

## Tema 4. Análisis comparativo del rendimiento

¿Qué servidor tiene mejor rendimiento?

Analistas, administradores y diseñadores

## Objetivos del tema

- Entender la problemática inherente al diseño de un índice de rendimiento cualquiera.
- Interpretar los índices clásicos de rendimiento usados en el ámbito de los procesadores.
- Entender el concepto de benchmark y sus distintos tipos.
- Conocer ejemplos reales de benchmarks.
- Conocer diferentes estrategias de análisis para hacer comparaciones de rendimiento así como las condiciones para hacer una comparación de rendimiento lo más ecuánime posible.

2

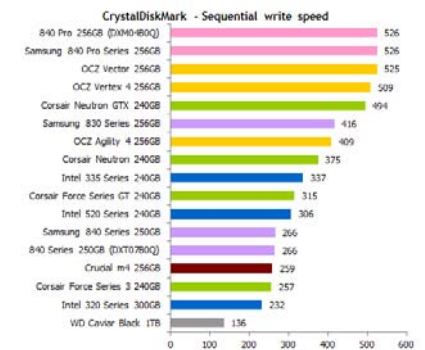
## Bibliografía

- *Evaluación y modelado del rendimiento de los sistemas informáticos*. Xavier Molero, C. Juiz, M. Rodeño. Pearson Educación, 2004. Capítulo 3.
- *Measuring computer performance: a practitioner's guide*. David J. Lilja, Cambridge University Press, 2000. Capítulos 2,5 y 7.
- *The art of computer systems performance analysis : Techniques for experimental design, measurement, simulation, and modeling*. Raj Jain, John Wiley & Sons, 1991. Capítulos 9, 13 y 20.
- *System Performance Tuning*. Gian-Paolo D. Musumeci, Mike Loukides, 2nd Edition - O'Reilly Media, 2002. Capítulo 2.
- *The Standard Performance Evaluation Corporation (SPEC)*, <http://www.spec.org>.
- *The Transaction Processing Performance Council (TPC)*, <http://www.tpc.org>.

3

## Contenido

- Introducción: Índices clásicos de rendimiento.
- Benchmarking.
- Análisis estadístico de los resultados de un benchmark.
- Diseño de experimentos.



4

## 4.1. Introducción: índices clásicos de rendimiento

## 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.
- **Linealidad:** Si el índice de rendimiento aumenta, el rendimiento real del sistema debe aumentar en la misma proporción.
- **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 debe ser fácil de medir y la forma de medirlo debe ser la misma para cualquier sistema.

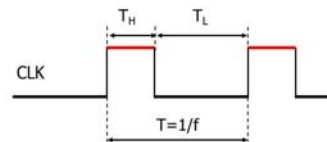


## Tiempo de ejecución, frecuencia de reloj y ciclos por instrucción

El tiempo de ejecución es el mejor índice de rendimiento a priori pero depende del programa o programas que se ejecuten. ¿Existen otros índices posibles para expresar el rendimiento del sistema? Históricamente se han usado  $f_{RELOJ}$ , CPI, MIPS y MFLOPS:

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

- NI = Número de instrucciones del programa/programas a ejecutar.
- $f_{RELOJ}$  = frecuencia de reloj.
- CPI = Nº medio de ciclos por instrucción.



### Desventajas:

- Ni la frecuencia de reloj ni los ciclos por instrucción son representativos del rendimiento de un sistema. Es posible encontrar ejemplos de sistemas con  $f_{RELOJ}$  (o CPI) peores que otros pero con mejores prestaciones.

## MIPS

- MIPS (*million of instructions per second*)
  - Se denominan MIPS nativos:

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

- Depende del juego de instrucciones y los MIPS medidos varían entre programas en el mismo computador.
- MIPS = *Meaningless Indicator of Processor Speed*.
- MIPS relativos: referidos a una máquina de referencia (proceso de normalización)

$$MIPS_{relativos} = \frac{T_{EJEC \text{ MÁQUINA REF}}}{T_{EJEC}} \times MIPS_{MÁQUINA REF}$$

# MFLOPS

- MFLOPS (*million of floating-point **operations** per second*)
  - Basado en operaciones y no en instrucciones.
  - El tiempo de ejecución de la fórmula es el del programa completo, incluyendo el tiempo consumido por las instrucciones de enteros.

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

- Problema: No todas las operaciones de coma flotante tienen la misma complejidad → MFLOPS normalizados. Ejemplo de normalización de operaciones en coma flotante:
  - ADD, SUB, COMPARE, MULT ⇒ 1 operación normalizada
  - DIVIDE, SQRT ⇒ 4 operaciones normalizadas
  - EXP, SIN, ATAN, ... ⇒ 8 operaciones normalizadas

9

# Cálculo de los MFLOPS de un programa

- Programa Spice: el computador DECStation 3100 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.

$$MFLOPS_{nativos} = \frac{109970178}{94 \times 10^6} = 1,2$$

$$MFLOPS_{normalizados} = \frac{94287845 \times 1 + 15682333 \times 4}{94 \times 10^6} = 1,7$$

- Problema: El formato de los números en coma flotante puede variar de una arquitectura a otra y, por tanto, tener diferente precisión.

10

## 4.2. Benchmarking

# La carga real

- Difícil de utilizar en la evaluación de sistemas.
  - Varía a lo largo del tiempo.
  - Resulta complicado reproducirla.
  - Interacciona con el sistema informático.



- Es más conveniente utilizar un **modelo** de la carga real como carga de prueba (test workload) para hacer comparaciones.

12

## Representatividad del modelo de carga

- Los modelos de carga son aproximaciones que representan una abstracción 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).



