PLUTO+: Near-Complete modeling of Affine Transformations for Parallelism and Locality

Aravind Acharya and Uday Bondhugula

Department of Computer Science and Automation
Indian Institute of Science
Bengaluru, India

09 Feb 2015

Outline

- Examples of affine functions of i, j: $i+j$, $i-j$, $i+1$, $2i+5$
- Not affine: ij, i^2, i^2+j^2, $a[j]$

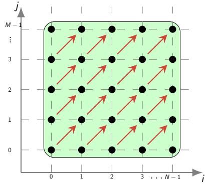


Figure: Iteration space

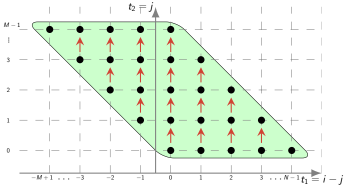


Figure: Transformed space

```
for (i = 0; i < N; i++){
    for (j = 0; j < M; j++){
          A[i+1][j+1] = f(A[i][j]);
    }
}
```

```
#pragma omp parallel for private(t2)
 for (t1=-M+1;t1<=N-1;t1++) {
   for(t2=max(0,-t1);t2<=min(M-1,N-1-t1);t2++){
           A[t1+t2+1][t2+1]=f(A[t1+t2][t2]);
       }
   }
```

- Transformation: $(i,j) \rightarrow (i-j, j)$

- Examples of affine functions of i, j: $i+j$, $i-j$, $i+1$, $2i+5$
- Not affine: ij, i^2, i^2+j^2, $a[j]$

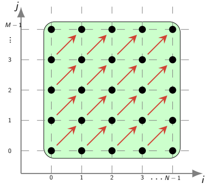


Figure: Iteration space

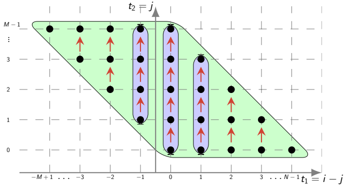


Figure: Transformed space

```
for (i = 0; i < N; i++){
    for (j = 0; j < M; j++){
          A[i+1][j+1] = f(A[i][j]);
    }
}
```

```
#pragma omp parallel for private(t2)
 for (t1=-M+1;t1<=N-1;t1++) {
   for(t2=max(0,-t1);t2<=min(M-1,N-1-t1);t2++){
            A[t1+t2+1][t2+1]=f(A[t1+t2][t2]);
        }
    }
```

- Transformation: $(i,j) \rightarrow (i-j,j)$

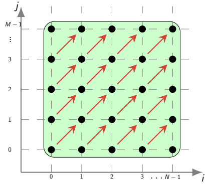


FIGURE: Iteration space

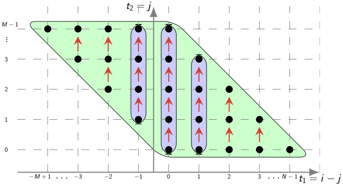


FIGURE: Transformed space

- Affine transformations are attractive because:
  - Preserve **collinearity** of points and **ratio of distances** between points
  - Code generation with affine transformations has thus been studied well (CLooG, ISL, OMEGA+)

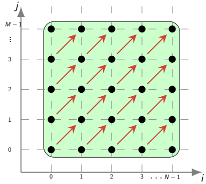


Figure: Iteration space

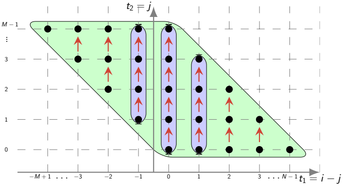


Figure: Transformed space

- Affine transformations are attractive because:
  - Preserve **collinearity** of points and **ratio of distances** between points
  - Code generation with affine transformations has thus been studied well (CLooG, ISL, OMEGA+)
  - Model a very rich class of loop re-orderings
  - Useful for several domains like dense linear algebra, stencils, image processing pipelines, Lattice Boltzmann Method

