

# **ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES**

## **Medidas de rendimiento en los computadores**

**Luis Rincón Córcoles  
Licesio J. Rodríguez-Aragón**



### **Programa**

Medidas de rendimiento en los computadores

- 1. Introducción.**
- 2. Definición de rendimiento.**
- 3. Medidas para evaluar el rendimiento.**
- 4. Programas para evaluar el rendimiento.**
- 5. Comparación y resumen del rendimiento.**
- 6. Ley de Amdahl.**
- 7. Otras medidas para evaluar el rendimiento.**
- 8. Comentarios finales.**



- ➔ D.A. PATTERSON, J.L HENNESSY. *Estructura y diseño de computadores*. Reverté, 2000.
- ➔ P. DE MIGUEL. *Fundamentos de los computadores*. 7ª edición, Paraninfo, 1999.
- ➔ W. STALLINGS. *Organización y Arquitectura de Computadores*. 5ª edición, Prentice-Hall, 2000.



## 1. Introducción

- ➔ Cuando vamos a adquirir un computador, es interesante que conozcamos el rendimiento que nos va a ofrecer.
- ➔ A la hora de diseñar un computador, es interesante contar con herramientas que nos permitan evaluar sus prestaciones con objeto de ponderar la relación coste / rendimiento del mismo.
- ➔ Vamos a estudiar algunas formas de caracterizar el rendimiento de un computador.



## 2. Definición de rendimiento

- ⇒ El concepto de rendimiento se puede percibir desde diferentes puntos de vista:
  - Tiempo de respuesta: velocidad (tiempo) de ejecución.
  - Productividad: Número de tareas completadas en la unidad de tiempo.

- ⇒ Consideraremos el rendimiento desde la perspectiva del tiempo de ejecución:

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

- ⇒ La máquina X es n veces más rápida que la máquina Y si:

$$\frac{\text{Rendimiento}_x}{\text{Rendimiento}_y} = n$$

- ⇒ El tiempo de ejecución y el rendimiento son recíprocos: aumentar el rendimiento implica disminuir el tiempo de ejecución.
- ⇒ Hablaremos de **mejorar el rendimiento**.

5



## 3. Medidas para evaluar el rendimiento

- ⇒ **Tiempo de reloj, tiempo de respuesta o tiempo transcurrido:**

- Tiempo desde que se lanza una tarea hasta que se completa.
- Incluye el tiempo de espera de entrada / salida, el tiempo consumido por otros procesos activos en el sistema, etc.

- ⇒ **Tiempo de ejecución de UCP o tiempo de ejecución:**

- Tiempo consumido por la UCP en ejecutar el programa.
- No incluye el tiempo de espera de entrada / salida, el tiempo consumido por otros procesos activos en el sistema, etc.
- Tiempo de ejecución de UCP = Tiempo de ejecución de UCP de usuario + tiempo de ejecución de UCP del sistema.

- ⇒ **Rendimiento de un sistema:**

- Tiempo transcurrido en un sistema sin carga.

- ⇒ **Rendimiento de UCP:**

- Recíproco del tiempo de ejecución.

- ⇒ **Tiempo de ejecución:** a veces se mide en ciclos de reloj.

- Frecuencia de reloj: inversa del ciclo de reloj.

6



## Medidas para evaluar el rendimiento

⇒ Tiempo de ejecución de UCP para un programa:

$$\text{Tiempo UCP} = \text{Ciclos UCP} \times \text{Tiempo de ciclo} = \frac{\text{Ciclos UCP}}{\text{Frecuencia de ciclo}}$$

⇒ CPI: ciclos de reloj por instrucción

⇒ CPI medio: media de ciclos de reloj por instrucción

$$\text{CPI}_{\text{medio}} = \text{CPI} = \frac{\sum_{i=1}^n (\text{CPI}_i \times C_i)}{\text{Número instrucciones}}$$

⇒ Entonces:

Ciclos UCP = Número de instrucciones × Media de ciclos por instrucción

$$\text{Ciclos UCP} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$$

7



## Medidas para evaluar el rendimiento

⇒ Se deduce que:

