

---

# **Tema**

# **Modelos de Rendimiento**

José Ranilla Pastor

ranilla@uniovi.es

<http://lear.inforg.uniovi.es/CyP>

- 1. Motivación**
- 2. Métricas absolutas: Tiempo de ejecución**
  - **Definición**
  - **Partes fundamentales**
- 3. Métricas Relativas**
  - **Incremento de Velocidad (SpeedUp)**
  - **Eficiencia**
  - **Coste**
- 4. Un ejemplo**

# Motivación

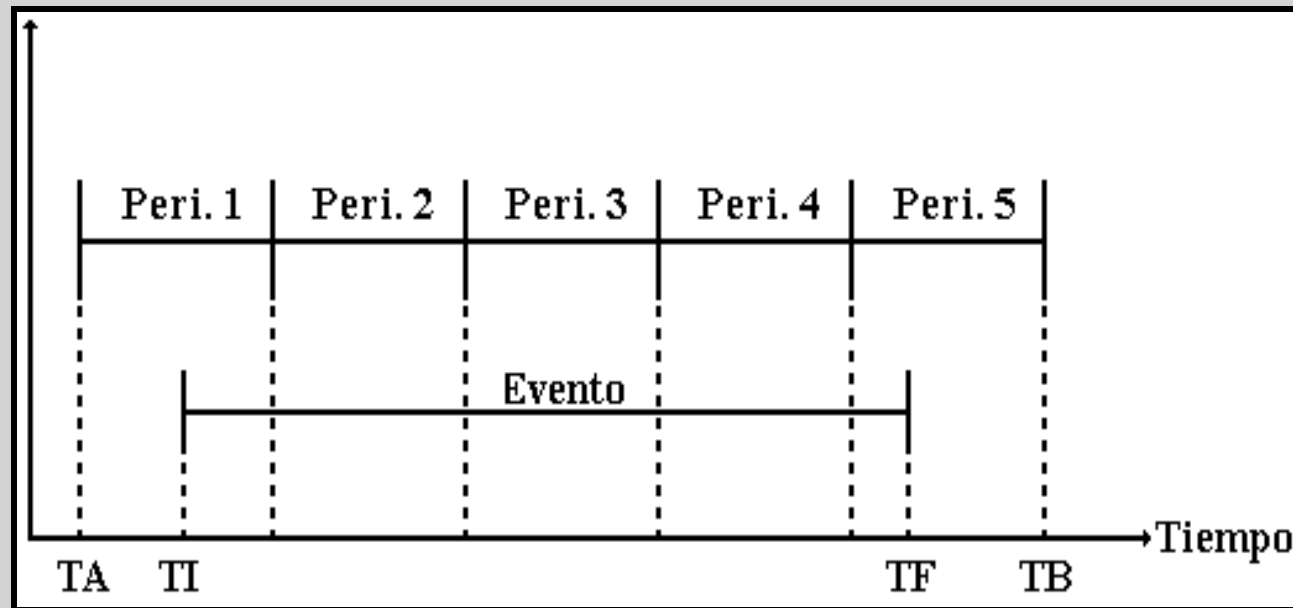
---

- Disponer de herramientas que permitan evaluar las prestaciones de los algoritmos paralelos de forma fiable y precisa.
- Dispones de una herramienta que permita comparar el rendimiento de los algoritmos paralelos (y secuenciales).
- Modelo de computación paralelo adoptado: MIMD-MD basado en paso de mensajes. Parte de lo aquí expuesto es aplicable a otros modelos.
- Enfoque muy básico.

# Tiempo de ejecución

## Paso previo: Relativo a cómo medir

- El error máximo que se comete al estimar la duración de un evento pertenece al intervalo  $(-t, t)$ , siendo  $t$  el periodo.



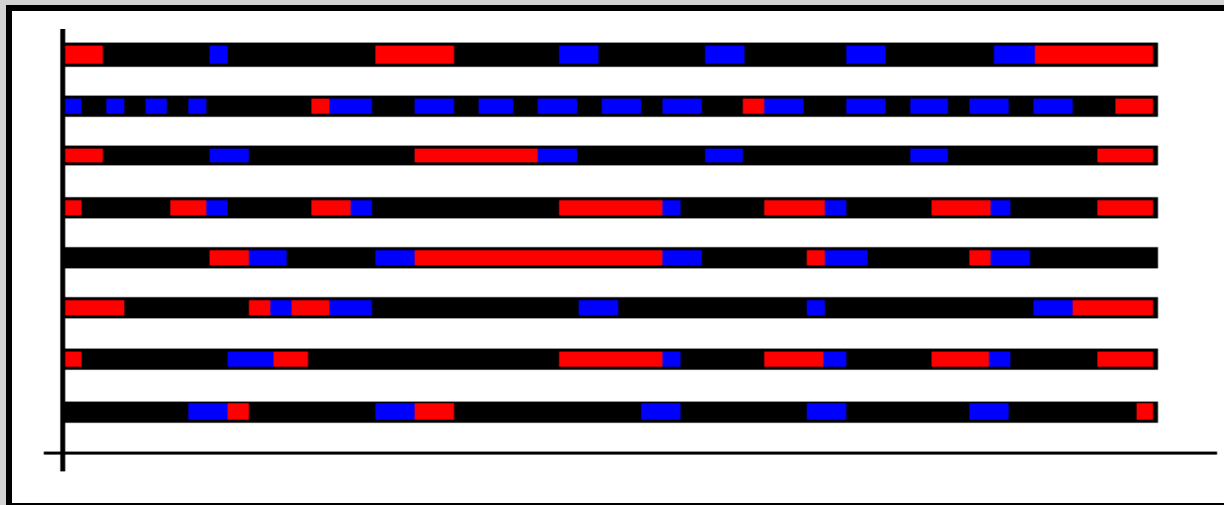
- Repetir los experimentos un número de veces adecuado controlando los posibles efectos colaterales (*nan*, excepciones, etc.) en los *flops* realizados.

