Automatic code generation for highly parallel multigrid solvers

Sebastian Kuckuk, Christian Schmitt, Harald Köstler

1 Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Department of Computer Science 10 (System Simulation)
2 Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Department of Computer Science 12 (Hardware-Software-Co-Design)

### Project ExaStencils
- Generation of efficient, robust and exa-scalable geometric multigrid solvers

### ExaStencils
- Sebastian Kuckuk
- Harald Köstler
- Ulrich Rüde
- Alexander Grebhahn
- Sven Apel
- Hannah Rittich
- Matthias Bolten
- Stefan Kronawitter
- Armin Größlinger
- Christian Lengauer

### Geometric Multigrid
- Smoothing of high frequency errors
- Coarsened representation of low frequency errors

### Multi-Layered DSL Approach
- From abstract problem specification on layer 1 to concrete solver implementation on layer 4
- L1: mathematical formulation of problem
- L2: discretization of the problem
- L3: specification of algorithmic components
- L4: complete program specification

### Code Generation with Scala
- Necessary due to the high variance of the multigrid domain
- Hardware - CPU, GPU or both? Number of nodes, sockets and cores?
- Cache characteristics? Network characteristics?
- Software - MPI, OpenMP or both? CUDA or OpenCL? Which version?
- MG components - Cycle type? Which smoother(s)? Which coarse grid solver? Which inter-grid operators?
- MG parameters - Relaxation? Number of smoothing steps?
- Optimizations - Vectorization? Temporal Blocking? Loop transformations?
- Problem description - Which PDE? Which boundary conditions?
- Discretization - Finite Differences, Finite Element or Finite Volumes?
- Domain - Uniform or block-structured? How to partition?

### References:

### Code Generation with Scala
- Modular and feature-rich code generation and transformation framework written in Scala [1]
- Automatic low-level optimization via polyhedral transformations [2]
- Interface to SPL and LFA prediction and optimization [3]