13

## Principales estrategias para obtener modelos de carga

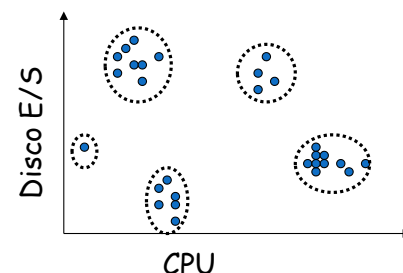
- Ajustar un modelo paramétrico “**personalizado**” a partir de la monitorización del sistema ante la carga real (*caracterización de la carga*).
- Usar programas de prueba que usen un modelo **genérico** de carga lo más similar posible al que se quiere reproducir (*referenciación o benchmarking*).



14

## Caracterización de la carga

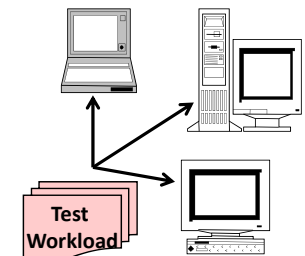
- Usualmente la caracterización de la carga de un sistema se realiza siguiendo los siguientes pasos:
  - Identificación de los recursos que más demande la carga (CPU, memoria, discos, red, etc.)
  - Elección de los parámetros característicos de dichos recursos (utilización de CPU, lecturas/escrituras que hay que hacer en cada disco, lecturas/escrituras a memoria, número de accesos a la red, etc.)
  - Recolección de datos (usando monitores de actividad).
  - Análisis y clasificación de los datos (medias, histogramas, agrupamiento o *clustering*, etc.).
  - Extracción de los representantes de la carga junto con información estadística sobre su distribución temporal.



15

## Referenciación (Benchmarking)

- Técnica usadas en la comparación del rendimiento de diferentes sistemas informáticos.
- Todos los sistemas se han de someter a la misma carga, por lo que ésta ha de ser suficientemente genérica.
- Un benchmark (*benchmark program*) es un programa o un conjunto de programas diseñados con el fin de comparar alguna característica del rendimiento entre equipos informáticos. Hay dos características principales que definen a un programa de 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 y validación de los resultados.



16

## 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 precisión de la solución de un determinado problema que se puede alcanzar en 1 minuto de ejecución.
- Programas que permiten variar tanto la cantidad de tareas como el tiempo de cómputo para adaptarlos a cada sistema.
  - HINT: Calcula los límites inferior y superior de una integral hasta que el sistema se quede sin recursos. Medida de rendimiento: QUIPS (*quality improvements per second*).

17

## Tipos de programas de benchmark: 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.
- Macrobenchmarks o benchmarks de sistema **completo** o de **aplicación real**: carga compuesta por un conjunto de aplicaciones, normalmente comerciales, habitualmente utilizadas en 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.



18

## Ejemplos de microbenchmarks

- 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 precisión prefijado.
- Dhrystone (1984)
  - Mide el rendimiento de operaciones con enteros, esencialmente por medio de operaciones de copia y comparación de cadenas de caracteres.

19

## Ejemplos de microbenchmarks (II)

- Stream: para medir el ancho de banda de la memoria <http://www.streambench.org/>.
- IOzone: rendimiento del sistema de ficheros (lecturas y escrituras a/desde el disco duro), <http://www.iozone.org/>. Igualmente HD Tune (Windows, <http://www.hdtune.com/>), Iometer (<http://www.iometer.org/>), fio (flexible I/O tester, Linux) o el comando 'hdparm -tT' (Linux).
- Netperf: rendimiento TCP y UDP (Linux y Windows). Basado en una arquitectura cliente-servidor, se usa en combinación con otro programa (netserver) que debe estar instalado en el servidor. <http://www.netperf.org/netperf/>. También pchar (=traceroute que calcula el ancho de banda por cada salto).
- También existen aplicaciones que incorporan varios **paquetes de microbenchmarks** para poder realizar diversos tests de forma cómoda:
  - LMBench (Unix, <http://lmbench.sourceforge.net>).
  - AIDA64 (Windows, <http://www.aida64.com>).
  - Passmark (Windows, <http://www.passmark.com>).
  - Sandra (Windows, <http://www.sissoftware.net>).

