

2º curso / 2º cuatr.

Grado en
Ing. Informática

Arquitectura de Computadores

Tema 2

Programación paralela

Material elaborado por los profesores responsables de la asignatura:

Mancia Anguita – Julio Ortega

Licencia Creative Commons



ugr

Universidad
de Granada

ETSIIT

Escuela Técnica Superior
de Ingenierías Informática
y de Telecomunicación



ATC

Departamento de Arquitectura
y Tecnología de Computadores
UNIVERSIDAD DE GRANADA



Lecciones

- Lección 4. Herramientas, estilos y estructuras en programación paralela
- Lección 5. Proceso de paralelización
- Lección 6. Evaluación de prestaciones en procesamiento paralelo
 - Ganancia en prestaciones y escalabilidad
 - Ley de Amdahl
 - Ganancia escalable

Bibliografía

➤ Fundamental

- Secc. 7.5. J. Ortega, M. Anguita, A. Prieto. “Arquitectura de Computadores”. ESII/C.1 ORT arq

Contenido Lección 6

- Ganancia en prestaciones y escalabilidad
- Ley de Amdahl
- Ganancia escalable

Evaluación de prestaciones

➤ Medidas usuales

➤ Tiempo de respuesta

- Real (*wall-clock time, elapsed time*) (/usr/bin/time)
- Usuario, sistema, CPU time = user + sys

➤ Productividad

➤ Escalabilidad

➤ Eficiencia

- Relación prestaciones/prestaciones máximas
- Rendimiento = prestaciones/nº_recursos
- Otras: Prestaciones/consumo_potencia,
prestaciones/área_ocupada

Ganancia en prestaciones. Escalabilidad

Ganancia en prestaciones:

$$S(p) = \frac{\text{Prestaciones}(p)}{\text{Prestaciones}(1)} = \frac{T_s}{T_p(p)}$$

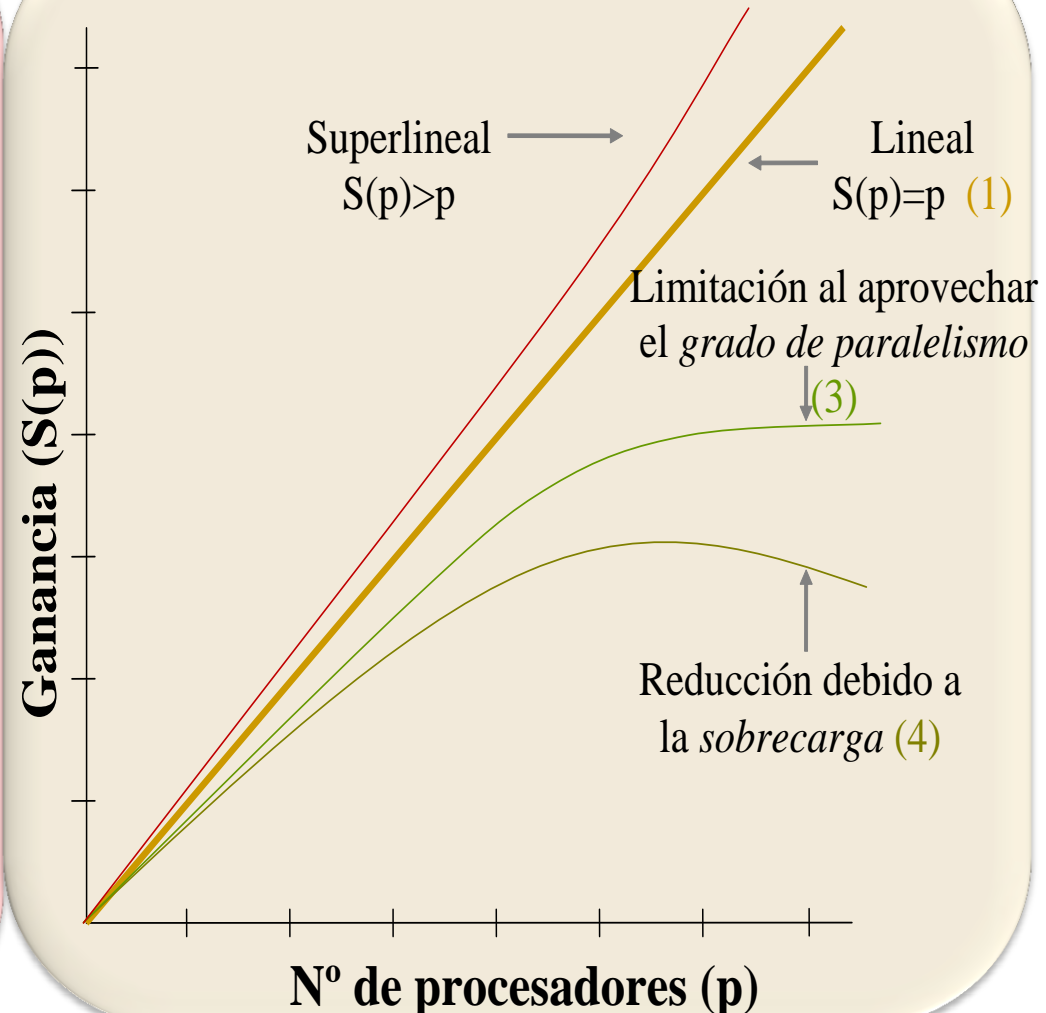
Ganancia en velocidad (Speedup)

