

# Tema 2: Paralelización de bucles

Sesión de problemas

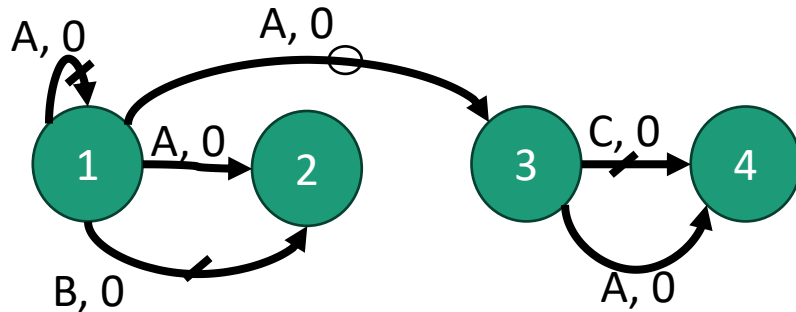
# Lista de ejercicios que quiero hacer

- Ejercicio 2.2 (parcialmente hecho en el alg. A-K)
- Ejercicio 2.3 (b)
- Ejercicio 2.4 (parcialmente hecho en el alg. A-K)
- Ejercicio 2.5 (parcialmente hecho en el resumen)
- Ejercicio 2.7
- Ejercicio 2.9
- Ejercicio 2.14

- 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 */
}
```

```
for(i=0,99)
  A1[i] = A[i]*B[i]
  B1[i] = A1[i]+c
  A2[i] = C[i]*c
  C1[i] = D[i] * A2[i]
Endfor
```



```
Rename A2 -> A
Rename B1 -> B
Rename C1 -> C
```

**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
```

```
doall i=0,19
```

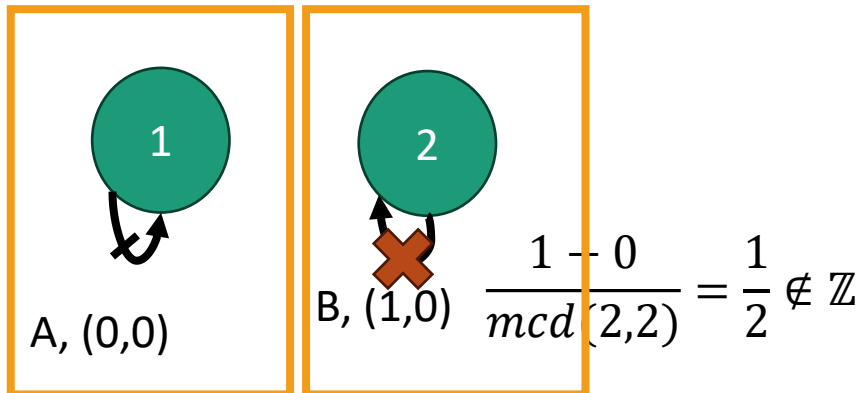
```
  doall j=0,199
```

```
    S1
```

```
doall i=0,19
```

```
  doall j=0,199
```

```
    S2
```

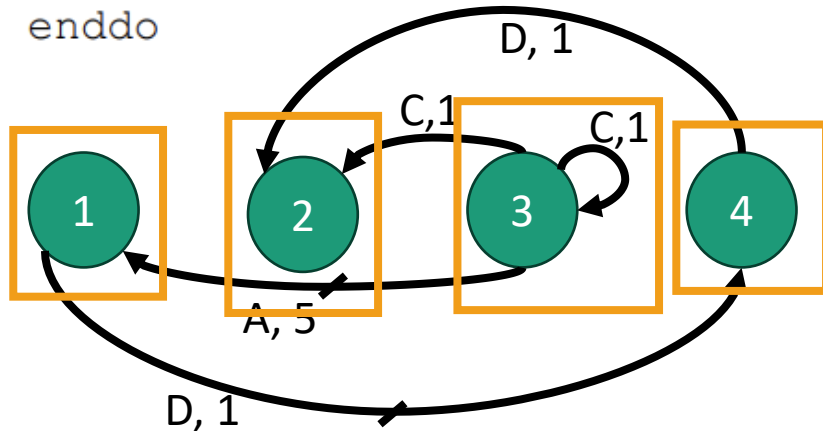


Hay  $20 \cdot 200 \cdot 2 = 8000$  tareas, luego cada procesador tiene que hacer  $\frac{8000}{40} = 200$ . Luego usando planificación estática consecutiva, P0 hará las 200 primeras, P1 las 200 siguientes y así...

$$Speedup = S(40) = \frac{T_s}{T_p} = \frac{200 * 20 * 2}{(200 * 20 * 2)/40} = 40$$

**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
```



C3 # Serie  
C1 # Paralelo  
C4 # Paralelo  
C2 # Paralelo