20

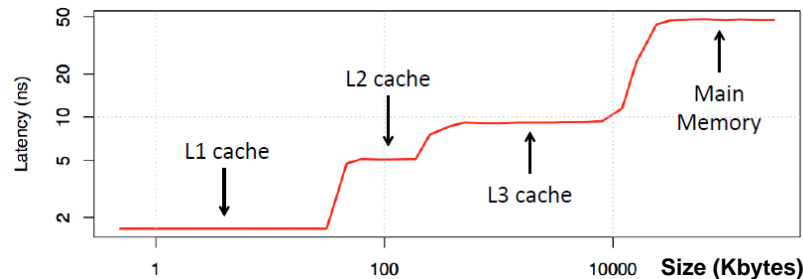


# Ejemplos de microbenchmarks (III)

fio

```
$ fio --name=seqwrite --rw=write --bs=128k --size=122374m
[...]
```

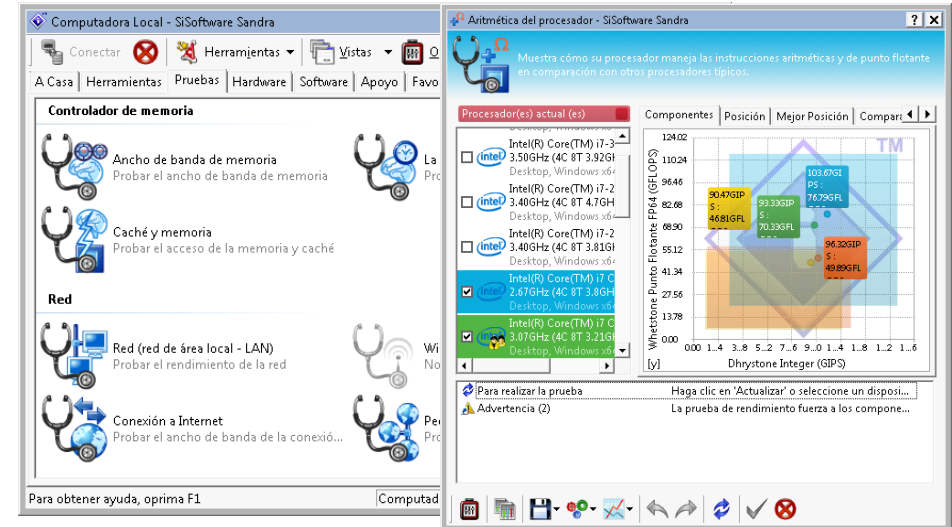
seqwrite: (groupid=0, jobs=1): err= 0: pid=22321  
 write: io=122374MB, bw=840951KB/s, iops=6569, runt=149011msec  
 clat (usec): min=41, max=133186, avg=148.26, stdev=1287.17  
 lat (usec): min=44, max=133188, avg=151.11, stdev=1287.21  
 bw (KB/s): min=10746, max=1983488, per=100.18%, avg=842503.94,  
 stdev=262774.35  
 cpu : usr=2.67%, sys=43.46%, ctx=14284, majf=1, minf=24  
 IO depths : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%  
 submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%  
 complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%  
 issued r/w/d: total=0/978992/0, short=0/0/0  
 lat (usec): 50=0.02%, 100=98.30%, 250=1.06%, 500=0.01%, 750=0.01%  
 lat (usec): 1000=0.01%  
 lat (msec): 2=0.01%, 4=0.01%, 10=0.25%, 20=0.29%, 50=0.06%  
 lat (msec): 100=0.01%, 250=0.01%



Imbench

21

# Paquetes de microbenchmarks: Sandra



22

# SPEC (<http://www.spec.org>)

The screenshot shows the homepage of the Standard Performance Evaluation Corporation (SPEC). It includes a navigation menu with links to Home, Benchmarks, Tools, Results, Contact, Site Map, Search, and Help. The main content area features a 'Benchmarks' section with a list of categories (Cloud, CPU, Graphics/Workstations, etc.) and a 'What's New' section with recent updates and news items.

23

# El paquete de microbenchmarks SPEC CPU 2017

- Compuesto por cuatro conjuntos de benchmarks con los que obtener 4 índices distintos (<http://www.spec.org/cpu2017/>):
  - SPECspeed®2017 Integer (rendimiento en aritmética entera)
  - SPECspeed®2017 Floating Point (rendimiento en coma flotante)
  - SPECrate®2017 Integer
  - SPECrate®2017 Floating Point
    - 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
  - Sistema de memoria
  - Compilador (C, Fortran y C++)



- Reglas estrictas para validar los resultados:
  - <https://www.spec.org/cpu2017/Docs/runrules.html>

24

## El paquete de microbenchmarks SPEC CPU 2017

- SPEC CPU2017 se distribuye como una imagen ISO que contiene:
  - Código fuente para los benchmarks de cada suite.
  - Data sets que necesitan algunos benchmarks para su ejecución.
  - Herramientas varias para compilación, ejecución, 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.

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

25

## Programas dentro de SPEC CPU 2017

- Criterios generales:
  - Han de ser aplicaciones reales.
  - 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
  - ...
- Ejemplo: SPECspeed®2017 Floating Point: 10 programas (la mayoría en Fortran y C)
  - 619.lbm\_s Dinámica de fluidos
  - 621.wrf\_s Predicción meteorológica
  - 638.imagick\_s Procesamiento de imágenes
  - ...

26

## Índices de prestaciones en SPEC CPU2017

- Índices de prestaciones (índices SPEC)
  - Aritmética entera: CPU2017IntegerSpeed\_peak, CPU2017IntegerSpeed\_base, CPU2017IntegerRate\_peak, CPU2017IntegerRate\_base.
  - Aritmética en coma flotante: CPU2017FP\_Speed\_peak, CPU2017FP\_Speed\_base, CPU2017FP\_Rate\_peak, CPU2017FP\_Rate\_base.
- Significado de “base” y “peak”:
  - Base: Compilación en modo conservador, es decir, con reglas estrictas para que todos usen 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 final es la media geométrica de esos tiempos de ejecución normalizados respecto a una máquina de referencia (Sun Fire V490 con procesador UltraSPARC IV+).

27

## Resultados de SPEC CPU2017IntegerSpeed



### 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: 2017-10-19T11:49

### CPU2017 Integer Speed (7):

[\[Search in CPU2017 Integer Speed results\]](#)

Test Sponsor	System Name	Parallel	Base Threads	Processor			Results	
				Enabled Cores	Enabled Chips	Threads/Core	Base	Peak
HPPE	Integrity Superdome X (384 core, 2.20 GHz, Intel Xeon E7-8890 v4) <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	No	384	384	16	2	5.31	5.86
HPPE	ProLiant DL580 Gen9 (2.20 GHz, Intel Xeon E7-8890 v4) <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	No	96	96	4	1	5.35	5.95
HPPE	ProLiant ML350 Gen9 (2.20 GHz, Intel Xeon E5-2699 v4) <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	No	44	44	2	1	5.80	6.43
HPPE	ProLiant DL380 Gen10 (2.10 GHz, Intel Xeon Platinum 8170) <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	Yes	52	52	2	1	8.96	Not Run
HPPE	ProLiant DL380 Gen10 (2.10 GHz, Intel Xeon Platinum 8176) <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	Yes	56	56	2	1	9.16	Not Run
Huawei	Huawei 2288H V5 (Intel Xeon Platinum 8180) <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	Yes	56	56	2	1	9.46	9.79
Oracle Corporation	Sun Fire V490 <a href="#">HTML</a>   <a href="#">CSV</a>   <a href="#">Text</a>   <a href="#">PDF</a>   <a href="#">PS</a>   <a href="#">Config</a>	Yes	1	8	4	1	1.00	Not Run

$$CPUint_{base} = \sqrt[10]{\frac{t_{1}^{REF}}{t_{1}^{base}} \times \frac{t_{2}^{REF}}{t_{2}^{base}} \times \dots \times \frac{t_{10}^{REF}}{t_{10}^{base}}}$$

$$CPUint_{peak} = \sqrt[10]{\frac{t_{1}^{REF}}{t_{1}^{peak}} \times \frac{t_{2}^{REF}}{t_{2}^{peak}} \times \dots \times \frac{t_{10}^{REF}}{t_{10}^{peak}}}$$