$$T_p(p) = T_c(p) + T_o(p)$$

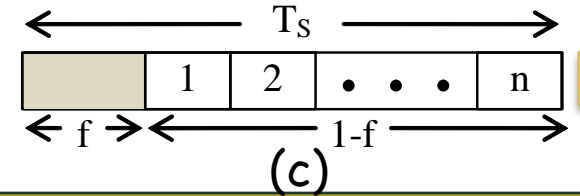
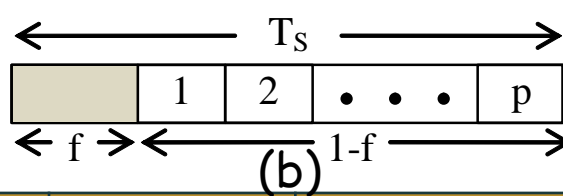
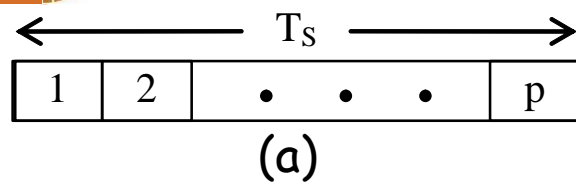
Sobrecarga (Overhead):

- Comunicación/sincronización.
- Crear/terminar procesos/threads.
- Cálculos o funciones no presentes en versión secuencial.
- Falta de equilibrado.

Escalabilidad:

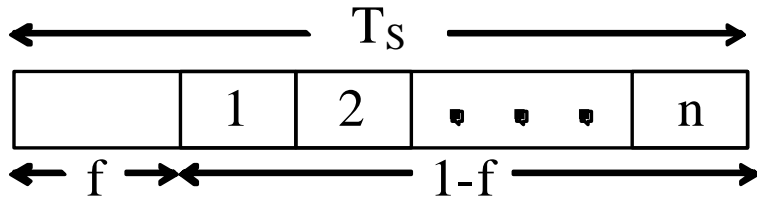


Ganancia en prestaciones. Ganancia máxima

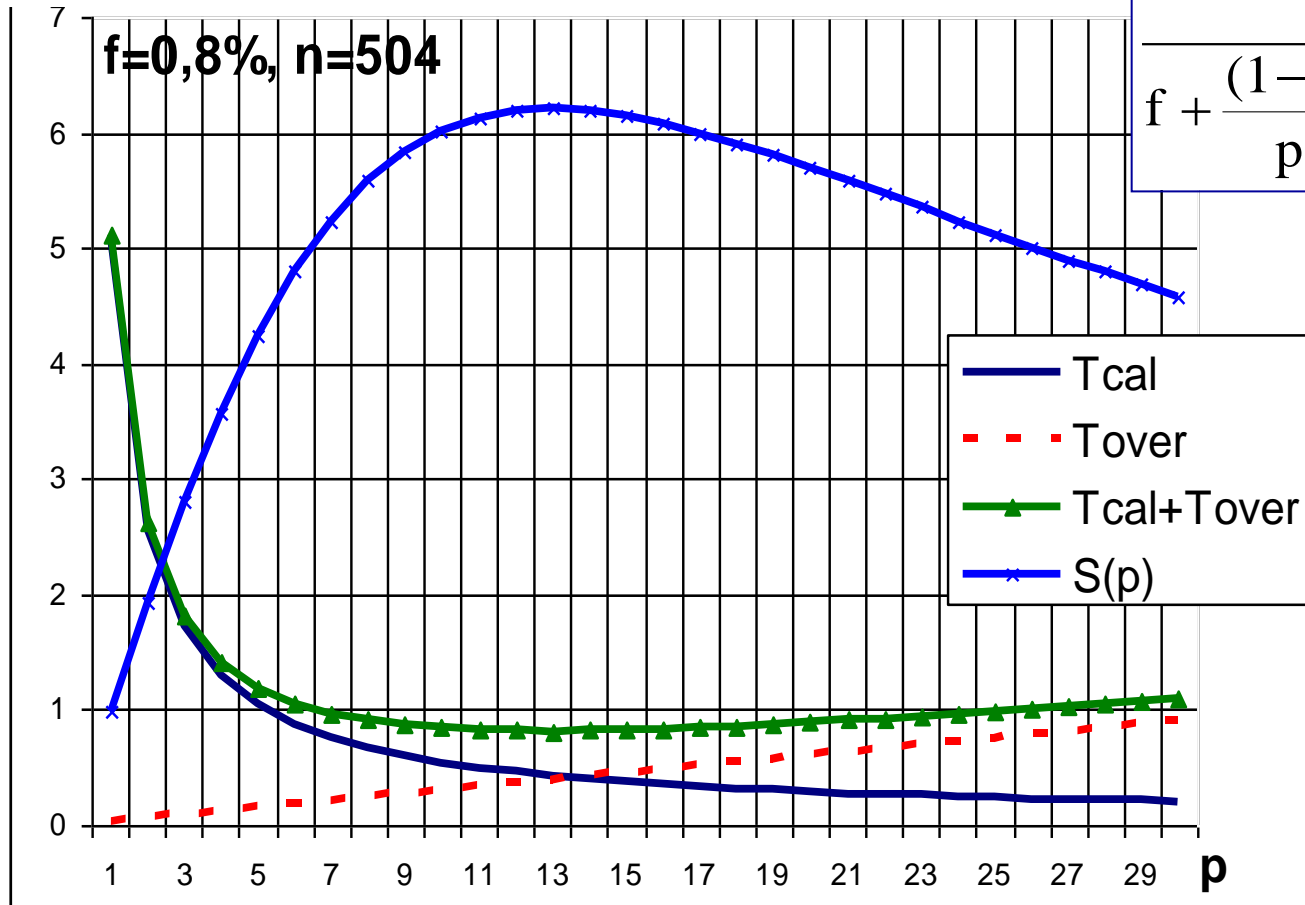


Modelo código	Fracción no paral. en T_s	Grado paralelismo	Overhead	Ganancia en función del número de procesadores p con T_s constante
(a)	0	ilimitado	0	$S(p) = \frac{T_s}{T_P(p)} = p$ Ganancia lineal (1)
(b)	f	ilimitado	0	$S(p) = \frac{1}{f + \frac{(1-f)}{p}} \xrightarrow{p \rightarrow \infty} \frac{1}{f}$ (2)
(c)	f	n	0	$S(p) = \frac{1}{f + \frac{(1-f)}{p}} \xrightarrow{p=n} \frac{1}{f + \frac{(1-f)}{n}}$ (3)
(b)	f	ilimitado	Incrementa linealmente con p	$S(p) = \frac{1}{f + \frac{(1-f)}{p} + \frac{T_O(p)}{T_s}} \xrightarrow{p \rightarrow \infty} 0$ (4)

Número de procesadores óptimo



$$S(p) = \frac{T_s}{T_P(p)} = \frac{T_s}{T_C(p) + T_O(p)} = \frac{1}{f + \frac{(1-f)}{p} + \frac{T_O(p)}{T_s}}$$