```
for i=2,93
  (3) C(i) = C(i-1)+A(i+5)
```

```
doall i=2,93
```

```
  (1) A(i) = D(i)-10
```

```
barrier
```

```
doall i=2,93
```

```
  (4) D(i-1) = E(i+1)*5
```

```
barrier
```

```
doall i=2,93
```

```
  (2) B(i+1) = C(i-1)*D(i-2)
```

Candidato a peeling

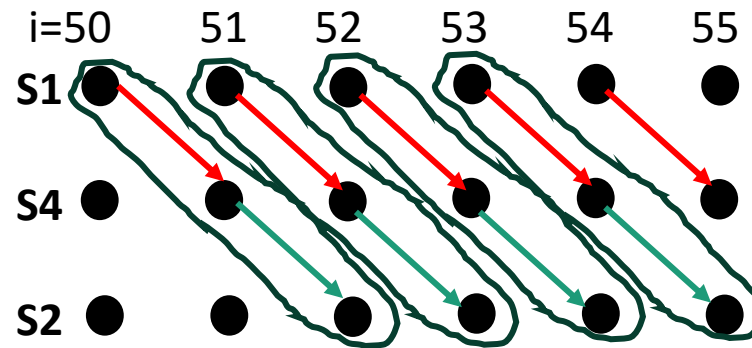
$$S(10) = \frac{92T \cdot 4}{92T + \frac{92T \cdot 3}{10} + 2T_{barr}} = \frac{368T}{119.6T} = 3.08$$

$$E(10) = \frac{S(10)}{10} = 0.3$$

# Peeling

¿Y si nos piden paralelización sin barreras o evitando las barreras posibles?

```
doall i=2,93
  (1) A(i) = D(i)-10
barrier
doall i=2,93
  (4) D(i-1) = E(i+1)*5
barrier
doall i=2,93
  (2) B(i+1) = C(i-1)*D(i-2)
```



```
doall i=2,93
  (1) A(i) = D(i)-10
barrier
doall i=1,92
  (4) D(i) = E(i+2)*5
barrier
doall i=0,91
  (2) B(i+3) = C(i+1)*D(i)
```

