



**Universidad
Europea de Madrid**

LAUREATE INTERNATIONAL UNIVERSITIES

INTRODUCCIÓN Y MEDIDAS DE RENDIMIENTO

MEDIDAS DE RENDIMIENTO

© Todos los derechos de propiedad intelectual de esta obra pertenecen en exclusiva a la Universidad Europea de Madrid, S.L.U. Queda terminantemente prohibida la reproducción, puesta a disposición del público y en general cualquier otra forma de explotación de toda o parte de la misma.

La utilización no autorizada de esta obra, así como los perjuicios ocasionados en los derechos de propiedad intelectual e industrial de la Universidad Europea de Madrid, S.L.U., darán lugar al ejercicio de las acciones que legalmente le correspondan y, en su caso, a las responsabilidades que de dicho ejercicio se deriven.

Índice

Presentación	4
Medidas de rendimiento vs. Niveles de descripción	5
Tiempo de ejecución	7
Rendimiento	9
Definiciones y fórmulas	11
Ejemplo I	14
CPI	14
Ejemplo II	15
MIPS	15
Recuento total de instrucciones	15
Otras medidas	17
Tiempo de ejecución, MIPS y CPI	19
Resumen	21

Presentación

Una competencia clave para un arquitecto de computadores es saber analizar el rendimiento de una máquina. Esta capacidad consiste en **saber medir cómo de eficiente es un sistema**. La forma más efectiva de decidir si un sistema es óptimo o no para el cometido que fue diseñado, es compararlo con otros sistemas similares.

No menos importante es **saber aplicar mejoras** sobre una arquitectura ya construida, para lo cual es necesario saber interpretar el diseño de una arquitectura y poder tomar decisiones sobre qué características de la máquina podemos mejorar y que a su vez impacten positivamente en el rendimiento de la máquina.

En este tema el estudiante aprenderá a:

- Conocer las distintas medidas de rendimiento.
- Saber distinguir qué partes hardware y qué nivel de abstracción influyen en cada medida de rendimiento.
- Saber calcular las medidas de rendimiento en base a unas características de partida.
- Conocer cuál es la medida de rendimiento a tener en cuenta, dependiendo de las características de la computadora que se quiere mejorar.



Medidas de rendimiento vs. Niveles de descripción

Dependiendo del nivel de abstracción o de descripción en el que queramos mejorar la máquina, deberemos fijarnos en una u otra medida de rendimiento.

A continuación, se muestran algunas medidas de rendimiento, clasificadas según el nivel de abstracción y la característica que queramos mejorar.

CAPA: SOFTWARE DE APLICACIÓN

NIVELES DE ABSTRACCIÓN	MEDIDAS DE RENDIMIENTO
Generadores de programas	<ul style="list-style-type: none"> • Número de Instrucciones de alto nivel. • Complejidad del algoritmo (Ω).
Programas de aplicación	<ul style="list-style-type: none"> • Tiempo de ejecución (T_{ej}). • Tiempo de respuesta de usuario. • Tiempo de CPU (T_{CPU}). • Operaciones por segundo. • Rendimiento (Performance). • Productividad o Throughput. • Tiempo de respuesta.



1/3 

CAPA: SOFTWARE DE SISTEMA

NIVELES DE ABSTRACCIÓN	MEDIDAS DE RENDIMIENTO
Compiladores e intérpretes	<ul style="list-style-type: none"> • Número de instrucciones máquina por programa (N).
Sistemas operativos	<ul style="list-style-type: none"> • Número de programas en paralelo. • Número de threads en paralelo.

CAPA: HARDWARE

NIVELES DE ABSTRACCIÓN	MEDIDAS DE RENDIMIENTO
Nivel digital	<ul style="list-style-type: none"> • Frecuencia de reloj (F). • Periodo o ciclo de reloj (t, T). • Inferencias lógicas por segundo (KLIPS).
Nivel físico	<ul style="list-style-type: none"> • Número de transistores por componente.

 2/3 

CAPA: INTERFAZ SW/HW

NIVELES DE ABSTRACCIÓN	MEDIDAS DE RENDIMIENTO
Arquitectura del conjunto de instrucciones	<ul style="list-style-type: none">• Ciclos por instrucción (CPI).• Instrucciones por segundo (MIPS).• Instrucciones de coma flotante por segundo (MFLOPS).
Organización de componentes hardware	<ul style="list-style-type: none">• Tiempo de acceso a memoria (Tacc, TAM).• Latencia.• Ancho de banda (Megabytes / segundo).• Transacciones por segundo (TPS).

