

Arquitectura de los Computadores

Laura Monroig, Alejandro Áyelo, Julia de la Cruz, Juan Sáez, Ander Dorado, Daniel Asensi

14 de febrero de 2021

1. FASE-1. Estudio 0

1.1. Introducción

1.1.1. Qué es el rendimiento

El rendimiento es la proporción entre el producto o el resultado obtenido y los medios utilizados [1], es decir que cuanto menor sea el coste que tenga el extraer un resultado de una operación mayor será el rendimiento.

Por lo tanto, si abstraemos la anterior información al mundo que nos incumbe, el de los computadores, podremos afirmar que cuanto mayor sea la calidad de una máquina, en relación con sus componentes y la sinergia de estos con el sistema operativo, menor será el tiempo de ejecución de las aplicaciones que este ejecute, ya que a mayor complicidad entre los mismos más rápido se transmitirá la información y menor será el tiempo de cálculo de las diferentes operaciones, lo que implicará un mayor rendimiento. [2] De hecho, la cantidad de cálculos que es capaz de realizar una computadora ya sea en programas en tiempo de ejecución o en largos procesos, es un indicador de rendimiento globalmente aceptado, por lo que a cuanto mayor tiempo de ejecución haya menor será el rendimiento siguiendo la siguiente formula:

$$Rendimiento = \frac{Prestaciones}{Coste}$$

1.1.2. ¿Por qué es importante el rendimiento?

El rendimiento de un ordenador es una parte especialmente importante, ya que de no ser este suficiente o estar limitado, sería un factor decisivo a la hora de elegir sus componentes o sistema operativo que maneje el computador.

Hemos de recalcar que no todos los equipos tienen la misma función, por lo tanto no todos los equipos necesitarán el mismo rendimiento ni tendrán el mismo gasto energético, por ejemplo: un estudiante no necesitará el mismo rendimiento gráfico y necesitará portabilidad, con respecto a un editor de películas que necesitará un rendimiento bruto elevado para que su trabajo sea eficiente, lo que explica que la elección de sus equipos en cuanto a rendimiento sea diferente, buscando el estudiante una maquina portable con un rendimiento menor para que el gasto energético de esta respecto a su rendimiento sea bajo y aguante toda la jornada de estudio, y buscando el editor un rendimiento y potencia altos para poder desempeñar su trabajo con la mayor fluidez posible.



Figura 1: Pc Estudiante.



Figura 2: Pc Editor.

1.2. Métricas para evaluación del rendimiento.

Una métrica de rendimiento son los **diferentes parámetros que permiten medir de forma cuantitativa el rendimiento de una computadora** o sus componentes, ya sea a nivel Hardware o Software. Por lo general, para determinar el rendimiento de un ordenador nos centramos en el estudio de tres componentes: **El procesador, la memoria principal y dispositivos de E/S.**

Existen 2 modelos de rendimiento:

- **Los modelos analíticos:** Usan fórmulas matemáticas para obtener y comparar parámetros claves en el rendimiento de un ordenador.
- **Los modelos de simulación:** Se encargan de imitar el comportamiento del sistema y observar el tiempo de procesamiento del ordenador ante las simulaciones.

Parámetros que afectan al del rendimiento del procesador en relación al tiempo:

Frecuencia de la CPU (fCPU): Es el número de ciclos por segundos al que trabaja el computador.

$$f_{cpu} = \frac{n \text{ ciclos}}{\text{segundos}}$$

Periodo de la CPU (TCPU): Tiempo que dura un ciclo, inversa de la frecuencia de la CPU.

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

Ciclos por instrucción (CPI): La CPI es el promedio de ciclos de reloj que se tarda en ejecutar una instrucción.

$$CPI = \frac{\text{ciclos de reloj de CPU para un programa}}{\text{Recuento de instrucciones}}$$

Detallando el CPI por cada tipo de instrucción estática i:

$$CPI = \frac{\sum_{i=1}^n Nro.Instruc_i * CPI_i}{Nro.InstrucTot}$$

- **Ii**= instrucciones dinámicas para cada tipo de instrucción estática i.
- **CPIi**= Número medio de ciclos de reloj para la instrucción tipo i.

