

COMPARACIÓN DEL RENDIMIENTO DE LOS COMPUTADORES: *BENCHMARKING*

Usuarios, administradores y diseñadores

¿Qué sistema informático
tiene mejor rendimiento para una
carga determinada?



1. MEDIDA Y RENDIMIENTO

Medidas de rendimiento



MEDIDAS DE RENDIMIENTO

Existen múltiples magnitudes para medir el rendimiento de un sistema informático, sin embargo, **el tiempo que necesita para llevar a cabo una cierta tarea es la más justa e intuitiva de las medidas.**

Otra métrica comúnmente aceptada para describir el rendimiento de un sistema informático es la **productividad**, la cual se define como **la cantidad de trabajo que ha completado un sistema por unidad de tiempo.**

- Por lo tanto, aquel sistema que tenga una productividad más alta será el que disponga de un mayor rendimiento.

MEDIDAS DE RENDIMIENTO

Por otra parte, es necesario tener en cuenta que un sistema informático está compuesto de una serie de **dispositivos**, tales como la CPU, la memoria principal, el sistema de E/S y las redes de comunicación, entre otros.

Tradicionalmente, la **CPU** se ha venido considerado como la unidad funcional que representa de manera intrínseca el rendimiento del sistema completo.

- Tanto es así que, en numerosos textos sobre arquitectura de computadores, se suele equiparar el rendimiento del procesador con el rendimiento del computador en su conjunto.

MEDIDAS DE RENDIMIENTO

Sin embargo, cada uno de los **dispositivos** dispone de sus propias **métricas de rendimiento**, y cada uno de ellos contribuye al desempeño del sistema en su totalidad.

Dispositivo	Medida
CPU	MIPS GFLOPS CPI Tiempo de ejecución
Memoria principal	Lecturas/s Escrituras/s Tiempo de espera Tiempo de lectura Tiempo de escritura
E/S	Lecturas/s Escrituras/s Tiempo de espera Tiempo de lectura Tiempo de escritura
Red de comunicación	Paquetes enviados/s Latencia (Tiempo de respuesta) Retraso (Delay) Fluctuación (Jitter)

ATRIBUTOS DE LA CALIDAD DE MEDIDA

- **Fácil de medir:** cuanto mayor sea la facilidad de medir, mayor será el uso de la medida.
- **Repetible:** si se realizan múltiples medidas, usando el mismo proceso de medición, se obtiene el mismo valor medido.
- **Confiable:** si la medida produce resultados seguros, con respecto a alguna propiedad, de tal forma que su resultado puede ordenar los sistemas medidos.
- **Lineal:** si los valores de mediciones son linealmente proporcionales a los grados de la propiedad que los sistemas exhiben.
- **Consistente:** si tiene las mismas unidades y la misma precisión entre diferentes sistemas medidos.
- **Independiente:** si no está influenciada por intereses espurios o particulares.

EL PROCESADOR Y SU RELOJ

La **frecuencia del reloj** de un procesador se refiere a la frecuencia en la que el reloj de la CPU funciona y sirve como un indicador de la velocidad del procesador.

Normalmente **se mide en ciclos de reloj por segundo** o su equivalente Hertzios (Hz) y un GHz son 10^9 Hz.

Desde el punto de vista de las características de calidad de una medida, se puede decir que la frecuencia de reloj es **repetible** (constante para un procesador dado), **fácil** de medir (gracias a las especificaciones del fabricante), **consistente e independiente**.

- Veamos el resto de características...

EL PROCESADOR Y SU RELOJ

El hecho de que los sistemas informáticos estén contruidos usando como base un reloj que funciona con un ratio constante, **hace que el procesador, o CPU**, tenga una gran repercusión a la hora de evaluar un sistema informático.

Ese reloj funciona con eventos discretos llamados ciclos de reloj o periodos de reloj.

Los diseñadores de computadores se refieren al **tiempo de reloj como la duración de su periodo**, por ejemplo 1 ns, o bien a su frecuencia, por ejemplo 1 GHz.

EL PROCESADOR Y SU RELOJ

Por lo tanto, el tiempo de CPU para un programa i determinado puede ser expresado como:

$$\text{Tiempo de CPU} = \text{Ciclos de CPU}_{\text{programa}_i} \cdot \text{Tiempo de ciclo de reloj}$$

O equivalentemente:

$$\text{Tiempo de CPU} = \frac{\text{Ciclos de CPU}_{\text{programa}_i}}{\text{Frecuencia del reloj}}$$

RELACIÓN ENTRE TIEMPO Y FRECUENCIA

Ejemplo de cómputo

- Procesador con un reloj a 500 MHz o 0,5 GHz
- ¿Cuál es la duración del tiempo de ciclo del reloj?

$$\text{Frecuencia : } 500 \text{ MHz} = 500 \times 10^6 \text{ Hz} = 500 \times 10^6 \text{ s}^{-1}$$

$$\text{Ciclo de reloj : } \frac{1}{500 \text{ MHz}} = \frac{1}{500 \times 10^6 \text{ Hz}} = 2 \times 10^{-9} \text{ s} = 2 \text{ ns}$$

CICLOS POR INSTRUCCIÓN DE UN PROGRAMA

Además del número de ciclos de reloj que necesita un programa para ser ejecutado, **podemos contar el número de instrucciones de bajo nivel que necesita ese programa para ser ejecutado.**

Si conocemos además el número de ciclos de reloj, es posible calcular el número medio de ciclos de reloj por instrucción (**CPI**, *cycles per instruction*).

$$CPI = \frac{\text{Ciclos de CPU}_{\text{programa}_i}}{\text{Numero de instrucciones}}$$

CÁLCULO DEL CPI DE UN PROGRAMA

Cada instrucción necesita un determinado número de ciclos, y por tanto, el valor de CPI depende de las instrucciones ejecutadas por cada programa

▪ Instrucciones de escritura (<i>store</i>):	12%	2 ciclos
▪ Instrucciones de lectura (<i>load</i>):	21%	2 ciclos
▪ Instrucciones de la ALU:	43%	1 ciclo
▪ Instrucciones de salto:	24%	2 ciclos

$$CPI = \sum_{i=1}^4 t_i \times f_i = 1,57$$

RENDIMIENTO DEL PROCESADOR

El **rendimiento del procesador depende de tres características**: el tiempo de ciclo de reloj (*clock time cycle*), los ciclos de reloj por instrucción (*clock cycles per instruction*), y el número de instrucciones.

$$\textit{Tiempo de CPU} = \textit{Numero de instrucciones} \cdot \textit{CPI} \cdot \textit{Tiempo ciclo reloj}$$

Es difícil cambiar o mejorar uno de estos tres parámetros de forma aislada, debido a la interdependencia de la tecnología involucrada en el proceso de mejora.

- **El tiempo de ciclo de reloj depende de la tecnología hardware y su organización, el CPI depende del repertorio de instrucciones de la arquitectura y el número de instrucciones depende del compilador.**

Operando nos quedaría:

$$\textit{Numero de instrucciones} / \textit{Tiempo de CPU} = \textit{Frecuencia reloj} / \textit{CPI}$$

RENDIMIENTO DEL PROCESADOR

La asociación errónea pero muy frecuente, es la de asociar un buen rendimiento con la frecuencia del procesador.

- Sin embargo, el hecho de obviar cantidad de detalles tecnológicos, no solo de la CPU, sino del sistema en conjunto, hace que dicha asociación sea errónea.
- Aunque, comparar computadores que tengan frecuencias muy diferentes, el rendimiento será diferente también, evidentemente.

2. ÍNDICES CLÁSICOS DE RENDIMIENTO

Índices de Arquitectura de Computadores
MIPS, MFLOPS

MIPS

MIPS (*million of instructions per second*)

- Se denominan MIPS nativos

$$MIPS = \frac{\text{Instrucciones ejecutadas}}{\text{Tiempo de ejecución} \times 10^6} = \frac{\text{Frecuencia de reloj de la CPU}}{CPI \times 10^6}$$

- **Depende del juego de instrucciones y los MIPS medidos varían entre programas en el mismo computador.**
- ***“Meaningless indicator of processor speed”***

MIPS relativos: referidos a una máquina de referencia (proceso de normalización)

$$MIPS_{\text{relativos}} = \left(\frac{\text{Tiempo de referencia}}{\text{Tiempo de ejecución}} \right) \times MIPS_{\text{referencia}}$$

EJEMPLO CON MIPS

El programa P contiene 200×10^6 instrucciones

- Procesador 1 (175 MHz): ejecuta P en 10 s
- Procesador 2 (300 MHz): ejecuta P en 5 s

$$MIPS_1 = \frac{\text{Instrucciones ejecutadas}}{\text{Tiempo de ejecución} \times 10^6} = \frac{200 \times 10^6}{10 \times 10^6} = 20$$

$$MIPS_2 = \frac{\text{Instrucciones ejecutadas}}{\text{Tiempo de ejecución} \times 10^6} = \frac{200 \times 10^6}{5 \times 10^6} = 40$$

$$CPI_1 = \frac{\text{Ciclos de reloj de CPU usados}}{\text{Instrucciones ejecutadas}} = \frac{10 \times 175 \times 10^6}{200 \times 10^6} = 8,75$$

$$CPI_2 = \frac{\text{Ciclos de reloj de CPU usados}}{\text{Instrucciones ejecutadas}} = \frac{5 \times 300 \times 10^6}{200 \times 10^6} = 7,5$$

MIPS

Los MIPS son una frecuencia fácil de medir, repetible e independiente.

Sin embargo no es lineal.

- Por ejemplo, un computador con el doble de MIPS no rinde necesariamente el doble.

Tampoco es confiable ni consistente.

- Aunque los MIPS han sido empleados tradicionalmente como índice para comparar las prestaciones con un mismo juego de instrucciones, presenta varios inconvenientes:
 - No tiene en cuenta el tipo de instrucciones, que puede variar según el programa ejecutado en un mismo computador.
 - El que computador obtenga más MIPS que otro no indica necesariamente que proporcione un tiempo de ejecución menor.

MFLOPS

MFLOPS (*million of floating-point operations per second*)

- Se denominan MFLOPS nativos
- Basado en operaciones y no en instrucciones
 - El tiempo de ejecución de la fórmula es el del programa, incluyendo el tiempo consumido por las instrucciones de enteros

$$MFLOPS = \frac{\text{Operaciones de coma flotante ejecutadas}}{\text{Tiempo de ejecución} \times 10^6}$$

MFLOPS

Los MFLOPS son **fáciles de medir y son una medida repetible**, pero no cumple ninguno de los otros cuatro atributos.

