

DESEMPEÑO DE LOS PROCESADORES

ARQUITECTURA DEL PROCESADOR II

1. DESEMPEÑO

Evaluar el desempeño de las computadoras es bastante desafiante.¹ Debido a lo intrincado y el tamaño de los sistemas de software modernos, sumado al amplio rango de técnicas empleadas por los diseñadores de hardware para mejorar el desempeño, se ha hecho mucho más difícil evaluar dicho desempeño. Cuando se trata de elegir entre diferentes computadoras, el desempeño es un atributo importante. Medir exactamente y comparar distintas computadoras es algo crítico para los compradores y por lo tanto para los diseñadores. Los vendedores de computadoras saben muy bien esto. Muchas veces los vendedores quieren que los clientes vean sus equipos bajo una determinada perspectiva, la que resulte más conveniente para ellos, pero no siempre esa perspectiva es la mejor para el comprador. Por eso es importante entender la mejor manera de medir el desempeño y cuáles son las limitaciones de esa medida al momento de elegir una computadora.

En la primer parte del apunte se describen diferentes formas para determinar el desempeño, luego se describen métricas para medir el desempeño desde el punto de vista del usuario y del diseñador del procesador. También se verá cómo estas métricas están relacionadas entre sí, y se presentará una importante ecuación de desempeño del procesador, la cual se usará a lo largo de la materia.

1.1. Definiendo desempeño. Cuando uno dice que una computadora tiene un mejor desempeño que otra. ¿Qué significa eso?. Aunque la pregunta puede parecer simple, una comparación con los aviones de pasajeros, muestra lo sutil que puede ser cuando se aplica al desempeño de las computadoras. La Tabla 1 muestra un listado de los aviones comerciales junto con algunos de sus datos: la capacidad, la autonomía y la velocidad crucero.

Avión	Pasajeros Transportados	Autonomía (Millas)	Velocidad (m.p.h)	Rendimiento en pasajeros (pasajeros X m.p.h.)
Boeing 777	375	4630	610	228.750
Boeing 747	470	4150	610	286.700
BAC/Sud Concorde	132	4000	1350	178.200
Douglas DC-8-50	146	8720	544	79.424

TABLA 1. Parámetros de los aviones comerciales

¹Este material de estudio es la traducción de la sección 1.6 del libro de Patterson y Hennessy [1]

Si se desea determinar que avión tiene el mejor desempeño, primero es necesario definir desempeño. Supongamos que se define desempeño en términos de la velocidad. Aún en este caso hay dos definiciones posibles. Se puede establecer que el avión más rápido es el que tiene la mayor velocidad crucero, ya que puede llevar un pasajero desde un punto a otro en el menor tiempo posible. Sin embargo, si se deben transportar 450 pasajeros desde un punto a otro, el 747 claramente será el más rápido (como se muestra en la última columna de la figura). De la misma manera es posible definir el desempeño de las computadoras según muchos criterios.

Si alguien debe correr un programa sobre dos computadoras de escritorio distintas, dirá que la computadora más rápida es la que termina primero el trabajo. Si se emplea un centro de cómputos, donde hay varios servidores corriendo los trabajos de varios usuarios, se dice que la computadora más veloz es la que finaliza la mayor cantidad de trabajos durante un día. El usuario que no comparte su computadora está interesado en reducir el *tiempo de respuesta*, el tiempo entre el comienzo y la finalización de una tarea, también conocido como el *tiempo de ejecución*. El administrador del centro de cómputos frecuentemente está interesado en incrementar el *rendimiento* (o en inglés *throughput*), que se define como la cantidad de trabajo realizado en un determinado intervalo de tiempo. Por eso, en muchos casos, necesitaremos diferentes métricas para medir el desempeño, también diferentes conjuntos de aplicaciones para probar computadoras personales (o dispositivos móviles), los cuales dan relevancia al tiempo de respuesta, a diferencia de los servidores que tienen más en cuenta el rendimiento (*throughput*).

Ejemplo. Rendimiento y tiempo de respuesta. ¿Los siguientes cambios a un sistema de computación, incrementan el rendimiento, disminuyen el tiempo de respuesta o ambas cosas?