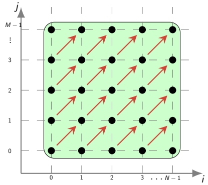


Figure: Iteration space

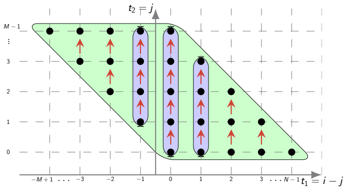


Figure: Transformed space

- Can express complex compositions of simpler transformations like permutation, skewing, reversal, scaling, shifting, tiling, fusion, distribution
- Affine transformations can improve **parallelism** and **locality** (Feautrier 1992, Lim and Lam 1997, Pluto 2008)

Pluto Algorithm

- Proposed by Bondhugula et al. CC 2008, PLDI 2008
- Improves locality and parallelism by minimizing dependence distance

- Proposed by Bondhugula et al. CC 2008, PLDI 2008
- Improves locality and parallelism by minimizing dependence distance
- Assumption: Transformation coefficients are **non-negative**

- Proposed by Bondhugula et al. CC 2008, PLDI 2008
- Improves locality and parallelism by minimizing dependence distance
- Assumption: Transformation coefficients are **non-negative**
- Optimization Problem: **Minimize dependence distance**

- Proposed by Bondhugula et al. CC 2008, PLDI 2008
- Improves locality and parallelism by minimizing dependence distance
- Assumption: Transformation coefficients are **non-negative**
- Optimization Problem: **Minimize dependence distance**
- Constraints:
  - Tile validity constraints
  - Dependence distance bounding constraints
  - Zero solution avoiding constraints
  - Linear Independence constraints

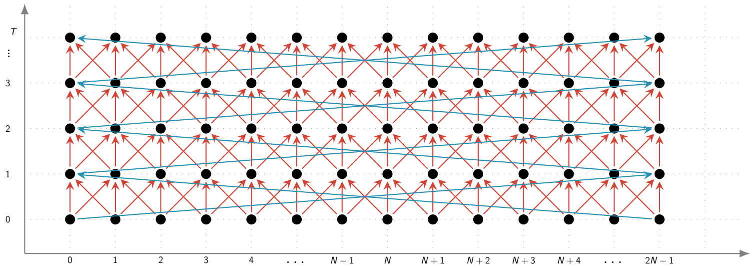
- Proposed by Bondhugula et al. CC 2008, PLDI 2008
- Improves locality and parallelism by minimizing dependence distance
- Assumption: Transformation coefficients are **non-negative**
- Optimization Problem: **Minimize dependence distance**
- Constraints:
  - Tile validity constraints
  - Dependence distance bounding constraints
  - Zero solution avoiding constraints
  - Linear Independence constraints

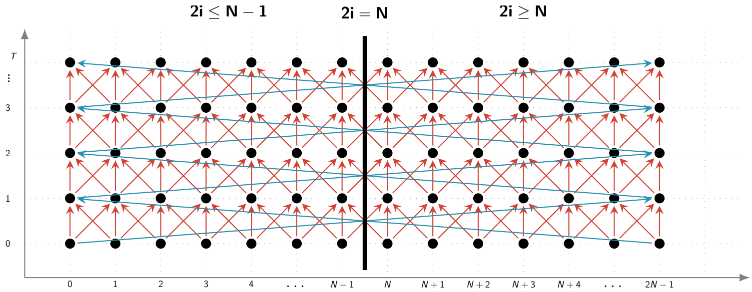
When transformation coefficients are negative, the above constraints miss useful solutions

Motivation: Periodic Stencils

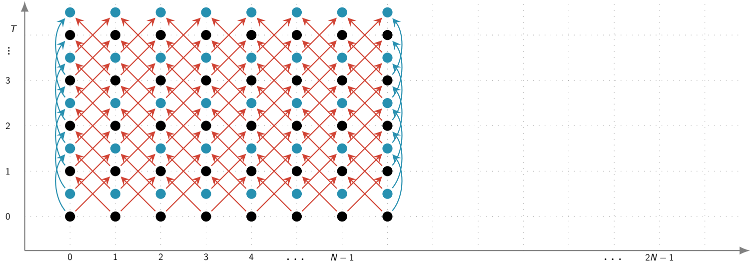
- Near-neighbor dependences and some long wraparound dependences
- Applications in fluid simulations of infinite domains
- Periodic Lattice Boltzmann Methods (LBM) used in fluid dynamics, Swim (shallow water equations) fall into this category

