Lecture 10
Loop Transformations for Communication-Free Parallelism

1. Examples of Loop Transformations
2. Affine Partitioning
3. Code Generation

Readings: Chapter 11-11.3, 11.6-11.7.4, 11.9-11.9.6
1. Parallelism and Locality

- Parallelism DOES NOT imply speed up!

- Parallel performance:
  - Improve locality with loop transformations
  - Minimize communication
  - Operations using the same data are executed on the same processor

- Sequential performance:
  - Improve locality with loop transformations
  - Minimize cache misses
  - Operations using the same data are executed close in time.

---

Loop Permutation (Loop Interchange)

\[
\begin{align*}
\text{for } I &= 1 \text{ to } 4 \\
\text{for } J &= 1 \text{ to } 3 \\
Z[I,J] &= Z[I-1,J]
\end{align*}
\]

\[
\begin{align*}
\begin{bmatrix} I' \\ J' \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} I \\ J \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\text{for } J' &= 1 \text{ to } 3 \\
\text{for } I' &= 1 \text{ to } 4 \\
Z[I',J'] &= Z[I'-1,J']
\end{align*}
\]
Loop Fusion

for $I = 1$ to $4$
T[$I$] = A[$I$] + B[$I$]  \hspace{1cm} \text{(s1)}
for $I' = 1$ to $4$
C[$I'$] = T[$I'$] $\times$ T[$I'$]  \hspace{1cm} \text{(s2)}

for $J = 1$ to $4$
T[$J$] = A[$J$] + B[$J$]  \hspace{1cm} \text{(s1)}
C[$J$] = T[$J$] $\times$ T[$J$]  \hspace{1cm} \text{(s2)}

M. Lam
CS243: Loop Transformations

Loop Transformations

• Unimodular transforms on loop nests
  – Interchange
  – Skewing
  – Reversal
• Cross statement transforms
  – Loop fusion
  – Loop fission
  – Re-indexing
• How to combine them to get parallelism and locality?
2. Affine Partitioning: 
A Contrived but Illustrative Example

FOR j = 1 TO n
    FOR i = 1 TO n
        A[i,j] = A[i,j]+B[i-1,j]; \(S_1\)
        B[i,j] = A[i,j-1]*B[i,j]; \(S_2\)

Best Parallelization Scheme

Algorithm finds affine partition mappings for each instruction:

S1: Execute iteration \((i, j)\) on processor \(i-j\).
S2: Execute iteration \((i, j)\) on processor \(i-j+1\).

SPMD code: Let \(p\) be the processor's ID number

\[
\text{if } (1-n \leq p \leq n) \text{ then}
\]
\[
\text{if } [1 \leq p) \text{ then}
\]
\[
B[p,1] = A[p,0] * B[p,1]; \quad (S_2)
\]
\[
\text{for } i_1 = \max[1,1+p) \text{ to } \min[n,n-1+p) \text{ do}
\]
\[
A[i_1,i_1-p] = A[i_1,i_1-p] + B[i_1-1,i_1-p]; \quad (S_1)
\]
\[
B[i_1,i_1-p+1] = A[i_1,i_1-p] * B[i_1,i_1-p+1]; \quad (S_2)
\]
\[
\text{if } (p \leq 0) \text{ then}
\]
\[
A[n+p,n] = A[n+p,N] + B[n+p-1,n]; \quad (S_1)
\]
Iteration Space

\[\text{FOR } i = 0 \text{ to } 5\]
\[\text{FOR } j = i \text{ to } 7\]

- \(n\)-deep loop nests: \(n\)-dimensional polytope
- Iterations: coordinates in the iteration space
- Assume: iteration index is incremented in the loop
- Sequential execution order: lexicographic order
  - \([0,0], [0,1], \ldots, [0,6], [0,7], [1,1], \ldots, [1,6], [1,7], \ldots\)

Maximum Parallelism & No Communication

For every pair of data dependent accesses \(F_{1 \text{l}1}+f_1\) and \(F_{j \text{l}2}+f_2\)

Find \(C_{j\text{r}}, C_{\text{l}1}, C_{2\text{r}}, C_{2\text{l}}\):

\[\forall \text{l}_1, \text{l}_2 \quad F_{1 \text{l}1}+f_1 = F_{2 \text{l}2}+f_2 \rightarrow C_{\text{l}1}+C_{\text{l}1} = C_{2\text{l}2}+C_{2\text{l}}\]

with the objective of maximizing the rank of \(C_{j\text{r}}, C_{2\text{r}}\)
**Rank of Partitioning = Degree of Parallelism**

Affine Mapping

\[
\begin{bmatrix}
0 & 0 \\
1 & 0 \\
0 & 1
\end{bmatrix}
\]

Rank

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example 1: Loop Transform

Find affine partitioning: \( c_1, c_2, c_0 \) such that

\[
p = \begin{bmatrix}
c_1 & c_2
\end{bmatrix}
\begin{bmatrix}
i \\
j
\end{bmatrix} + c_0
\]

Suppose iteration \( i, j \) & \( i', j' \) refer to same location

\[
i = i' - 1 \\
j = j'
\]

No communication means:

\[
c_1 i + c_2 j + c_0 = c_1 i' + c_2 j' + c_0
\]

\[
c_1 (i' - 1) + c_2 j' + c_0 = c_1 i' + c_2 j' + c_0
\]

\[
c_1 = 0
\]

\[
p = c_2 j + c_0
\]

Pick simplest \( c_2, c_0 \): \( c_2 = 1, c_0 = 0 \)

\[
p = j
\]
3. Code Generation

- Naive
  - Each processor visits all the iterations
  - Executes only if it owns that iteration
- Optimization
  - Removes unnecessary looping and condition evaluation