1. Reemplazar el procesador de una computadora por uno más rápido.
2. Agregar procesadores adicionales en un sistema que usa múltiples procesadores para distintas tareas - por ejemplo buscar en la red.

Respuesta. Decrementar el tiempo de respuesta siempre mejora el rendimiento. Por eso, en el caso 1, tanto el tiempo de respuesta como el rendimiento son mejorados. En el caso 2, ninguna tarea consigue hacer el trabajo más rápido, sólo se incrementa el rendimiento.

Sin embargo, si la demanda de procesamiento en el segundo caso es casi tan grande como el rendimiento, el sistema puede forzar que los requerimientos queden esperando en una cola. En este caso el incremento del rendimiento podría también mejorar el tiempo de respuesta, porque se podría reducir el tiempo de espera en la cola. Por eso en varios sistemas de computadoras reales, ya sea cambiar el tiempo de ejecución o el rendimiento, frecuentemente cada uno afecta al otro. $\square$

En la primer parte de este documento la discusión del desempeño estará principalmente orientada al tiempo de respuesta. Para maximizar el desempeño se debe minimizar el tiempo de respuesta (o tiempo de ejecución) para una tarea. De esa manera es posible relacionar el desempeño y el tiempo de ejecución para una computadora X :

$$Desempeño_X = \frac{1}{Tiempo\ de\ ejecución_X}$$

Esto significa que para dos computadoras X e Y , si el desempeño de X es mejor que el desempeño de Y , vale que:

$$Desempeño_X > Desempeño_Y$$

$$\frac{1}{Tiempo\ de\ ejecución_X} > \frac{1}{Tiempo\ de\ ejecución_Y}$$

$$Tiempo\ de\ ejecución_Y > Tiempo\ de\ ejecución_X$$

Es decir que: el tiempo de ejecución de Y es mayor que el tiempo de ejecución de X , si X es más rápida que Y . Al discutir sobre el diseño de computadoras, la relación del desempeño de dos computadoras diferentes se hace de manera cuantitativa. Se usa la frase “ X es n veces más rápida que Y ” o lo que es equivalente “ X es n veces tan rápida como Y ”, para establecer que:

$$\frac{Desempeño_X}{Desempeño_Y} = n$$

Si X es n veces más rápida que Y , entonces el tiempo de ejecución en Y es n veces mayor que la ejecución en X :

$$\frac{Desempeño_X}{Desempeño_Y} = \frac{Tiempo\ de\ ejecución_Y}{Tiempo\ de\ ejecución_X} = n$$

Ejemplo. Desempeño relativo. Si una computadora A corre un programa en 10 segundos y una computadora B corre el mismo programa en 15 segundos, ¿Cuánto más rápida es A que B?

Respuesta. Se sabe que A es n veces más rápida que B si:

$$\frac{Desempeño_A}{Desempeño_B} = \frac{Tiempo\ de\ ejecución_B}{Tiempo\ de\ ejecución_A} = n$$

Así la relación de desempeño es:

$$\frac{15}{10} = 1,5$$

y A, es por lo tanto, 1,5 veces más rápida que B. $\square$

En el ejemplo de arriba, se puede decir que la computadora B es 1,5 veces más lenta que la computadora A debido a que:

$$\frac{Desempeño_A}{Desempeño_B} = 1,5$$

quiere decir que

$$\frac{Desempeño_A}{1,5} = Desempeño_B$$

Por simplicidad se usa la terminología *tan rápida como* cuando se está tratando de comparar computadoras de manera cuantitativa. Debido a que el desempeño y el tiempo de ejecución son recíprocos, incrementar el desempeño requiere decrementar el tiempo de ejecución. Para evitar la confusión potencial entre los términos *incrementar* y *decrementar*, habitualmente

se dice “mejorar el desempeño” o “mejorar el tiempo de ejecución” cuando se quiere decir “incrementar el desempeño” o “decrementar el tiempo de ejecución”.