Tiempo UCP = Número instrucciones × CPI × Tiempo de ciclo ⇒

$$\text{Tiempo UCP} = \frac{\text{Número de instrucciones} \times \text{CPI}}{\text{Frecuencia de ciclo}}$$

⇒ Y por tanto:

$$\text{Tiempo UCP} = \sum_{i=1}^n (\text{CPI}_i \times C_i) \times \text{Tiempo de ciclo} \Rightarrow$$

$$\text{Tiempo UCP} = \frac{\sum_{i=1}^n (\text{CPI}_i \times C_i)}{\text{Frecuencia de ciclo}}$$

8



## 4. Programas para evaluar el rendimiento

⇒ Carga de trabajo de un sistema: conjunto de programas ejecutados en el mismo a lo largo del día.

- Para comprobar el rendimiento de un sistema, lo mejor es medir el tiempo de ejecución de la carga de trabajo (difícil).

⇒ Programas de prueba (*benchmarks*): programas pequeños específicamente escogidos para medir el rendimiento.

- Ventajas:
  - A menudo se pueden ensamblar a mano (útil si aún no hay compilador).
  - Son fáciles de estandarizar en diferentes máquinas.
- Desventajas: se prestan a fraudes.
  - Compiladores específicamente diseñados para optimizar un *benchmark*.
  - Mejoras específicas en la circuitería para optimizar fragmentos pequeños de código.

⇒ Pruebas sintéticas: programas artificiales contruidos para intentar englobar las características de un conjunto de programas.

- Son irreales y también se prestan a optimizaciones fraudulentas.

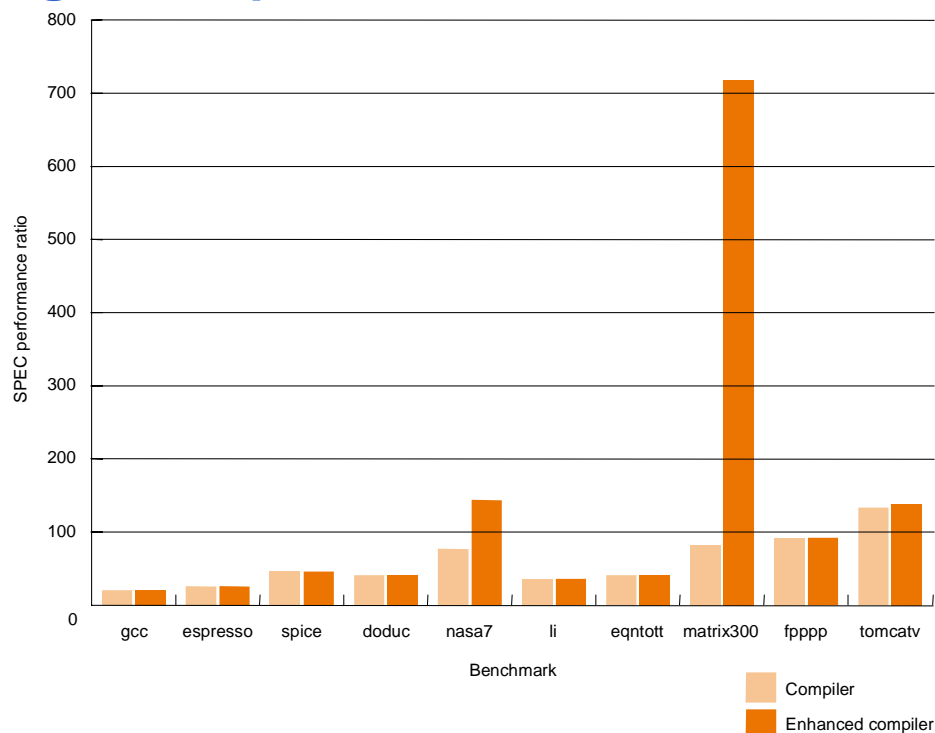
⇒ Lo mejor es realizar pruebas con programas reales (utilizados regularmente, o bien programas típicos), sobre todo si la máquina está ya funcionando.

- Reproducibilidad de las medidas: fundamental (documentar la prueba).