$$T_C(p) = O\left(\frac{1}{p}\right)$$

$$T_O(p) = O(p)$$

```
For(i=0;i<n;i++) {
    Código para i
}
```


Contenido Lección 6

- Ganancia en prestaciones y escalabilidad
- Ley de Amdahl
- Ganancia escalable

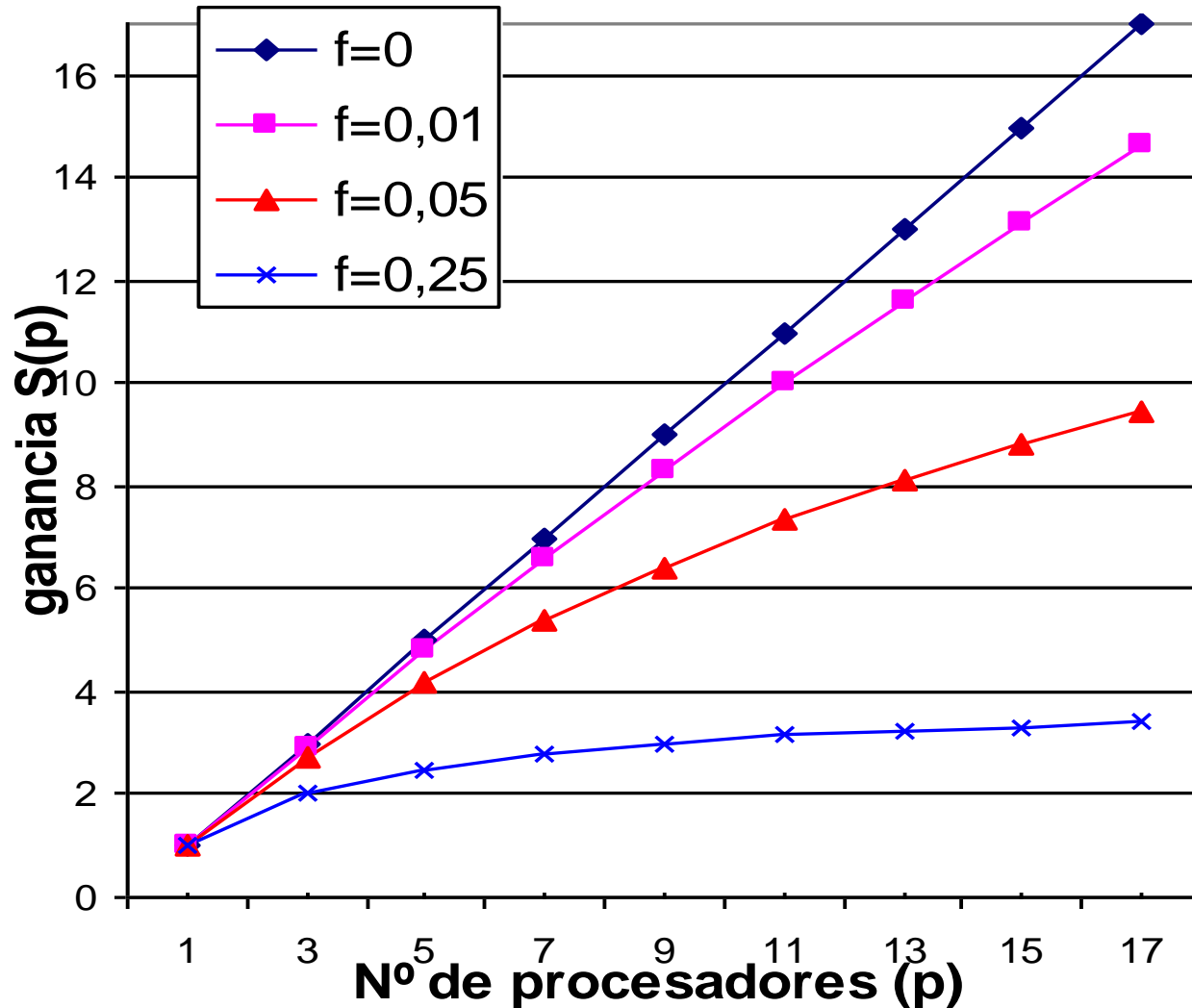
Ley de Amdahl

- Ley de Amdahl: la ganancia en prestaciones utilizando p procesadores está limitada por la fracción de código que no se puede paralelizar (2):

$$S(p) = \frac{T_s}{T_P(p)} \leq \frac{T_s}{f \cdot T_s + \frac{(1-f) \cdot T_s}{p}} = \frac{p}{1 + f(p-1)} \rightarrow \frac{1}{f} (p \rightarrow \infty)$$

- S : Incremento en velocidad que se consigue al aplicar una mejora. (paralelismo)
- p : Incremento en velocidad máximo que se puede conseguir si se aplica la mejora todo el tiempo. (número de procesadores)
- f : fracción de tiempo en el que no se puede aplicar la mejora. (fracción de t. no paralelizable)

