

ISE-TEMA-4.pdf



Phantone



Ingeniería de Servidores



3º Grado en Ingeniería Informática

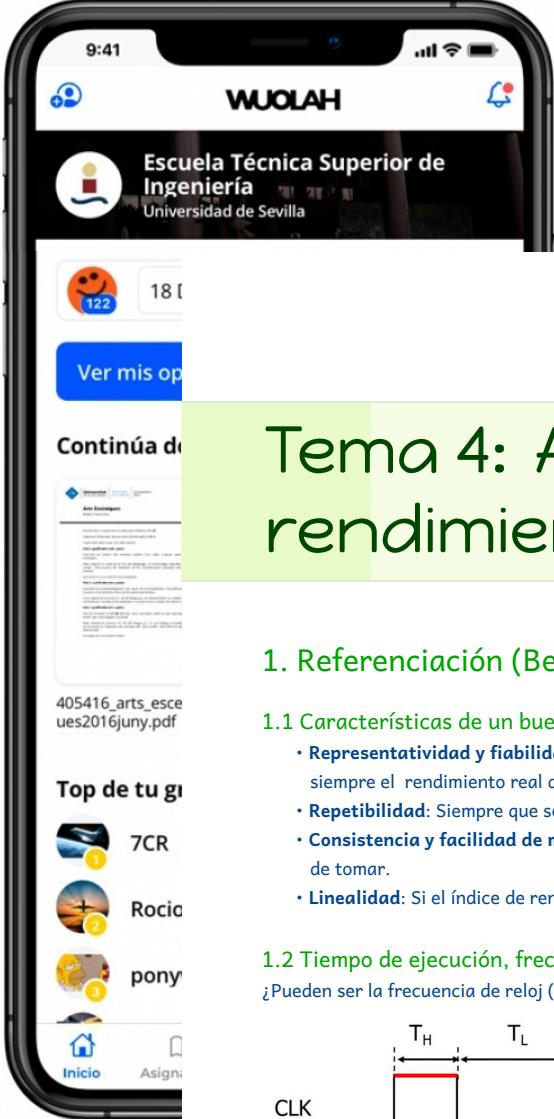


Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the App Store GET IT ON Google Play



18

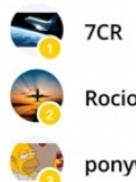
Ver mis op

Continúa d



405416_arts_esce
ues2016juny.pdf

Top de tu g



Inicio Asigna

Tema 4: Análisis comparativo del rendimiento

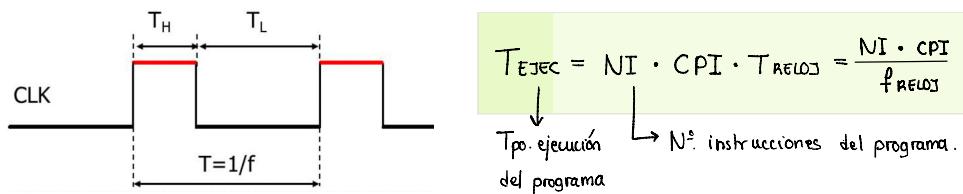
1. Referenciación (Benchmarking)

1.1 Características de un buen índice de rendimiento de un sistema informático

- **Representatividad y fiabilidad:** Si un sistema A siempre presenta un índice de rendimiento mejor que el sistema B, es porque siempre el rendimiento real de A es mejor que el de B.
- **Repetibilidad:** Siempre que se mida el índice en las mismas condiciones, el valor de éste debe ser el mismo.
- **Consistencia y facilidad de medición:** El índice se debe poder medir en cualquier sistema informático y esta medida debe ser fácil de tomar.
- **Linealidad:** Si el índice de rendimiento aumenta, el rendimiento real del sistema debe aumentar en la misma proporción.

1.2 Tiempo de ejecución, frecuencia de reloj y CPI

¿Pueden ser la frecuencia de reloj (fRELOJ) o el número medio de ciclos por instrucción (CPI) buenos índices de rendimiento?



No lo son. Es posible encontrar ejemplos de sistemas con fRELOJ (o CPI) peores que otros pero con mejores prestaciones.

¿Y si usamos directamente el tiempo de ejecución (TEJEC) de un determinado programa?

- ¿Consistencia? El programa debería estar escrito en un lenguaje de alto nivel.
- ¿Repetibilidad? El programa debería ejecutarse en un entorno muy controlado.
- ¿Representatividad y fiabilidad? Dependería del programa a ejecutar.

1.3. MIPS (million of instructions per second)

En principio, parece una medida prometedora ya que representa cómo de rápido ejecuta las instrucciones un microprocesador.

Inconvenientes:

- ¿Representatividad y fiabilidad? Depende del juego de instrucciones (ej. RISC vs CISC).
- ¿Repetibilidad? Los MIPS medidos varían incluso entre diferentes programas en el mismo computador.

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

Basado en operaciones y no en instrucciones.

$$MFLOPS = \frac{\text{Operaciones de coma flotante realizadas}}{T_{EJEC} \cdot 10^6}$$

Inconvenientes:

- ¿**Representatividad y fiabilidad?**: no todas las operaciones de coma flotante tienen la misma complejidad. Posible solución: MFLOPS normalizados: cada operación se multiplica por un peso que es proporcional a su complejidad. EJEMPLO:
 - ADD, SUB, COMPARE, MULT → 1 operación normalizada
 - DIVIDE, SQRT → 4 operaciones normalizadas
 - EXP, SIN, ATAN, ... → 8 operaciones normalizadas
- ¿**Consistencia**? El formato de los números en coma flotante puede variar de una arquitectura a otra y, por tanto, los resultados de las operaciones podrían tener diferente exactitud. Además, ¿y si no necesito las operaciones en coma flotante en mi servidor?

Conclusión final: Tampoco nos vale y no hay más candidatos. Nos contentaremos con el tiempo de ejecución (TEJEC) de un determinado programa o conjunto de programas → **El índice de rendimiento va a depender de la carga con la que se haga la comparación**

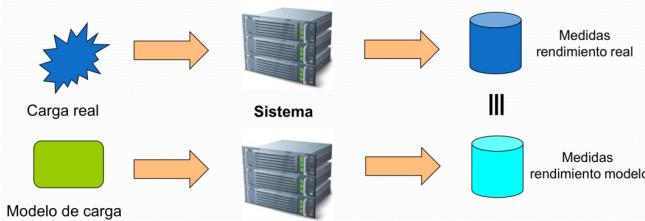
1.5 La carga real

- Difícil de utilizar en la evaluación de sistemas.
 - Varía a lo largo del tiempo: ¿Uso la carga de hoy, la de ayer...?
 - Resulta complicado reproducirla: Si la uso con un servidor muy lento, es posible que llegue una solicitud basada en una respuesta que todavía no ha llegado.
 - Interacciona con el sistema informático: Si un servidor contesta muy rápido, es posible que reciba las solicitudes antes o incluso más solicitudes que si fuera muy lento.
- Es más conveniente utilizar un modelo de la carga real como carga de prueba (test workload) para hacer comparaciones.



1.6 Modelo de carga

Representatividad del modelo de carga



Los modelos de carga son representaciones aproximadas de la carga que recibe un sistema informático. El modelo de la carga:

- Debe ser lo más representativo posible de la carga real.
- Debe ser lo más simple/compacto que sea posible (tiempos de medición y espacio en memoria razonables).

Principales estrategias para obtener modelos de carga

- Ajustar un modelo paramétrico "personalizado" a partir de la monitorización del sistema ante la carga real (caracterización de la carga).
- Usar programas de prueba que usen un modelo genérico de carga lo más similar posible al que se quiere reproducir (referenciación o benchmarking).