Tiling Periodic Domains

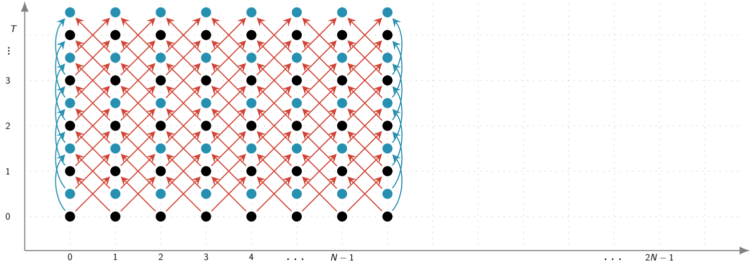




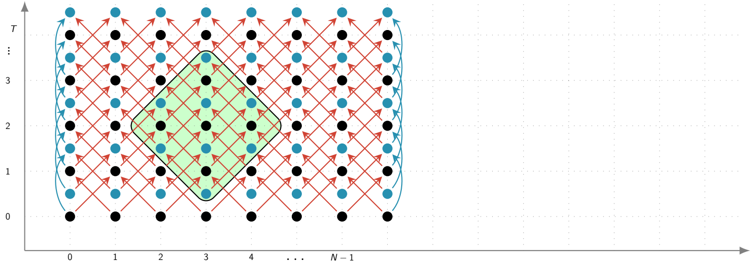
- Find a cut close to the mid point: $(2i = N)$



- Find a cut close to the mid point: $(2i = N)$
- Reverse the second domain and shift it right by N: $(t, i) \rightarrow (t, N - i)$



- Find a cut close to the mid point: $(2i = N)$
- Reverse the second domain and shift it right by N: $(t, i) \rightarrow (t, N - i)$
- Now all dependences are **short**



- Find a cut close to the mid point: $(2i = N)$
- Reverse the second domain and shift it right by N: $(t, i) \to (t, N - i)$
- Now all dependences are **short**
- Tile the time dimension (parallelogram, diamond)

Transformation coefficients are **non-negative integers**.

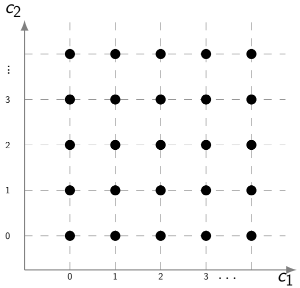


FIGURE: Search space of transformation coefficients

Transformation coefficients are **non-negative integers**.

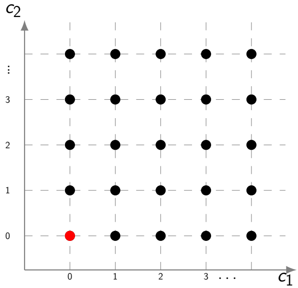


FIGURE: Search space of transformation coefficients

$$c_1, c_2 \geq 0,$$
$$c_1 + c_2 \geq 1.$$

Transformation coefficients are **integers**.

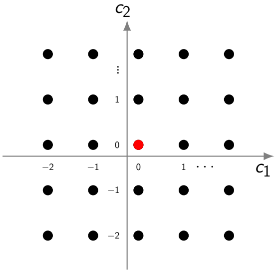


Figure: Search space of transformation coefficients

$$c_1, c_2 \geq 0,$$
$$c_1 + c_2 \geq 1.$$

With coefficients being **negative**, we may miss valid solutions. Eg: $c_1 = 1, c_2 = -1$

Transformation coefficients are **integers**.

