Towards Compositional and Generative Tensor Optimizations
Adilla Susungi, Norman Rink, Jerónimo Castrillón, Immo Huismann, Albert Cohen, Claude Tadonki, Jörg Stiller, Jochen Fröhlich

To cite this version:
Towards Compositional and Generative Tensor Optimizations
Adilla Susungi, Norman A. Rink, Jerónimo Castrillón, Immo Huismann, Albert Cohen, Claude Tadonki, Jörg Stiller and Jochen Fröhlich
adilla.susungi@mines-paristech.fr — norman.rink@tu-dresden.de

Tensors in Computational Fluid Dynamics (CFD)
- Loop characteristics:
  - 3 to 4 dimensions nesting
  - Few iterations per dimension (e.g., 17 or 33 iterations)
- Type of computations:
  - Tensor contractions
  - Outer products
  - Element-wise multiplications
  - Computations on each element of a structured mesh

Inverse Helmholtz
\[
t_{ijk} = \sum_{l,m,n} A_{kn}^T \cdot A_{jm}^T \cdot A_{il}^T \cdot u_{lmn}
\]
\[
p_{ijk} = D_{ijk} \cdot t_{ijk}
\]
\[
v_{ijk} = \sum_{l,m,n} A_{kn} \cdot A_{jm} \cdot A_{il} \cdot p_{lmn}
\]

Tensor Optimization Frameworks
- Domain-specific expressivity
  - Flexible/Adaptive optimization heuristics
- Hidden and/or rigid expressivity
  - Generic expressivity

Related Work
- Different levels of expressiveness and control on optimizations
  - Flexible/adaptive
  - Hidden/rigid

Chill • Pluto • TensorFlow • TVM • Tensor Contraction Engine • Numpy • Tensor Algebra Compiler

Optimizing CFD Kernels with Existing Tools
- Several limitations
- Few opportunities for adaptations
- Unadapted constructs
- Unadapted heuristics
- Limited expressivity
- Limited optimizations

Goal
A cross-domain intermediate language for tensor optimizations

Intermediate Language
- Modular constructs
- First-class citizens:
  - Arrays
  - Tensor operators
  - Loop iterators
  - Transformations

Envisioned Tool
- Meta-programming
- Iterative search

Future Work
- Applications to other domains
  - Syntax refinement
  - Formal semantics

Search Space Exploration
- Evaluation order of tensor contractions
- Fusions
- Permutations
- Vectorization
- Collapsing
- Unrolling

Inverse Helmholtz by Example

Example of assessment: Different heuristics of loop interchanges (+ parallelization)

Variant L1: Loop interchanges only;
Variant L2: Loop interchanges + data transpositions with copying;
Variant L3: Loop interchanges + data transpositions without copying.

Baseline: sequential execution (3.32s). Machine: 24-core Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz (Haswell)

This work was partially funded by the German Research Council (DFG) through the Cluster of Excellence ‘Center for Advancing Electronics Dresden’ (cfaed) and by PSL Research University through the ACOPAL project.