CG-Kit, Code Generation Toolkit for Performant and Maintainable Variants of Source Code

Scientific Achievement

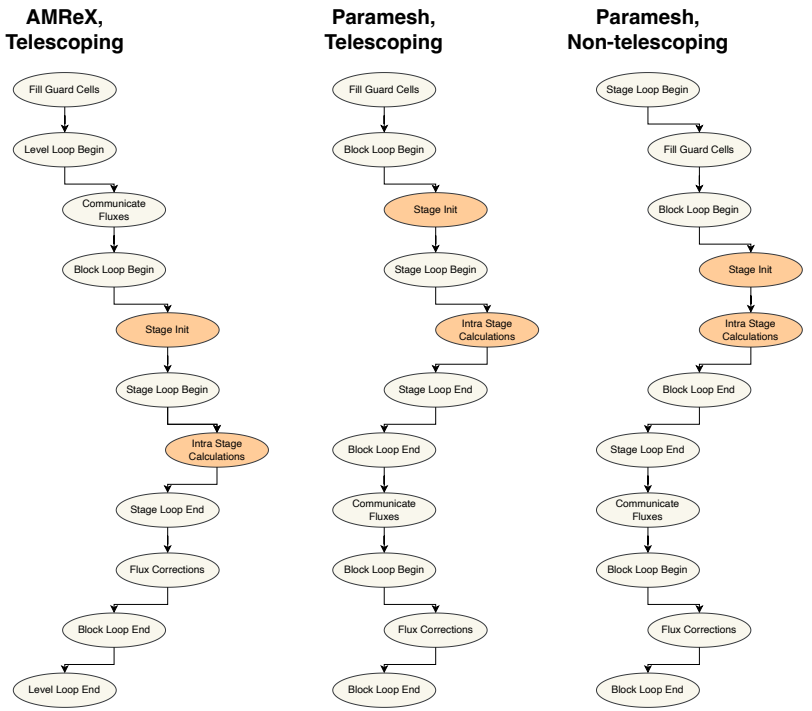
The toolkit enables different realizations of numerical algorithms that lead to the same solution outcome but differ in the details of algorithm design and/or the implementation of how the solution is obtained. The variants enabled by CG-Kit go beyond the kind of code unification achieved by C++ based abstractions. With CG-Kit it is possible to maintain succinct descriptions of control flow that avoid inclusion of numerical details. Code generation stitches in numerical components while translating the recipe into compilable code.

Significance and Impact

- As heterogeneity grows with different accelerators, more variants of control flow are needed, sometime even different algorithms may be needed.
- CG-Kit enables easy way to express map of computation to resources and separates numerical details control flow at various granularities.

Technical Approach

- Express control flow and map using a recipe interface in Python
- Translate recipes into parameterized source trees – substitute for abstract syntax tree to avoid having to use parsers and compiler technology
- Algorithmic and platform specific information encoded in templates
- Code emitted in a compilable format



Simplified graph representations for three variants of hydrodynamics solver (Spark) in Flash-X. The nodes shown with gray background denote code generation operations. Spark's numerical algorithms are represented in orange, which are subgraphs of control flow consisting of multiple nodes, and these subgraphs are reused in all three variants.

PI(s)/Facility Lead(s): Person Name; Anshu Dubey
Collaborating Institutions: Virginia Tech, Riken
ASCR Program: [ECP, SciDAC.]
ASCR PM: Lali Chatterji
Publication(s) for this work: https://doi.org/10.48550/arXiv.2401.03378