

# **Tema 2**

## **Análisis del rendimiento**

**Arquitectura de los Computadores**

# Tema 2. Análisis del rendimiento

## 🏠 Objetivos

- 🏠 Entender el concepto de rendimiento, la evolución del rendimiento en los computadores en los últimos años y su relación con el coste
- 🏠 Saber cuantificar la ganancia de rendimiento o aceleración que puede obtenerse al mejorar alguna característica de un computador
- 🏠 Mostrar al alumno distintas métricas para evaluar el rendimiento de una arquitectura, observando la relación que existe entre ellas.
- 🏠 Adquirir conciencia de la necesidad de establecer métricas para llevar a cabo procesos de evaluación y comparación objetiva y contrastada de sistemas computacionales

# Tema 2. Análisis del rendimiento

## 🏠 Contenido

### 🏠 2.1. Rendimiento. Concepto y definiciones

- 🏠 Concepto de rendimiento
- 🏠 Ley de Amdhal
- 🏠 Relación entre rendimiento y coste

### 🏠 2.2. Evaluación del rendimiento

- 🏠 Medidas del rendimiento
- 🏠 Programas para evaluar el rendimiento
- 🏠 Formulación de resultados

# **2.1. Rendimiento. Concepto y definiciones**

**Tema 2 Análisis del rendimiento**

**Arquitectura de los Computadores**

## 2.1 Rendimiento. Concepto y definiciones

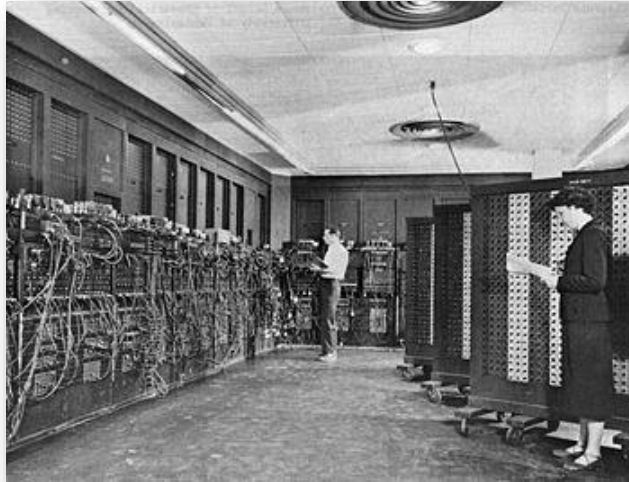
### Concepto

Amdahl

Relación

Análisis de  
rendimiento

### Evolución del rendimiento



- La tecnología de computadores ha tenido un increíble progreso en los últimos 65 años
- Por menos de 500€ es posible comprar un portátil que tiene más rendimiento, memoria y capacidad de disco que un computador que costaba casi 1 millón de € en 1985.

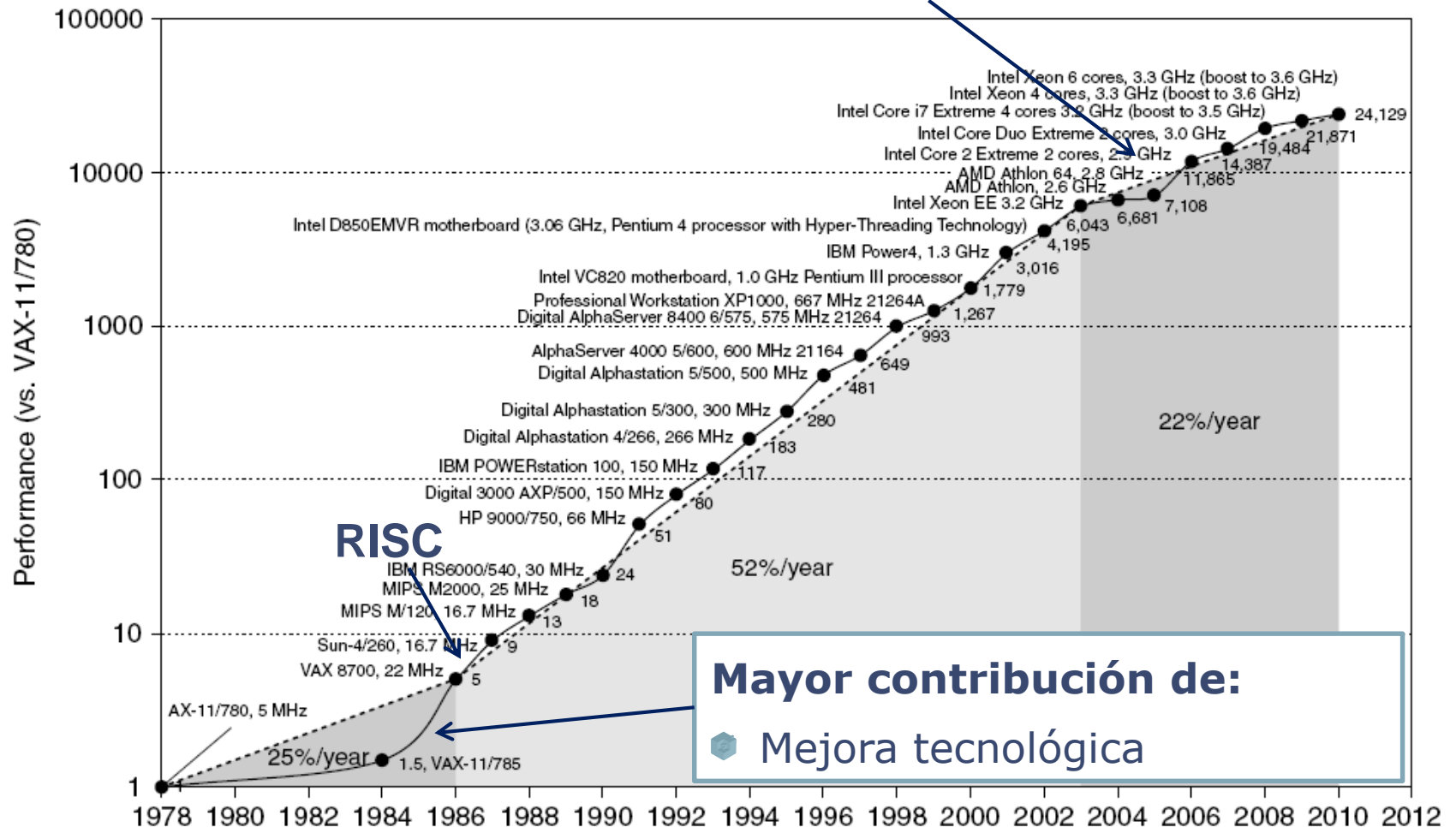
- Esta rápida mejora se debe fundamentalmente a:
  - Avances en la tecnología** (casi constante) usada para construir computadores
    - Tamaño de los elementos en el chip (feature size), velocidad del reloj
  - Innovaciones en el diseño** (menos consistentes)
    - Compiladores de lenguajes de alto nivel, UNIX
    - Provocado por arquitecturas RISC

# 2.1 Rendimiento. Concepto y definiciones

## Concepto

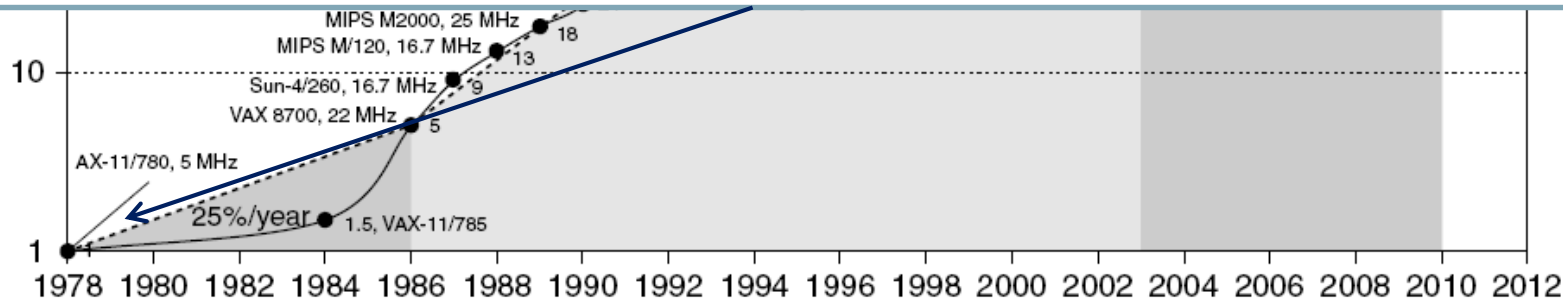
## Evolución del rendimiento

## Multiprocesadores



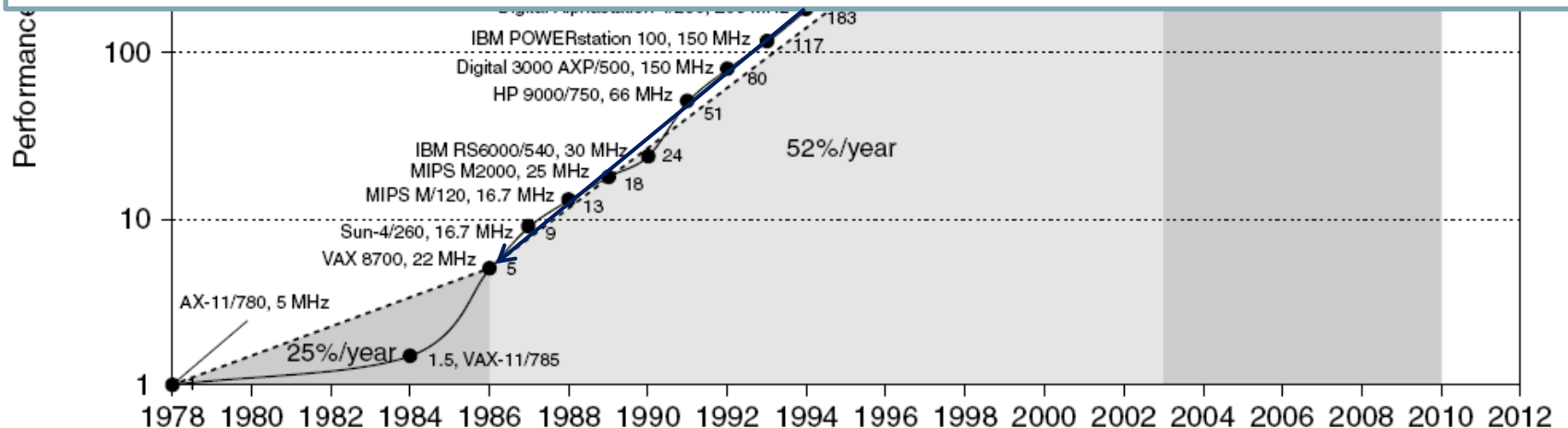
## Aparición del microprocesador ( $\mu$ P) (finales 1970)

- Capacidad de dirigir los avances en la tecnología de circuitos integrados
- Tasa más alta de mejora del rendimiento (35% anual)
- Ventajas en el coste debido a la producción masiva de  $\mu$ Ps.
- Aumenta el número de computadores basados en  $\mu$ Ps.
- Además, dos cambios significantes:
  - Eliminación virtual de la programación en lenguaje ensamblador
    - Reduce la necesidad de la compatibilidad en el código objeto
  - Aparición de sistemas operativos estandarizados como UNIX
    - Reducen el coste y el riesgo en la aparición de una nueva arquitectura



## RISC (Principios 1980)

- Cambios anteriores permiten el desarrollo de forma satisfactoria de un nuevo conjunto de arquitecturas con instrucciones más simples: arquitecturas RISC (Reduced Instruction Set Computer).
- Los diseñadores de máquinas RISC se centraron en dos técnicas clave para la mejora del rendimiento:
  - La explotación del paralelismo a nivel de instrucción
  - El uso de cachés
- El aumento del rendimiento forzó a las arquitecturas previas a mantener el ritmo o a desaparecer





# 2.1 Rendimiento. Concepto y definiciones

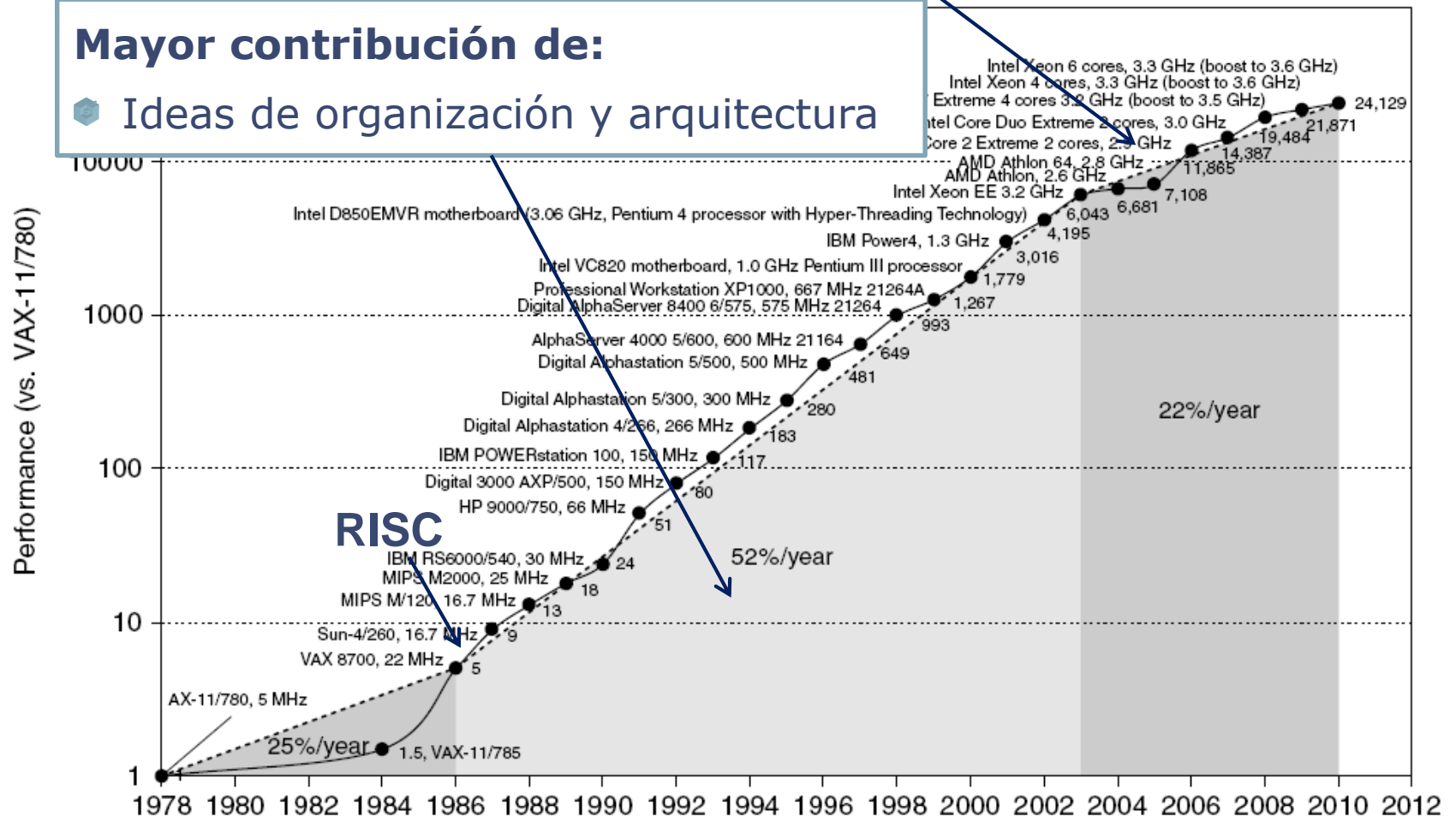
## Concepto

## Evolución del rendimiento

## Multiprocesadores

Mayor contribución de:

➡ Ideas de organización y arquitectura



## 2.1 Rendimiento. Concepto y definiciones

### Cae la mejora del rendimiento para un sólo procesador

- Máxima disipación de potencia de chips refrigerados por aire
- Falta de más paralelismo a nivel de instrucción a explotar eficientemente.