28

## Resultados de SPEC CPU2017 IntegerSpeed (II)

Hardware				Software										
CPU Name:	Intel Xeon E7-8890 v4			OS:	SUSE Linux Enterprise Server 12 (x86_64) SP1									
Max MHz:	3400				3.12.53-60.30-default									
Nominal:	2200			Compiler:	C/C++: Version 17.0.0.098 of Intel C/C++ Compiler for Linux; Fortran: Version 17.0.0.098 of Intel Fortran Compiler for Linux									
Enabled:	384 cores, 16 chips, 2 threads/core													
Orderable:	2 to 16 chips			Parallel:	No									
Cache L1:	32 KB I + 32 KB D on chip per core			Firmware:	HP Bundle: 008.004.084 SFW: 043.025.000 08/16/2016									
L2:	256 KB I+D on chip per core			File System:	xfs									
L3:	60 MB I+D on chip per chip			System State:	Run level 5 (multi-user, w/GUI)									
Other:	None			$CPU2017IntSpeed_{base} = \sqrt[10]{4,96 \times 7,29 \times 5,45 \times \dots}$ $CPU2017IntSpeed_{peak} = \sqrt[10]{6,01 \times 7,45 \times 6,75 \times \dots}$										
Memory:	4 TB (128 x 32 GB 2Rx4 PC4-2400T-L, running at 1600 MHz)													
Storage:	8 x C8S59A, 900 GB 10 K RPM SAS													
Other:	None													
Results Table														
Benchmark	Base						Peak							
	Threads	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Threads	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
600.perlbench_s	384	365	4.86	358	4.96	357	4.98	384	298	5.95	295	6.02	295	6.0
602.gcc_s	384	553	7.20	546	7.29	546	7.29	384	540	7.37	535	7.45	534	7.4
605.mcf_s	384	866	5.45	866	5.45	898	5.26	384	708	6.67	700	6.75	699	6.7
620.omnetpp_s	384	276	5.90	271	6.03	289	5.65	384	251	6.50	247	6.61	246	6.6
623.xalanbmk_s	384	189	7.50	188	7.52	187	7.57	384	179	7.91	179	7.93	180	7.8
625.x264_s	384	283	6.24	282	6.25	283	6.23	384	271	6.51	272	6.49	270	6.5
631.deepsjeng_s	384	407	3.52	408	3.52	407	3.52	384	343	4.18	343	4.18	343	4.1
641.1eala_s	384	460	3.64	460	3.64	460	3.63	384	438	3.90	430	3.88	440	3.8

29

## Benchmarks de sistema completo: TPC

- TPC (Transactions Processing Performance Council, <http://www.tpc.org>): Organización sin ánimo de lucro especializada en benchmarks relacionados con comercio electrónico y con bases de datos.

The screenshot shows the TPC website interface. At the top, it says 'developing data-centric benchmark standards and disseminating objective, verifiable performance data to the industry... The TPC is a'. Below this, there are links for 'Home', 'About the TPC', and 'Benchmarks'. The 'Benchmarks' section is highlighted, showing a list of benchmarks: TPC-C, TPC-DS, TPC-H, TPC-V, TPC-DB, TPC-ED, TPC-EE, TPC-ER, TPC-ES, TPC-ET, TPC-EX, TPC-F, TPC-G, TPC-I, TPC-J, TPC-K, TPC-L, TPC-M, TPC-N, TPC-O, TPC-P, TPC-Q, TPC-R, TPC-S, TPC-T, TPC-U, TPC-V, TPC-W, TPC-X, TPC-Y, TPC-Z. The 'TPC Benchmarks & Benchmark Results' section is also visible, with a list of benchmarks and their results.

30

## Benchmarks de sistema completo: TPC

- Principales benchmarks:
  - TPC-C: Tipo OLTP (*on-line transaction processing*). Simula una gran compañía con varios almacenes, cada uno con 100.000 productos y tiene 3000 clientes. Peticiones que involucran acceso a las bases de datos tanto locales como distribuidas.
  - TPC-E: Tipo OLTP. Simula una correduría de bolsa en donde hay una única base de datos central. El benchmark es escalable de modo que se pueden simular transacciones de compañías de diversos tamaños.
  - 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.
- Métricas: peticiones/transacciones procesadas por unidad de tiempo (*tps/tpm/tpH*) superando unos ciertos requisitos de tiempos de respuesta. También: coste por petición procesada (incluido mantenimiento) y consumo de potencia por petición procesada.

31

## TPC-C: Los mejores resultados

TPC Transaction Processing Performance Council

TPC-C - Top Ten Performance Results  
Version 5 Results As of 18-Mar-2014 12:40 PM [GMT]

Note 1: The TPC believes it is not valid to compare prices or price/performance of results in different currencies.

