Tema-4-Analisis-comparativo-del-...

fer__luque

Ingeniería de Servidores

3º Grado en Ingeniería Informática

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

Tema 4: Análisis comparativo del rendimiento

1. Referenciación (*Benchmarking*)

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.

Tiempo de ejecución, frecuencia de reloj y CPI

¿Pueden ser la frecuencia o el número medio de ciclos por instrucción buenos índices de rendimiento?

$$T_{EJEC} = NI \times CPI \times T_{RELOJ} = \frac{NI \times CPI}{f_{RELOJ}}$$

No lo son, hay sistemas con f_{RELOJ} (o CPI) peores que otros pero con mejores prestaciones.

Inconvenientes del T_{EJEC}:

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

Por estas razones, no es un buen índice.

MIP (*million of instructions per second*)

Parece una medida prometedora, ya que representa cómo de rápido ejecuta las instrucciones un microprocesador:

$$MIPS = \frac{NI}{T_{EJEC} \times 10^6} = \frac{f_{RELOJ}}{CPI \times 10^6}$$

Inconvenientes:

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

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

Está basado en operaciones y no en instrucciones:

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

Inconvenientes:

- Representatividad y fiabilidad: No todas las operaciones de coma flotante tienen la misma complejidad. Se pueden usar MFLOPS normalizados (cada operación se multiplica por un peso proporcional a su complejidad)
- Consistencia: El formato puede variar de una arquitectura a otra. Además, ¿y si estas operaciones no son necesarias en mi servidor?

Conclusión final: tampoco nos vale y no hay más candidatos. Nos contentaremos con el tiempo de ejecución (T_{EJEC}) de un determinado programa o conjunto de programas $\rightarrow$ **El índice de rendimiento va a depender de la carga con la que se haga la comparación**

La carga real

Es difícil de utilizar en la evaluación de sistemas por los siguientes motivos:

- Varía con el tiempo
- Resulta complicado reproducirla
- Interacciona con el sistema informático

Es por eso que es más conveniente utilizar un **modelo** de la carga real como carga de prueba (*test workload*) para hacer comprobaciones.

Representatividad del modelo de carga

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

- Debe ser lo más representaativo 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

Hay dos opciones:

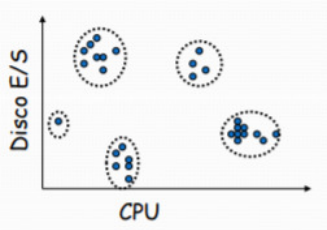
Caracterización de la carga

Se trata de ajustar un modelo paramétrico **personalizado** a partir de la monitorización del sistema ante la carga real.

Consiste en:

- Identificar los recursos que más demande la carga (CPU, memoria, discos...)
- Elegir los parámetros característicos de dichos recursos
- Medir el valor de dichos parámetros usando monitores de actividad (muestreo)
- Analizar los datos: medias, histogramas, agrupamiento o *clustering*

- General el modelo de carga seleccionando representantes de la carga (solicitudes al servidor) junto con información estadística sobre su distribución temporal



Referenciación (*Benchmarking*)

Usar programas de prueba que usen un modelo **genérico** de carga lo más similar posible al que se quiere reproducir. Tienen 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*) eespecífica con la que se 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.

Tipos de programas de *benchmark*: 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

Tipos de programas de *benchmark*: según la generalidad del test

- ***Microbenchmarks*** o *benchmarks* para **componentes**: estresan componentes o agrupaciones de ellos concretos
- ***Macrobenchmarks*** o *benchmarks* para el sistema **completo** o de **aplicación real**: la carga intenta imitar situaciones reales típicas de algún área (comercio, servidores web, de ficheros...)

