

Rendimiento

Rendimiento y tiempo de ejecución

- El rendimiento de una computadora A es inversamente proporcional al tiempo de ejecución:

$$\text{Rendimiento}_A = \frac{1}{\text{Tiempo de ejecución}_A}$$

Si la maquina A es n veces más rápida que la máquina B significa que:

$$\frac{RENDIMIENTO_A}{RENDIMIENTO_B} = n$$

$$\frac{TIEMPO_DE_EJECUCIÓN_B}{TIEMPO_DE_EJECUCIÓN_A} = n$$

Ejemplo

Una máquina A ejecuta un programa en 10 segundos y una máquina B ejecuta el mismo programa en 15, ¿Cuánto es mas rápida A que B?

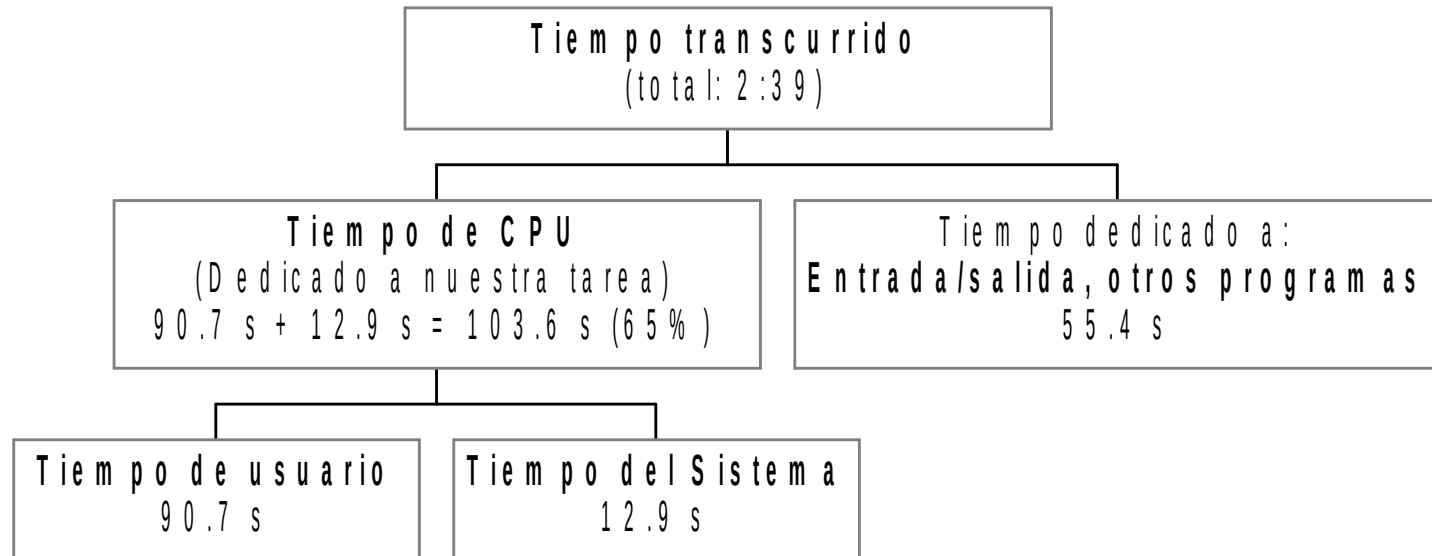
De manera que la razón de rendimiento es:
 $15/10 = 1.5$, y A es por lo tanto, 1.5 veces más rápida que B

Tiempo de ejecución

- Tiempo (segundos)
- Comando Unix/Linux:

time "programa" --> 90.7 u 12.9 s 2:39 65%
(159 s)

¿Qué tiempo medimos?



- El porcentaje de tiempo transcurrido es el de CPU es :

$$(90,7+12,9)/159=0.65$$

Entonces mas de 1/3 se gasto esperando las E/S y/o ejecutando otras cosas.

- En ocasiones no se considera al tiempo de CPU del sistema debido a la poca precisión de los sistemas operativos para obtenerlo, por lo que resultará poco realista considerar el tiempo de CPU del sistema cuando se comparan rendimientos entre máquinas con sistemas operativos diferentes.

