
EL RENDIMIENTO EN LOS COMPUTADORES

2

2.1- Introducción.....	2
2.2- Rendimiento en los computadores.....	2
2.3.- Comparación del rendimiento.....	6
2.4- Unidades de medida del rendimiento global	8
2.4.1- MIPS.....	8
2.4.2- MIPS VAX	9
2.4.3- MEGAFLOPS nativos y relativos	10
2.5- Programas de prueba o Benchmarks	12
2.5.1- Benchmarks Sinteticos.....	12
2.5.2- Benchmarks Reducidos o de juguete.....	13
2.5.3- Kernel Benchmarks.....	13
2.5.4- Programas Reales.....	13
2.5.4.1 Programa de pruebas SPEC (System Performance Evaluation Cooperative)	13
2.5.4.2 Otros benchmarks	16
2.5.5- Programas propios de fabricantes.....	16
2.6- Leyes de Amdahl	17
2.6.1- Primera ley de amdahl.....	17
2.6.2- Segunda ley de amdahl.....	18
2.7- Media aritmética y geométrica.....	19
2.8- Relación coste rendimiento	20
2.8.1- Análisis de la distribución de los costes de un computador.....	20
2.9- Ejercicios propuestos.....	22

2.1- INTRODUCCIÓN

En este capítulo se analiza el concepto y las unidades de medida del rendimiento de los computadores.

Definimos en general rendimiento de un elemento como una medida de productividad o capacidad de trabajo de dicho elemento en función de unos parámetros o características.

El rendimiento no es una medida única, sino una relación entre varias medidas, la valoración de varios parámetros, y depende del objetivo al que se oriente el trabajo.

EJEMPLO 1

Para entender mejor el concepto de rendimiento analizaremos el siguiente ejemplo, se trata de determinar cuál de los dos automóviles tiene mejor rendimiento.

Coche	Consumo	Velocidad	Plazas	C.V	Precio
A	4.5	145	4	90	13.400€
B	7.1	185	7	110	15.300€

Figura 2.1. Prestaciones de los vehículos.

Como hemos comentado anteriormente, el rendimiento dependerá de la finalidad a la que se quiera destinar el vehículo y de unos parámetros, en este caso consumo, velocidad, etc :

- Si se utiliza para transportar 2 personas diariamente teniendo en cuenta el coste, el rendimiento mejor será el del vehículo A.
- Si el vehículo es requerido para recorrer 100 km y transportar 6 personas, el rendimiento mejor será el del vehículo B.
- Si lo que prima es el tiempo en un determinado recorrido la elección acertada es el B.

2.2-RENDIMIENTO EN LOS COMPUTADORES

Se define **rendimiento de un sistema** como la capacidad que tiene dicho sistema para realizar un trabajo en un determinado tiempo. Es inversamente proporcional al tiempo, es decir, cuanto mayor sea el tiempo que necesite, menor será el rendimiento.

Los computadores ejecutan las instrucciones que componen los programas, por lo tanto el **rendimiento de un computador** está relacionado con el tiempo que tarda en ejecutar los programas. De esto se deduce que el tiempo es la medida del rendimiento de un computador.

El *rendimiento del procesador* depende de los siguientes parámetros:

1. Frecuencia de la CPU (f_{CPU}) : es el número de ciclos por segundo al que trabaja el procesador o CPU. No confundir la frecuencia de la CPU con la frecuencia del sistema, el bus del sistema trabaja a menor frecuencia que la CPU.

$$f_{CPU} = \frac{\text{n}^\circ \text{ ciclos}}{\text{segundo}}$$

2. Periodo de la CPU (T_{CPU}) : es el tiempo que dura un ciclo y es la inversa de la frecuencia de la CPU.

$$T_{CPU} = \frac{1}{f_{CPU}}$$

3. Ciclos por instrucción (CPI) : las instrucciones se descomponen en microinstrucciones, que son operaciones básicas que se hacen en un ciclo de reloj. En un programa se llama CPI al promedio de microinstrucciones que tienen las instrucciones del programa, es decir, los ciclos de reloj que se tarda de media en ejecutar una instrucción.

$$CPI = \frac{\sum_{i=1}^n N^\circ \text{ Instruc}_i * CPI_i}{N^\circ \text{ InstrucTot}}$$

4. Número de instrucciones del programa : cuantas más instrucciones haya en el programa más tiempo se tarda en ejecutarlo luego baja el rendimiento. El que tengamos un número reducido de instrucciones dependerá del programador y de que dispongamos de un buen compilador.

5. Multitarea : hace referencia a la capacidad que tiene un computador de atender simultáneamente varias tareas.

Como anteriormente hemos comentado, el *rendimiento de un procesador* para un programa concreto es un factor inversamente proporcional al tiempo que tarda en ejecutar dicho programa.

$$\eta_{\text{prog}} = \frac{1}{T_{\text{prog}}}$$

El *tiempo de programa* depende a su vez del número de instrucciones del programa y del tiempo que se tarda en ejecutar cada instrucción.

$$T_{\text{Prog}} = N^\circ \text{instruc.} * T_{\text{instrucciones}}$$

El **tiempo** que tarda en ejecutarse cada **instrucción** depende del número de microinstrucciones o ciclos en los que se descompone. Cada microinstrucción tarda distintos ciclos de reloj en ejecutarse, por lo que se hace un promedio ponderado de ciclos de instrucción.

$$T_{\text{instrucción}} = \text{CPI} * T_{\text{CPU}}$$

$$T_{\text{prog}} = N^{\circ} \text{ instrucciones} * \text{CPI} * T_{\text{CPU}}$$

$$\eta_{\text{prog}} = \frac{1}{T_{\text{prog}}} = \frac{1}{N^{\circ} \text{instruc} * \text{CPI} * T_{\text{CPU}}} = \frac{f_{\text{CPU}}}{N^{\circ} \text{instruc} * \text{CPI}} \text{ s}^{-1}$$

En resumen, el rendimiento de un procesador para un determinado programa queda en función de tres factores:

- **Frecuencia de la CPU** la cual depende fundamentalmente de la tecnología de fabricación del procesador. Cuanto mayor sea la frecuencia de la CPU, mejor será el rendimiento.
- **Número de instrucciones del programa** el cual depende del programador, del lenguaje de programación y del compilador. Cuanto mayor sea el número de instrucciones del programa peor rendimiento tendrá.
- **CPI** que depende de diseño interno o arquitectura del computador y del software o instrucciones que se hayan elegido. Es importante optimizar el programa con instrucciones que tengan pocos ciclos. Cuanto mayor sea el CPI, peor será el rendimiento.

