Programación Paralela

Gabriel Astudillo Muñoz

Medidas de rendimiento de arquitecturas de computadores

Rendimiento

Costo computacional de un programa que se ejecuta en un procesador

- tiempo
- memoria

Tiempo de Ejecución

Teórico

$$T_{ejec} = \frac{I_c \cdot CPI}{f}$$

- I_c: # de instrucciones de máquina
- CPI: nro promedio de ciclos por instrucción
- f: frecuencia de reloj del sistema computacional

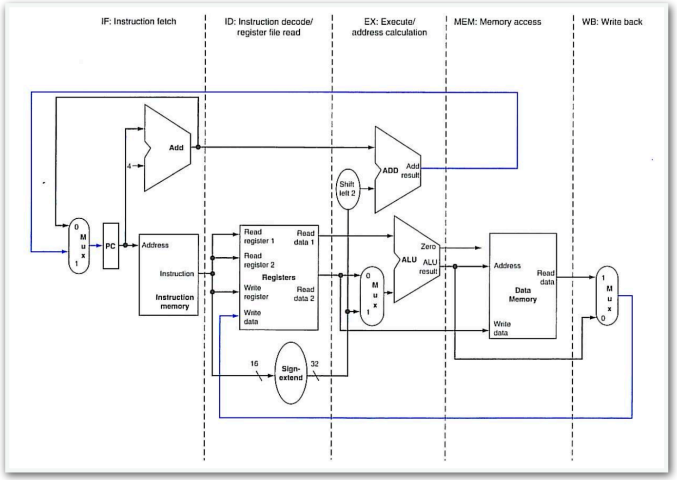
MFLOPS

$$MFLOPS = \frac{nro\ de\ instrucciones\ en\ pto.\ flotante}{T_{ejec} \cdot 10^6}$$

Tiempo de ejecución

$$T_{ejec} = \frac{I_c \cdot CPI}{f}$$

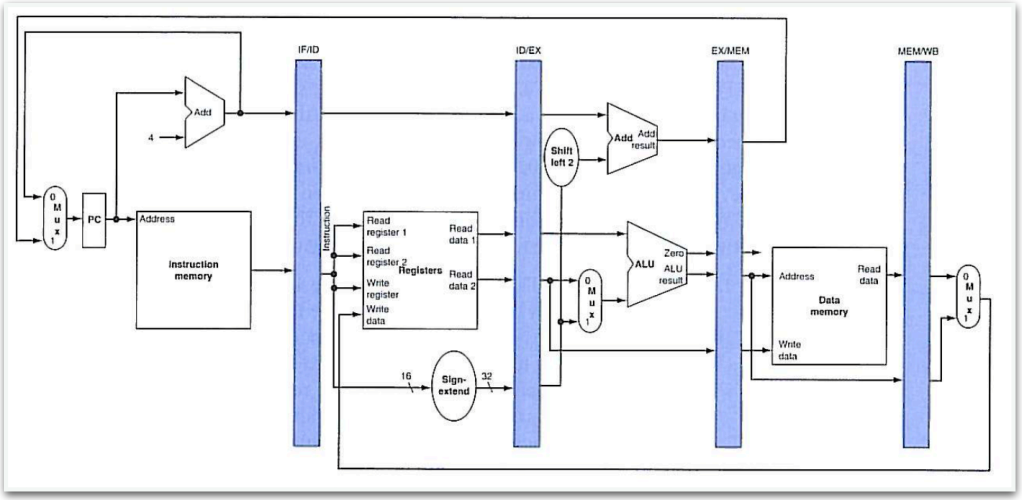
Interesa disminuir el **Tiempo de Ejecución**

