



Módulo 3: Diseño del procesador. Rendimiento

1



3.4 Rendimiento

1. Rendimiento en los procesadores
2. Medidas de rendimiento
3. Ley de Amdahl

■ Bibliografía

- Hennessy Patterson Apendice A, 4ª ed.

2

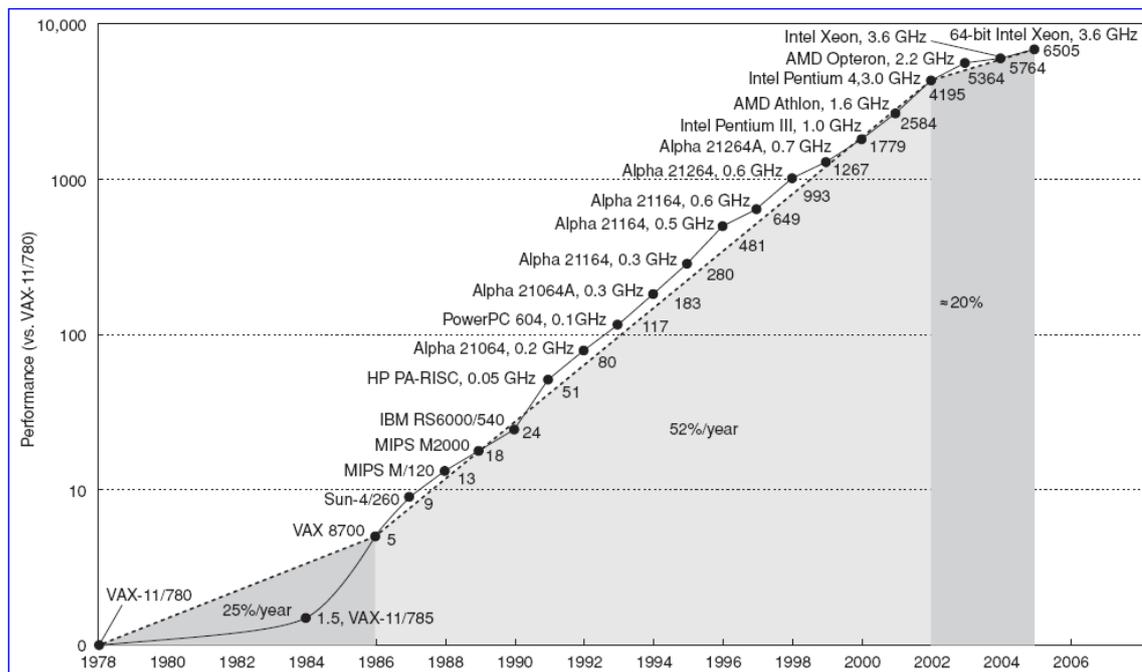
El papel del rendimiento



“Los buenos programadores se han preocupado siempre por el rendimiento de sus programas porque la rápida obtención de resultados es crucial para crear programas de éxito”

D. A. Patterson y J. L. Henessy

Crecimiento del rendimiento de los procesadores



Rendimiento de los procesadores



- **Cuántos ciclos tarda en ejecutarse este programa**
 - **Depende del procesador:** por ejemplo en el MIPS multiciclo

```
lw r1, 0(r0) → 5
lw r2, 4(r0) → 5
add r3, r1, r2 → 4
beq r3, r5, 1 → 4
sub r3, r3, r5 → 4
sw r3, 8(r0) → 4
```

- **Y cuánto Tiempo**
 - Depende de la frecuencia del procesador

Medidas del rendimiento



Para poder **comparar diferentes procesadores** hace falta establecer una medida del rendimiento que permita cuantificar los resultados de la comparación

- **Métrica:** establece la unidad de medida, que casi siempre es el tiempo, aunque hay que considerar dos aspectos diferentes del tiempo:
 - **Tiempo de ejecución:** tiempo que tarda en realizarse una tarea determinada
 - **Productividad** (throughput): tareas realizadas por unidad de tiempo

El interés por uno u otro aspecto dependerá del punto de vista de quien realiza la medida: un usuario querrá minimizar el tiempo de respuesta de su tarea, mientras que el responsable de un centro de datos le interesará aumentar el número de trabajos que realiza el centro por unidad de tiempo (productividad)

- **Patrón de medida:** establece los programas que se utilizan para realizar la medida (**benchmarks**). Existen muchos posibles benchmarks aunque los más utilizados son:
 - Nucleos de programas reales: SPEC
 - Programas sintéticos: TPC

