Otimizações em Sistemas Vetoriais

Tópicos:
- Inversão de Loops
- Fusão de Loops
- Distribuição de Loops
- Junção de Loops

Referência: Padua,D.A. & Wolfe,M.J. "Advanced Compiler Optimizations for Supercomputers", *Communications of the ACM*, 29(12), Dec. 1986, pp.1184-1200.

Inversão de Loops

Idéia: Trocar o ordem de loops aninhados, de modo a facilitar a vetorização ou a paralelização

Exemplo:

```
L1:     do j=1,N
          do i=2,N
            A(i,j) = A(i-1,j) + B(i)
          enddo
        enddo

L2:     do i=2,N
          do j=1,N
            A(i,j) = A(i-1,j) + B(i)
          enddo
        enddo
```


Inversão de Loops (cont.)

L1: Loop **j** **pode** ser paralelizado; Loop **i** **não** pode ser vetorizado

```
L1':    doall j=1,N
          do i=2,N
            A(i,j) = A(i-1,j) + B(i)
          enddo
        enddoall
```

L2: Loop **i** **não** pode ser paralelizado; Loop **j** **pode** ser vetorizado

```
L2':    do i=2,N
          A(i,1:N) = A(i-1,1:N) + B(i)
        enddo
```

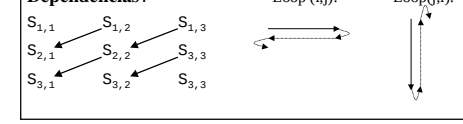

Inversão de Loops (cont.)

Obs: Inversão pode não ser possível em alguns casos

Exemplo:

```
L3:     do i=1,N
          do j=i,N
S:          A(i,j) = A(i-1,j+1)
          enddo
        enddo
```

Dependências: Loop (i,j): Loop(j,i):



Fusão de Loops

Def: Transformação de loops adjacentes em um único loop

Objetivo: Minimizar o overhead de inicialização dos loops

Exemplo:

```
    do i=2,N
S1:   A(i) = B(i) + C(i)
    enddo
    do i=2,N
S2:   D(i) = A(i-1)
    enddo
```

Obs: A fusão destes dois loops é válida, pois a dependência $S_1 \ \delta \ S_2$ continua sendo respeitada

Fusão de Loops (cont.)

Obs: Fusão pode não ser possível em alguns casos

Exemplo:

```
    do i=2,N
S1:   A(i)=B(i)+C(i)
    enddo
    do i=2,N
S2:   D(i)=A(i+1)
    enddo
```

Loop após fusão ficaria:

```
    do i=2,N
S1:   A(i)=B(i)+C(i)
S2:   D(i)=A(i+1)
    enddo
```

- Antes da fusão: $S_1 \ \delta \ S_2$
- Após fusão: $S_2 \ \delta^a \ S_1$

Dependência original é violada no novo loop

⇒ Fusão **não** é válida neste caso

Distribuição de Loops

Def: Transformação de um loop único em loops adjacentes

Objetivo: Possibilitar vetorização ou paralelização parcial do loop

Exemplo:

```
      do i=2,N
S1:      A(i) = B(i) + C(i)
S2:      D(i) = D(i-1) * 2
      enddo
```

Pode ser transformado em

```
S1:  A(2:N) = B(2:N) + C(2:N)
      do i=2,N
S2:      D(i) = D(i-1) * 2
      enddo
```


Junção de Loops

Idéia: Transformar um loop duplo num loop simples, para facilitar a vetorização

Exemplo:

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

Pode ser transformado em

```
do ij=1,25
  AA(ij) = BB(ij) + 2
enddo
```