Module 17: Loops

Lecture 33: Data Dependence in Parallel Loops

The Lecture Contains:

Previous Next

Module 17: Loops

Lecture 33: Data Dependence in Parallel Loops

Data Dependence in Parallel Loops

- Two statements or loops have data access conflict when they refer to the same location
- It is resolved by completing the first access before initiating the second one

Forall Loop

- Each statement computes rhs for all index values before store
  forall i = 2 to 10 do
  x[i] = x[i-1] + x[i+1]
  endall
- If it were a sequential loop
  flow dependence from x[i] to x[i-1]
  anti dependence from x[i+1] to x[i]
- Semantics of forall loop
  - Fetch all old values before writing therefore, no flow dependence
  - Two anti dependence from S to S with distances (-1, 1)

Dopar Loop

- Each iteration starts with copies of variables with the values available before the loop
- Value computed in one iteration can not be fetched in another iteration
- If there is conflict between two stores, language model does not resolve. It is likely to be a programmer error

Previous Next

Module 17: Loops

Lecture 33: Data Dependence in Parallel Loops

Consider:

dopar i = 2, 20

X[i] = Y[i] + 1

Z[i] = X[i-1] + X[i] + X[i+1]

enddopar

I	S2	S3
2	X[2]=Y[2]+1	Z[2]=X[1]+X[2]+X[3]
3	X[3]=Y[3]+1	Z[3]=X[2]+X[3]+X[4]
4	X[4]=Y[4]+1	Z[4]=X[3]+X[4]+X[5]

- Within each iteration $S_2 \ \delta_\infty^f \ S_3$
- Across the iteration anti dependence from X[i] to X[i-1] and X[i+1]

Dosingle Loop

- Single assignment rules excludes output dependence
- There can not be anti dependence since each variable is defined only once
- Any access conflict must resolve in favour of definition first followed by use, therefore, only flow dependencies

Consider:

dosingle i = 2, 20

X[i] = Y[i] + 1

Z[i] = X[i-1] + X[i] + X[i+1]

enddosingle

Def	Use	Dependence	Distance
X[i]	X[i-1]	flow	1
	X[i]	flow	0
	X[i+1]	flow	-1

Previous Next

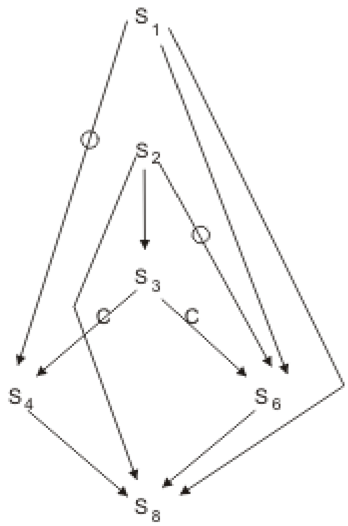
Module 17: Loops

Lecture 33: Data Dependence in Parallel Loops

Program Dependence Graph (PDG)

Control dependence edges in data dependence graph.

S_1	X = 1
S_2	Y = 2
S_3	if Y < T then
S_4	X = 2
S_5	else
S_6	Y = X
S_7	endif
S_8	Z = X + Y



Data Dependence Analysis For Arrays

- Concentrate on linear subscripts:

Linear	[I], [I + J - 1], [10 * I - 1, J * 2]
Non-Linear	[I * J], [I/J], [mod(I, 2) + 1]
	[IP[I] + 1]
Linear & Non-Linear	[2 * I - 1, I * J]

- In case of non-linear subscripts
  - Ignore the subscript
  - Use special solvers(very little work available)

Building Dependence Systems

- Form dependence equations
- Unknowns are loop induction variables
- Coefficients are compile time constants
- One coefficient for each loop induction variable plus a constant coefficient
- Generally use induction variable corresponding to normalized or semi normalized loops.
  i^d indicates a definition
  i^u indicates a use
- Express dependence equation as a matrix notation
  AI = C
  where A: Coefficient matrix; I: Vector of unknowns C: Constant vector

If there is no solution, there can be no dependence.

Previous Next

Module 17: Loops

Lecture 33: Data Dependence in Parallel Loops

Additional Constraints

- A solution must lie within limits of induction variable value
- There must be an integer solution
- Compiler may have to determine dependence distance or direction vector

Example:

Find out dependence eqn.

for I = 2, N

for J = 1, I

B[I,J] = B[I-1,J] + A[I]*C[J]

endfor

endfor

There are 2 eqns., one for each dimension

$$B[2+i_1, 1+i_2] = B[1+i_1, 1+i_2] + A[2+i_1] * C[1+i_2]$$

$$2 + i_1^d = 1 + i_1^u$$

$$1 + i_2^d = 1 + i_2^u$$

$$\begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} i_1^d \\ i_1^u \\ i_2^d \\ i_2^u \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

Previous Next

Module 17: Loops

Lecture 33: Data Dependence in Parallel Loops

Loop Limit Constraints

- Loop limits add constraints
- Lower limit is zero for normalized loop
- Constraints are realities
- Dependence system can be expressed in matrix notation.

A i = C Linear equalities
B i = b Linear inequalities
(+)ve coefficient for unknown Upper limit
(-)ve coefficient for unknown Lower limit

Example:

```
for I = 2, 100
for J = 1, I-1
B[I, J] = B[J, I]
endfor
endfor
```

The 2 eqns. for equality are:

$$2 + i_1^d = 1 + i_2^u$$
$$1 + i_2^d = 2 + i_1^u$$
$$\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} i_1^d \\ i_1^u \\ i_2^d \\ i_2^u \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Previous Next