Ejemplos de *microbenchmarks*

- Whetstone: rendimiento en operaciones en coma flotante con pequeñas aplicaciones científicas
- Linpack: rendimiento en operaciones en coma flotante a traves de un algoritmo para resolver un sistema denso de ecuaciones lineales. Comprueba que la solución es correcta con un grado de exactitud prefijado
- Dhrystone: operaciones con enteros (copia y comparación de cadenas de caracteres)
- Stream: medir ancho de banda de la memoria
- IOzone, HD Tune, Iometer: rendimiento del sistema de ficheros
- Netperf: rendimiento TCP y UDP. Se usa en combinación con netserver y pchar
- Existen también aplicaciones que incorporan varios **paquetes de *microbenchmarks*** para poder realizar diversos tests de forma cómoda:
    - Phoronix Test Suite
    - AIDA54
    - Sandra

SPEC (*Standard Perfomance 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 distintos:

- SPECspeed 2017 Integer
- SPECspeed 2017 Floating poing
- SPECrate 2017 Integer
- SPECrate 2017 Floating point

¿Qué significa cada uno?

- **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++)

Este paquete incluye reglas estrictas para validar los resultados. Se distribuye como una imagen ISO que contiene:

- Código fuente de todos los programas de *benchmark*
- Data sets que necesitan algunos programas
- 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 que se ejecuta y cuántas copias o subprocesos se eligen

Programas dentro de SPECspeed 2017

Criterios generales:

- Han de ser aplicaciones reales
- Portabilidad a muchas arquitecturas

SPECspeed 2017 **Integer**:

- Incluye 10 programas (la mayoría en C y C++)

SPECspeed 2017 **Floating Point**:

- Incluye 10 programas (la mayoría en Fortran y C)

Índices de prestaciones de SPECspeed 2017

También llamados, de forma genérica, **índices SEPC**:

- **Integer**: $CPU_{2017}IntegerSpeed\_peak$, $CPU_{2017}IntegerSpeed\_base$
- **Floating Point**: $CPU_{2017}FP\_Speed\_peak$, $CPU_{2017}FP\_Speed\_base$

Significado de *base* y *peek*:

- **Base**: Se han utilizado los mismos argumentos para compilar todos los programas escritos en el mismo lenguaje (compilación en modo conservador)
- **Peak**: Rendimiento pico, permitiendo que cada uno escoja las opciones de compilación óptimas para cada programa (rendimiento pico)

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**

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

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

Nota: Revisar y entender resultados de SPECspeed 2017 Integer (diapositivas 27, 28 y 29)

Benchmarks de sistema completo: TPC

Organización sin ánimo de lucro especializada en benchmarks relacionados con comercio electrónico y con bases de datos.

Entre sus principales *benchmarks* encontramos (todos son escalables):

- TPC-C: Tipo OLTP (*on-line transaction processing*). Simula una gran compañía que vende productos en varios almacenes, cada uno a cargo de zonas con un número de clientes por zona. Las peticiones involucran acceso a las bases de datos tantos locales como distribuidas.
- 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 ejecutan consultas altamente complejas a una base de datos y analiza enormes volúmenes de datos. *Revisar búsqueda de resultados TPC-H (diapositivas 33-37)*

Métricas:

- Transacciones procesadas por unidad de tiempo superando ciertos requisitos de tiempo de respuesta.
- Coste por transacción procesada y consumo de potencia por transacción procesada

Benchmarks de sistema completo: SPEC y SYSmark25

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.

Pero surge la pregunta de ¿cómo concentrar todos los índices en uno solo? Normalmente se utiliza algún tipo de **media**

La media aritmética

Se define la media aritmética para n medidas, $t_1, \ldots, t_n$:

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

Si no todas las medidas tienen la misma importancia, se puede asociar a cada medida un peso w_k, obteniéndose la **media aritmética ponderada**

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

Para aplicarlo en nuestro caso, podemos pensar: si t_k es el tiempo de ejecución del programa de benchmark k-ésimo en la máquina a testar, w_k podría escogerse, inversamente proporcional a dicho tiempo de ejecución en una determinada máquina de referencia:

$$w_k \propto \frac{C}{t_k^{REF}} \Rightarrow C = \frac{1}{\sum_{k=1}^{n} \frac{1}{t_k^{REF}}}$$