EJEMPLO 2

Dadas las características de un programa calcular el tiempo que tarda en ejecutarse el programa y su rendimiento. Se sabe que la CPU trabaja a 1GHz. El programa se compone de 19 instrucciones de tres tipos: A, B y C y cada tipo de instrucción tiene un número de microinstrucciones diferente según indica la tabla:

Tipo de instrucción	Instrucciones/Prog.	CPI _i
A	5	2
B	6	3
C	8	1

Tiempo de programa:

$$\text{CPI} = \frac{\sum N^{\circ} \text{instruc.} * \text{CPI}_i}{N^{\circ} \text{instruc. TOTAL}} = \frac{(5 * 2 + 6 * 3 + 8 * 1)}{19} = 1.89$$

$$T_{prog} = N^{\circ}instruc. * CPI * T_{CPU} = 19 * 1.89 * \frac{1}{1000 * 10^6} = 36ns$$

Rendimiento:

$$\eta_{prog} = \frac{1}{T_{prog}} = \frac{1}{36ns} = 27777777.78 \text{ s}^{-1}$$

EJEMPLO 3

Se dispone de un computador basado en un Pentium IV a 2GHz. Se ejecuta un programa compuesto por 1000 instrucciones. Según los datos de la siguiente tabla, calcular el rendimiento:

Tipo de instrucción	Instrucciones/Prog.	CPI _i
Lógico-Aritmética	20%	4
Salto incondicional	10%	3
Salto condicional	10%/10%	7/5
Carga (Reg<-Mem)	30%	3
Almacenamiento (Mem<-Reg)	20%	4

Tiempo de programa:

$$CPI_{Pentium} = \frac{(200 * 4 + 100 * 3 + (100 * 7 + 100 * 5) + 300 * 3 + 200 * 4)}{1000} = 4$$

$$T_{prog} = N^{\circ}instruc. * CPI * T_{CPU} = 1000 * 4 * \frac{1}{2000 * 10^6} = 2000ns$$

Rendimiento:

$$\eta_{prog} = \frac{1}{T_{prog}} = \frac{1}{2000ns} = \frac{10^9}{2 * 10^3} = 500000 \text{ s}^{-1}$$

2.3.- COMPARACION DEL RENDIMIENTO

El objetivo principal de los compradores, diseñadores y otros usuarios de computadores, es conseguir el mayor rendimiento posible con el menor coste.

El rendimiento, por tanto, ofrece una medida cuantitativa que permite comparar unos computadores frente a otros y decidir cual nos aporta mayores ventajas.

Cuando comparemos 2 máquinas hablaremos de **aceleración** :

1.- Aceleración del rendimiento : Es la relación que existe entre el rendimiento del computador A y el rendimiento del computador B. Indica que el computador A es un n% mejor que el B.

$$A_{\eta} = \frac{\eta_A}{\eta_B}$$

2.- Aceleración del tiempo : Es la relación entre el tiempo que tarda el computador A en ejecutar un programa y el tiempo que tarda el computador B, y es inversamente proporcional al rendimiento.

$$A_T = \frac{T_A}{T_B} = \frac{\eta_B}{\eta_A}$$

EJEMPLO 4

Tenemos un PentiumIV a 500 MHz y ejecuta un programa de 1000 instrucciones, según la siguiente tabla calcular el rendimiento. Si sustituimos el procesador por un PentiumIV a 1GHz, ¿cómo varia el rendimiento?.

Tipo de instrucción	Instrucciones/Prog.	CPI _i
Carga y Almacenamiento	30%	3
Lógico Aritméticas	50%	4
Salto	20%	2

$$a) CPI = \frac{(300 * 3 + 500 * 4 + 200 * 2)}{1000} = 3,3$$

$$\eta_{prog} = \frac{f_{cpu}}{n^{\circ} inst. * cpi} = \frac{500 * 10^6}{1000 * 3,3} = 151515,15 \text{ s}^{-1}$$

b) El rendimiento se multiplicará por 2 ya que es directamente proporcional a la frecuencia.

EJEMPLO 5

Tenemos un PentiumIV a 2GHz y ejecuta un programa de 1000 instrucciones, según la siguiente tabla calcular el rendimiento. Para aumentar el rendimiento se añade posteriormente una caché que ahorra 1 ciclo en accesos si el dato está en la caché y si no está dura lo mismo. El porcentaje de acierto es del 90%. Calcular el rendimiento y la aceleración del rendimiento con respecto del anterior sin caché.

Tipo de instrucción	Instrucciones/Prog.	CPI _i
Lógico-Aritmética	300	5
Llamadas a subrutinas	100	9
Salto incondicional	100	6
Salto condicional	50/50	9/3
Carga (Reg<-Mem)	100	2
Almacenamiento (Mem<-Reg)	100	4
Mov (Reg<-Reg)	200	2

$$CPI_{\text{sin caché}} = \frac{(300 * 5 + 100 * 9 + 100 * 6 + (50 * 9 + 50 * 3) + 100 * 2 + 100 * 4 + 200 * 2)}{1000} = 4,6$$

$$\eta_{\text{sin caché}} = \frac{2 * 10^9}{1000 * 4,6} = 434708,6 \text{ s}^{-1}$$

$$CPI_{\text{con caché}} = \frac{(\dots + 100 * 6 + (50 * 9 + 50 * 3) + (90 * 1 + 10 * 2) + (90 * 3 + 10 * 4) + 200 * 2)}{1000} = 4,42$$

$$\eta_{\text{con caché}} = \frac{2 * 10^9}{1000 * 4,42} = 452488 \text{ s}^{-1}$$

$$CPI_{\text{sin caché}} = \frac{(300 * 5 + 100 * 9 + 100 * 6 + (50 * 9 + 50 * 3) + 100 * 2 + 100 * 4 + 200 * 2)}{1000} = 4,6$$

$$A\eta = \frac{\text{rend.concaché}}{\text{rend.sin caché}} = \frac{452488}{434782} = 1,0407 \text{ s}^{-1}$$

Hemos conseguido un aumento en el rendimiento de un 1% al incorporar caché.

