CS 314 Principles of Programming Languages

Lecture 24

Zheng Zhang

Department of Computer Science
Rutgers University

Wednesday 7th December, 2016
Class Information

- Project 2 due today, 11:55pm EST.
- Project 3 and Homework 8 will be posted soon.
Lexicographical (sequential) order for the following iteration space:

\((1,1), (1,2), \ldots, (1,6)\)
\((2,2), (2,3), \ldots (2,6)\)
\[\ldots\]
\((5,5), (5,6)\)

\[
\begin{align*}
&\text{do } I = 1, 5 \\
&\quad \text{do } J = I, 6 \\
&\quad \quad \ldots \\
&\quad \quad \text{enddo} \\
&\quad \text{enddo} \\
&1 \leq I \leq 5 \\
&I \leq J \leq 6
\end{align*}
\]

Given \(l = (i_1, \ldots, i_n)\) and \(l' = (i'_1, \ldots, i'_n)\),

\(l < l'\) iff \((i_1, i_2, \ldots, i_k) = (i'_1, i'_2, \ldots, i'_k)\) \& \(i_{k+1} < i'_{k+1}\)
Review: Dependence Testing

Given
\[
\begin{align*}
\text{do } & i_1 = L_1, U_1 \\
\ldots &
\text{do } i_n = L_n, U_n \\
S_1 & A(f_1(i_1, \ldots, i_n), \ldots, f_m(i_1, \ldots, i_n)) = \ldots \\
S_2 & \ldots = A(g_1(i_1, \ldots, i_n), \ldots, g_m(i_1, \ldots, i_n))
\end{align*}
\]

A dependence between statement $S_1$ and $S_2$, denoted $S_1 \delta S_2$, indicates that $S_1$, the source, must be executed before $S_2$, the sink on some iteration of the nest.

Let $\alpha$ & $\beta$ be a vector of $n$ integers within the ranges of the lower and upper bounds of the $n$ loops.

Does $\exists \alpha \leq \beta$, s.t.

\[
f_k(\alpha) = g_k(\beta) \quad \forall k, 1 \leq k \leq m
\]
Review: Dependence Testing

SIV - Single Induction Variable Test

1. Single loop nest with constant lower (LB) and upper (UB) bounds, and step 1

   for i = LB, UB, 1
     ...
   endfor

   The loop bounds define the iteration space for loop induction variable i.

2. Two array references with array subscript (index) expressions of the form (true dependence)

   for i = LB, UB, 1
     R1: X(a*i + c1) = ... \ write
     R2: ... = X(a*i + c2) ... \ read
   endfor

   where a, c1, and c2 are integer constants, R1 and R2 are references to the same array, i is the loop induction variable, and a \( \neq 0 \).

Question: Is there a true dependence between R1 and R2?
There is a dependence between R1 and R2 iff

$$\exists i, i' : i \leq i' \text{ and } (a \times i + c_1) = (a \times i' + c_2)$$

where $i$ and $i'$ are two iterations in the iteration space of the loop. This means that in both iterations, the same element of array $X$ would be accessed.

So let’s just solve the equation:

$$\frac{c_1 - c_2}{a} = i' - i = \Delta d$$

There is a dependence with distance $\Delta d$ iff

1. $\Delta d$ is an integer value and
2. $\text{UB} - \text{LB} \geq \Delta d \geq 0$
Review: Dependence Testing Examples

1. for i = LB, UB, 1
   R1: X(i) = ... \ write
   R2: ... = X(i - 2) ... \ read

   a=1, c₁=0, c₂=-2 ⇒ Δd = 2 (dependence)

2. for i = LB, UB, 1
   R1: X(2*i) = ... \ write
   R2: ... = X(2*i - 1) ... \ read

   a=2, c₁=0, c₂=-1 ⇒ Δd = \frac{1}{2} (no dependence)

Assume R1 executes before R2.

Classification of dependences:

- R1 is write, R2 is read ⇒ true dependence
- R1 is read, R2 is write ⇒ anti dependence
- R1 is write, R2 is write ⇒ output dependence
Distance Vectors

Distance Vector: number of iterations between accesses to the same memory location

\[
\begin{align*}
&\text{do } I = 1, N \\
&\text{do } J = 1, N \\
&S_1 \ A(I,J) = A(I,J-1) \\
&\text{enddo} \\
&\text{enddo} \\
&S_2 \ A(I,J) = A(I-1,J-1) \\
&S_3 \ B(I,J) = B(I-1,J+1) \\
&\text{enddo} \\
&\text{enddo}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Distance Vector</th>
<th>Distance</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_1(\text{wt } A)\delta S_1(\text{rd } A))</td>
<td>(0,1)</td>
<td>(=,&lt;)</td>
</tr>
<tr>
<td>(S_2(\text{wt } A)\delta S_2(\text{rd } A))</td>
<td>(1,1)</td>
<td>(&lt;,&lt;&lt;)</td>
</tr>
<tr>
<td>(S_3(\text{wt } B)\delta S_3(\text{rd } B))</td>
<td>(1,-1)</td>
<td>(&lt;,&gt;)</td>
</tr>
</tbody>
</table>

Zheng Zhang
A dependence \( D = (d_1, \ldots, d_k) \) is carried at level \( i \), if \( d_i \) is the first nonzero element of the distance/direction vector.

A loop \( l_i \) is parallel, if there is no dependence vector \( D \) carried at level \( i \) for any pair of memory references.

\[
\begin{align*}
\text{do } I & = 1, N \\
& \quad \text{do } J = 1, N \\
S_1 & \quad A(I,J) = A(I,J-1) \\
\text{do } I & = 1, N \\
& \quad \text{do } J = 1, N \\
S_2 & \quad A(I,J) = A(I-1,J-1) \\
\text{do } I & = 1, N \\
& \quad \text{do } J = 1, N \\
S_3 & \quad B(I,J) = B(I-1,J+1)
\end{align*}
\]
Dependence Analysis for Array Reference

A **loop-independent** dependence exists regardless of the loop structure. The source and sink of the dependence occur on the same loop iteration.

A **loop-carried** dependence is induced by the iterations of a loop. The source and sink of the dependence occur on different loop iterations. *Loop-carried dependences can inhibit parallelization and loop transformations*

```plaintext
do I = 1, 100
   A(I) = ...
       ...= A(I-1)
enddo

```

```plaintext
do I = 1, 100
   A(I) = ...
       ...= A(I) + 1
enddo
```
Next Lecture

Next time:

- Project 3 and OpenMP, Review on Dependence Analysis, Automatic Vectorizer