Tiempo de ejecución

Durante todo este tema, hablaremos de muchas medidas de rendimiento como las que enumerábamos en el apartado anterior.

Estudiaremos algunas de ellas, aprendiendo su definición y memorizando fórmulas para calcularlas. Sin embargo, cuando llegue la hora de medir el rendimiento global de un computador, la prueba definitiva sobre si una arquitectura es óptima o si hemos mejorado sus prestaciones lo suficiente, sólo nos fijaremos en una medida: el tiempo de ejecución.

El tiempo de ejecución de programas reales es la única medida de rendimiento fiable.

El tiempo de ejecución es por tanto, *la medida de la verdad*. De su resultado dependerá que todos los recursos utilizados para la compra y construcción de componentes para diseñar la arquitectura, diseñar algoritmos óptimos o buscar mejoras en las prestaciones, hayan merecido o no la pena.

Definimos el tiempo de ejecución como el tiempo de respuesta del sistema desde que se ejecuta un programa hasta que se obtiene una respuesta.

Hay que distinguir entre el tiempo de ejecución de usuario (T_{ej}), y el tiempo de ejecución del procesador (T_{CPU}).

Tiempo de ejecución de usuario (T_{ej})	Tiempo que el usuario tarda en recibir una respuesta del sistema desde que ejecuta un programa hasta que recibe el resultado por la pantalla o por la interfaz de usuario correspondiente.
Tiempo de ejecución de procesador (T_{CPU})	Tiempo que medimos por software desde que se inicia la ejecución de un programa hasta que se genera el resultado buscado como salida.

Rendimiento

Como ya anunciara Einstein en su teoría de la relatividad: "*dos observadores que se mueven relativamente uno al lado del otro con distinta velocidad, a menudo obtendrán diferentes medidas del tiempo*".

Cuando tratamos de evaluar las prestaciones de una arquitectura de computadores ocurre algo similar. Cuando decimos que el rendimiento de un computador es óptimo, ¿Respecto a qué medida? ¿Comparado con qué?



El rendimiento de un computador es *relativo*, dependiendo del fin para que sea utilizado y de cómo de bueno sea comparado con máquinas similares que fueron construidas para el mismo fin o similar.

Para medir el rendimiento de un computador, ya tenemos claro que la única medida fiable es el tiempo de ejecución, pero no hemos delimitado cuál es el programa que ejecutamos, ni cuál es la medida de tiempo que se espera. Para demostrar que una computadora es óptima, debemos ejecutar un programa real (o un programa de prueba de complejidad similar), y comparar los tiempos de ejecución con una computadora similar. Cuanto menor sea el tiempo de ejecución, mejor es nuestro sistema.

Reducir el tiempo de ejecución, mejora el rendimiento de un sistema.

Para comparar el rendimiento de los computadores se usa como medida estándar el *número de veces más rápido que el VAX-11/780*.

El VAX-11/780 fue la primera máquina de la familia VAX. Ejecutaba aproximadamente 1 millón de instrucciones por segundo (1 MIPS) con programas reales. El procedimiento consiste en ejecutar un conjunto de programas de prueba llamados *benchmarks* que implementan algoritmos ya definidos, y comparar sus resultados con las tablas de tiempos ya calculadas para el VAX.

Definiciones y fórmulas

En el elemento interactivo siguiente, podrás explorar las definiciones de las medidas de rendimiento más importantes y las fórmulas asociadas a cada medida, para saber cómo calcularlas.

Al final de este tema debemos ser expertos conocedores de estas medidas y ser capaces de calcularlas.

TIEMPO DE EJECUCIÓN

Definición


- Tiempo que el usuario tarda en recibir una respuesta del sistema desde que ejecuta un programa, hasta que recibe el resultado por la pantalla o interfaz de usuario (Acrónimo: Tej).
- Tiempo que el procesador tarda en ejecutar un programa (Acrónimo: T_{CPU}).

Fórmulas

- $T_{CPU} = N * CPI * t$
- $T_{CPU} = t * \sum_{i=1}^n (CPI_i * I_i)$

Unidad: segundos

LEYENDA	
N	Recuento de instrucciones=instrucciones/programa.
t	Período de reloj.
CPI	Ciclos por instrucción.
CPI_i	Ciclos por instrucción de tipo i.
I_i	Nº de Instrucciones de tipo i.
n	Nº de tipos de instrucciones.

