-source translation tool translates code based on the configuration



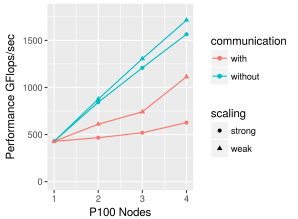
Highly Configuable Translation Process

- The set of the language extensions can be easily extended, for example
  - User defines groups of declaration specifiers, e.g. Dimension(2D or 3D)
  - Access operators are defined by the user
    - Simplifies definition of grid connectivity, e.g. cell.neighbor, cell.edge
    - Allows the user to add any needed operators, and control their bahavior
- Users control the optimization procedures, for example
  - Memory layout is completely controlled by user
    - Memory allocation
    - Index transformations (including mathematical transformations)
  - Parallelization is controlled by user
    - User can mark code with parallelization directives(tested with OpenMP & OpenACC)
    - Single node and multiple nodes (tested with MPI)
    - User controls communication libraries initialization & halo exchange code

Performance Evaluation

Performance on P100 (and V100) GPUs, with OpenACC and MPI

	Serial	P100			V100		
		performance GF/s	Memory throughput GB/s		performance GF/s	Memory throughput GB/s	
			read	write		read	write
3D	1.97	220.38	91.34	56.10	854.86	242.59	86.98
3D-1D	1.99	408.15	38.75	43.87	1240.19	148.49	57.12



Performance on Broadwell processors with OpenMP and MPI