CS671 Parallel Programming in the Many-Core Era

Polyhedral Framework for Compilation: Polyhedral Model Representation, Data Dependence Analysis, Scheduling and Data Locality Optimizations

December 3, 2013
Table of Contents

Polyhedral Program Optimizations

Code to Model: Polyhedral Model
  Polyhedral Model Overview
  Modeling Dependence

Transformations in the Model
  Classical Affine Transformations
  Affine Scheduling - One Dimensional Time
  Affine Scheduling - Multi-Dimensional Time

Model To Code
  The Problem Definition
  Polyhedral Code Generation Approaches

References
Polyhedral Program Optimization:

A three-stage process

1. Analysis: from code to model
   ▶ Existing prototype tools
     PoCC (Clan-Candl-LetSee-Pluto-Cloog-Polylib-PIPLib-ISL-FM)
     URUK, Omega, Loopo
   ▶ GCC GRAPHITE
   ▶ Reservoir Labs R-Stream, IBM XL/Poly

2. Transformation in the model
   ▶ Build and select a program transformation

3. Code generation: from model to code
   ▶ Apply the transformation in the model
   ▶ Regenerate syntactic (AST-based) code
Lecture 7 Table of Contents

Polyhedral Program Optimizations

Code to Model: Polyhedral Model
  Polyhedral Model Overview
  Modeling Dependence

Transformations in the Model
  Classical Affine Transformations
  Affine Scheduling - One Dimensional Time
  Affine Scheduling - Multi-Dimensional Time

Model To Code
  The Problem Definition
  Polyhedral Code Generation Approaches

References
Motivation

An Example of Loop

```c
for ( i = 0; i < 3; ++i )
    for ( j = 0; j < 3; ++j )
        A[i][j] = i * j;
```

Loop execution:

```
A[0][0]=0*0  A[0][1]=0*1  A[0][2]=0*2
```
Capture Loop Characteristics

What will be needed?

- Need to indicate total number of iterations
- Need to distinguish every different iteration
- Need to imply an order on these different iterations

Challenges

- Compact (memory space consumption)
- Parametric loop bound / unbounded loop
- Non-unit loop stride
- Conditionals
Overview of Polyhedral Model

Iteration domain:

- A set of *n-dimensional* vectors
  - *Iteration Vector*: \( \vec{x} = (i, j) \)
  - Iteration domain: the set of values of \( \vec{x} \)

- **Polytopes** model sets of totally ordered n-dimensional vectors
- The set must be convex
- Example:

**Code**

```plaintext
for ( i=0; i < 5; i++ )
    for ( j=0; j < 5; j++ )
        ....
    endfor
endfor
```

**Iter. Domain Graph**
**Convexity**

**Definition: Convex Set**
Let $\mathbb{R}$ be a vector space over the real numbers. A set $S$ in $\mathbb{R}$ is convex iff, $\forall \mu, \lambda \in S$ and given any $t \in [0,1]$, the point

$$(1 - t) \ast \mu + t \ast \lambda \in S$$

**In words:**
Drawing a line segment between any two points of $S$, each point of this segment is also in $S$. 
Convexity in Loop Iteration Domains

A statement surrounded by loops with unit-stride, no conditional and constant loop bounds has a convex iteration domain, if the iteration domain exists.

\[ \text{for ( } i=0; i < N; i++ \text{ )} \]

\[ \ldots \]

In general, in loops

- The polytope set must be convex
- Convexity is the central concept of polyhedral optimization
Definition

A function \( f: \mathbb{K}^m \rightarrow \mathbb{K}^n \) is affine if there exists a vector \( \vec{b} \in \mathbb{K}^n \) and matrix \( A \in \mathbb{K}^{n \times m} \) such that:

\[
\forall \vec{x} \in \mathbb{K}^m, f(\vec{x}) = A\vec{x} + \vec{b}
\]

