

# Computación de Altas Prestaciones HPC

## Problemas Tema 2: Vectorización y paralelización de bucles

**2.1.-** El siguiente bucle tiene varios tipos de dependencias. Encuentra todas las dependencias verdaderas, antidependencias y dependencias de salida, y elimine las antidependencias y dependencias de salida cambiando el nombre.

```
for (i=0; i<100; i=i+1) {  
    Y[i] = X[i] / c;      /* S1 */  
    X[i] = X[i] + c;      /* S2 */  
    Z[i] = Y[i] + c;      /* S3 */  
    Y[i] = c - Y[i];      /* S4 */  
}
```

**2.2.-** El siguiente bucle se va a ejecutar en una máquina multithread:

```
do i = 1, N-2  
    (1) A(i) = B(i)  
    (2) C(i) = A(i) + B(i-1)  
    (3) D(i) = C(i+1)  
    (4) B(i) = C(i) * 2  
enddo
```

Genere el grafo de dependencias

Escribe una versión paralela de dicho bucle y suponiendo que el tiempo de ejecución de una instrucción es T, ¿Cuántas veces más rápido es la ejecución al paralelizar en P procesadores?. ¿Cuál será la máxima aceleración con infinitos procesadores y despreciando el tiempo de sincronización.

**2.3.-** Examine los siguientes bucles y analice su posible paralelización:

a) ¿ Tiene dependencias las iteraciones del siguiente bucle?

```
for (i=0; i<100; i++) {  
    A[i] = B[2*i+4];  
    B[4*i+5] = A[i];  
}
```

I

b) Encuentra todas las dependencias verdaderas, antidependencias y dependencias de salida, y elimine las antidependencias y dependencias de salida cambiando el nombre.

```
for (i=0; i<100; i++) {  
    A[i] = A[i] * B[i]; /* S1 */  
    B[i] = A[i] + c; /* S2 */  
    A[i] = C[i] * c; /* S3 */  
    C[i] = D[i] * A[i]; /* S4 */  
}
```

c) Considere el siguiente bucle:

```
for (i=0; i < 100; i++) {  
    A[i] = A[i] + B[i]; /* S1 */  
    B[i+1] = C[i] + D[i]; /* S2 */  
}
```

¿Hay dependencias entre S1 y S2? ¿Es paralelizable? En caso negativo, muestre como hacerlo paralelizable.

**2.4.** Represente el grafo de dependencias de los siguientes bucles:

a)

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

b)

```
do i = 1, N-2  
do j = 1, N-2  
    (1) A(i,j) = A(i,j-1) * 2  
    (2) C(i,j) = A(i-2,j+1) + 1  
Enddo  
enddo
```

**2.5.-** El siguiente bucle se va a ejecutar en una máquina SMP de 4 procesadores:

```
do i = 1, 994  
    (1) A(i) = D(i) - 10  
    (2) B(i+1) = B(i+1) * A(i-1)  
    (3) D(i+1) = D(i+1) + D(i+1)  
Enddo
```

- Genere el grafo de dependencias y el espacio de iteraciones correspondientes al bucle. Escriba una versión paralela de dicho bucle que pueda ejecutarse en P procesadores sin tener que utilizar funciones de sincronización.
- Para el caso de esta máquina de 4 procesadores, escriba el código que ejecutará cada procesador (el bucle paralelo) si el reparto de las iteraciones entre los procesadores es (1) estático consecutivo; y (2) dinámico por chunks ( trozos), con trozos de 50 iteraciones. ¿Cuántas iteraciones ejecutará el procesador P1 en el primer caso? ¿y en el segundo?

**2.6-** Para cada uno de los siguientes bucles indique en qué casos existen dependencia entre las dos instrucciones dadas. Justifique la respuesta.

a)

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

b)

```
do i = 5, 100  
    A(i-5) = ...  
    ... = A(2*i+90)  
enddo
```

```

c)
do i = 1, 100
    A(3*i+100) = ...
    ... = A(2*i-1)
enddo
d)
do i = 1, 100
    A(6*i+3) = ...
    ... = A(3*i+81)
enddo

```

**2.7.-** Indique cómo paralelizar de la manera más efectiva el bucle siguiente para una máquina con 40 procesadores. Escriba el pseudocódigo para el caso de planificación estática consecutiva y realice una estimación de la aceleración que se obtendrá.

```

do i = 0, 19
    do j = 0, 199
        A(i, j) = A(i, j) + 1
        B(2i+1, j) = B(2i, j) * 2
    enddo
enddo