Número de instrucciones del programa: Si hay más instrucciones, más tiempo tardará en ejecutar, por lo que baja el rendimiento.

Multitarea: Capacidad que tiene un computador de realizar varias tareas al mismo tiempo.

$$n_{prog} = \frac{1}{T_{prog}}$$

Tiempo de ejecución: Es el tiempo que tarda el computador en realizar una tarea (medido en segundos).

$$\frac{\text{Tiempo de CPU usuario} + \text{Tiempo CPU sistema}}{\text{tiempo transcurrido en segundos}}$$

Rendimiento de la CPU: Se mide dividiendo los ciclo de reloj de la CPU entre la frecuencia que tenga dicho reloj.

$$\text{Tiempo de CPU} = \frac{\text{ciclos de reloj de la CPU para un programa}}{\text{frecuencia del reloj}}$$

Alternativas para la medida del rendimiento (alternativa como medida al tiempo):

MIPS : Son los millones de instrucciones por segundos que ejecuta un procesador para un programa. Puede considerarse como la inversa del tiempo de ejecución.

$$\text{MIPS} = \frac{\text{recuento de instrucciones}}{\text{tiempo de ejecución} * 10^6} = \frac{\text{frecuencia de reloj}}{\text{CPU} * 10^6}$$

Así mismo, la relación de la variable y el programa con el tiempo es:

$$\text{Tiempo de ejecucion} = \frac{\text{recuento de instrucciones}}{\text{MIPS} * 10^6}$$

MFLOPS: Se utiliza para medir el rendimiento de un procesador en millones de instrucciones en coma flotante por segundo.

$$\text{MFLOPS} = \frac{\text{numero de operaciones de tipo flotante}}{\text{tiempo de ejecucion} * 10^6}$$

Instrucciones por ciclo (IPC): Cantidad de instrucciones ejecutadas en un ciclo de reloj. El rendimiento del CPU depende de la frecuencia, de la cantidad de ciclos de reloj y del recuento de instrucciones.

$$\text{IPC} = \frac{N * f}{T}$$

[3] [4]

1.3. Programas de prueba.

Benchmark → En términos informáticos, un benchmark es una aplicación destinada a medir el rendimiento de un dispositivo o algún elemento del mismo.

Para ello, se pone a prueba el equipo con tareas muy exigentes y variadas, con la intención de medir cómo de bien las cumple. De esta forma, se puede estimar bajo qué tareas o estímulos un determinado smartphone, tablet u ordenador se comporte de una manera fiable y efectiva. Cabe destacar que generalmente hay dos tipos de pruebas las de sistema (las que se realizan a todo el dispositivo) y las de componentes. Su funcionamiento consiste en realizar una prueba de estrés a determinados componentes mediante la ejecución de imágenes 3D. [5] Algunos objetivos de los benchmark son:

- Comprobar si las cualidades de algunos componentes están dentro de los parámetros normales de los mismos.
- Maximizar el rendimiento.
- Obtener la mejor relación coste-beneficio.