Medidas del rendimiento: Tiempo de ejecución

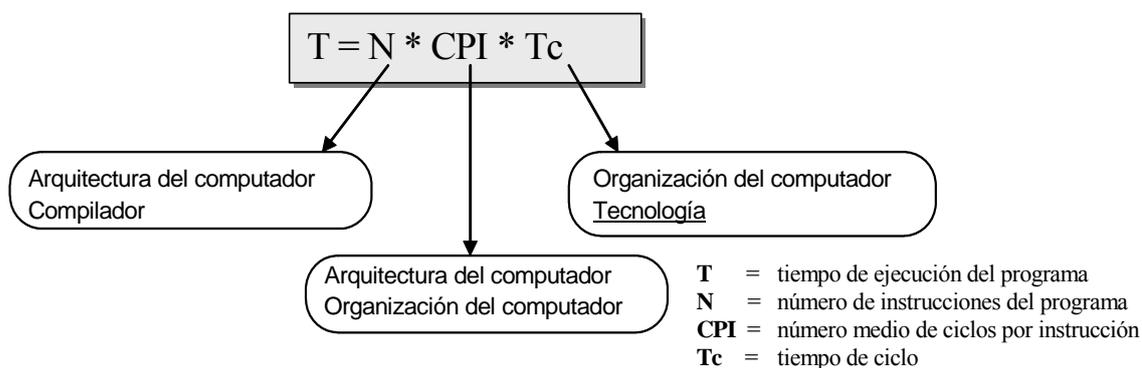


- Formado por :
 - **Tiempo de respuesta:** tiempo para completar una tarea (que percibe el usuario).
 - **Tiempo de CPU:** tiempo que tarda en ejecutarse un programa, sin contar el tiempo de E/S o el tiempo utilizado para ejecutar otros programas. Se divide en:
 - **Tiempo de CPU utilizado por el usuario:** tiempo que la CPU utiliza para ejecutar el programa del usuario sin tener en cuenta el tiempo de espera debido a la E/S
 - **Tiempo de CPU utilizado por el S.O.:** tiempo que el S.O. emplea para realizar su gestión interna.
 - La función **time** de Unix visualiza estas componentes: 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

Medidas del rendimiento: Tiempo de ejecución



Tiempo de ejecución de un programa



- **CPI = Ciclos medios por instrucción**
 - Una instrucción necesita varios ciclos de reloj para su ejecución
 - Además, diferentes instrucciones tardan diferentes cantidades de tiempo
 - **CPI** = Es una suma ponderada del número de ciclos que tarda por separado cada tipo de instrucción

Medidas del rendimiento: Tiempo de ejecución



Cálculo del CPI

- El número total de ciclos de reloj de la CPU se calcula como:

$$\text{Número de ciclos de la CPU} = \text{CPI} \cdot \text{NI} = \sum_{i=1}^n \text{CPI}_i \cdot \text{NI}_i$$

- NI_i = número de veces que el grupo de instrucciones i es ejecutado en un programa
 - CPI_i = número medio de ciclos para el conjunto de instrucciones i
- Podemos calcular el CPI multiplicando cada CPI_i individual por la fracción de ocurrencias de las instrucciones i en el programa.

$$\text{CPI} = \frac{\left(\sum_{i=1}^n \text{CPI}_i \cdot \text{NI}_i \right)}{\text{NI}} = \sum_{i=1}^n \left(\text{CPI}_i \cdot \frac{\text{NI}_i}{\text{NI}} \right)$$

- CPI_i debe ser medido, y no calculado a partir de la tabla del manual de referencia

Pérdida de rendimiento



- EL **CPI ideal es 1**
- Hay pérdidas de rendimiento por las paradas del pipe

$$\text{CPI}_{\text{real}} = \text{CPI}_{\text{ideal}} + \text{Penaliz. media por instrucción} = 1 + \sum_{i=1}^{\text{\#tipos de instr}} \text{Penaliz}_i \times \text{Frec}_i$$

- Caso de los saltos. Un programa típico 30% de saltos

$$\text{CPI} = 1 + (1 \times 0.3) = 1.3$$

$$\text{Speedup} = \frac{\text{Nº Instrucciones} \times \text{nº de etapas}}{\text{Nº instrucciones} \times \text{CPI}} = \frac{5}{1.3} = 3.84$$

- Eficiencia :

$$\text{Speedup}_{\text{real}} / \text{Speedup}_{\text{ideal}} = 3.84 / 5 = 0.76$$