1/4 

CICLOS POR INSTRUCCIÓN**Definición**

- Número de ciclos de instrucción que transcurren en un segundo (Acrónimo: CPI).

Fórmulas

- $CPI = C / N$
- $CPI = \sum_{i=1}^n (CPI_i * F_i)$
- $CPI = (Tej * F) / N$

Unidad: ciclos/instrucción

LEYENDA

C	Número ciclos / programa.
N	Recuento de instrucciones=instrucciones/ programa.
F _i	Frecuencia (% 0..1) de aparición de instrucciones de tipo i.
F	Frecuencia de Reloj (Hz).

◀ 2/4 ▶

INSTRUCCIONES POR SEGUNDO**Definición**

- Millones de Instrucciones ejecutadas en un segundo (Acrónimo MIPS).

Fórmulas

- $MIPS = (N * 10^{-6}) / Tej$
- $MIPS = F \text{ (MHz)} / CPI$

Unidad: MIPS = 10⁶ instr./ seg

LEYENDA

N	Recuento de instrucciones=instrucciones/ programa.
Tej	Tiempo de ejecución.
F (MHz)	Frecuencia de reloj medida en MHz.
CPI	Ciclos por instrucción.

◀ 3/4 ▶

PERFORMANCE**Definición**

- Número de veces más rápido que el VAX 11/780 (Acrónimo: Performance (X) = P(X)).
- X es n veces más rápido que Y (Acrónimo: n = Performance (X/Y)).

Fórmulas

- $P(X) = \text{Tej (VAX)} / \text{Tej (X)}$
Unidad: Performance de X. X es P(X) veces más rápido que el VAX.
- $n = \text{Tej (Y)} / \text{Tej (X)} = P(X) / P(Y)$
Unidad: X es n veces más rápido que Y.

LEYENDA

Tej (X)

Tiempo de ejecución de X

Ejemplo I

Tras ejecutar un programa de prueba en un Intel Pentium de 166 MHz, se obtuvo un tiempo de ejecución de 6 segundos, y la siguiente distribución de cada tipo de instrucciones.

Tipo de Instrucción	Frecuencia de uso	Ciclos de reloj
Operaciones Aritmético-lógicas	43%	1
Carga desde Memoria	21%	2
Almacenamiento en Memoria	12%	2
Instrucciones de salto	24%	2

Queremos calcular el CPI, MIPS, y el recuento total de instrucciones.

CPI

Como los datos que nos dan incluyen la distribución de instrucciones (en porcentaje) de cada tipo, cuántos ciclos dura cada tipo de instrucción, usaremos la fórmula basada en tipos de instrucción:

$$CPI = \sum_{i=1}^n (CPI_i \cdot F_i) = 0,43 \cdot 1 + 0,21 \cdot 2 + 0,12 \cdot 2 + 0,24 \cdot 2 = 1,57 \text{ ciclos / instrucción}$$

$$CPI = 1,57 \text{ CPI}$$

(Continúa en la pantalla siguiente).

Ejemplo II

MIPS

Para el cálculo de MIPS, usaremos la fórmula basada en la frecuencia del procesador y el CPI que acabamos de calcular

$$\text{MIPS} = F \text{ (MHz)} / \text{CPI} = 166 / 1,57 = 105,76 \text{ Millones de Instrucciones / segundo}$$

$$\text{MIPS} = 105,76$$

Recuento total de instrucciones

Para calcular el número total de instrucciones, tendremos que basarnos en los datos ya calculados y aplicar alguna fórmula de las que ya conocemos.

Por ejemplo, podemos usar la fórmula del tiempo de ejecución, que es un dato que sí conocemos (6 segundos):

$$T_{\text{CPU}} = N * \text{CPI} * t$$

Despejamos N (recuento de instrucciones): $N = T_{\text{CPU}} / (\text{CPI} * t)$

No sabemos el ciclo de reloj t , pero conocemos la frecuencia de reloj $F = 166 \text{ MHz} = 166 * 10^6 \text{ Hz}$

Por tanto el ciclo de reloj es: $t = 1 / (166 * 10^6)$

$N = 6 * (166 * 10^6) / 1,57 = 634,39$ millones de instrucciones

$N = 634,39 \text{ M instr.}$

¿Tú lo habrías calculado de otra forma?

Otras medidas

A parte de las medidas de rendimiento que ya hemos estudiado, existen otras medidas de rendimiento que son más específicas y que sirven para medir características concretas de una máquina.