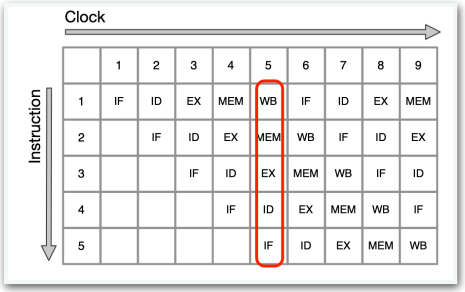


Procesador no segmentado

CPI ≈ 5

Procesador sub-escalar





pipelining

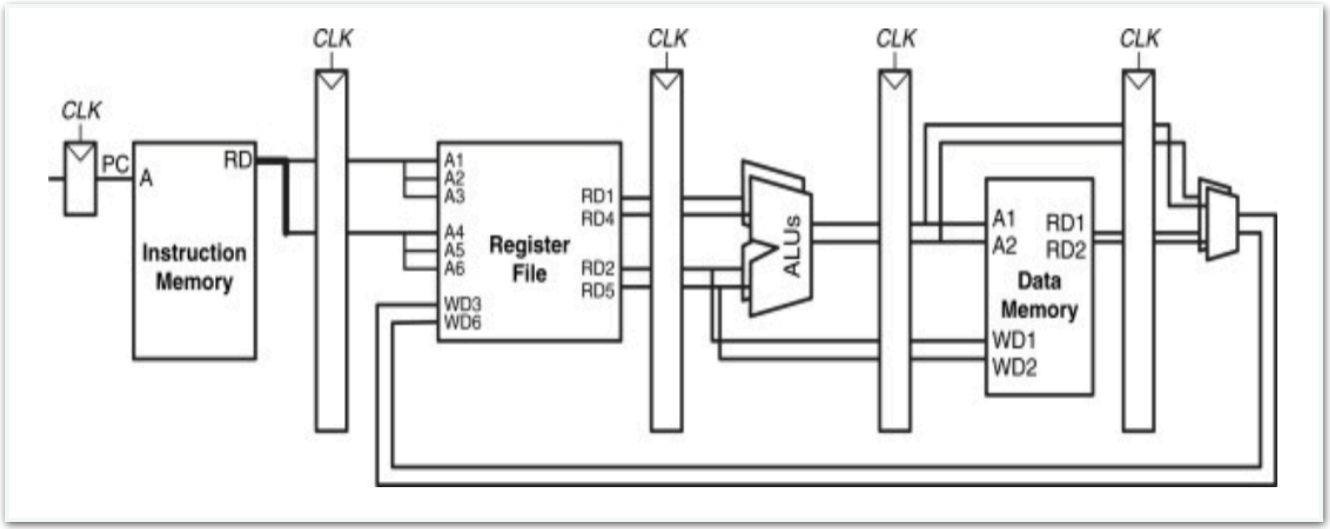
CPI ≈ 1

Procesador escalar

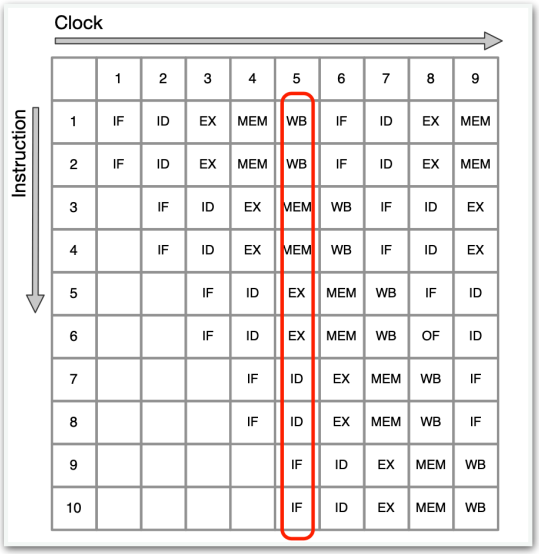
Tiempo de ejecución

$$T_{ejec} = \frac{I_c \cdot CPI}{f}$$

Interesa disminuir el **Tiempo de Ejecución**



pipelining



	1	2	3	4	5	6	7	8	9
1	IF	ID	EX	MEM	WB	IF	ID	EX	MEM
2	IF	ID	EX	MEM	WB	IF	ID	EX	MEM
3		IF	ID	EX	MEM	WB	IF	ID	EX
4		IF	ID	EX	MEM	WB	IF	ID	EX
5			IF	ID	EX	MEM	WB	IF	ID
6			IF	ID	EX	MEM	WB	OF	ID
7				IF	ID	EX	MEM	WB	IF
8				IF	ID	EX	MEM	WB	IF
9					IF	ID	EX	MEM	WB
10					IF	ID	EX	MEM	WB