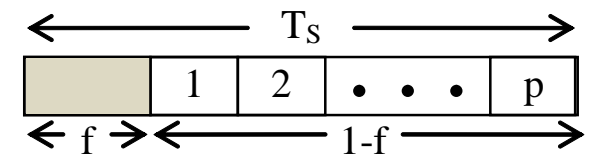
Ley de Amdahl



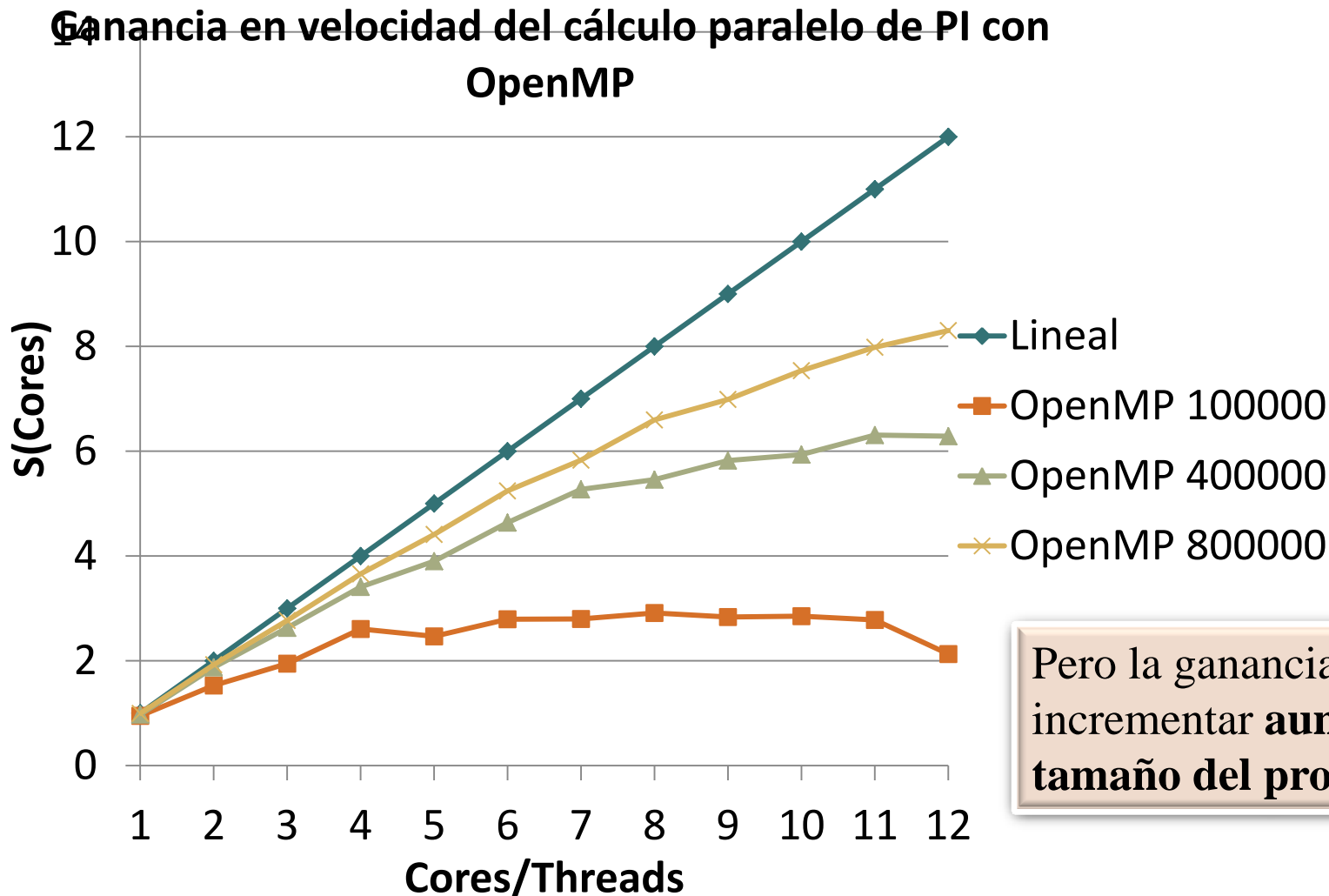
$$S(p) = \frac{p}{1 + f(p-1)}$$
$$S(p) \rightarrow \frac{1}{f} (p \rightarrow \infty)$$

Es pesimista.