9



## Programas para evaluar el rendimiento



⇒ Pruebas de conjunto de *benchmarks* de prueba en una máquina con dos compiladores distintos.

10



## 5. Comparación y resumen del rendimiento

⇒ A menudo los vendedores y los compradores necesitan un único número para evaluar el rendimiento de una máquina.

- Este número resume el rendimiento del conjunto de programas de prueba seleccionado.
- Problema: ¿cómo calcular la medida resumen?

⇒ Comparación los rendimientos relativos de dos máquinas X e Y respecto de un programa:

$$\frac{\text{Rendimiento}_X}{\text{Rendimiento}_Y} = \frac{\text{tiempo de ejecución}_Y}{\text{tiempo de ejecución}_X} = p$$

⇒ Si la carga está formada por varios programas, el tiempo de ejecución puede ser la media aritmética del tiempo de ejecución de todos ellos:

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



## Comparación y resumen del rendimiento

⇒ Método sugerente para comparar los rendimientos de dos máquinas:

1. Normalizar los tiempos de ejecución para una máquina de referencia.
2. Tomar el promedio de los tiempos de ejecución normalizados.

⇒ Problema: el resultado depende de cuál sea la máquina de referencia.

⇒ Solución: usar la media geométrica en vez de la media aritmética.

$$MG = \sqrt[n]{\prod_{i=1}^n \text{Ratio del tiempo de ejecución}_i}$$

⇒ La media geométrica es independiente de la normalización:

$$\frac{MG(X_i)}{MG(Y_i)} = MG \frac{X_i}{Y_i}$$



## 6. Ley de Amdahl

⇒ Sirve para evaluar el impacto de una mejora parcial en una máquina en relación al rendimiento global de la misma.

$$\text{Tiempo final mejorado} = \frac{\text{Tiempo afectado por la mejora}}{\text{Cantidad de la mejora}} + \text{Tiempo no afectado}$$

⇒ Una mejora puede repercutir en un aumento del rendimiento que puede compensar el aumento de coste al incorporarla.

$$\text{Mejora en rendimiento} = \frac{\text{Rendimiento con mejora}}{\text{Rendimiento sin mejora}} = \frac{\text{Tiempo inicial sin mejora}}{\text{Tiempo final mejorado}}$$

⇒ Al mejorar un aspecto de la máquina, el impacto de dicha mejora va en función de la fracción de tiempo en que ésta pueda aplicarse, y está limitado por ella.

13



## 7. Otras medidas para evaluar el rendimiento

⇒ MIPS: millones de instrucciones ejecutadas por segundo.

$$\text{MIPS} = \frac{\text{Número de instrucciones}}{\text{Tiempo de ejecución} \times 10^6}$$

- No se tienen en cuenta las características de las instrucciones: no se pueden comparar máquinas con diferentes repertorios de instrucciones.
- Los MIPS varían entre diferentes programas de un mismo computador.
- Los MIPS pueden variar inversamente al rendimiento.

⇒ MIPS relativos: MIPS normalizados respecto de una máquina de referencia.

$$\text{MIPS relativos} = \frac{\text{Tiempo en máquina de referencia}}{\text{Tiempo en la máquina}} \times \text{MIPS referencia}$$

⇒ MIPS de pico: velocidad máxima de ejecución de instrucciones.

- Se obtienen con el CPI menor posible.

⇒ MFLOPS: millones de operaciones de coma flotante por segundo.

⇒ MOPS: millones de operaciones por segundo.

14



## 8. Comentarios finales

➡ Es importante diseñar mejorando el rendimiento sin perder de vista el coste que ello supone: equilibrio coste-rendimiento.

- Diseño de alto rendimiento: el coste es secundario.
- Diseño de bajo coste: el rendimiento es secundario.
- Diseño coste/rendimiento: busca el equilibrio.

➡ Método correcto de calcular el rendimiento: tiempo de ejecución de programas reales.

➡ Factores que influyen en el rendimiento:

- Número de instrucciones de los programas.
- Número de ciclos de reloj por instrucción.
- Frecuencia del reloj.

➡ Arte del diseño de computadores: determinar de forma precisa cómo las alternativas influyen en el coste y el rendimiento.