```

**2.8.-** Considere el siguiente bucle a ejecutar en una máquina paralela:

```

do i = 1, 24
    do j = 1, i
        (1) A(i, j) = B(i, j) + 1
        (2) C(i, j) = A(i-1, j) * 2
    enddo
enddo

```

a) Genere el grafo de dependencias y decida cómo paralelizar el código de la manera más eficiente posible. Escriba el código correspondiente al caso de un sistema de 4 procesadores, en el que la planificación de las tareas es dinámica por trozos de tamaño fijo, 4 iteraciones en concreto.

b) Indique el número de tareas de planificación necesarias y haz una estimación razonada del speed-up que se puede conseguir al ejecutar el programa de esa manera. Supón que el coste de ejecución de una instrucción como las del bucle es 1, y que una operación de sincronización tiene un coste de 10.

**2.9.-** Dado el siguiente bucle, indique cómo ejecutarlo en paralelo de la manera más eficiente posible. Escriba el código resultante. Si el tiempo de ejecución de cada instrucción es T, realice una estimación de la aceleración y de la eficiencia que se conseguirá al utilizar 10 procesadores. Justifique los resultados.

```

do i = 2, 93
    (1) A(i) = D(i) - 10
    (2) B(i+1) = C(i-1) * D(i-2)
    (3) C(i) = C(i-1) + A(i+5)
    (4) D(i-1) = E(i+1) * 5
enddo

```

**2.10.-** Dado el siguiente bucle, genera el grafo de dependencias e indica cómo paralelizar el código sin necesidad de sincronización (efectuando peeling.)

do i= 1, 198

$A(i) = B(i-1) * 3$

$B(i+1) = C(i-1) + 1$

$C(i) = C(i) * 0,5$

enddo

**2.11.-** Dado el siguiente bucle, genera el grafo de dependencias e indica cómo paralelizar el código sin necesidad de sincronizar los procesos.

do i= 1, 1000

$A(i) = A(i) + 1$

$B(i+1) = C(i) * A(i)$

$C(i+1) = C(i+1) + D(i)$

$F(i) = C(i+1) / F(i)$

enddo

**2.12.-** El siguiente bucle se ejecuta en una máquina de 10 procesadores. Analiza cómo paralelizar el bucle de la manera más efectiva y escribe el código que ejecutará cada procesador para el caso de planificación estática consecutiva

Nota: cada procesador se identifica por la variable privada  $pid = 0..9$ .

do i = 2, 101

    do j = 0, 199

$A(i,j) = A(i,j+2) * 2$

$B(i,j) = B(i-2,j) * 3$

    enddo

enddo

**2.13.-** Se Se ejecuta en un sistema paralelo de 4 procesadores el siguiente bucle:

do i = 1, 40

    <código>

enddo

Todas las iteraciones son independientes entre sí y se sabe que el tiempo de ejecución de las mismas es  $T(i) = i$  (es decir, la primera iteración 1, la segunda 2, etc.).

Calcula el tiempo de ejecución en paralelo del programa, y el factor de aceleración respecto al caso serie, si la planificación de tareas es:

a) estática: 1. consecutiva; y 2. entrelazada

b) dinámica: 1. Autoplanificación con:  $Z = 1$  y  $Z = 5$

2. autoplanificación guiada,  $Z_s = \text{ceil}[(N - i) / P]$

3. trapezoidal,  $Z_1 = 9$ ,  $k = 1$

Considere que el tiempo de cada operación de planificación dinámica es 1.

**2.14.-** Se ejecutan en paralelo, entre 4 procesadores, 100 iteraciones de un determinado bucle, independientes entre sí.

El tiempo de ejecución de cada iteración es de 1 s, salvo una de las iteraciones, que tiene un tiempo de ejecución de 20 s.

Calcula, de manera aproximada, el speed-up y la eficiencia que se consigue en los siguientes casos:

- (a) el reparto de iteraciones es estático consecutivo (con trozos del mismo tamaño).
- (b) el reparto de iteraciones es estático entrelazado (con trozos del mismo tamaño).
- (c) el reparto de iteraciones es dinámico, 1 a 1 (calcula el peor y el mejor caso); coste de una operación de asignación de tareas, 50 ms.