1. **Sintéticos** → Son más específicos, están **especialmente diseñados para medir el rendimiento de un componente individual** de un ordenador, normalmente llevando el mismo escogido a su máxima capacidad. Este tipo de programas **pueden probar diferentes subsistemas de nuestro ordenador de forma separada**, con otros trozos de código o benchmarks específicos para realizar cierto tipo de prueba. Ejemplos: Cinebench, 3DMark, SuperPi...
 - **Cinebench** → Cinebench es una herramienta muy poderosa para evaluar el rendimiento de un PC y compararlo con otros equipos. Una vez ejecutadas las pruebas, los resultados de Cinebench se muestran utilizando el propio sistema de puntuación de la herramienta. [5]
 - **Dhrystone** → Es un pequeño benchmark sintético que pretende ser representativo de programación entera de sistemas. Está basado en estadísticas publicadas sobre uso de particularidades de los lenguajes de programación, sistemas operativos, compiladores, editores, etc. [6]
 - **Whetstone** → Se lo considera uno de los padres de los benchmarks sintéticos por ser el primero diseñado específicamente con ese fin. Su código fuente es corto y relativamente fácil de entender, con una estructura limpia y que se explica por sí misma. El tiempo de ejecución es corto: 100 segundos (por diseño). Es muy preciso (tiene pequeñas variaciones en los resultados). [7]
 - **SuperPi** → Super PI es un popular software de benchmark que calcula pi con un número específico de dígitos después del punto decimal, hasta un máximo de 32 millones. Mide el tiempo que tarda un procesador en calcular una determinada cantidad de dígitos. [8]
2. **De aplicaciones** → Utilizan una serie de herramientas basadas en apps reales, que simulan una carga de trabajo para medir el comportamiento global del equipo. [9]
3. **De bajo nivel** → Miden directamente el rendimiento de los componentes. Ejemplo: el reloj de la CPU, los tiempos de la DRAM y de la caché SRAM, tiempo de acceso medio al disco duro, latencia, tiempo de cambio de pista, etc. Ejemplos: El reloj de la CPU, los tiempos de la DRAM y de la caché SRAM, tiempo de acceso medio al disco duro, latencia etc. [10]
4. **De alto Nivel** → Están más enfocados a medir el rendimiento de la combinación componente/controlador/SO de un aspecto específico del sistema, como por ejemplo el rendimiento de E/S con ficheros, o el rendimiento de una determinada combinación de componentes/controlador/SO/aplicación. [11]
5. **Otros tipos** → Consumo de energía, Disipación de calor, Reducción de ruido... Otros tipos de programas para medir el rendimiento del equipo o componentes son por Ejemplo: Linpack, SPE, Ciusbet...
 - **SPEC (Standard Performance Evaluation Corporation)** [12] → Se creó sin ánimo de lucro y entre marcas, universidades ... con el objetivo de definir unos benchmark que consistieran en programas reales y que pudieran ser utilizados como referentes o estándares para todas las marcas. Los benchmark SPEC sirven para medir el rendimiento de componentes concretos o de los microprocesadores en genera. Estos han ido mejorando y evolucionando con el tipo como se muestra en la siguiente tabla:

FAMILIA SPEC	CON NÚMEROS ENTEROS	CON NÚMEROS REALES	LÍMITES DE COMPILACIÓN
SPEC89	10 programas de prueba		No
SPEC92	SPECint92: 6 programas	SPECfp92: 14 programas	No
SPEC95	SPECint95: 10 programas	SPECfp95: 10 programas	2 Opciones: 1- La mejor posible 2- La misma para todos
SPEC2000	CINT 2000: 12 programas	CFP 2000: 14 programas	2 Opciones: <u>baseline</u> y libre
SPEC2006	CINT 2006: 11 programas	CFP 2006: 17 programas	2 Opciones: <u>baseline</u> y libre

Figura 3: Evolución SPEC

1.4. Presentación de los resultados.

Para presentar los resultados queremos resumir el rendimiento de una computadora, la forma más fácil es usar un solo número que se pueda comparar con otras máquinas. Por lo general, esto implica ejecutar pruebas

en la máquina y realizar algún tipo de media. Existen varias formas diferentes de definir un valor medio, entre ellos **la media aritmética, la media geométrica y la media armónica.**

En el sentido matemático, la media geométrica de un conjunto de n valores es la longitud de un lado de un cubo de n dimensiones que tiene el mismo volumen que un rectángulo de n dimensiones cuyos lados están dados por los n valores. Como esto no es ni intuitivo ni informativo, la utilidad de usar la media geométrica para cualquier cosa es cuestionable. Su única ventaja aparente es que **no se ve afectado por la normalización: si normaliza primero por un conjunto de pesos o por la media geométrica del pesos después, el resultado es el mismo.**

Esta propiedad se ha utilizado para sugerir que la media geométrica es superior, ya que produce los mismos resultados cuando comparar varios equipos independientemente de los tiempos de los equipos que se utilicen como factor de normalización [13]. Sin embargo, el argumento fue refutado en [14], donde se ilustró por primera vez la **falta de significado de la media geométrica.**[15]

- **Media aritmética.** Problema cuando se promedian velocidades en lugar de tiempos. Cuando se quiere caracterizar el comportamiento de un computador que funciona con 2 o más modos:

P_1 ejecuta 10,000 instrucciones en 5 segundos $\Rightarrow r_1 = 2,000$ instrucciones/segundo

P_2 ejecuta 10,000 instrucciones en 2 segundos $\Rightarrow r_1 = 5,000$ instrucciones/segundo

La velocidad de ejecución mediante la media aritmética es $R_a = \frac{(2,000 + 5,000)}{2} = 3,500$ instrucciones

Si con esta media calculamos las instrucciones ejecutadas en 7 segundos:

$$3,500 * 7 = 24,500 \text{ instrucciones}$$

[16]

- **Media geométrica:** La media geométrica es consistente y no importa la máquina que se tome como referencia. [17]
 - **Ventajas:** La media geométrica es consistente (no importa la máquina que se tome como referencia). Es independiente de la frecuencia de ejecución (w_i) de cada programa en el workload.
 - **Desventajas:** Muchas veces es necesario contemplar la frecuencia w_i de cada programa dentro del workload. No siempre es consistente respecto a los tiempos totales de ejecución.
- **Media armónica:** Dada una cantidad finita de números es igual al recíproco, o inverso, de la media aritmética de los recíprocos de dichos valores y es recomendada para promediar velocidades.
 - **Ventajas:** Considera todos los valores de la distribución y en ciertos casos, es más representativa que la media aritmética.
 - **Desventaja:** La influencia de los valores pequeños y el hecho de que no pueda ser determinada en distribuciones con valores iguales a cero; por eso su empleo no es aconsejable en distribuciones donde existan valores muy pequeños.[18]

Ejemplo de la presentación de los resultados de benchmark:

Hardware	Software
CPU: IBM Power3-II CPU MHz: 375 FPU: Integrated CPU(s) enabled: 1 core, 1 chip, 1 core/chip CPU(s) orderable: 1,2,3,4 Parallel: No Primary Cache: 32KBI+64KBD (on chip) Secondary Cache: 4MB (I+D) (off chip) L3 Cache: None Other Cache: None Memory: 4GB Disk Subsystem: 1x9.1GB IBM Ultra-2 SCSI, 10000RPM Other Hardware: None	Operating System: AIX 4.3.3 + beta of APAR IY12051 Compiler: Fortran 77 and 90: IBM xl Fortran 6.1 invoked as 'xlf90' C: IBM VAC 5.0 invoked as 'xlc' File System: AIX/JFS System State: Multi-user
Notes/Tuning Information	
Portability Flags -qfixed used in: wupwise, swim, mgrid, applu, galgel, sixtrack, apsi -qsuffix=f-f90 used in: galgel, facerec, lucas, fma3d NOTE: The flag combination 'qpdl/qpdl2' indicates a 2 step compilation process, the exact process is as follows: fdo pre0 = rm -rf \${PDFDIR}; mkdir -p \${PDFDIR} PASS1_FLAGS =-qpdl {optimization flags} PASS1_LD_FLAGS =-l{libraries} -L[pdf library location] -lpdf PASS2_FLAGS =-qpdl2 {optimization flags} PASS2_LD_FLAGS =-l{libraries} Base Optimization Flags: Fortran: -qpdl/pdl2 -O3 -qarch=pwr3 -lmass C: -O5 -lmass Peak Optimization Flags: 168.wupwise: -qpdl/pdl2 -O4 -qipa=level=2 -lmass 171.swim: -qpdl/pdl2 -O4 -qipa=level=2 -qhot -lmass 172.mgrid: -O3 -qarch=pwr3 -qhot	
Standard Performance Evaluation Corporation	

Figura 4: Listado de software y hardware. [19]

Class of Instances	#	OpenSAT		Chaff	Ratio	
		RefImpl	JQuest2		RefImpl	JQuest2
		Time X	Time X	Time X	RefImpl	JQuest2
sss-sat-1.0	100	72015.77 45	929.73 0	301.09 0	293.18	3.09
vliw-sat-1.0	100	115678.68 76	14050.59 0	6271.24 0	18.44	2.24
vliw-sat-1.1	100	116394.43 76	12648.07 0	6588.11 0	17.67	1.42
fvp-unsat-1.0	4	4508.79 3	1071.16 0	731.12 0	6.17	1.47
bounded model checking	34	18417.69 10	8954.27 0	3383.24 0	5.44	2.64
equivalence checking	25	23176.40 14	4953.91 2	3636.46 2	6.37	1.36
beijing	16	3788.14 2	3453.96 2	3150.05 2	1.20	1.10

