Loop Dependence and Parallelism

Announcements

- Reading assignments for next week due
- Midterm is Monday March 6th, example midterms posted
- Assignment 2 is due the week after that. Start now!

Today

- Data dependencies between loop iterations
- Determining when a loop is parallelizable
- LLVM and vectorization

Loop Data Dependence Analysis

Loop transformations

- Parallelization
- Rescheduling to improve data locality

Data dependence analysis

- What is the partial ordering between iterations based on data dependences?
- That partial ordering must be maintained.

Data Dependences

Recall

- A data dependence defines ordering relationship two between statements
- In executing statements, data dependences must be respected to preserve correctness

Example

s_1	`a := 5;`		s_1	`a := 5;`
s_2	`b := a + 1;`	$\overset{?}{\equiv}$	s_3	`a := 6;`
s_3	`a := 6;`		s_2	`b := a + 1;`

Dependences and Loops

Loop-independent dependences

```
do i = 1,100
     A(i) = B(i)+1
     C(i) = A(i)*2
enddo
```

Dependences within the same loop iteration

Loop-carried dependences

```
do i = 1,100
   A(i) = B(i)+1
  C(i) = A(i-1)*2
enddo
```

Dependences that cross loop iterations

Data Dependence Terminology

We say statement s_2 depends on s_1

- **True (flow) dependence**: s_1 writes memory that s_2 later reads
- **Anti-dependence**: s_1 reads memory that s_2 later writes
- **Output dependences**: s_1 writes memory that s_2 later writes
- **Input dependences**: s_1 reads memory that s_2 later reads

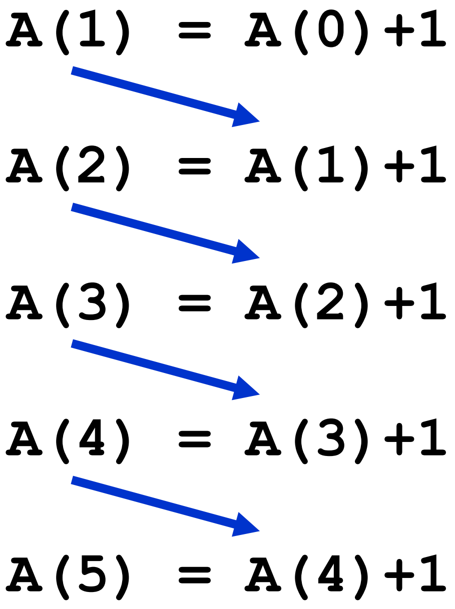
Notation: $s_1 \delta s_2$

- s_1 is called the **source** of the dependence
- s_2 is called the **sink** or **target**
- s_1 must be executed before s_2

Data Dependences and Loops

How do we identify dependences in loops?

```
do i = 1,5
   A(i) = A(i-1)+1
enddo
```



```
A(1) = A(0)+1
A(2) = A(1)+1
A(3) = A(2)+1
A(4) = A(3)+1
A(5) = A(4)+1
```

Simple view

- Imagine that all loops are fully unrolled
- Examine data dependences as before

Problems

- Impractical and often impossible
- Lose loop structure

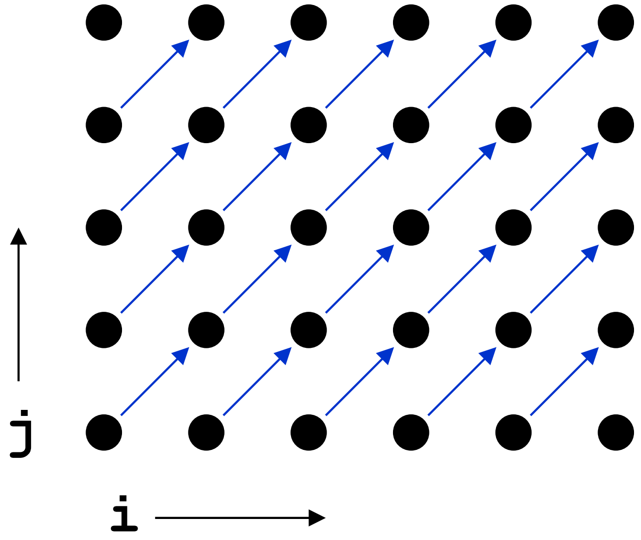
Iteration Spaces

Idea

- Explicitly represent the iterations of a loop nest

Example

```
do i = 1,6
    do j = 1,5
        A(i,j) = A(i-1,j-1)+1
    enddo
enddo
```


