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

Towards Compositional and Generative Tensor Optimizations

**Tensor Optimization Frameworks**

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

**Inverse Helmholtz**

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

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

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

**Optimizing CFD Kernels with Existing Tools**

- Several limitations
- Few opportunities for adaptations

### Related Work

- Different levels of expressiveness and control on optimizations

- **Flexible/adaptive**
  - Chill
  - Pluto
  - TensorFlow
  - TVM
  - Tensor Contraction Engine
  - Numpy
  - Tensor Algebra Compiler

- **Hidden/rigid**

<table>
<thead>
<tr>
<th>Specific</th>
<th>Generic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Must adapt constructs</td>
<td>Must adapt heuristics</td>
</tr>
<tr>
<td>Unadapted constructs</td>
<td>Unadapted heuristics</td>
</tr>
<tr>
<td>Limited optimizations</td>
<td>Limited expressivity</td>
</tr>
</tbody>
</table>

**Intermediate Language**

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

**Envisioned Tool**

- Meta-programming
- Iterative search

- **Flexible/Adaptive expressivity**
- Iterative search

**Search Space Exploration**

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

**Intermediate Language vs. Optimized C**

The intermediate language allows for flexible and adaptive optimizations, while the optimized C code can achieve similar performance but requires more manual effort.

**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

**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.