Caracterización de la carga

La forma más fácil para obtener un modelo de la carga a la que está sometido un servidor durante un determinado periodo de tiempo consiste en:

- **Identificar** los recursos que más demande la carga (CPU, memoria, discos, red, etc.)
- **Elegir** los parámetros característicos de dichos recursos (utilización de CPU, lecturas/escrituras que hay que hacer en cada disco, lecturas/escrituras a memoria, número de accesos a la red, etc)
- **Medir** el valor de dichos parámetros usando monitores de actividad (muestreo).
- **Analizar** los datos: medias, histogramas, agrupamiento o clustering, etc.
- **Generar** el modelo de carga seleccionando representantes de la carga (=solicitudes al servidor) junto con información estadística sobre su distribución temporal.

1.7 Referenciación (benchmarking)

Consiste en utilizar un programa o un conjunto de programas (benchmark programs) con el fin de comparar alguna característica del rendimiento entre equipos informáticos. Hay dos características principales que definen a un benchmark:

- La **carga de prueba** (test workload) específica con la que estresa el sistema evaluado.
- El **conjunto de reglas** que se deben seguir para la correcta ejecución, obtención y validación de los resultados.

Ventajas de usar benchmarking

- Existen muchos benchmarks diferentes para distintos tipos de servidores y cargas. Hay una **alta probabilidad** de encontrar uno que reproduzca unas condiciones parecidas a las que experimenta nuestro servidor.
- Las comparaciones entre el rendimiento de varios servidores son **justas** ya que todas las ejecuciones se realizan de forma idéntica siguiendo las **reglas del benchmark**.
- Muchos benchmarks permiten ajustar la carga de tal forma que podemos medir la **escalabilidad** de nuestro servidor.
- Al poder conocer tanto el rendimiento para un determinado benchmark que obtienen diferentes servidores como cómo están diseñados y configurados dichos servidores, obtenemos una **información muy valiosa sobre cómo diseñar y/o configurar nuestros propios servidores**.

1.8 Tipos de programas de benchmarkings



**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

Según la estrategia de medida

- Programas que miden el tiempo necesario para ejecutar una cantidad pre- establecida de tareas.
 - > La mayoría de benchmarks.
- Programas que miden la cantidad de tareas ejecutadas para un tiempo de cómputo pre-establecido.
 - > SLALOM: Mide la exactitud de la solución de un determinado problema que se puede alcanzar en 1 minuto de ejecución.
- Programas que permiten modificar sus parámetros para adaptarlos a cada sistema.
 - > TPC-C: Calcula cuántas consultas por segundo se realizan, de media, a un servidor de base de datos permitiendo aumentar tanto el no de usuarios como el tamaño de la base de datos. Exige un tiempo mínimo de respuesta para un tanto por ciento de usuarios.

Según la generalidad del test

- **Microbenchmarks** o benchmarks para componentes: estresan componentes o agrupaciones de componentes concretos del sistema: procesador, caché, memoria, discos, red, procesador+caché, procesador+compilador+memoria virtual, etc. (es difícil aislar un componente de otro y ver cómo lo gestiona el SO)

EJEMPLOS:

- **Whetstone (1976)**
 - Mide el rendimiento de las operaciones en coma flotante por medio de pequeñas aplicaciones científicas que usan sumas, multiplicaciones y funciones trigonométricas.
- **Linpack (1983)**
 - Mide el rendimiento de las operaciones en coma flotante a través de un algoritmo para resolver un sistema denso de ecuaciones lineales. El benchmark incorpora una rutina para comprobar que la solución a la que se llega es la correcta con un grado de exactitud prefijado. Se utiliza para confeccionar la lista de los 500 mejores supercomputadores del mundo.
- **Dhrystone (1984)**
 - Mide el rendimiento de operaciones con enteros, esencialmente por medio de operaciones de copia y comparación de cadenas de caracteres.
- **Macrobenchmarks** o benchmarks de sistema completo o de aplicación real: la carga intenta imitar situaciones reales (normalmente servidores con muchos clientes) típicas de algún área. P.ej. e-comercio, servidores web, servidores de ficheros, servidores de bases de datos, sistemas de ayuda a la decisión, paquetes ofimáticos + correo electrónico + navegación, etc.

1.9 SPEC (Standart Performance Evaluation Corporation)

Es una corporación sin ánimo de lucro cuyo propósito es establecer, mantener y respaldar la estandarización de benchmarks y herramientas para evaluar el rendimiento y la eficiencia energética de los equipos informáticos.

El paquete de microbenchmarks SPEC CPU 2017

Compuesto por cuatro conjuntos de benchmarks distintos :

- SPECspeed®2017 Integer (rendimiento en aritmética entera)
- SPECspeed®2017 Floating Point (rendimiento en coma flotante)
- SPECrate®2017 Integer (rendimiento en aritmética entera)
- SPECrate®2017 Floating Point (rendimiento en coma flotante)
 - > Speed: cuánto tarda en ejecutarse un programa (tiempo de respuesta).
 - > Rate: cuántos programas puedo ejecutar por unidad de tiempo (productividad).

¿Qué componentes se evalúan?

- Procesador (enteros o coma flotante según el caso).
- Sistema de memoria.
- Compilador (C, Fortran y C++).

SPEC CPU2017 se distribuye como una imagen ISO que contiene:

- Código fuente de todos los programas de benchmark.
- Data sets que necesitan algunos benchmarks para su ejecución.
- Herramientas varias para compilación, ejecución, obtención de resultados, validación y generación de informes.
- Documentación, incluyendo reglas de ejecución y de generación de informes.

El tiempo de ejecución depende del índice a obtener, la máquina en la que se ejecuta y cuántas copias o subprocesos se eligen.

Los programas dentro de SPECspeed 2017 tienen dos criterios generales: han de ser aplicaciones y deben tener portabilidad a muchas arquitecturas (Intel y AMD x86 & x86-64, Sun SPARC, IBM POWER e IA-64)

EJEMPLO: SPECspeed®2017 Integer: 10 programas (la mayoría en C y C++)

- 600.perlbench_s - Intérprete de Perl
- 657.xz_s - Utilidad de compresión
- 602.gcc_s - Compilador de C
- 623.xalancbmk_s - Conversión XML a HTML
- ...

Metric	Config Tested	Individual benchmarks	Full Run (Reportable)
SPECrate2017_int_base	1 copy	6 to 10 minutes	2.5 hours
SPECrate2017_fp_base	1 copy	5 to 36 minutes	4.8 hours
SPECspeed2017_int_base	4 threads	6 to 15 minutes	3.1 hours
SPECspeed2017_fp_base	16 threads	6 to 75 minutes	4.7 hours

Índices de prestaciones de SPECspeed 2017

También llamados índices SPEC:

- Aritmética entera: CPU2017IntegerSpeed_peak, CPU2017IntegerSpeed_base.
- Aritmética en coma flotante: CPU2017FP_Speed_peak, CPU2017FP_Speed_base.

Significado de “base” y “peak”:

- **Base:** Compilación en modo conservador: todos los programas escritos en el mismo lenguaje usan las mismas opciones de compilación.

- **Peak:** Rendimiento pico, permitiendo que cada uno escoja las opciones de compilación óptimas para cada programa.

Cálculo: Cada programa del benchmark se ejecuta 3 veces y se escoge el resultado intermedio (se descartan los 2 extremos). El índice SPEC es la media geométrica de las ganancias en velocidad con respecto a una máquina de referencia (en SPEC CPU2017 una Sun Fire V490).