Sin embargo, este índice depende del tiempo de ejecución de cada test en la máquina de referencia escogida

La media geométrica

La media geométrica de n medidas, $S_1, \ldots, S_n$ se define como:

$$\bar{S}_g = \sqrt[n]{\prod_{k=1}^{n} S_k} = \left(\prod_{k=1}^{n} S_k\right)^{\frac{1}{n}}$$

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 (aunque cambian los valores de la media geométrica, el orden no). Usado en los *benchmarks* de SPEC y SYSMARK.

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

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

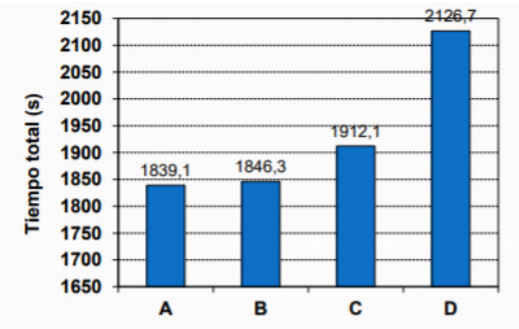
Ejemplo de comparación con tiempos

Para explicar el por qué escogemos la media geométrica, analizaremos este ejemplo con datos inventados:

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

Comparación con el tiempo total

La máquina más rápida es A ya que es la que menos tarda en ejecutar, uno tras otro, todos los programas del *benchmark*. En orden A, B, C, D



Esto no significa que A sea siempre la más rápida

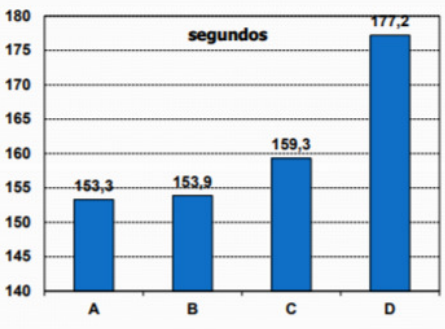
Comparación con la media aritmética

$$\overline{t_A} = \frac{1}{12}\sum_{k=1}^{12} t_k^A = 153{,}3s$$

$$\overline{t_B} = \frac{1}{12}\sum_{k=1}^{12} t_k^B = 153{,}9s$$

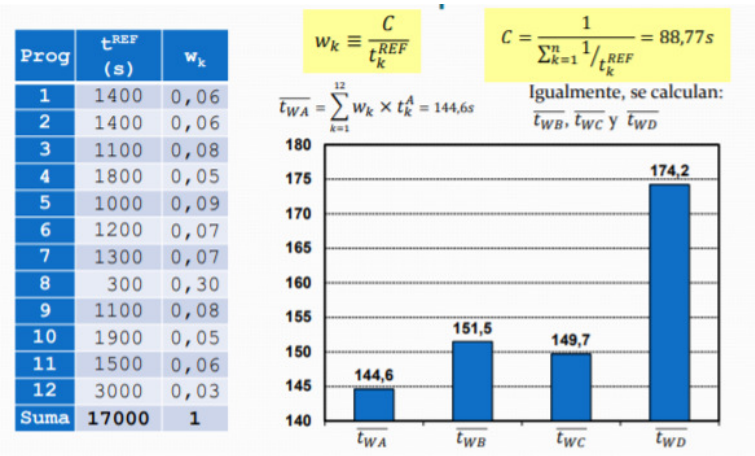
$$\overline{t_C} = \frac{1}{12}\sum_{k=1}^{12} t_k^C = 159{,}3s$$

$$\overline{t_D} = \frac{1}{12}\sum_{k=1}^{12} t_k^D = 177{,}2s$$



El orden se mantiene

Comparación con la media aritmética 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

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 este criterio, se mantiene la máquina A como la más rápida pero B y C intercambian posiciones