Code Generation

for I = 1 to 4
for J = 1 to 3
Z[I,J] = Z[I-1,J]

for P = 1 to 3
for I = 1 to 4
for J = 1 to 3
if (j == P)
  Z[I,J] = Z[I-1,J]

for P = 1 to 3
for I = 1 to 4
Z[I,P] = Z[I-1,P]

SPMD (single program multiple data) code:
for I = 1 to 4
Z[I,P] = Z[I-1,P]
**Loop Permutation (Loop Interchange)**

for $I = 1$ to $4$
for $J = 1$ to $3$
$Z[I, J] = Z[I-1, J]$

for $P = 1$ to $3$
for $I = 1$ to $4$
$Z[I, P] = Z[I-1, P]$

Example 2: Loop Fusion

Find affine partitioning: $c_{1,1}, c_{1,0}, c_{2,1}, c_{2,0}$ such that

s1: $[p] = \begin{bmatrix} c_{1,1} \\ i \end{bmatrix} [i] + c_{1,0}$

s2: $[p] = \begin{bmatrix} c_{2,1} \\ i' \end{bmatrix} [i'] + c_{2,0}$

Suppose iteration $i$ & $i'$ refer to the same location
$i = i'$

No communication means:
$c_{1,1} i + c_{1,0} = c_{2,1} i' + c_{2,0}$

$c_{1,1} = c_{2,1}$
$c_{1,0} = c_{2,0}$

Pick simplest values: $c_{1,1} = c_{2,1} = 1$, $c_{1,0} = c_{2,0} = 0$

$p = i$; $p = i'$
Loop Fusion

for I = 1 to 4
T[I] = A[I] + B[I] \hspace{1cm} (s1)

for I' = 1 to 4
C[I'] = T[I'] x T[I'] \hspace{1cm} (s2)

Example 3: 2 Nested, Parallel Loops

Find affine partitioning: \( c_1, c_2, c_0 \) such that

\[
p = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + c_0
\]

Suppose iteration \( i, j \) & \( i', j' \) refer to same location \( i = i' \) \quad \text{and} \quad j = j' \)

No communication means:

\[
\begin{align*}
&c_1 i + c_2 j + c_0 = c_1 i' + c_2 j' + c_0 \\
&c_1 i' + c_2 j' + c_0 = c_1 i + c_2 j + c_0
\end{align*}
\]

No constraints

Two basis vectors: \( [c_1, c_2]=[1 \ 0] \), or \( [c_1, c_2]=[0 \ 1] \)

Two answers for \( p \): two degrees of parallelism

\[
\begin{bmatrix}
p_1 \\
p_2
\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix}
\]
Example 3: 2 Nested, Parallel Loops

\[ \text{for } I = 1 \text{ to } 4 \]
\[ \text{for } J = 1 \text{ to } 3 \]
\[ Z[I,J] = Z[I,J]+1 \]

\[ \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} =\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} \]

for \( p_1 = 1 \text{ to } 4 \)
for \( p_2 = 1 \text{ to } 3 \)
if \( (I==p1 \& J == p2) \)
\[ Z[I,J] = Z[I,J]+1 \]

for \( p_1 = 1 \text{ to } 4 \)
for \( p_2 = 1 \text{ to } 3 \)
\[ Z[p1,p2] = Z[p1,p2]+1 \]

Optimizing Arbitrary Loop Nesting Using Affine Partitions (chotst, NAS)

\[ \text{DO 1 } J = 0, N \]
\[ I_0 = \text{MAX} (-M, -J) \]
\[ \text{DO 2 } I = I_0, -1 \]
\[ \text{DO 3 JJ = } I_0 - I, -1 \]
\[ A(L,J,J) = A(L,J,J) - A(L,J,J+1) * A(L,J,J) \]
\[ A(L,J,J) = A(L,J,J) * A(L,0,I+J) \]
\[ A(L,0,J) = A(L,0,J) - A(L,0,J) ** 2 \]
\[ A(L,0,J) = 1. / \text{SQRT} (\text{ABS} (\text{EPSS}(L) + A(L,0,J))) \]
\[ \text{DO 6 } I = 0, \text{NRHS} \]
\[ \text{DO 7 } K = 0, N \]
\[ \text{DO 8 L = 0, N} \]
\[ B(I,L,K) = B(I,L,K) - A(L,0,K) \]
\[ \text{DO 9 L = 0, N} \]
\[ B(I,L,K) = B(I,L,K) * A(L,0,K) \]
\[ \text{DO 6 } K = N, 0, -1 \]
\[ \text{DO 7 } L = 0, N \]
\[ B(I,L,K+JJ) = B(I,L,K+JJ) - A(L,-JJ,K+JJ) * B(I,L,K) \]
\[ B(I,L,K) = B(I,L,K) * A(L,0,K) \]
\[ \text{DO 6 } J = 1, \text{MIN}(M, N-K) \]
\[ B(I,L,K-JJ) = B(I,L,K-JJ) - A(L,-JJ,K) * B(I,L,K) \]
Chotst: Results with Affine Partitioning + Blocking

(Unimodular: a subset of affine partitioning for perfect loop nests)

Maximum Parallelism & No Communication

For every pair of data dependent accesses $F_{1i1}+f_1$ and $F_{j2}+f_2$:

Find $C_{j1}, C_{i1}, C_{21}, C_{22}$:

$\forall i_1, i_2 \ F_{1i1}+f_1 = F_{j2}+f_2 \rightarrow C_{i1} = C_{j2}+C_{2}$

with the objective of maximizing the rank of $C_{j1}, C_{22}$