EJEMPLO: Si llamamos t_i al tiempo que tarda la máquina a evaluar en ejecutar el programa de benchmark i -ésimo y t_{REF} lo que tardaría la máquina de referencia para ese programa (y hay 10 programas en el benchmark):

$$\text{índice SPEC} = \sqrt[10]{\frac{t_1^{REF}}{t_1} \times \frac{t_2^{REF}}{t_2} \times \dots \times \frac{t_{10}^{REF}}{t_{10}}}$$

All SPEC CPU2017 Integer Speed Results Published by SPEC										
These results have been submitted to SPEC; see the disclaimer before studying any results.										
Search published CPU2017 results										
Last update: 2017-10-19 11:49										
CPU2017 Integer Speed (7):										
Search in CPU2017 Integer Speed results										
Test Sponsor	System Name	Parallel	Base Threads	Enabled Cores	Enabled Chips	Threads/Core	Results	Base	Peak	
HPE	Integrity Superdome X (384 core, 2.20 GHz, Intel Xeon E7-8890 v4)	No	384	384	16	2	5.31	5.86		
	HTML CSV Text PDF PS Config									
HPE	ProLiant DL580 Gen9 (2.20 GHz, Intel Xeon E7-8890 v4)	No	96	96	4	1	5.35	5.95		
	HTML CSV Text PDF PS Config									
HPE	ProLiant ML350 Gen9 (2.20 GHz, Intel Xeon E5-2699 v4)	No	44	44	2	1	5.80	6.42		
	HTML CSV Text PDF PS Config									
HPE	ProLiant DL380 Gen10 (2.10 GHz, Intel Xeon Platinum 8176)	Yes	52	52	2	1	8.96	Not Run		
	HTML CSV Text PDF PS Config									
HPE	ProLiant DL380 Gen10 (2.10 GHz, Intel Xeon Platinum 8176)	Yes	56	56	2	1	9.16	Not Run		
	HTML CSV Text PDF PS Config									
Huawei	Huawei 2288H V5 (Intel Xeon Platinum 8180)	Yes	56	56	2	1	9.46	9.79		
	HTML CSV Text PDF PS Config									
Oracle Corporation	Sun Fire V490	Yes	1	8	4	1	1.00	Not Run		
	HTML CSV Text PDF PS Config									

1.10 Benchmarks de sistema completo

TCP

TPC (Transactions Processing Performance Council): Organización sin ánimo de lucro especializada en benchmarks relacionados con comercio electrónico y con bases de datos.

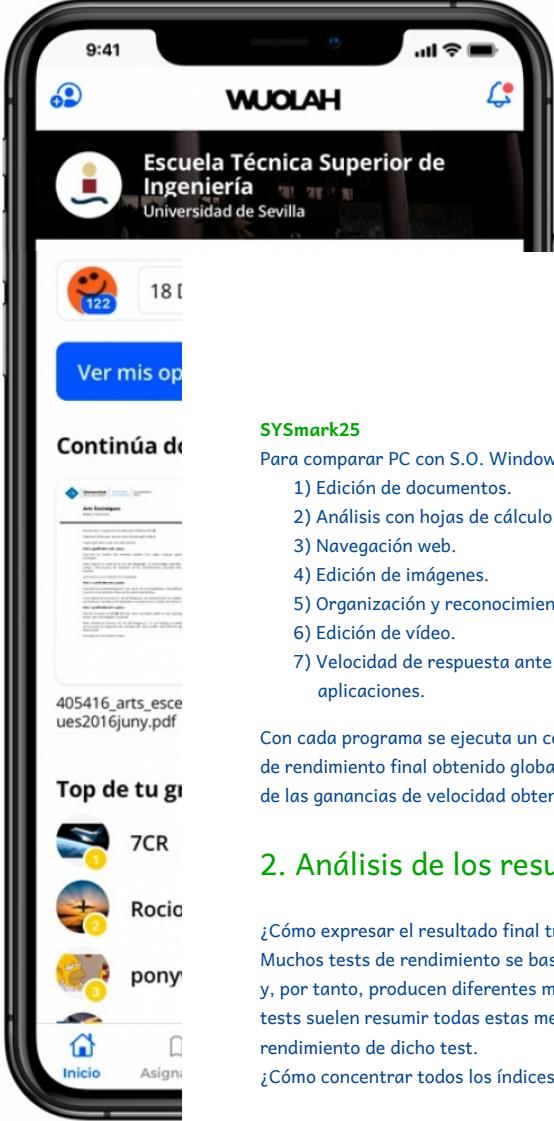
Principales benchmarks (todos son escalables):

- TPC-C: Tipo OLTP (on-line transaction processing). Simula una gran compañía que vende 100.000 productos y que tiene varios almacenes (configurable), cada uno a cargo de 10 zonas, con 3000 clientes/zona. Las peticiones involucran acceso a las bases de datos tanto locales como distribuidas (a veces el producto no está en el almacén más cercano).
- TPC-E: Tipo OLTP. Simula una correduría de bolsa en donde hay una única base de datos central.
- TPC-H, TPC-DS: Tipo DS (decision support). Se deben ejecutar consultas altamente complejas a una gran base de datos y analizar enormes volúmenes de datos (minería de datos, big-data).

Métricas: transacciones procesadas por unidad de tiempo (tps(tpm/tph) superando unos ciertos requisitos de tiempos de respuesta (ej. el 90% deben tener un tiempo de respuesta inferior a 1s). También: coste por transacción procesada (incluido mantenimiento) y consumo de potencia por transacción procesada.

SPEC

- **File Server: SFS2014:** Tiempos de respuesta y productividades de servidores de ficheros.
- **High Performance Computing, OpenMP, MPI, OpenCL**
 - SPEC MPI2007: Message Passing Interface (MPI).
 - SPEC OMP2012: Open MultiProcessing (OpenMP).
 - SPEC ACCEL: OpenCL y OpenACC
- **JAVA Cliente/Servidor**
 - SPECjEnterprise2010: Java Enterprise Edition (JEE).
 - SPECjms2007: Java Message Service (JMS).
 - SPECjvm2008: Java Runtime Environment (JRE).
- **Cloud: SPEC Cloud_IaaS 2018** (Servicios en la nube)
- **Virtualization: SPECvirt_sc2013** (Virtualización en Centros de Procesamiento de Datos).
- **Consumo de potencia: SPECpower_ssj2008** (Rendimiento de un servidor ejecutando aplicaciones JAVA frente al consumo de potencia).



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

Continúa de



Top de tu grupo



SYSmark25

Para comparar PC con S.O. Windows. Considera la carga en varios escenarios:

- 1) Edición de documentos.
- 2) Análisis con hojas de cálculo.
- 3) Navegación web.
- 4) Edición de imágenes.
- 5) Organización y reconocimiento de imágenes.
- 6) Edición de vídeo.
- 7) Velocidad de respuesta ante ejecución simultánea de múltiples aplicaciones.

Programas utilizados:

- Adobe: Acrobat, Photoshop, Lightroom, Premiere Pro.
- Microsoft: Word, Excel, PowerPoint, Outlook.
- Google Chrome.
- Corel WinZip.
- Audacity installer.
- Autolt (AU3 scripting files).

Con cada programa se ejecuta un conjunto de tareas de acuerdo con un modelo de comportamiento de un usuario "habilidoso". El índice de rendimiento final obtenido global para todos los escenarios se calcula igual que el índice SPEC, es decir, mediante la media geométrica de las ganancias de velocidad obtenidas con respecto a una determinada máquina de referencia.

2. Análisis de los resultados de un test de rendimiento

¿Cómo expresar el resultado final tras la ejecución de un test de rendimiento?

Muchos tests de rendimiento se basan en la ejecución de diferentes programas y, por tanto, producen diferentes medidas de rendimiento. Sin embargo, estos tests suelen resumir todas estas medidas en un único valor: el índice de rendimiento de dicho test.

¿Cómo concentrar todos los índices en uno solo? -> Método habitual de síntesis: uso de algún tipo de **media**.