1.2. Midiendo el desempeño. El tiempo es la medida de desempeño de las computadoras: la computadora más rápida es la que realiza la misma cantidad de trabajo en menos tiempo. El *tiempo de ejecución* de un programa se mide en segundos por programa. Sin embargo, el tiempo puede ser definido de diferentes maneras, dependiendo de lo que se esté contando. La definición más clara del tiempo es el llamado *tiempo del reloj de pared*, *tiempo de respuesta*, o *tiempo transcurrido*. Estos términos se refieren al tiempo total para completar una tarea, incluyendo los accesos a disco, los accesos a memoria, las actividades de entrada/salida, las demoras del sistema operativo – todo eso.

Las computadoras con frecuencia son compartidas, es decir, que un procesador debe trabajar sobre varios programas simultáneamente. En esos casos, el sistema puede tratar de mejorar el rendimiento para un programa, más que optimizar el tiempo de respuesta. Por eso se marca la diferencia entre el tiempo transcurrido y el tiempo durante el cual el procesador se dedica a una tarea en particular. El tiempo de ejecución de CPU o simplemente tiempo de CPU, que reconoce esa diferencia, es el tiempo que la CPU destina a esa tarea y no incluye el tiempo destinado a e/s o ejecutar otros programas. (Recordar que el tiempo de respuesta que experimenta el usuario es el tiempo transcurrido, no el tiempo de CPU). El tiempo de CPU puede ser a su vez dividido en el tiempo que la CPU destina a ejecutar el programa, llamado *tiempo de CPU del usuario* y el tiempo que la CPU gasta haciendo las tareas que el Sistema Operativo destina al programa, llamado *tiempo de CPU del sistema*. Resulta difícil establecer la diferencia entre el tiempo de CPU del usuario y del sistema, porque en general es difícil asignar la responsabilidad de las tareas del Sistema Operativo a una tarea y no a otra, también por la diferencia de funcionamiento entre los distintos Sistemas Operativos.

Por consistencia se mantiene la diferencia entre desempeño basado en el tiempo transcurrido y aquel que se basa en el tiempo de ejecución de CPU. En este documento se usará el término *desempeño del sistema* para hacer referencia al tiempo transcurrido en un sistema sin carga y *desempeño de CPU* para referirse al tiempo de CPU del usuario. El interés de la materia está principalmente orientado al tiempo de CPU del usuario, aun cuando las consideraciones sobre desempeño son generalmente aplicables a las dos mediciones: tiempo transcurrido y tiempo de CPU.

Si bien los usuarios de computadoras están interesados en el tiempo, cuando se examina en detalle una computadora, es conveniente pensar acerca del desempeño de la misma empleando otras métricas. En particular, los diseñadores de computadoras, prefieren una medida que se relaciona a cuán rápido el hardware hace las funciones básicas. Casi todas las computadoras son creadas usando un reloj que determina cuando ocurren los eventos en el hardware. Estos intervalos discretos de tiempo son llamados *ciclos de reloj* (o *ticks*, o *ticks de reloj*, o *período de reloj*, o *reloj*, o *ciclos*). Los diseñadores se refieren al *período del reloj* o bien como el tiempo para un ciclo de reloj completo (por ejemplo 250 picosegundos o 250 ps) o bien como la frecuencia del reloj (por ejemplo 4GigaHerz o 4GHz), la cual es la inversa del período del reloj. Más adelante se formalizará la relación entre los ciclos de

reloj usados por el diseñador del hardware y los segundos del usuario de las computadoras.

No hay que caer en la trampa. Existe la expectativa de que mejorando un único aspecto de una computadora, se incrementa el desempeño total en una cantidad proporcional al tamaño de la mejora. La gran idea de **hacer más rápido el caso común** tiene un corolario desmoralizante que ha atormentado a los diseñadores de software y de hardware. Este corolario nos recuerda que la posibilidad de mejorar el desempeño es afectada por la cantidad total de tiempo que consuma el evento mejorado.

Un problema de diseño simple lo ilustra bien. Supongamos que un programa corre en 100 segundos sobre una computadora, donde las operaciones de multiplicación son responsables de 80 de esos segundos. ¿Cuánto hay que mejorar la multiplicación si se desea que el programa corra cinco veces más rápido?

