Elements: Transformation-Based High-Performance Computing in Dynamic Languages

Andreas Klöckner <andreask@illinois.edu>

Department of Computer Science · University of Illinois at Urbana-Champaign · Urbana, IL

OAC-1931577

Comp. Science on Today's Hardware

Computational science software must run at an appreciable fraction of peak performance on many different machines:

- Nvidia GPUs
- Intel/AMD wide-vector CPUs (AVX-256, AVX-512)
- Intel/AMD GPUs
- ...

Programming Challenges

- Array/vector architectures
- High memory latency
- Limited on-chip memory

Simplification: Use CL/CUDA abstract machine model for all of them
Still: Machine-specific trade-offs/capacities
Therefore: Machine-specific code needed
And: Many dialects (OpenCL, CUDA, ISPC, OpenMP+Pragma SIMD, ...)

Competing Approaches

- C++ Metaprogramming
- Libraries

PyOpenCL/PyCUDA: HPC in Python

Describe computation

Write/Generate code

Organize computation

Compute result

Loopy

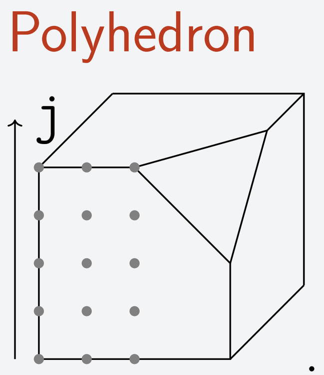
Loopy is a code generator for computation with arrays.

Performance: human 'in the loop' for the foreseeable future.

- Capture math at a high level; target **number crunching**
- Progressively 'lower' through manual transformations
- Observe and control optimization steps
- 'Help me write the CUDA C/ISPC/... I would write'

Loopy: Program Representation

Polyhedron




```
{[i,j]:
  0<=i<n and
  0<=j<n and
  ...}
```

Statement(s)

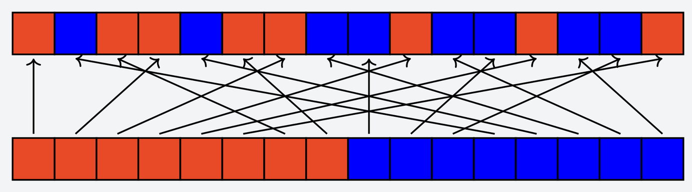
```
b[i] = sum(j,
    A[i,j] * x[j])
```

Summary
Tree of Polyhedra
+ Statements
+ Dependencies
= Semantics

Vector Shuffles via Array Access

Vector shuffles complement shared memory as an efficient means of communication.

- less synchronization
- less energy expense
- harder to program

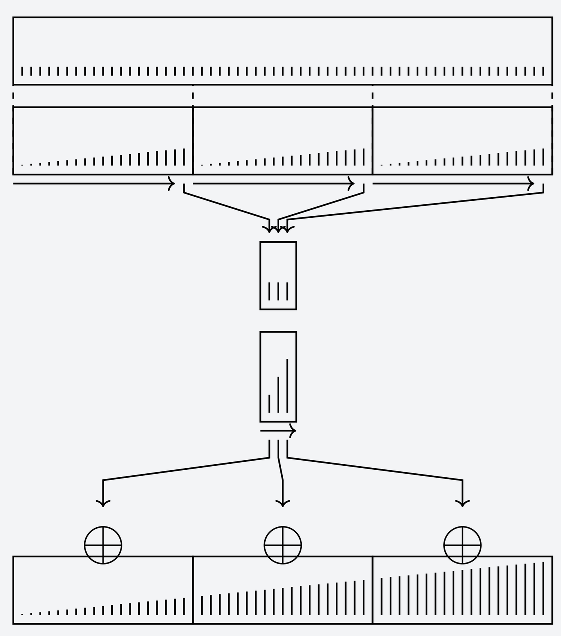


Idea: Represent them in *Loopy*'s intermediate representation, make them reachable by transforms (e.g. for SoA/AoS transpose)

Transformation, Efficieny for Scan

Scan/prefix sums are a core parallel primitive [Blelloch '93], with uses in, e.g.:

- threads with variable-size output
- sorting
- filtering



Idea: Allow *Loopy* to represent and transform scans

Transform Addressing

Core difficulty in program transformation: **Transform addressing**, i.e. specifying *where* a transform should apply.

Notation must be:

- compact
- specific
- human-readable

Idea: A *query language* acting on the program representation

A Transform-Capable Array Package

numpy code is widespread, e.g.:

- neural nets
- computational science
- image processing

numpy code could use GPUs well, but not robustly high performance *without help*.