- Diferentes fabricantes cuentan las operaciones de coma flotante de distinta forma.
- El conjunto de operaciones de coma flotante suele variar de una arquitectura de procesador a otra (ocurre lo mismo con los MIPS)
- No es lo mismo, desde un punto de vista estrictamente temporal, hacer una suma que una división en coma flotante.
 - Estos problemas se resuelven parcialmente con el establecimiento de los denominados MFLOPS normalizados, que permiten ponderar de manera más justa la complejidad de cada operación en coma flotante.

MFLOPS NORMALIZADOS

Consideran la complejidad de las operaciones en coma flotante

- Suma, resta, multiplicación, comparación, negación: poco costosas
- División, raíz cuadrada: costosas
- Trigonómicas: muy costosas

Ejemplo de normalización de operaciones en coma flotante

- ADD, SUB, COMPARE, MULT \Rightarrow 1 operación normalizada
- DIVIDE, SQRT \Rightarrow 4 operaciones normalizadas
- EXP, SIN, ATAN, ... \Rightarrow 8 operaciones normalizadas

CÁLCULO DE LOS MFLOPS DE UN PROGRAMA

Programa Spice: el computador tarda en 94 segundos en ejecutarlo.

- Contiene 109.970.178 operaciones en coma flotante de las cuales:
 - 15.682.333 son divisiones (DIVD)
 - El resto tiene una complejidad similar a la de la suma

$$MFLOPS_{nativos} = \frac{109.970.178}{94 \times 10^6} = 1,2$$

$$MFLOPS_{normalizados} = \frac{94.287.845 + 15.682.333 \times 4}{94 \times 10^6} = 1,7$$

ATRIBUTOS DE MEDIDAS DE RENDIMIENTO DE PROCESADOR

- **Fácil de medir:** cuanto mayor sea la facilidad de medir, mayor será el uso de la medida.
- **Repetible:** si se realizan múltiples medidas, usando el mismo proceso de medición, se obtiene el mismo valor medido.
- **Confiable:** si la medida produce resultados seguros con respecto a alguna propiedad de tal forma que su resultado puede ordenar los sistemas medidos.
- **Lineal:** si los valores de mediciones son linealmente proporcionales a los grados de la propiedad que los sistemas exhiben.
- **Consistente:** si tiene las mismas unidades y la misma precisión entre diferentes sistemas medidos.
- **Independiente:** si no está influenciada por intereses espurios o particulares.

	Facilidad	Repetitividad	Confiability	Linealidad	Consistencia	Independencia
GHz						
MIPS						
MFLOPS						



3. RESUMEN DE RENDIMIENTOS: APROXIMACIONES

Uso de medias
Normalización de rendimientos



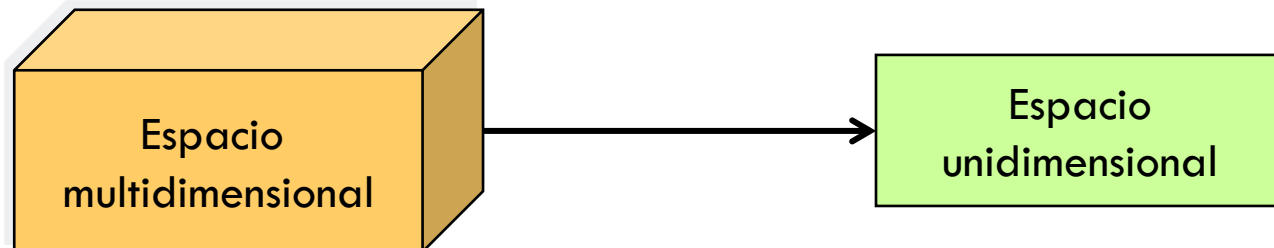
¿CÓMO EXPRESAR EL RENDIMIENTO?

El rendimiento es una variable multidimensional

- Tendría que expresarse mediante múltiples índices
- Sin embargo, las comparaciones son más sencillas si se usa un único índice de rendimiento (a minimizar o maximizar)

¿Cómo concentrar todas las variables en una sola?

- Utilizar la *mejor* variable que represente el rendimiento
- Asegurar que su obtención es válida
- Método habitual de síntesis: uso de medias
 - Hay que tener especial cuidado con la normalización de valores



PUNTOS DE PARTIDA

Está aceptado universalmente que

- El tiempo necesario para llevar a término un determinado cómputo refleja la esencia del rendimiento de un computador

Dos propiedades indispensables

- Si resumimos el rendimiento de un computador para un conjunto de ejecuciones de programas en una única cantidad expresada en **unidades de tiempo**, entonces ha de ser **directamente proporcional** al tiempo total (ponderado) consumido por los programas.
- Si resumimos el rendimiento de un computador para un conjunto de ejecuciones de programas en una única cantidad expresada en **una frecuencia**, entonces ha de ser **inversamente proporcional** al tiempo total (ponderado) consumido por los programas.

LA MEDIA ARITMÉTICA

Útil para tiempos de respuesta



$$x_a = \frac{1}{n} \sum_{i=1}^n x_i$$

$$x_a = \sum_{i=1}^n w_i \times x_i, \text{ con } \sum_{i=1}^n w_i = 1$$

No se ha de utilizar con frecuencias

Recomendación: normalizar el resultado final (no cada x_i)

LA MEDIA ARMÓNICA

Útil cuando los valores son frecuencias con unidades de tiempo en el denominador (MIPS, MFLOPS)

$$x_h = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$$

$$x_h = \frac{1}{\sum_{i=1}^n \frac{w_i}{x_i}}, \text{ con } \sum_{i=1}^n w_i = 1$$

No se ha de utilizar con tiempos de respuesta

Recomendación: normalizar el resultado final (no cada x_i)

LA MEDIA GEOMÉTRICA

No es útil ni para tiempos de respuesta ni para frecuencias

$$x_g = \left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}}$$

$$x_g = \prod_{i=1}^n x_i^{w_i}, \text{ con } \sum_{i=1}^n w_i = 1$$

Única virtud: mantiene el mismo orden en las comparaciones con valores normalizados (consistencia)

- Usado en los benchmarks de SPEC y BAPCO

Sin embargo, este orden puede no ser el correcto

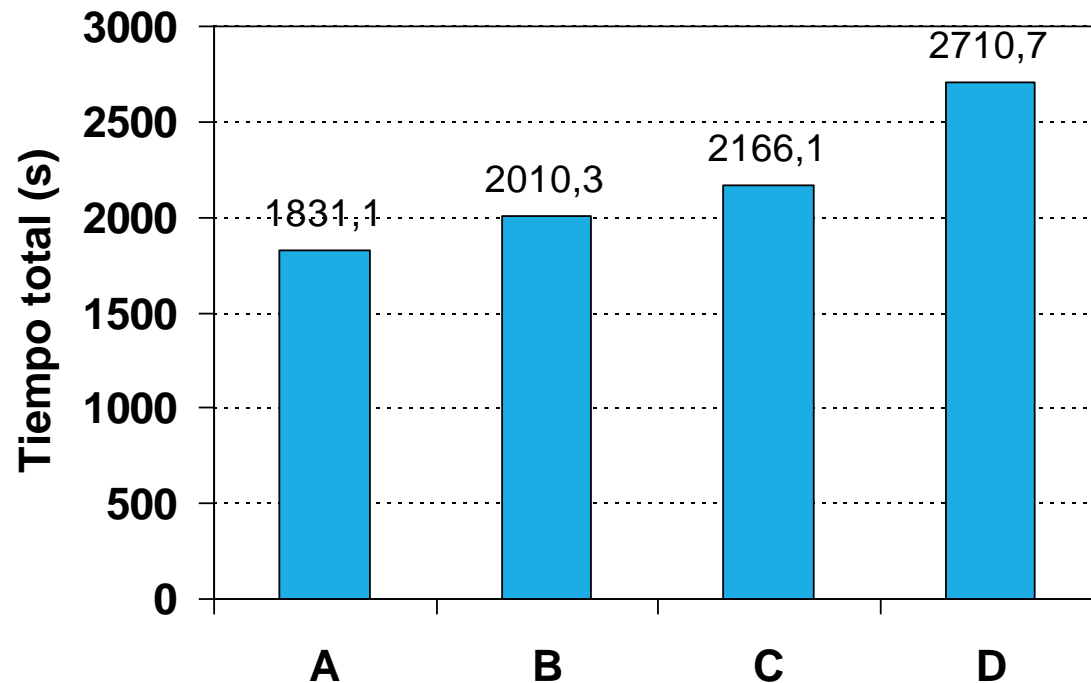
EJEMPLO DE COMPARACIÓN CON TIEMPOS

Programa	R	A	B	C	D
1	1400	141	170	136	134
2	1400	154	166	215	301
3	1100	96.8	94.2	146	201
4	1800	271	283	428	523
5	1000	83.8	90.1	77.4	81.2
6	1800	179	189	199	245
7	1300	112	131	87.7	75.5
8	1800	151	158	138	142
9	1100	93.5	122	104	118
10	1900	133	173	118	142
11	1500	173	170	179	240
12	3000	243	264	338	508
Suma	19100	1831.1	2010.3	2166.1	2710.7

COMPARACIÓN CON EL TIEMPO TOTAL

Ordenación con el **tiempo total**:

- De más rápida a más lenta: A, B, C, D
 - Esto no significa que A sea siempre la más rápida (depende del programa), aunque, en conjunto, sí que lo es