El tiempo de ejecución del programa luego de hacer la mejora está dado por la siguiente ecuación simple, conocida como la ley de Amdahl:²

$$\begin{aligned} \text{Tiempo de ejecución después de la mejora} &= \\ &= \frac{\text{Tiempo de ejecución afectado por la mejora}}{\text{Cantidad de mejora}} + \text{Tiempo de ejecución no afectado} \end{aligned}$$

Para este problema:

$$\text{Tiempo de ejecución después de la mejora} = \frac{80 \text{ seg}}{n} + (100 - 80 \text{ seg})$$

Debido a que se desea que el desempeño sea cinco veces mejor, el nuevo tiempo de ejecución debiera ser 20 segundos, lo que sería:

$$\begin{aligned} 20 \text{ seg} &= \frac{80 \text{ seg}}{n} + (20 \text{ seg}) \\ 0 &= \frac{80 \text{ seg}}{n} \end{aligned}$$

Si la multiplicación consume el 80% del tiempo total, no existe una cantidad por la cual mejorar la multiplicación para lograr incrementar cinco veces el desempeño. El incremento posible del desempeño con una mejora está limitado por la cantidad que la característica mejorada se pueda usar. La ley de Amdahl se puede usar para estimar la mejora en el desempeño cuando se conoce el tiempo que consume alguna función y su aceleración potencial. La ley de Amdahl y la ecuación de desempeño de la CPU son herramientas prácticas para evaluar potenciales mejoras en el desempeño.

²La ley de Amdahl es una regla que establece cual es el posible incremento del desempeño para una mejora determinada, el límite es establecido por las oportunidades de aplicar la mejora.

1.3. El desempeño de la CPU y sus factores. Los usuarios y los diseñadores con frecuencia analizan el desempeño usando métricas distintas. Si fuese posible relacionar estas métricas distintas, sería posible determinar el efecto que tiene un cambio en el diseño sobre el desempeño percibido por el usuario. Debido a que nuestro interés es el desempeño de CPU, la medida de desempeño tomada como base es el tiempo de ejecución de CPU. Una fórmula simple relaciona las dos métricas básicas (ciclos de reloj y tiempo de ciclo de reloj) con el tiempo de CPU:

$$\begin{array}{c} \text{Tiempo de ejecución de CPU} \\ \text{para un programa} \end{array} = \begin{array}{c} \text{Ciclos de CPU para} \\ \text{un programa} \end{array} \times \text{Tiempo de ciclo de reloj}$$

Alternativamente, como la frecuencia de reloj y el tiempo de ciclo de reloj son inversas:

$$\begin{array}{c} \text{Tiempo de ejecución de CPU} \\ \text{para un programa} \end{array} = \frac{\text{Ciclos de CPU para un programa}}{\text{Frecuencia de reloj}}$$

Esta fórmula deja en claro que el diseñador puede mejorar el desempeño, reduciendo el número de ciclos de reloj requeridos para un programa o la duración del ciclo de reloj. Como se verá más adelante, los diseñadores deben hacer un balance entre el número de ciclos de reloj requeridos para un programa y la duración de cada ciclo. Muchas técnicas que decrementan el número de ciclos también incrementan el tiempo de ciclo de reloj.

1.4. Tiempo y Frecuencia. Frecuencia, es una medida para indicar el número de repeticiones de cualquier fenómeno o suceso periódico en una unidad de tiempo. Para calcular la frecuencia de un evento, se contabilizan un número de ocurrencias de éste teniendo en cuenta un intervalo temporal, luego estas repeticiones se dividen por el tiempo transcurrido. Según el Sistema Internacional, la frecuencia se mide en hercios (Hz), en honor a Heinrich Rudolf Hertz. Un hercio es aquel suceso o fenómeno repetido una vez por segundo, 2 Hz son dos sucesos (periodos) por segundo y así sucesivamente. Esta unidad se llamó originariamente como *ciclos por segundo* (cps) y aún también se sigue utilizando. Otras unidades para indicar la frecuencia son *revoluciones por minuto* (rpm) y *radianes por segundo* (rad/s). Las pulsaciones del corazón o el *tempo* musical se mide como golpes por minuto (bpm, del inglés beats per minute).

$$1\text{Hz} = \frac{1}{\text{Seg}}$$

Un método alternativo para calcular la frecuencia es medir el tiempo entre dos repeticiones (periodo) y luego calcular la frecuencia (f) recíproca de esta manera:

$$f = \frac{1}{T}$$

donde T es el periodo de la señal.

