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 optimization heuristics

Related Work

- Different levels of expressiveness and control on optimizations

Flexible/adaptive

Hidden/rigid

Specific

Generic

Tensor Optimization Frameworks

- Several limitations
- Few opportunities for adaptations

Optimizing CFD Kernels with Existing Tools

- Limited expressivity
- Limited optimizations
- Unadapted heuristics
- Unadapted constructs

Should we create yet another domain-specific solution?

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

Search Space Exploration

- Evaluation order of tensor contractions
- Fusions
- Permutations
- Vectorization
- Collapsing
- Unrolling

Inverse Helmholtz by Example

# Basic array declaration
A = array(2, double, [N, N])
u = array(3, double, [N, N, N])
D = array(3, double, [N, N, N])

# Transposition
At = vtranspose(A, 1, 2)

# Tensor contractions
tmp1 = contract(At, u, [2, 1])
tmp2 = contract(At, tmp1, [2, 2])
tmp3 = contract(At, tmp2, [2, 3])

# Tensor contractions
tmp4 = entrywise(D, tmp3)

# Tensor contractions
tmp5 = contract(A, tmp4, [2, 1])
tmp6 = contract(A, tmp5, [2, 2])

# Loop interchanges
interchange(i4, i3)
interchange(i4, i2)
interchange(j2, j1)
interchange(j1, j4)

# Transpositions
tmp2t = vtranspose(tmp2, 1, 2)

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

Future Work

- Applications to other domains
- Syntax refinement
- Formal semantics

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.