FLOPS	<p>Número de operaciones en coma flotante.</p> <p>Sirve para medir el rendimiento de la máquina cuando trabaja con números reales. Es una medida de rendimiento muy importante, similar a los MIPS.</p>
Productividad (Throughput)	<p>Número de tareas realizadas por el computador en unidad de tiempo.</p> <p>Por ejemplo, 1000 tareas / segundo. Está directamente relacionado con el rendimiento global de la máquina y el tiempo de ejecución.</p>
KLIPS	<p>Número de operaciones lógicas que se realizan en un segundo, medido en miles de operaciones.</p> <p>Sirve para medir el rendimiento de la máquina respecto a las operaciones lógicas.</p>

TPS	<p>Número de transferencias de información por un segundo.</p> <p>Sirve para medir la latencia o velocidad de transferencia de información a través del Bus de Datos. Los TPS miden cuántos envíos realizamos por segundo. Por tanto, el ancho de banda dependerá del ancho del bus (número de hilos o bits que retransmitimos por envío) y del TPS.</p>
-----	--



Tiempo de ejecución, MIPS y CPI

Ya sabemos que el tiempo de ejecución es la única medida fiable, pero hay otras medidas íntimamente relacionadas como son los MIPS y los CPI.

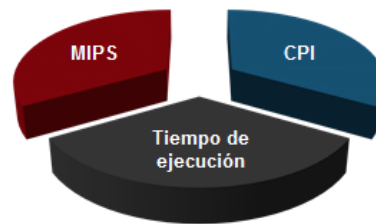
¿Podemos fiarnos de los MIPS para comparar el rendimiento de dos sistemas? Aparentemente los MIPS son una buena medida, pues cuántas más instrucciones ejecutemos, mejor.

Pero fijémonos en una cuestión: puede que, al implementar un mismo algoritmo en dos arquitecturas distintas, el programa resultante tenga muchas más instrucciones en el lenguaje máquina que usa una computadora que la otra.

¿Cómo es posible? El número de instrucciones dependerá del juego de instrucciones. Las arquitecturas RISC (Reduced Instruction Set Computer) tienen un juego reducido de instrucciones mucho más genéricas que las arquitecturas CISC (Complex Instruction Set Computing). Por tanto, en las arquitecturas RISC necesitaremos más instrucciones (aunque duran menos tiempo) para llevar a cabo el mismo programa.

Esto implica que aunque ejecutemos muchas más instrucciones por segundo en una arquitectura RISC, puede que no ejecutemos el programa en menos tiempo. Es decir, dependemos del CPI (ciclos por instrucción).

Las instrucciones en las arquitecturas CISC tienen un CPI mucho mayor que las arquitecturas RISC, pues las instrucciones RISC son mucho más sencillas y genéricas. Ahora bien, para hacer la misma operación, por ejemplo multiplicar o calcular la raíz cuadrada, necesitamos varias instrucciones en RISC, y una sola en CISC.



Como conclusión, encontramos que el rendimiento al comparar dos máquinas, depende de la frecuencia del procesador (¿cuánto dura un ciclo?), y del número de ciclos ejecutados al finalizar el programa.

Debemos por tanto, ejecutar el programa en las dos máquinas y comparar los tiempos de ejecución.

Resumen

Para poder diseñar una arquitectura óptima que ofrezca el rendimiento esperado para el fin para el que se construyó, es imprescindible conocer cuáles son las medidas de rendimiento que me permiten medir las prestaciones globales de un sistema. Igualmente, este conocimiento me permitirá mejorar una característica concreta o el rendimiento global de una arquitectura ya diseñada o incluso ya fabricada.

La única medida de rendimiento fiable es el tiempo de ejecución. El tiempo de ejecución medirá el tiempo que tarda en ejecutarse un programa de prueba en dos máquinas distintas. Esta medida nos permitirá comparar el rendimiento de una máquina respecto a la otra, o de una configuración respecto de otra dentro de una misma máquina. El programa de prueba o programas de prueba deben tener una complejidad similar a los programas reales que normalmente se ejecutarán, para que las medidas sean lo más fiables posibles.

Otras medidas importantes son los MIPS, el CPI, y la frecuencia del procesador. Estas medidas nos permitirán, por un lado estimar cuál será el tiempo de ejecución final, dependiendo del programa ejecutado. Por otro lado, permitirán medir características particulares del sistema como el número de instrucciones que nuestro sistema ejecuta por unidad de tiempo, la complejidad del juego de instrucciones (genérico o específico), o la velocidad del procesador (el ciclo de reloj).