### Multiprocesadores



# 2.1 Rendimiento. Concepto y definiciones

## Concepto

Amdahl

Relación

Análisis de  
rendimiento

## Concepto de rendimiento

- Qué significa cuando decimos que un computador es más rápido que otro?:
  - **Usuario:**
    - Un computador es más rápido cuando un programa se ejecuta en **menos tiempo**
    - Interesado en reducir el **tiempo de respuesta/tiempo de ejecución**
      - Tiempo transcurrido entre el inicio y el final de un evento
  - **Administrador de un Cluster:**
    - Un computador es más rápido cuando completa **más transacciones por hora** (Google, Amazon...)
    - Interesado en aumentar la **productividad/throughput**
      - Cantidad total de trabajo realizado en un tiempo determinado

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

Análisis de  
rendimiento

### Ejemplo:

- ¿Las siguientes mejoras en el rendimiento afectan a la productividad, al tiempo de respuesta o a ambas cosas?
  - 1. Ciclo de reloj más rápido
  - 2. Múltiples procesadores para tareas separadas (sistema de reservas de una compañía aérea)
  - 3. Procesamiento paralelo de problemas científicos
  - **La disminución del tiempo de respuesta habitualmente mejora la productividad.**
    - 1 y 3 mejoran el tiempo de respuesta y en consecuencia la productividad. En el caso 2 no mejora el tiempo de respuesta pero si la productividad.
- La influencia de factores no determinísticos aconseja hablar de estas medidas de rendimiento con **distribuciones de probabilidad.**
  - Por ejemplo el tiempo de respuesta en un disco para una operación de entrada salida depende de:
    - Actividad del disco en el instante de petición.
    - Número de tareas intentando acceder al disco
- Esta situación aconseja hablar de tiempo medio de respuesta de un acceso al disco

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

### Concepto de rendimiento

- ⚙ El tiempo es la medida más fiable del rendimiento
- ⚙ El tiempo de ejecución de un programa se mide en segundos
- ⚙ La relación entre el tiempo y el rendimiento es inversa
- ⚙ El rendimiento se mide como una frecuencia de eventos por segundo

$$\text{Rendimiento} = \frac{1}{\text{tiempo}}$$

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

### Relación de rendimientos entre máquinas

"X es más rápida que Y"

$$t_{ex} < t_{ey}$$

"X es n % más rápida que Y"

$$t_{ex} \cdot n\% < t_{ey}$$

### Porcentaje incremental

$$tiempo\ ejecución_X + \frac{n}{100} tiempo\ ejecución_X = tiempo\ ejecución_Y$$

### Aceleración

$$\frac{tiempo\ ejecución_Y}{tiempo\ ejecución_X} = 1 + \frac{n}{100}$$

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

### Relación de rendimientos entre máquinas

#### En términos de rendimiento

$$1 + \frac{n}{100} = \frac{\text{tiempo ejecución}_Y}{\text{tiempo ejecución}_X} = \frac{\frac{1}{\text{Re } n \text{ dim iento}_Y}}{\frac{1}{\text{Re } n \text{ dim iento}_X}} = \frac{\text{Re } n \text{ dim iento}_X}{\text{Re } n \text{ dim iento}_Y}$$

#### Despejando

$$\frac{n}{100} = \frac{\text{Re } n \text{ dim iento}_X}{\text{Re } n \text{ dim iento}_Y} - 1 = \frac{\text{Re } n \text{ dim iento}_X - \text{Re } n \text{ dim iento}_Y}{\text{Re } n \text{ dim iento}_Y}$$

$$n = 100 \frac{\text{Re } n \text{ dim iento}_X - \text{Re } n \text{ dim iento}_Y}{\text{Re } n \text{ dim iento}_Y}$$

#### Expresado con tiempos de ejecución

$$n = 100 \frac{\text{tiempo ejecución}_Y - \text{tiempo ejecución}_X}{\text{tiempo ejecución}_X}$$

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

Análisis de  
rendimiento

### Relación de rendimientos entre máquinas

- Ejemplo: Si la máquina A ejecuta un programa en 10 segundos y la máquina B ejecuta un programa en 30 segundos. ¿Cuál de las siguientes afirmaciones es correcta?
  - A es el 30% más rápida que B
  - A es el 200% más rápida que B



## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

### Relación de rendimientos entre máquinas

❖ "La productividad de X es el 30 por 100 superior que la de Y"

$$P = nt/t; P_x = 1,3 * P_y$$

**Ejemplo:** Si la máquina A ejecuta un programa en 10 segundos y la máquina B ejecuta el mismo programa en 15 segundos. ¿En que porcentaje es la máquina A más rápida que la B?

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

### Relación de rendimientos entre máquinas

❖ "La productividad de X es el 30 por 100 superior que la de Y"

$$P = nt/t; P_x = 1,3 * P_y$$

**Ejemplo:** Si la máquina A ejecuta un programa en 10 segundos y la máquina B ejecuta el mismo programa en 15 segundos. ¿En que porcentaje es la máquina A más rápida que la B?

$$n = 100 \frac{\text{tiempo ejecución}_B - \text{tiempo ejecución}_A}{\text{tiempo ejecución}_A}$$

$$n = 100 \frac{15 - 10}{10} = 50$$

**"A es el 50% más rápida que B"**

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

Análisis de  
rendimiento

- ❖ **Ejercicio 1:** Se dan los tiempos de ejecución en segundos del benchmark linpack y de 10.000 iteraciones del benchmark Dhrystone en dos modelos VAX y un procesador actual :

Modelo	Año	Tiempo linpack	Tiempo Dhrystone
VAX- 11/780	1978	4,9	5,69
VAX 8600	1985	1,43	1,35
Intel Xeon 4 cores	2010	$2,24 \times 10^{-4}$	$2,74 \times 10^{-4}$

- ❖ **A) ¿Cuántas veces es más rápido el 8600 que el 780 utilizando Linpack? ¿Que ocurre cuando se utiliza Dhrystone?**
- ❖ B) ¿Cuántas veces es más rápido el Xeon que el 780 utilizando Linpack? ¿Que ocurre cuando se utiliza Dhrystone?
- ❖ **C) ¿Cual es el crecimiento medio del rendimiento por año entre el 780 y el 8600 utilizando linpack? ¿Que ocurre cuando se utiliza Dhrystone?**
- ❖ D) ¿Cual es el crecimiento medio del rendimiento por año entre el 780 y el Xeon utilizando linpack? ¿Que ocurre cuando se utiliza Dhrystone?

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

### • a y c) Comparación de rendimientos para el VAX 780 y el VAX 8600 (Según Linpack)

$$te_{VAX780} = 4,9 \text{ s}$$
$$rend_{VAX780} = \frac{1}{te_{VAX780}} = 0,204$$

$$te_{VAX8600} = 1,43 \text{ s}$$
$$rend_{VAX8600} = \frac{1}{te_{VAX8600}} = 0,699$$

### • Porcentaje de incremento del rendimiento entre las dos arquitecturas:

$$porcentaje = n = 100 \frac{te_{VAX780} - te_{VAX8600}}{te_{VAX8600}} = 100 \frac{4,9 - 1,43}{1,43} = 243\%$$

$$porcentaje = n = 100 \frac{rend_{VAX8600} - rend_{VAX780}}{rend_{VAX780}} = 100 \frac{0,699 - 0,204}{0,204} = 243\%$$

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

### ⬢ Aceleración del rendimiento entre las dos arquitecturas

$$aceleracion = \left(1 + \frac{n}{100}\right) = \frac{te_{VAX780}}{te_{VAX8600}} = \frac{4,9}{1,43} = 3,43 = \frac{rend_{VAX8600}}{rend_{VAX780}} = \frac{0,699}{0,204}$$

$$rend_{VAX8600} = 3,43 \cdot rend_{VAX780} \Rightarrow 0,699 = 3,43 \cdot 0,204$$

└──────────┘  
7 años

### ⬢ Los incrementos anuales se aplican cada año sobre el anterior

$$rend_{an} = \Delta_{anual} \cdot rend_{an-1} \rightarrow rend_{a1} = \Delta_{anual} \cdot rend_{a0}$$

$$rend_{a2} = \Delta_{anual} \cdot rend_{a1} = (\Delta_{anual})^2 rend_{a0}$$

$$rend_{an} = \Delta_{anual} \cdot rend_{an-1} = (\Delta_{anual})^n rend_{a0}$$

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

### El incremento anual es:

$$\Delta_{anual} = \sqrt[n]{\frac{rend_{an}}{rend_{a0}}} = \sqrt[n]{\frac{te_{a0}}{te_{an}}}$$

### Si consideramos $n=7$

$$te_{a0} = te_{VAX780} \qquad rend_{a0} = rend_{VAX780}$$

$$te_{a7} = te_{VAX8600} \qquad rend_{a7} = rend_{VAX8600}$$

$$\Delta_{anual} = \sqrt[n]{\frac{rend_{a7}}{rend_{a0}}} = \sqrt[n]{\frac{te_{a0}}{te_{a7}}} = \sqrt[7]{\frac{rend_{VAX8600}}{rend_{VAX780}}} = \sqrt[7]{\frac{te_{VAX780}}{te_{VAX8600}}} = \sqrt[7]{3,43} = 1,193$$

$$aceleración = \left(1 + \frac{n}{100}\right) = \Delta_{anual} \Leftrightarrow n = (\Delta_{anual} - 1) \cdot 100 = 19,3\%$$

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

Análisis de  
rendimiento

- **Ejercicio 2:** Hay dos equipos de diseño en dos compañías diferentes. La gestión de la compañía más pequeña y más agresiva pide un ciclo de diseño de dos años para sus productos. La gestión de la compañía más grande y menos agresiva apuesta por un ciclo de diseño de cuatro años. Supongamos que en el mercado de hoy día se demanda 1.5 veces el rendimiento de un Intel Xeon 4 cores según Linpack.
- ¿Cuales deberían ser los objetivos de rendimiento para cada producto, si las frecuencias de crecimiento necesarias son el 22% por año?
- Supongamos que las compañías acaban de empezar a utilizar chips de DRAM de 2048 Megabits. Dado que la capacidad por chip se ha incrementado entre un 25% y un 40% por año recientemente y casi doblándose de 2 a 3 años. Supongamos que las frecuencias de crecimiento serán 30% y se duplicará en 3 años ¿Que tamaños de DRAM hay que planificar para utilizar en estos proyectos?. Obsérvese que el crecimiento de las DRAM es discreto

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

- a) Actualmente 1.5 veces mas rendimiento que el Intel Xeon (según Linpack)

$$1.5 = \frac{te_{Xeon}}{te_{actual}} \Rightarrow te_{actual} = \frac{te_{Xeon}}{1.5} = \frac{2.24 \cdot 10^{-4}}{1.5} = 1.49 \cdot 10^{-4} s$$

- El rendimiento crece un 22% anual

$$rend_{a1} = rend_{act} + 0.22 \cdot rend_{act} = 1.22 \cdot rend_{act} \Rightarrow 1.22 = \frac{rend_{a1}}{rend_{act}} = \frac{te_{act}}{te_{a1}}$$

- El primer año  $te_{a1} = \frac{1}{1.22} \cdot te_{act} = 0.82 \cdot te_{act}$

- El segundo año  $te_{a2} = \frac{1}{1.22} \cdot te_{a1} = 0.82 \cdot te_{a1} = (0.82)^2 \cdot te_{act}$

- El año **n**  $te_{an} = \frac{1}{1.22} \cdot te_{an-1} = 0.82 \cdot te_{an-1} = (0.82)^n \cdot te_{act}$



## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

- a) Actualmente 1.5 veces mas rendimiento que el Intel Xeon (según Linpack)

$$1.5 = \frac{te_{Xeon}}{te_{actual}} \Rightarrow te_{actual} = \frac{te_{Xeon}}{1.5} = \frac{2.24 \cdot 10^{-4}}{1.5} = 1.49 \cdot 10^{-4} s$$

- El rendimiento crece un 22% anual

$$rend_{a1} = rend_{act} + 0.22 \cdot rend_{act} = 1.22 \cdot rend_{act} \Rightarrow 1.22 = \frac{rend_{a1}}{rend_{act}} = \frac{te_{act}}{te_{a1}}$$

- Empresa A (2 años)

$$te_{a2} = (0.82)^2 \cdot te_{act} = (0.82)^2 \cdot 1.49 \cdot 10^{-4} = 1 \cdot 10^{-4} s = 100 \mu s$$

- Empresa B (4 años )

$$te_{a4} = (0.82)^4 \cdot te_{act} = (0.82)^4 \cdot 1.49 \cdot 10^{-4} = 6.74 \cdot 10^{-5} s = 67.4 \mu s$$

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

- b) El tamaño de la memoria actual es 2048 Mb

- Tamaño de la memoria año 1

$$tm_{a1} = tm_{act} + 0.3 \cdot tm_{act} \Rightarrow tm_{a1} = 1.3 \cdot tm_{act}$$

- Tamaño de la memoria año 2

$$tm_{a2} = tm_{a1} + 0.3 \cdot tm_{a1} \Rightarrow tm_{a2} = 1.3 \cdot tm_{a1} = (1.3)^2 \cdot tm_{act}$$

- Tamaño de la memoria año n

$$tm_{an} = tm_{an-1} + 0.3 \cdot tm_{an-1} \Rightarrow tm_{an} = 1.3 \cdot tm_{an-1} = (1.3)^n \cdot tm_{act}$$

- Evolución continua

$$tm_{a2} = (1.3)^2 \cdot tm_{act} = (1.3)^2 \cdot 2048 = 3461,12Mb$$

$$tm_{a3} = (1.3)^3 \cdot tm_{act} = (1.3)^3 \cdot 2048 = 4449,45Mb$$

$$tm_{a4} = (1.3)^4 \cdot tm_{act} = (1.3)^4 \cdot 2048 = 5849,29Mb$$

$$tm_{a6} = (1.3)^6 \cdot tm_{act} = (1.3)^6 \cdot 2048 = 9885,30Mb$$

## 2.1 Rendimiento. Concepto y definiciones

### Concepto

Amdahl

Relación

Análisis de  
rendimiento

- b) El tamaño de la memoria actual es 2048 Mb

- Tamaño de la memoria año 1

$$tm_{a1} = tm_{act} + 0.3 \cdot tm_{act} \Rightarrow tm_{a1} = 1.3 \cdot tm_{act}$$

- Tamaño de la memoria año 2

$$tm_{a2} = tm_{a1} + 0.3 \cdot tm_{a1} \Rightarrow tm_{a2} = 1.3 \cdot tm_{a1} = (1.3)^2 \cdot tm_{act}$$

- Tamaño de la memoria año n

$$tm_{an} = tm_{an-1} + 0.3 \cdot tm_{an-1} \Rightarrow tm_{an} = 1.3 \cdot tm_{an-1} = (1.3)^n \cdot tm_{act}$$

- Evolución discreta (x2 cada 3 años)

- Empresa A (salto superior más cercano 3 años x 2)

$$tm_{EmpA} = 2048Mb_{act} \cdot 2 = 4096Mb$$

- Empresa B (salto superior más cercano 6 años x 2)

$$tm_{EmpB} = 2048Mb_{act} \cdot 2 \cdot 2 = 8192Mb$$

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

### Ley de Amdhal

- Relacionado con el principio de diseño de **favorecer el caso frecuente**
- Define la ganancia** de rendimiento o aceleración que puede obtenerse al mejorar alguna característica de un computador
- La mejora** obtenida en el rendimiento al utilizar algún modo de ejecución más rápido **está limitada por la fracción de tiempo** en **que se puede utilizar** ese modo más rápido

$$\text{Aceleración Rendimiento} = \frac{\text{Rendimiento con mejora}}{\text{Rendimiento sin mejora}} = \frac{\text{Tiempo ejecución sin mejora}}{\text{Tiempo ejecución con mejora}}$$

Análisis de  
rendimiento

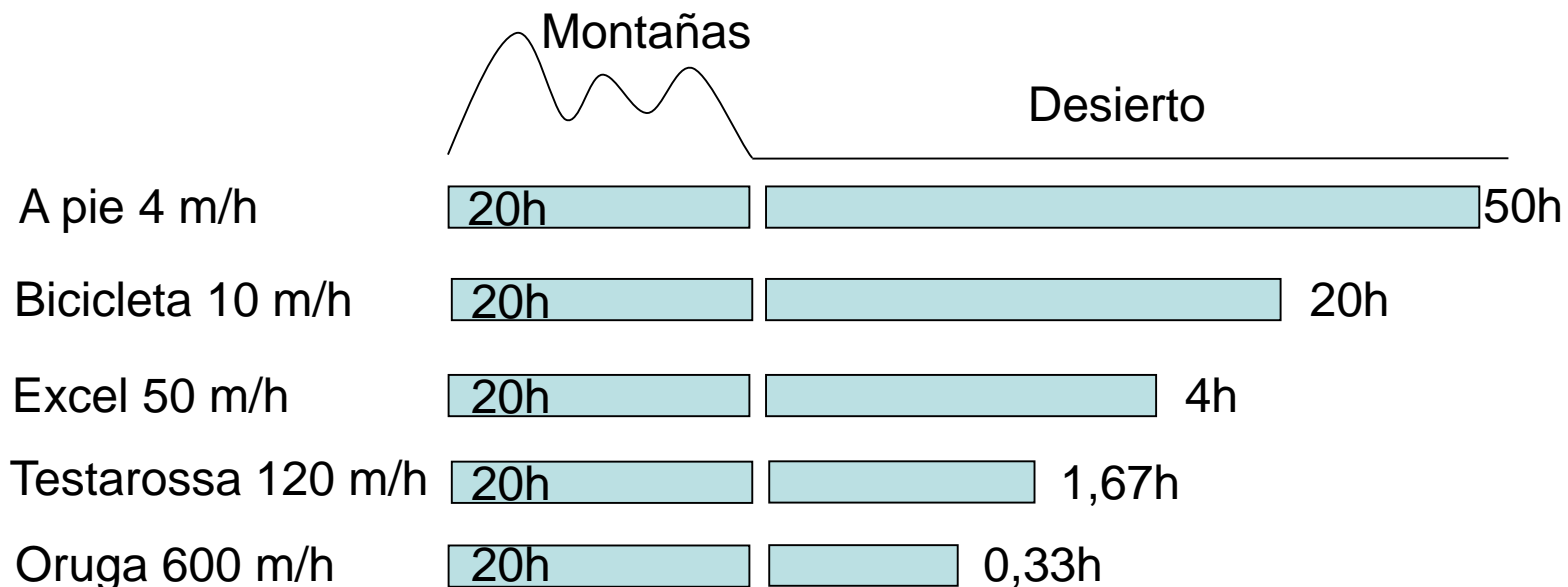
## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