# Tiempo de ejecución

---

## Definición

- Tiempo transcurrido desde que el primer procesador inicia la ejecución del algoritmo hasta que el último la finaliza.
- Depende de varios factores: computación, comunicaciones, solapamientos, esperas, etc.



# Tiempo de ejecución

---

## En fórmulas

$$T = T_{\text{comp}}^j + T_{\text{comu}}^j + T_{\text{idle}}^j$$

## O bien

$$T = \frac{1}{p} \left( \sum_{i=1}^p T_{\text{comp}}^i + \sum_{i=1}^p T_{\text{comu}}^i + \sum_{i=1}^p T_{\text{idle}}^i \right)$$

## Globalmente

$$T = T_{\text{comp}} + T_{\text{comu}} + T_{\text{idle}} - T_{\text{solapamiento}}$$

# Tiempo de ejecución

---

## Algoritmos síncronos

$$T = T_{\text{comp}} + T_{\text{comu}} + T_{\text{idle}} - 0$$

$$T \cong T_{\text{comp}} + T_{\text{comu}}$$

## Algoritmos asíncronos

$$T = \max(T_{\text{comp}}, T_{\text{comu}}) + T_{\text{idle}}$$

Hipótesis  
de Trabajo



$$\frac{T_{\text{comp}} + T_{\text{comu}}}{2} \leq \max(T_{\text{comp}}, T_{\text{comu}}) \leq T_{\text{comp}} + T_{\text{comu}}$$

# Tiempo de ejecución

---

## Tiempo de Computación

- Es el tiempo que el algoritmo emplea realizando cálculos. Generalmente se expresa en *flops*.

## Según el enfoque teórico/empírico

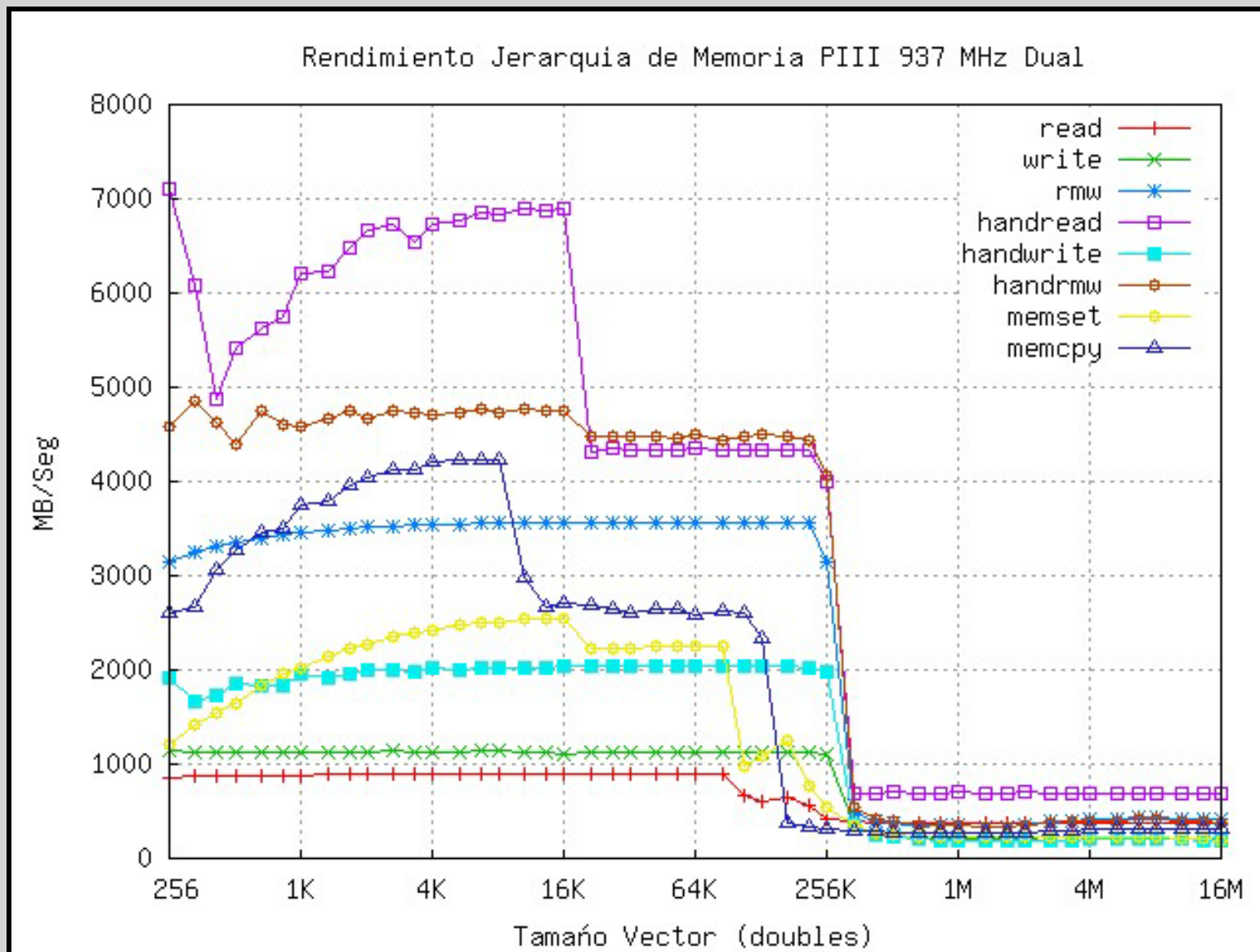
- Depende del tamaño del problema y se expresa en función de él, y del número de procesadores.
- Depende del número de tareas por procesador, de las características de los procesadores, de los subsistemas de memoria, etc.