2.4- UNIDADES DE MEDIDA DEL RENDIMIENTO GLOBAL

En la búsqueda de una medida estándar del rendimiento de los computadores, se han desarrollado una serie de métricas populares como alternativa al uso del tiempo, el cual ha conducido en alguna ocasión a resultados distorsionados o interpretaciones incorrectas. En esta sección se explican algunas de las métricas más comúnmente usadas.

2.4.1- MIPS

Los MIPS son los millones de instrucciones por segundo que ejecuta un procesador para un programa determinado.

$$MIPS = \frac{N^{\circ} \text{instrucprog}}{T_{\text{prog}} * 10^6} = \frac{N^{\circ} \text{instrucprog}}{N^{\circ} \text{instrucprog} * CPI * T_{\text{CPU}} * 10^6} = \frac{f_{\text{CPU}}}{CPI * 10^6}$$

La ventaja de esta unidad de medida es su fácil comprensión ya que un mayor número de MIPS indicará una mayor velocidad de la máquina. Sin embargo, no es una medida buena ya que puede producir resultados erróneos, un mismo programa en un mismo computador puede dar resultados diferentes según el programador o el repertorio de instrucciones utilizadas, además los MIPS pueden variar inversamente al rendimiento. Por ese motivo, se pasó a utilizar los modelo de referencia el VAX11/780.

EJEMPLO 6

Se dispone de una máquina con dos clases de instrucciones, la instrucción 1 tendrá un CPI de 3 y la instrucción 2 un CPI de 2. Al medir el código para el mismo programa para dos compiladores diferentes se obtienen los siguientes datos.

Código	Instrucción 1	Instrucción 2
Compilador A	6	2
Compilador B	4	2

Se supone que la frecuencia de reloj de la máquina es de 200MHz. ¿Qué secuencia de código se ejecuta con más rapidez de acuerdo con los MIPS?. Realizar lo mismo de acuerdo al tiempo de ejecución. Explicar los resultados.

Para calcular los MIPS utilizamos la ecuación:

$$MIPS = \frac{f_{\text{CPU}}}{CPI * 10^6}$$

$$CPI = \frac{\sum_{i=1}^N (CPI_i * I_i)}{N^{\circ} \text{instruc TOTAL}}$$

$$CPI_A = \frac{(6 * 3 + 2 * 2)}{(6 + 2)} = \frac{22}{8} = 2.75$$

$$MIPS_A = \frac{200 * 10^6}{2.75 * 10^6} = 72.72$$

$$CPI_B = \frac{(4 * 3 + 2 * 2)}{(4 + 2)} = \frac{16}{6} = 2.66$$

$$MIPS_B = \frac{200 * 10^6}{2.66 * 10^6} = 75.18$$

Por consiguiente, el código producido por el compilador B tiene una frecuencia en MIPS más elevada.

Calcularemos ahora el tiempo de ejecución y para ello utilizaremos la formula:

$$T_{\text{Prog}} = \frac{CPI * N^{\circ} \text{Instruc}}{f_{\text{CPU}}}$$

$$T_{\text{PROG A}} = \frac{2.75 * (6 + 2)}{200 * 10^6} = 1,1^{-7} s^{-1}$$

$$T_{\text{PROG B}} = \frac{2.66 * (4 + 2)}{200 * 10^6} = 79,8^{-7} s^{-1}$$

Por tanto, el compilador A es claramente más rápido lo cual contradice lo que habíamos observado con los MIPS.

2.4.2- MIPS VAX

Los MIPS VAX son la relación entre el tiempo que tarda un computador en realizar un programa y el tiempo que tarda en realizarlo el computador VAX11/780. Por ejemplo, un valor de MIPS VAX = 1 que toma el 80286 significa que dicho computador ejecuta un programa en el mismo tiempo que el VAX11/780.

$MIPS VAX = \frac{T_{\text{Prog VAX11/780}}}{T_{\text{Prog CPU}}}$
--

2.4.3- MEGAFLOPS NATIVOS Y RELATIVOS

Surgen ya que los MIPS no hacen distinción entre operaciones normales y operaciones en coma flotante.

Los **Megaflops nativos** indican los de millones de instrucciones en coma flotante por segundo que se realizan para un determinado programa.

$$MFLOPS_{nativos} = \frac{\text{Millones instruc}}{T_{Prog}} * 10^6$$

Los Megaflops nativos tampoco son fiables, ya que dependen mucho de programas y computadores. Tuvieron muchas críticas de los fabricantes de computadores, porque, hay computadores que solo tienen operaciones sencillas (ADD, SUB, MUL) y generan pocas operaciones en coma flotante, en cambio hay otros computadores que tienen más operaciones en coma flotante (DIV, SQRT, EXP, SIN, ...). Las operaciones sencillas tardan menos en ejecutarse que las operaciones complejas, por lo que apareció el concepto de Megaflops relativos.

Los **Megaflops relativos o normalizados** indican los millones de operaciones en coma flotante por segundo pero teniendo en cuenta la equivalencia que tienen las operaciones complejas con las simples.

$$MFLOPS_{Relativos} = \sum (\text{MillonesInstruc}_i * \text{Peso}_i)$$

Instrucciones	Peso
ADD, SUB, MULT	1 operación en coma flotante
DIV, SQRT...	4 operación en coma flotante
EXP, SIN,...	8 operación en coma flotante

Figura 2.2. Equivalencias.

EJEMPLO 7

Hallar los MFLOPS nativos y relativos que se obtienen al utilizar un computador que ejecuta un programa en 66 segundos y que consta de las siguientes instrucciones en coma flotante.