El *segundo* fue definido en 1967 por la Comisión Internacional de Pesos y Medidas como el tiempo que necesita un electrón para girar sobre su propio eje dentro de un átomo de cesio. El llamado reloj atómico puede medir la longitud de un segundo con una exactitud de 13 cifras decimales. Más exacto aún es el movimiento del electrón en una molécula de hidrógeno que consigue una exactitud de 15 cifras decimales. Por debajo del segundo existen otras unidades: milisegundo (milésima parte de un segundo, 10^{-3} segundos), microsegundo (millonésima de segundo, 10^{-6} segundos), nanosegundo (milmillonésima de segundo,

10^{-9} segundos), piconsegundo (billonésima de segundo, 10^{-12} segundos) y femtosegundo (milbillonésima de segundo, 10^{-15} segundos).

Una señal que completa un ciclo por segundo tiene un Hz de frecuencia.

Unidad	Símbolo	Valor en seg.	Frecuencia	Símb.Frec.
milisegundos	ms	1×10^{-3}	kiloHertz	kHz
microsegundos	Ms	1×10^{-6}	megaHertz	MHz
nanosegundo	ns	1×10^{-9}	gigaHertz	GHz
piconsegundo	ps	1×10^{-12}	tetraHertz	THz

Un clock es simplemente una señal continua con un tiempo de ciclo fijo, la frecuencia de reloj (Fr) es sencillamente la inversa del *tiempo de ciclo de reloj* (T_c):

$$Fr = \frac{1}{T_c}$$

El tiempo de ciclo de reloj está dividido en dos partes: cuando la señal es alta y cuando la señal es baja.

Ejemplo. Mejorando el desempeño. Nuestro programa favorito corre en 10 segundos en la computadora A, la cual tiene un reloj de 2 GHz. Tratamos de ayudar a un diseñador de computadoras para construir una computadora B, la cual correrá nuestro programa en 6 segundos. El diseñador ha determinado que un incremento sustancial en la frecuencia del reloj es posible, pero este incremento afectará el resto del diseño de la CPU, causando que la computadora B requiera para este programa 1,2 veces el número de ciclos de instrucción que la computadora A. ¿Qué frecuencia de reloj le decimos al diseñador que debe alcanzar?

Respuesta. Primero debemos encontrar el número de ciclos requeridos por el programa en la computadora A:

$$\text{Tiempo de CPU}_A = \frac{\text{Ciclos de reloj}_A}{\text{Frecuencia de reloj}_A}$$

$$10 \text{ seg} = \frac{\text{Ciclos de reloj}_A}{2 \times 10^9 \frac{\text{ciclos}}{\text{segundos}}}$$

$$\text{Ciclos de reloj}_A = 10 \text{ seg} \times 2 \times 10^9 \frac{\text{ciclos}}{\text{segundos}} = 20 \times 10^9 \text{ ciclos}$$

El tiempo de CPU para B puede ser encontrado usando la siguiente ecuación:

$$\text{Tiempo de CPU}_B = \frac{1,2 \times \text{Ciclos de reloj}_A}{\text{Frecuencia de reloj}_B}$$

$$6 \text{ seg} = \frac{1,2 \times 20 \times 10^9 \text{ ciclos}}{\text{Frecuencia de reloj}_B}$$

$$\begin{aligned} \text{Frecuencia de reloj}_B &= \frac{1,2 \times 20 \times 10^9 \text{ ciclos}}{6 \text{ seg}} = \frac{0,2 \times 20 \times 10^9 \text{ ciclos}}{\text{segundos}} = \\ &= \frac{4 \times 10^9 \text{ ciclos}}{\text{segundos}} = 4 \text{ GHz} \end{aligned}$$

Para correr el programa en 6 segundos, la computadora B debe tener el doble de la frecuencia de A. $\square$