## No olvidar

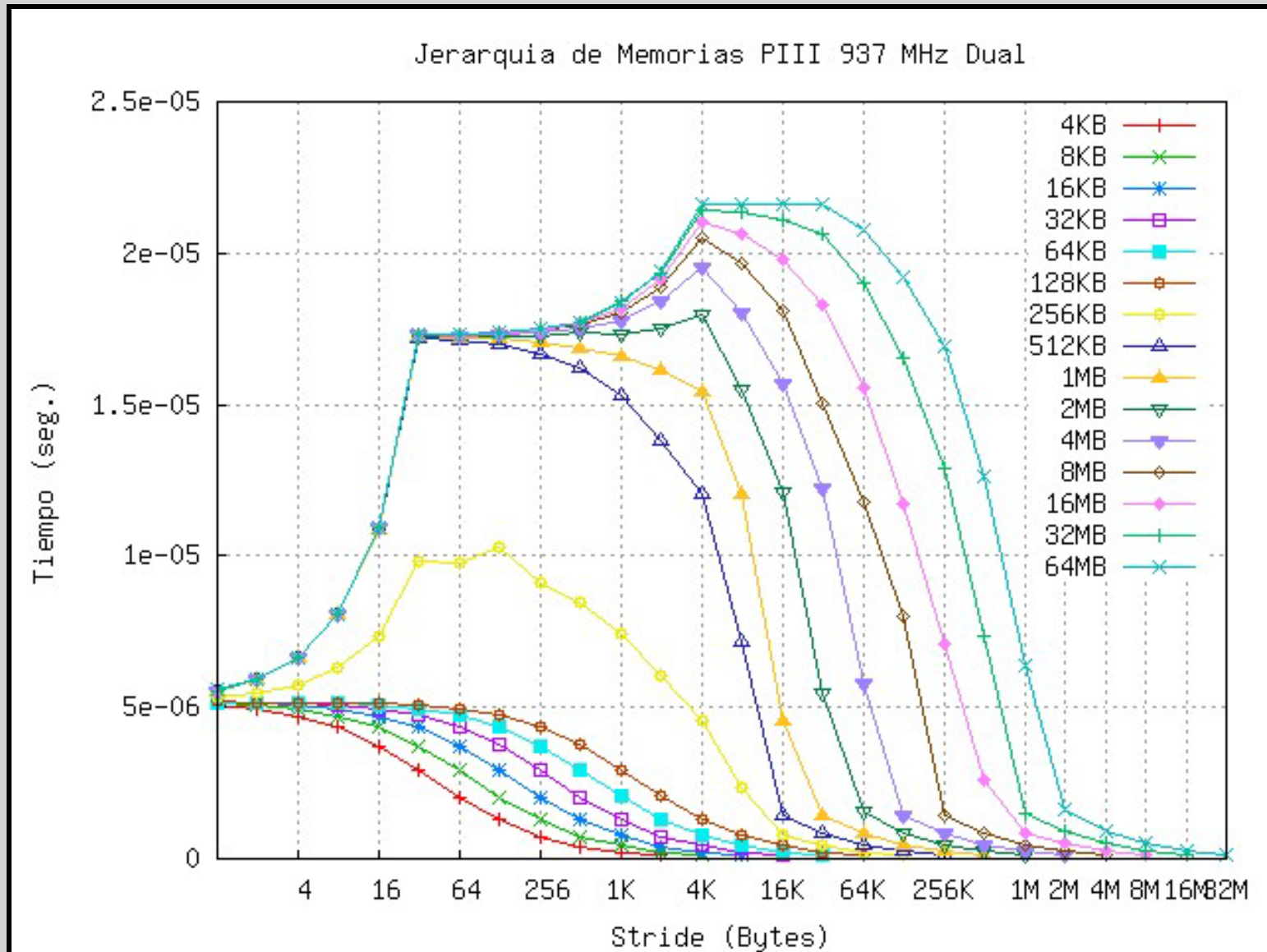
- Su comportamiento dinámico (efecto variable) por causas ajenas al algoritmo pero inherentes al sistema.  $\Rightarrow$



# Tiempo de ejecución



# Tiempo de ejecución



# Tiempo de ejecución

---

## Tiempo de Comunicación

- Tiempo que las tareas emplean en enviar/recibir.