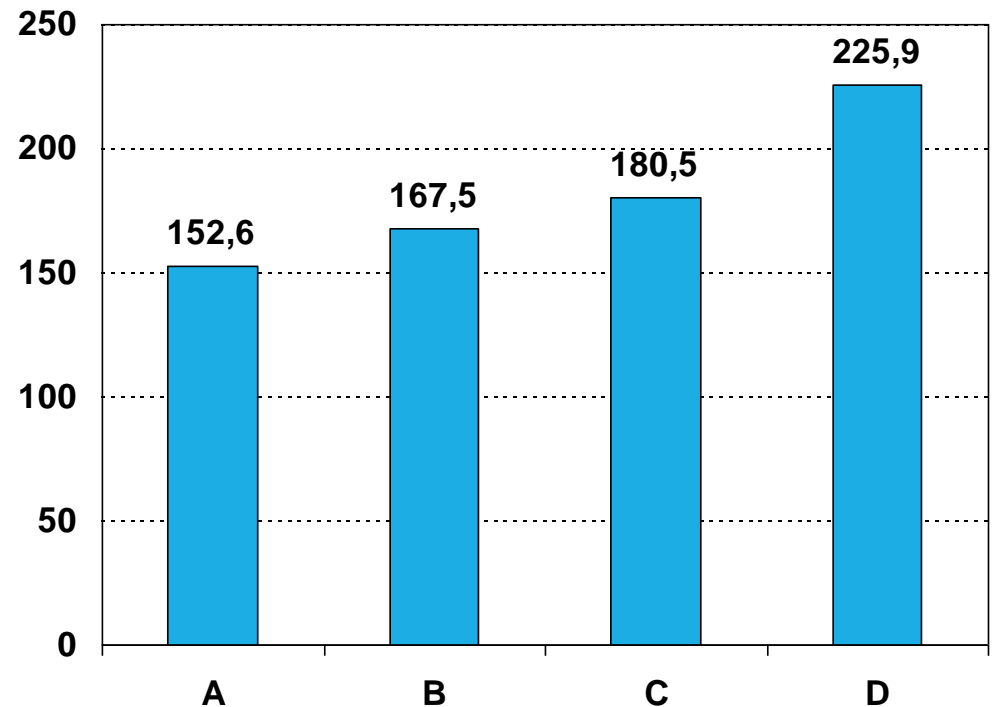
RESUMEN CON LA MEDIA ARITMÉTICA

$$x_A = \frac{1}{12} \sum_{i=1}^{12} t_{A,i} = \frac{1831,1}{12} = 152,6$$

$$x_B = \frac{1}{12} \sum_{i=1}^{12} t_{B,i} = \frac{2010,3}{12} = 167,5$$

$$x_C = \frac{1}{12} \sum_{i=1}^{12} t_{C,i} = \frac{2166,1}{12} = 180,5$$

$$x_D = \frac{1}{12} \sum_{i=1}^{12} t_{C,i} = \frac{2710,7}{12} = 225,9$$



De más rápida a más lenta: A, B, C, D

Esto no significa que A sea siempre la más rápida, pero sí que lo es en media

NORMALIZACIÓN DE RENDIMIENTOS: RATIOS

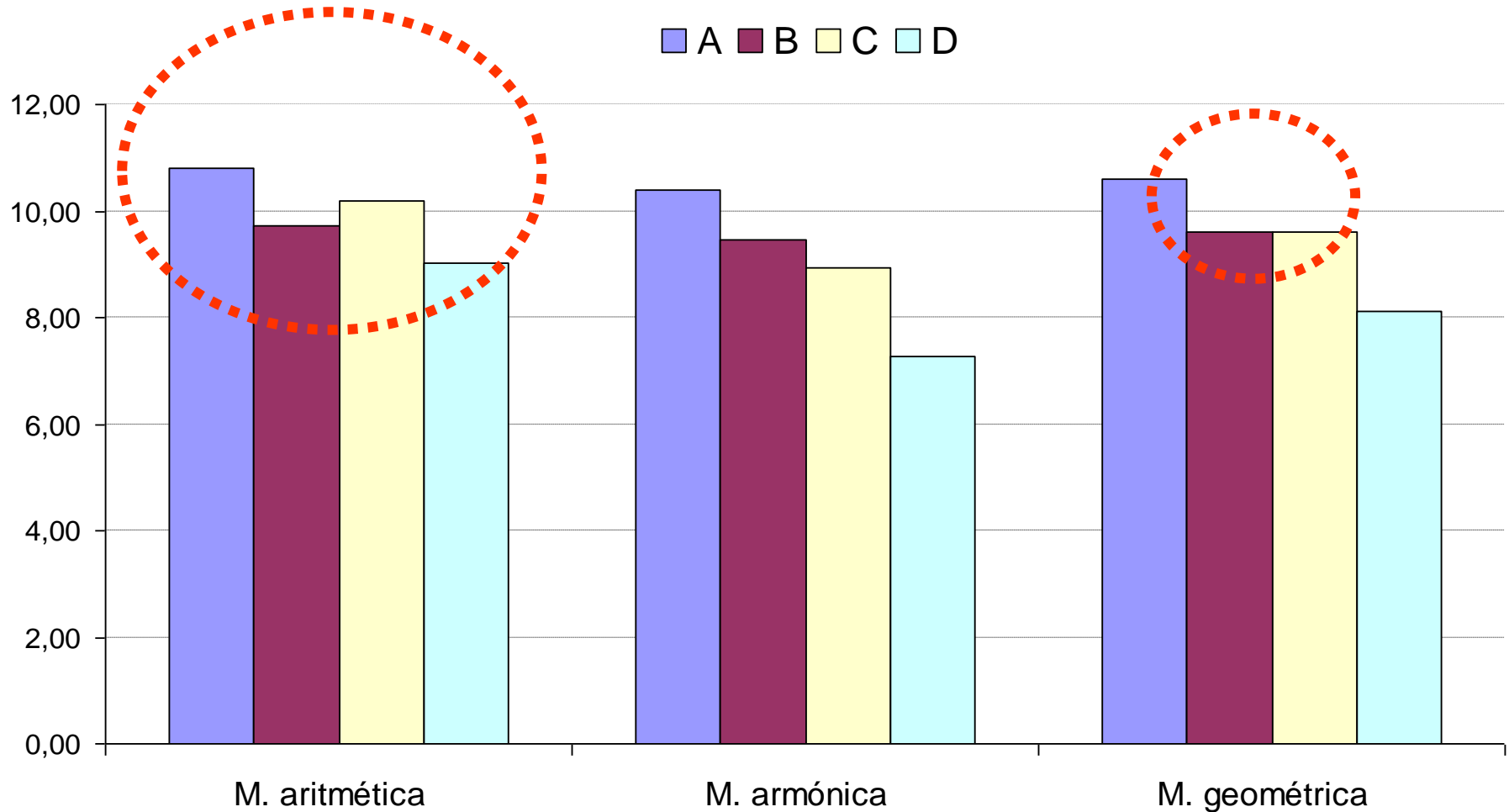
Si dividimos los valores de cada fila del valor obtenido en una máquina de referencia (aceleraciones individuales de los programas)

Programa	R	A	B	C	D
1	1400	9.9	8.2	10.3	10.4
2	1400	9.1	8.4	6.6	4.7
....					

A partir de aquí el resumen con las medias se hace con los valores normalizados sobre la máquina de referencia (aceleraciones individuales, ratios obtenidos por filas)

- Ahora **la aceleración es un índice a maximizar.**

COMPARACIÓN CON VALORES NORMALIZADOS (RATIOS POR FILAS, MEDIAS DE COLUMNAS)





COMPARANDO SERVIDORES

ACTIVIDAD TEMA 3.1 (OBLIGATORIA)

Carga	Sistema R	Sistema A	Sistema B	Sistema C	Sistema D
File I/O	70	7,05	8,5	6,8	6,7
CPU	70	7,70	8,3	10,75	15,05
Threads	55,5	4,84	4,71	7,30	10,05
Mutex	90	13,55	14,15	21,4	26,15
Memory	50	4,19	4,50	3,87	4,06

¿Son equivalentes las medidas de rendimiento de los tiempos de respuesta de los cuatro sistemas (A,B,C,D) utilizando los tres tipos de medias? ¿Por qué?

Si quisieras calcular las aceleraciones de los sistemas sobre el sistema de referencia R, ¿Qué tipo de media harías para obtener una medida de rendimiento correcta?

Y si quisieras calcular las aceleraciones de los tipos cargas sobre la carga de referencia ¿Qué tipo de media harías para poder obtener una medida de rendimiento correcta?

3. REFERENCIACIÓN (*BENCHMARKING*)

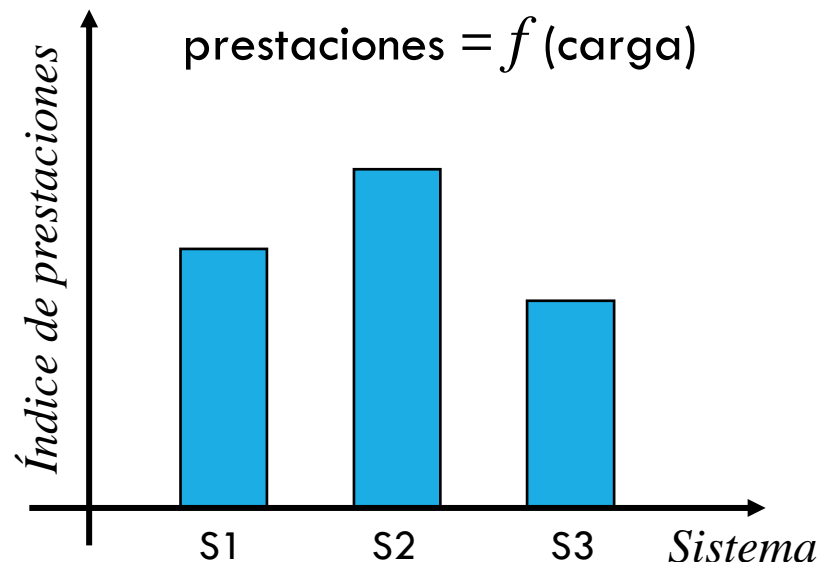
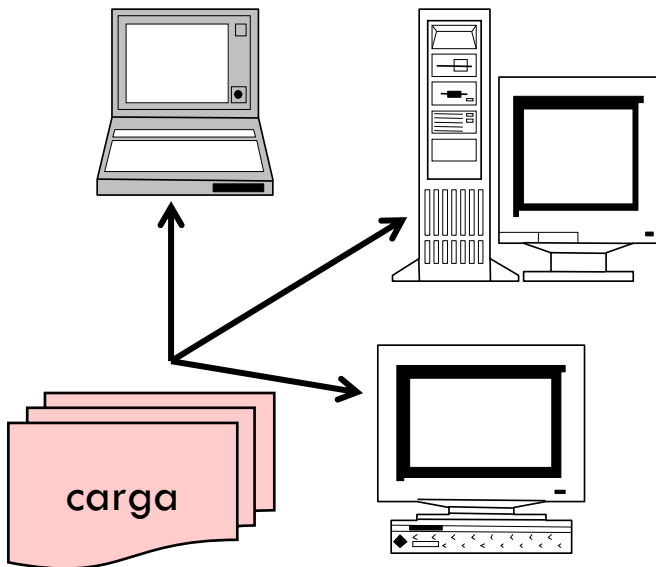
Comparar rendimientos a través de la
ejecución de programas

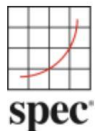


¿QUÉ ES LA REFERENCIACIÓN?