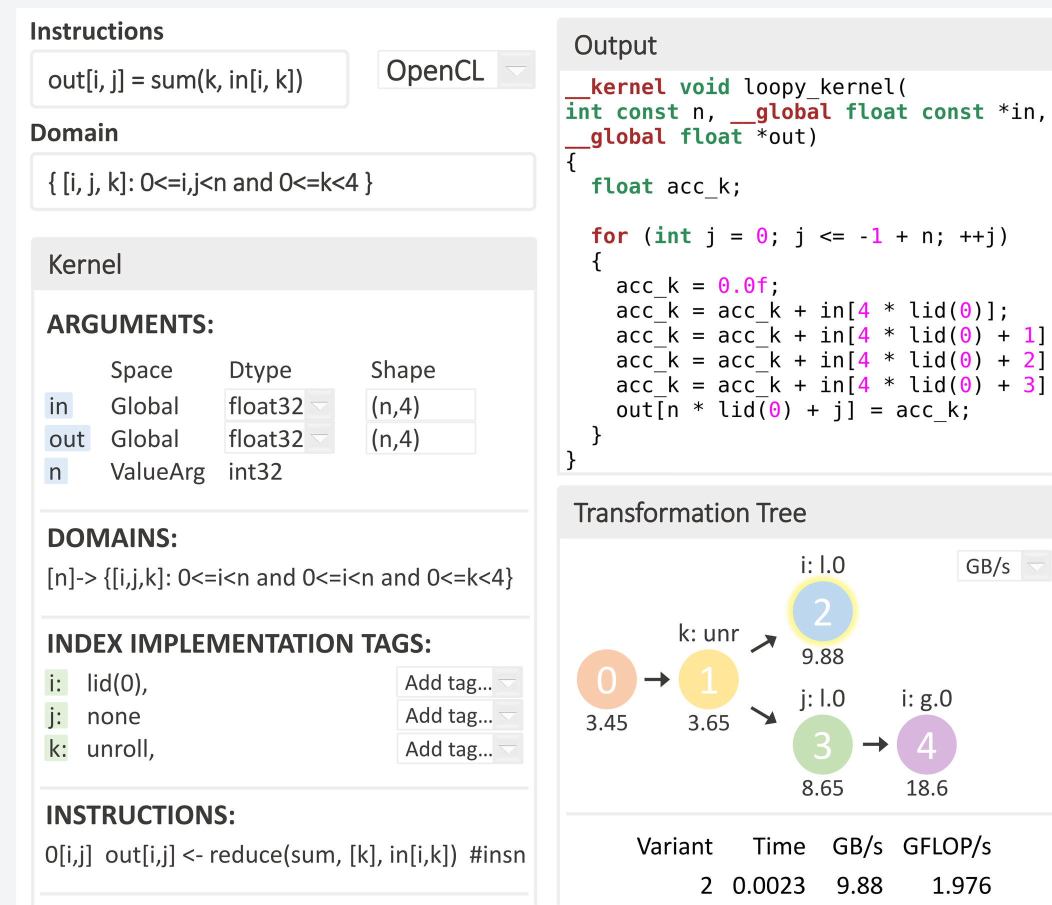
```
result[1:-1] = v[2:] - v[:-2]
result[0]    = v[1]  - v[0]
result[-1]   = v[-1] - v[-2]
```

- Many realizations of "lazy *numpy*"
- For automatic differentiation, performance

Ideas:

- a reusable lazy array package
- connect it to *Loopy* codegen
- allow user intervention for performance
- replace PyOpenCL/PyCUDA array objects

Graphical Transform User Interface



- Improves Transform Discoverability
- Eases Experimentation
- Supports Performance Modeling

(Ask for a demo!)

Loopy Usage Example

```
import loopy as lp
k = lp.make_kernel(
    "{[i]:0<=i<n}",
    "out[i]=2*a[i]")

k = lp.split_iname(k, "i", 128)
k = lp.tag_inames(k, "i_outer:g.0")
k = lp.tag_inames(k, "i_inner:l.0")
```

Loopy: Summary

Github:
https://github.com/inducer/loopy

- **One** intermediate representation from math to low-level machine code
  - Shared medium between human and machine
- Transformations to cover the difference
- Make near-peak performance accessible by manual transformation
- Allow autotuning/automated search to be implemented "on top"

A user quote about Loopy's IR:

> *We believe that loopy's level of abstraction hits this sweet spot needed for a high performance code generation workflow. [Kempf et al. '18]*

Known Science Users

- Firedrake finite element framework: https://arxiv.org/abs/1903.08243
- Dune PDElab finite element framework: http://arxiv.org/abs/1812.08075
- Pystella stencil-based cosmology solver: https://arxiv.org/abs/1909.12843, https://arxiv.org/abs/1909.12842
- Computational neuroscience: https://doi.org/10.3389/fninf.2018.00068
- SIMD/SIMT for chemical kinetics: https://doi.org/10.1016/j.combustflame.2018.09.008

ILLINOIS