## Tipos

- Internas
- Externas

(usamos el mismo para ambos tipos de comunicaciones)

## Modelos

- **PRAM**: poco realista.
- **LogP** y **LogGP**: complejos para este curso.
- $\alpha$ - $\beta$ : el que usaremos.

# Tiempo de ejecución

## Latencia y Ancho de Banda o $\alpha$ - $\beta$

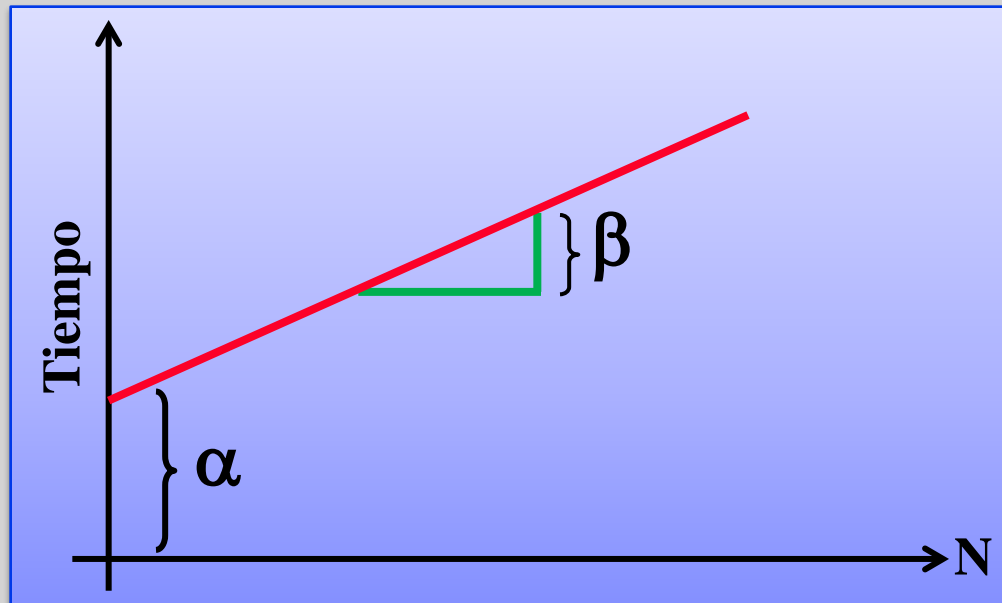
- Parámetros

$\alpha$  Latencia de la red.

$\beta$  coste por *word* (1 / ancho de banda).

$$T = \alpha + N\beta$$

$$\alpha \gg \beta$$



# Tiempo de ejecución

---

## Algunos valores *experimentales*

Entorno	$\alpha$	$\beta$
T3E/MPI	6.7	0.003
IBM/MPI	7.6	0.004
Quadrics/MPI	7.3	0.005
Myrinet/MPI	7.2	0.006
Dolphin/MPI	7.8	0.005
GigE/MPI	5.9	0.009

$\alpha$  en  $\mu\text{seg.}$

$\beta$  en  $\mu\text{seg. por Byte}$

## Problemas del modelo $\alpha$ - $\beta$


- Conduce a estimaciones *poco finas* en:
  - Entornos heterogéneos.
  - En computadoras donde las comunicaciones internas están altamente optimizadas.
- Necesidad de que los valores de las constantes sean reales.

## Algunas recomendaciones para el modelo $\alpha$ - $\beta$

- Constantes con efecto *variable* según el tipo de red, el tráfico, ...
- Disminuir los efectos de la constante de establecimiento. Un mensaje grande es más *barato* que muchos pequeños:

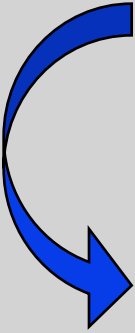
$$\alpha + n\beta \ll n(\alpha + \beta)$$

## Problemas del Tiempo de ejecución

- 
- Métrica Absoluta
  - Depende, al menos, del tamaño del problema  $\Rightarrow$  **Normalizar**

**Dificultad para comparar algoritmos**

## Estas alternativas son/representan

- 
- Métricas Relativas
  - Estiman la efectividad con que los algoritmos usan los recursos

**Permiten comparar algoritmos**

# Speedup, Eficiencia

---

## Definición

- El incremento de velocidad (o *Speedup*) de un algoritmo paralelo cuando se ejecuta sobre  $p$  procesadores es:

$$S(n;p) = \frac{\text{Tiempo de ejecución sobre un procesador}}{\text{Tiempo de ejecución en } p \text{ procesadores}}$$