Referenciación (*benchmarking*)

- Técnicas usadas en la comparación del rendimiento de diferentes sistemas informáticos
- Todos los sistemas se han de someter a la misma carga





First Quarter 2018 SPEC CPU2017 Results

These results have been submitted to SPEC; see [the disclaimer](#) before studying any results.

[Search published CPU2017 results](#)

Last update: 2018-05-04t14:08

Published Results (1571):

[Search](#) | [CPU2017 Floating Point Rates](#) | [CPU2017 Floating Point Speed](#) | [CPU2017 Integer Rates](#) | [CPU2017 Integer Speed](#) |

CPU2017 Floating Point Rates (412):

[\[Search in CPU2017 Floating Point Rates results\]](#)

Índices de rendimiento

Test Sponsor	System Name		Base Copies	Processor			Results		Energy	
				Enabled Cores	Enabled Chips	Threads/ Core	Base	Peak	Base	Peak
ASUSTeK Computer Inc.	ASUS RS700-E9(Z11PP-D24) Server System (2.70 GHz, Intel Xeon Gold 6150) HTML CSV Text PDF PS Config		72	36	2	2	199	202	--	--
ASUSTeK Computer Inc.	ASUS RS700-E9(Z11PP-D24) Server System (2.10 GHz, Intel Xeon Platinum 8176) HTML CSV Text PDF PS Config		112	56	2	2	233	236	--	--
ASUSTeK Computer Inc.	ASUS WS C621E SAGE Server System (2.50 GHz, Intel Xeon Platinum 8180) HTML CSV Text PDF PS Config		112	56	2	2	252	257	--	--

Hardware

CPU Name: Intel Xeon Gold 6152
Max MHz: 3700
Nominal: 2100
Enabled: 44 cores, 2 chips, 2 threads/core
Orderable: 1,2 Chips
Cache L1: 32 KB I + 32 KB D on chip per core
L2: 1 MB I+D on chip per core
L3: 30.25 MB I+D on chip per chip
Other: None
Memory: 384 GB (24 x 16 GB 2Rx4 PC4-2666V-R)
Storage: 1 x 600 GB SAS HDD, 10K RPM
Other: None

Software

OS: SUSE Linux Enterprise Server 12 SP2 (x86_64)
 4.4.21-69-default
Compiler: C/C++: Version 18.0.0.128 of Intel C/C++
 Compiler for Linux;
 Fortran: Version 18.0.0.128 of Intel Fortran
 Compiler for Linux
Parallel: No
Firmware: Version 3.1.1d released Jun-2017
File System: xfs
System State: Run level 3 (multi-user)
Base Pointers: 64-bit
Peak Pointers: 32/64-bit
Other: jemalloc: jemalloc memory allocator library
 V5.0.1;
 jemalloc: configured and built at default for
 32bit (i686) and 64bit (x86_64) targets;
 jemalloc: built with the RedHat Enterprise 7.4,
 and the system compiler gcc 4.8.5
 jemalloc: sources available at
 releases

Hardware specification

Executed workload

Software specification

Results Table

Benchmark	Base								Peak							
	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
500.perlbench r	88	828	169	827	169	834	168	88	681	206	683	205	680	206		
502.gcc r	88	657	190	658	189	667	187	88	541	230	542	230	541	230		
505.mcf r	88	549	259	550	258	556	256	88	552	257	553	257	560	254		
520.omnetpp r	88	820	141	829	139	845	137	88	885	130	885	130	887	130		
523.xalanbmk r	88	459	202	461	201	462	201	88	373	249	372	250	373	249		
525.x264 r	88	353	437	353	437	352	438	88	333	462	336	459	338	456		
531.deepsjeng r	88	535	189	537	188	538	188	88	539	187	539	187	537	188		
541.leela r	88	788	185	804	181	789	185	88	776	188	776	188	779	187		
548.exchange2 r	88	543	425	542	425	542	426	88	542	425	544	424	543	424		
557.xz r	88	622	153	623	153	622	153	88	625	152	625	152	625	152		

SPECrate2017 int base 217
 SPECrate2017 int peak 230

Performance summary

Results appear in the order in which they were run. Bold underlined text indicates a median measurement.

FACTORES INFLUYENTES EN LA REFERENCIACIÓN

Elementos hardware de cada equipo

- Procesador, placa base y *chipset*, memoria principal, discos, ...

Sistema operativo

Sistema de memoria

- Configuraciones de las memorias cache y virtual

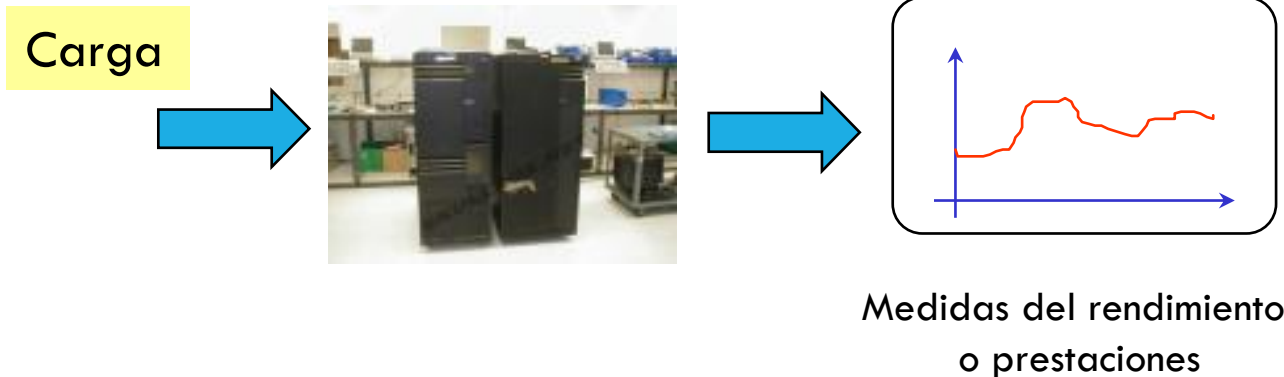
Efecto del compilador sobre el código ejecutable

- Cuidado con las posibles optimizaciones
 - `j=log(5.0); por for(i=1;i<=5000;i++) j=log(5.0);`
 - `x=exp(log(x)/2*T1)` por la expresión `x=sqrt(exp(log(x)/T1))`, ya que `sqrt(exp(x))=exp(x/2)`
- Hay compiladores “listos” que detectan los *benchmarks* clásicos

CARGA DE TRABAJO Y CARGA DE PRUEBA

Carga de trabajo (*workload*)

- Conjunto de todas las demandas a que está sometido un sistema informático durante un periodo de tiempo
 - Programas, datos, órdenes, usuarios, ...



Carga de prueba (*test workload*)

- Carga a que se somete un sistema informático durante una sesión de medida de rendimiento

LA CARGA REAL

Difícil de utilizar en la evaluación de sistemas

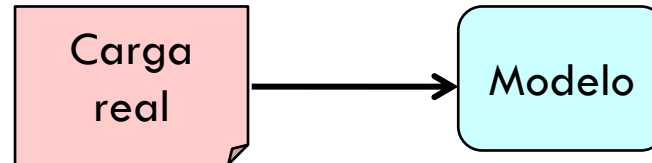
- Resulta complicado reproducirla
- Varía a lo largo del tiempo
- Interacciona con el sistema informático



