Loop Transformations for Parallelism & Locality

Last time
- Unimodular transformation framework
- Loop permutation
- Loop reversal
- Loop skewing

Today
- Kelly & Pugh transformation framework
- Loop transformations
  - Loop fusion
  - Loop fission

What are the dependences?
\[
\begin{align*}
\text{do } i = 1, n & \\
\text{s}_1 & \quad A(i) = B(i) + 1 \\
\text{enddo} & \\
\text{do } i = 1, n & \\
\text{s}_2 & \quad C(i) = A(i)/2 \\
\text{enddo} & \\
\text{do } i = 1, n & \\
\text{s}_3 & \quad D(i) = 1/C(i+1) \\
\text{enddo}
\end{align*}
\]

Kelly and Pugh Transformation Framework

Specify iteration space as a set of integer tuples
\[\{(i,j) \mid 1 \leq i, j \leq n\}\]

Specify data dependences as mappings between integer tuples (i.e., data dependence relations)
\[\{\{i,j\} \rightarrow \{i',j'\} \mid (i = i' - 1) \land (j = j' - 1) \land (1 \leq i, j, i', j' \leq n)\}\]

Specify transformations as mappings between integer tuples
\[\{\{i,j\} \rightarrow \{i',j'\} \mid (i' = j) \land (j' = i)\}\]

Execute iterations in transformed iteration space in lexicographic order

Loop Fusion Example

What are the dependences?
\[
\begin{align*}
\text{do } i = 1, n & \\
\text{s}_1 & \quad A(i) = B(i) + 1 \\
\text{enddo} & \\
\text{s}_2 & \quad C(i) = A(i)/2 \\
\text{s}_3 & \quad D(i) = 1/C(i+1) \\
\text{enddo}
\end{align*}
\]

Fusion changes the dependence between \(s_2\) and \(s_3\), so fusion is illegal

Specifying Loop Fusion in Kelly and Pugh Framework

Specify iteration space as a set of integer tuples
\[IS_1 = \{(i_1, i_2) \mid 1 \leq i_1 \leq n\}\]
\[IS_2 = \{(i_2, i_3) \mid 1 \leq i_2 \leq n\}\]
\[IS_3 = \{(i_3, i_4) \mid 1 \leq i_3 \leq n\}\]
\[IS = IS_1 \cup IS_2 \cup IS_3\]

Specify data dependences as mappings between integer tuples (i.e., data dependence relations)
\[D_{12} = \{\{1, i_1, 1\} \rightarrow \{2, i_2, 1\} \mid i_1 = i_2\}\]
\[D_{23} = \{\{2, i_2, 1\} \rightarrow \{3, i_3, 1\} \mid i_2 = i_3 + 1\}\]
\[D = D_{12} \cup D_{23}\]

Specify transformations as mappings between integer tuples
\[T_1 = \{\{1, i_1, 1\} \rightarrow \{1, i'_1, 1\} \mid i'_1 = i_1\}\]
\[T_2 = \{\{2, i_2, 1\} \rightarrow \{1, i'_2, 1\} \mid i'_2 = i_2\}\]
\[T_3 = \{\{3, i_3, 1\} \rightarrow \{1, i'_3, 3\} \mid i'_3 = i_3\}\]
\[T = T_1 \cup T_2 \cup T_3\]
Checking Legality in Kelly & Pugh Framework

For each dependence, \([I] \rightarrow [J]\) the transformed \(I\) iteration must be executed after the transformed \(J\) iteration.

Loop Fusion Example (cont)

Loop reversal is legal for the original loops
- Does not change the direction of any dep in the original code
- Will reverse the direction in the fused loop: \(s_1\) \(s_2\) \(s_3\) will become \(s_2\) \(s_1\) \(s_3\)

\[
\begin{align*}
do i = n, 1, -1 \\
s_1 & \quad A(i) = B(i) + 1 \\
enddo \\
s_2 & \quad C(i) = A(i)/2 \\
enddo \\
s_3 & \quad D(i) = 1/C(i+1) \\
enddo
\end{align*}
\]

After reversal and fusion all original dependences are preserved

Loop Fission (Loop Distribution)

Idea
- Split a loop nest into multiple loop nests (the inverse of fusion)

Example
\[
\begin{align*}
do i = 1, n \\
A(i) &= B(i) + 1 \\
C(i) &= A(i)/2 \\
enddo
\end{align*}
\]

Motivation?
- Produces multiple (potentially) less constrained loops
- May improve locality
- Enable other transformations, such as interchange

Legality?
- Fission is legal when the loop body contains no cycles in the dependence graph

Cycles cannot be preserved because after fission all cross-loop dependences flow from body1 to body2
Recall our fusion example

\[
\begin{align*}
do i = 1, n \\
& s_1 A(i) = B(i) + 1 \\
& \text{endo} \\
& s_2 C(i) = A(i)/2 \\
& \text{endo} \\
& s_3 D(i) = 1/C(i+1) \\
& \text{endo}
\end{align*}
\]

Can we perform fission on this loop?

\[
\begin{align*}
do i = 1, n \\
& s_1 A(i) = B(i) + 1 \\
& \text{endo} \\
& s_2 C(i) = A(i)/2 \\
& \text{endo} \\
& s_3 D(i) = 1/C(i+1) \\
& \text{endo}
\end{align*}
\]

If there are no cycles, we can reorder the loops with a topological sort

\[
\begin{align*}
do i = 1, n \\
& s_1 A(i) = B(i) + 1 \\
& \text{endo} \\
& s_2 C(i) = A(i)/2 \\
& \text{endo} \\
& s_3 D(i) = 1/C(i+1) \\
& \text{endo}
\end{align*}
\]

Can we perform fission on this loop?

Next Time

- Lecture
  - Tiling