- ◆ **Ejemplo:** Se pretende viajar entre dos puntos cuyo trayecto involucra a dos tipos de terreno. El primer tramo es a través de las montañas y el segundo por el desierto. Tenemos varios tipos de vehículos pero las montañas tienen que ser recorridas necesariamente a pie, empleando para ello 20 horas. El segundo tramo de 200 millas puede ser recorrido de cinco formas diferentes:



Análisis de  
rendimiento

## 2.1 Rendimiento. Concepto y definiciones

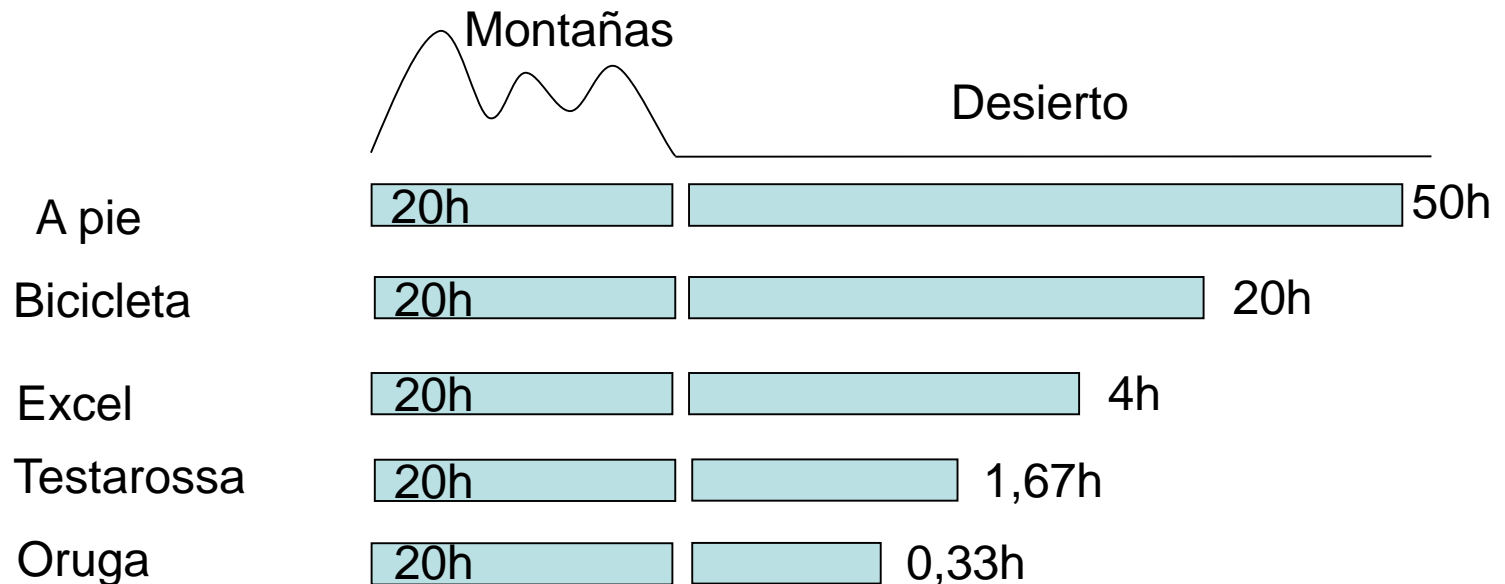
Concepto

Amdahl

Relación

Tomando como referencia el recorrido a pie de la distancia completa

Vehículo segunda parte del viaje	Horas de la segunda parte del viaje	Aceleración en el desierto	Horas del viaje completo	Aceleración en el viaje completo
A pie	50	$1=4/4=50/50$	$70=50+20$	$1,0=70/70$
Bicicleta	20	$2,5=10/4=50/20$	$40=20+20$	$1,8=70/40$
Excel	4	$12,5=50/4=50/4$	$24=20+4$	$2,9=70/24$
Testarossa	1,67	$30=120/4=50/1,67$	$21,67=20+1,67$	$3,2=70/21,67$
Vehículo oruga	0,33	$150=600/4=50/0,33$	$20,33=20+0,33$	$3,4=70/20,33$



Análisis de  
rendimiento

## 2.1 Rendimiento. Concepto y definiciones

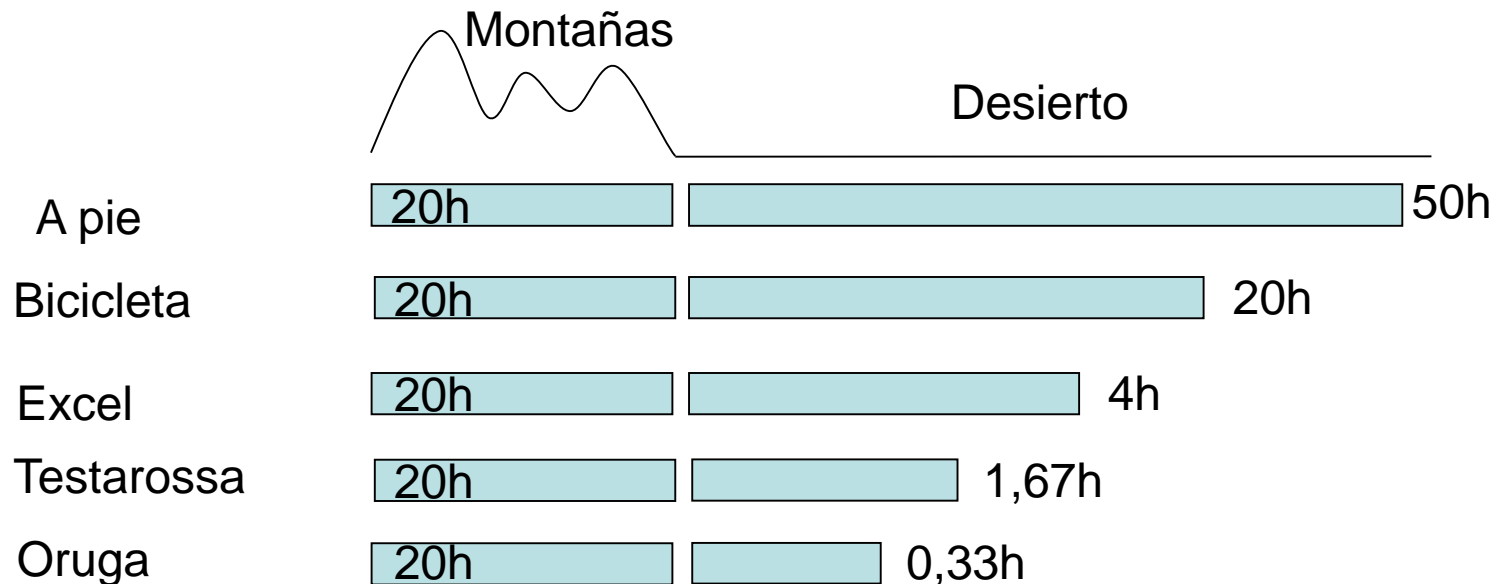
Concepto

Amdahl

Relación

Análisis de  
rendimiento

- ❖ **Ley de Amdahl:** aceleración dependiente de dos factores:
- ❖ **Fracción mejorada:** fracción de tiempo de la opción sin la mejora que puede utilizarse para aprovechar la mejora (siempre menor o igual que uno). En el ejemplo anterior 50/70.
- ❖ **Aceleración mejorada:** Aceleración del modo mejorado (mayor que uno). En el ejemplo anterior aceleración en el desierto.



## 2.1 Rendimiento. Concepto y definiciones

Concepto

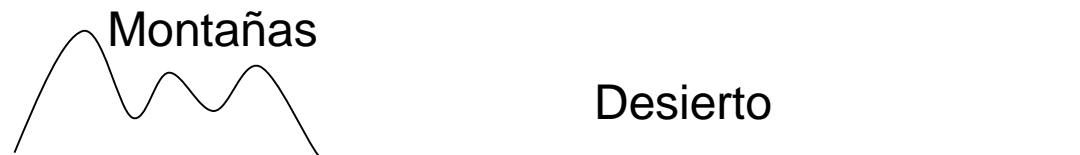
Amdahl

Relación

- El **tiempo de ejecución nuevo** (máquina original con el modo mejorado) (21,67) es el tiempo empleado sin utilizar la parte mejorada (20) mas el tiempo empleado utilizando la parte mejorada (1,67) (tiempo a pie + tiempo en Testarossa)

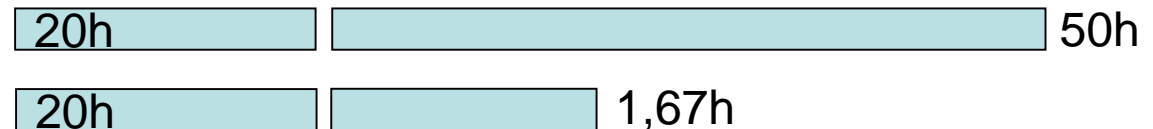
$$TE_{nuevo} = TE_{antiguo} \left( \underbrace{(1 - Fracción_{mejorada})}_{\text{TE antiguo - TE antiguo utilizado en modo mejorado}} + \underbrace{\frac{Fracción_{mejorada}}{Aceleración_{mejorada}}}_{\text{TE antiguo utilizado en modo mejorado / Aceleración mejorada}} \right)$$

70h	TE antiguo	TE antiguo utilizado en modo mejorado	50h
50h	-	/	
	TE antiguo utilizado en modo mejorado	Aceleración mejorada	30
	=	=	
20h	TE antiguo utilizado en modo no mejorado	TE nuevo utilizando modo mejorado	1,67h



A pie

Testarossa



Análisis de  
rendimiento



## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

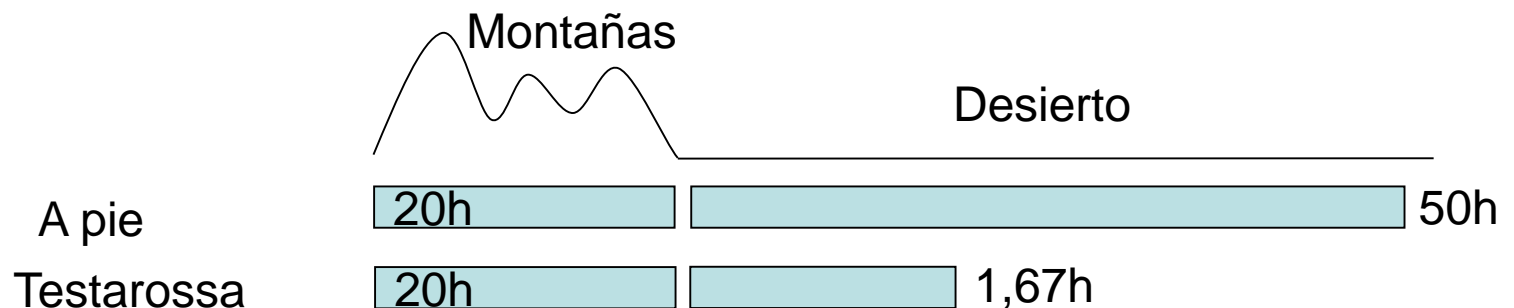
Relación

- El **tiempo de ejecución nuevo** (máquina original con el modo mejorado) (21,67) es el tiempo empleado sin utilizar la parte mejorada (20) mas el tiempo empleado utilizando la parte mejorada (1,67) (tiempo a pie + tiempo en Testarossa)

$$Aceleración_{global} = \frac{TE_{antiguo}}{TE_{nuevo}} = \frac{1}{(1 - Fracción_{mejorada}) + \frac{Fracción_{mejorada}}{Aceleración_{mejorada}}}$$

$$fracción_{mejorada}=1 \rightarrow aceleración_{global}=aceleración_{mejorada}$$

$$fracción_{mejorada}=0 \rightarrow aceleración_{global}=1$$



Análisis de  
rendimiento

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

Análisis de  
rendimiento

- **Ejemplo:** Suponer que estamos considerando una mejora que corra diez veces más rápida que la máquina original, pero sólo es utilizable el 40% del tiempo. ¿Cual es la aceleración global lograda al incorporar la mejora?

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

- **Ejemplo:** Suponer que estamos considerando una mejora que corra diez veces más rápida que la máquina original, pero sólo es utilizable el 40% del tiempo. ¿Cual es la aceleración global lograda al incorporar la mejora?

$$\text{Fracción}_{\text{mejorada}} = 0,4$$

$$\text{Aceleración}_{\text{mejorada}} = 10$$

$$\text{Aceleración}_{\text{global}} = \frac{1}{0,6 + \frac{0,4}{10}} = \frac{1}{0,64} \approx 1,56$$

- **Ejemplo:** Suponer que una cache es 5 veces más rápida que la memoria principal y supongamos que la cache puede ser utilizada el 90% del tiempo ¿Qué aumento de velocidad se logrará al utilizar la cache?

Análisis de  
rendimiento

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

Análisis de  
rendimiento

- **Ejemplo:** Suponer que estamos considerando una mejora que corra diez veces más rápida que la máquina original, pero sólo es utilizable el 40% del tiempo. ¿Cual es la aceleración global lograda al incorporar la mejora?

$$\text{Fracción}_{\text{mejorada}} = 0,4$$

$$\text{Aceleración}_{\text{mejorada}} = 10$$

$$\text{Aceleración}_{\text{global}} = \frac{1}{0,6 + \frac{0,4}{10}} = \frac{1}{0,64} \approx 1,56$$

- **Ejemplo:** Suponer que una cache es 5 veces más rápida que la memoria principal y supongamos que la cache puede ser utilizada el 90% del tiempo ¿Qué aumento de velocidad se logrará al utilizar la cache?

$$\text{Fracción}_{\text{mejorada}} = 0,9$$

$$\text{Aceleración}_{\text{mejorada}} = 5$$

$$\text{Aceleración}_{\text{global}} = \frac{1}{0,1 + \frac{0,9}{5}} = \frac{1}{0,25} = 3,6$$

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

Análisis de  
rendimiento

- **Ejercicio 3:** Para las cuatro siguientes preguntas, supongamos que se está considerando mejorar una máquina añadiéndole un modo vectorial. Cuando se ejecuta un cálculo en modo vectorial, es 20 veces más rápido que en el modo normal de ejecución. Llamamos al porcentaje de tiempo que puede emplearse el modo vectorial porcentaje de vectorización.
- 1. Dibujar un gráfico donde se muestre la aceleración como porcentaje del cálculo realizado en modo vectorial. Rotular el eje y con "aceleración neta" y el eje x con "porcentaje de vectorización".
- 2. ¿Que porcentaje de vectorización se necesita para conseguir una aceleración de 2?
- 3. ¿Que porcentaje de vectorización se necesita para conseguir la mitad de la aceleración máxima alcanzable utilizando el modo vectorial?
- 4. Supongamos que hemos medido el porcentaje de vectorización de programas, obteniendo que es del 70%. El grupo de diseño hardware dice que puede duplicar la velocidad de la parte vectorizada con una inversión significativa de ingeniería adicional. Se desea saber si el equipo de compilación puede incrementar la utilización del modo vectorial como otra aproximación para incrementar el rendimiento. ¿Que incremento en el porcentaje de vectorización (relativo a la utilización actual) se necesitará para obtener la misma ganancia de rendimiento? ¿Que inversión es recomendable?

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

Análisis de  
rendimiento

### ● Ejercicio 3:

- 1. Dibujar un gráfico donde se muestre la aceleración como porcentaje del cálculo realizado en modo vectorial. Rotular el eje y con "aceleración neta" y el eje x con "porcentaje de vectorización".