Operación	Millones de instrucciones	Peso
ADD	20	1
SUB	10	1
MUL	10	1
DIV	12	4
EXP	2	8
SIN	99	8
SQRT	1	4

$$MFLOPS_{nativos} = \frac{(20 + 10 + 10 + 12 + 2 + 99 + 1) * 10^6}{66 * 10^6} = 2,33$$

$$MFLOPS_{relativos} = \frac{(20 * 1 + 10 * 1 + 10 * 1 + 12 * 4 + 2 * 8 + 99 * 8 + 1 * 4) * 10^6}{66 * 10^6} = 13,64$$

El resultado de los Mflops relativos es mayor porque utiliza muchas instrucciones de alta complejidad.

Tabla de rendimiento comparativa en Megaflops (millones de operaciones por segundo), obtenida de la revista PC Today para índices PC y estimación Apple para Mac:

Equipo	Rendimiento
Pentium III a 800 MHz	234,9 Megaflops
Pentium IV a 1.400 MHz	362,4 Megaflops
AMD Athlon a 1.000 MHz	486,4 Megaflops
PowerPC G4 a 450 MHz	3.000 Megaflops
PowerPC G4 a 500 MHz	7.000 Megaflops

Figura 2.3.Rendimiento en Megaflops.

Comparando:

PowerPC G4 a 450 MHz es:

- 12 veces superior al de un Pentium III a 800 MHz
- 8 veces superior al de un Pentium IV a 1,4 GHz
- 6 veces superior al de un AMD Athlon a 1 GHz

PowerPC G4 a 500 MHz es:

- 29 veces superior al de un Pentium III a 800 MHz
- 19 veces superior al de un Pentium IV a 1,4 GHz
- 14 veces superior al de un AMD Athlon a 1 GHz

2.5- PROGRAMAS DE PRUEBA O BENCHMARKS

Las medidas de rendimiento vistas hasta ahora no son válidas hoy en día dado que algunas como los MIPS tienden a dar resultados erróneos y a que los computadores actuales tienen una elevada velocidad. La mejor y mas fiable forma de calcular el rendimiento es medir el tiempo que los diversos computadores tardan en ejecutar los programas que realmente el usuario va a utilizar posteriormente. Ese será el mejor rendimiento para ese usuario, pero no para todos los usuarios, ya que el rendimiento es un valor relativo de acuerdo con la aplicación que se va a hacer.

El rendimiento de una estación de trabajo se mide analizando una serie de componentes físicos que determinan el rendimiento completo del sistema. A la hora de determinar el rendimiento global de un sistema, también hay que evaluar el sistema operativo, los equipos lógicos de red, los compiladores y las librerías gráficas, etc.

Para la evaluación del rendimiento de los sistemas se utilizan pruebas de rendimiento o benchmarks, que son programas modelo que efectúa la industria para comparar factores de rendimiento y relaciones rendimiento / precio de los diferentes modelos de computadores. No obstante, estas evaluaciones no son siempre directamente comparables, y en ocasiones ofrecen poca información, porque las configuraciones con las que se realizan las evaluaciones no son expuestas con claridad.

Hay multitud de programas de prueba o benchmarks. Estos programas se dividen principalmente en 4 grupos, los tres primeros tipos han quedado en desuso.

- BENCHMARK SINTÉTICOS
- BENCHMARKS REDUCIDOS
- BENCHMARK KERNEL O DE NÚCLEO
- PROGRAMAS REALES

2.5.1- Benchmarks Sinteticos

El objeto de este tipo de programas de prueba es simular el comportamiento de aplicaciones del mundo real. Para elaborar estas pruebas sintéticas se estudian una serie de aplicaciones y se desarrolla un código artificial que mezcla los cálculos matemáticos, bucles, llamadas a funciones, etc. Las series de programas de prueba sintéticos más conocidas son Whetstone y Dhrystone. Los Benchmark Sintéticos están formados por las rutinas más repetitivas de los programas más utilizados.

- **Dhrystone Benchmark (MIPS)** : miden la eficiencia del procesador y del compilador en un entorno de desarrollo de sistemas con lenguajes de alto nivel. Su valor es expresado en instrucciones Dhrystone por segundo (Dhrystone MIPS, millones de instrucciones Dhrystone por segundo). No realiza operaciones en coma flotante, por lo que muchos fabricantes no lo consideran como una medida adecuada para definir el rendimiento de hoy en día. Los resultados se relativizan respecto al número de instrucciones Dhrystone por segundo que son alcanzadas en un VAX 11/780.
- **Whetstone Benchmark** : predecesora del Dhrystone, es una medida desarrollada para evaluar sistemas que se vayan a destinar a la ejecución de pequeños programas científicos y de ingeniería. Sus programas se han implementado en FORTRAN e incluyen cálculos con enteros y en coma flotante, manipulación de arrays y saltos condicionales. Esta prueba predice cómo serán ejecutadas aplicaciones que hacen un uso intensivo de la unidad central de proceso. Los resultados son expresados en Kwips (miles de instrucciones Whetstone por segundo)

Estos dos tipos de Benchmarks, a partir de los 80, cayeron en desuso y aparecieron los Benchmark reducidos.

2.5.2- Benchmarks Reducidos o de juguete

Los programas reducidos tienen entre 10 y 100 líneas de código y producen un resultado que el usuario conoce antes de ejecutarlo. Algunos ejemplos de este tipo de Benchmarks serían: el Towers, que resuelve el problema de las torres de Hanoi con muchas llamadas recursivas; el Perm, que calcula permutaciones de 7 tornadas de 5 en 5 y los programas Criba de Eratóstenes, Puzzle y Quicksort, que son los más populares porque son pequeños, fáciles de introducir y de ejecutar en cualquier computador.

Estos programas, al ser tan pequeños y sencillos, eran muy vulnerables, era muy fácil mejorar el rendimiento para un programa concreto, por lo que se pasó a los Benchmark Kernel o de núcleo.

2.5.3- Kernel Benchmarks

Son programas de pruebas formados por pequeñas piezas clave de programas reales que evalúan el rendimiento y lo aíslan de las características individuales de una máquina, permitiendo explicar las razones de las diferencias en los rendimientos de programas reales.