2.1 La media aritmética

Dado un conjunto n de medias, t_1, \dots, t_n , definimos su **media aritmética**:

$$\bar{t} = \frac{1}{n} \sum_{k=1}^n t_k$$

Si no todas las medidas tienen la misma importancia, se puede asociar a cada medida tk un peso wk, obteniendo la **media aritmética ponderada**:

$$\bar{t}_w = \sum_{k=1}^n w_k \times t_k, \text{ con } \sum_{k=1}^n w_k = 1$$

Si t_k es el tiempo de ejecución del programa de benchmark k-ésimo en la máquina a testar, w_k , podría escogerse, por ejemplo, inversamente proporcional a dicho tiempo de ejecución en una determinada máquina de referencia:

$$w_k \equiv \frac{C}{t_k^{\text{REF}}} \rightarrow C = \frac{1}{\sum_{k=1}^n 1/t_k^{\text{REF}}}$$

2.2 La media geométrica

Dado un conjunto n de medidas S_1, \dots, S_n , definimos su **media geométrica**:

$$S_g = \sqrt[n]{\prod_{k=1}^n S_k} = \left(\prod_{k=1}^n S_k \right)^{1/n}$$

Propiedad: cuando las medidas son ganancias en velocidad (speedups) con respecto a una máquina de referencia, este índice mantiene el mismo orden en las comparaciones independientemente de la máquina de referencia elegida. Usado en los benchmarks de SPEC y SYSMARK.

$$\text{SPEC}(M) = \sqrt[n]{\frac{t_1^{\text{REF}}}{t_1^M} \times \frac{t_2^{\text{REF}}}{t_2^M} \times \dots \times \frac{t_n^{\text{REF}}}{t_n^M}} = \frac{\sqrt[n]{t_1^{\text{REF}} \times t_2^{\text{REF}} \times \dots \times t_n^{\text{REF}}}}{\sqrt[n]{t_1^M \times t_2^M \times \dots \times t_n^M}}$$

Si la máquina de referencia es la misma, el valor no cambia entre máquinas.

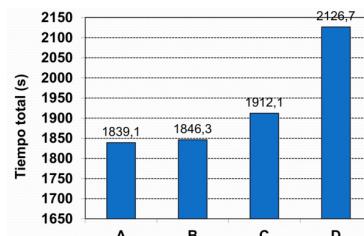
$$\text{SPEC}(M1) > \text{SPEC}(M2) \iff \sqrt[n]{t_1^{M1} \times t_2^{M1} \times \dots \times t_n^{M1}} < \sqrt[n]{t_1^{M2} \times t_2^{M2} \times \dots \times t_n^{M2}}$$

Ejemplo de comparación:

Programa	t_{REF} (s)	t^A (s)	t^B (s)	t^C (s)	t^D (s)
1	1400	141	170	136	134
2	1400	154	166	215	25
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	1200	179	189	199	245
7	1300	120	131	87,7	75,5
8	300	151	158	138	192
9	1100	93,5	122	88	118
10	1900	133	173	118	142
11	1500	173	170	179	240
12	3000	243	100	100	150
Suma	17000	1839,1	1846,3	1912,1	2126,7

Con tiempos: la máquina más rápida es “A” ya que es la que tarda menos en ejecutar, uno tras otro, todos los programas del benchmark (1839,1 segundos).

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.



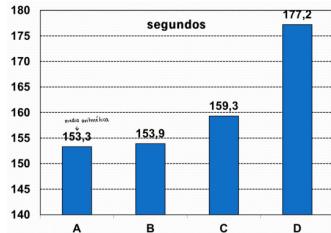
Con la media aritmética: la máquina que ejecuta los programas del benchmark, uno tras otro, en menor tiempo es la de menor media aritmética de los tiempos de ejecución.

$$\bar{t}_A = \frac{1}{12} \sum_{k=1}^{12} t_k^A = 153,3 \text{ s}$$

$$\bar{t}_B = \frac{1}{12} \sum_{k=1}^{12} t_k^B = 153,9 \text{ s}$$

$$\bar{t}_C = \frac{1}{12} \sum_{k=1}^{12} t_k^C = 159,3 \text{ s}$$

$$\bar{t}_D = \frac{1}{12} \sum_{k=1}^{12} t_k^D = 177,2 \text{ s}$$



Con la media ponderada:

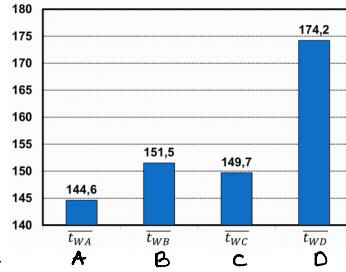
Prog	t_{REF} (s)	w_k
1	1400	0,06
2	1400	0,06
3	1100	0,08
4	1800	0,05
5	1000	0,09
6	1200	0,07
7	1300	0,07
8	300	0,30
9	1100	0,08
10	1900	0,05
11	1500	0,06
12	3000	0,03
Suma	17000	1

$$w_k \equiv \frac{C}{t_k^{REF}}$$

$$C = \frac{1}{\sum_{k=1}^n 1/t_k^{REF}} = 88,77 \text{ s}$$

$$\bar{t}_{WA} = \sum_{k=1}^{12} w_k \times t_k^A = 144,6 \text{ s}$$

↳ Igual para \bar{t}_{WB} , \bar{t}_{WC} y \bar{t}_{WD}



Según este criterio, la máquina “más rápida” sería la de menor tiempo medio ponderado de ejecución. Nótese que esta ponderación depende, en este ejemplo, de la máquina de referencia.

Con la media geométrica de speedups: Calculamos la ganancia en velocidad de cada máquina con respecto a la máquina de referencia (tal y como lo hacen SPEC y Sysmark):

Programa	t_{REF} (s)	S^A speedup	S^B speedup	S^C speedup	S^D speedup
1	1400	9,9	8,2	10,3	10,4
2	1400	9,1	8,4	6,5	56,0
3	1100	11,4	11,7	7,5	5,5
4	1800	6,6	6,4	4,2	3,4
5	1000	11,9	11,1	12,9	12,3
6	1200	6,7	6,3	6,0	4,9
7	1300	10,8	9,9	14,8	17,2
8	300	2,0	1,9	2,2	1,6
9	1100	11,8	9,0	12,5	9,3
10	1900	14,3	11,0	16,1	13,4
11	1500	8,7	8,8	8,4	6,3
12	3000	12,3	30,0	30,0	20,0
M. Geom.		8,78	8,66	8,97	9,00

¿A quién beneficia la decisión de usar la media geométrica de speedups?

J8	:	X ✓ fx	=MEDIA.GEOM(J2:J5)						
A	B	C	D	E	F	G	H	I	J
Prog-Bench.	tREF(s)	tA(s)	tB(s)	tC(s)	tD(s)	tREF/tA	tREF/tB	tREF/tC	tREF/tD
1	200	100	99	1	1	2,00	2,02	200,0	200,0
2	200	100	101	133	1	2,00	1,98	1,50	200,0
3	200	100	100	133	1	2,00	2,00	1,50	200,0
4	200	100	100	133	397	2,00	2,00	1,50	0,50
5	800	400	400	400	400				
6	Suma					Media Geométrica	2,0000	2,0001	5,11
7									44,81
8									

Se premian las mejoras sustanciales. No se castigan empeoramientos no tan sustanciales. Debemos ser MUY cuidadosos con las comparaciones y saber qué estamos haciendo realmente.

Conclusiones del análisis

- Intentar reducir un conjunto de medidas de un test de rendimiento a un solo “valor medio” final no es una tarea trivial.
- La media aritmética de los tiempos de ejecución es una medida fácilmente interpretable e independiente de ninguna máquina de referencia. El menor valor nos indica la máquina que ha ejecutado el conjunto de programas del test, uno tras otro, en un tiempo menor.
- La media aritmética ponderada nos permite asignar más peso a algunos programas que a otros. Esta ponderación debería realizarse, idealmente, según las necesidades del usuario. Si se hace de forma dependiente de los tiempos de ejecución de una máquina de referencia, la elección de ésta puede influir significativamente en los resultados.
- La media geométrica de las ganancias en velocidad con respecto a una máquina de referencia es un índice de interpretación compleja cuya comparación no depende de la máquina de referencia. Premia mejoras sustanciales con respecto a algún programa del test y no castiga al mismo nivel los empeoramientos.