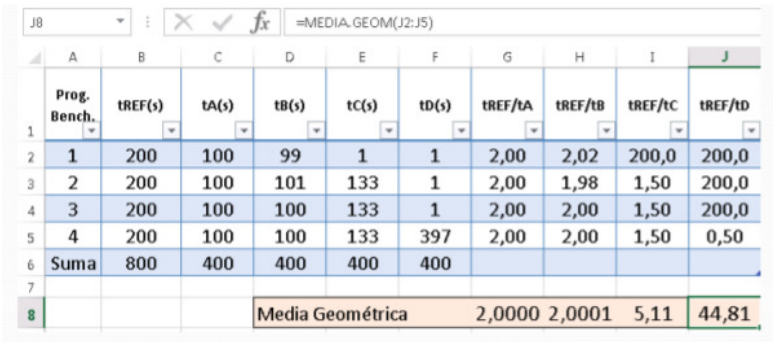
Comparación con la media geométrica de *speedups*

Calculamos la ganancia en velocidad de cada máquina con respecto a la máquina de referencia (tal y como 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

El *speedup* es un índice **a maximizar**, por lo que, según este criterio, la "mejor máquina" es la D.

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



J8 =MEDIA.GEOM(J2:J5)

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

Se puede observar que todos los programas tardan en total lo mismo en ejecutar el total del *benchmark*, pero distribuidos de diferente manera

Como podemos ver, la media geométrica de *speedups* premia las **mejoras sustanciales**, pero **no se castigan empeoramientos no tan sustanciales**. Debemos ser muy cuidadosos con las comparaciones y saber qué estamos haciendo realmente

Conclusiones de este 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. Esa 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

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** $\rightarrow$ 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

Distribución t de Student

Si extraemos n muestras $\{d_1, \ldots, d_n\}$ pertenecientes a una distribución Normal de media $\bar{d}_{real}$ y calculo la siguiente medida:

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

siendo $\bar{d}$ la media muestral y s la desviación típica muestral

$$\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n} \qquad s = \sqrt{\frac{\sum_{i=1}^{n} (d_i - \bar{d})^2}{n-1}}$$

y repetimos el experimento muchas veces, veremos que esos t_{exp} pertenecen a una distribución t-Student con $n-1$ grados de libertad.

Pero **¿para qué me puede servir esto?**

Ejemplo 1: Test de rendimiento entre A y B

Tiempos de ejecución de 6 programas en dos máquinas diferentes 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

$\bar{t_A} = 144{,}5s$
$\bar{t_B} = 120{,}2s$
¿Son significativas estas diferencias?

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

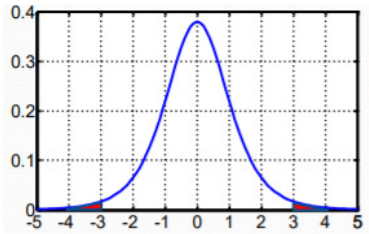
Si partimos de la hipótesis (hipótesis "nula", H_0) de que las máquinas tienen rendimientos equivalentes, entonces las diferencias se debe n a una suma (=una media) de factores aleatorios independientes. En este caso d_i (diferencias entre tiempos de ejecución), serán muestras de una distribución normal de media cero ($\bar{d}_{real} = 0$). Por tanto, por lo explicado previamente:

$$t_{exp} = \frac{\bar{d}}{s/\sqrt{n}} = \frac{24{,}3s}{8{,}12s} = 2,99$$

pertenecerá a una distribución t-Student con $6 - 1 = 5$ grados de libertad. **¿Qué probabilidad hay de que esto sea realmente así?**

Nivel o Grado de Significatividad

Distribución t de Student con 5 grados de libertad:



$P - value = P(|t| \geq |t_{exp}|)$ en T_{n-1}
$= 2 \times P(t \leq -|t_{exp}|)$ en T_{n-1}

=DISTR.T.2C(2,99;5) = 0,03 *(Excel).*
=DISTR.T(2,99;5;2) = 0,03 *(Calc).*
=2·tcdf(-2.99,5) = 0,03 *(Matlab).*

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

Conclusión: Si $P - value < \alpha$ diremos que, para un grado de significatividad α o para **un nivel de confianza** $(1 - \alpha) * 100$ (normalmente 95%), las máquinas tienen **rendimientos estadísticamente diferentes**.