All Results Clustered Results Non-Clustered Results Currency All

Rank	Company	System	Performance (tpmC)	Price/tpmC	Watts/KtpmC	System Availability	Database	Operating System	TP Monitor	Date Submitted	Cluster
1	ORACLE	SPARC T5-8 Server	8,552,523	.55 USD	NR	09/25/13	Oracle 11g Release 2 Enterprise Edition with Oracle Partitioning	Oracle Solaris 11.1	Oracle Tuxedo CFSR	03/26/13	N
2	ORACLE	Sun Server X2-8	5,055,888	.89 USD	NR	07/10/12	Oracle Database 11g R2 Enterprise Edition w/Partitioning	Oracle Linux w/Unbreakable Enterprise Kernel R2	Tuxedo CFSR	03/27/12	N
3	IBM	IBM System x3850 X5	3,014,684	.59 USD	NR	09/22/11	IBM DB2 ESE 9.7	SUSE Linux Enterprise Server 11 SP1 for x86_64	Microsoft COM+	07/11/11	N
4	CISCO	Cisco UCS C240 M3 Rack Server	1,609,186	.47 USD	NR	09/27/12	Oracle Database 11g Standard Edition One	Oracle Linux w/Unbreakable Enterprise Kernel R2	Microsoft COM+	09/27/12	N
5	IBM	IBM Flex System x240	1,503,544	.53 USD	NR	08/16/12	IBM DB2 ESE 9.7	Red Hat Enterprise Linux 6.2	Microsoft COM+	04/11/12	N
6	IBM	IBM System x3650 M4	1,320,082	.51 USD	NR	02/25/13	IBM DB2 ESE 9.7	Red Hat Enterprise Linux 6.4 with xVM	Microsoft COM+	02/22/13	N
	HP	HP ProLiant Blade					Microsoft SQL Server	Microsoft Windows	Microsoft		

32



## Benchmarks de sistema completo: 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).
- **Virtualization: SPECvirt\_sc2010** (Virtualización en Centros de Procesamiento de Datos).
- **Cloud: SPEC Cloud\_IaaS 2016** (Servicios en la nube)
- **Consumo de potencia: SPECpower\_ssj2008** (Rendimiento de un servidor ejecutando aplicaciones JAVA frente al consumo de potencia).

33

## Benchmarks de sistema completo: SYSMark 2012

- Para comparar PC con S.O. Windows.
- Considera la carga en 6 escenarios:
  - Office Productivity: Word, PowerPoint, Outlook, Acrobat ...
  - Media Creation: Adobe Photoshop, Adobe Premiere...
  - Web Development: Dreamweaver, IE, Firefox...
  - Data/Financial Analysis: Excel.
  - 3D Modeling: Autodesk 3ds Max, AutoCAD, Google SketchUp...
  - System Management: Winzip, Firefox installer.
- Con cada programa se ejecuta un conjunto de tareas de acuerdo con un modelo de comportamiento de un usuario “habilitado”.
- El tiempo medio de ejecución de los benchmarks de cada categoría se normaliza (ratio) respecto de una máquina de referencia. Finalmente, el índice SYSMark2012 se calcula mediante la media geométrica de los ratios obtenidos.

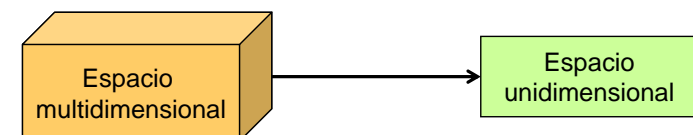


34

## 4.3. Análisis de los resultados de un benchmark

### ¿Cómo expresar el rendimiento final tras la ejecución de un benchmark?

- El rendimiento es una variable multidimensional.
  - Habría de 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 algún tipo de **media**.



36

## La media aritmética

- Dado un conjunto de  $n$  medidas,  $t_1, \dots, t_n$ , 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  $t_k$  un peso  $w_k$ , obteniéndose la **media aritmética ponderada**:

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

Si  $t_k$  es el tiempo de ejecución del programa de benchmark  $k$ -ésimo,  $w_k$  podría escogerse, por ejemplo, inversamente proporcional a  $t_{REF\_k}$ , el tiempo de ejecución en la máquina de referencia:

$$w_k \equiv \frac{C}{t_{REF\_k}} \quad C = \frac{1}{\sum_{k=1}^n 1/t_{REF\_k}}$$

37

## La media geométrica

- Dado un conjunto de  $n$  medidas,  $r_1, \dots, r_n$ , definimos su media geométrica:

$$\bar{r}_g = \sqrt[n]{\prod_{k=1}^n r_k} = \left( \prod_{k=1}^n r_k \right)^{1/n}$$

- Propiedad: cuando los valores son medidas de ganancias en velocidad con respecto a un sistema de referencia, este índice mantiene el mismo orden en las comparaciones independientemente del sistema de referencia usado. Usado en los benchmarks de SPEC y SYSMARK.

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

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

38

## Ejemplo de comparación con tiempos

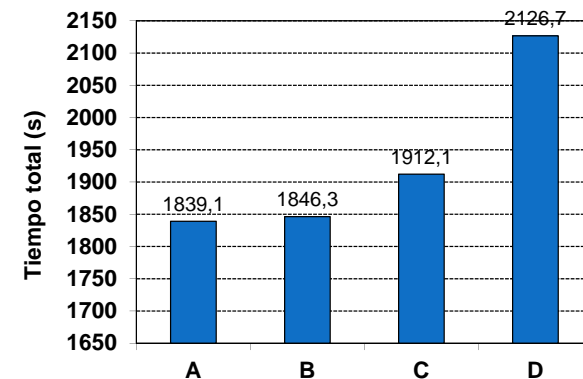
Programa	Ref. (s)	A (s)	B (s)	C (s)	D (s)
1	1400	141	170	136	134
2	1400	154	166	215	25
3	1100	96,8	94,2	146	201
4	1800	271	283	428	523
5	1000	83,8	90,1	77,4	81,2
6	1200	179	189	199	245
7	1300	120	131	87,7	75,5
8	300	151	158	138	192
9	1100	93,5	122	88	118
10	1900	133	173	118	142
11	1500	173	170	179	240
12	3000	243	100	100	150
Suma	17000	1839,1	1846,3	1912,1	2126,7

- La máquina más rápida es "A" ya que es la que tarda menos en ejecutar todos los programas del benchmark (1839,1 segundos).

39

## 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.



40

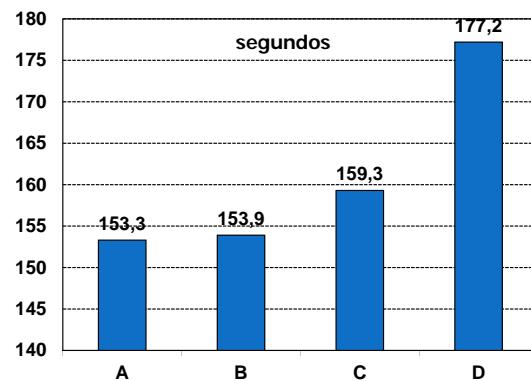
## Comparación con la media aritmética

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

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

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

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



- La máquina más rápida (la que ejecuta los programas del benchmark en menor tiempo) es la de menor media aritmética de los tiempos de ejecución.