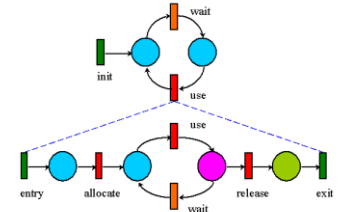
EL MODELADO DE LA CARGA

Caracterización o modelado de la carga

- Descripción cuantitativa

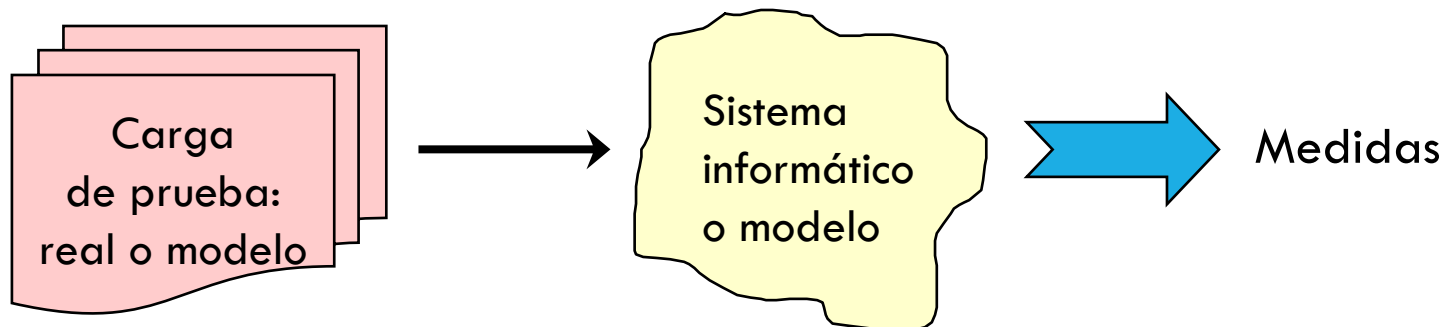


Structured Workload

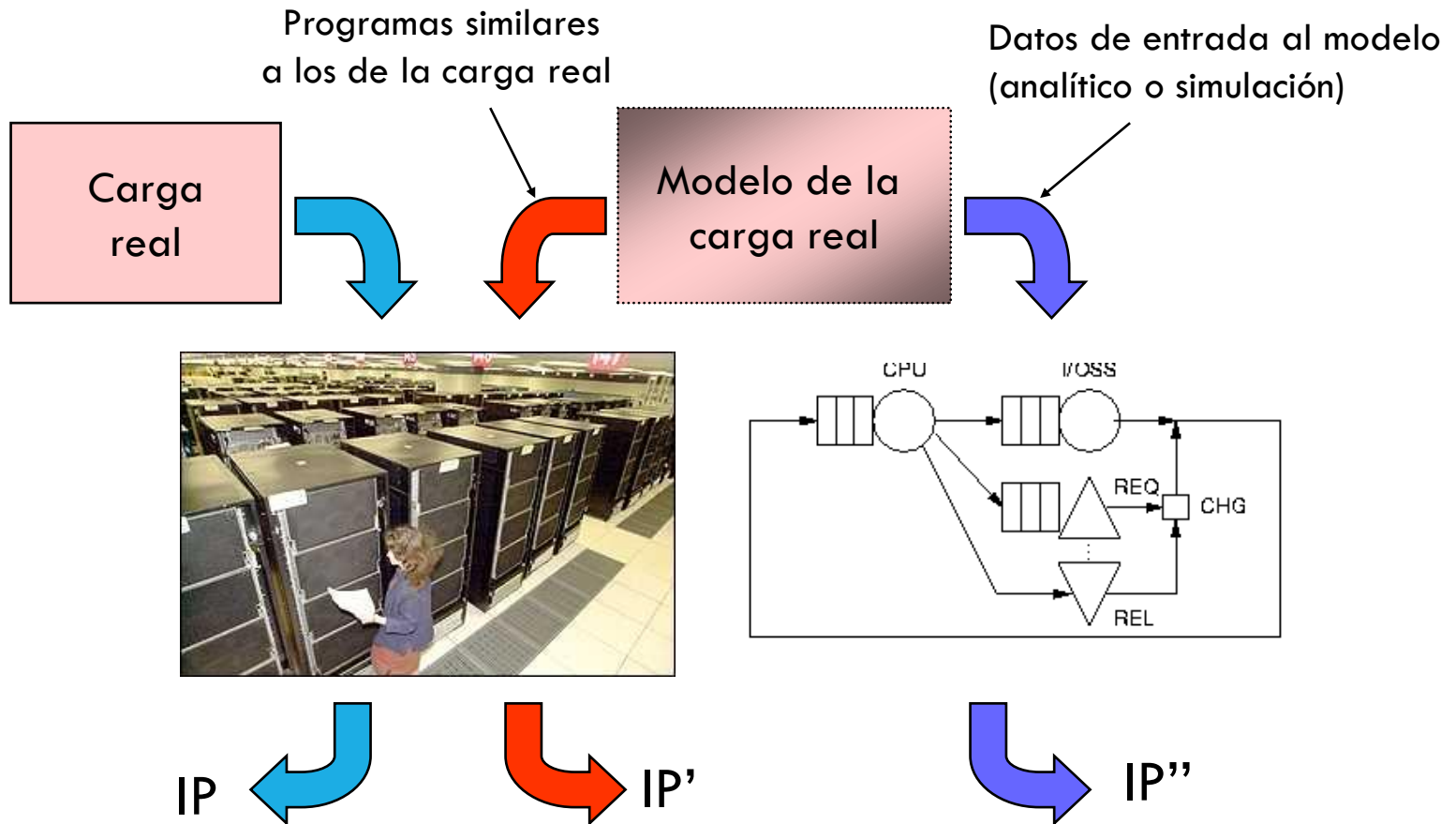


Toma de medidas de rendimiento

- A partir del sistema y carga reales, o bien a partir de modelos



OBTENCIÓN DE LOS ÍNDICE DE PRESTACIONES



¿QUÉ SISTEMA ES MÁS RÁPIDO?

El rendimiento del computador A es n veces mejor que el rendimiento del computador B si

$$\frac{R_A}{R_B} = \frac{T_B}{T_A} = n$$

Ejemplo: $T_A=10$ s y $T_B=15$ s

$$\frac{T_B}{T_A} = 1.5$$

- El computador A es 1.5 veces más rápido que el B

¿QUÉ SISTEMA ES MÁS RÁPIDO?

El rendimiento del computador A es un $n\%$ superior al rendimiento del computador B si

$$\frac{R_A}{R_B} = \frac{T_B}{T_A} = 1 + \frac{n}{100}, \text{ o también, } \left(\frac{R_A - R_B}{R_B} \right) \times 100 = n$$

Ejemplo: $T_A = 10$ s y $T_B = 15$ s

$$\frac{T_B}{T_A} = 1 + \frac{n}{100} \Rightarrow n = \left(\frac{T_B - T_A}{T_A} \right) \times 100 = 50$$

- El computador A es un 50% más rápido que el B

¿RENDIMIENTOS DIFERENTES?

Ejecución de n programas en dos máquinas A y B

¿Son significativas las diferencias obtenidas?

- Hay que usar mecanismos estadísticos

Intervalo de confianza para las diferencias

$$\bar{x} \pm t_{1-\frac{\alpha}{2}, n-1} \frac{s}{\sqrt{n}}$$

Nivel de confianza, por ejemplo, del 95% ($\alpha=0.05$)

- Si incluye el cero, entonces no hay diferencias significativas
- Si no incluye el cero, entonces las máquinas tienen rendimientos significativamente diferentes

EJEMPLO DE RENDIMIENTOS SIMILARES

Programa	A	B	Diferencias (A-B)
P1	5.4	19.1	-13.7
P2	16.6	3.5	13.1
P3	0.6	3.4	-2.8
P4	1.4	2.5	-1.1
P5	0.6	3.6	-3.0
P6	7.3	1.7	5.6

¿Son significativas estas diferencias?

CÁLCULO DEL INTERVALO DE CONFIANZA

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = -0.32, s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} = 9.03, t_{1-\frac{0.1}{2}, n-1} = t_{0.975, 5} = 2.571$$

$$\bar{x} \pm t_{1-\frac{\alpha}{2}, n-1} \frac{s}{\sqrt{n}} = -0.32 \pm 2.571 \times \frac{9.03}{\sqrt{6}} = -0.32 \pm 9.478$$

Como el intervalo calculado es (-9.80, 9.17) e incluye el cero podemos afirmar, con un nivel de confianza del 95%, que las diferencias **NO SON SIGNIFICATIVAS**



4. ESTRATEGIAS DE ANÁLISIS

Efectos de la normalización y de la ponderación



ESTRATEGIAS DE ANÁLISIS

Muy raramente un estudio comparativo de prestaciones es inocente, sobre todo cuando intervienen los fabricantes y priman intereses económicos

Caso de estudio

- Tres sistemas: A, B y C
- Dos programas de prueba: P1 y P2
- **Tiempo de ejecución en segundos de cada programa**

	A	B	C
P1	20	10	40
P2	40	80	20

TIEMPO INDIVIDUAL DE CADA PROGRAMA

Tiempo individual de cada máquina

	A	B	C
P1	20	10	40
P2	40	80	20

Vamos a comprobar, una vez más, cómo las conclusiones de un estudio de comparación de rendimiento se pueden alterar empleando, de manera incorrecta la aplicación de valores medios y normalizaciones.

- La comparación entre los tres sistemas puede resultar confusa, porque A y C se comportan al revés y B es extremo.

TIEMPO TOTAL DE EJECUCIÓN

	A	B	C
P1	20	10	40
P2	40	80	20
Suma	60	90	60

La medida más fiable para comparar rendimientos viene dada por el tiempo de ejecución suma de los programas de prueba.

En este sentido la suma del tiempo de ejecución de estos programas será la que determine qué máquina es más rápida.

A y C son equivalentes y 1,5 veces más rápidas que B

NORMALIZAR EL TIEMPO TOTAL DE EJECUCIÓN

	A	B	C
P1	20	10	40
P2	40	80	20

Suma	60	90	60
Respecto de A	1,0	1,5	1,0
Respecto de B	0,67	1,0	0,67
Respecto de C	1,0	1,5	1,0

Si nos fijamos en las normalizaciones (ratio) podemos comprobar que la ordenación del tiempo suma se mantiene.

MEDIA ARITMÉTICA Y NORMALIZADA DEL TIEMPO DE EJECUCIÓN

	A	B	C
P1	20	10	40
P2	40	80	20

Media aritmética	30	45	30
Respecto de A	1,0	1,5	1,0
Respecto de B	0,67	1,0	0,67
Respecto de C	1,0	1,5	1,0

La media aritmética se puede utilizar para promediar tiempos de ejecución porque mantiene el significado físico de las cantidades que se resumen.

La media aritmética calculada sobre los tiempos de ejecución, así como el valor normalizado de esta media respecto de los tres sistemas mantiene el orden.

MEDIA GEOMÉTRICA DEL TIEMPO TOTAL DE EJECUCIÓN

	A	B	C
P1	20	10	40
P2	40	80	20
Suma	60	90	60
Media geométrica	28,3	28,3	28,3

En los tres casos la media geométrica ofrece el mismo resultado, y como consecuencia, la conclusión que se puede dirimir es que los tres computadores tienen un rendimiento equivalente, **hecho que contrasta con la suma de los tiempos de ejecución.**

NORMALIZACIÓN DE LOS TIEMPOS Y LA MEDIA ARITMÉTICA

Normalización sobre diferentes bases A, B y C

	Referencia A			Referencia B			Referencia C		
	A	B	C	A	B	C	A	B	C
P1 (20/10/40)	1.0	2.0	0.5	0.5	1.0	0.25	2.0	4.0	1.0
P2 (40/80/20)	1.0	0.5	2.0	2.0	1.0	4.0	0.5	0.25	1.0
Suma (60/90/60)	2.0	2.5	2.5	2.5	2.0	4.25	2.5	4.25	2.0
Media aritmética (30/45/30)	1.0	1.25	1.25	1.25	1.0	2.125	1.25	2.125	1.0

Podemos observar que el uso de la suma o la media aritmética de los tiempos de ejecución normalizados mejora ostensiblemente la máquina que se toma como referencia y, peor aún, no mantiene el orden establecido por la suma de los tiempos de ejecución sin normalizar.

NORMALIZACIÓN DE LOS TIEMPOS Y LA MEDIA GEOMÉTRICA

Normalización sobre diferentes bases A, B y C

	Referencia A			Referencia B			Referencia C		
	A	B	C	A	B	C	A	B	C
P1 (20/10/40)	1.0	2.0	0.5	0.5	1.0	0.25	2.0	4.0	1.0
P2 (40/80/20)	1.0	0.5	2.0	2.0	1.0	4.0	0.5	0.25	1.0
Suma (60/90/60)	2.0	2.5	2.5	2.5	2.0	4.25	2.5	4.25	2.0
Media geométrica (28.3/28.3/28.3)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Como se puede observar, **la media geométrica establece el mismo orden entre los tres computadores independientemente de si se utilizan los tiempos de ejecución sin normalizar o normalizados** (no importa qué máquina se tome como referencia).

NORMALIZACIÓN DE LOS TIEMPOS Y LA MEDIA GEOMÉTRICA

Este tipo de media se emplea en el cálculo de índices como SPEC, donde los tiempos de ejecución se normalizan respecto de una máquina de referencia.