En este 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.

Intervalos de confianza para t_{exp}

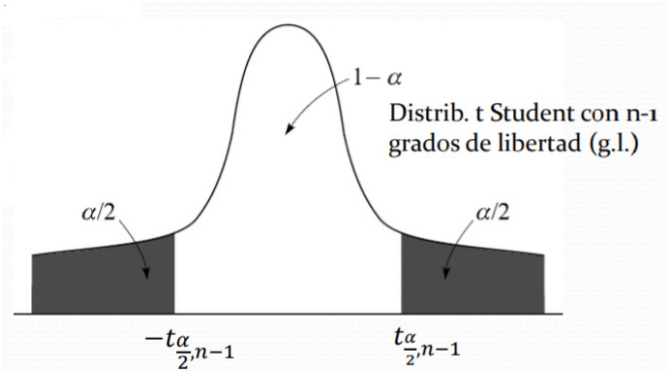
Para un nivel de significatividad α (típ. $0,05 = 5\%$), buscamos el valor $t_{\alpha/2,n-1}$ que cumpla $Prob(|t| > t_{alpha/2,n-1})$ o equivalentemente:

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

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.



En nuestro ejemplo, para un nivel de significatividad de $\alpha = 0,05$, buscamos $t_{\alpha/2,n-1}$ tal que

$$Prob(t \leq -t_{\alpha/2,n-1}) = \alpha/2 = 0,025$$

para una distribución t de Student con 5 grados de libertad.

Lo obtenemos con excel por ejemplo y obtenemos que $t_{\alpha/2,n-1} = 2,57$

Como $t_exp = 2,99$ no está en ese rango pues **rechazamos la hipótesis** H_0

Intervalos de confianza para $\bar{d}_{real}$

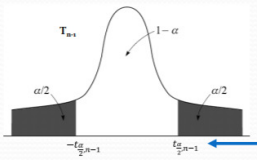
Básicamente, se basa en "traducir" el intervalo de t_{exp} en el que se cumple la hipótesis, al intervalo en que **debe estar el** $\bar{d}_{real}$.

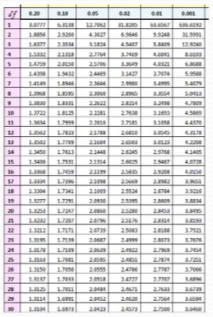
En caso de que el 0 no se encuentre en este intervalo, podemos **rechazar la hipótesis**.

Resumen test t para muestras pareadas

- Partimos de:

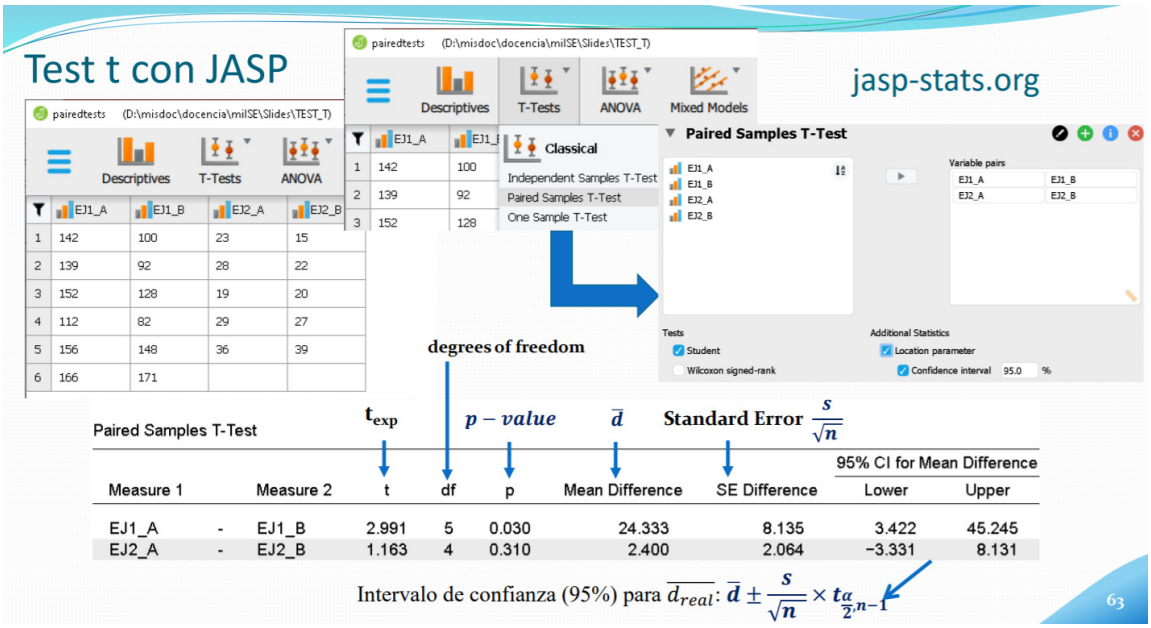
Exp.	tA	tB	$d_i = tA_i - tB_i$
P_1	tA_1	tB_1	d_1
P_2	tA_2	tB_2	d_2
...	...	...	...
P_n	tA_n	tB_n	d_n





- Ho: Rendimiento A ≡ Rendimiento B, es decir, $d_i \sim \mathcal{N}(\bar{d}_{real}, \sigma^2)$ con $\bar{d}_{real} = 0$
- Calculo $t_{exp} = \frac{\bar{d} - \bar{d}_{real}}{s/\sqrt{n}} \sim T_{n-1}$ siendo $\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n}$ $s = \sqrt{\frac{\sum_{i=1}^{n}(d_i - \bar{d})^2}{n-1}}$
- Definimos el nivel o grado de significatividad α.
- Rechazamos Ho para un nivel de confianza (1- α)*100(%) si:
  1. Método 1: p-value < α. Siendo p-value = $P(|t| \geq |t_{exp}|)$ en T_{n-1} ≈ Prob (Ho podría ser cierta).
  2. Método 2: $t_{exp} \notin \left[-t_{\frac{\alpha}{2},n-1}, t_{\frac{\alpha}{2},n-1}\right]$. Siendo $t_{\frac{\alpha}{2},n-1}$ el valor que hace que $Prob(t \leq -t_{\frac{\alpha}{2},n-1}) = \frac{\alpha}{2}$ para una distribución t de Student con n-1 grados de libertad.
  3. Método 3: $0 \notin \left[\bar{d} - \frac{s}{\sqrt{n}} \times t_{\frac{\alpha}{2},n-1}, \bar{d} + \frac{s}{\sqrt{n}} \times t_{\frac{\alpha}{2},n-1}\right]$. Intervalo de confianza para $\bar{d}_{real}$.

Test t con JASP



Otra utilidad del test t: Estimación de intervalos de confianza de medias de medidas experimentales

Hipótesis: Realizamos n medidas $\{d_1, \ldots, d_n\}$ de un mismo fenómeno. Si estas 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.

Por tanto, hay un $(1-\alpha) * 100$ de probabilidad de que el valor medio real $\bar{d}_{real}$ se encuentre en el intervalo visto anteriormente.