3. Comparación de prestaciones en presencia de aleatoriedad

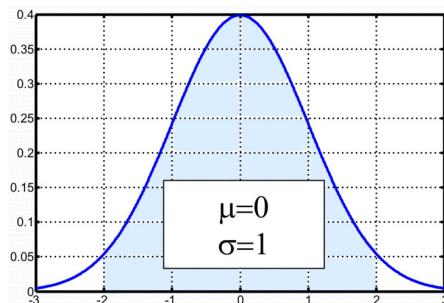
3.1 Distribución normal

Independientemente de qué índice se escoja, un buen ingeniero debería, en primer lugar, determinar si las diferencias entre las medidas obtenidas por un test de rendimiento en presencia de aleatoriedad son estadísticamente significativas [2] Necesitaremos repasar algunos conceptos de estadística.

- **Distribución normal:** Es una distribución de probabilidad caracterizada por su media μ y su varianza σ^2 cuya función de probabilidad viene dada por:

$$Prob(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

La probabilidad de obtener un elemento en el rango $[\mu - 2\sigma, \mu + 2\sigma]$ es del 95%



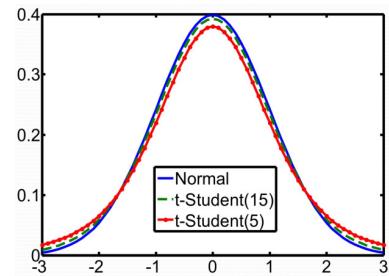
Teorema del límite central: la media de un conjunto grande de muestras aleatorias de cualquier distribución e independientes entre sí pertenece una distribución normal.

3.2 Distribución t de Student

Si extraemos n muestras $\{d_1, d_2, \dots, d_n\}$ pertenecientes a una distribución Normal de media d_{real} , y calculo la siguiente medida (=estadístico):

$$t_{exp} = \frac{\bar{d} - \bar{d}_{real}}{s/\sqrt{n}} \quad / \quad \bar{d} = \frac{\sum_{i=1}^n d_i}{n} \quad / \quad s = \sqrt{\frac{\sum_{i=1}^n (d_i - \bar{d})^2}{n-1}}$$

$\left\{ \begin{array}{l} Prob(t) = \frac{\Gamma(\nu/2)}{\sqrt{\nu\pi}\Gamma(\nu/2)} \left(1 + \frac{t^2}{\nu}\right)^{-\nu/2} \\ s/\sqrt{n} \equiv \text{Error estándar} \end{array} \right.$



EJEMPLO 1: Test de rendimiento entre A y B

Tiempos de ejecución (en segundos) de 6 programas (P1...P6) en dos máquinas diferentes (A y B) en condiciones donde puede haber alta aleatoriedad.

Programa	t _A (s)	t _B (s)	d = t _A -t _B (s)
P1	142	100	42
P2	139	92	47
P3	152	128	24
P4	112	82	30
P5	156	148	8
P6	166	171	-5

$$\bar{d} = 24,3 \text{ s} \quad s = 19,9 \text{ s} \quad s/\sqrt{n} = 8,12 \text{ s}$$

$\bar{t}_A = 144,5 \text{ s}$
 $\bar{t}_B = 120,2 \text{ s}$
 ¿Son significativas estas diferencias?

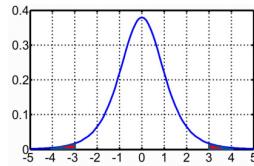
Si partimos de la hipótesis (H_0 , hipótesis nula) de que las máquinas tienen rendimientos equivalentes, entonces las diferencias se deben a una suma (media) de factores aleatorios independientes. En ese caso serán muestras de una distribución normal de media cero ($d_{real} = 0$).

Por tanto $t_{exp} = \frac{\bar{d}}{s/\sqrt{n}} = \frac{24,3 \text{ s}}{8,12 \text{ s}} = 2,99$ pertenecerá a una distribución t de Student con $6-1=5$ grados de libertad.

¿Qué probabilidad hay de que esto sea realmente así?

3.3 Nivel o Grado de Significatividad

Distribución t de Student con 5 grados de libertad (T5)



$$\begin{aligned} P\text{-value} &= P(|t| \geq |t_{exp}|) \text{ en } T_{n-1} \\ &= 2 \times P(t \leq -|t_{exp}|) \text{ en } T_{n-1} \\ &= \text{DISTR.T.2C}(2,99;5) = 0,03 \text{ (Excel).} \\ &= \text{DISTR.T}(2,99;5;2) = 0,03 \text{ (Calc).} \\ &= 2 \cdot \text{tcdf}(-2,99,5) = 0,03 \text{ (Matlab).} \end{aligned}$$

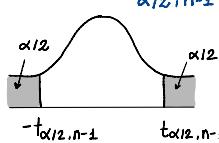
La probabilidad de obtener un valor de $|t|$ igual o superior a 2,99 de una distribución t de Student con 5 grados de libertad es de 0,03 (P-value (Valor-P)=0,03). ¿Es eso mucho o poco? Debemos definir un umbral: **nivel o grado de significatividad α** . Normalmente, $\alpha=0,05$ (5%).

Conclusión: Si P-value < α diremos que, para un grado de significatividad α o para un **nivel de confianza** $(1-\alpha) \times 100$ (normalmente 95%), las máquinas tienen rendimientos estadísticamente diferentes. En ese caso, B sería, de media, 1,2 veces más rápida que A en ejecutar cada programa ($144,5/120,2 = 1,2$). En caso contrario, no podríamos descartar la hipótesis de que las máquinas tengan rendimientos equivalentes.

3.4 Intervalos de confianza para t_{exp}

Para un nivel de significatividad α (típicamente $0,05 = 5\%$), buscamos el valor $t_{\alpha/2, n-1}$ que cumpla $\text{Prob}(|t| > t_{\alpha/2, n-1}) = \alpha$ o equivalente:

$$\text{Prob}(-t_{\alpha/2, n-1} \leq t \leq t_{\alpha/2, n-1}) = 1 - \alpha$$



Diremos que para un nivel de confianza $1-\alpha$, para aceptar H_0 el valor de t_{exp} debería situarse en el intervalo: $[-t_{\alpha/2, n-1}, t_{\alpha/2, n-1}]$

A dicho intervalo se le denomina intervalo de confianza de la medida para un nivel de significatividad α . Teniendo en cuenta que:

$$\text{Prob}(-t_{\alpha/2, n-1} \leq t \leq t_{\alpha/2, n-1}) = 1 - 2 \times \text{Prob}(t \leq -t_{\alpha/2, n-1}) = 1 - 2 \times \text{Prob}(t > t_{\alpha/2, n-1})$$

es fácil demostrar que $t_{\alpha/2, n-1}$ cumple que $\text{Prob}(t \leq -t_{\alpha/2, n-1}) = \text{Prob}(t > t_{\alpha/2, n-1}) = \alpha/2$

En el **EJEMPLO 1**, para un nivel de significatividad de $\alpha = 0,05$, buscamos $t_{\alpha/2, n-1}$ tal que $\text{Prob}(t \leq t_{\alpha/2, n-1}) = \alpha/2 = 0,025$ para una distribución t de Student con 5 grados de libertad. Eso lo podemos conseguir, por ejemplo, consultando las tablas estadísticas.

Dicho de otra manera, si las diferencias entre los tiempos de ejecución de ambas máquinas se debieran a factores aleatorios, existiría un 95% de probabilidad de que:

$$t_{exp} = \frac{\bar{d}}{s/\sqrt{n}} \text{ se encuentre en el rango } [-t_{\alpha/2, n-1}, t_{\alpha/2, n-1}]$$

→ En el **EJEMPLO 1** $[-t_{0,025,5}, t_{0,025,5}] = [-2,57, 2,57]$. Como 2,99 no pertenece a dicho rango, rechazamos la hipótesis de que ambas máquinas tienen un rendimiento equivalente con el 95% de confianza.