**Representa la bondad del diseño paralelo**

- El incremento de velocidad (o *Speedup*) de un algoritmo paralelo cuando se ejecuta sobre  $p$  procesadores respecto al mejor algoritmo secuencial es:

$$S'(n;p) = \frac{\text{Tiempo de ejecución del secuencial más rapido}}{\text{Tiempo de ejecución del paralelo en } p \text{ procesadores}}$$

**Indica la eficacia (prestaciones) del algoritmo paralelo**



## Definición

- La eficiencia de un algoritmo paralelo, respecto a sí mismo, es:

$$E(n;p) = \frac{S(n;p)}{p}$$

- Siendo la eficiencia respecto al mejor algoritmo secuencial:

$$E'(n;p) = \frac{S'(n;p)}{p}$$

Evidentemente:  $E'(n;p) \leq E(n;p) \leq 1$

## Objetivo

- Algoritmos óptimos  $E'(n;p) \in \theta(1)$
- Algoritmos eficientes  $E'(n;p) \in \Omega\left(\frac{1}{\log n}\right)$

## Definición

- Relación entre el tiempo de ejecución del secuencial óptimo y el tiempo de ejecución paralelo, multiplicado por el número de procesadores.

$$C(n;p) = \frac{T(n;1)}{T(n;p)}p$$

## Propiedad

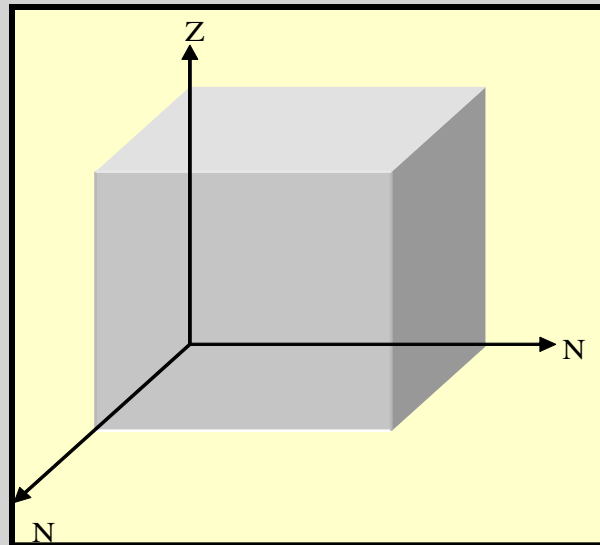
- Un algoritmo paralelo es de **coste-óptimo**, eficiencia máxima, si el tiempo para resolver un problema en una computadora paralela es **proporcional** al tiempo de resolución del mismo problema por el algoritmo secuencial óptimo.

# Un ejemplo

---

## El problema

- Dominio definido por una malla de  $N*N*Z$  puntos.

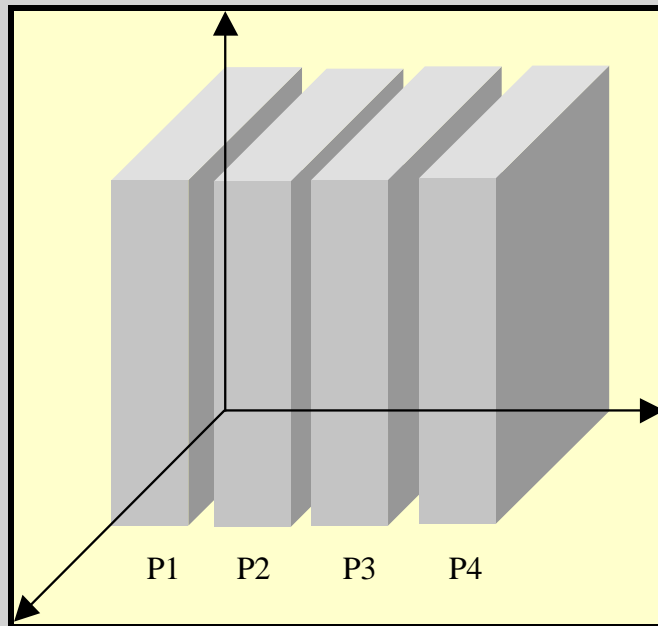


- Sea  $t_s$  la constante que denota el coste computacional para actualizar cada punto de la malla en cada etapa.

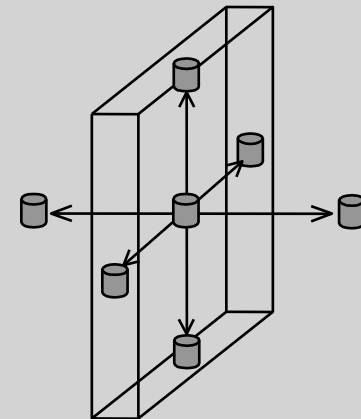
# Un ejemplo

## El problema

- Descomposición y/o mapeado realizado:  $(N*N*Z)/P$  puntos por procesador.



Comunicaciones locales.