Los ejemplos más conocidos son el “Livermore Loops”, una serie de 21 fragmentos de bucles pequeños, y el “Linpack”, formado por un paquete de subrutinas de álgebra lineal. Sólo tratan algunos aspectos y son antiguos. No existen núcleos para evaluar prestaciones gráficas.

2.5.4- Programas Reales

Son programas hechos con partes de programas que realmente se utilizan mucho (procesadores de texto, compiladores, herramientas CAD, etc). Los problemas que presentan son que dependen mucho de los datos de entrada, suelen ser complejos de usar, los S.O. sobre los que se prueban suelen ser incompatibles y no son ni estándar ni de libre distribución.

2.5.4.1 Programa de pruebas SPEC (System Performance Evaluation Cooperative)

Para evitar el caos que había al medir el rendimiento, en 1986 se asociaron una serie de empresas (IBM, SUN, INTEL, APPLE, ...) y constituyeron una organización sin ánimo de lucro para la evaluación del rendimiento de los computadores. Esta organización tenía la función de diseñar y proponer los programas y pruebas que debían cumplir los computadores para la evaluación de su rendimiento. La mayoría de los fabricantes de ordenadores incluyen en sus páginas Web las medidas SPEC de sus equipos, eso sí, con las críticas correspondientes y poniendo claramente de relieve donde sobresalen. Incluso se basan en los resultados dados en estos benchmarks para tomar decisiones de diseño con respecto a sus máquinas.

En 1986 apareció la primera normativa y cada 3 años aproximadamente se modifican los programas teniendo en cuenta partes de programas actuales. SPEC consiste en realidad en tres grupos diferentes:

- OSG (Open Systems Group), que crea benchmarks para procesadores y sistemas que ejecutan UNIX, Windows NT y VMS.
- HPC (High Performance Group), que mide prestaciones de ordenadores dedicados a cálculo intensivo.
- GPC (Graphics Performance Characterization Group), que mide prestaciones de subsistemas gráficos, OpenGL y XWindows.

La evolución de los SPEC ha sido :

Primera versión 1989

La primera normativa se basó en 10 programas de los cuales 4 eran con números enteros y 6 con números en coma flotante. Se tomó como referencia el VAX11/780 y teniendo en cuenta lo que tarda en ejecutar los 10 programas se obtuvo una medida SPEC. Inicialmente se escogieron tres índices:

- **SPECint86** : es la media geométrica de los 4 programas de números enteros.
- **SPECfp89** : es la medida geométrica de los 6 programas para números en coma flotante.
- **SPECmark89** : es la media de las dos anteriores.

En 1989 se cambió la normativa porque se daba demasiada importancia a las operaciones en coma flotante.

Segunda versión 1992

En 1992 se redefinieron, incrementando el número de Benchmarks a 20, de los cuales 6 eran de números enteros y 14 con números en coma flotante. Se eliminó el concepto de SPECmark ya que favorecía a máquinas con alto rendimiento de punto flotante e introdujo el concepto de SPECrate basado en la ejecución multitarea. Estos programas tenían una variedad y un tamaño tal que era muy difícil que el fabricante mejorara el índice SPEC variando una parte de su computador.

- **SPECint92** : es un indicador del rendimiento de la UCP en un entorno comercial gcc, compress, espresso, etc.
- **SPECfp92** : es una buena medida del comportamiento en entornos de operación en coma flotante, como pueden ser entornos de ingeniería y científicos spice2g6, programas de astrofísica, partículas, procesamiento de señales, etc.

Tercera versión 1995

- **SPECint95** : 8 programas enteros.
- **SPECfp95** : 10 programas F.P..

Cuarta versión 2000, SPEC CPU2000

- **CINT2000** : 11 programas enteros.
- **CFP2000** : 14 programas F.P..

Hoy en día nos basamos en los SPEC de los años 95 y 2000. A continuación se ofrecen comparativas para los SPECint95, SPECfp95 y los SPECint2000, SPECfp2000 de distintos tipos de procesadores:

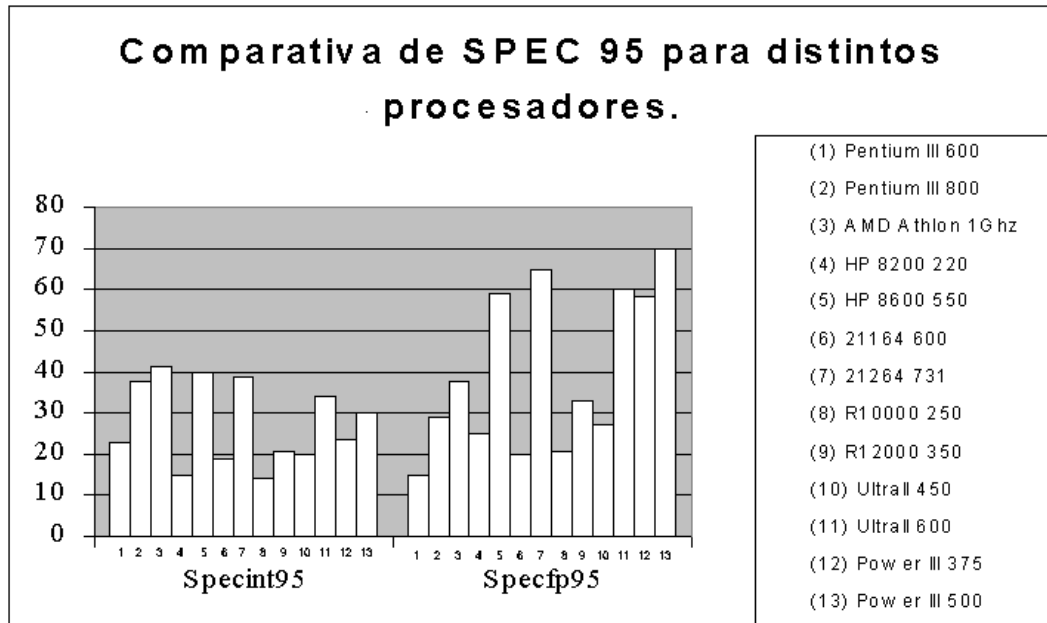


Figura 2.4. Spec95

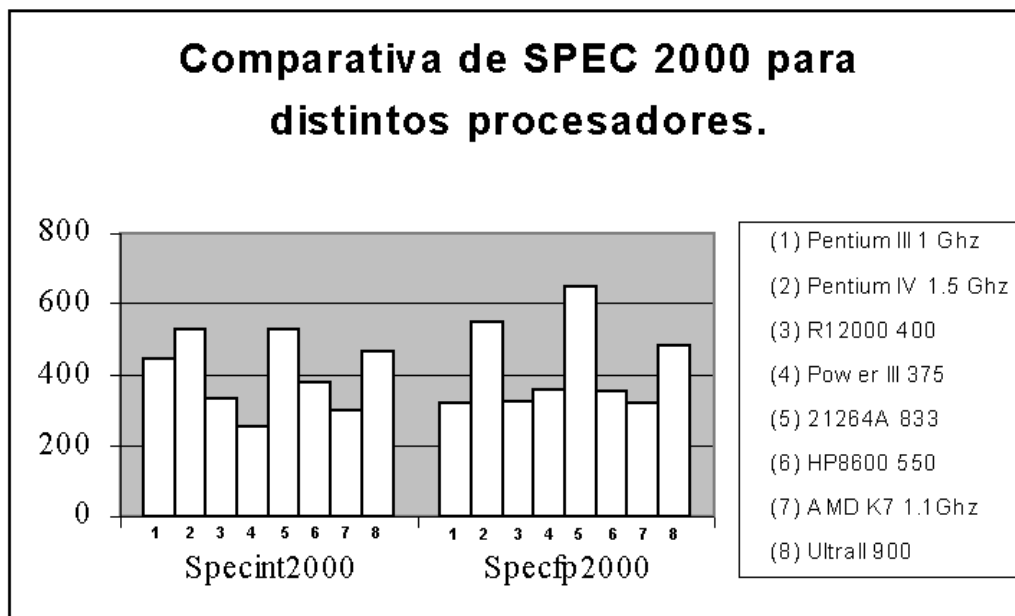


Figura 2.5

2.5.4.2 Otros benchmarks

- **AIM Suite III** : es un Benchmark que mide la eficiencia de sistemas multiusuario, en entornos servidores, UNIX, para atender varios usuarios ejecutando cada uno de ellos un proceso diferente.
- **Xmark** : es un Benchmark de dominio público basado en UNIX y disponible bajo el sistema Xwindow, para evaluar el rendimiento de aplicaciones propias de entornos gráficos y de autoedición.
- **Graphstone** : es una prueba gráfica que evalúa el rendimiento del subsistema de vídeo. El resultado se expresa en número de operaciones ejecutadas por segundo.
- **Khornerstone** : es una medida que prueba tanto el subsistema gráfico, como el rendimiento de la UCP y del disco duro, así como la capacidad para la ejecución de operaciones en coma flotante.
- **SYSmark** : se trata de un benchmark para evaluar CPU's bajo el sistema Windows. Hay diferentes versiones como la SYSmark98 y la SYSmark2002.
- **CPUmark99** : mide el rendimiento de los Pentium bajo Windows.
- **Wintune98** : dedicado a procesamientos de texto y hojas de calculo.
- **Winstone99, Multimediamark99, J.Mark 2.0, etc.**

Pentium® 4 processor	SYSmark 2002
1.5GHz	159
1.6GHz	166
1.7GHz	174
1.8GHz	179
1.9GHz	186
2GHz	193
2AGHz	212
2.2GHz	227
2.4GHz	242

Figura 2.6 Comparativa sobre diferentes frecuencias del Pentium IV del SYSmark2002.

2.5.5- Programas propios de fabricantes

Dentro de este tipo de programas está por ejemplo el índice iCOMP. Se crea en 1992 y proporciona una medición sencilla y relativa del rendimiento del microprocesador. No es un programa de pruebas, sino un conjunto de ellos que se utiliza para calcular un índice de rendimiento relativo que ayude a aquellos que vayan a comprar un PC a decidir qué microprocesador Intel satisface mejor sus necesidades.

Para elaborar el índice iCOMP, Intel toma uno de sus procesadores como referencia y le asigna el índice iCOMP 100. El procesador que se toma como referencia es el 486 SX a 25MHz. Los índices iCOMP de los demás procesadores se obtienen comparándoles con el microprocesador que se toma como referencia.

A continuación se presenta una comparativa el índice iCOMP 3.0, que incluye software y tecnología 3D, multimedia e Internet, para algunos de los procesadores de la familia Intel.

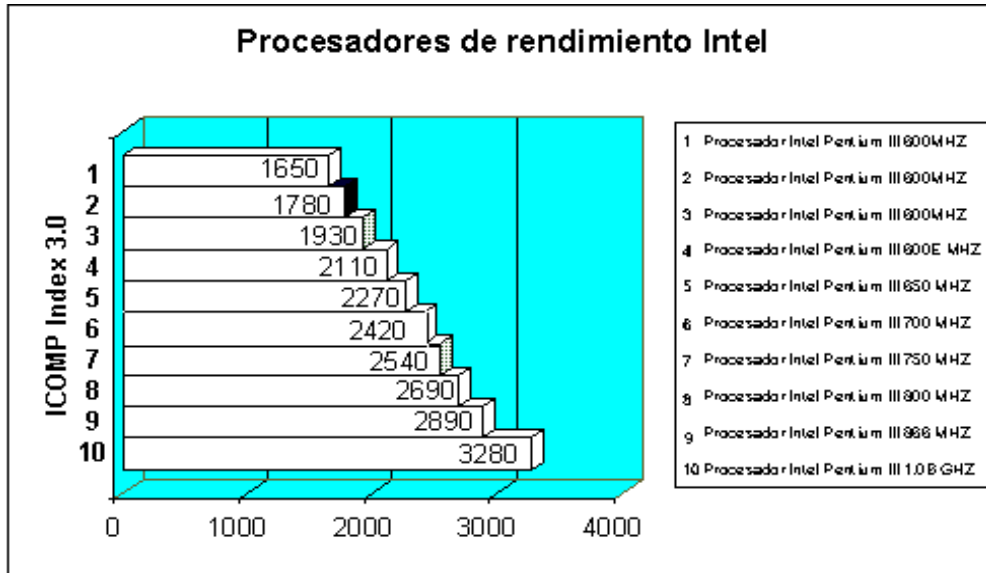


Figura 2.6 El índice iCOMP 3.0 basado en productividad, multimedia, 3D e internet.