- Se pierde un 24 % respecto al caso ideal

Medidas del rendimiento: MIPS



MIPS (Millones de Instrucciones Por Segundo)

$$MIPS = \frac{NI}{Tiempo_de_ejecución \cdot 10^6} = \frac{1}{CPI \cdot 10^6 \cdot T_c} = \frac{Fc}{CPI \cdot 10^6}$$

$$Tiempo\ de\ ejecución = \frac{NI}{MIPS \cdot 10^6}$$

- **Dependen del repertorio de instrucciones**, por lo que resulta un parámetro difícil de utilizar para comparar máquinas con diferente repertorio de instrucciones
- **Varían entre programas ejecutados en el mismo computador**
- **Pueden variar inversamente al rendimiento**, como ocurre en máquinas con hardware especial para punto flotante

$$MIPS_{relativos} = \frac{Tiempo\ de\ ejecución\ en\ la\ máquina\ de\ referencia}{Tiempo\ de\ ejecución\ en\ la\ máquina\ a\ medir} \cdot MIPS_{referencia}$$

$$MIPS_{referencia} = 1 \text{ (MIPS del VAX-11/780)}$$

Medidas del rendimiento: MFLOPS



MFLOPS (Millones de Operaciones en punto FLOTante Por Segundo)

$$MFLOPS = \frac{Número\ de\ operaciones\ en\ coma\ flo\ tan\ te\ de\ un\ programa}{Tiempo\ de\ ejecución \cdot 10^6}$$

- Existen operaciones en coma flotante rápidas (como la suma) o lentas (como la división), por lo que puede resultar una medida poco significativa.
- Se han utilizado los MFLOPS normalizados, que dan distinto peso a las diferentes operaciones.
- Por ejemplo: suma, resta, comparación y multiplicación se les da peso 1; división y raíz cuadrada peso 4; y exponenciación, trigonométricas, etc. peso 8:

Productividad (throughput)

- Número de tareas ejecutadas en la unidad de tiempo

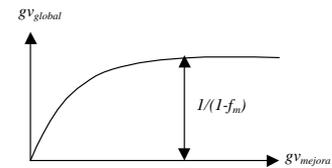
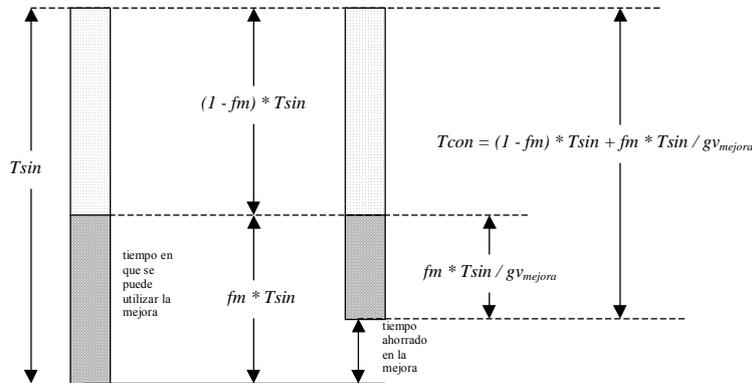
Medidas del rendimiento. Ley de Amdahl



Ganancia de velocidad (*speedup*): Ley de Amdahl

La mejora obtenida en el rendimiento global de un computador al utilizar un modo de ejecución más rápido está limitada por la fracción de tiempo que se puede utilizar dicho modo

•Definición de ganancia de velocidad: $gv_{global} = \frac{\text{Tiempo de ejecución sin mejora } (T_{sin})}{\text{Tiempo de ejecución con mejora } (T_{con})}$



$$\lim_{gv_{mejora} \rightarrow \infty} gv_{global} = \frac{1}{1-f_m}$$

$$gv_{mejora} \rightarrow \infty$$

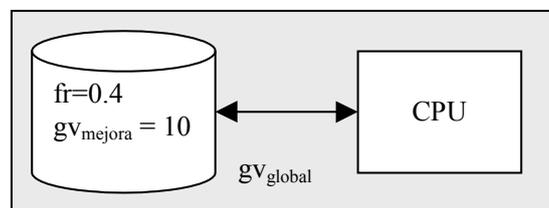
$$gv_{global} = \frac{T_{sin}}{T_{con}} = \frac{T_{sin}}{(1-f_m) * T_{sin} + \frac{f_m * T_{sin}}{gv_{mejora}}} = \frac{1}{(1-f_m) + \frac{f_m}{gv_{mejora}}}$$

Medidas del rendimiento. Ley de Amdahl



Ejemplo

- En un computador se sustituye el disco magnético por otro 10 veces más rápido.
- El disco se utiliza sólo el 40% del tiempo de ejecución.
- ¿Cuál es la ganancia de velocidad global?



$$gv_{mejora} = 10; f_m = 0.4 \implies gv_{global} = \frac{1}{(1-0.4) + \frac{0.4}{10}} = \frac{1}{0.64} = 1.56$$

$$\lim_{gv_{mejora} \rightarrow \infty} gv_{global} = \frac{1}{(1-0.4)} = 1.666 \text{ (máxima ganancia para velocidad del disco infinita)}$$