3.5 Intervalos de confianza para dreal

Acabamos de ver que si las diferencias entre los tiempos de ejecución de ambas máquinas se debieran a factores aleatorios, existiría un 95% de probabilidad de que t_{exp} se encuentre en el rango $[-t_{\alpha/2, n-1}, t_{\alpha/2, n-1}] = [-2,57, 2,57]$

Como

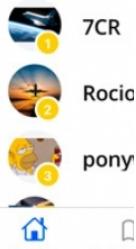
$$t_{exp} = \frac{\bar{d} - \bar{d}_{real}}{s/\sqrt{n}} \in \left[-\left| t_{\alpha/2, n-1} \right|, \left| t_{\alpha/2, n-1} \right| \right] = [-2,57, 2,57]$$

sin más que identificar t_{exp} con los valores límite $\pm t_{\alpha/2, n-1}$ sabemos que, de ser H_0 cierta, habrá un 95% de probabilidad de que el valor medio real dreal de las diferencias entre los tiempos se encuentre en el intervalo:

$$\bar{d}_{real} \in \left[\bar{d} - \frac{s}{\sqrt{n}} \times \left| t_{\alpha/2, n-1} \right|, \bar{d} + \frac{s}{\sqrt{n}} \times \left| t_{\alpha/2, n-1} \right| \right] = 24,3 \mp 20,9 = [3,4, 45,2] \text{ s}$$

Y el problema se transforma simplemente en comprobar si ese valor medio real dreal puede o no ser cero.

En nuestro ejemplo, como el intervalo no incluye el cero, concluiremos una vez más que la **hipótesis de que ambas máquinas pueden tener rendimientos equivalentes no es cierta al 95% de confianza**.



RESUMEN

- Partimos de:

	t_A	t_B	$d_i = t_{A_i} - t_{B_i}$
P_1	t_{A_1}	t_{B_1}	d_1
P_2	t_{A_2}	t_{B_2}	d_2
...
P_n	t_{A_n}	t_{B_n}	d_n

- Rechazamos H_0 para un nivel de confianza $(1-\alpha) \times 100 (\%)$ si:

- Método 1: $p\text{-value} < \alpha$. Siendo $p\text{-value} = P(|t| \geq |t_{exp}|)$ en $T_{n-1} \approx \text{Prob}(H_0 \text{ es cierta})$.

- Método 2: $t_{exp} \notin [-t_{\alpha/2, n-1}, t_{\alpha/2, n-1}]$. Siendo $t_{\alpha/2, n-1}$ el valor que hace que $\text{Prob}(t \leq -t_{\alpha/2, n-1}) = \frac{\alpha}{2}$ para una distribución t de Student con $n-1$ grados de libertad.

- Método 3: $0 \notin [\bar{d} - \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1}, \bar{d} + \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1}]$ (Intervalo de confianza para d_{real}).

EJEMPLO 2: ¿Influye el parámetro swappiness del S.O. en este servidor?

Productividades (en páginas web/s) obtenidas por el servidor en 5 experimentos diferentes para dos valores diferentes (A y B) del parámetro proxy_cache_min_use de Nginx en condiciones donde puede haber alta aleatoriedad.

Experimento	X_A (pág/s)	X_B (pág/s)	$d = X_A - X_B$ (pág/s)
Exp1	23	15	8
Exp2	28	22	6
Exp3	19	20	-1
Exp4	29	27	2
Exp5	36	39	-3

- Hago la siguiente hipótesis:

Rendimiento A \equiv Rendimiento B, es decir, $d_i \sim N(\bar{d}_{real}, \sigma^2)$ con $\bar{d}_{real} = 0$.

- Calculo $t_{exp} = \frac{\bar{d} - \bar{d}_{real}}{s/\sqrt{n}} \sim T_{n-1}$

Usando como criterio la media aritmética ($X_A=27$ págs/s, $X_B=25$ págs/s) parece que el parámetro A obtiene mejor productividad que el B pero, ¿son significativas las diferencias para un nivel de confianza del 95%? $(1-\alpha) \times 100 = 95\% \rightarrow$ grado de significatividad $\alpha = 0,05$.

Experimento	X_A (pág/s)	X_B (pág/s)	$d = X_A - X_B$ (pág/s)
Exp1	23	15	8
Exp2	28	22	6
Exp3	19	20	-1
Exp4	29	27	2
Exp5	36	39	-3

$$\bar{d} = \frac{\sum_{i=1}^n d_i}{n} = \frac{8+6-1+2-3}{5} = 2,4 \text{ págs/s}$$

$$s = \sqrt{\frac{\sum_{i=1}^n (d_i - \bar{d})^2}{n-1}} = \sqrt{\frac{\sum_{i=1}^n d_i^2 - n \cdot \bar{d}^2}{n-1}} = 4,6 \text{ págs/s}$$

$$\frac{s}{\sqrt{n}} = \frac{4,6}{\sqrt{5}} = 2,06 \text{ págs/s} \quad t_{exp} = \frac{\bar{d}}{s/\sqrt{n}} = \frac{2,4}{2,06} = 1,16$$

Método 1:

- $p\text{-value} = P(|t| \geq |t_{exp}|)$ en $T_{n-1} = P(|t| \geq |1,16|)$ en $T \approx \text{Prob}(H_0 \text{ podría ser cierta})$.
- Uso Calc (por ejemplo): $p\text{-value} = \text{DISTR.T}(1,16; 4; 2) = 0,31$.
- Como $p\text{-value} > \alpha (0,31 > 0,05)$ no podemos rechazar la hipótesis H_0 al 95% de nivel de confianza (los parámetros A y B sí podrían tener rendimientos equivalentes).

Método 2 (intervalo de confianza para t_{exp}):

- Calculo $t_{\alpha/2, n-1}$ que es el valor que hace que $\text{Prob}(t \leq -t_{\alpha/2, n-1}) = \frac{\alpha}{2}$ para una distribución t de Student con $n-1$ grados de libertad.
- Miro en la tabla del p-valor sabiendo que df (grado libertad) = 4 y $\alpha = 0,05 \rightarrow p\text{-value} = 2,78$.
- $t_{exp} = 1,16 \in [-2,78, 2,78]$ no podemos rechazar H_0 al 95% de confianza.

Método 3 (intervalo de confianza para \bar{d}_{real}):

$$\left[\bar{d} - \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1}, \bar{d} + \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1} \right] = [2,4 - 2,06 \times 2,78, 2,4 + 2,06 \times 2,78] = [-3,3, 8,1] \text{ págs/s.}$$

- Como $0 \in [-3,3, 8,1]$ no podemos rechazar la hipótesis H_0 al 95% de confianza.

Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

3.8 Test t con JASP

Paired Samples T-Test		degrees of freedom						
Measure 1	Measure 2	t	df	p	Mean Difference	Standard Error $\frac{s}{\sqrt{n}}$	95% CI for Mean Difference	
EJ1_A	- EJ1_B	2.991	5	0.030	24.333	8.135	3.422	45.245
EJ2_A	- EJ2_B	1.163	4	0.310	2.400	2.064	-3.331	8.131

Intervalo de confianza (95%) para \bar{d}_{real} : $\bar{d} \pm \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1}$

Hipótesis: Realizamos n medidas $\{d_1, d_2, \dots, d_n\}$ de un mismo fenómeno (p.ej. tiempos de ejecución de un programa, tiempos acceso de un disco duro, productividades de red,...). Si éstas pueden diferir debido a una suma de efectos aleatorios, podemos suponer que se distribuyen según una normal de media \bar{d}_{real} , que es el valor que buscamos. En ese caso, sabemos que:

$t_{exp} = \frac{\bar{d} - \bar{d}_{real}}{s/\sqrt{n}}$ pertenece a la distribución t-Student con $n-1$ grados de libertad, siendo \bar{d} y s la media y la desviación típica muestrales, respectivamente.