CPI < 1

Procesador super-escalar

Aceleración (SpeedUp)

Comparación entre los tiempos de ejecución del caso base (secuencial) y del caso mejorado.

$$S(N) = \frac{T_{seq}(N)}{T_{improved}(N)}$$

Caso general

Aceleración Paralela

$$S(N,p) = \frac{T_{seq}(N)}{T_p(N,p)}$$

tiempo secuencial para resolver el problema de tamaño **N** en un LP (la ejecución es con un sólo thread)

tiempo para resolver el problema de tamaño **N** en **p** LP (p.e. se ejecuta con **p** threads).

Eficiencia Paralela

Para un problema de tamaño **N** en **p** procesadores:

$$E(N,p) = \frac{1}{p} \cdot \frac{T_{seq}(N)}{T_p(N,p)} = \frac{S(N,p)}{p}$$

E(N,p) →0 : el programa se ejecuta en un procesador en forma secuencial.

E(N,p) →1 : el programa se ejecuta en todos los procesadores

Programa escala linealmente ssi **E=1**

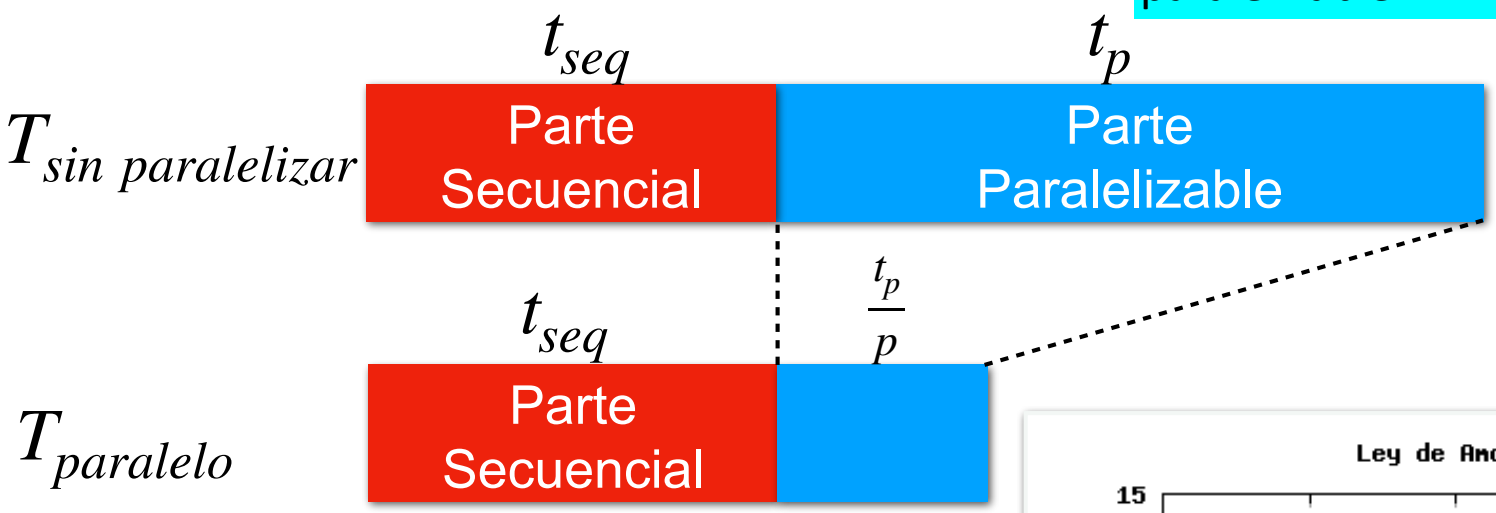
En general, la eficiencia permite identificar el nivel de paralelismo máximo para un problema determinado

Para un valor α, una solución escala para p procesadores, si se cumple:

$$E(N,p) \geq \alpha$$

Límites a la Aceleración (S)