2.6- LEYES DE AMDAHL

Las leyes de Amdahl evalúan las modificaciones en el rendimiento de un computador cuando se introducen mejoras o más recursos. El criterio fundamental de Amdahl es que lo que hay que mejorar o modificar siempre es lo que se usa más frecuentemente, ya que es lo que más afecta al rendimiento. En el estudio de las leyes de Amdahl se utilizan dos formulas:

$Aceleración_{\eta} = \frac{\eta_{sist.nuevo}}{\eta_{sist.antiguo}}$	$Ganancia\ Velocidad = \frac{Tiempo_{sist.nuevo}}{Tiempo_{sist.antiguo}}$
--	---

2.6.1- Primera ley de amdahl

La primera ley de Amdahl dice que el aumento del rendimiento debido a la inclusión de una mejora con un nuevo recurso en el sistema está limitado por el tiempo que se utiliza dicha mejora en la ejecución de la tarea.

$$T_{nuevo} = T_{antiguo} * \left(\frac{Fracción\ de\ tiempo\ mejorada}{Ganancia\ Velocidad} + Fracción\ de\ tiempo\ sin\ mejora \right)$$

2.6.2- Segunda ley de amdahl

La segunda ley de Amdahl dice que cuando se introduce una mejora a un computador previamente mejorado, el incremento del rendimiento es menor que si se introduce la mejora sobre el sistema sin mejorar. Dicho de otra forma, la mejora incremental en la aceleración conseguida con la mejora de una parte se va reduciendo a medida que se van introduciendo nuevas mejoras.

EJEMPLO 11

Se cambia la ALU de un procesador dedicado a una tarea en la cual el 50% del tiempo lo hace la ALU. La nueva ALU es dos veces más rápida que la anterior. Calcular el tiempo de mejora.

$$T_{\text{con mejora}} = T_{\text{sin mejora}} \left(\frac{0,5}{2} + 0,5 \right)$$

$$T_{\text{con mejora}} = T_{\text{sin mejora}} * 0,75$$

Se ha mejorado el tiempo en un 75%.

Sustituimos la ALU por una nueva un 30% mayor. Si inicialmente un computador tardaba 2sg. en hacer un programa, ¿cuánto tarda ahora sabiendo que la ALU para este programa está funcionando el 50% del tiempo?

$$T_{\text{nuevo}} = T_{\text{antiguo}} * \left(\frac{\text{Fracción de tiempo mejorada}}{\text{Ganancia Velocidad}} + \text{Fracción de tiempo sin mejora} \right)$$

$$T_{\text{nuevo}} = 2 * \left(\frac{0,5}{1,3} + 0,5 \right) = 1,76 \text{sg.}$$

EJEMPLO 12

Calcular la aceleración del rendimiento de un sistema computador cuando se cambia la ALU del procesador por otra 10 veces más rápida, sabiendo que la ALU participa durante el 40% del tiempo que dura la tarea.

$$\text{Aceleración}_{\eta} = \frac{\eta_{\text{con mejora}}}{\eta_{\text{sin mejora}}} = \frac{T_{\text{sin mejora}}}{T_{\text{con mejora}}}$$

$$T_{\text{con mejora}} = T_{\text{sin mejora}} * (0,4 / 10 + 0,6) = 0,64 * T_{\text{sin mejora}}$$

$$\text{Aceleración}_{\eta} = \frac{T_{\text{sin mejora}}}{0,64 * T_{\text{sin mejora}}} = 1,56$$

Luego el rendimiento aumenta en un 56%.

2.7- MEDIA ARITMÉTICA Y GEOMETRICA

En la mayoría de las evaluaciones de rendimiento de computadores se mide el tiempo que tarda en ejecutar diversos programas y después se calcula la media. Hay dos tipos de media:

- **Media aritmética:** se suman los tiempos de las diversas pruebas y se divide por el número de pruebas.

$$M_{\text{aritmética}} = \frac{\sum t_i}{n}$$

- **Media geométrica:** se calcula la raíz del producto de todos los tiempos.

$$M_{\text{geométrica}} = \sqrt[n]{\prod_{i=1}^n t_i}$$

La media aritmética no interesa en el calculo del rendimiento porque aunque proporciona una medida de lo que tarda el computador en ejecutar los programas, los resultados son contradictorios, por lo que se debe usar la media geométrica.

2.8- RELACION COSTE RENDIMIENTO

Aunque típicamente el rendimiento es un parámetro fundamental por el que se selecciona un computador, en la práctica el coste tiene una importancia decisiva. A la hora de elegir un computador, lo que determina su compra es la relación coste-rendimiento, es decir, el coste por unidad de rendimiento. La tendencia del mercado es construir computadores con mayor rendimiento y menor coste.

2.8.1- Análisis de la distribución de los costes de un computador

Vamos a analizar el coste de los componentes de un computador poniendo como ejemplo Pentium IV:

Placa DFI Pentium IV ATX NB72-SC	94,00€
CPU INTEL PIV 2,2GHZ	228,00€
Ventilador Extra PIV Superball	9,00€
DIMM 128 MB PC133 MHZ SDRAM	24,50€
Tarjeta Gráfica VGA TNT2 32MB AGP	31,50€
Disketera 3 1/2 HD 1,44 MB	9,62€
Caja Semitorre ATX LUJO P IV L2000-300W	38,50€
Monitor Color 15" Digital KFC (3 años de Garantía)	105,00€
Teclado 106 Teclas Win96 Multimedia PS2	5,71€
Tarjeta de sonido SOUND BLASTER 128 Bits PNP	14,90€
MODEM 56KB Interno V90 MOTOROLA	13,25€
CDROM 52x LG IDE OEM	25,50€
Disco Duro 40GB SEAGATE ATA100	63,00€
Altavoces 180 Watios AZUL 3D	4,81€
TOTAL	667,29€

Figura 2.7. Costes de un Pentium IV a 2,2 GHZ.