# Un ejemplo

---

## El estudio del rendimiento

- Tiempo de computación
  - Desde la versión secuencial

$$T_{\text{comp}} = \frac{T_{\text{Secuencial}}}{P} = \frac{t_s(N \cdot N \cdot Z)}{P} = \frac{t_s(N^2 \cdot Z)}{P}$$

- Observando un procesador arbitrario

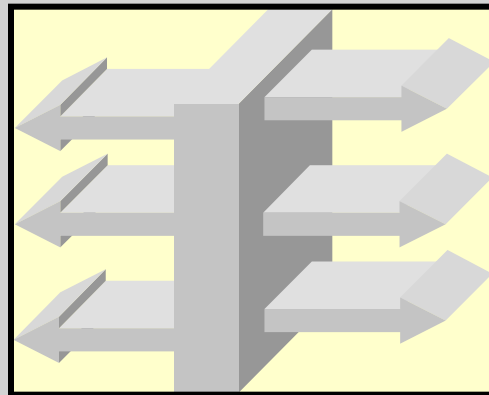
$$T_{\text{comp}} = T_{\text{comp}}^j = \frac{t_s(N^2 \cdot Z)}{P}$$

# Un ejemplo

---

## El estudio del rendimiento

- Tiempo de comunicación
  - Comunicaciones en un sólo plano; 2 por iteración.



- El tamaño del mensaje es el número de puntos que posee cada procesador en sus fronteras, es decir,  $N*Z$

$$T_{comu}^j = 2(\alpha + NZ\beta)$$

## El estudio del rendimiento

- Tiempo de comunicación
  - Como todos los procesadores realizan las mismas operaciones de entrada/salida y, además, de forma simultanea:

$$T_{\text{comu}} = 2(\alpha + NZ\beta)$$

- Tiempo de inactividad
  - Todas las iteraciones son iguales.
  - Cada procesador es responsable de la misma carga de computación y comunicaciones.
  - Ejecución sincronizada con sus vecinos.

# Un ejemplo

---

## El estudio del rendimiento

- En resumen

$$T = \left[ \frac{t_s(N^2Z)}{P} \right] + [2(\alpha + NZ\beta)] + 0$$

- En el límite

$$\lim_{N \rightarrow \infty} T \cong \frac{N^2Z}{P}$$

- Respecto al secuencial

$$\frac{T_{\text{Secuencial}}}{T_{\text{Paralelo}}} = \frac{N^2Z}{NZ \frac{N}{P}} = P$$



# Un ejemplo

---

## El estudio del rendimiento

- Speedup

$$S(N;p) = \frac{N^2 Z t_s}{\frac{t_s (N^2 Z)}{p} + 2(\alpha + NZ\beta)} = \frac{p N^2 Z t_s}{N^2 Z t_s + 2p(\alpha + NZ\beta)}$$

- Eficiencia

$$E(N;p) = \frac{S(N;p)}{p} = \frac{N^2 Z t_s}{N^2 Z t_s + 2p(\alpha + NZ\beta)}$$