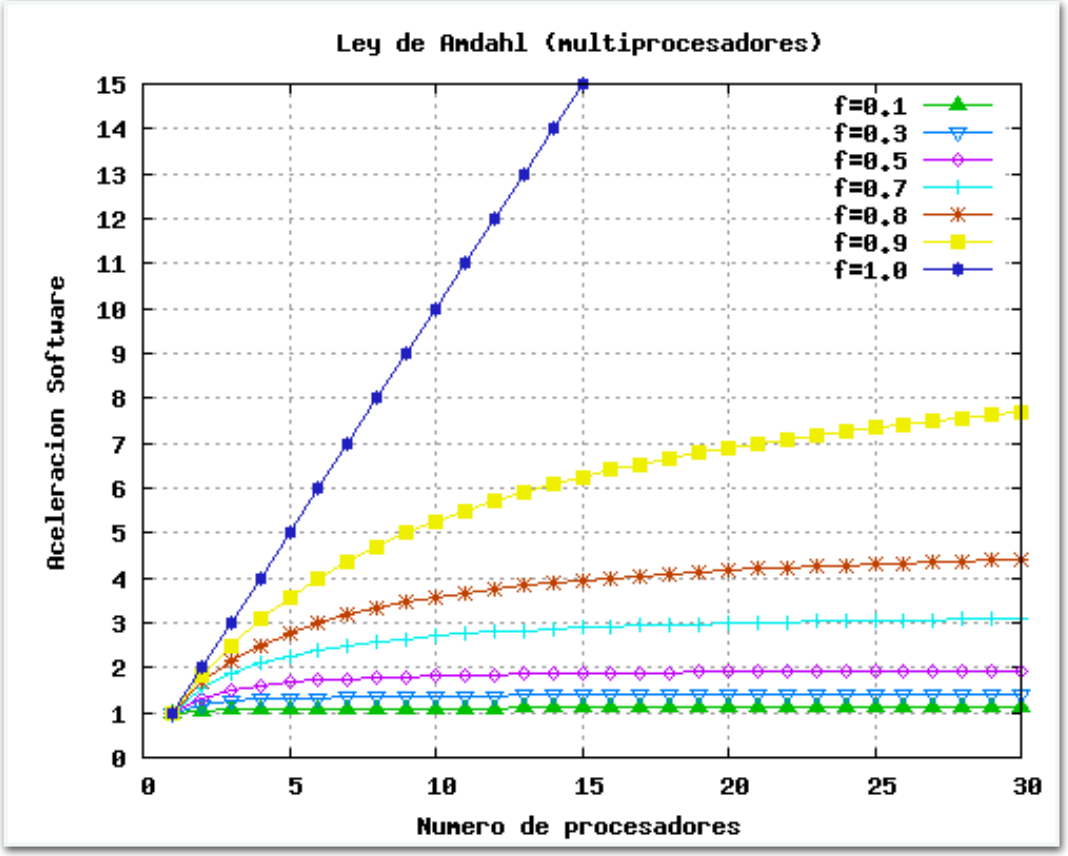
Sea **f** la fracción del programa paralelizable → **1-f** es la fracción del programa secuencial



Ley de Amdahl

$$A = \frac{T_{sin\ paralelizar}}{T_{paralelo}} = \frac{1}{1 - f + \frac{f}{p}}$$