- Sin embargo, el orden que proporciona es erróneo, ya que según los valores que acabamos de calcular, las tres máquinas presentan rendimientos totalmente equivalentes, aunque sabemos que no tardan lo mismo en ejecutar los dos programas.
- Es una de las razones por las que, en ocasiones, una máquina con un índice SPEC más elevado que otra puede tardar más tiempo en ejecutar un determinado conjunto de programas de prueba.

CÓMO UTILIZAR LAS MEDIAS (RECOMENDACIONES)

Cuando se dispone de **tiempos**, se utiliza la media **aritmética** (o la media aritmética ponderada, según el caso)

Cuando se dispone de **frecuencias** o **velocidades**, se emplea la media **armónica** (o la media armónica ponderada, según el caso)

Cuando se dispone de **índices** de rendimiento, se puede emplear la media **geométrica**, **pero a riesgo de reordenar los resultados**

Cuando se dispone de **porcentajes**, se utiliza la media **aritmética** (o la media aritmética ponderada, según el caso).

Cuando se dispone de **tiempos** **primero se hace la media aritmética y luego se normaliza y no al revés**

Cuando se dispone de **valores** muy **extremos**, no se utiliza la media aritmética directamente, se **eliminan** los extremos, si se puede...

CÓMO UTILIZAR LAS MEDIAS (RECOMENDACIONES)

El uso de cada tipo de media depende del significado físico que represente cada medida de rendimiento x_i .

- Por ejemplo, si lo que se mide es el tiempo de ejecución de diversos programas, el valor medio calculado deberá ser proporcional a la suma de los tiempos de ejecución.
- Al contrario, si lo que se mide es la frecuencia o ratio entre el número de operaciones y el tiempo total de ejecución, el valor medio que se computa habrá de ser inversamente proporcional a la suma de los tiempos de ejecución.
- Por ejemplo, la media aritmética es adecuada para promediar tiempos de ejecución y la media armónica para ratios, como por ejemplo MFLOPS.
- Sin embargo, ni la media aritmética sirve para promediar frecuencias, ni la media armónica para promediar tiempos.

CÓMO UTILIZAR LAS MEDIAS (RECOMENDACIONES)

Por su parte, la media geométrica no resulta apropiada para promediar tiempos ni ratios.

- Sin embargo, algunos autores han defendido su uso para resumir valores normalizados porque mantiene un orden consistente cuando se comparan las prestaciones de varios computadores, aunque a veces este orden puede resultar erróneo.
- El adjetivo consistente significa en este contexto que el orden es el mismo, independientemente de la máquina que se tome como referencia para hacer la comparación.

CÓMO UTILIZAR LAS MEDIAS (RECOMENDACIONES)

En ocasiones, las medias anteriores se llevan a cabo sobre valores normalizados previamente (por ejemplo, dividiendo los tiempos de ejecución entre los obtenidos en una máquina de referencia), pero hay que saber que **la máquina de referencia no puede ser parte del experimento de comparación.**

En cualquier caso, **la mejor alternativa a seguir consiste en resumir el rendimiento en una media y aplicar después la normalización al valor obtenido.**

CÁLCULO DEL ÍNDICE SPECFP_BASE

Benchmark	Tiempo refer.	Tiempo medido	Relación (x100)
168.wupwise	1600	470	340
171.swim	3100	616	503
172.mgrid	1800	704	256
173.applu	2100	715	294
177.mesa	1400	334	419
178.galgel	2900	645	457
179.art	2600	1232	211
183.quake	1300	500	260
187.facerec	1900	488	390
188.amp	2200	756	291
189.lucas	2000	869	230
191.fma3d	2100	594	354
200.sixtrack	1100	465	237
301.apsi	2600	975	267

Hardware

- Intel Xeon Gold 6150 Cache
- 768 GB (24 x 32 GB 2Rx4 PC4-2666V-R) RAM

Software

- Red Hat Enterprise Linux Server release 7.3
- C/C++: Version 18.0.0.128 of Intel C/C++

$$SPECfp_base = \sqrt[14]{\prod_{i=1}^{14} \left(\frac{T_{ref}}{T_i} \times 100 \right)} = 311$$



BENCHMARKING DE SERVIDORES

ACTIVIDAD TEMA 3.2 (VOLUNTARIA)

En la tabla amarilla de la transparencia del cálculo del índice SPECfp_base hay una serie de tiempos de ejecución de programas, los de la máquina de referencia y la aceleración que representa la ejecución en un sistema ($\times 100$).

Propón un índice más conveniente que el que se calcula con la media geométrica del recuadro verde ¿Por qué crees que tu proposición es más justa para calcular el rendimiento de ese sistema?



5. BENCHMARKS

Microbenchmarks: Sysbench, Linpack

Macrobenchmarks: SPEC_CPU

Consortios SPEC, TPC



MICROBENCHMARKS VS MACROBENCHMARKS

*“**Macro-benchmarks** measure the **entire system**, and usually model some workload; they can be either synthetic or application benchmarks. **Micro-benchmarks** measure a **specific part of the system**. They can be thought of as a subset of synthetic benchmarks in that they are artificial; however, they do not try to model any real workload whatsoever.”*

* Gregg, B. (2014). *Systems performance: enterprise and the cloud*. Pearson Education.

MICROBENCHMARKS: *SYSBENCH*

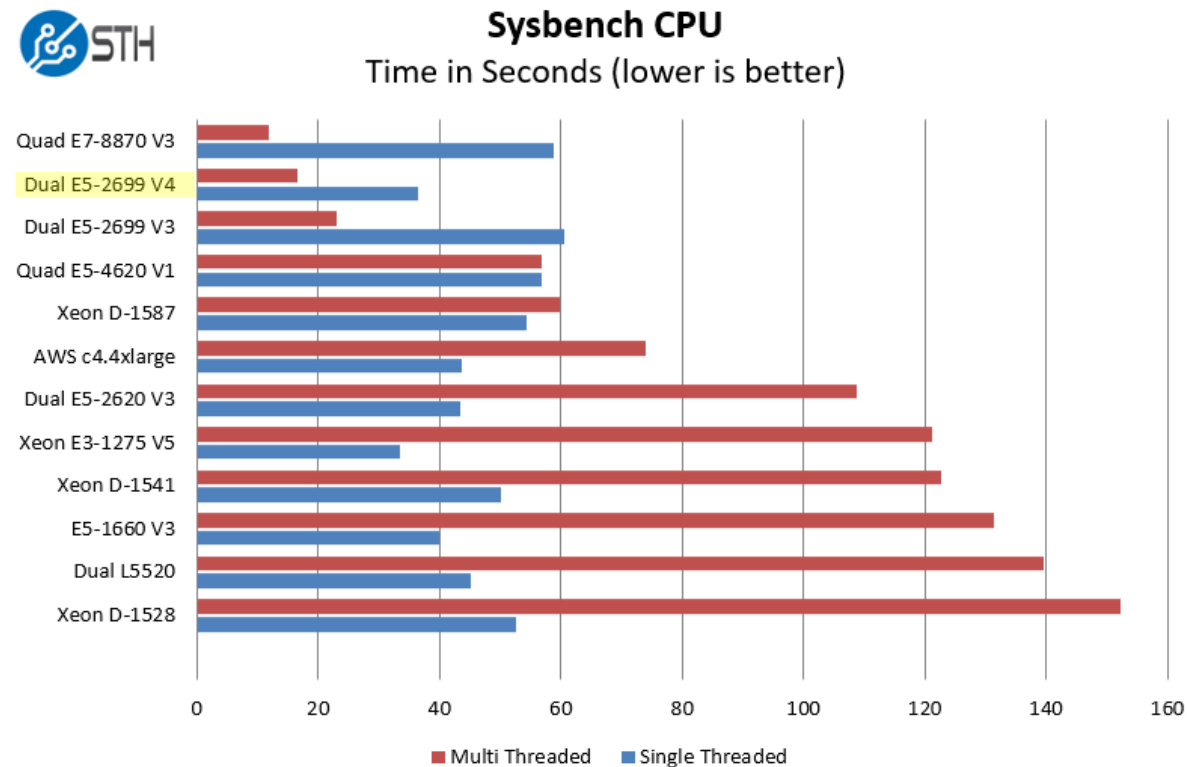
Sysbench permite testear componentes individuales de un sistema.

- CPU
 - Tiempo de respuesta, GFlops
- Memoria principal
 - Tiempo de respuesta, productividad
- I/O
 - Tiempo de respuesta, productividad
- SO (sistema mutex)
 - Número de bloqueos
- Base de datos MySQL
 - Tiempo de respuesta, productividad
- ...

MICROBENCHMARKS: *SYSBENCH*

Sysbench permite testear componentes individuales de un sistema.