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

4. Diseño de experimentos de comparación de rendimiento

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 repuesta o dependiente (métrica)**: El índice de rendimiento que usamos para las comparaciones. Ej: tiempos de respuesta (R), productividades (X).
- **Factor**: Cada una de las variables que pueden afectar a la variable respuesta. Ej: SO, tamaño memoria, tipo procesador...
- **Nivel**: Cada uno de los valores que puede asumir un factor. Ej: Para el SO (Windows, CentOS...), para tipo de disco (SATA, IDE...)
- **Interacción**: Una interacción ocurre cuando el efecto de un factor cambia para diferentes niveles de otro factor. Ej: El hecho de utilizar un tipo de SO puede afectar a cómo de importante sea la cantidad de memoria

Tipos de diseños experimentales

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

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

Se trata por tanto de lo ya visto para un número de factores mayor

Ejemplo

#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	$m_{global} = 119.9s$
Efectos (ε_j)	-10.9	-5.1	16.1	

Análisis de la Varianza (ANOVA) de un factor

Este test ANOVA parte de una hipótesis similar: **el factor a considerar no influye en el rendimiento**.

Usa el siguiente **modelo**: $y_{ij} = m_{global} + \varepsilon_j + r_{ij}$, donde:

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_{niv}=3 niveles: SAS, SATA e IDE). El índice i recorre las distintas repeticiones para cada uno de esos niveles (en nuestro caso, n_{rep}=5 repeticiones).

m_{global}: Media global de todas las observaciones:

$$m_{global} = \frac{1}{n_{rep} \times n_{niv}} \sum_{i=1}^{n_{rep}} \sum_{j=1}^{n_{niv}} y_{ij}$$

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

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

La principal pregunta a contestar es: ¿Tiene influencia el factor sobre la variable respuesta o, dicho de otro modo, algún ε_j es distinto de cero?

Tras realizar algunos cálculos estadísticos se obtiene el llamado como **estadístico** (F_{exp}), que, en caso de que pertenezca a una distribución **F de Snedecor** con $n_{niv} - 1$ grados de libertad en el numerador y $n_{niv} \times (n_{rep} - 1)$ en el denominador.

Igual que con el test T, obtendremos el P-value que nos indica la probabilidad de que un valor de F sea mayor o menor que $|F_{exp}|$. Este valor P-value debe ser menor que el nivel de significatividad para aceptar la hipótesis.

En caso de que sea mayor (sí influye el factor), no podemos quedarnos con el que parezca mejor, ya que la hipótesis nos dice solo que **los 3 nos son equivalentes**, pero puede darse el caso de que dos sean equivalentes y otro no.

Por tanto, ahora debemos comparar las medias de cada nivel unas con otras usando un *test t*: **prueba de múltiples rangos** o de comparaciones múltiples

***post-hoc test*: Prueba de múltiples rangos**

Se trata de un tipo especial de test t (el error estándar es el mismo para todos los pares), que nos da una tabla como la siguiente:

Intervalo de confianza (95%) para $\overline{d_{real}}$ ↓ — **p-value de cada test t realizado** ↓

		95% CI for Mean Difference				
	Mean Difference	Lower	Upper	SE	t	Ptukey
SAS → SATA	−5.800	−19.480	7.880	5.128	−1.131	0.514
SAS ↘ IDE	−27.000	−40.680	−13.320	5.128	−5.266	5.399e −4
SATA → IDE	−21.200	−34.880	−7.520	5.128	−4.134	0.004

Vemos que H_0 es cierta para:

- SAS $\equiv$ SATA: $0,514 \geq 0,05$
- SAS $\not\equiv$ IDE: $5,4 \cdot 10^{-4} < 0,05$: Como $Mean\ Difference(SAS - IDE) < 0$, significa que $\bar{t}_{SAS} - \bar{t}_{IDE} < 0$, por lo que IDE tarda más que SAS. Por tanto SAS mejor que IDE al 95% de confianza
- SATA $\not\equiv$ IDE: $0,004 < 0,05$. $\bar{t}_{SATA} - \bar{t}_{IDE} < 0$, por lo que IDE tarda también más que SATA. Por tanto SATA mejor que IDE al 95% de confianza

Como SAS y SATA mejor que IDE y equivalentes entre ellos, escogeríamos el más barato.

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.