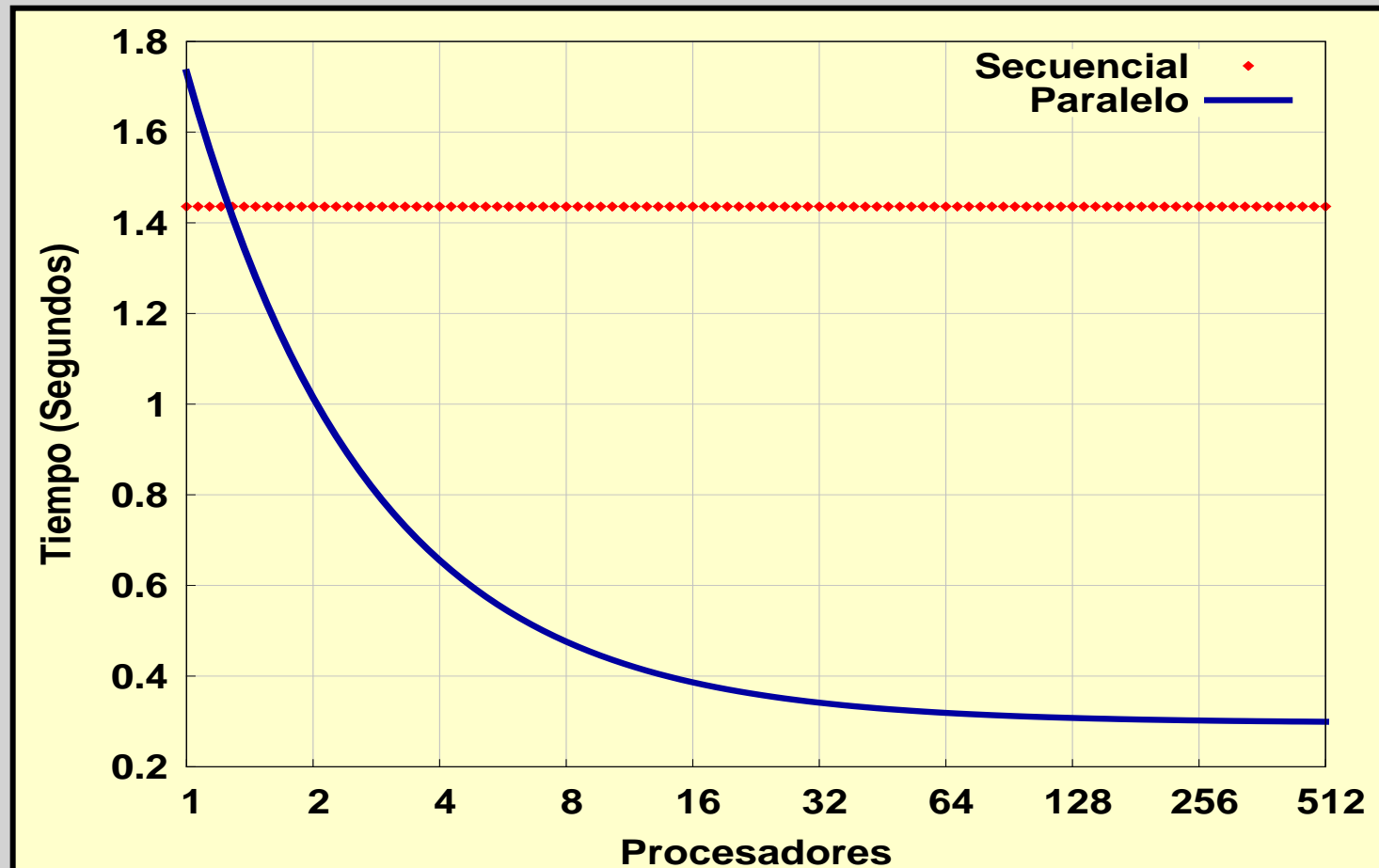
- Coste

$$C(N;p) = S(N;p)p = \frac{p^2 N^2 Z t_s}{N^2 Z t_s + 2p(\alpha + NZ\beta)}$$

# Un ejemplo

## El estudio del rendimiento

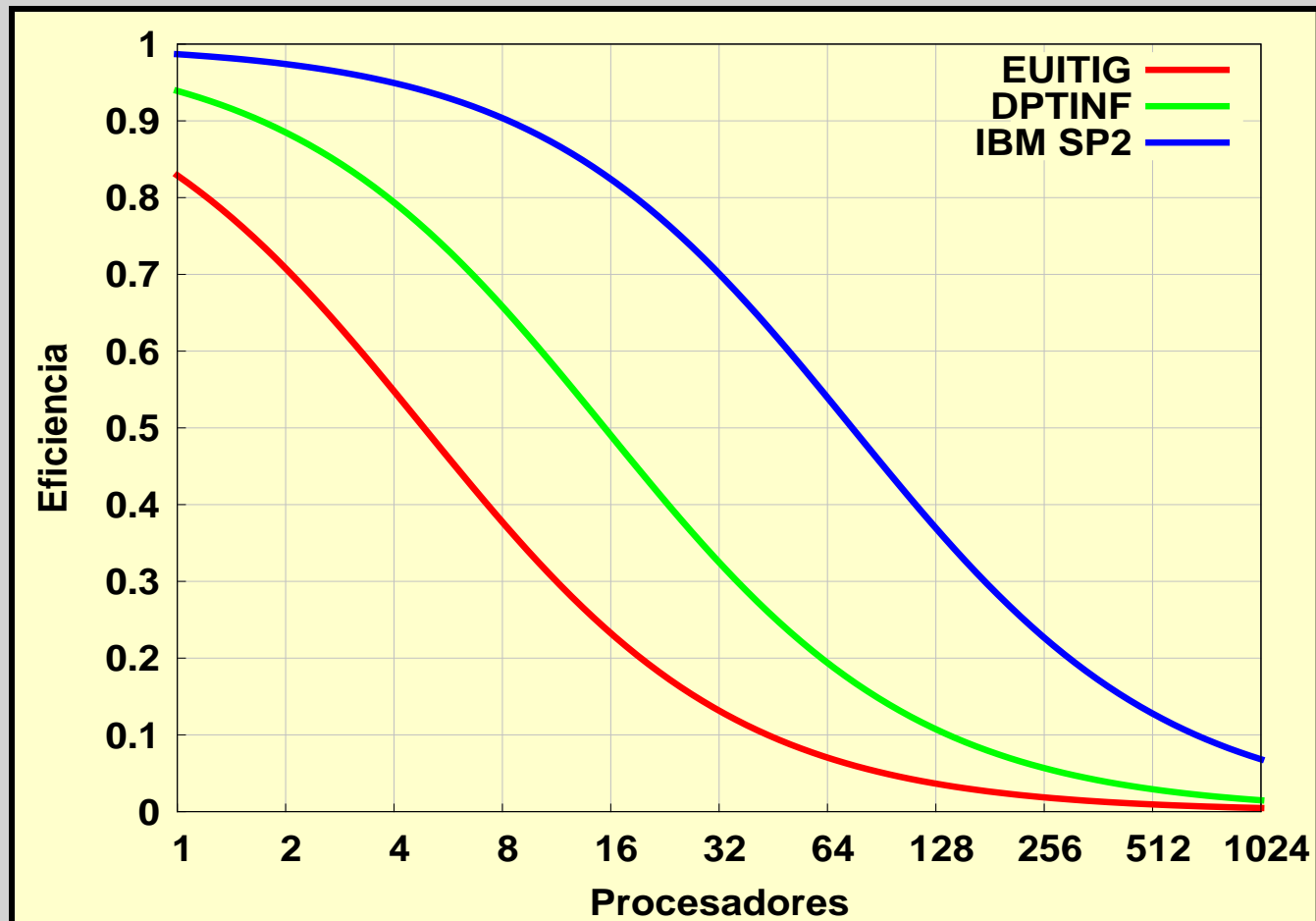
- $t_s=1\text{E-}8$ ,  $\alpha=6.3\text{E-}5$ ,  $\beta=1.1\text{E-}6$ ,  $Z=128$ ,  $N=1024$ ,  $k \in [1, 512]$