Nos dice que la *escalabilidad* está limitada por f (*fracción de T_s que no se puede paralelizar*)



Ganancia escalable

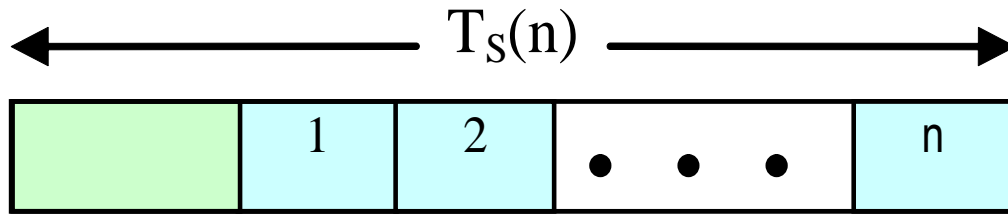


Pero la ganancia se puede incrementar **aumentando el tamaño del problema**

Contenido Lección 6

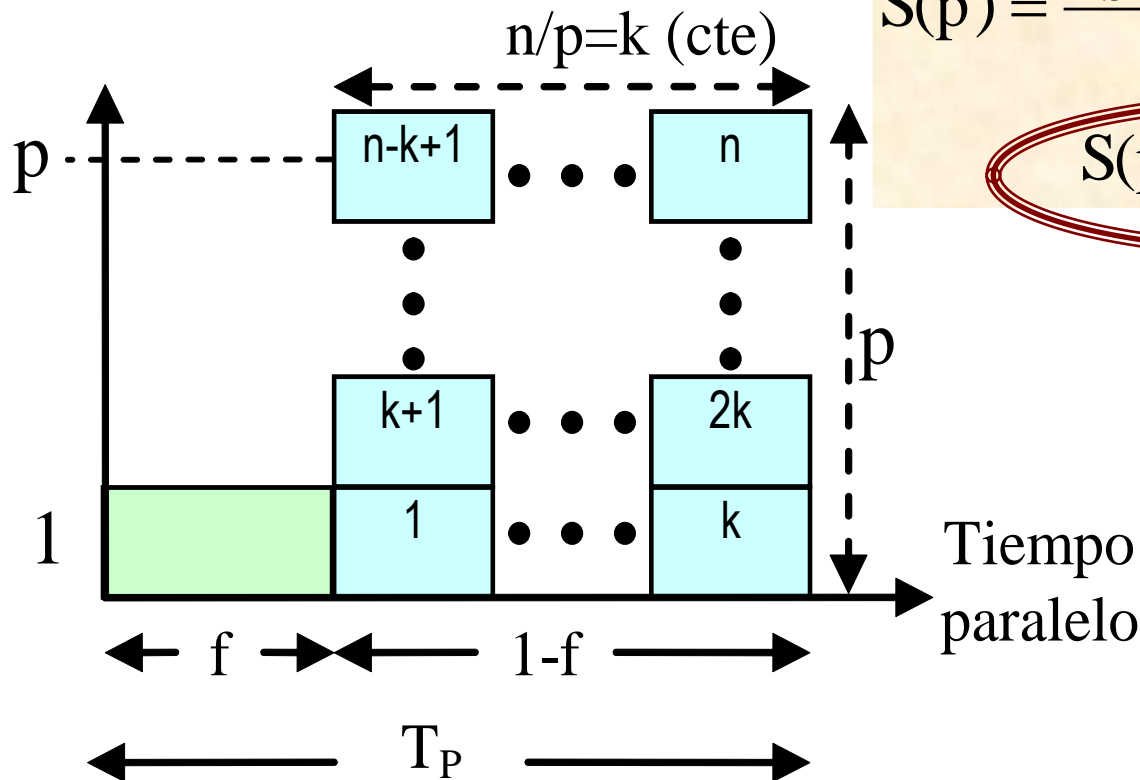
- Ganancia en prestaciones y escalabilidad
- Ley de Amdahl
- Ganancia escalable

Ganancia escalable o Ley de Gustafson



Amdahl mantiene constante T_S ,
Gustafson mantiene constante T_P

Procesadores



$$S(p) = \frac{T_S(n = kp)}{T_P} = \frac{fT_P + p(1-f)T_P}{T_P}$$

$$S(p) = p(1-f) + f$$

Para ampliar ...

➤ Páginas Web:

- http://en.wikipedia.org/wiki/Parallel_computing

➤ Artículos en revistas

- Gene M. Amdahl. 1967. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference (AFIPS '67 (Spring))*. ACM, New York, NY, USA. Disponible en línea (biblioteca UGR):
<http://doi.acm.org/10.1145/1465482.1465560>
- John L. Gustafson. 1988. Reevaluating Amdahl's law. *Commun. ACM* 31, 5 (May 1988), 532-533. Disponible en línea (biblioteca UGR):
<http://doi.acm.org/10.1145/42411.42415>