Otras medidas

- CPI (ciclos por instrucción)
- MIPS (millones de instrucciones por segundo)
- MFLOPS (Millones de operaciones en punto flotante por segundo)
- MOPS

Rendimiento de los sistemas paralelos

- Rendimiento puede verse como una medida de que tan “bien” un sistema, o los componentes que lo constituyen, lleva a cabo las tareas asignadas.

Escalabilidad.

- Este término se refiere al estudio del cambio en las medidas de rendimiento de un sistema cuando una o varias características del sistema son variadas.

Características que generalmente se cambian es

- número de procesadores.
- Granularidad del problema

Métricas de rendimiento

¿Para qué en ambientes paralelos?

- Conocer cuánto más rápido se ejecuta una aplicación en una computadora paralela.
- Conocer cuál es el beneficio obtenido cuando se usa paralelismo y cuál es la aceleración que resulta por el uso de dicho paralelismo.

Métricas de rendimiento

Tiempo de Ejecución:

Tiempo en un programa **secuencial**, tiempo que transcurre entre el comienzo y la finalización de la ejecución del programa.

Tiempo en un programa **paralelo**, el tiempo que transcurre entre el comienzo y la finalización del último proceso que termina la ejecución.

Métricas de rendimiento

Aceleración (Speedup)

- La aceleración puede ser definida como el ratio

- $$S = \frac{\text{Tiempo Ejecución Secuencial}}{\text{Tiempo Ejecución Paralelo}}$$

- Ejemplo: Si el programa secuencial se ejecuta en 6 seg, y el paralelo en 2 seg, la aceleración es 3

Pintando una casa

- Supongamos que deseamos pintar una casa que tiene 10 habitaciones. Si me demoro 1 día por habitación, la casa completa me demorara 10 días.
- Si contrato un equipo de 10 personas, cada una podría pintar una pieza y la casa estaría terminada en poco más de 1 día; no exactamente 1 día, pues el equipo demorara un pequeño tiempo en asignación de piezas,
- Si contrato 20 personas, la casa estaría terminada en un poco más de 5 días.
- Si contrato 200?

Límite Speed-UP

- Todo programa paralelo tiene una parte secuencial que eventualmente limita la aceleración que se puede alcanzar en una plataforma paralela.

Algunos factores que limitan el speed-up son:

- Overhead de software
- Des-balance de carga
- Overhead de comunicación/sincronización
- Overhead secuencial que no puede paralelizarse

Ley de Amdahl

- La ley de Amdahl es un modelo matemático que describe la relación entre la aceleración esperada de la implementación paralela de un algoritmo y la implementación serial del mismo algoritmo.
- Indica la mejora de rendimiento que se puede obtener al incrementar las unidades de procesamiento.

Ley de Amdahl

$$S_n = \frac{T_s + T_p}{T_s + \frac{T_p}{n}}$$

Ley de Amdahl

El speedup puede ser reescrito como

$$\begin{aligned} S(n) &= \frac{T_s + T_p}{T_s + T_p/n} \\ &= \frac{1}{\frac{T_s}{T_s + T_p} + \frac{T_p}{(T_s + T_p)n}} \\ &= \frac{1}{f + \frac{(1-f)}{n}} \\ &\leq \frac{1}{f} \end{aligned}$$

