

IV

Programación Paralela

Clase 10: Introducción a la Programación Paralela

Ramiro Martínez D'Elía

2021

Índice general

1. Conceptos	2
1.1. Programa paralelo	2
1.2. Speedup	2
1.3. Eficiencia	2
1.4. Escalabilidad	3
1.5. Ley de Amdhal	3
1.6. Granularidad	3

Capítulo 1

Conceptos

1.1. Programa paralelo

Tipo de programa, concurrente, escrito para resolver un problema en menos tiempo que su análogo secuencial. El objetivo principal es reducir el tiempo de ejecución, o resolver problemas más grandes o con mayor precisión en el mismo tiempo.

Un programa paralelo puede escribirse usando variables compartidas o pasaje de mensajes. La elección la dicta el tipo de arquitectura.

Para determinar si una solución paralela tiene mejores prestaciones que su análogo secuencial, existen nociones como: *speedup*, *eficiencia*, *escalabilidad*, y *ley de Amndhal*.

1.2. Speedup

La métrica de speedup (o eficiencia) **evalúa el tiempo total de ejecución** de un programa. El speedup está dado por la siguiente fórmula:

$$S = \text{Speedup} = T_1/T_p$$

Donde T_1 es el tiempo de una solución secuencial ejecutando en 1 CPU y T_p es el tiempo de una solución paralela con p CPUs.

Del resultado obtenido, podemos concluir las siguientes afirmaciones:

- Si $S = P \rightarrow$ speedup *lineal* (o *perfecto*).
- Si $S < P \rightarrow$ speedup *sublineal*
- Si $S > P \rightarrow$ speedup *superlineal*.

Por ejemplo, si el tiempo de una solución secuencial es de 600 segundos y el de una solución paralela es de 60 segundos utilizando 10 CPU.

$$\text{Speedup} = T_1/T_p = 600/60 = 10$$

Como $S = P \rightarrow$ speedup lineal

1.3. Eficiencia

Esta métrica permite conocer **qué tan bien aprovechará procesadores extras** un programa paralelo. La eficiencia, está dada por la siguiente fórmula:

$$E = \text{Eficiencia} = \text{Speedup}/p$$

Donde p es la cantidad de CPUs. Del resultado obtenido, podemos concluir las siguientes afirmaciones:

- Con speedup perfecto $\rightarrow E = 1$.
- Con speedup sublineal $\rightarrow E < 1$.
- Con speedup superlineal $\rightarrow E > 1$.

Por ejemplo, si el tiempo de una solución secuencial es de 600 segundos y el de una solución paralela es de 60 segundos utilizando 10 CPU.

$$Speedup = T_1/T_p = 600/60 = 10$$

$$Eficiencia = Speedup/P = 10/10 = 1$$

Como el speedup obtenido fue lineal ($S = P$), entonces se cumplió que $E = 1$.

1.4. Escalabilidad

Las métricas de **speedup** y **eficiencia**, son **relativas**. Ellas dependen del número de procesadores, el tamaño de los datos y el algoritmo utilizado. Por esto, de forma general, se dice que **un programa paralelo es escalable si; su eficiencia se mantienen constante para un rango amplio de CPU**.

Por ejemplo: una solución es paralelizada sobre P procesadores de dos maneras. En la primera, el speedup está regido por la función $S = P - 5$. Mientras que, en la otra $S = P/2$. ¿Qué solución se comportará más eficientemente al crecer P ?

P	S = P - 5	S = P / 2
6	$(6 - 5)/6 = 1,66$	$(6/2)/6 = 0,5$
10	$(10 - 5)/10 = 0,5$	$(10/2)/10 = 0,5$
20	$(20 - 5)/20 = 0,75$	$(10/2)/20 = 0,5$

La solución regida por el speedup $S = P/2$ es la más escalable. Ya que, el valor de E se mantiene constante al incrementar el número de procesadores.

1.5. Ley de Amdhal

Un programa típico consta de 3 (tres) etapas; entrada de datos, cómputo y salida de resultados.

La ley de Amdhal postula que; para todo algoritmo **existe un speedup máximo alcanzable**, independiente del número de procesadores. Ese valor dependerá de la cantidad de código paralelizable.

$$Limite = 1/1 - Paralelizable$$

Por ejemplo: suponga un programa secuencial donde las etapas (1) y (2) consumen cada una el 10% del tiempo de ejecución y no pueden ser paralelizadas. La etapa (3) consume el 80% restante. Si el programa tarda 100 unidades de tiempo (ut), ¿cuál es el límite de mejora alcanzable?

$$Limite = 1/1 - 0,8 = 1/0,2 = 5$$

El mejor speedup alcanzable es 5. Esto quiere decir que; el tiempo de ejecución, del algoritmo paralelizado, será como máximo hasta 5 veces más rápido ($100/5 = 20ut$)

1.6. Granularidad