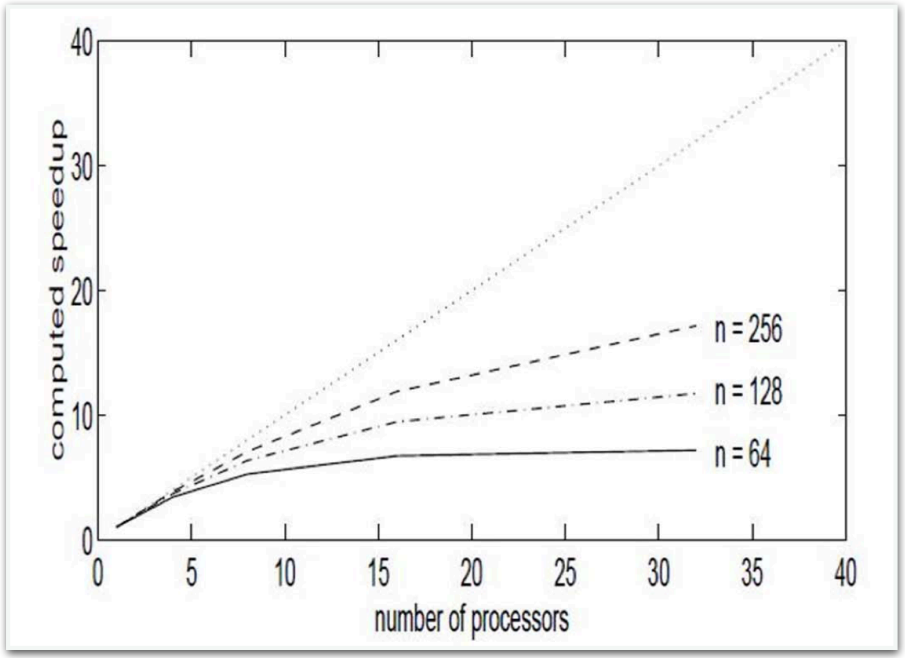
f: fracción del programa que es paralelizable



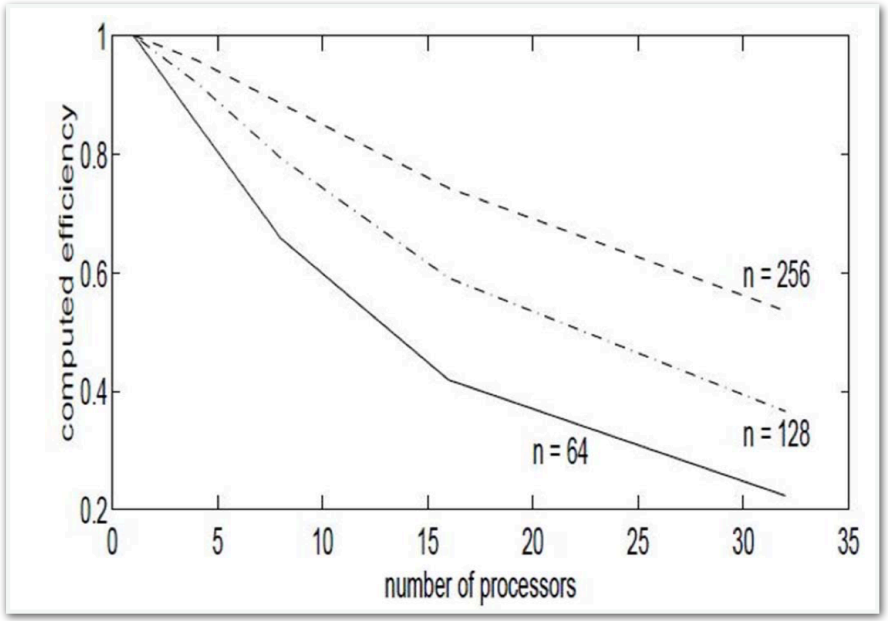
http://webhome.phy.duke.edu/~rgb/brahma/Resources/als/als/node3.html

Ejemplos

Aceleración (SpeeUp)



Eficiencia Paralela



Escalabilidad

Es una medida para ver qué tan bien un programa utiliza múltiples LP para:

- Disminuir el tiempo de ejecución
- Ejecutar problemas más grandes

Cómo medir

Ejecutar programa de control en múltiples LP y medir el tiempo de ejecución.

Determinar la variación de los tiempos de ejecución según la variación de los LP y comparar con el comportamiento ideal.

Comportamiento ideal

Se alcanza en ausencia de cualquier overhead

Tipos de Escalabilidad

Escalabilidad débil

Incrementar el tamaño de problema según la cantidad de LP a utilizar.

¿Puedo ejecutar problemas grandes?