Iteration Space

Iteration Space

- A set of tuples that represents the iterations of a loop
- Can visualize the dependences in an iteration space

Distance Vectors

Idea

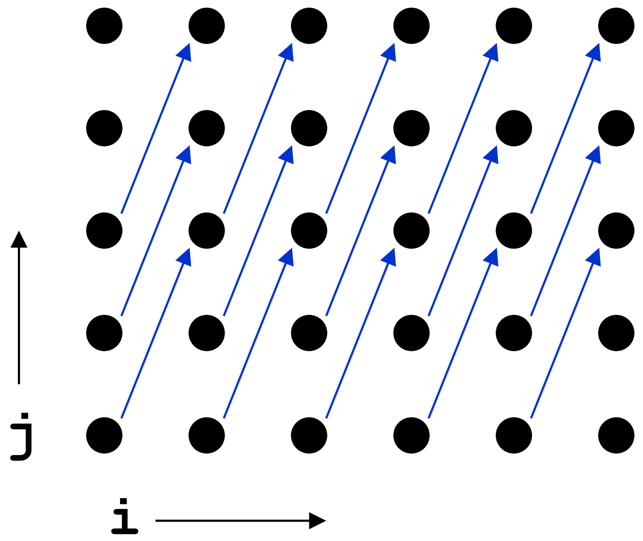
- Concisely describe dependence relationships between iterations of an iteration space
- For each dimension of an iteration space, the distance is the number of iterations between accesses to the same memory location

Definition

- $\mathbf{v} = \mathbf{i}^T - \mathbf{i}^S$

Example

```
do i = 1,6
   do j = 1,5
        A(i,j) = A(i-1,j-2)+1
     enddo
enddo
```



Distance Vector: (1,2)

outer loop

inner loop

Distance Vectors: Legality

Definition

- A dependence vector, *v*, is **lexicographically nonnegative** when the left-most entry in *v* is positive or all elements of *v* are zero
  Yes: (0,0,0), (0,1), (0,2,-2)
  No: (-1), (0,-2), (0,-1,1)

- A dependence vector is **legal** when it is lexicographically nonnegative (assuming that indices increase as we iterate)

Why are lexicographically negative distance vectors illegal?

What are legal distance vectors?

Loop-Carried Dependences

Definition

- A dependence $D=(d_1,...d_n)$ is **carried** at loop level i if d_i is the first nonzero element of D

Example

```
do i = 1,6
  do j = 1,6
      A(i,j) = B(i-1,j)+1
      B(i,j) = A(i,j-1)*2
      enddo
enddo
```

Distance vectors: (0,1) for accesses to **A**
(1,0) for accesses to **B**

Loop-carried dependences

- The j loop carries dependence due to **A**
- The i loop carries dependence due to **B**

Direction Vector

Definition

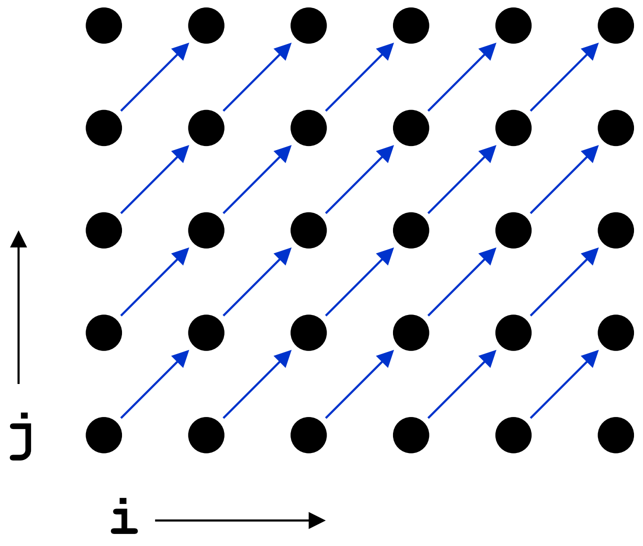
- A direction vector serves the same purpose as a distance vector when less precision is required or available
- Element *i* of a direction vector is <, >, or = based on whether the source of the dependence precedes, follows or is in the same iteration as the target in loop *i*

Example

```
do i = 1,6
     do j = 1,5
         A(i,j) = A(i-1,j-1)+1
     enddo
enddo
```

Direction vector: (<,<)

Distance vector: (1,1)



Data dependence analysis examples

Do some examples in class.

Concepts

Loops

- Data dependences including distance vectors,
- loop carried dependences, and
- direction vectors.
- How to determine a loop can be parallelized.

Ways to Parallelize

What ways can you parallelize a computation?