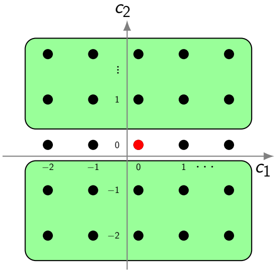


Figure: Search space of transformation coefficients

$$c_1 \neq 0 \vee c_2 \neq 0.$$

Transformation coefficients are **integers**.

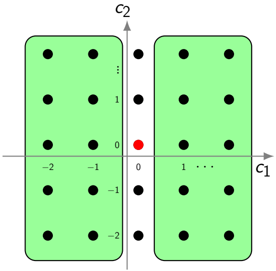


Figure: Search space of transformation coefficients

$$c_1 \neq 0 \vee c_2 \neq 0.$$

Transformation coefficients are **integers**.

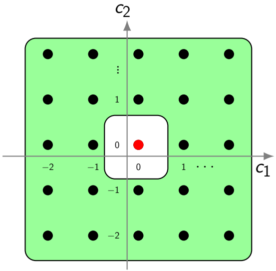
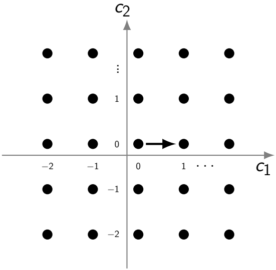


FIGURE: Search space of transformation coefficients

$$c_1 \neq 0 \vee c_2 \neq 0.$$

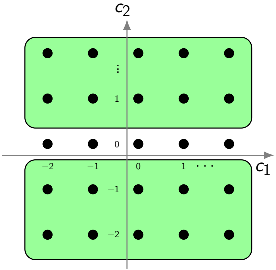
This constraint results in a **non convex space**. Approach does not scale.

Challenge 2: Finding Linearly independent transformations



- Let $(0, 1)$ be the first hyperplane

Challenge 2: Finding Linearly independent transformations



- Let $(0, 1)$ be the first hyperplane
- For the next hyperplane to be linearly independent: $c_2 \neq 0$

Outline

PLUTO+: Avoiding the Zero Solution

Assume that c_0, c_1, c_2 are bounded by -4 and $+4$.

- c_0, c_1, c_2 can be considered to be in base 5
- If $(c_0, c_1, c_2) = \vec{0}$, then $5^2 c_2 + 5c_1 + c_0 = 0$ and vice versa

PLUTO+: Avoiding the Zero Solution

Assume that c_0, c_1, c_2 are bounded by -4 and $+4$.

- c_0, c_1, c_2 can be considered to be in base 5
- If $(c_0, c_1, c_2) = \vec{0}$, then $5^2 c_2 + 5c_1 + c_0 = 0$ and vice versa
- To avoid the zero solution,
  $(c_0, c_1, c_2) \neq \vec{0} \iff |\mathbf{5^2 c_2 + 5c_1 + c_0}| \geq \mathbf{1}$. Make this constraint convex by using a **decision variable**.

Assume that c_0, c_1, c_2 are bounded by -4 and +4.

- c_0, c_1, c_2 can be considered to be in base 5
- If $(c_0, c_1, c_2) = \vec{0}$, then $5^2c_2 + 5c_1 + c_0 = 0$ and vice versa
- To avoid the zero solution, $(c_0, c_1, c_2) \neq \vec{0} \iff |\mathbf{5^2c_2 + 5c_1 + c_0}| \geq \mathbf{1}$. Make this constraint convex by using a **decision variable**.
- Since the coefficients are in base 5,
  the maximum value of $5^2c_2 + 5c_1 + c_0$ is $5^3 - 1 = 124$.
  the minimum value of $5^2c_2 + 5c_1 + c_0$ is $1 - 5^3 = -124$.
- Hence, upper and lower bounds for $5^2c_2 + 5c_1 + c_0$ are known

- Introduce a decision variable to obtain a convex space representing the constraint on the absolute value
- We then have:

  $c_0, c_1, c_2 \neq 0 \iff |25c_2 + 5c_1 + c_0| \geq 1.$

PLUTO+: Avoiding the Zero Solution

- Introduce a decision variable to obtain a convex space representing the constraint on the absolute value
- We then have:

  $c_0, c_1, c_2 \neq 0 \iff |25c_2 + 5c_1 + c_0| \geq 1.$

Consider the following expressions where $\delta \in \{\mathbf{0}, \mathbf{1}\}$ is a decision variable:

$$5^2 c_2 + 5c_1 + c_0 \geq 1 - \delta * 5^3,$$
$$-(5^2 c_2 + 5c_1 + c_0) \geq 1 - (1 - \delta) * 5^3.$$

- Introduce a decision variable to obtain a convex space representing the constraint on the absolute value
- We then have:

  $c_0, c_1, c_2 \neq 0 \iff |25c_2 + 5c_1 + c_0| \geq 1.$

Consider the following expressions where $\delta \in \{\mathbf{0}, \mathbf{1}\}$ is a decision variable:

$$5^2c_2 + 5c_1 + c_0 \geq 1 - \delta * 5^3,$$
$$-(5^2c_2 + 5c_1 + c_0) \geq 1 - (1 - \delta) * 5^3.$$

If $\delta = \mathbf{0}$ then,

$$5^2c_2 + 5c_1 + c_0 \geq 1.$$
$$5^2c_2 + 5c_1 + c_0 \leq 5^3 - 1.$$

Positive half-space

- Introduce a decision variable to obtain a convex space representing the constraint on the absolute value
- We then have:

  $c_0, c_1, c_2 \neq 0 \iff |25c_2 + 5c_1 + c_0| \geq 1.$

Consider the following expressions where $\delta \in \{\mathbf{0}, \mathbf{1}\}$ is a decision variable:

$$5^2 c_2 + 5c_1 + c_0 \geq 1 - \delta * 5^3,$$
$$-(5^2 c_2 + 5c_1 + c_0) \geq 1 - (1 - \delta) * 5^3.$$

If $\delta = \mathbf{0}$ then,

$$5^2 c_2 + 5c_1 + c_0 \geq 1.$$
$$5^2 c_2 + 5c_1 + c_0 \leq 5^3 - 1.$$

Positive half-space

If $\delta = \mathbf{1}$ then,

$$5^2 c_2 + 5c_1 + c_0 \geq 1 - 5^3.$$
$$5^2 c_2 + 5c_1 + c_0 \leq -1.$$

Negative half-space

- The hyperplanes for every statement have to be linearly independent

- The hyperplanes for every statement have to be linearly independent
- Assume the hyperplane that has been found is

$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

Linear independence is given by

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \neq \vec{0}$$

PLUTO+: Linear Independence

- The hyperplanes for every statement have to be linearly independent
- Assume the hyperplane that has been found is

$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

  Linear independence is given by

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \neq \vec{0}$$

- Newly found hyperplane must have a non-zero component in the orthogonal subspace

- The hyperplanes for every statement have to be linearly independent
- Assume the hyperplane that has been found is

$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

Linear independence is given by

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \neq \vec{0}$$

- Newly found hyperplane must have a non-zero component in the orthogonal subspace

$$|\mathbf{c_1} - \mathbf{c_2}| + |\mathbf{c_3}| \geq \mathbf{1}$$

PLUTO+: Linear Independence

- The hyperplanes for every statement have to be linearly independent
- Assume the hyperplane that has been found is

$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

Linear independence is given by

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \neq \vec{0}$$

- Newly found hyperplane must have a non-zero component in the orthogonal subspace

$$|\mathbf{c_1} - \mathbf{c_2}| + |\mathbf{c_3}| \geq \mathbf{1}$$

- Get a convex space corresponding to the absolute value using a decision variable.