41

## Usando la media aritmética ponderada

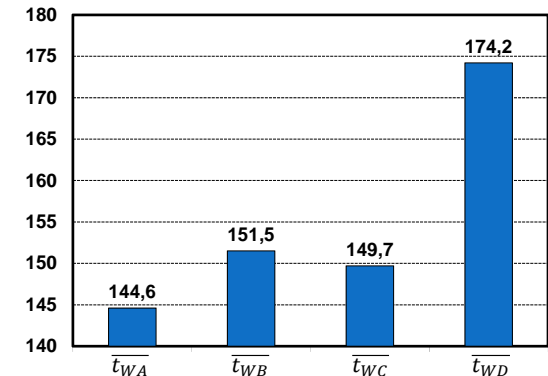
Prog	Ref. (s)	$w_k$
1	1400	0,06
2	1400	0,06
3	1100	0,08
4	1800	0,05
5	1000	0,09
6	1200	0,07
7	1300	0,07
8	300	0,30
9	1100	0,08
10	1900	0,05
11	1500	0,06
12	3000	0,03
Suma	17000	1

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

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

$$w_k \equiv \frac{C}{t_{REF,k}}$$

Igualmente, se calculan:  
 $\bar{t}_{WB}, \bar{t}_{WC}$  y  $\bar{t}_{WD}$



- Según este criterio, la máquina más "rápida" sería la de mejor tiempo medio ponderado de ejecución. Nótese que esta ponderación depende, en este ejemplo, de la máquina de referencia.

42

## Normalización de rendimientos: ratios

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

Programa	Ref (s)	A (ratio)	B (ratio)	C (ratio)	D (ratio)
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

- Ahora la ratio (=ganancia en velocidad con respecto a la máquina de referencia) es un índice a maximizar. Según los resultados, la mejor máquina es ¡¡¡la D!!!

43

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

J8									
=MEDIA.GEOM(J2:J5)									
	A	B	C	D	E	F	G	H	I
Prog. Bench.	Tiempo Ejecución Máquina	A(s)	B(s)	C(s)	D(s)	R/A	R/B	R/C	R/D
Ref (s)	Ref (s)								
1	1	200	100	99	1	2,00	2,02	200,0	200,0
2	2	200	100	101	133	2,00	1,98	1,50	200,0
3	3	200	100	100	133	2,00	2,00	1,50	200,0
4	4	200	100	100	133	397	2,00	2,00	1,50
5	Suma	800	400	400	400				
6									
7									
8									
Media Geométrica						2,0000	2,0001	5,11	44,81

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.

44

## Conclusiones de este análisis

- Intentar reducir un conjunto de medidas de un benchmark a un solo “valor medio” final no es una tarea trivial.
- La media aritmética de los tiempos de ejecución de un benchmark 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 benchmark 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 que no depende la máquina de referencia. Premia mejoras sustanciales con respecto a algún programa del benchmark y no castiga al mismo nivel los empeoramientos.
- Independientemente de qué índice se escoja, un buen ingeniero debería en primer lugar determinar si las diferencias entre las diferentes medidas obtenidas son **estadísticamente significativas**. ¿Qué significa eso?

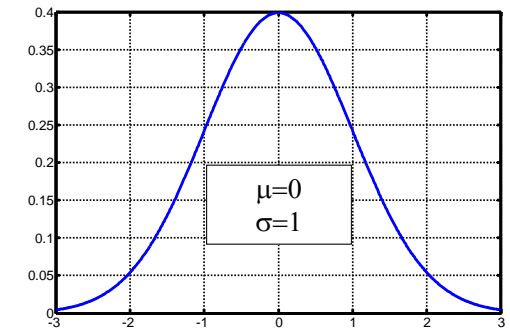
45

## Repaso de Estadística: Distribución Normal

- Es una distribución caracterizada por su media  $\mu$  y su varianza  $\sigma^2$  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 suma de un conjunto de muestras aleatorias pertenecientes a cualquier distribución e independientes entre sí tiende a una distribución normal.

46

## Repaso de Estadística: Distribución t de Student

Si disponemos de  $n$  muestras  $d_i$  pertenecientes a una distribución Normal de media  $\bar{d}_{real}$ , el número (=estadístico):

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

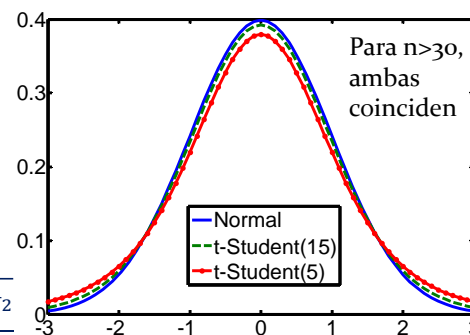
siendo  $\bar{d}$  la media muestral:

$$\bar{d} = \frac{\sum_{i=1}^n d_i}{n}$$

y  $s$  la desviación típica muestral:

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

se distribuye según la distribución t-Student con  $n-1$  grados de libertad. **¿Para qué me puede servir esto?**



47

## Ejemplo1

- Tiempos de ejecución (en segundos) de un benchmark compuesto por 6 programas (P1...P6) en dos máquinas diferentes (A y B)

Programa	tA (s)	tB (s)	$d_i = tA_i - tB_i$
P1	142	100	42
P2	139	92	47
P3	152	128	24
P4	112	82	30
P5	156	148	8
P6	166	171	-5
Suma	867	721	

¿Son significativas estas diferencias?