Comportamiento ideal

Tiempo de ejecución es independiente de la cantidad de LP

Escalabilidad fuerte

Fijar un tamaño de problema y aumentar la cantidad de LP

¿Es posible disminuir el tiempo de ejecución?

Comportamiento ideal

Tiempo de ejecución es proporcional a 1/LP

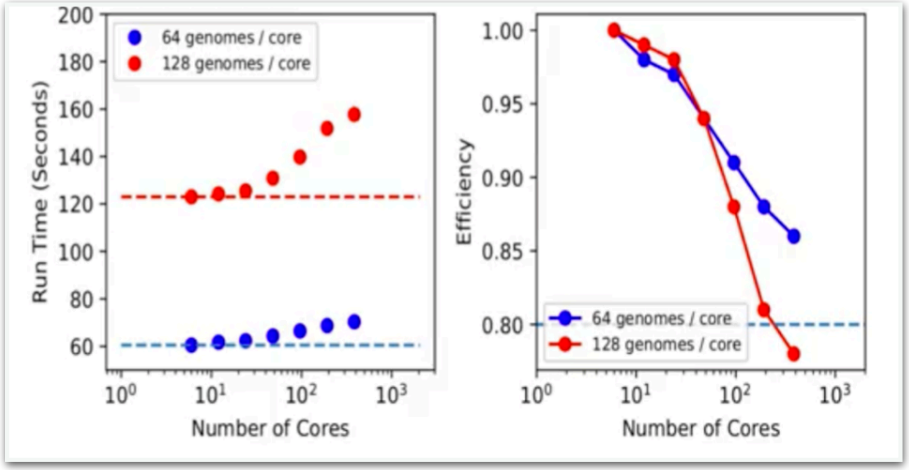
Tipos de Escalabilidad

Escalabilidad débil

Incrementar el tamaño de problema según la cantidad de LP a utilizar.

Cantidad de cómputos por LP es constante

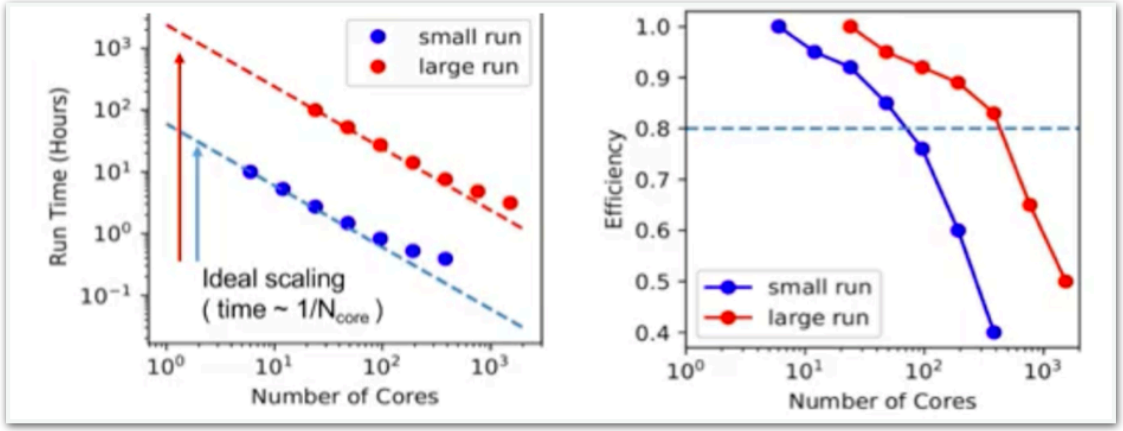
Desempeño ideal: T_{exec} constante



Escalabilidad fuerte

Fijar un tamaño de problema y aumentar la cantidad de LP

Muestra cómo el T_{exec} disminuye a medida que aumentan los LP



Estudio de la escalabilidad

Tomar un tamaño de problema adecuado

Decidir la duración de las pruebas. Debe ser adecuado.

Ejecutar el código en intervalos de tiempo adecuados por cada tamaño de problema y en varios LP.

Por cada experimento, calcular el tiempo de ejecución según las variables independientes que se requieran, determinar speedup, eficiencia, etc.

Graficar los resultados