Por tanto, hay un $(1-\alpha) * 100\%$ de probabilidad de que el valor medio real \bar{d}_{real} se encuentre en el intervalo: $\bar{d} \pm \frac{s}{\sqrt{n}} \cdot t_{\alpha/2, n-1}$

Utilidad: Podemos usar esta información para determinar un intervalo de confianza para \bar{d}_{real} , y no quedarnos simplemente con el valor medio muestral.

EJEMPLO:

Queremos determinar un intervalo de confianza del 95% para el tiempo medio de escritura de un determinado fichero en un disco duro. Para ello, se han realizado $n=8$ medidas experimentales:

#exp	d (ms)	$\bar{d} = \frac{\sum_{i=1}^n d_i}{n} = 820ms$	$s = \sqrt{\frac{\sum_{i=1}^n (d_i - \bar{d})^2}{n-1}} = 14ms$	<ul style="list-style-type: none"> En Excel, haciendo: ABS(INV.T(alfa/2;n-1)). En Calc, DISTR.T.INV(alfa;n-1). En Matlab, haciendo: abs(tinv(alfa/2,n-1)).
1	835			
2	798			
3	823			
4	803			
5	834			
6	825			
7	813		$ t_{\alpha/2, n-1} = t_{0,025, 7} = 2,36$	
8	829			

df	0.20	0.10	0.05	0.02	0.01	0.001
1	3.0777	6.3138	12.7062	31.8205	63.6567	636.6192
2	1.8856	2.9200	4.3027	6.9646	9.9248	31.5991
3	1.6377	2.3534	3.1824	4.5407	5.8409	12.9240
4	1.5332	2.1318	2.7764	3.7469	4.6041	8.6103
5	1.4759	2.0150	2.5706	3.3649	4.0321	6.8688
6	1.4396	1.9432	2.4469	3.1427	3.7074	5.9588
7	1.4149	1.8946	2.3646	2.9980	3.4995	5.4079

Por tanto, hay un 95% ($\alpha=0.05$) de probabilidad de que el tiempo medio de escritura **real** de ese fichero se encuentre en el intervalo:

$$\bar{d} \pm \frac{s}{\sqrt{n}} t_{\alpha/2, n-1} = 820 \pm \frac{14}{\sqrt{8}} t_{0.05, 7} = [808, 832]ms$$

4. Diseño de experimentos de comparación de rendimiento

4.1 Planteamiento del problema

Supongamos que queremos determinar cuáles de los siguientes factores afectan significativamente al rendimiento de un determinado servidor:

1. Sistema Operativo: Windows Server, CentOS, Debian, Ubuntu.
2. Memoria RAM: 32GB, 64GB, 128GB.
3. Discos duros: SAS, SATA, IDE (=P-ATA).

Y, en el caso de que afecten, cuál de los niveles del factor es significativamente mejor que el resto.

¿Qué experimentos debemos diseñar para ello y cómo debemos analizar los resultados?

Terminología

- Variable respuesta o dependiente (métrica): El índice de rendimiento que usamos para las comparaciones. P.ej. tiempos de respuesta (R), productividades (X).
- Factor: Cada una de las variables que pueden afectar a la variable respuesta. P.ej. sistema operativo, tamaño de memoria, tipo de disco duro, tipo de procesador, número de microprocesadores, número de cores, tamaño de cada caché, compilador, algún parámetro configurable del S.O. o del servidor, etc.
- Nivel: Cada uno de los valores que puede asumir un factor. P.ej. para un S.O.: Windows, CentOS, Debian, Ubuntu; para un tipo de disco duro: SATA, IDE, SAS; para un parámetro del sistema operativo: ON, OFF, etc.
- Interacción: Una interacción ocurre cuando el efecto de un factor cambia para diferentes niveles de otro factor. P.ej. el hecho de usar un tipo determinado de S.O. puede afectar a cómo de importante sea usar una mayor cantidad de memoria DRAM.

Tipos de diseños experimentales

• **Diseños con un solo factor:** Se utiliza una configuración determinada como base y se estudia un factor cada vez, midiendo los resultados para cada uno de sus niveles. Problema: solo válida si descartamos que haya interacción entre factores. Número total de experimentos = $1 + \sum_{i=1}^k (n_i - 1)$ donde k es el número de factores y n_i el número de niveles del factor i. En nuestro ejemplo, habría que hacer 8 experimentos.

• **Diseños multi-factoriales completos:** Se prueba cada posible combinación de niveles para todos los factores. Ventaja: se analizan las interacciones entre todos los factores. Número total de experimentos = $\prod_{i=1}^k n_i$. En nuestro ejemplo: 36 experimentos.

• **Diseños multi-factoriales fraccionados:** Término medio entre los anteriores. No todas las interacciones se verán reflejadas en los resultados, solo las de las interacciones que se consideren más probables.

- Todos ellos se pueden realizar con diferentes niveles de repetición: a) sin repeticiones, b) con todos los experimentos repetidos el mismo número de veces, c) con un número de repeticiones diferentes para cada nivel o cada factor.

3.2 Diseños con un solo factor

EJEMPLO: Para el servidor principal de nuestra empresa, queremos saber si la elección del tipo de disco duro afecta al rendimiento. Para ello, se ha escogido tres discos duros con tres interfaces distintas: SAS, SATA e IDE y se ha realizado un experimento que consiste en ejecutar, en condiciones reales y en presencia de aleatoriedad, un conjunto de programas usados habitualmente por el servidor y medir el tiempo de ejecución. Este experimento se ha repetido 5 veces:

#Exp.	SAS (s)	SATA (s)	IDE (s)
1	103	115	143
2	97	102	134
3	123	120	139
4	106	115	135
5	116	122	129
Medias	109.0	114.8	136.0
Efectos (ε_j)	-10.9	-5.1	16.1

$$m_{\text{global}} = 119.95$$

¿Tiene influencia el factor “interfaz del disco duro” sobre el rendimiento? ¿Son las diferencias entre los discos duros significativas?

-> **Test ANOVA de un factor.**

3.3 Análisis de la Varianza (ANOVA) de un factor

$$\text{Modelo: } y_{ij} = m_{\text{global}} + \varepsilon_j + r_{ij} \quad i=1, \dots, n_{\text{rep}}; \quad j=1, \dots, n_{\text{niv}}$$

y_{ij} : Las observaciones. En nuestro caso los tiempos de ejecución obtenidos en cada prueba. El índice j recorre los distintos niveles del factor cuya influencia se quiere medir (en nuestro caso hay $n_{\text{niv}}=3$ niveles: SAS, SATA e IDE). El índice i recorre las distintas repeticiones para cada uno de esos niveles (en nuestro caso, $n_{\text{rep}}=5$ repeticiones).

$$m_{\text{global}} : m_{\text{global}} = \frac{1}{n_{\text{rep}} \times n_{\text{niv}}} \sum_{i=1}^{n_{\text{rep}}} \sum_{j=1}^{n_{\text{niv}}} y_{ij} \quad \rightarrow \text{Media global de las operaciones}$$

$$\varepsilon_j : \text{Efecto al nivel j-ésimo: } \varepsilon_j = \frac{1}{n_{\text{rep}}} \sum_{i=1}^{n_{\text{rep}}} y_{ij} - m_{\text{global}} \quad / \quad \text{Se cumple que: } \sum_{j=1}^{n_{\text{niv}}} \varepsilon_j = 0$$

r_{ij} : Perturbaciones o error experimental (ruido). Deben cumplir:

- Que tengan varianza constante, independiente del nivel (si no de exp./nivel no es homogéneo).
- Que su distribución sea normal.