- CPU

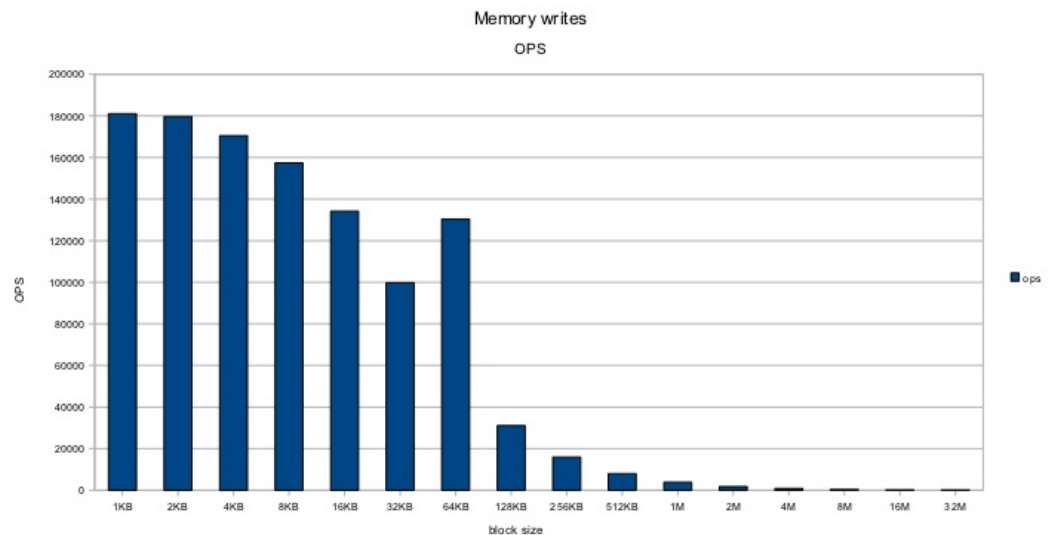


MICROBENCHMARKS: *SYSBENCH*

Sysbench permite testear

- Memoria principal

sysbench
memory test

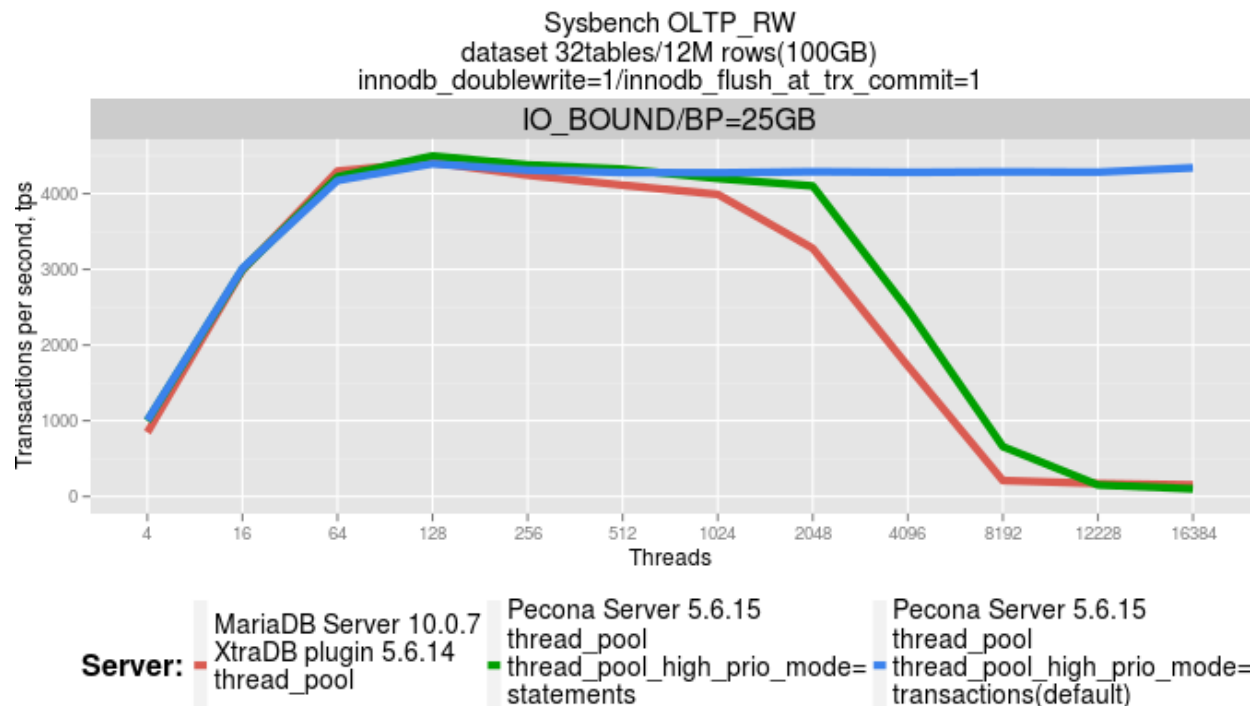


PALOMINODB Proven Database Excellence

MICROBENCHMARKS: *SYSBENCH*

Sysbench permite testear componentes individuales de un sistema.

- I/O



MICROBENCHMARKS: *LINPACK*

Benchmark clásico: desarrollado en 1975

Uso en sistemas de carácter científico (altas prestaciones)

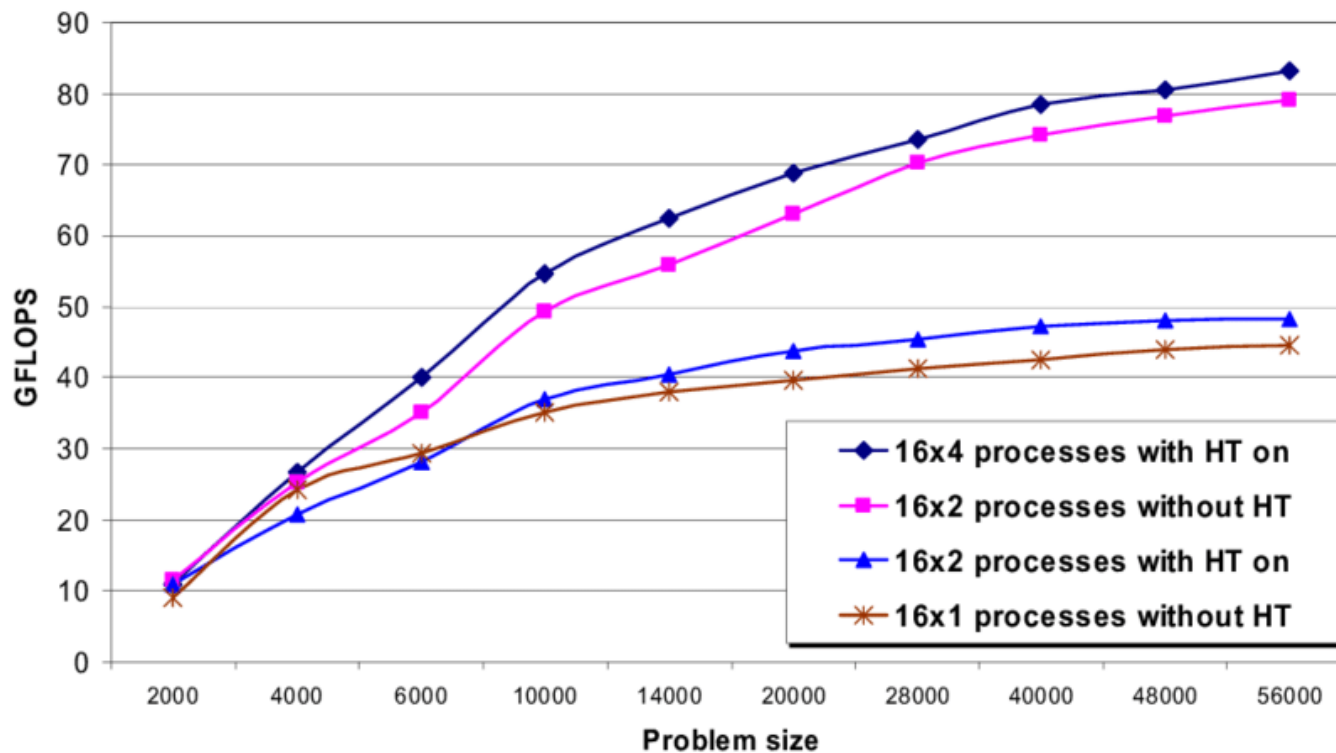
Resolución de sistemas de ecuaciones diferenciales

Resultados del benchmark: Mflops

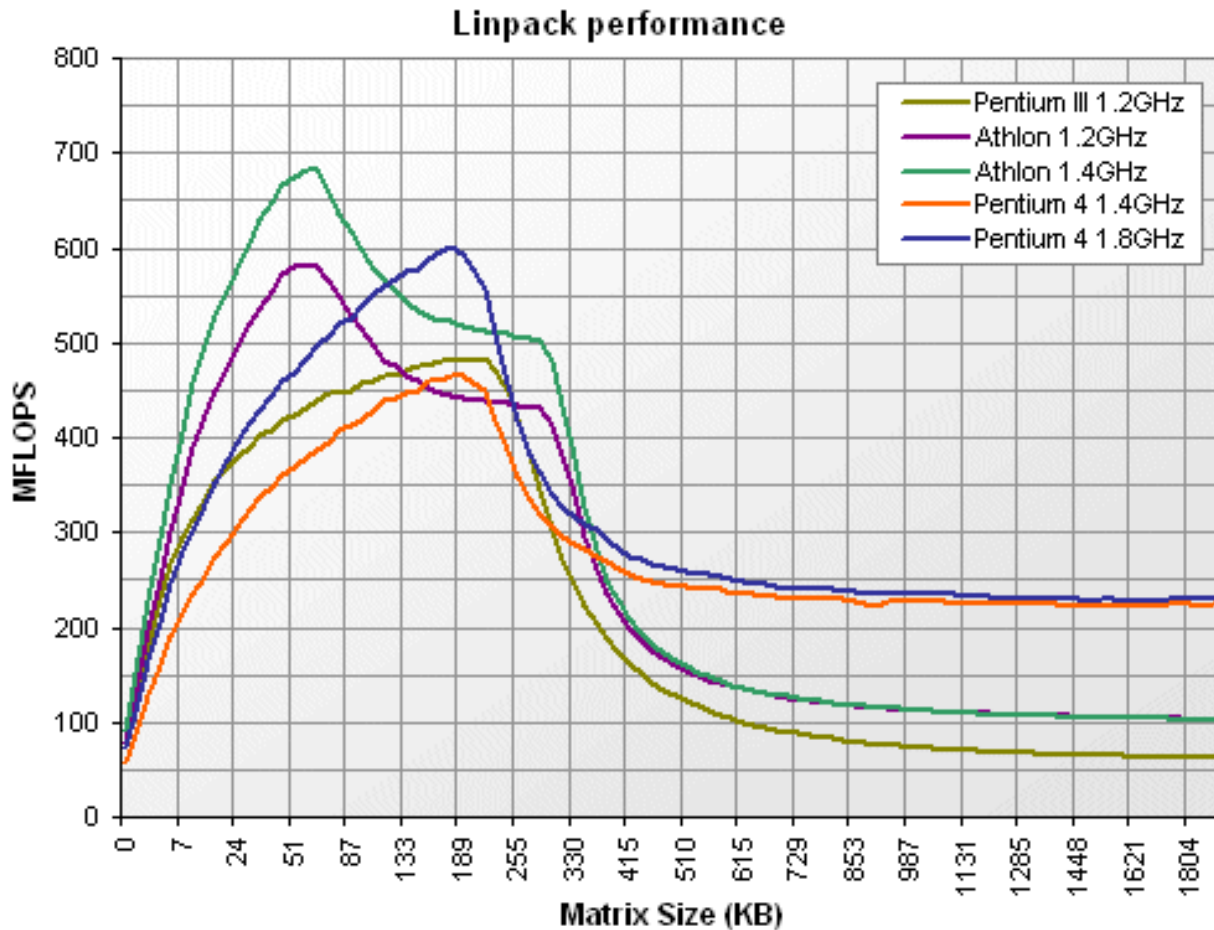
<http://www.netlib.org/linpack/>

MICROBENCHMARKS: *LINPACK*

Linpack Performance Results
on a 16-node Dual-XEON cluster



MICROBENCHMARKS: *LINPACK*



MACROBENCHMARKS: *SPEC_CPU2017*

- ⇒ Benchmark estándar en la industria
- ⇒ Mide y compara el rendimiento bajo diferentes niveles de intensidad
 - Sistema de CPUs
 - Sistema y subsistema de memoria
 - Sistema de compilación
 - <https://www.spec.org/cpu2017/>