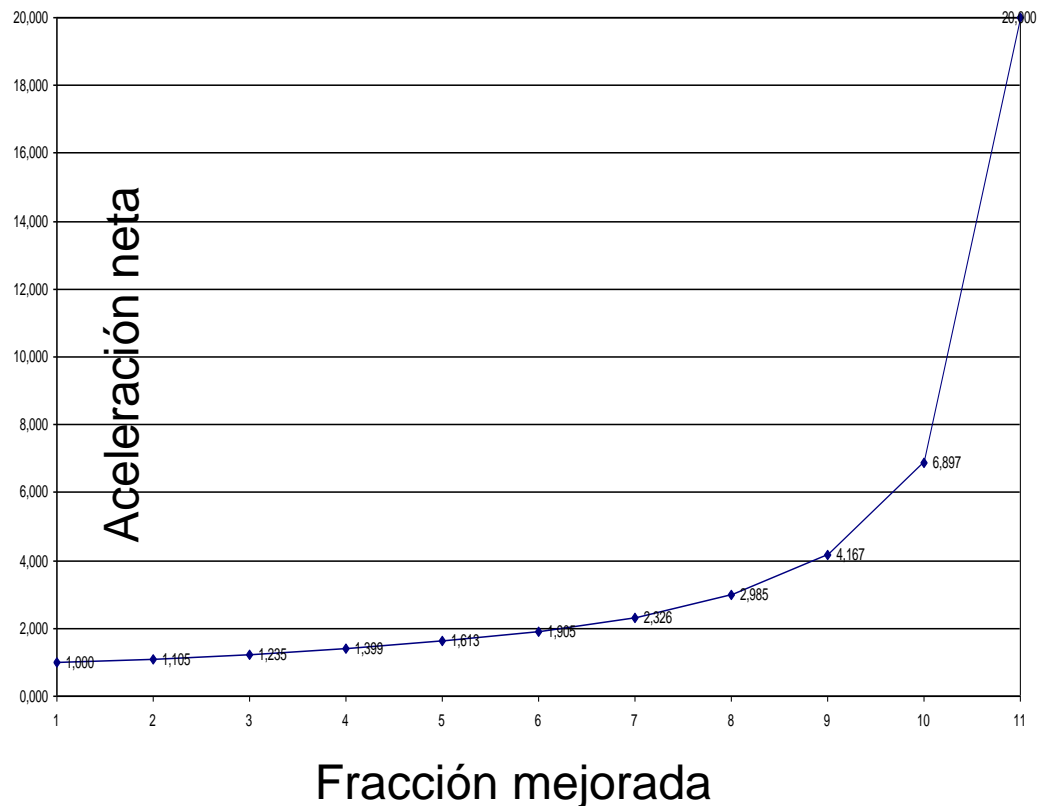
$$0\% \quad a_g = \frac{1}{1 + \frac{0}{20}} = 1$$

$$10\% \quad a_g = \frac{1}{0,9 + \frac{0,1}{20}} = 1,105$$

$$20\% \quad a_g = \frac{1}{0,8 + \frac{0,2}{20}} = 1,235$$

$$90\% \quad a_g = \frac{1}{0,1 + \frac{0,9}{20}} = 6,9$$

$$100\% \quad a_g = \frac{1}{0 + \frac{1}{20}} = 20$$



## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

Análisis de  
rendimiento

### • Ejercicio 3:

- 2.¿Que porcentaje de vectorización se necesita para conseguir una aceleración de 2?

$$2 = \frac{1}{(1 - f_m) + \frac{f_m}{20}} \Rightarrow f_m = 0,526 = 52,6\%$$

- 3.¿Que porcentaje de vectorización se necesita para conseguir la mitad de la aceleración máxima alcanzable utilizando el modo vectorial?.

$$10 = \frac{1}{(1 - f_m) + \frac{f_m}{20}} \Rightarrow f_m = 0,947 \approx 95\%$$

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

- 4. Hemos medido el porcentaje de vectorización de programas, obteniendo que es del 70%. El grupo de diseño hardware dice que puede duplicar la velocidad de la parte vectorizada con una inversión significativa de ingeniería adicional. Se desea saber si el equipo de compilación puede incrementar la utilización del modo vectorial como otra aproximación para incrementar el rendimiento. ¿Que incremento en el porcentaje de vectorización (relativo a la utilización actual) se necesitará para obtener la misma ganancia de rendimiento? ¿Que inversión es recomendable?.

$$TE_{\text{ant}} = TE_{\text{vec}} * 20 = TE_{\text{vec.r}} * 40 \rightarrow TE_{\text{vec}} = TE_{\text{vec.r}} * 2$$

- Aceleración global conseguida por el grupo de hardware

$$a_g = \frac{1}{(1 - 0,7) + \frac{0,7}{40}} = 3,15$$

Análisis de  
rendimiento



## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

- 4. Hemos medido el porcentaje de vectorización de programas, obteniendo que es del 70%. El grupo de diseño hardware dice que puede duplicar la velocidad de la parte vectorizada con una inversión significativa de ingeniería adicional. Se desea saber si el equipo de compilación puede incrementar la utilización del modo vectorial como otra aproximación para incrementar el rendimiento. ¿Que incremento en el porcentaje de vectorización (relativo a la utilización actual) se necesitará para obtener la misma ganancia de rendimiento? ¿Que inversión es recomendable?

$$TE_{ant} = TE_{vec} * 20 = TE_{vec.r} * 40 \rightarrow TE_{vec} = TE_{vec.r} * 2$$

- Incremento del % de vectorización para alcanzar  $a_g = 3,15$

$$a_g = 3,15 = \frac{1}{(1 - f_m) + \frac{f_m}{20}} \Rightarrow f_m = 0,719$$

$$\Delta\%vector = |0,7 - 0,719| = 0,019 = 1,9\%$$

Análisis de  
rendimiento

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

Análisis de  
rendimiento

- **Ejercicio 4:** El coprocesador de un computador mejora en un factor de 5 el procesamiento de números en coma flotante. El tiempo de ejecución de cierto programa es de 1 minuto con el coprocesador instalado, y de 2,5 minutos sin este.
- a) Calcula el porcentaje del tiempo de ejecución, sin el coprocesador instalado, que el programa realiza operaciones en coma flotante.
- b) Calcula el tiempo de ejecución del programa sin el coprocesador instalado, para realiza las operaciones en coma flotante.
- c) Calcula el tiempo de ejecución del programa con el coprocesador instalado, para realiza las operaciones en coma flotante.
- d) Calcula el tiempo de ejecución del programa para realiza las operaciones enteras.
- e) Comprueba la coherencia del planteamiento sumando los resultados de los apartados c y d, obteniendo el tiempo de ejecución del programa con el coprocesador instalado.





## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

- **Ejercicio 4:** El coprocesador de un computador mejora en un factor de 5 el procesamiento de números en coma flotante. El tiempo de ejecución de cierto programa es de 1 minuto con el coprocesador instalado, y de 2,5 minutos sin este.
- a) Calcula el porcentaje del tiempo de ejecución, sin el coprocesador instalado, que el programa realiza operaciones en coma flotante.

	Tarea sin punto flotante	Tarea con punto flotante
Sin copro		 2,5m
Con copro		 1m

$$a_g = \frac{t_{e.ant}}{t_{e.nue}} = \frac{2,5}{1} = \frac{1}{(1 - f_m) + \frac{f_m}{5}} \Rightarrow f_m = 0,75$$

Análisis de  
rendimiento

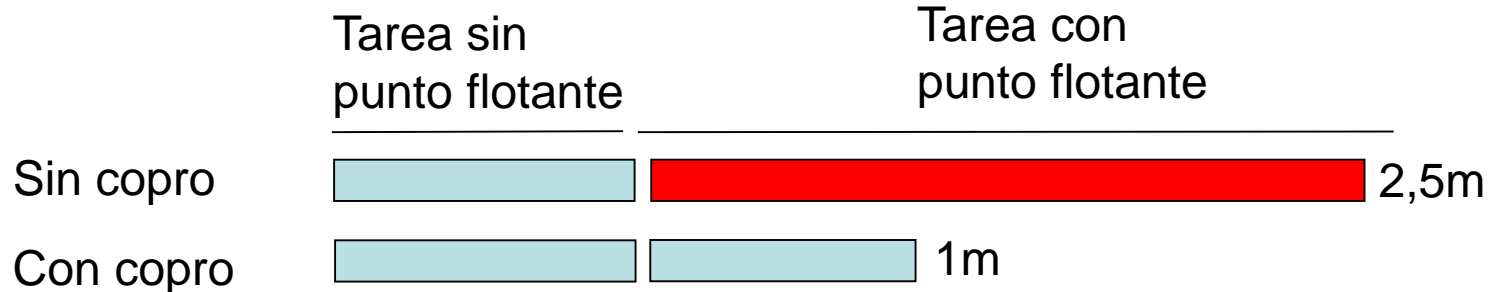
## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

- **Ejercicio 4:** El coprocesador de un computador mejora en un factor de 5 el procesamiento de números en coma flotante. El tiempo de ejecución de cierto programa es de 1 minuto con el coprocesador instalado, y de 2,5 minutos sin este.
- b) Calcula el tiempo de ejecución del programa sin el coprocesador instalado, para realiza las operaciones en coma flotante.



$$t_{e.ant} = 2,5$$

$$t_{e.ant.flot} = 2,5 \square 0,75 = 1,875m$$

Análisis de  
rendimiento





## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

- **Ejercicio 4:** El coprocesador de un computador mejora en un factor de 5 el procesamiento de números en coma flotante. El tiempo de ejecución de cierto programa es de 1 minuto con el coprocesador instalado, y de 2,5 minutos sin este.
- c) Calcula el tiempo de ejecución del programa con el coprocesador instalado, para realiza las operaciones en coma flotante.

	Tarea sin punto flotante	Tarea con punto flotante
Sin copro		 2,5m
Con copro		 1m

$$t_{e.ant} = 2,5$$

$$t_{e.nue.flot} = \frac{1,875}{5} = 0,375m$$

Análisis de  
rendimiento

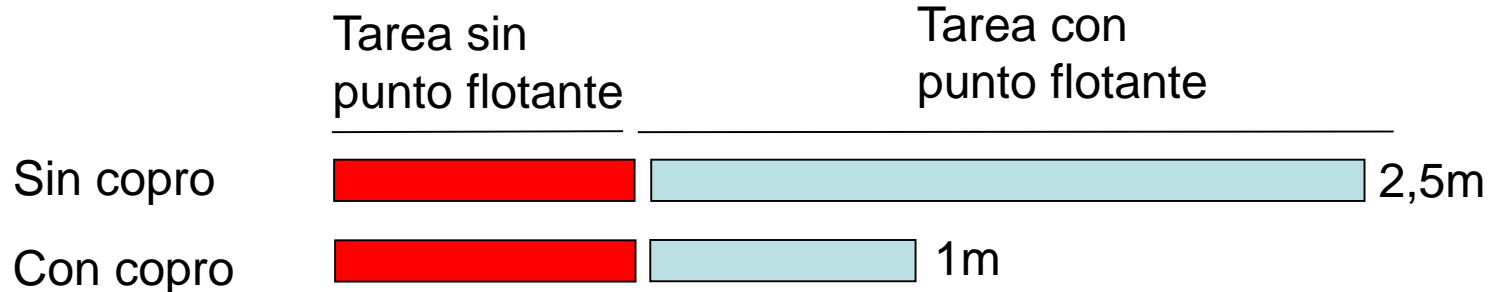
## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

- **Ejercicio 4:** El coprocesador de un computador mejora en un factor de 5 el procesamiento de números en coma flotante. El tiempo de ejecución de cierto programa es de 1 minuto con el coprocesador instalado, y de 2,5 minutos sin este.
- d) Calcula el tiempo de ejecución del programa para realiza las operaciones enteras.



$$t_{e.int} = 2,5 - 1,875 = 0,625m$$

Análisis de  
rendimiento





## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

- **Ejercicio 4:** El coprocesador de un computador mejora en un factor de 5 el procesamiento de números en coma flotante. El tiempo de ejecución de cierto programa es de 1 minuto con el coprocesador instalado, y de 2,5 minutos sin este.
- e) Comprueba la coherencia del planteamiento sumando los resultados de los apartados c y d, obteniendo el tiempo de ejecución del programa con el coprocesador instalado.

	Tarea sin punto flotante	Tarea con punto flotante
Sin copro		 2,5m
Con copro		 1m

$$t_{nue} = t_{int} - t_{nue.flo} = 0,375 + 0,625 = 1 m$$

Análisis de  
rendimiento

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

Análisis de  
rendimiento

### Relación rendimiento y coste

¿Por qué el cliente se decide por un computador en lugar de otro?

- ❖ Las medidas estándares del rendimiento permiten comparar cuantitativamente a los diseñadores
- ❖ Las comparativas basan sus informes en aspectos relacionados con el rendimiento y coste de compra



## 2.1 Rendimiento. Concepto y definiciones

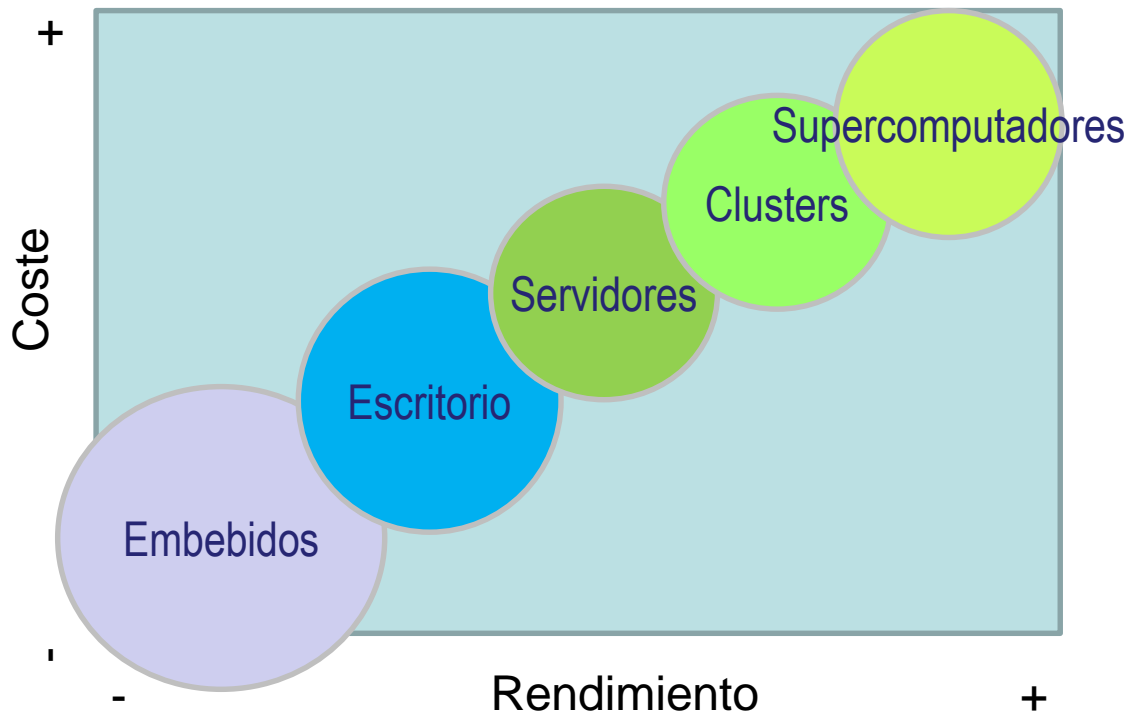
Concepto

Amdahl

Relación

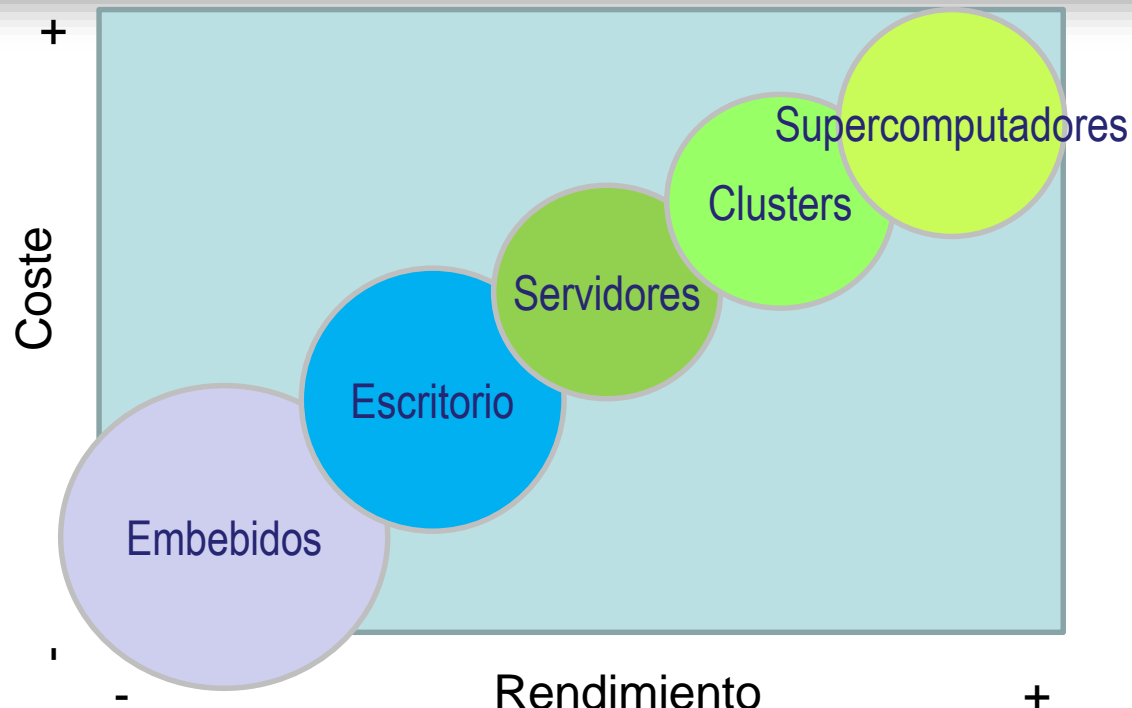
### Relación rendimiento y coste

- El diseño de computadores abarca desde el alto coste alto rendimiento hasta el bajo coste bajo rendimiento