Si no se cumplen, hay otros test alternativos: test de Kruskal-Wallis, test de Friedman.

La principal pregunta que intenta contestar el test ANOVA es: ¿Tiene influencia el factor sobre la variable respuesta, es decir, algún ε_j es distinto de cero?

El método de ANOVA se basa en descomponer la varianza de las muestras en:

$$\sum_{i=1}^{n_{\text{rep}}} \sum_{j=1}^{n_{\text{niv}}} (y_{ij} - m_{\text{global}})^2 = n_{\text{rep}} \sum_{j=1}^{n_{\text{niv}}} (\varepsilon_j)^2 + \sum_{i=1}^{n_{\text{rep}}} \sum_{j=1}^{n_{\text{niv}}} (r_{ij})^2$$

Utilizando la notación abreviada:

$$\text{SST} = \text{SSA} + \text{SSE}$$

- SST= Varianza total de las muestras. (Sum-of-Squares Total)
- SSA= Varianza explicada por los efectos o alternativas (intergrupos). (Sum-of-Squares Alternatives)
- SSE= Varianza residual o del error (intragrupos) (Sum-of-Squares Error)

El objetivo es contrastar la hipótesis (H_0) de que el factor no influye sobre los resultados ($\varepsilon_j \simeq 0 \forall j = 1 \dots n_{\text{niv}}$).

Si esto es cierto, entonces el resultado de hacer...

$$F_{\text{exp}} \equiv \frac{\text{SSA}/(n_{\text{niv}} - 1)}{\text{SSE}/(n_{\text{niv}} \times (n_{\text{rep}} - 1))} \sim F_{n_{\text{niv}} - 1, n_{\text{niv}} \times (n_{\text{rep}} - 1)}$$

... debería ser una muestra de una distribución F de Snedecor con $n_{\text{niv}} - 1$ grados de libertad en el numerador y $n_{\text{niv}} \times (n_{\text{rep}} - 1)$ en el denominador.

En nuestro ejemplo:

$$\begin{aligned} \text{SST} &= 2809 \\ \text{SSA} &= 2020 \\ \text{SSE} &= 789 \end{aligned}$$

$$F_{\text{exp}} \equiv \frac{\frac{\text{SSA}}{n_{\text{niv}} - 1}}{\frac{\text{SSE}}{(n_{\text{niv}} \times (n_{\text{rep}} - 1))}} = \frac{\frac{2020}{3 - 1}}{\frac{789}{(3 \times (5 - 1))}} = 15,37$$

¿Qué probabilidad hay de que la muestra 15.37 o superior se haya extraído de una distribución $F_{2,12}$?

$P\text{-value} = P(F \geq 15.37; 2, 12) = 0.00049$. Identifico ese valor como la probabilidad de que la hipótesis H_0 de que el valor del factor no influye pueda ser cierta:

- Si la probabilidad es menor que $\alpha = 0.05$ diremos que descartamos la hipótesis de que el factor no influya a un $(1-\alpha) \times 100\% = 95\%$ de confianza.
- Si el factor influye, a continuación (análisis post-hoc) comparamos las medias de cada nivel unas con otras usando un test t: prueba de múltiples rangos o de comparaciones múltiples.

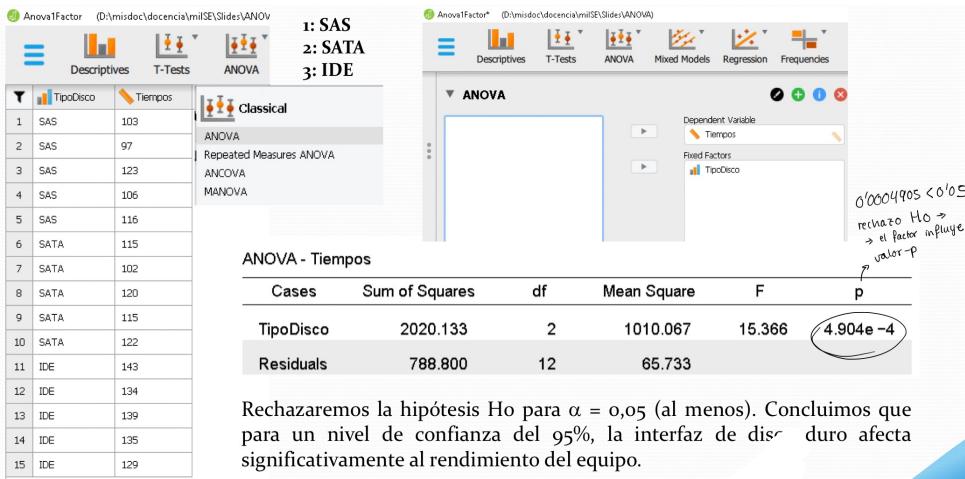
RESUMEN

• Partimos de:

#Experimento	Rendimiento nivel 1	...	Rendimiento nivel n_{niv}
1	y_{11}		$y_{1n_{\text{niv}}}$
2	y_{12}		$y_{2n_{\text{niv}}}$
...
n_{rep}	$y_{n_{\text{rep}}1}$		$y_{n_{\text{rep}}n_{\text{niv}}}$

- **Modelo:** $y_{ij} = m_{\text{global}} + \varepsilon_j + r_{ij} \quad i=1, \dots, n_{\text{rep}}; j=1, \dots, n_{\text{niv}}$
- H_0 : Rendimiento de todos los niveles del factor es equivalente ($\varepsilon_j = 0, j=1, \dots, n_{\text{niv}}$) \Rightarrow El factor no influye en el rendimiento.
- Se calcula $F_{\text{exp}} \equiv \frac{\text{SSA}/(n_{\text{niv}} - 1)}{\text{SSE}/(n_{\text{niv}} \times (n_{\text{rep}} - 1))} \sim F_{n_{\text{niv}} - 1, n_{\text{niv}} \times (n_{\text{rep}} - 1)}$
- p-value \approx Prob (H_0 podría ser cierta).
- Rechazamos H_0 para un nivel de confianza $(1 - \alpha) \times 100\%$ si valor-p < α .
- En ese caso, comparamos las medias de cada nivel unas con otras usando un test t: prueba de múltiples rangos o de comparaciones múltiples.

EJEMPLO



Rechazaremos la hipótesis H_0 para $\alpha = 0.05$ (al menos). Concluimos que para un nivel de confianza del 95%, la interfaz de disco duro afecta significativamente al rendimiento del equipo.

Intervalo de confianza (95%) para \bar{d}_{real}		probabilidad de que tengan el mismo rendimiento					
		p-value de cada test t realizado					
		95% CI for Mean Difference	Lower	Upper	SE	t	Ptukkey
SAS	SATA	-5.800	-19.480 <small>incluye 0</small>	7.880	5.128	-1.131	0.514 <small>acepto</small>
IDE	(SAS - IDE)	-27.000	-40.680 <small>no incluye 0</small>	13.320	5.128	-5.266	5.399e-4 <small>rechazo</small>
SATA	IDE	-21.200	-34.880 <small>no incluye 0</small>	-7.520	5.128	-4.134	0.004 <small>rechazo</small>

Como el factor influye, hacemos ahora un test t entre cada combinación de niveles para comparar el efecto en el rendimiento de cada tipo de disco duro: prueba de múltiples rangos o de comparaciones múltiples.

• La primera fila de la tabla es un test t entre SAS y SATA, la p-value de cada test t realizado segunda entre SAS de IDE y la tercera entre SATA e IDE.

• La última columna es el p-value de cada test t realizado.

Concluimos que, al 95% de confianza, el disco IDE es claramente peor que los otros dos, pero que las diferencias entre SAS y SATA, para este problema, no son estadísticamente significativas, por lo que podríamos decidirnos por el más barato (o hacer más pruebas para estar más seguros).