Es decir, el speedup alcanzable está limitado por su parte secuencial

Ley de Amdahl

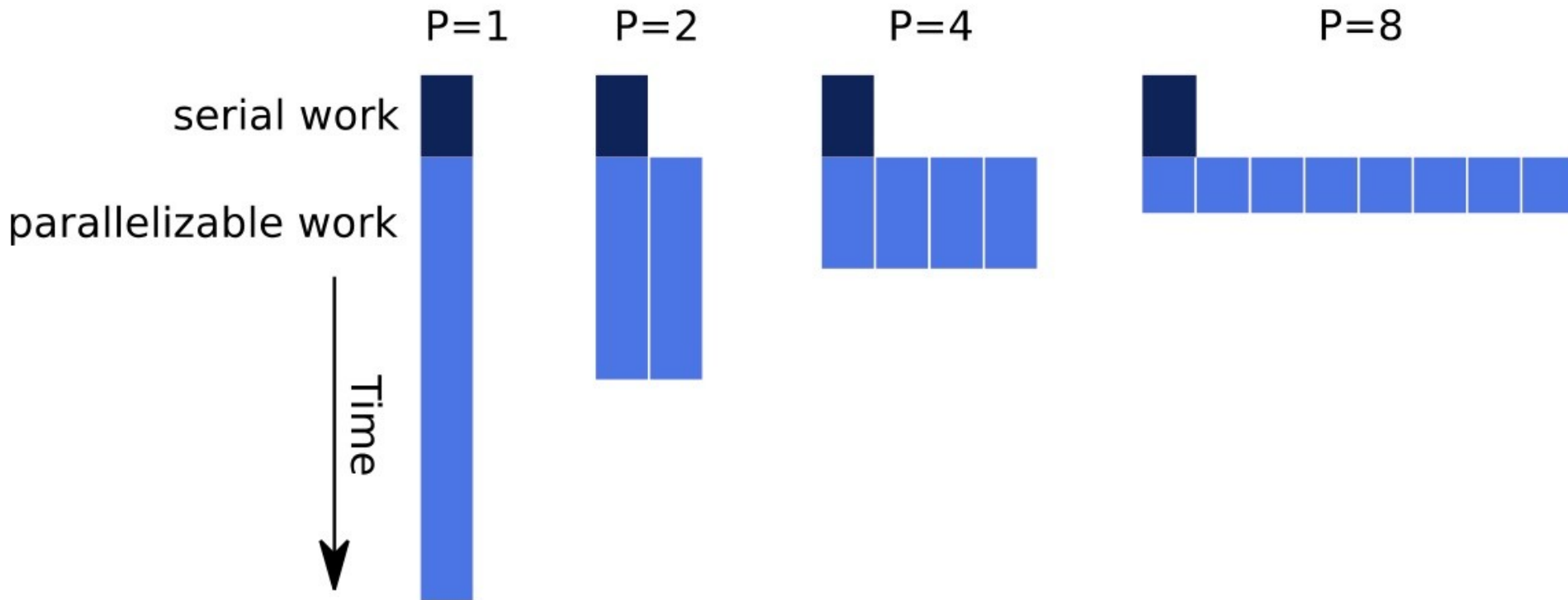
- Amdahl concluyó que las computadoras paralelas no podrían entregar *speedups* cada vez mayores, pues el speedup estaba limitado por la porción secuencial de los problemas, independientemente de la computadora paralela.
- La ley de Amdahl es una ley de speedup para cargas de trabajo fijas.

Ley de Gustafson-Barsis

- Speed-up escalable
- Se utiliza cuando el tamaño del problema puede incrementar al incrementar el numero de procesadores