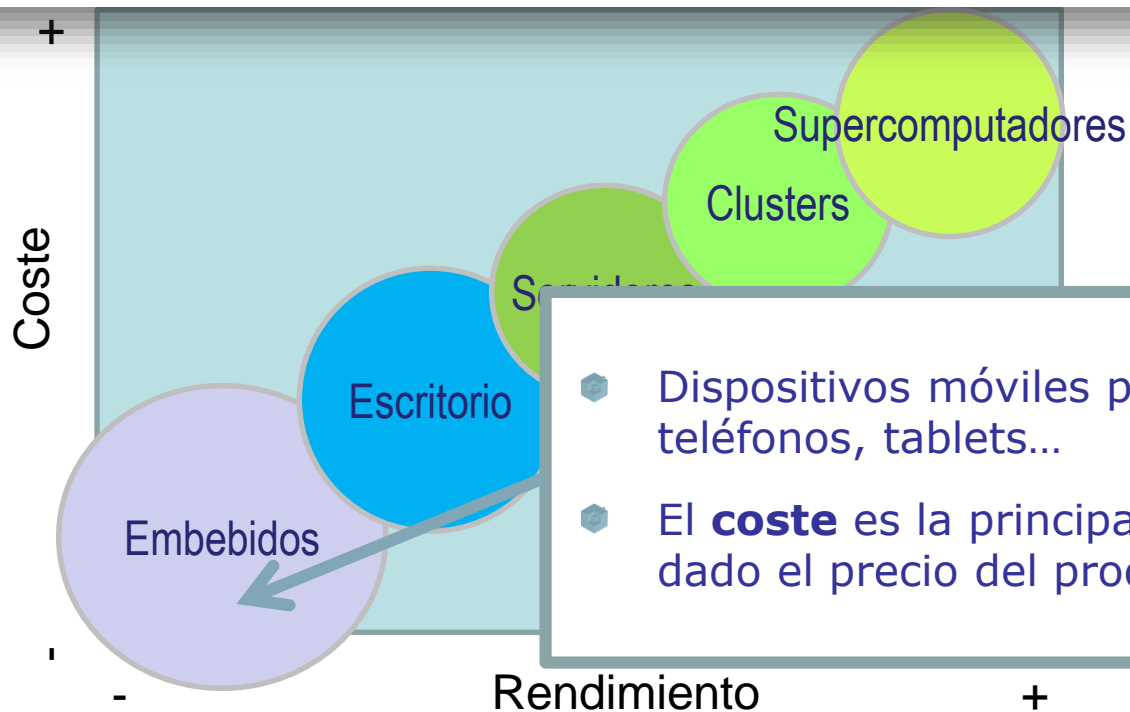


Análisis de  
rendimiento

Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Embedded
Price of system	\$100–\$1000	\$300–\$2500	\$5000–\$10,000,000	\$100,000–\$200,000,000	\$10–\$100,000
Price of micro-processor	\$10–\$100	\$50–\$500	\$200–\$2000	\$50–\$250	\$0.01–\$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance

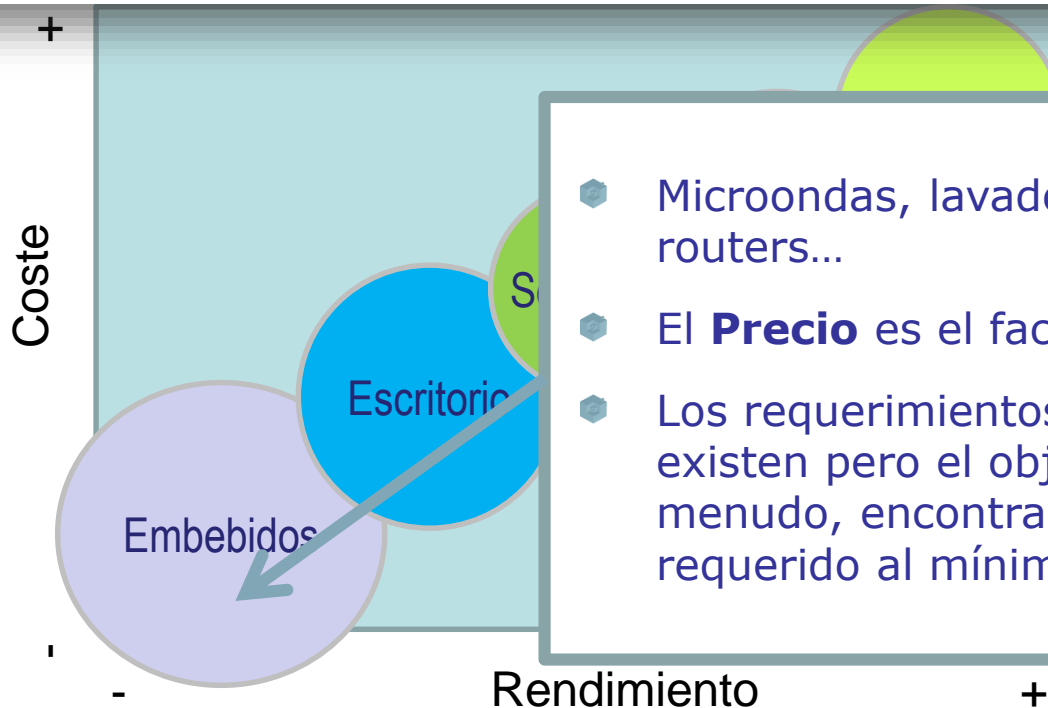


Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Embedded
Price of system	\$100–\$1000	\$300–\$2500	\$5000–\$10,000,000	\$100,000–\$200,000,000	\$10–\$100,000
Price of micro-processor	\$10–\$100	\$50–\$500	\$200–\$2000	\$50–\$250	\$0.01–\$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance



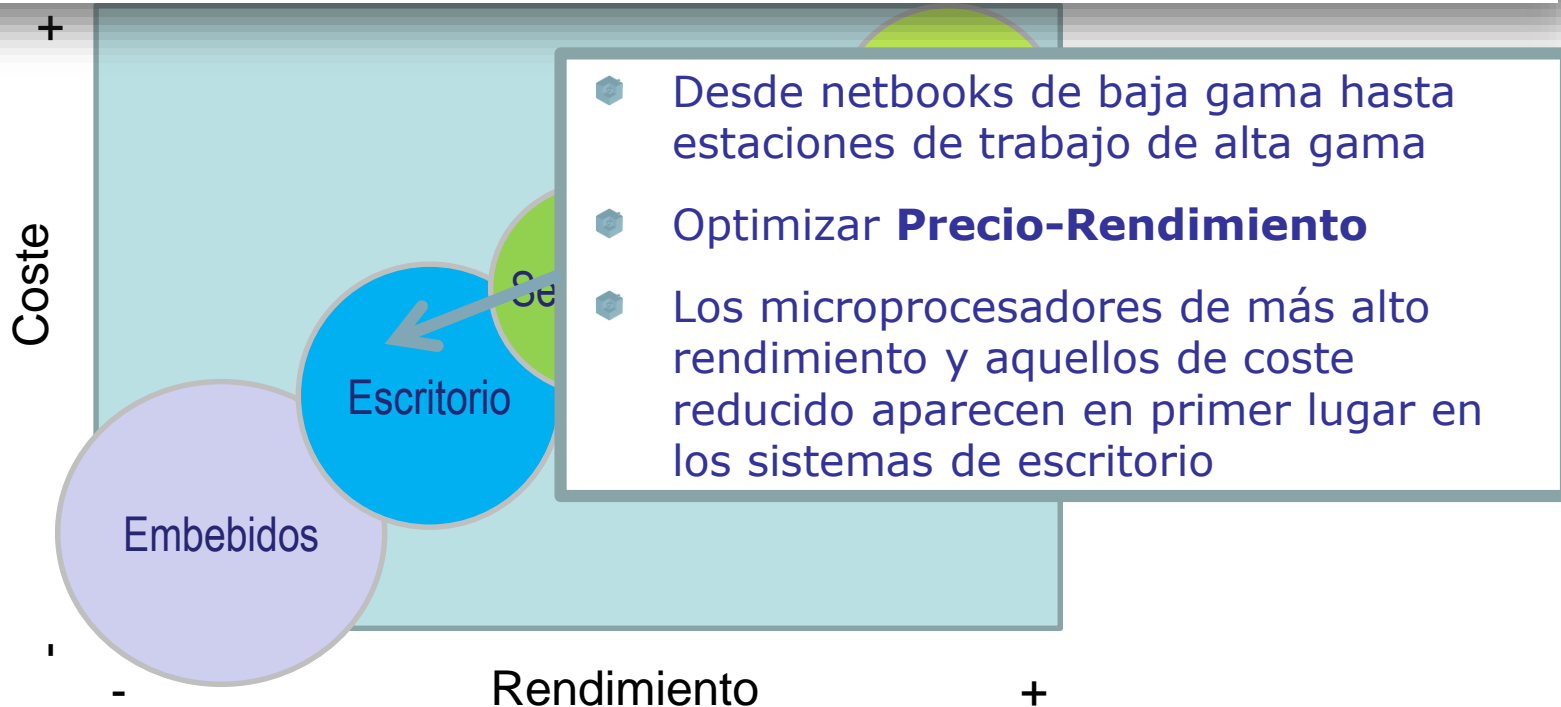
- Dispositivos móviles personales: teléfonos, tablets...
- El **coste** es la principal preocupación dado el precio del producto

Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Embedded
Price of system	\$100–\$1000	\$300–\$2500	\$5000–\$10,000,000	\$100,000–\$200,000,000	\$10–\$100,000
Price of micro-processor	\$10–\$100	\$50–\$500	\$200–\$2000	\$50–\$250	\$0.01–\$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance



- Microondas, lavadoras, impresoras, routers...
- El **Precio** es el factor clave
- Los requerimientos de rendimiento existen pero el objetivo principal es, a menudo, encontrar el rendimiento requerido al mínimo precio

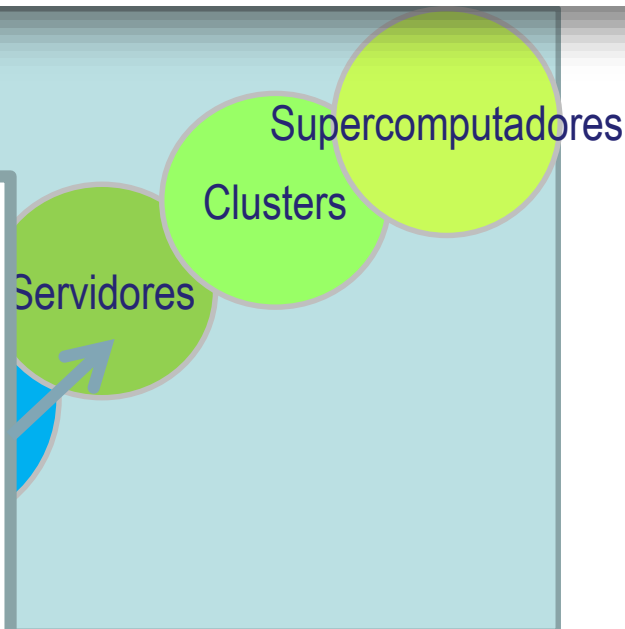
Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Embedded
Price of system	\$100–\$1000	\$300–\$2500	\$5000–\$10,000,000	\$100,000–\$200,000,000	\$10–\$100,000
Price of micro-processor	\$10–\$100	\$50–\$500	\$200–\$2000	\$50–\$250	\$0.01–\$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance



Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Embedded
Price of system	\$100–\$1000	\$300–\$2500	\$5000–\$10,000,000	\$100,000–\$200,000,000	\$10–\$100,000
Price of micro-processor	\$10–\$100	\$50–\$500	\$200–\$2000	\$50–\$250	\$0.01–\$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance

+

- ◆ Diseñados para una **productividad** eficiente
- ◆ Rendimiento global del servidor—en términos de transacciones por minuto o páginas web servidas por segundo—es crucial

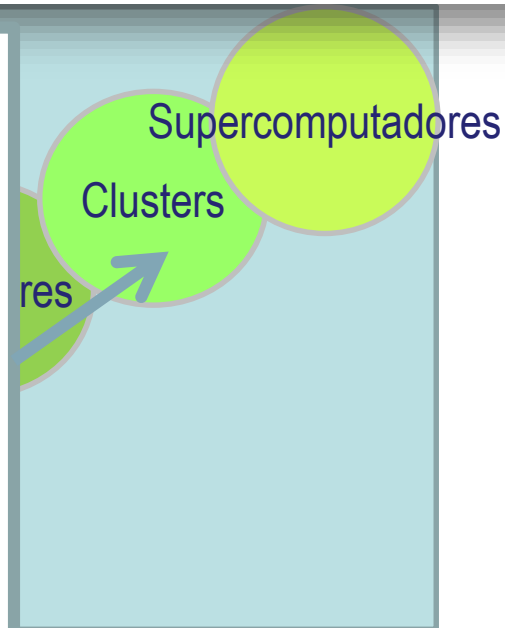


Rendimiento

+

Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Embedded
Price of system	\$100–\$1000	\$300–\$2500	\$5000–\$10,000,000	\$100,000–\$200,000,000	\$10–\$100,000
Price of micro-processor	\$10–\$100	\$50–\$500	\$200–\$2000	\$50–\$250	\$0.01–\$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance

- Colecciones de computadores de escritorio o servidores conectados por redes de área local que actúan como un solo computador más grande
- Relacionados con aplicaciones de búsqueda, redes sociales, compartición de video, juegos multijugador...
- **Precio-rendimiento** es crítico



Rendimiento

+

Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Embedded
Price of system	\$100–\$1000	\$300–\$2500	\$5000–\$10,000,000	\$100,000–\$200,000,000	\$10–\$100,000
Price of micro-processor	\$10–\$100	\$50–\$500	\$200–\$2000	\$50–\$250	\$0.01–\$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance

- Énfasis en el rendimiento del punto flotante, capaces de ejecutar grandes programas por lotes que pueden correr por semanas
- **Rendimiento** es crítico. El coste es menos importante

Supercomputadores  
clusters



Rendimiento

+



## 2.1 Rendimiento. Concepto y definiciones

### Coste

Concepto

Amdahl

Relación

- El coste es un parámetro a tener muy en cuenta al diseñar un nuevo procesador o al modificar uno existente
- Los factores principales que influyen en el coste de un computador son:
  - Curva de aprendizaje:**
    - Costes de manufacturación decrecen a lo largo del tiempo incluso sin mejoras en la tecnología de implementación básica
    - El porcentaje de dispositivos manufacturados que pasan los procedimientos de prueba se incrementa a lo largo del tiempo

Análisis de  
rendimiento

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

Análisis de  
rendimiento

### Coste

- El coste es un parámetro a tener muy en cuenta al diseñar un nuevo procesador o al modificar uno existente
- Los factores principales que influyen en el coste de un computador son:
  - Curva de aprendizaje:**
  - Volumen:** El incremento del volumen afecta al coste en:
    - Reduciendo el tiempo necesario para bajar la curva de aprendizaje
    - Incrementando la eficiencia de compra y manufactura (10% menos por cada doble de volumen)
    - Reduciendo la cantidad del coste de diseño que debe ser amortizado para cada computador

## 2.1 Rendimiento. Concepto y definiciones

### Coste

Concepto

Amdahl

Relación

- El coste es un parámetro a tener muy en cuenta al diseñar un nuevo procesador o al modificar uno existente
- Los factores principales que influyen en el coste de un computador son:
  - Curva de aprendizaje:**
  - Volumen:**
  - Mercado altamente competitivo:**
    - Las DRAMs, discos, monitores, teclados son vendidos por diferentes fabricantes y son esencialmente idénticos
    - La competencia reduce la distancia entre el precio de venta y el coste, pero también reduce el coste porque:
      - Los componentes tienen tanto un gran volumen como una clara definición