$$\bar{d} = 24,3 \text{ seg}$$

$$s = 19,9 \text{ seg}$$

- Si partimos de la hipótesis de que las máquinas tienen rendimientos equivalentes, entonces las diferencias se deben a factores aleatorios independientes. En ese caso  $d_i$  serán muestras de una distribución normal de media cero. Por tanto:

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

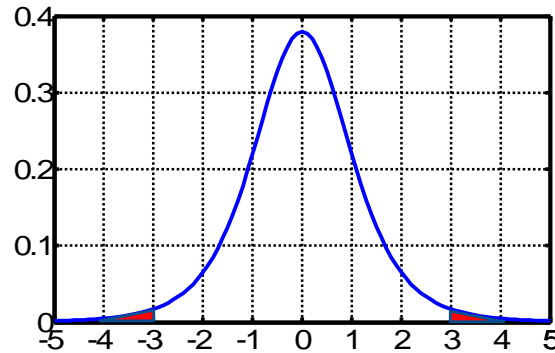
pertenecerá a una distribución t de Student con  $6-1=5$  grados de libertad. ¿Qué probabilidad hay de que esto sea realmente así?

48



# Nivel o Grado de Significatividad ( $\alpha$ )

- Distribución t de Student con 5 grados de libertad:



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

$$P\text{-value} = P(|t| \geq 2,99; 5) = 2 \times P(t < -2,99; 5)$$

$$= 2 \cdot \text{tcdf}(-|2,99|, 5) = 0.0304 \quad (\text{Matlab})$$

$$= \text{DISTR.T.2C}(|2,99|; 5) = 0.0304 \quad (\text{Excel})$$

$$= \text{DISTR.T}(|2,99|; 5; 2) = 0.0304 \quad (\text{Calc})$$

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 3,04% (**P-value** (Valor-P)= 0.0304). ¿Es eso mucho o poco? Debemos definir un umbral: **nivel o grado de significatividad  $\alpha$** .

Conclusión: Si  $P\text{-value} < \alpha$  diremos que, para un grado de significatividad  $\alpha$  o para un **nivel de confianza**  $(1-\alpha) \cdot 100 = 95\%$ , las máquinas tienen rendimientos estadísticamente diferentes. En ese caso, B sería 1.2 veces más rápida que A en ejecutar el benchmark ( $867/721=1.2$ ). En caso contrario, no podríamos descartar la hipótesis de que las máquinas tengan rendimientos equivalentes.

49

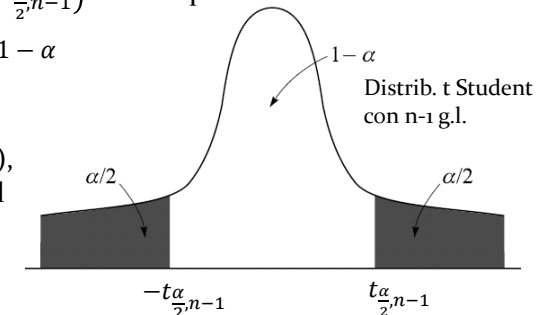
## Intervalos de confianza

- Para un nivel de significatividad  $\alpha$  (típ.  $0.05 = 5\%$ ), buscamos el valor  $t_{\alpha/2, n-1}$  que cumpla  $Prob(|t| > t_{\alpha/2, n-1}) = \alpha$  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  $1-\alpha$  (típ.  $0.95 = 95\%$ ), el valor de t debe 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  $\alpha$ . Teniendo en cuenta que:

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

es fácil demostrar que  $t_{\alpha/2, n-1}$  cumple que (ver figura):

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

50

## Intervalos de confianza para $t_{exp}$

- En el caso del *Ejemplo 1*, 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. Eso se puede obtener, por ejemplo:

- En Matlab, haciendo:  $\text{abs}(\text{tinv}(\alpha/2, n-1)) = \text{abs}(\text{tinv}(0.025, 5)) = 2.57$
- En Excel, haciendo:  $\text{ABS}(\text{INV.T}(\alpha/2; n-1)) = \text{ABS}(\text{INV.T}(0,025; 5)) = 2.57$  ó directamente:  $\text{INV.T.2C}(\alpha; n-1) = \text{INV.T.2C}(0,05; 5) = 2.57$
- En Calc,  $\text{DISTR.T.INV}(\alpha; n-1) = \text{DISTR.T.INV}(0,05; 5) = 2.57$

- 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

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

se encuentre en el rango  $[-t_{\alpha/2, n-1}, t_{\alpha/2, n-1}] = [-t_{0.025, 5}, t_{0.025, 5}] = [-2.57, 2.57]$ .

➔ Como  $t_{exp}=2.99$  no está en ese rango, concluiremos que la hipótesis de que ambas máquinas pueden ser iguales no es cierta al 95% de confianza.

51

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

- 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

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

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

$$\bar{d}_{real} \in \left[ \bar{d} - \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1}, \bar{d} + \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1} \right] = 24.3 \mp 20.9 = [3.4, 45.2] \text{ s}$$

Y el problema se transforma simplemente en comprobar si ese valor medio real  $\bar{d}_{real}$  puede o no ser **cero**.

➔ En nuestro ejemplo, como el intervalo no incluye el cero, concluiremos que la hipótesis de que ambas máquinas pueden ser iguales no es cierta al 95% de confianza.

52

## En resumen: Test t (valor-p o p-value)

- Ejecución de  $n$  programas en dos máquinas A y B.
- ¿Son significativas las diferencias obtenidas ( $d_i = tA_i - tB_i$ )? Hay que usar mecanismos estadísticos.

- Calculo:

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

$$P - value = P(|t| \geq t_{exp}; n-1)$$

- Concluiremos, para un nivel de confianza del  $(1-\alpha) \times 100\%$  (típ. 95%) o para un nivel de significatividad de  $\alpha$  (típ. 5%):
  - Si  $P\text{-value} \geq \alpha$ , entonces no hay diferencias significativas (es posible que los valores de  $d_i$  sean aleatorios  $\rightarrow$  las dos alternativas pueden tener rendimientos equivalentes).
  - Si  $P\text{-value} < \alpha$ , entonces las alternativas presentan rendimientos significativamente diferentes. La que sea mejor dependerá del índice de rendimiento que se considere (tiempos medios, SPEC, etc.)