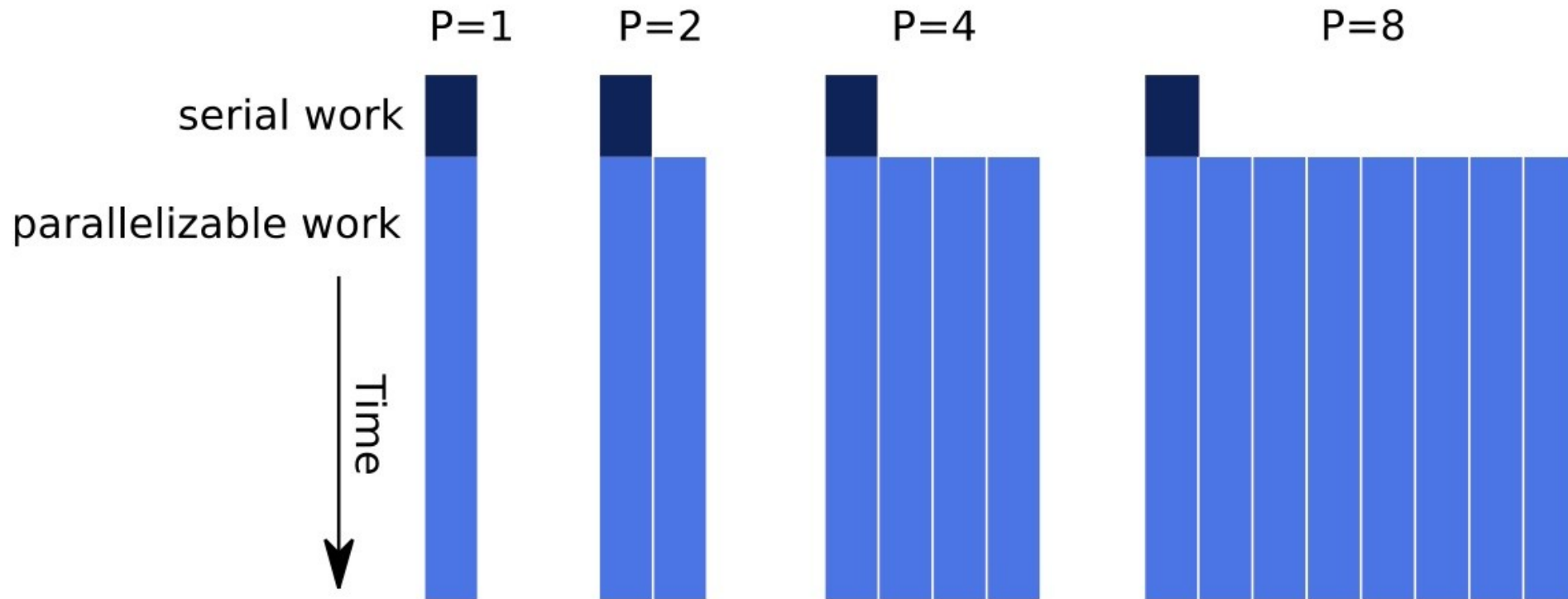
Amdahl vs Gustafson-Baris

Amdahl

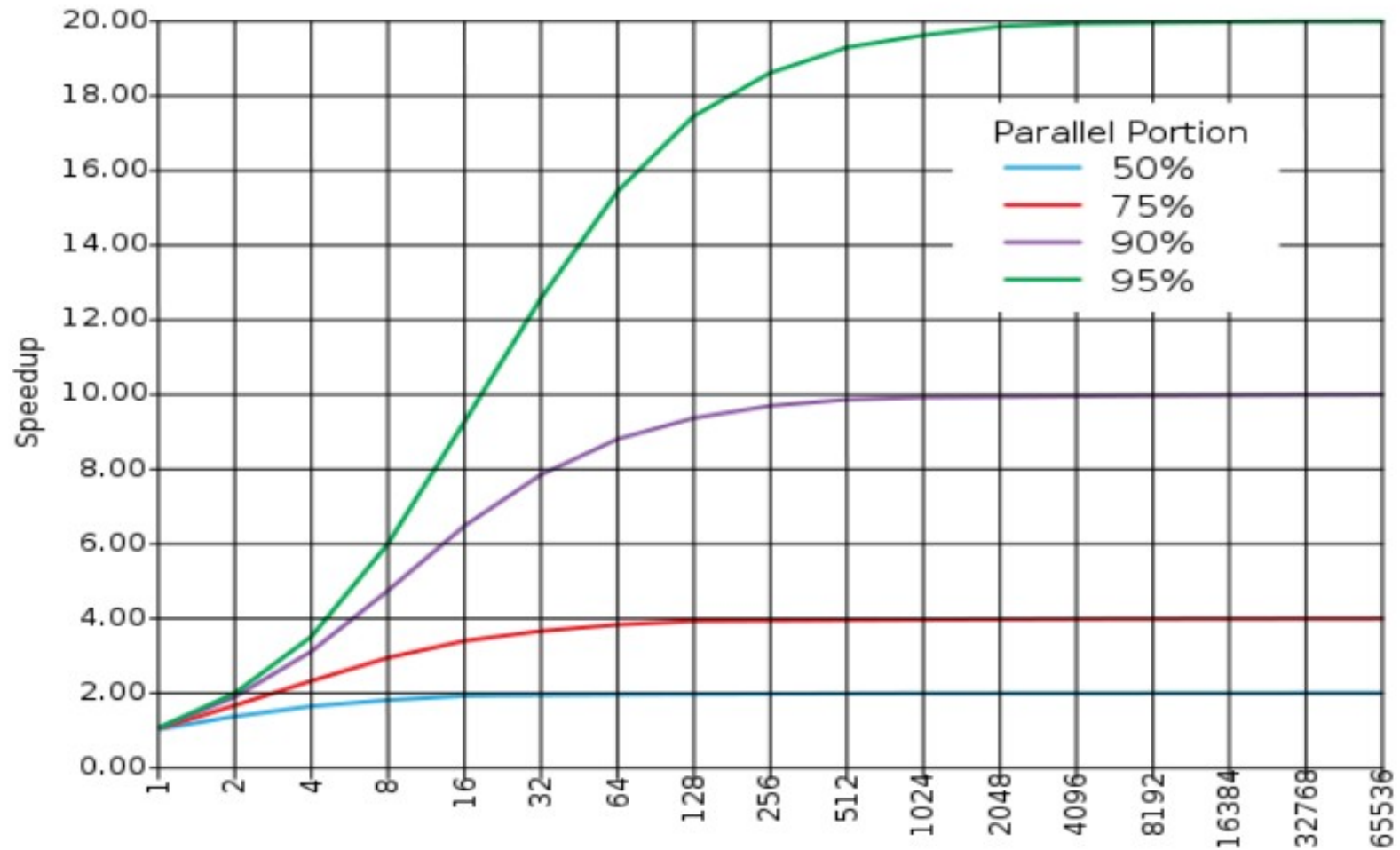


Amdahl vs Gustafson-Baris

Gustafson-Baris



Amdahl's Law



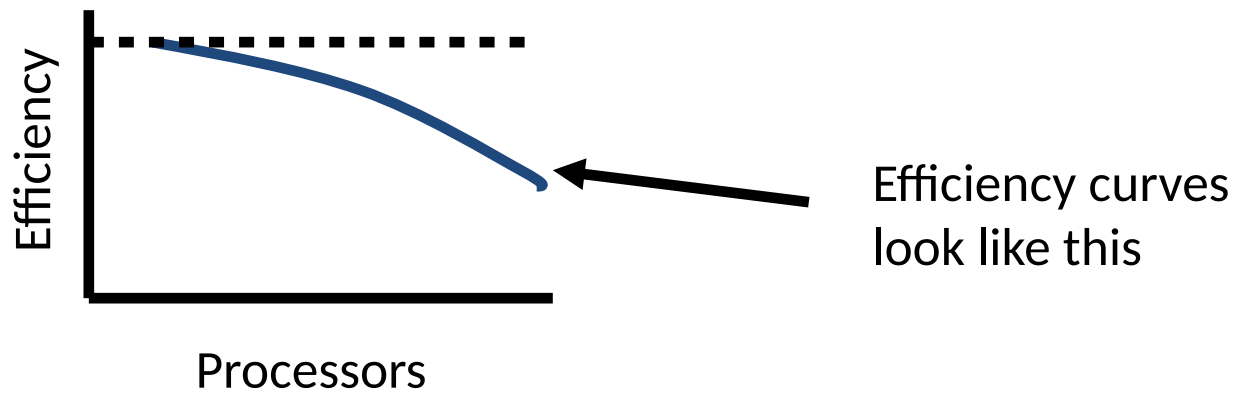
Eficiencia

- La Eficiencia de un sistema paralelo se define como:

$$E = \frac{S_N}{N}$$

- La eficiencia permite conocer que tan bien se están aprovechando los recursos de la máquina.
- Medida de utilización del procesador
- Para cargas de trabajo fijas $0 < E < 1$
- Para cargas de trabajo escalable ley de Gustafon, es posible que $E > 1$

Eficiencia



Ejercicio

- Obtener el speedup teórico para fracciones paralelas de 50%, 90% y 99% y un número de unidades de procesamiento de 10,100,1000 y 10000 (graficar)

n	P=0.50	P=0.90	P=0.99
10			
100			
1000			
10000			

Métrica de Karp-Flatt o fracción secuencial

- Se utiliza para determinar si la eficiencia se decrementa debido a:
 - Cómputo en componentes secuenciales
 - Incremento en overhead paralelo

Interpretación

- Si “f” se mantiene + o - constante al aumentar el número de up, entonces el speedup está restringido por el cómputo en la parte secuencial.
- Si “f” se incrementa al aumentar el número de up el speedup está restringido por el overhead paralelo , tal como:
- A menudo una combinación de ambos factores.

Ejemplos

- Suponer que del análisis de un programa se obtiene

procesadores	Speed-up	eficiencia
2	1.5	
3	1.8	
4	2	

Otro ejemplo

Processors	Speedup	Efficiency	f
2	1.87		
3	2.60		
4	3.16		

- Se puede predecir el speedup para 6 procesadores
- Extrapolando f a 0.11 y después calcular S_n