1.5. Desempeño de instrucción. Las ecuaciones de desempeño de arriba no incluyen ninguna referencia al número de instrucciones necesarias para el programa. Sin embargo, debido a que el compilador genera instrucciones para ejecutar y que la computadora debe ejecutar las instrucciones para correr el programa, el tiempo de ejecución debe depender del número de instrucciones del programa. Una forma de pensar acerca del tiempo de ejecución es que es igual al número de instrucciones ejecutadas multiplicadas por el tiempo promedio por instrucción. Por lo tanto, el número de ciclos de reloj requeridos para un programa se puede escribir como:

$$\text{Ciclos de reloj de CPU} = \text{Instrucciones del programa} \times \frac{\text{Promedio de ciclos de reloj}}{\text{por instrucción}}$$

El término *ciclos de reloj por instrucción*, que es el número promedio de ciclos de reloj que cada instrucción toma para ejecutar, es frecuentemente abreviado como **CPI**. Debido a que diferentes instrucciones pueden tomar diferente cantidad de tiempo, dependiendo de lo que hagan, el CPI provee una forma de comparar dos implementaciones distintas de la misma arquitectura del conjunto de instrucciones, debido a que el número de instrucciones ejecutadas por el programa será la misma.

Ejemplo. Usando la ecuación del desempeño. Supongamos que existen dos implementaciones de la misma arquitectura del conjunto de instrucciones. La computadora A tiene un tiempo de ciclos de reloj de 250ps y un CPI de 2 para un determinado programa, y una computadora B que tiene un tiempo de ciclo de reloj 500 ps y un CPI de 1,2 para el mismo programa. ¿Qué computadora es más rápida para este programa y por cuanto?

Respuesta Se sabe que cada computadora ejecuta el mismo número de instrucciones para el programa; digamos que este número es I . Primero se debe encontrar la cantidad de ciclos de reloj del procesador para cada una de las computadoras:

$$\text{Ciclos de reloj de CPU}_A = I \times 2$$

$$\text{Ciclos de reloj de CPU}_B = I \times 1,2$$

Ahora es posible calcular el tiempo de CPU para cada computadora:

$$\begin{aligned} \text{Tiempo de CPU}_A &= \text{Ciclos de reloj de CPU}_A \times \text{Tiempo de ciclo de reloj}_A \\ &= I \times 2 \times 250 \text{ ps} = I \times 500 \text{ ps} \end{aligned}$$

De la misma manera para la computadora B:

$$= I \times 1,2 \times 500 \text{ ps} = I \times 600 \text{ ps}$$

Claramente la computadora A es la más rápida. ¿Cuánto más rápida?, eso está dado por la relación:

$$\frac{\text{Desempeño}_A}{\text{Desempeño}_B} = \frac{\text{Tiempo de ejecución}_B}{\text{Tiempo de ejecución}_A} = \frac{600 \times I \text{ ps}}{500 \times I \text{ ps}} = 1,2$$

Se puede concluir que la computadora A es 1,2 veces más rápida que la computadora B para este programa. $\square$

1.6. Ecuación clásica del desempeño de CPU. Es posible ahora escribir la ecuación clásica del desempeño de CPU en términos de la *cantidad de instrucciones* (el número de instrucciones ejecutadas por el programa), el CPI, y el tiempo de ciclo de reloj:

$$\text{Tiempo de CPU} = \text{Cantidad de instrucciones} \times \text{CPI} \times \text{Tiempo de ciclo de reloj}$$

o, dado que la frecuencia es la inversa del tiempo de CPU:

$$\text{Tiempo de CPU} = \frac{\text{Cantidad de instrucciones} \times \text{CPI}}{\text{Frecuencia de reloj}}$$

Estas fórmulas son particularmente útiles porque separan los tres factores claves que afectan el desempeño. Es posible usar estas dos fórmulas para comparar dos implementaciones diferentes o para evaluar un diseño alternativo si conocemos su impacto sobre estos tres parámetros.