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

Análisis de  
rendimiento

### Coste de un circuito integrado (IC)

- Los costes de los ICs se están convirtiendo en una porción cada vez más grande del coste que varía entre computadores
- Los factores que influyen en el coste del silicio son:
  - El número de puertas:** influye en el número de transistores necesarios. Un aumento de estos requiere un área de silicio mayor
  - Conexiones** entre elementos: el número y la longitud
  - Regularidad** del diseño: cuanto más regular sea el diseño, menos área ocupará
- Los procesos de IC están caracterizados por el "feature size":
  - Tamaño mínimo de un transistor o conexión sea en la dimensión X o Y (10 micras en 1971 a 0.032 micras en 2011)

## 2.1 Rendimiento. Concepto y definiciones

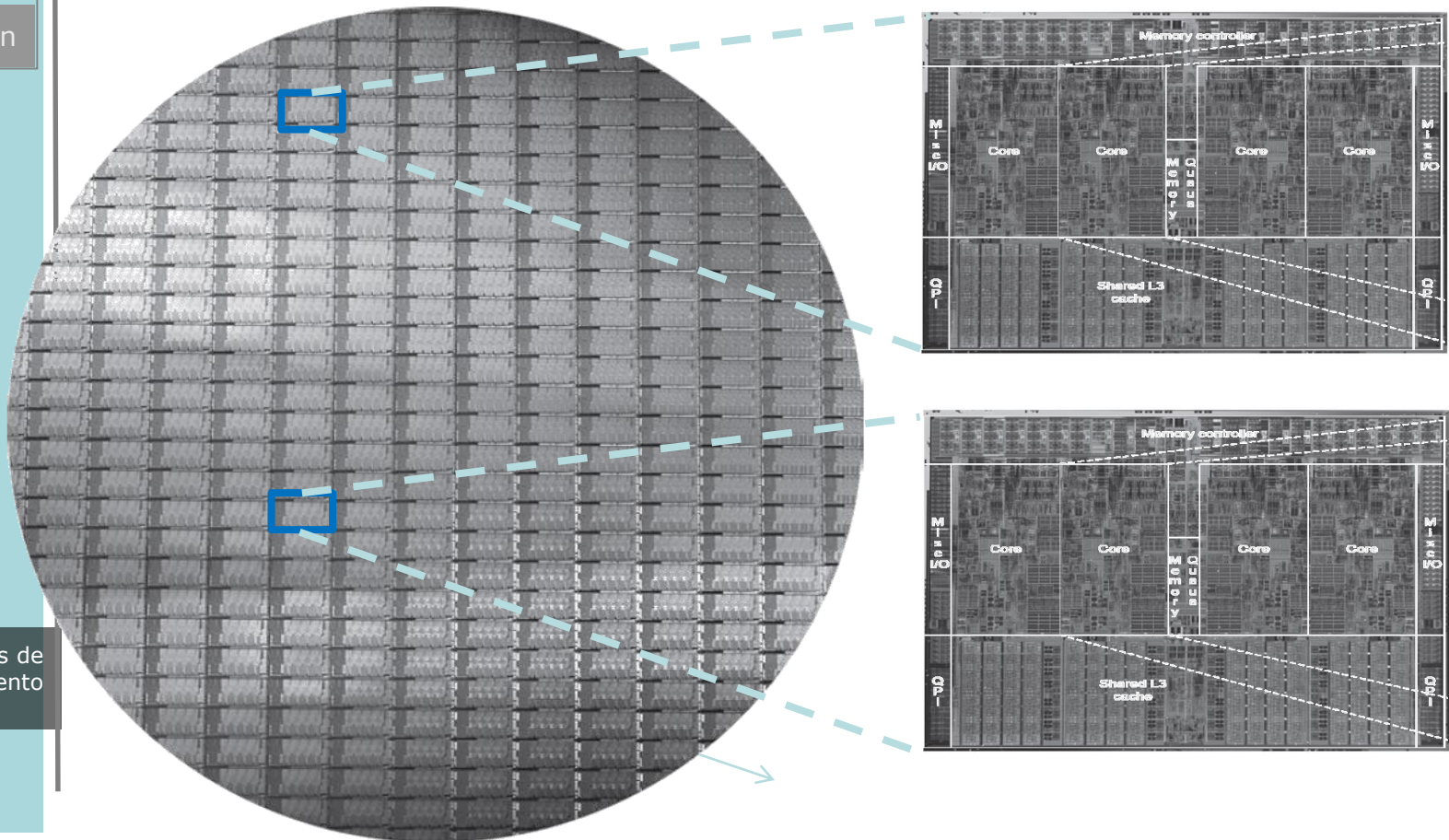
Microprocessor	16-bit address/ bus, microcoded	32-bit address/ bus, microcoded	5-stage pipeline, on-chip I & D caches, FPU	2-way superscalar, 64-bit bus	Out-of-order 3-way superscalar	Out-of-order superpipelined, on-chip L2 cache	Multicore OOO 4-way on chip L3 cache, Turbo
Product	Intel 80286	Intel 80386	Intel 80486	Intel Pentium	Intel Pentium Pro	Intel Pentium 4	Intel Core i7
Year	1982	1985	1989	1993	1997	2001	2010
Die size (mm <sup>2</sup> )	47	43	81	90	308	217	240
Transistors	134,000	275,000	1,200,000	3,100,000	5,500,000	42,000,000	1,170,000,000
Processors/chip	1	1	1	1	1	1	4
Pins	68	132	168	273	387	423	1366
Latency (clocks)	6	5	5	5	10	22	14
Bus width (bits)	16	32	32	64	64	64	196
Clock rate (MHz)	12.5	16	25	66	200	1500	3333
Bandwidth (MIPS)	2	6	25	132	600	4500	50,000
Latency (ns)	320	313	200	76	50	15	4

- Los procesos de IC están caracterizados por el "feature size":
  - Tamaño mínimo de un transistor o conexión sea en la dimensión X o Y (10 micras en 1971 a 0.032 micras en 2011)

## 2.1 Rendimiento. Concepto y definiciones

### Coste de un circuito integrado (IC)

- El proceso básico de fabricación del silicio no ha cambiado: la oblea es testeada y cortada en dados que son empaquetados



Concepto

Amdahl

Relación

Análisis de  
rendimiento



## 2.1 Rendimiento. Concepto y definiciones

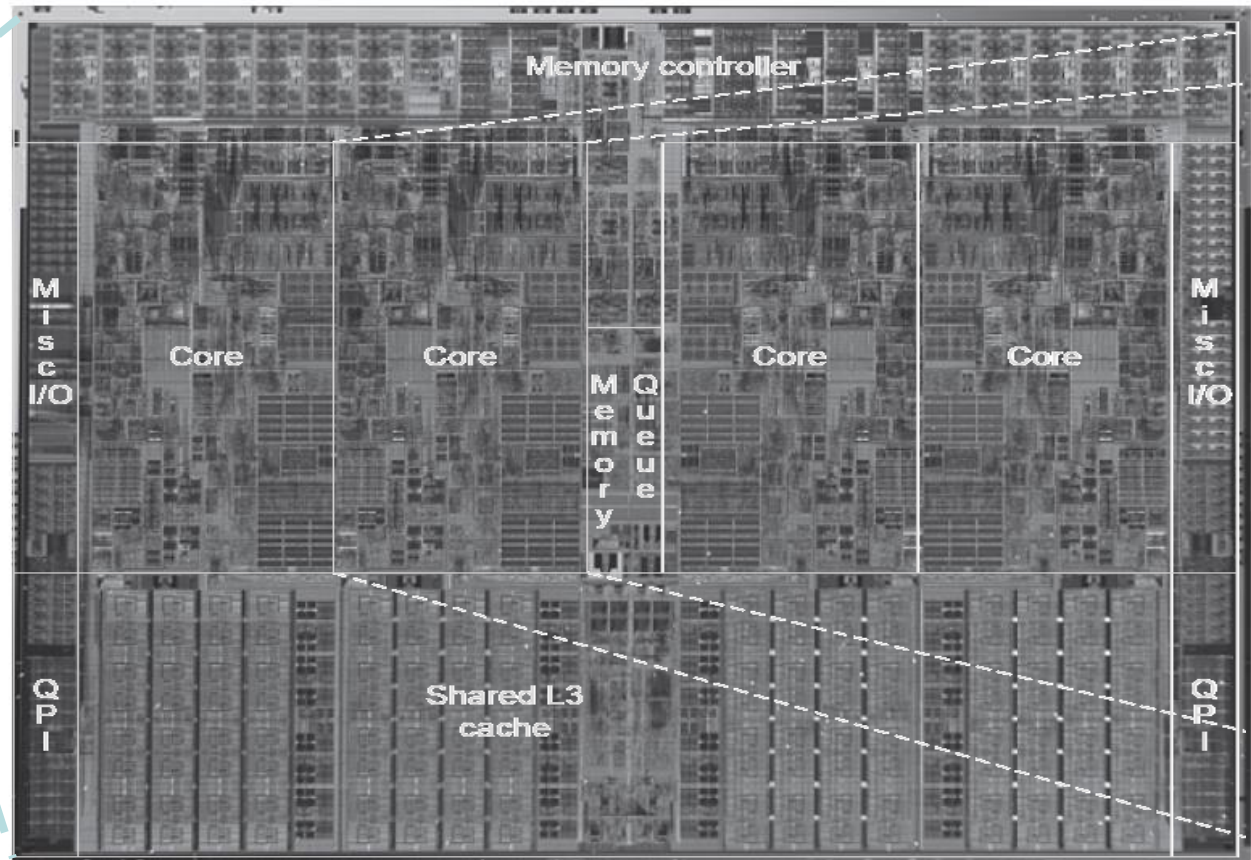
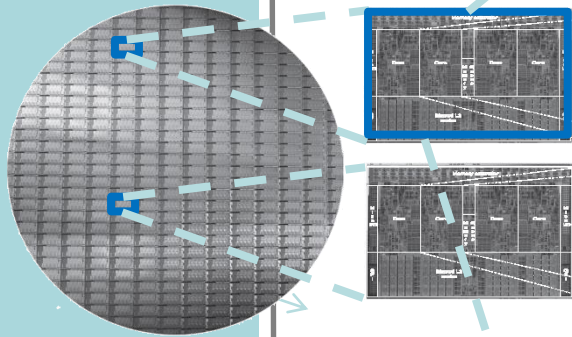
Concepto

Amdahl

Relación

### Coste de un circuito integrado (IC)

- El proceso básico de fabricación del silicio no ha cambiado: la oblea es testeada y cortada en dados que son empaquetados



Análisis de  
rendimiento

## 2.1 Rendimiento. Concepto y definiciones

Concepto

Amdahl

Relación

### Coste de un circuito integrado (IC)

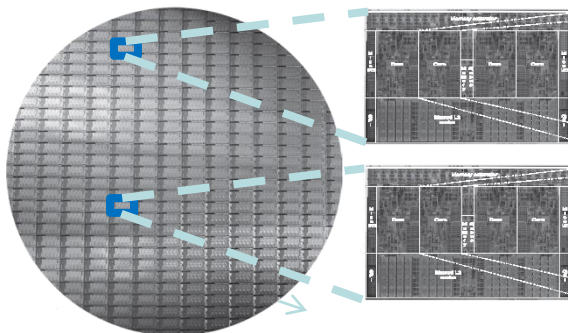
- El proceso básico de fabricación del silicio no ha cambiado: la oblea es testeada y cortada en dados que son empaquetados

$$\text{Cost of integrated circuit} = \frac{\text{Cost of die} + \text{Cost of testing die} + \text{Cost of packaging and final test}}{\text{Final test yield}}$$

$$\text{Cost of die} = \frac{\text{Cost of wafer}}{\text{Dies per wafer} \times \text{Die yield}}$$

- El número de dados por oblea es aproximadamente el área de la oblea dividida por el área del dado. De forma más precisa, puede estimarse por:

$$\text{Dies per wafer} = \frac{\pi \times (\text{Wafer diameter}/2)^2}{\text{Die area}} - \frac{\pi \times \text{Wafer diameter}}{\sqrt{2} \times \text{Die area}}$$



- Compensa los dados cerca de la periferia de la oblea. Aproximadamente el número de dados a lo largo del borde

Análisis de  
rendimiento



## 2.1 Rendimiento. Concepto y definiciones

### Coste de un circuito integrado (IC)

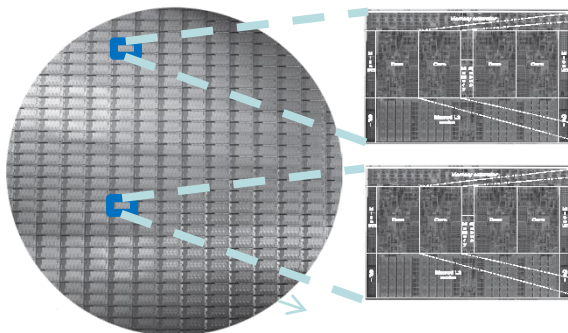
- Este es el numero máximo de dados por oblea:

$$\text{Dies per wafer} = \frac{\pi \times (\text{Wafer diameter}/2)^2}{\text{Die area}} - \frac{\pi \times \text{Wafer diameter}}{\sqrt{2} \times \text{Die area}}$$

- ¿Cual es la fracción de dados no defectuosos en la óblea, o el rendimiento del dado (die yield)? Asumiendo que:
  - Los defectos están distribuidos aleatoriamente
  - el rendimienndo es inversamente proporcional a la complejidad del proceso de fabricación

$$\text{Die yield} = \text{Wafer yield} \times 1 / (1 + \text{Defects per unit area} \times \text{Die area})^N$$

- Fórmula Bose–Einstein: modelo empírico teniendo en cuenta el rendimiento en muchas líneas de fabricación



- Defectos por unidad de área = 0.016-0.057 defectos por cm<sup>2</sup> (2010)
- N = factor de complejidad del proceso = 11.5-15.5 (40 nm, 2010)

Concepto

Amdahl

Relación

Análisis de  
rendimiento

## 2.1 Rendimiento. Concepto y definiciones

### Ejercicio

- Intel utiliza obleas de 300mm para construir su Core i7. Estas obleas tienen un coste de 20000€. El dado del Core i7 tiene unas dimensiones 20.7 mm x 10.5 mm. Asumiendo que el proceso de fabricación de Intel tiene una densidad de defectos de 0.023 por  $\text{cm}^2$  y que el factor de complejidad del proceso para un tamaño mínimo del transistor de 32 nm es 13.5, calcula el coste del dado. ¿Cuál sería el coste del dado dentro de 2 años, sabiendo que el factor de complejidad del proceso para la tecnología de 32nm decrece un 5% por año?

Concepto

Amdahl

Relación

Análisis de  
rendimiento

## 2.2. Evaluación del rendimiento

**Tema 2 Análisis del rendimiento**

**Arquitectura de los Computadores**

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

### Tiempo de ejecución

- El tiempo es la medida más fiable del rendimiento
- El tiempo de ejecución de un programa se mide en segundos

### El rendimiento

- La relación entre el tiempo y el rendimiento es inversa
- El rendimiento se mide como una frecuencia de eventos por segundo

$$\text{Rendimiento} = \frac{1}{\text{tiempo}}$$

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

Análisis de  
rendimiento

### Tiempo de programa / Tiempo de CPU

- ⚙ **Tiempo de reloj, tiempo de respuesta, tiempo transcurrido:** Latencia para completar una tarea incluyéndolo todo: accesos a disco, accesos a memoria, actividades de entrada salida, gastos del sistema operativo, multiprogramación...
- ⚙ **Rendimiento del sistema:** Este término se utiliza para referenciar el tiempo transcurrido en un sistema no cargado.
- ⚙ **Tiempo de CPU**
  - ⚙ CPU usuario + CPU sistema
  - ⚙ Tiempo en que la CPU está calculando sin incluir tiempos de espera para E/S o para ejecución de otros programas
  - ⚙ **Tiempo de programa:** Este término se refiere al tiempo de CPU del usuario (nos centraremos en rendimiento de la CPU)

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

### Tiempo de programa / Tiempo de CPU

- **Tiempo de reloj, tiempo de respuesta, tiempo transcurrido:** Latencia para completar una tarea incluyéndolo todo: accesos a disco, accesos a memoria, actividades de entrada salida, gastos del sistema operativo, multiprogramación...
- **Rendimiento del sistema:** Este término se utiliza para referenciar el tiempo transcurrido en un sistema no cargado.

