



Rendimiento

Resumen

Conceptos

- o **Rendimiento:** Un ordenador tiene más rendimiento que otro si puede ejecutar una tarea en menos tiempo.
 - o Rendimiento = inversa del tiempo de ejecución
 - o Cociente de rendimientos=inversa cociente de tiempos de ejecución
 - o Se suele expresar en porcentajes
- o **Productividad:** Un ordenador tiene más productividad que otro si puede ejecutar más tareas en el mismo tiempo.

Tiempos

- **Tiempo transcurrido:** Tiempo total necesario para completar una tarea, incluyendo accesos a discos, a memoria, E/S, etc...
- **Tiempo de CPU:** Tiempo que la CPU emplea en una determinada tarea. No incluye el tiempo de E/S ni de atención a otros programas.
- **Tiempo de CPU de usuario:** Tiempo que la CPU pasa en el programa del usuario.
- **Tiempo de CPU de sistema:** Tiempo que la CPU pasa ejecutando tareas de sistema operativo en beneficio del programa del usuario.
- **Rendimiento del sistema:** En este caso utilizamos como base el tiempo transcurrido en un sistema sin cargas (monotarea).
- **Rendimiento de CPU:** En este caso se utiliza como base el tiempo de CPU.

Rendimiento CPU

- o Tiempo de ejecución de CPU en función de
 - o Ciclos de reloj
 - o Periodo de reloj o frecuencia de reloj
- o Considerando instrucciones de programa queda en función de
 - o Número de instrucciones
 - o Ciclos por instrucción
 - o Periodo de reloj

¿Cómo afecta cada componente al tiempo de ejecución?

Componente	Parámetro	
Algoritmo	Número de instrucciones, CPI	Dependiendo del algoritmo necesitaremos más o menos instrucciones. El algoritmo puede favorecer instrucciones más rápidas o no (afecta al CPI)
Lenguaje	Número de instrucciones, CPI	El lenguaje implementa el algoritmo. Idem del anterior.
Compilador	Número de instrucciones, CPI	El compilador efectúa la traducción a lenguaje máquina. Idem de los anteriores
Arquitectura del juego de instrucciones	Número de instrucciones, CPI, frecuencia de reloj	El juego de instrucciones afecta a los tres parámetros

Ley de Amdhal

- o Acelerar el caso común: Se debe favorecer el caso frecuente sobre el infrecuente.
 - o Overflow es infrecuente, por tanto la suma sin overflow (add unsigned) debe ser más rápida que con overflow (add).
- o “La mejora obtenida en el rendimiento al utilizar algún modo de ejecución más rápido está limitada por la fracción de tiempo en que se pueda utilizar ese modo más rápido

Aceleración del rendimiento

- o La aceleración depende de dos factores:
 - o **Fracción mejorada:** La fracción de tiempo durante la cual puede utilizarse la mejora.
 - o **Aceleración mejorada:** Cuanto más rápido se ejecuta la tarea si sólo se utiliza el modo mejorado.

Aceleración del rendimiento

- Tiempo de ejecución sin mejora = Tiempo de ejecución antiguo – Tiempo de ejecución con mejora.

$$Tejec_{sinmejora}$$

$$= Tejec_{antiguo} - Tejec_{conmejora}$$

$$Tejec_{sinmejora} = Tejec_{antiguo} * (1 - Fm)$$

Aceleración del rendimiento

- Tiempo de ejecución con mejora=Tiempo de ejecución antiguo*Fracción de tiempo durante la cual se puede utilizar la mejora

○

$$\begin{aligned} &Tejec_{conmejora} \\ &= Tejec_{antiguo} * \frac{Tejec_{conmejora}}{Tejec_{antiguo}} \end{aligned}$$

Aceleración del rendimiento

- Tiempo de ejecución nuevo = Tiempo antiguo durante la fracción que no se utiliza la mejora + tiempo nuevo durante el cual se utiliza la mejora
- El tiempo nuevo durante el cual se utiliza la mejora es $1/AM$ multiplicado por el tiempo antiguo.
 - Si se ejecuta el doble de rápido tarda $\frac{1}{2}$ del tiempo antiguo

Aceleración del rendimiento

$$o \ AG = \frac{T_{antiguo}}{T_{nuevo}} = \frac{1}{\left((1-Fm) + \left(\frac{Fm}{Am}\right)\right)}$$

Resumen de Rendimiento (1)

- Definición:

- $$\text{Rendimiento}_X = \frac{1}{\text{Tiempo_de_ejecución}_X}$$

- Condición:

- Si

- $$\text{Rendimiento}_X > \text{Rendimiento}_Y$$

- Entonces

- $$\text{Tiempo_de_ejecución}_X < \text{Tiempo_de_ejecución}_Y$$

Resumen de Rendimiento (2)

o Si X es más rápida que Y

$$o \frac{\text{Rendimiento}_X}{\text{Rendimiento}_Y} = n = 1 + \frac{(m)}{100}$$

$$o \frac{\text{Tiempo_de_ejecución}_Y}{\text{Tiempo_de_ejecucion}_X} = n = 1 + \frac{(m)}{100}$$

$$o N=1.5 \rightarrow m=50$$

o El ordenador x es 1.5 veces más rápido

o El ordenador X es un 50% más rápido

Resumen de Rendimiento (3)

$$\circ \text{Tiempo}_{CPU_{programa}} = \text{NumCiclosReloj}_{CPU_{programa}} \times \text{PeriodoReloj}$$

$$\circ \text{Tiempo}_{CPU_{programa}} = \frac{\text{NumCiclosReloj}_{CPU_{programa}}}{\text{FrecuenciaReloj}}$$

○ Podemos disminuir el tiempo

- Reduciendo el periodo de reloj == aumentar la frecuencia de reloj
- Reducir el número de ciclos de reloj

Resumen de Rendimiento (4)

$$\text{NumCiclosRelej}_{CPU_{programa}} = \text{NumInstrucciones}_{programa} \times \text{PromedioCiclosRelej}_{instruccion}$$

$$\text{Tiempo}_{CPU_{programa}} = \text{NumCiclosRelej}_{CPU_{programa}} \times \text{PeriodoRelej}$$

$$\text{Tiempo}_{CPU_{programa}} = \text{NumInstrucciones}_{programa} \times \text{CPI}_{instruccion} \times \text{PeriodoRelej}$$

$$\frac{\text{instrucciones}}{\text{programa}} \times \frac{\text{ciclos}}{\text{instruccion}} \times \frac{\text{Segundos}}{\text{ciclo}} = \frac{\text{Segundo}}{\text{programa}}$$

Podemos disminuir el tiempo

- Reduciendo el periodo de reloj == aumentar la frecuencia de reloj
- Reducir el número de ciclos de reloj que tarda cada instrucción
- Reducir el número de instrucciones de cada programa