Affine half-space definition

An affine half-space of \( \mathbb{K}^m \) is defined as the set of points:

\[
\{ \vec{x} \in \mathbb{K}^m | \vec{a} \vec{x} \leq b \}
\]
Example of an affine half-space:

\[ 3x_1 + 2x_2 \geq 11 \]

\[ 3x_1 + 2x_2 \leq 11 \]

\[ 3x_1 + 2x_2 = 11 \]
Polyhedron definition

A set $S \in \mathbb{K}^n$ is a polyhedron if there exists a system of a finite number of inequalities $A\vec{x} \leq \vec{b}$ such that:

$$P = \{\vec{x} \in \mathbb{K}^n | A\vec{x} \leq \vec{b}\}$$

Equivalently, it is the intersection of finitely many half-spaces.
Example of a polyhedron generated by intersection of three half-spaces:
Polyhedron III

A polytope is a bounded polyhedron

Integer Polyhedron ($\mathbb{Z}$-polyhedron)
It is a polyhedron where all its extreme points are integer valued

Integer Hull
The integer hull of a rational polyhedron $P$ is the largest set of integer points such that each of these points is in $P$. 
Polyhedron Example

Example of a Polytope and an Integer Hull

\[-3x_1 + 4x_2 = 4\]
\[3x_1 + 2x_2 = 11\]
\[2x_1 - x_2 = 5\]
Polyhedron Example

Example of a Polytope and an Integer Hull

\[-3x_1 + 4x_2 = 4\]
\[3x_1 + 2x_2 = 11\]
\[2x_1 - x_2 = 5\]
Polyhedron Example

Example of a Polytope and an Integer Hull

\[ -3x_1 + 4x_2 = 4 \]

\[ 3x_1 + 2x_2 = 11 \]

\[ 2x_1 - x_2 = 5 \]
Define Affine Loops

A special class of loop:
The lowerbound and upper bound of the loop are expressed as affine expressions of the surrounding loop variables and symbolic constants.

Given

\[
\begin{align*}
\text{do } & i_1 = L_1, U_1 \\
& \ldots \\
& \text{do } i_n = L_n, U_n \\
& S_1 : \ldots \\
& S_2 : \ldots
\end{align*}
\]

where \( L_m = f_m(i_1, i_2, \ldots, i_{m-1}) \) and \( f_m(\ldots) \) are all affine functions of variables \( (i_1, \ldots, i_{m-1}) \).
Modeling Affine Loop Iteration Domains

Corresponding loop constructs: unit stride, no conditionals

- Polytope dimension: number of surrounding loops
- Constraints: set by loop bounds
- Let’s all formulate it in this way: \( A \times x - b \geq 0 \)
- Add symbolic constants if necessary

Previous Loop Example

```c
for ( i = 0; i \leq 5; ++i )
for ( j = 0; j \leq 5; ++j )
    A[i][j] = i * j;
```

\[
D_R : \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 5 \\ 0 & 1 & 0 \\ 0 & -1 & 5 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} \geq \vec{0}
\]
Lexicographical Order of Affine Index

Given \( I = (i_1, i_2, ..., i_n) \) and \( I' = (i'_1, i'_2, ..., i'_n) \),
\( I < I' \) iff
\[ (i_1, i_2, ..., i_k) = (i'_1, i'_2, ..., i'_k) \) \& \( i_{k+1} < i'_{k+1} \)

An Example of Loop

for ( \( i = 0; \ i < 3; \ ++i \) )
    for ( \( j = 0; \ j < 3; \ ++j \) )
        \( A[i][j] = i \times j; \)

Loop execution:

(0,0)
(0,1)
(0,2)
(1,0)
(1,1)
(1,2)
...