# Un ejemplo

## El estudio del rendimiento

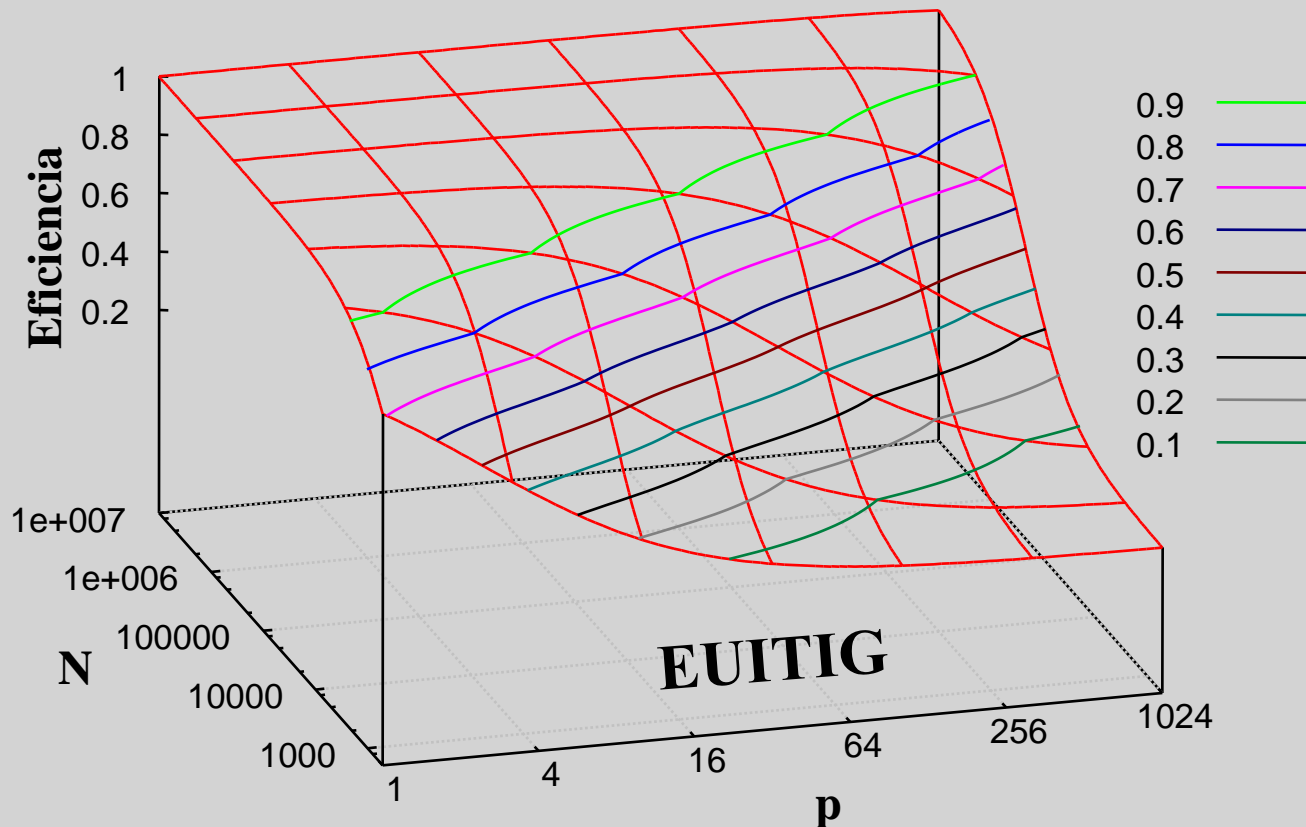
- $Z=128$ ,  $N=1024$ ,  $p \in [1, 1024]$



# Función de sobrecarga

## El estudio del rendimiento

- $N \in [512, 1E+07]$ ,  $Z=N$ ,  $p \in [1, 1024]$



# Función de sobrecarga

## El estudio del rendimiento

- $N \in [512, 1E+07]$ ,  $Z=N$ ,  $p \in [1, 1024]$