La función *time* de Unix produce una salida de la forma: 90.7u 12.9s 2:39 65%, donde:

- Tiempo de CPU del usuario = 90.7 segundos
- Tiempo de CPU utilizado por el sistema = 12.9 segundos
- Tiempo de CPU = 90.7 seg. + 12.9seg = 103.6
- Tiempo de respuesta = 2 minutos 39 segundos = 159 segundos
- Tiempo de CPU = 65% del tiempo de respuesta = 159 segundos \* 0.65 = 103.6
- Tiempo esperando operaciones de E/S y/o el tiempo ejecutando otras tareas 35% del tiempo de respuesta = 159 segundos \* 0.35 = 55.6 segundos

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

### Tiempo de programa / Tiempo de CPU

- El tiempo de CPU de un programa puede expresarse en función del ciclo de reloj

$$\text{Tiempo de CPU} = \frac{\text{Ciclos de reloj de CPU para un programa}}{\text{Duración del ciclo de reloj}}$$

$$\text{Tiempo de CPU} = \frac{\text{Ciclos de reloj de CPU para un programa}}{\text{Frecuencia de reloj}}$$

- No tiene sentido mostrar el tiempo transcurrido en función del ciclo de reloj ya que la latencia de los dispositivos de entrada salida es independiente del ciclo de reloj de la CPU

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

Análisis de  
rendimiento

### CPI

- Número medio de ciclos de reloj por instrucción. Se expresa en función del número de ciclos de reloj y el número de instrucciones ejecutadas

$$\text{CPI} = \frac{\text{Ciclos de reloj de CPU para un programa}}{\text{Recuento de instrucciones}}$$

- Podemos expresar el tiempo de CPU en función del CPI:

$$\text{Tiempo de CPU} = \frac{\text{Ciclos de reloj de CPU para un programa}}{\text{Ciclos de reloj de CPU para un programa}} * \text{Duración del ciclo de reloj}$$

$$\text{Tiempo de CPU} = \frac{\text{Recuento de instrucciones}}{\text{Recuento de instrucciones}} * \text{CPI} * \text{Duración del ciclo de reloj}$$

$$\text{Tiempo de CPU} = \text{RI} * \text{CPI} * \text{clk}$$



## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

**En ocasiones se detalla el CPI por cada tipo de instrucción estática  $i$**

$$\text{Ciclos de reloj de la CPU} = \sum_{i=1}^n (CPI_i \cdot I_i)$$

$I_i$  = instrucciones dinámicas para cada tipo de instrucción estática  $i$ .

$CPI_i$  = Número medio de ciclos de reloj para la instrucción tipo  $i$ .

Esto nos permite expresar

$$\text{Tiempo de CPU} = \sum_{i=1}^n (CPI_i \cdot I_i) \cdot \text{Duración del ciclo de reloj}$$

$$CPI = \frac{\sum_{i=1}^n (CPI_i \cdot I_i)}{\text{recuento de instrucciones}} = \sum_{i=1}^n (CPI_i \cdot \frac{I_i}{\text{recuento de instrucciones}})$$

•  $CPI_i$  medido, no obtenido de tabla de referencia. Deben considerarse fallos de cache y demás incidencias del sistema de memoria.

## 2.2 Evaluación del rendimiento

Métricas

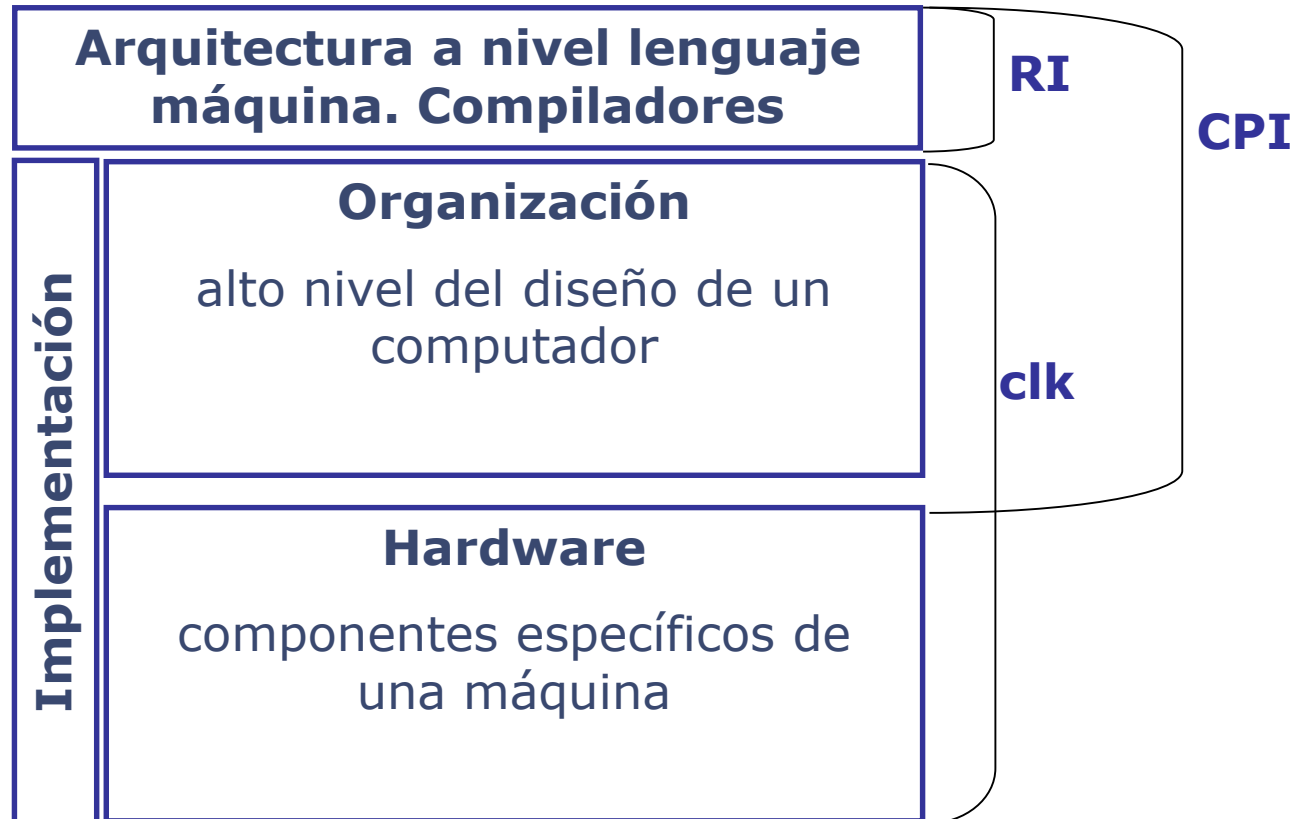
Benchmarks

Formulación

Análisis de  
rendimiento

### Tres parámetros interdependientes

$$\text{Tiempo de CPU} = \text{RI} * \text{CPI} * \text{clk}$$



## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

**Ejemplo:** Suponer que estamos considerando dos alternativas para una instrucción de salto condicional:

CPU A. Una instrucción de comparación inicializa un código de condición y es seguida por un salto que examina el código de condición.

CPU B. Se incluye una comparación en el salto.

En ambas CPU, la instrucción de salto condicional emplea 2 ciclos de reloj, y las demás instrucciones 1 (en este sencillo ejemplo se están despreciando las pérdidas del sistema de memoria). En la CPU A, el 20% de todas las instrucciones ejecutadas son saltos condicionales; como cada salto necesita una comparación, otro 20% de las instrucciones son comparaciones. Debido a que la CPU A no incluye la comparación en el salto, su ciclo de reloj es un 25% más rápido que el de la CPU B. ¿Qué CPU es más rápida?.

# Arquitectura de Computadores

Métricas

Benchmarks

Rendimiento

Análisis de  
rendimiento

**CPU A 20% saltos (2); 20% compara (1); 60% restos (1)**

$$CPI_A = 0,2 * 2 + 0,8 * 1 = 1,2$$

Saltos

resto

$$CLK_B = CLK_A + 0,25 * CLK_A = 1,25 CLK_A$$

$$T_{CPU.A} = RI_A * 1,2 * CLK_A$$

**CPU B**

No ejecuta las comparaciones -> 20% saltos (2); 60% restos (1) = 80% total

20/80=0,25 -> saltos (2) ; 60/80=0,75 -> resto (1)

$$CPI_B = 0,25 * 2 + 0,75 * 1 = 1,25$$

Saltos

resto

$$RI_B = 80/100 * RI_A$$

$$T_{CPU.B} = (0,8 * RI_A) * 1,25 * (1,25 * CLK_A) = 1,25 * RI_A * CLK_A$$

$$a = T_{CPU.B} / T_{CPU.A} = 1,25 / 1,2 = 1,042 \rightarrow n = 4,2\%$$

# Arquitectura de Computadores

Métricas

Benchmarks

Rendimiento

**Ejemplo:** Después de ver el análisis, un diseñador consideró que, volviendo a trabajar en la organización, la diferencia de las duraciones de los ciclos de reloj podía reducirse, fácilmente, a un 10%. ¿Qué CPU es más rápida ahora?.

$$\text{CLK}_B = \text{CLK}_A + 0,1 * \text{CLK}_A = 1,1 \text{ CLK}_A$$

## CPU A

$$T_{\text{CPU.A}} = 1,2 * \text{RI}_A * \text{CLK}_A$$

## CPU B

$$T_{\text{CPU.B}} = (0,8 * \text{RI}_A) * 1,25 * (1,1 * \text{CLK}_A) = 1,1 * \text{RI}_A * \text{CLK}_A$$

$$a = T_{\text{CPU.A}} / T_{\text{CPU.B}} = 1,2 / 1,1 = 1,09 \rightarrow n = 9 \%$$

Análisis de  
rendimiento

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

**Ejemplo:** Supongamos que estamos considerando otro cambio en un repertorio de instrucciones. La máquina, inicialmente, sólo tiene instrucciones de carga y de almacenamiento en memoria, y, después, todas las operaciones se realizan en los registros. Tales máquinas se denominan máquinas de carga almacenamiento (load/store). A continuación observamos medidas de la máquina de carga almacenamiento que muestran la frecuencia de instrucciones, denominada mezcla de instrucciones (instruction mix) y número de ciclos de reloj por instrucción.

Operación	Frecuencia	Cuenta de ciclos de reloj
Ops ALU	43%	1
Load	21%	2
store	12%	2
Saltos	24%	2

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

Supongamos que el 25% de las operaciones de la unidad aritmético lógica (ALU) utilizan directamente un operando cargado que no se utiliza de nuevo.

Proponemos añadir instrucciones a la ALU que tengan un operando fuente en memoria. Estas nuevas instrucciones de registro memoria emplean dos ciclos de reloj. Supongamos que el repertorio extendido de instrucciones incrementa en 1 el número de ciclos de reloj para los saltos, pero sin afectar a la duración del ciclo de reloj. ¿Mejorará este cambio el rendimiento de la CPU?.

Operación	Frecuencia	Cuenta de ciclos de reloj
Ops ALU	43%	1
Load	21%	2
store	12%	2
Saltos	24%	2

# Arquitectura de Computadores

Métricas

Benchmarks

Rendimiento

CPU<sub>A</sub>

$$CPI_A = 0,43 * 1 + 0,21 * 2 + 0,12 * 2 + 0,24 * 2 = 1,57$$

alu

load

store

saltos

$$T_{CPU.A} = RI_A * 1,57 * CLK_A$$

CPU<sub>B</sub>

Alu

Alu-mem

Load mejor

alm

salt

$$CPI_B = \frac{(0,75 * 0,43 * 1) + (0,25 * 0,43 * 2) + (0,21 - 0,25 * 0,43) * 2 + 0,12 * 2 + 0,24 * 3}{1 - 0,25 * 0,43}$$

$$= 1,703 / 0,893 = 1,908$$

$$T_{CPU.B} = (0,893 * RI_A) * 1,908 * CLK_A = 1,703 * RI_A * CLK_A$$

$$a = T_{CPU.B} / T_{CPU.A} = 1,703 / 1,57 = 1,085 \rightarrow n = 8,5\%$$

Análisis de  
rendimiento



## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

### Alternativas para la medida del rendimiento

- La medida más fiable del rendimiento es el **tiempo de ejecución** de los **programas reales**
- Alternativas al tiempo como medida del rendimiento y a los programas reales como objetos de medida han conducido a errores en el diseño de computadores

**MIPS** Millones de instrucciones por segundo

$$MIPS = \frac{\text{recuento de instrucciones}}{\text{Tiempo de ejecución} \cdot 10^6}$$

- Los MIPS se muestran como un parámetro intuitivo para reflejar el rendimiento. Máquinas más rápidas tienen MIPS más altos.

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

## Alternativas para la medida del rendimiento

### Considerando

$$\text{Re cuenta de instrucciones} = \frac{\text{tiempo de ejecución}}{\text{CPI} \cdot \text{ciclo de reloj}}$$

$$\text{MIPS} = \frac{\text{recuento de instrucciones}}{\text{Tiempo de ejecución} \cdot 10^6}$$

$$\text{MIPS} = \frac{\frac{\text{tiempo de ejecución}}{\text{CPI} \cdot \text{ciclo de reloj}}}{\text{Tiempo de ejecución} \cdot 10^6} = \frac{1}{\text{CPI} \cdot \text{ciclo de reloj} \cdot 10^6}$$

$$\text{MIPS} = \frac{\text{Frecuencia de reloj}}{\text{CPI} \cdot 10^6}$$

## 2.2 Evaluación del rendimiento

### Métricas

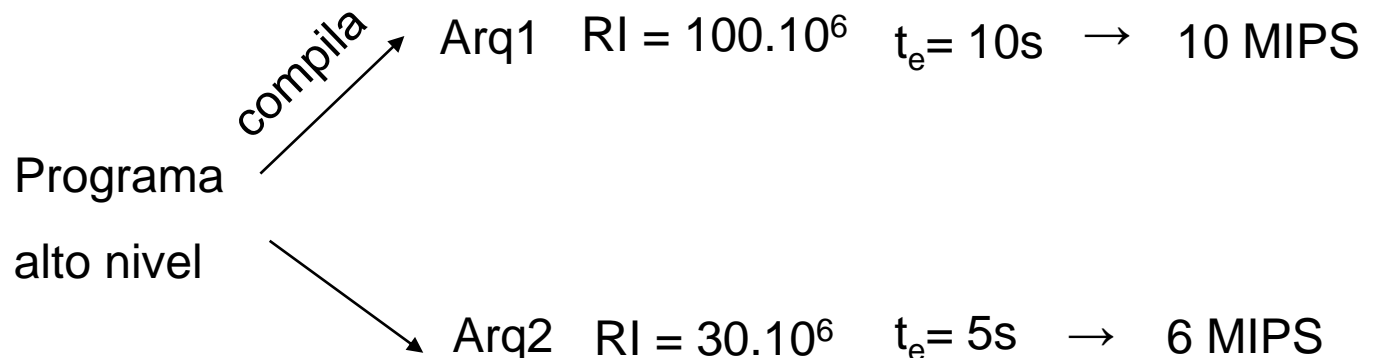
### Benchmarks

### Formulación

Análisis de  
rendimiento

### Problemas derivados de la utilización de los MIPS

- ❖ **Dependientes del repertorio de instrucciones.** No es aconsejable comparar los MIPS de computadores con repertorios de instrucciones diferentes.
  - ❖ Reflejan el ritmo de ejecución de instrucciones
  - ❖ No reflejan la efectividad del repertorio RI
  - ❖ Los MIPS pueden variar inversamente al rendimiento.
- ❖ **Dependencia del programa** en el mismo computador.
- ❖ **Ejemplo:**



## 2.2 Evaluación del rendimiento

### Métricas

Benchmarks

Formulación

Análisis de  
rendimiento



**Ejemplo:** Supongamos que construimos un compilador optimizado para la máquina de carga/almacenamiento descrita en el ejemplo anterior. El compilador descarta el 50% de las instrucciones de la ALU aunque no pueda reducir cargas, almacenamientos ni saltos. Ignorando las prestaciones del sistema y suponiendo una duración del ciclo de reloj de 20 ns (frecuencia de reloj de 50Mh). ¿Cuál es la frecuencia en MIPS para el código optimizado frente al código sin optimizar? ¿Está el criterio de los MIPS de acuerdo con el del tiempo de ejecución?.

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

### MIPS relativos y MIPS nativos

- ⚙ **Tiempo referencia**= tiempo de ejecución de un programa en la máquina de referencia
- ⚙ **Tiempo no estimado** = tiempo de ejecución del mismo programa en la máquina que se va a medir
- ⚙ **MIPS** = estimación de los MIPS de la máquina de referencia

$$MIPS_{relativos} = \frac{Tiempo_{referencia}}{Tiempo_{no\ estimado}} \cdot MIPS_{referencia}$$

- ⚙ Los MIPS relativos se apoyan en el tiempo de ejecución
- ⚙ En los años 80 la máquina dominante como referencia era la VAX-11/780, denominada máquina de 1 MIPS.

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

### Análisis de rendimiento

## FLOPS

**FLOPS** = Operaciones de punto flotante por segundo

$$MFLOPS = \frac{\text{Número de operaciones en punto flotante de un programa}}{\text{Tiempo de ejecución} \cdot 10^6}$$

$$GFLOPS = \frac{MFLOPS}{10^6}$$

❖ **Problemas** derivados de la utilización de los FLOPS

❖ **Dependencia del repertorio**

❖ Término basado en operaciones con objetivo de poder utilizarlo para comparar diferentes máquinas.

