Complex data structures scheduling

Paul Iannetta
ENS de Lyon, France
Ph.D. advisors: Laure Gonnord, Lionel Morel

Our Team: CASH

- **Optimization and Dataflow Parallelism for HPC Applications**
  - Better understand HPC Applications and their needs
  - Automatically reschedules instructions at statement-level to harness data locality and cache mechanisms
- **Scalable Static Analyses**
  - Design new low-cost analyses
  - Revisit syntax-based optimizations and polyhedral model within the Abstract Interpretation framework

High Performance Computing

High Performance Computing uses supercomputers to achieve computations. Supercomputers have:
- Many Cores: GPUs and CPUs
- Complex Memory Architecture

Need to have efficient compiler optimizations that can harness the extreme degree parallelism offered by these architectures.

A Static Analysis Framework: The Polyhedral Model

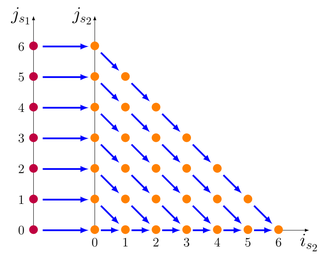
The polyhedral model is a framework used to extract the hidden parallelism of computation intensive kernels. Many kernels come from linear algebra and can be expressed with affine loops and array accesses.

Example 1. Polynomial Multiplication
Let A, B and C be the polynomials defined as follows:

$$A(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$
$$B(x) = b_0 + b_1x + b_2x^2 + \cdots + b_mx^m$$
$$C(x) = A(x)B(x)$$

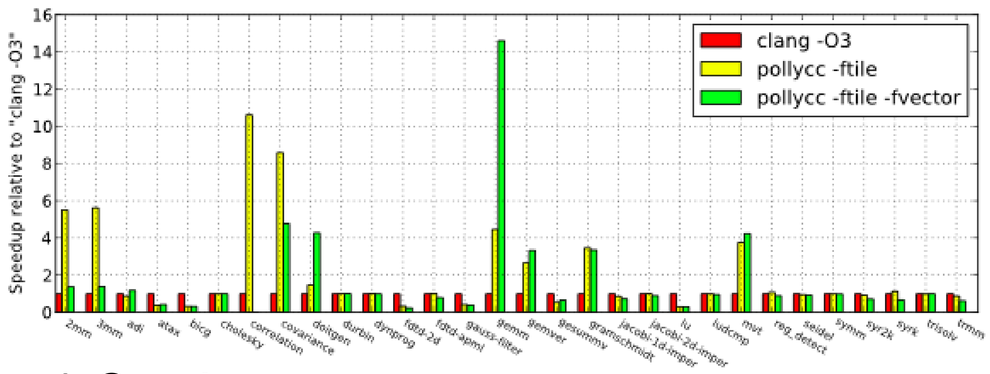
```
    for (int j = 0 ; j <= n + m ; ++j)
s1:     c[j] = 0

    for (int i = 0 ; i <= n ; ++i)
      for (int j = 0 ; j <= m ; ++j)
s2:     c[i+j] = c[i+j] + a[i] * b[j]
```

Computation of C



Dependencies

Below is a benchmark with optimizations provided by Polly[1], an extension of llvm, which enable polyhedral optimizations at compile time.



Research Questions:

- How to deal with non regular control flow? WIP: Approximation
- How to handle complex data structures? WIP: Balanced Binary Trees

[1] http://polly.llvm.org/performance.html

WIP 1: Approximate Program Analysis

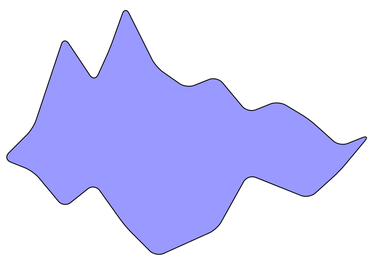
Why Approximate Programs?

- Polyhedral programs is a small class of programs.
- Polyhedral programs can be greatly optimized.
- Use polyhedral programs as approximations of general programs.

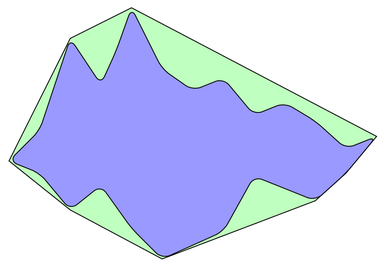
What kind of approximations?

- What interests us are iteration variables.
- As those variables change they draw a domain in space.
- In the best case this domain is a convex polyhedron.
- If not we approximate it as a polyhedron.

Original Domain



Polyhedral Approximation



Challenges

- How to abstract the area as a polyhedron properly especially the evolution of polyhedron when in a loop?
- How to deal with the spurious data point added by the transformation when analysing the behaviour of the inter-operations dependencies when generating the code?

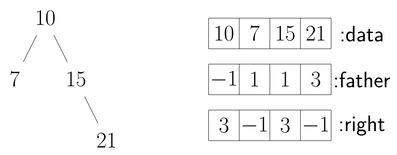
WIP 2: Balanced Binary Tree

Optimizations on Balanced binary trees

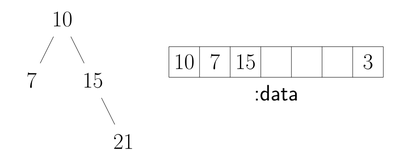
- Change memory layout to improve data locality of trees on primitive datatypes.
  - Tree as an array which uses depth-first order.
  - Tree as an array which uses breadth-first order.
- Merge independent or partially-independent tree traversals.

Properties of different representations:

Properties	Representation: Breadth First	Depth First
Indexes are predicable	✓	✗
Rotations are cache-friendly	✓	✓
No holes	✗	✓
A subarray is a subtree	✗	✓
Insertion is cheap	✓	✗



Depth-first representation



Breadth-first representation

Challenges

- Make the change of layout seamless to the programmer.
- Make the change compatible with the polyhedral model.
- Express tree traversal merges into the polyhedral model.

http://perso.ens-lyon.fr/paul.iannetta
paul.iannetta@ens-lyon.fr