Outline

- Implemented as PLUTO+
- The optimization problem is solved using an open-source ILP solver (ISL, GLPK)

- Implemented as PLUTO+
- The optimization problem is solved using an open-source ILP solver (ISL, GLPK)
- Eg: Transformation obtained for LBM D2Q9 (periodic)

Original schedule	Transformed schedule
S1: (t, i, j)	S1: (t-i, t+i, t+j) S2: (t+i-N, t-i+N, t+j) S3: (t-i, t+i, t-j+N) S4: (t+i-N, t-i+N, t-j+N)

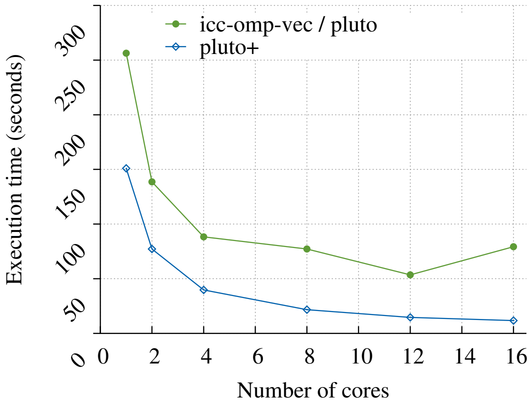
- Performance evaluation: Heat equation benchmarks with periodic conditions from Pochoir, Swim from SPEC 2000fp, several Lattice Boltzmann Method (LBM) simulations
- Comparison with Intel C compiler, Palabos in addition for LBM

- Performance evaluation: Heat equation benchmarks with periodic conditions from Pochoir, Swim from SPEC 2000fp, several Lattice Boltzmann Method (LBM) simulations
- Comparison with Intel C compiler, Palabos in addition for LBM
- Analysis of impact on polyhedral automatic transformation time and overall compilation time

EXPERIMENTAL SETUP

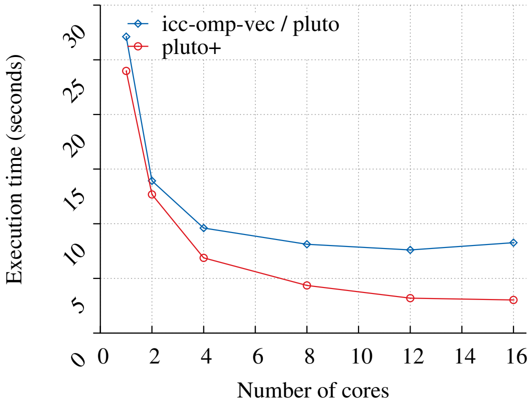
- Codes were run on Intel SandyBridge Machine with the following configuration.

	Intel Xeon E5-2680
Clock	2.7 GHz
Cores / socket	8
Total cores	16
L1 cache / core	32 KB
L2 cache / core	512 KB
L3 cache / socket	20 MB
Peak GFLOPs	172.8
Compiler	Intel C compiler (icc) 14.0.1
Compiler flags	-O3 -xHost -ipo
	-restrict -fno-alias -ansi-alias
	-fp-model precise -fast-transcendentals
Linux kernel	3.8.0-44



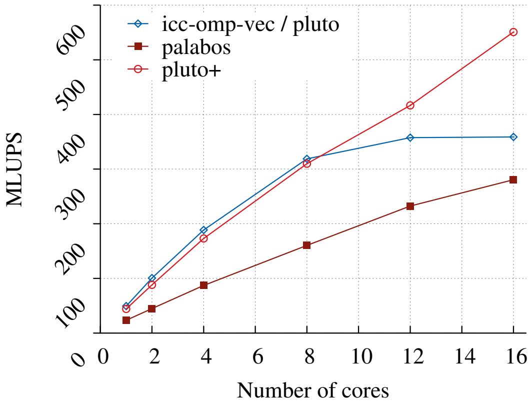
- Speedup of 1.7× on single core and 6.7× on 16 cores against icc

Swim Benchmark (Spec 2000fp)



- Speedup of 2.73× over on 16 cores icc

Performance: LBM D2Q9



- Speedup of 1.5× over icc and 1.9× over Palabos

Compile Times

- Comparison of Pluto+ with Pluto ($\frac{\text{Pluto+ time}}{\text{Pluto time}}$)

