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:

HAL Id: hal-01666818
https://hal-mines-paristech.archives-ouvertes.fr/hal-01666818
Submitted on 18 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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

\[ t_{ijk} = \sum_{l,m,n} A_{kn}^T \cdot A_{jm}^T \cdot A_{il}^T \cdot w_{lmn} \]

\[ p_{ijk} = D_{ijk} \cdot t_{ijk} \]

\[ v_{ijk} = \sum_{l,m,n} A_{kn} \cdot A_{jm} \cdot A_{il} \cdot p_{lmn} \]

Search Space Exploration

- Several limitations
- Few opportunities for adaptations

Should we create yet another domain-specific solution?

Intermediate Language

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

Envisioned Tool

Meta-programming

Iterative search

Flexible/Adaptive optimization heuristics

Generic expressivity

Optimized C

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
\( tmp1 = contract(At, u, [2, 1]) \)
\( tmp2 = contract(At, tmp1, [2, 2]) \)
\( v = contract(A, tmp6, [2, 3]) \)

# Iterator declaration
\( i1 = iterator(0, N, 1) \)
\( i2 = iterator(0, N, 1) \)

# ... other iterator declarations

# Association of iterators to computations
build(D, \{td1, td2, td3\})
build(tmp1, \{i1, i2, i1, i4\})
# Also applies to tmp5, ... , tmp6
build(v, \{k12, k22, k32, k42\})

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.