Porcentualmente:

Placa Base, CPU y Ventilador	50 %
Disketera, Teclado, Modem y Monitor	22 %
Caja semitorre	6 %
Tarjeta de sonido, Tarjeta Gráfica y Altavoces	8 %
Disco Duro y Memoria SDRAM	14 %

Figura 2.8 Análisis porcentual de los costes del computador.

Como conclusión observamos que la mitad del coste del computador recae en la CPU, su placa Base y el ventilador para refrigerarla. En los establecimientos informáticos a los precios de fabrica de los componentes deberemos incluir los costes directos de los distribuidores como son mano de obra, stocks y garantías. Asimismo hay que incluir los márgenes de beneficio que se lleva el distribuidor que están entre el 25 y el 40 % del coste del computador.

2.9- EJERCICIOS PROPUESTOS

EJERCICIO 1

Se propone a un 386 añadirle una memoria caché con una tasa de acierto de un 90%, de forma que, cuando el acceso se haga en la caché, el CPI de las instrucciones que afectan a la memoria se decremента en una unidad. Al poner la caché, en los fallos se pierde un ciclo, es decir, se le suma al CPI 1 unidad. Se pide:

- Calcular el rendimiento del 386 sin caché.
- Calcular el rendimiento del 486 con caché.
- Calcular la relación de rendimientos entre el 386 con caché y el 386 sin caché.

Las diferentes instrucciones tienen la frecuencia y el CPI que se refleja en la tabla:

Nº de instrucciones	Tipo de instrucciones	CPI386
20	Carga	2
10	Almacenar	4
15	Reg/Reg	2
8/7	Salto condicional	9/3
10	Call	9
30	Operaciones Aritméticas	5

EJERCICIO 2

Suponiendo que tenemos 2 máquinas con las siguientes características para un determinado programa R:

- Máquina A : Duración del ciclo de reloj de 23 ns. Con un CPI de 3,2
- Máquina B : Duración del ciclo de reloj de 15 ns. Con un CPI de 4

¿Cuál de las dos máquinas tiene mayor rendimiento para el programa R?

EJERCICIO 3

Estamos interesados en dos implementaciones de una máquina. Una con hardware especial de punto flotante y otra sin él. Considerar un programa P, con la siguiente mezcla de operaciones:

Multiplicación en punto flotante	10%
Suma en punto flotante	15%
División en punto flotante	5%
Instrucciones enteras	70%

La máquina MFP (máquinas con punto flotante), tiene hardware de punto flotante y además puede implementar directamente las operaciones en punto flotante.

Necesita el siguiente número de ciclos para cada clase de instrucción:

Multiplicación en punto flotante	6
Suma en punto flotante	4
División en punto flotante	20
Instrucciones enteras	2

La máquina MNFP (máquina sin puntos flotante) no tiene hardware de punto flotante y por ello debe las operaciones en punto flotante utilizando instrucciones enteras. Todas las instrucciones enteras necesitan dos ciclos de reloj. El número de instrucciones enteras necesarias para implementar cada una de las operaciones en punto flotante es como sigue:

Multiplicación en punto flotante	30
Suma en punto flotante	20
División en punto flotante	50

Ambas máquinas tienen una frecuencia de reloj de 100 MHz. Calcular las frecuencias en MIPS nativos para ambas máquinas.

EJERCICIO 4

Se va a mejorar una máquina y se barajan dos opciones: hacer que la ejecución de las instrucciones de multiplicación tarden 4 veces menos, o que la ejecución de las de acceso a memoria tarde 2 veces menos. Se ejecuta un programa de prueba antes de realizar la mejora y se obtienen las siguientes medidas de tiempo de uso de la CPU: el 20% del tiempo es utilizado para multiplicar, el 50% para instrucciones de acceso a memoria y el 30% restante para otras tareas.

¿Cuál será el incremento de velocidad si sólo se mejora la multiplicación? ¿Y si sólo se mejoran los accesos a memoria? ¿Y si se realizan ambas mejoras?

EJERCICIO 5

Se ejecutan sobre una máquina dos programas A y B utilizados como test para medir su rendimiento. Los recuentos de instrucciones tienen la siguiente distribución en ambos:

	PROGRAMA A	PROGRAMA B
Instrucciones de proceso	37%	48%
Instrucciones de transferencia	45%	36%
Instrucciones de salto	18%	16%

La máquina presenta los siguientes CPI (ciclos por instrucción) medios para cada grupo de instrucciones sin memoria caché de 2º nivel y con ella.

	CPI MEDIO	
	SIN CACHE DE 2º NIVEL	CON CACHE DE 2º NIVEL
Instrucciones de proceso	1.0	1.0
Instrucciones de transferencia	5.2	2.4
Instrucciones de salto	1.1	1.0

Determinar la ganancia de rendimiento (aceleración o speed up) que presenta la mejora de la jerarquía de memoria introducida en la máquina con respecto a la situación sin mejora.

EJERCICIO 6

Una vez graduado, el lector se preguntará cómo llegar a ser un líder en el diseño de computadores. Su estudio sobre la utilización de construcciones de los lenguajes de alto nivel sugiere que las llamadas a los procedimientos son una de las operaciones más caras. Suponga que ha inventado un esquema que reduce las operaciones de carga y almacenamiento normalmente asociadas con las llamadas y vueltas de procedimientos. Lo primero que hace es ejecutar algunos experimentos con y sin esta optimización. Sus experimentos utilizan el mismo compilador optimizador en ambas versiones del computador.

Los experimentos realizados revelan lo siguiente:

- La duración del ciclo de reloj de la versión no optimizada es el 5% más rápido.
- El 30% de las instrucciones de la versión no optimizada son operaciones de carga o almacenamiento.
- La versión optimizada ejecuta 1/3 menos de operaciones de carga y almacenamiento que la versión no optimizada. Para las demás instrucciones, el recuento de ejecución dinámica es inalterable.
- Todas las instrucciones (incluyendo las de carga y almacenamiento) emplean un ciclo de reloj.

¿Qué versión es más rápida? Justificar cuantitativamente la decisión.