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
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:

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

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

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

Tensor Optimization Frameworks

- Domain-specific expressivity
- Flexible/Adaptive optimization heuristics
- Hidden and/or rigid optimization heuristics

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(A, u, [2, 1])
tmp2 = contract(A, tmp1, [2, 2])
tmp3 = contract(A, tmp2, [2, 3])

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

# Association of iterators
build(D, [td1, td2, td3])
build(tmp1, [i1, i2, i3, i4])

# Element-wise multiplication
tmp4 = entrywise(D, tmp3)

# Tensor contractions
tmp5 = contract(A, tmp4, [2, 1])
tmp6 = contract(A, tmp5, [2, 2])
v = contract(A, tmp6, [2, 3])

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

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

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

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.