Figura 5: Resultados benchmark. [20]

2. Referencias.

Referencias

- [1] Real Academia de la Lengua Española. *Definición Rae*. <https://dle.rae.es/rendimiento>.
- [2] Facultad Informática UA. *Tipos rendimiento*. <https://cvnet.cpd.ua.es/uaMatDocente/Materiales/MaterialesAlumno>.
- [3] Facultad Informática UA. *Métricas del Rendimiento*. <https://cvnet.cpd.ua.es/uaMatDocente/Materiales/MaterialesAlumno>.
- [4] karmencliment. *Métricas para la evaluación del Rendimiento*. <https://cutt.ly/dkEgagx>.
- [5] MAXON. *Cinebench*. <https://www.maxon.net/es/cinebench>.
- [6] Wikiwand. *Benchmark*. <https://www.wikiwand.com/es/Dhrystone>.
- [7] Wikiwand. *Whetstone*. [https://www.wikiwand.com/es/Whetstone_\(benchmark\)](https://www.wikiwand.com/es/Whetstone_(benchmark)).
- [8] TechPowerPi. *SuperPi*. <https://www.techpowerup.com/download/super-pi/>.
- [9] EcuRed. *De aplicaciones*. https://www.ecured.cu/Pruebas_Benchmark#Otros_tipos_de_Benchmark.
- [10] EcuRed. *Bajo Nivel*. https://www.ecured.cu/Pruebas_Benchmark#Otros_tipos_de_Benchmark.
- [11] Wikiwand. *Alto Nivel*. [https://www.wikiwand.com/es/Whetstone_\(benchmark\)](https://www.wikiwand.com/es/Whetstone_(benchmark)).
- [12] Universidad Europea de Madrid. *SPEC*. https://www.cartagena99.com/recursos/alumnos/apuntes/ININF1_M10_U1_T3.pdf.
- [13] Philip J. Fleming y John J. Wallace. “How Not to Lie with Statistics: The Correct Way to Summarize Benchmark Results”. En: *Commun. ACM* 29.3 (mar. de 1986), 218–221. ISSN: 0001-0782. DOI: 10.1145/5666.5673. URL: <https://doi.org/10.1145/5666.5673>.
- [14] J. E. Smith. “Characterizing Computer Performance with a Single Number”. En: *Commun. ACM* 31.10 (oct. de 1988), págs. 1202-1206. ISSN: 0001-0782. DOI: 10.1145/63039.63043.
- [15] Bruce Jacob y Trevor Mudge. *Notes on Calculating Computer Performance*. <https://www.eecs.umich.edu/techreports/cse/95/CSE-TR-231-95.pdf>.
- [16] *Tema 705Benchmark – UNNE*. http://exa.unne.edu.ar/informatica/evalua_ant/TEMA_07_05_BENCHMARK.PDF.
- [17] Iñigo Arce. *Evaluación de Performance mediante Benchmarks*. <https://slideplayer.es/slide/1675941/>.
- [18] Wikipedia. *Media armónica — Wikipedia, La enciclopedia libre*. [Internet; descargado 11-febrero-2021]. 2020. URL: [\url{https://es.wikipedia.org/w/index.php?title=Media_arm%C3%B3nica&oldid=129487542}](https://es.wikipedia.org/w/index.php?title=Media_arm%C3%B3nica&oldid=129487542).
- [19] <https://player.slideplayer.es/7/1675941/data/images/img4.jpg>. <https://player.slideplayer.es/7/1675941/data/images/img4.jpg>.
- [20] Daniel Le Berre. https://www.researchgate.net/figure/Results-on-benchmark-examples_tbl12833521. https://www.researchgate.net/figure/Results-on-benchmark-examples_tbl12833521.