...
Another Iteration Space Example

for ( i = 0; i ≤ 4; ++i )
    for ( j = i + 1; j ≤ 10 - i; ++j )
        A[i][j] = i * j;

Inequalities set by loop bounds:
i ≥ 0
i ≤ 4 → −i ≥ −4
j ≥ i + 1
j ≤ 10 − i → −j ≥ −10 + i

\[ D_R : \begin{bmatrix}
1 & 0 \\
-1 & 0 \\
-1 & 1 \\
-1 & -1 \\
\end{bmatrix}
\begin{bmatrix}
i \\
j \\
\end{bmatrix}
+ \begin{bmatrix}
0 \\
4 \\
-1 \\
10 \\
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
-1 & 0 & 4 \\
-1 & 1 & -1 \\
1 & -1 & 10 \\
\end{bmatrix}
\begin{bmatrix}
i \\
j \\
1 \\
\end{bmatrix}
\geq \vec{0} \]
The loop iteration space in graph:

Inequalities set by loop bounds:

\[ i \geq 0 \]
\[ i \leq 4 \]
\[ j \geq i + 1 \]
\[ j \leq 10 - i \]
Scan the Loop Iteration Space

Loop code generation:

- Using the lexicographical order:
- Assume no dependence: what if we want to sweep the space using another order?

Scan the iteration space from loop $i$ first.
What will be the loop bounds?
Fourier-Motzkin elimination:

Input: A polyhedron $S$ with variables $x_1, x_2, x_3, \ldots, x_n$.
Output: A polyhedron $S'$ with variables $x_1, x_2, \ldots, x_{m-1}, x_{m+1}, \ldots, x_n$

- Generally speaking, find a projection onto dimensions other than $x_m$
  - Take all pairs of inequalities with opposite sign coefficients of $x_m$, and for each generate a new valid inequality that eliminates $x_m$
  - Also take all inequalities from the original set which do not depend on $x_m$

**Fourier-Motzkin Theorem:**
The above collection of inequalities defines exactly the projection of $S$ onto $x_m = 0$. 
Assignment (2) I

More examples:

Example I

\[
\begin{align*}
&\text{for ( } i = 10; i \leq 1000; ++i \text{ )} \\
&\text{for ( } j = i; j \leq i + 10; ++j \text{ )} \\
&\hspace{10pt} A[i,j] = 0;
\end{align*}
\]

Example II

\[
\begin{align*}
&\text{for ( } i = 0; i \leq 100; ++i \text{ )} \\
&\text{for ( } j = 0; j \leq i+100; ++j \text{ )} \\
&\text{for ( } k = i+j; k \leq 100 - i - j; k++ \text{ )} \\
&\hspace{10pt} X[i,j,k] = 0;
\end{align*}
\]
for ( i = 0; i < N; ++i )
  for ( j = 0; j < i; ++j )
    if ( i > M )
      A[j] = 0;
More complicated loops I

- Unbounded domains: use polyhedra!
- Parametric loop bounds: use parametric polyhedra!
- Non-unit loop bounds: normalize the loop!
- Conditionals:
  - Those which preserve convexity
  - Problem remain for those which do not preserve convexity
Examples

Example I

for ( i = 0; i < N; i += 2 )
  for ( j = 0; j < N; ++j )
    A[i] = 0;

Example II

for ( i = 0; i < N; i += 2 )
  for ( j = 0; j < N; ++j )
    if ( i % 3 == 1 && j % 5 == 0 )
      A[i] = 0;
Generalized Conditionals

Different categories:

- Conjunctions ($a \&\& b$)
- Disjunctions ($a |\| b$)
- Non-affine ($i \times j < 2$)
- Data-dependent ($a[i] == 0$)
Relation with Operations on Polyhedra

**Conjunction:**
Definition: intersection
The intersection of two convex sets $\mathcal{P}_1$ and $\mathcal{P}_2$ is a convex set $\mathcal{P}$.

$$\mathcal{P} = \{ \bar{x} \in \mathbb{K}^m | \bar{x} \in \mathcal{P}_1 \land \bar{x} \in \mathcal{P}_2 \}$$

**Disjunction:**
Definition: union
The union of two convex sets $\mathcal{P}_1$ and $\mathcal{P}_2$ is a set $\mathcal{P}$:

$$\mathcal{P} = \{ \bar{x} \in \mathbb{K}^m | \bar{x} \in \mathcal{P}_1 \lor \bar{x} \in \mathcal{P}_2 \}$$

The union of two convex sets may not be a convex set.