❖ El conjunto de operaciones en punto flotante no es consistente en diferentes máquinas (CRAY-2 no tiene instrucciones de dividir mientras que el motorola 68882 si).

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

### Análisis de rendimiento

## FLOPS

**FLOPS** = Operaciones de punto flotante por segundo

$$MFLOPS = \frac{\text{Número de operaciones en punto flotante de un programa}}{\text{Tiempo de ejecución} \cdot 10^6}$$

$$GFLOPS = \frac{MFLOPS}{10^6}$$

- ❖ **Problemas** derivados de la utilización de los FLOPS
  - ❖ **Dependencia del repertorio**
  - ❖ **Dependencia del programa**
  - ❖ La estimación de los MFLOPS cambia según la mezcla de operaciones rápidas y lentas en punto flotante del programa.
  - ❖ Si el 100% de las operaciones en punto flotante son sumas, la estimación será mayor que si el 100% son divisiones

## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

Análisis de  
rendimiento

### FLOPS Normalizados

- 🔧 Solución a los problemas de FLOPS
- 🔧 Operaciones normalizadas (Livermore Loops)

Operaciones reales PF	Operaciones normalizadas PF
ADD, SUB, COMPARE, MULT	1
DIVIDE, SQRT,	4
EXP, SIN	8

#### 🔧 Programa 1

$4 \cdot 10^6$  sumas

$3 \cdot 10^6$  div

$t_e = 10s$

#### MFLOPS Nativos

$$\frac{7 \cdot 10^6}{10 \cdot 10^6} = 0,7 MFLOPS$$

#### MFLOPS Normalizados

$$\frac{16 \cdot 10^6}{10 \cdot 10^6} = 1,6 MFLOPS$$

#### 🔧 Programa 2

$7 \cdot 10^6$  div

$t_e = 17s$

#### MFLOPS Nativos

$$\frac{7 \cdot 10^6}{17 \cdot 10^6} = 0,4 MFLOPS$$

#### MFLOPS Normalizados

$$\frac{28 \cdot 10^6}{17 \cdot 10^6} = 1,65 MFLOPS$$



## 2.2 Evaluación del rendimiento

### Métricas

### Benchmarks

### Formulación

Análisis de  
rendimiento



**Ejemplo:** EL programa SPICE se ejecuta en la DECstation 3100 en 94 segundos. El número de operaciones en punto flotante ejecutadas en ese programa es el de la tabla. ¿Cuántos son los MFLOPS nativos para ese programa? Usando las conversiones ¿Cuántos son los MFLOPS normalizados?.

ADDD	25.999.440
SUBD	18.266.439
MULD	33.880.810
DIVD	15.682.333
COMPARED	9.745.930
NEGD	2.617.846
ABSD	2.195.930
CONVERTD	1.581.450
<i>Total</i>	109.970.178

# Arquitectura de Computadores

Métricas

Benchmarks

Rendimiento

- ❖ **Ejemplo:** EL programa SPICE se ejecuta en la DECstation 3100 en 94 segundos. El número de operaciones en punto flotante ejecutadas en ese programa es el de la tabla.  
¿Cuántos son los MFLOPS nativos para ese programa? Usando las conversiones ¿Cuántos son los MFLOPS normalizados?.

$$\text{MFLOPS}_{\text{NATIVOS}} = n^{\circ} \text{ OPF} / T_e = 110 \text{ M} / 94 \cdot 10^6 = 1,2 \text{ MFLOPS}$$

$$\text{MFLOPS}_{\text{NORM}} = 157 \text{ M} / 94 \cdot 10^6 = 1,7 \text{ MFLOPS}$$

Análisis de  
rendimiento

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

Análisis de  
rendimiento

### Programas para evaluar el rendimiento

- ⚙️ Tiempo de ejecución de la carga de trabajo del usuario (workload) (mezcla de programas y órdenes del S.O.)
- ⚙️ **Programas reales.** compiladores de C, software de tratamiento de textos como TeX y herramientas CAD como Spice
- ⚙️ **Núcleos (Kernels).** pequeños fragmentos clave de programas. Livermore Loops y Linpack.
- ⚙️ **Benchmarks reducidos (toys).** 10 y 100 líneas de código. Criba de Eratóstenes, Puzzle y clasificación rápida (quicksort).
- ⚙️ **Benchmarks Sintéticos.** se crean artificialmente intentando simular la frecuencia media de operaciones y operandos de un gran conjunto de programas. Whetstone y Dhrystone.
- ⚙️ Whetstone: instrucciones Algol principios de los años setenta.
- ⚙️ Dhrystone: Originalmente en ADA y más tarde en C y Pascal.

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

Análisis de  
rendimiento

### Programas reales frente a otros benchmarks

- ❖ Importancia para las empresas de los benchmarks: empleo de recursos para optimizar el funcionamiento de estos pero no de programas reales.
- ❖ Ejemplo extremo: empleo de optimizadores de compiladores sensibles a los benchmarks que aplican optimizaciones.
- ❖ Si se utilizasen programas reales para evaluar el rendimiento, las mejoras repercutirían en el usuario final.

### Razones de la utilización de benchmarks pequeños

- ❖ **Portabilidad:** En el pasado lenguajes de programación inconsistentes entre máquinas dificultando el transporte
- ❖ **Fácil simulación:** Cuando se diseña una nueva máquina
- ❖ **Estandarización:** Los pequeños benchmarks más fácilmente
- ❖ En la actualidad la popularidad de los sistemas operativos estándares (UNIX, WINDOWS...) elimina la principal dificultad.

## 2.2 Evaluación del rendimiento

Métricas

**Benchmarks**

Formulación

Análisis de  
rendimiento

### **Colecciones de benchmarks**

- ⚙ Medir el rendimiento de los procesadores con una variedad de aplicaciones
- ⚙ Ventajas clave: la debilidad de algún benchmark es minimizada por la presencia de otros
- ⚙ Las colecciones de benchmarks formadas por programas que pueden ser núcleos, pero fundamentalmente programas reales

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

Análisis de  
rendimiento

### Colecciones de benchmarks

- **SPEC:** El grupo System Performance Evaluation Cooperative se formó en 1988 con representantes de diversas compañías (Hewlett-Packard, DEC, MIPS, Sun) que llegaron al acuerdo de ejecutar un conjunto de programas y entradas reales.
- SPEC89, SPEC92, SPEC95, SPEC CPU2000, SPEC CPU2006.
- Desktop Benchmarks
  - SPEC CPU 2006 (CINT, CFP)
  - SPECviewperf (OpenGL Graphics Library)
  - SPECapc (ProEngineer;Solidworks;Unigraphics...)
- BenchMarks Para Servidores
  - SPECrate (productividad)
  - SPECSFS (Rendimiento sistema ficheros de red)
  - SPECWeb (Rendimiento servidores web)
- BenchMarks sistemas embebidos

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

Análisis de  
rendimiento

### Colecciones de benchmarks

- ⚙ **Otras colecciones**
- ⚙ **Business Winstone:** Proceso por lotes Netscape, Corel, WordPerfect, Microsoft...
- ⚙ **CC Winstone:** Aplicaciones creación contenido multimedia (Photoshop, Premiere, edición audio...)
- ⚙ **Winbench:** Procesos por lotes miden rendimiento CPU, video, disco,... orientado a medida por subsistemas.

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

### Formulación de los resultados de la evaluación del rendimiento

- **Reproducibilidad:** enumerar todo lo necesario para repetir los experimentos.
- SPICE tarda 94 segundos en una DECstation 3100
- Este enunciado prescinde de aspectos como:
  - Entradas del programa
  - Versión del programa
  - Versión del compilador
  - Nivel de optimización y código compilado
  - Versión del sistema operativo
  - Cantidad de memoria principal
  - Número y tipos de disco
  - Versión de la CPU

Análisis de  
rendimiento



## 2.2 Evaluación del rendimiento

Métricas





Benchmarks

**Formulación**

Análisis de  
rendimiento

### **Formulación de los resultados de la evaluación del rendimiento**

#### **Informe SPEC:**

-  **Descripción hardware**
-  **Descripción software**
-  **Descripción parámetros compilación**
-  **Publicación resultados básicos y optimizados**

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

### Formulación de los resultados de la evaluación del rendimiento

#### ● Informe SPEC:

#### ● Descripción hardware y software

##### Hardware

CPU Name: Intel Core i7-965 Extreme Edition  
CPU Characteristics: Intel Turbo Boost Technology up to 3.46 GHz  
CPU MHz: 3200  
FPU: Integrated  
CPU(s) enabled: 4 cores, 1 chip, 4 cores/chip, 2 threads/core  
CPU(s) orderable: 1 chip  
Primary Cache: 32 KB I + 32 KB D on chip per core  
Secondary Cache: 256 KB I+D on chip per core  
L3 Cache: 8 MB I+D on chip per chip  
Other Cache: None  
Memory: 12 GB (6 x 2GB Samsung M378B5673DZ1-CF8 DDR3-1066 CL7)  
Disk Subsystem: 80 GB Intel X-25M SATA Solid-State Drive  
Other Hardware: None

##### Software

Operating System: Windows Vista Ultimate w/ SP1 (64-bit)  
Compiler: Intel C++ Compiler Professional 11.0 for IA32  
Build 20080930 Package ID: w\_cproc\_p\_11.0.054  
Microsoft Visual Studio 2008 (for libraries)  
Auto Parallel: Yes  
File System: NTFS  
System State: Default  
Base Pointers: 32-bit  
Peak Pointers: 32-bit  
Other Software: None  
SmartHeap Library Version 8.1 from  
<http://www.microquill.com/>

Análisis de  
rendimiento

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

Análisis de  
rendimiento

### Formulación de los resultados de la evaluación del rendimiento

#### Informe SPEC:

- Descripción hardware y software
- Descripción parámetros compilación

#### Compiler Invocation

C benchmarks:  
icl -Qvc9 -Qc99

C++ benchmarks:  
icl -Qvc9

#### Base Optimization Flags

C benchmarks:  
-QxSSE4.2 -Qipo -O3 -Qprec-div- -Qopt-prefetch -Qparallel  
-Qpar-runtime-control -Qvec-guard-write /F512000000

C++ benchmarks:  
-QxSSE4.2 -Qipo -O3 -Qprec-div- -Qopt-prefetch -Qcxx-features  
/F512000000 shlw32m.lib -link /FORCE:MULTIPLE

#### Peak Optimization Flags

C benchmarks:

400.perlbench: -QxSSE4.2(pass 2) -Qprof\_gen(pass 1) -Qprof\_use(pass 2)  
-Qipo -O3 -Qprec-div- -Qansi-alias -Qopt-prefetch  
/F512000000 shlw32m.lib -link /FORCE:MULTIPLE

401.bzip2: -QxSSE4.2(pass 2) -Qprof\_gen(pass 1) -Qprof\_use(pass 2)  
-Qipo -O3 -Qprec-div- -Qopt-prefetch -Qansi-alias  
/F512000000

403.gcc: -QxSSE4.2(pass 2) -Qprof\_gen(pass 1) -Qprof\_use(pass 2)  
-Qipo -O3 -Qprec-div- /F512000000

429.mcf: -QxSSE4.2 -Qipo -O3 -Qprec-div- -Qopt-prefetch

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

### Formulación de los resultados de la evaluación del rendimiento

#### Informe SPEC:

- Descripción hardware y software
- Descripción parámetros compilación
- Publicación resultados básicos y optimizados

Results Table

Benchmark	Base						Peak					
	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
400.perlbench	426	22.9	426	22.9	<u>426</u>	<u>22.9</u>	305	32.0	305	32.1	<u>305</u>	<u>32.0</u>
401.bzip2	526	18.3	<u>526</u>	<u>18.3</u>	526	18.3	517	18.7	518	18.6	<u>517</u>	<u>18.7</u>
403.gcc	<u>305</u>	<u>26.4</u>	305	26.4	302	26.7	264	30.5	267	30.1	<u>264</u>	<u>30.5</u>
429.mcf	189	48.3	<u>189</u>	<u>48.3</u>	190	48.0	189	48.2	192	47.5	<u>190</u>	<u>48.0</u>
445.gobmk	444	23.6	444	23.6	<u>444</u>	<u>23.6</u>	<u>396</u>	<u>26.5</u>	395	26.5	396	26.5
456.hmmer	496	18.8	495	18.9	<u>495</u>	<u>18.9</u>	392	23.8	<u>392</u>	<u>23.8</u>	392	23.8
458.sjeng	494	24.5	494	24.5	<u>494</u>	<u>24.5</u>	472	25.6	472	25.6	<u>472</u>	<u>25.6</u>
462.libquantum	99.5	208	<u>99.6</u>	<u>208</u>	99.9	207	99.5	208	<u>99.6</u>	<u>208</u>	99.9	207
464.h264ref	599	37.0	599	36.9	<u>599</u>	<u>37.0</u>	538	41.1	<u>539</u>	<u>41.1</u>	539	41.1
471.omnetpp	255	24.5	<u>254</u>	<u>24.6</u>	254	24.6	218	28.7	<u>218</u>	<u>28.7</u>	217	28.7
473.astar	395	17.8	<u>395</u>	<u>17.8</u>	395	17.8	355	19.8	<u>356</u>	<u>19.7</u>	356	19.7
483.xalancbmk	232	29.8	231	29.9	<u>231</u>	<u>29.8</u>	232	29.8	231	29.9	<u>231</u>	<u>29.8</u>

Análisis de  
rendimiento

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

Análisis de  
rendimiento

### Formulación de los resultados de la evaluación del rendimiento

	Computador A	Computador B	Computador C
Programa 1	1	10	20
Programa 2	1000	100	20
Tiempo total	1001	110	40

#### ■ Afirmaciones:

- A es 900% más rápido que B para el programa 1.
- B es 900% más rápido que A para el programa 2.
- A es 1900% más rápido que C para el programa 1.
- C es 4900% más rápido que A para el programa 2.
- B es 100% más rápido que C para el programa 1.
- C es 400% más rápido que B para el programa 2.
- El contraste de las afirmaciones presenta un cuadro confuso.

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

### Resúmenes del rendimiento

#### ⚙️ Tiempo total de ejecución

	Computador A	Computador B	Computador C
Programa 1	1	10	20
Programa 2	1000	100	20
Tiempo total	1001	110	40

- ⚙️ B es 810% más rápido que A para los programa 1 y 2.
- ⚙️ C es 2400% más rápido que A para los programa 1 y 2.
- ⚙️ C es 175% más rápido que B para los programa 1 y 2.

#### ⚙️ Tiempo medio de ejecución

$$\frac{1}{n} \sum_{i=1}^n \text{Tiempo}_i$$

Análisis de  
rendimiento

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

Análisis de  
rendimiento

### Resúmenes del rendimiento

#### ⬢ Tiempo de ejecución ponderado

- ⬢ Asignar a cada programa un factor de peso  $w_i$  que indique la frecuencia relativa del programa en la carga de trabajo.

$$\sum_{i=1}^n w_i \cdot \text{Tiempo}_i$$

- ⬢  $w_i$  = frecuencia del programa iésimo de la carga de trabajo
- ⬢ **Tiempo<sub>i</sub>** = tiempo de ejecución del programa i-ésimo

#### ⬢ Media Geométrica

$$\sqrt[n]{\prod_{i=1}^n \text{Tiempo}_i}$$

$$\frac{MG(x_i)}{MG(y_i)} = MG\left(\frac{x_i}{y_i}\right)$$

## 2.2 Evaluación del rendimiento

Métricas

Benchmarks

Formulación

Análisis de  
rendimiento

### Intel Core i7-965

Benchmark	Base Ref Time	Base Run Time	Ratio
400.perlbench	9770	425.9	22.9396572
401.bzip2	9650	526.2	18.33903459
403.gcc	8050	305.2	26.37614679
429.mcf	9120	188.8	48.30508475
445.gobmk	10490	443.6	23.64743012
456.hmmer	9330	494.7	18.8599151
458.sjeng	12100	494.2	24.48401457
462.libquantum	20720	99.6	208.0321285
464.h264ref	22130	598.9	36.95107697
471.omnetpp	6250	253.9	24.61599055
473.astar	7020	395.1	17.76765376
483.xalancbmk	6900	231.3	29.83138781

<b>Media Aritmética</b>	10960.83333	371.45	41.67912673	29.5082335
	<b>MA(TR)</b>	<b>MA(TE)</b>	<b>MA(TR/TE)</b>	<b>MA(TR)/MA(TE)</b>

<b>Média Geométrica</b>	10108.21274	334.1591446	30.2496966	30.2496966
	<b>MG(TR)</b>	<b>MG(TE)</b>	<b>MG(TR/TE)</b>	<b>MG(TR)/MG(TE)</b>