Ejemplo. Comparando segmentos de código. Un diseñador de compiladores está tratando de decidir entre dos secuencias de código para una computadora particular. Los diseñadores de hardware han provisto la siguiente información:

CPI para cada clase de instrucción			
	A	B	C
CPI	1	2	3

Para una sentencia en lenguaje de alto nivel particular, quien escriba el compilador debe considerar dos secuencias de código que tienen el siguiente conteo de instrucciones:

Conteo de instrucciones para cada clase de instrucción			
Secuencias de código	A	B	C
1	2	1	2
2	4	1	1

¿Qué secuencia de código ejecuta más instrucciones? ¿Cuál será más rápida? ¿Cuál es el CPI para cada secuencia?

Respuesta. La secuencia 1 ejecuta $2 + 1 + 2 = 5$ instrucciones. La secuencia 2 ejecuta $4 + 1 + 1 = 6$ instrucciones. Se puede usar la ecuación de ciclos de reloj de CPU, basada en la cantidad de instrucciones y el CPI, para encontrar el número total de ciclos de reloj para cada secuencia:

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

Esto produce

$$\text{Ciclos de reloj de CPU}_1 = (2 \times 1) + (1 \times 2) + (2 \times 3) = 2 + 2 + 6 = 10 \text{ ciclos}$$

$$\text{Ciclos de reloj de CPU}_2 = (4 \times 1) + (1 \times 2) + (1 \times 3) = 4 + 2 + 3 = 9 \text{ ciclos}$$

Se verifica que la secuencia 2 es más rápida, aun cuando ejecuta una instrucción adicional. Dado que la secuencia de código 2 toma en total menos ciclos de reloj pero tiene más instrucciones, debe tener un CPI menor. Los valores del CPI pueden ser calculados por

$$CPI = \frac{\text{Ciclos de reloj de CPU}}{\text{Cantidad de instrucciones}}$$

$$CPI_1 = \frac{\text{Ciclos de reloj de CPU}_1}{\text{Cantidad de instrucciones}_1} = \frac{10}{5} = 2,0$$

$$CPI_2 = \frac{\text{Ciclos de reloj de CPU}_2}{\text{Cantidad de instrucciones}_2} = \frac{9}{6} = 1,5$$

□

¿Cómo se pueden determinar los valores de estos factores en la ecuación de desempeño? Es posible medir el tiempo de ejecución de CPU ejecutando el programa y la información del tiempo de ciclo de reloj es provista generalmente como parte de la documentación que viene con la computadora. La cantidad de instrucciones y el CPI es más difícil de obtener. Desde luego, si se sabe la frecuencia de reloj y el tiempo de ejecución de CPU, sólo es necesario conocer, o bien la cantidad de instrucciones, o bien el CPI, para determinar el otro.

Es posible medir la cantidad de instrucciones usando herramientas de software que controlan la ejecución o usando un simulador de la arquitectura. Como alternativa es posible usar contadores de hardware, los cuales son incluidos en muchos procesadores, para registrar una variedad de medidas, incluyendo el número de instrucciones ejecutadas, el CPI promedio y en general el origen de la pérdida de desempeño. Debido a que la cantidad de instrucciones depende de la arquitectura, pero no de la implementación exacta, es posible medir la cantidad de instrucciones sin conocer el detalle de la implementación. El CPI depende sin embargo de una variedad de detalles de diseño de la computadora, incluyendo el sistema de memoria, la estructura del procesador y de la combinación de instrucciones ejecutadas en una aplicación. Así, el CPI varía para cada aplicación como también entre implementaciones del mismo conjunto de instrucciones.

Los ejemplos anteriores muestran los riesgos de usar solamente un factor para evaluar el desempeño. Cuando se comparan dos computadoras, es necesario considerar los tres componentes, los cuales se combinan para determinar el tiempo de ejecución. Si alguno de estos factores se repite, como la frecuencia de reloj en el ejemplo de arriba, el desempeño puede ser determinado comparando los factores que no son iguales. Debido a que el CPI varía de acuerdo a la mezcla de instrucciones, tanto la cantidad de instrucciones como el CPI deben ser comparados, aun si la frecuencia de reloj es la misma.

REFERENCIAS

- [1] Patterson, David A. and Hennessy, John L., Computer Organization and Design, Fifth Edition: The Hardware/Software Interface, Morgan Kaufmann Publishers Inc., 2013