```
for i=2,93
  (3) C(i) = C(i-1)+A(i+5)
D(1)=...,B(3)=...,B(4)=...
doall i=2,91
  (1) A(i) = D(i)-10
  (4) D(i) = E(i+2)*5
  (2) B(i+3) = C(i+1)*D(i)
```

$$S(10) = \frac{368T}{92T + 6T + \frac{90T \cdot 3}{10}}$$

$$= \frac{368T}{125T} = 2.94$$

**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.

$$T(1) = T_s = 99 \cdot (1s) + 1 \cdot (20s) = 119s$$

(a) y (b) Si el reparto es estático, todos los procesadores recibirán  $\frac{100}{4} = 25$  iteraciones. Entonces, algún procesador recibirá la iteración “mala” que tarda 20s.

$$T(proc_{malo}) = 24 \cdot (1s) + 1 \cdot (20s) = 44s$$

$$T(proc_{resto}) = 25 \cdot (1s) = 25s$$

Como el tiempo estará limitado por lo que tarda el peor de ellos,

$$T(p = 4) = 44s$$

$$S(4) = \frac{119}{44} = 2.70 \rightarrow E(4) = 0.68$$

Da igual si el reparto es consecutivo o entrelazado, el procesador que reciba la iteración mala, tardará 44s y determina el tiempo de ejecución paralelo

**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.

$$T(1) = T_s = 99 \cdot (1s) + 1 \cdot (20s) = 119s$$

(c) Si el reparto es dinámico, los procesadores van pidiendo iteraciones de una en una (cada petición introduce un overhead de 50ms). Con esto, intentamos compensar el tiempo de ejecución de la tarea mala dejando que el resto de procesadores cojan más tareas. En el caso mejor, la tarea mala es de las primeras. Durante los primeros 20.05s observaremos:

$$W(proc_{malo}) = 1 \text{ its}$$

$$W(proc_{bueno}) = \frac{20.05}{1.05} \text{ its} = 19.10 \text{ its}$$

¿Y después?

Se reparten de una en una entre los 4.  $W(rest) = 100 - \left( \frac{20.05}{1.05} \cdot 3 + 1 \right) = 41.71 \text{ its}$

$$T_{p=4}(rest) \approx \frac{41.71}{4} \cdot 1.05 = 10.95s$$

$$T(4) = 20.05s + 10.95s = 31s \rightarrow S(4) = \frac{119}{31} = 3.8 \rightarrow E(4) = \frac{3.8}{4} = 0.96$$

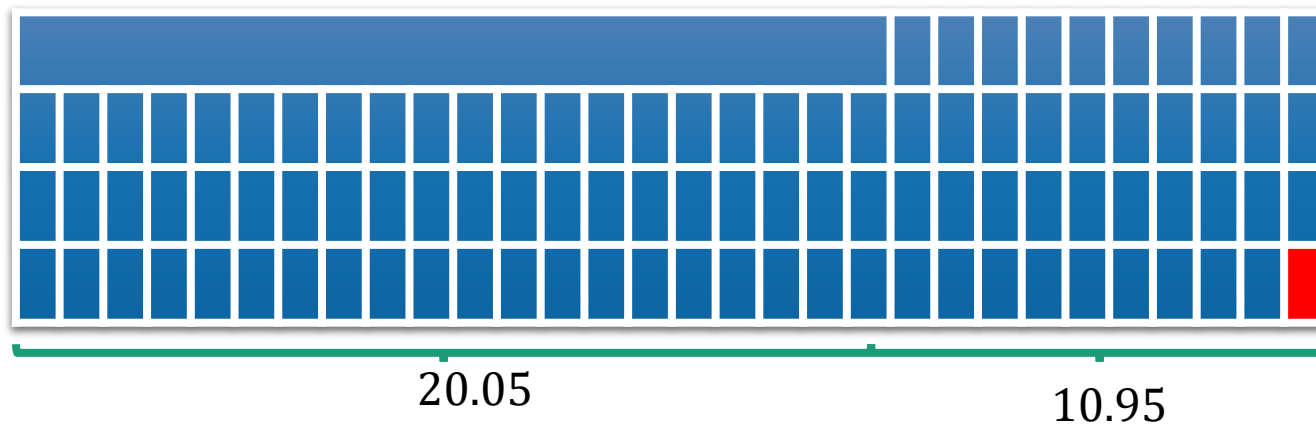


**2.14.-** Se ejecutan en paralelo, entre 4 procesadores, 100 iteraciones de un determinado bucle, independientes entre sí.

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.



$$T(4) = 20.05s + 10.95s = 31s \rightarrow S(4) = \frac{119}{31} = 3.8 \rightarrow E(4) = \frac{3.4}{4} = 0.96$$

**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.

$$T(1) = T_s = 99 \cdot (1s) + 1 \cdot (20s) = 119s$$

(c2) Si el reparto es dinámico, los procesadores van pidiendo iteraciones de una en una (cada petición introduce un overhead de 50ms).

En el caso peor, la tarea mala será de las últimas. Todos los procesadores harán aproximadamente el mismo número de tareas y al final, uno de ellos, la tarea mala. En concreto, ese procesador procesará 24 tareas de 1s, y el resto 25:

$$T(proc_{malo}) = 1.05 \cdot 24 + 20.05 \cdot 1 = 45.25s$$

$$T(proc_{bueno}) = 1.05 \cdot 25 = 26.25s$$

Después ese procesador, se encargará de la tarea mala mientras el resto están en idle porque no hay más tareas.

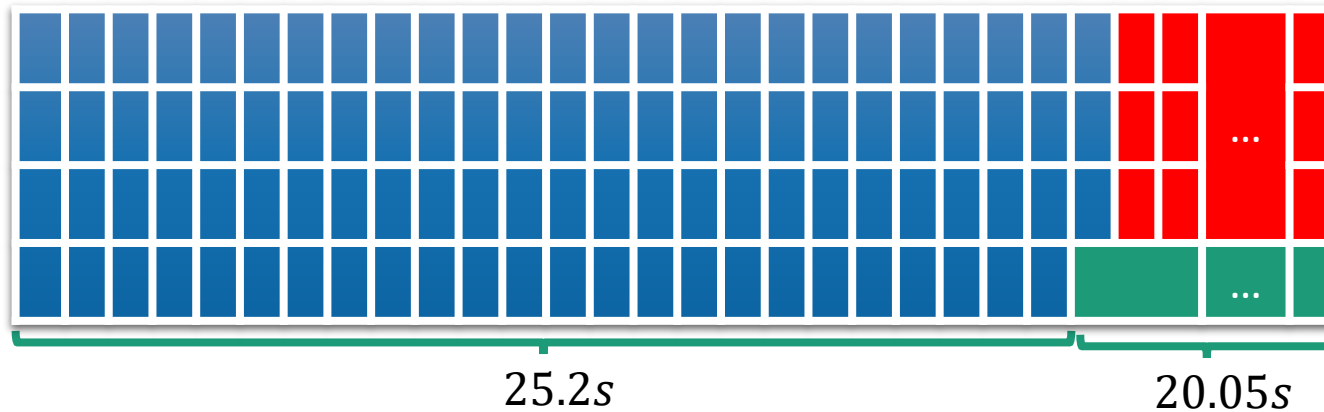
$$T(4) = 25.2 + 20.05 = 45.25s \rightarrow S(4) = \frac{119}{45.25} = 2.6 \rightarrow E(4) = \frac{2.6}{4} = 0.66$$

**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.



$$T(4) = 25.2 + 20.05 = 45.25s \rightarrow S(4) = \frac{119}{45.25} = 2.6 \rightarrow E(4) = \frac{2.6}{4} = 0.66$$