MACROBENCHMARKS: *SPEC_CPU2017*

⇒ Resultados en tiempo de respuesta

Results Table

Results Table														
Benchmark	Base							Peak						
	Threads	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Threads	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
600.perlbenc s	72	285	6.24	<u>286</u>	<u>6.22</u>	286	6.21	72	240	7.41	239	7.42	<u>239</u>	<u>7.42</u>
602.gcc s	72	<u>423</u>	<u>9.42</u>	417	9.55	424	9.39	72	412	9.67	<u>413</u>	<u>9.65</u>	413	9.63
605.mcf s	72	<u>426</u>	<u>11.1</u>	429	11.0	426	11.1	72	425	11.1	<u>421</u>	<u>11.2</u>	418	11.3
620.omnetpp s	72	<u>257</u>	<u>6.35</u>	265	6.15	254	6.43	72	<u>248</u>	<u>6.58</u>	247	6.61	249	6.54
623.xalancbmk s	72	150	9.46	149	9.51	<u>150</u>	<u>9.46</u>	72	139	10.2	<u>140</u>	<u>10.1</u>	140	10.1
625.x264 s	72	<u>150</u>	<u>11.8</u>	150	11.7	149	11.8	72	150	11.8	150	11.8	<u>150</u>	<u>11.8</u>
631.deepsjeng s	72	280	5.11	<u>280</u>	<u>5.11</u>	280	5.12	72	282	5.08	282	5.08	<u>282</u>	<u>5.08</u>
641.leela s	72	<u>393</u>	<u>4.34</u>	393	4.34	393	4.34	72	<u>392</u>	<u>4.36</u>	392	4.36	392	4.36
648.exchange2 s	72	219	13.4	<u>220</u>	<u>13.4</u>	220	13.4	72	220	13.4	<u>220</u>	<u>13.4</u>	219	13.4
657.xz s	72	277	22.3	280	22.1	<u>280</u>	<u>22.1</u>	72	<u>277</u>	<u>22.3</u>	277	22.3	277	22.3
SPECspeed2017 int base			8.87											
SPECspeed2017 int peak			9.16											
Results appear in the order in which they were run. Bold underlined text indicates a median measurement.														

MACROBENCHMARKS: *SPEC_CPU2017*

⇒ Resultados en productividad

Results Table

Results Table														
Benchmark	Base							Peak						
	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Copies	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
500.perlbench_r	72	630	182	633	181	629	182	72	517	222	523	219	523	219
502.gcc_r	72	572	178	568	180	564	181	72	455	224	455	224	453	225
505.mcf_r	72	438	266	451	258	450	258	72	451	258	455	256	455	256
520.omnetpp_r	72	765	124	763	124	764	124	72	826	114	825	115	826	114
523.xalancbmk_r	72	366	208	365	208	367	207	72	285	266	286	266	286	266
525.x264_r	72	262	481	259	487	264	478	72	248	508	248	509	249	506
531.deepsjeng_r	72	386	214	395	209	399	207	72	403	205	404	204	403	205
541.leela_r	72	597	200	587	203	587	203	72	590	202	602	198	591	202
548.exchange2_r	72	401	470	401	470	401	470	72	401	470	401	470	401	470
557.xz_r	72	477	163	517	150	517	150	72	519	150	518	150	519	150
SPECrate2017_int_base			224											
SPECrate2017_int_peak			237											
Results appear in the order in which they were run. Bold underlined text indicates a median measurement.														

SPEC

SPEC (Standard Performance Evaluation Corporation)

- *MISSION: to establish, maintain, and endorse a standardized set of relevant benchmarks and metrics for performance evaluation of modern computer systems*

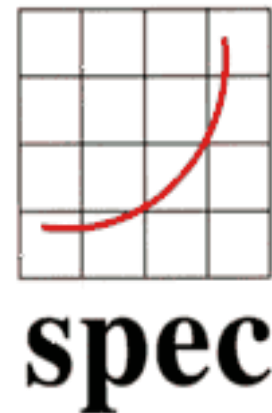
Funciones de SPEC

- Desarrollar benchmarks
- Hacer públicos los resultados

Intervienen muchas empresas

- HP, MIPS, IBM, Intel,
- Motorola, DEC, etc.

<http://www.spec.org>





Standard Performance Evaluation Corporation

[Home](#) [Benchmarks](#) [Tools](#) [Results](#) [Contact](#) [Site Map](#) [Search](#) [Help](#)

Benchmarks

- [Cloud](#)
- [CPU](#)
- [Graphics/Workstations](#)
- [ACCEL/MPI/OMP](#)
- [Java Client/Server](#)
- [Mail Servers](#)
- [Storage](#)
- [Power](#)
- [Virtualization](#)
- [Web Servers](#)

Results Search

Submitting Results

[Cloud/CPU/Java/Power](#)
[SFS/Virtualization](#)
[ACCEL/MPI/OMP](#)
[SPECapc/SPECviewperf/SPECwpcc](#)

The Standard Performance Evaluation Corporation (SPEC) is a non-profit corporation formed to establish, maintain and endorse standardized benchmarks and tools to evaluate performance and energy efficiency for the newest generation of computing systems. SPEC develops benchmark suites and also reviews and publishes submitted results from our [member organizations](#) and other benchmark licensees.

What's New:

01/16/2019: The [SPECjbb 2015](#) benchmark has been updated to version 1.02. This update adds support for Java SE 11. Licensees of versions 1.0 and 1.01 are entitled to a complimentary V1.02 upgrade. SPEC will be retiring SPECjbb2015 V1.01 in April 2019, after which time all result submissions must be made with V1.02.

01/16/2019: [SPEC Cloud IaaS 2016](#) is being retired in favor of its successor, [SPEC Cloud IaaS 2018](#). The last date for submitting benchmark results to SPEC for publication on its website is March 6, 2019. On March 6, 2019, the product will no longer be actively supported by SPEC.

12/20/2018: The [Kaivalya Dixit Distinguished Dissertation](#) Selection committee has chosen this year to select two winners

TPC

TPC (*Transactions Processing Performance Council*)

- Entornos transaccionales (OLTP: *on-line transaction processing*)
 - <http://www.tpc.org>
- Compuesto por varios programas
 - TPC-App, TPC-C, TPC-H
- Aspectos tratados
 - Sistemas de bases de datos distribuidas
 - Arquitectura cliente/servidor, servidores web
- Unidades: peticiones procesadas por segundo (tps)





dedicated on developing data-centric benchmark standards and disseminating objective, verifiable performance data to the industry... The TPC is a non-profit corporation focused

[Document Search](#)[Member Login](#)[Home](#)[About the TPC](#)[Benchmarks](#)[Newsletter](#)[Join the TPC](#)[Downloads](#)[Technical Articles](#)[TPCTC](#)

What's New

02/10/2019 [TPC announces TPCTC 2019 in Los Angeles, CA](#)

01/30/2019 [TPC releases TPC-DS Version 2.10.1](#)

01/30/2019 [TPC is looking for database providers to add support in TPCx-IoT](#)

TPC Benchmarks & Benchmark Results

Please select any of the active TPC benchmarks below. All available options will be displayed.

Transaction
Processing -
OLTP

[TPC-C](#)[TPC-E](#)

Decision
Support

[TPC-H](#)[TPC-DS](#)[TPC-DI](#)

Most Visited Pages

[TPC - Current Specifications](#)[TPC-C - Advanced Sort](#)[TPC - Benchmark Overview](#)[TPC-E - Top Ten Performance Results](#)[TPC - Spreadsheets of TPC Results](#)[TPC-H - Advanced Sort](#)[TPC-C - All Results - Sorted by Performance](#)[TPC - Who We Are](#)[TPC-Tools Download TPC-DS](#)

Copyright © 1988-2019 TPC. All rights reserved. Web-Design and Maintenance by: [Hotea Solutions](#)



EL PAQUETE SPEC CPU2017

Compuesto por dos partes

- CINT2000: rendimiento en aritmética entera intensiva
- CFP2000: rendimiento en aritmética de coma flotante intensiva

¿Qué quiere decir la “C”?

- Se evalúan componentes, no el sistema entero

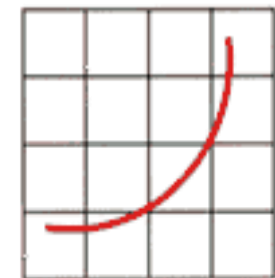
¿Qué componentes se evalúan?

- Procesador
- Arquitectura de memoria
- Compilador

¿Qué componentes no se evalúan?

- Subsistema de disco, red o gráficos

<https://www.spec.org/cpu2017/Docs/overview.html>



spec

CRITERIOS DE SELECCIÓN DE PROGRAMAS

Portabilidad a muchas arquitecturas

- 32 y 64 bits: Alpha, Intel, PA-RISC, Rxx00, SPARC, etc

Portabilidad a muchos sistemas operativos

- Unix y NT

No han de contener I/O, ni red ni gráficos

Han de caber en 256 MB de memoria principal para no provocar intercambio (*swapping*) con el disco

No han de pasar más del 5% del tiempo ejecutando código que no sea especificado por SPEC

PROGRAMAS DENTRO DE CPU2017

CINT2000: 12 programas

- Todos en lenguaje C
 - 164.zip Utilidad de compresión
 - 176.gcc Compilador de C
 - 186.crafty Programa para jugar al ajedrez
 - 197.parser Procesador de lenguaje natural
 - 254.gap Teoría computacional de grupos

CFP2000: 14 programas, en lenguajes FORTRAN y C

- 6 en FORTRAN77, 4 en FORTRAN90 y 4 en C
 - 173.applu Ecuaciones diferenciales parciales
 - 178.galgel Dinámica de fluidos
 - 200.sixtrack Modelo de acelerador de partículas

ÍNDICES PRESTACIONES EN SPEC CPU2017

Índices de prestaciones

- Aritmética entera
 - SPECint2000, SPECint_base2000
- Aritmética en coma flotante
 - SPECfp2000, SPECfp_base2000

Significado de “base” y “non-base”

- Compilación en modo conservativo o agresivo

Cálculo

- Media geométrica de los valores normalizados respecto de la máquina de referencia Sun Ultra5_10 con un procesador a 300 MHz
 - Tarda aproximadamente 2 días en ejecutar todos los benchmarks del paquete