Benchmark	Auto-transformation	Total Time
Polybench	0.89	1.15
Heat equation	0.39	2.25
Swim (SPEC2000fp)	9.71	2.83
LBM benchmarks	0.49	1.80

- Pluto+ scales very well
- Improvement in auto-transformation time in several cases due to bounds on transformation coefficients
- In most cases, the increase in compile time was due to an increase in code generation time
- Total compilation time varied from 0.013s (jacobi-1d-imper) to 56.36s (LBM D3Q27)

Benchmarks	Increase in compilation time	Speedup in running time
Heat equation	2.25×	2.91×
Swim	2.83×	2.73×
LBM benchmarks (mean)	1.8×	1.33×

- Enables **time tiling** in the presence of periodic boundary conditions

Performance Summary

Benchmarks	Increase in compilation time	Speedup in running time
Heat equation	2.25×	2.91×
Swim	2.83×	2.73×
LBM benchmarks (mean)	1.8×	1.33×

- Enables **time tiling** in the presence of periodic boundary conditions
- Scales like Pluto

Benchmarks	Increase in compilation time	Speedup in running time
Heat equation	2.25×	2.91×
Swim	2.83×	2.73×
LBM benchmarks (mean)	1.8×	1.33×

- Enables **time tiling** in the presence of periodic boundary conditions
- Scales like Pluto
- **Availability:** Code and benchmarks available at `http://mcl.csa.iisc.ernet.in/`

- Griebl [Habilitation thesis 2008]: Forward Communication Only (FCO) constraints. Finds extremal solutions of the space of valid transformations. Expensive, practicality and scalability not demonstrated

Related Work

- Griebl [Habilitation thesis 2008]: Forward Communication Only (FCO) constraints. Finds extremal solutions of the space of valid transformations. Expensive, practicality and scalability not demonstrated
- Lim and Lam [POPL'97 and ICS'99]: Objective is to reduce frequency of synchronization and improve locality. No specific objective function to choose among valid solutions

- Griebl [Habilitation thesis 2008]: Forward Communication Only (FCO) constraints. Finds extremal solutions of the space of valid transformations. Expensive, practicality and scalability not demonstrated
- Lim and Lam [POPL'97 and ICS'99]: Objective is to reduce frequency of synchronization and improve locality. No specific objective function to choose among valid solutions
- R-STREAM compiler [Encl. of Par. Computing 2011]: larger number of decision variables per statement.

Summary

- Presented PLUTO+, models a significantly larger space of affine transformations in the presence of a cost function, complete in practice

Summary

- Presented PLUTO+, models a significantly larger space of affine transformations in the presence of a cost function, complete in practice
- Increase in overall compilation time by only 15%

Summary

- Presented PLUTO+, models a significantly larger space of affine transformations in the presence of a cost function, complete in practice
- Increase in overall compilation time by only 15%
- Faster than Pluto in auto-transformation time on average

Summary

- Presented PLUTO+, models a significantly larger space of affine transformations in the presence of a cost function, complete in practice
- Increase in overall compilation time by only 15%
- Faster than Pluto in auto-transformation time on average
- Speedup of 2.73× over icc on Swim benchmark

Summary

- Presented PLUTO+, models a significantly larger space of affine transformations in the presence of a cost function, complete in practice
- Increase in overall compilation time by only 15%
- Faster than Pluto in auto-transformation time on average
- Speedup of 2.73× over icc on Swim benchmark
- Speedup of 1.33× for Lattice Boltzmann Method computations

- Presented PLUTO+, models a significantly larger space of affine transformations in the presence of a cost function, complete in practice
- Increase in overall compilation time by only 15%
- Faster than Pluto in auto-transformation time on average
- Speedup of 2.73× over icc on Swim benchmark
- Speedup of 1.33× for Lattice Boltzmann Method computations

Acknowledgments

- PPoPP'15 reviewers for their comments
- Albert Cohen, INRIA, for discussions
- Intel Labs, India, for equipment and software used for experimentation
- Microsoft Research, India and ACM SIGPLAN for travel grants