53

## Resumen: Test t (Intervalos de confianza para $t_{exp}$ )

- Ejecución de  $n$  programas en dos máquinas A y B.
- ¿Son significativas las diferencias obtenidas ( $d_i = tA_i - tB_i$ )? Hay que usar mecanismos estadísticos.

- Calculo:  $t_{exp} = \frac{\bar{d}}{s/\sqrt{n}}$  siendo  $\bar{d} = \frac{\sum_{i=1}^n d_i}{n}$   $s = \sqrt{\frac{\sum_{i=1}^n (d_i - \bar{d})^2}{n-1}}$

- Intervalo de confianza para  $t_{exp}$  (para un nivel de significatividad  $\alpha$  predeterminado, típ. 0.05):

$$[-t_{\alpha/2, n-1}, t_{\alpha/2, n-1}]$$

siendo  $t_{\alpha/2, n-1}$  el valor que hace que  $Prob(t \leq -t_{\alpha/2, n-1}) = \alpha/2$  para una distribución t de Student con  $n-1$  grados de libertad.

- Concluiremos, para un nivel de confianza del  $(1-\alpha) \times 100\%$  (típ. 95%) o para un nivel de significatividad  $\alpha$  (típ. 5%):
  - Si  $t_{exp}$  está en el intervalo, entonces no hay diferencias significativas.
  - Si no lo está, entonces las alternativas presentan rendimientos significativamente diferentes.

54

## Resumen: Test t (Intervalos de confianza para $\bar{d}_{real}$ )

- Ejecución de  $n$  programas en dos máquinas A y B.
- ¿Son significativas las diferencias obtenidas ( $d_i = tA_i - tB_i$ )? Hay que usar mecanismos estadísticos.
- Intervalo de confianza para la media real de las diferencias  $\bar{d}_{real}$  (para un nivel de significatividad  $\alpha$  predeterminado, típ. 0.05):

$$\bar{d}_{real} \in \left[ \bar{d} - \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1}, \bar{d} + \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1} \right] \equiv \bar{d} \pm \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1}$$

siendo  $t_{\alpha/2, n-1}$  el valor que hace que  $Prob(t \leq -t_{\alpha/2, n-1}) = \alpha/2$  para una distribución t de Student con  $n-1$  grados de libertad.

- Concluiremos, para un nivel de confianza del  $(1-\alpha) \times 100\%$  (típ. 95%) o para un nivel de significatividad  $\alpha$  (típ. 5%):
  - Si el intervalo incluye el cero, entonces no hay diferencias significativas.
  - Si no incluye el cero, entonces las alternativas presentan rendimientos significativamente diferentes.

55

## Ejemplo 2

- Tiempos de ejecución (en segundos) de un benchmark compuesto por 5 programas (P1...P5) en dos máquinas diferentes (A y B)

Programa	tA (s)	tB (s)	$d_i = tA_i - tB_i$
P1	23	15	8
P2	28	22	6
P3	19	20	-1
P4	29	27	2
P5	36	39	-3
Suma	135	123	

¿Son significativas estas diferencias?  
dato:  $|t_{0.025, 4}| = 2.78$

$$\bar{d} = 2.4s$$

$$s = 4.6s$$

$$t_{exp} = \frac{\bar{d}}{s/\sqrt{n}} = 1.16$$

- $P - value = P(|t| \geq t_{exp}; n-1) = P(|t| \geq 1.16; 4) = 0.31 (> 0.05)$
- Para un nivel de significatividad de  $\alpha=0.05$ :
  - Intervalo de confianza para  $t_{exp}$ :  $[-2.78, 2.78]$  (**dentro del intervalo**)
  - Intervalo para  $\bar{d}_{real}$ : (**incluye el cero**)

$$\bar{d} \pm \frac{s}{\sqrt{n}} \times t_{\alpha/2, n-1} = 2.4 \pm \frac{4.6}{\sqrt{5}} \times 2.78 = 2.4 \pm 5.72 = [-3.3, 8.1]s$$

➔ NO podemos descartar, al 95% de nivel de confianza, que ambas máquinas puedan ser iguales.

56

# Test T con Statgraphics

	ej1A	ej1B	ej2A	ej2B	Cc
1	142	100	23	15	
2	139	92	28	22	
3	152	128	19	20	
4	112	82	29	27	
5	156	148	36	39	
6	166	171			
7					

## Prueba de Hipótesis para ej1A - ej1B

Prueba t

Hipótesis Nula: media = 0

Alternativa: no igual

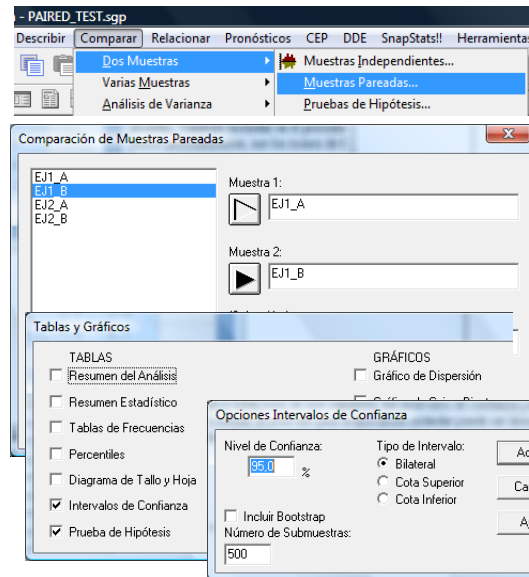
Estadístico t = 2,9912

Valor-P (P-value) = 0,0304056

Se rechaza la hipótesis nula para  $\alpha = 0,05$ .

## Intervalos de Confianza para ej1A - ej1B

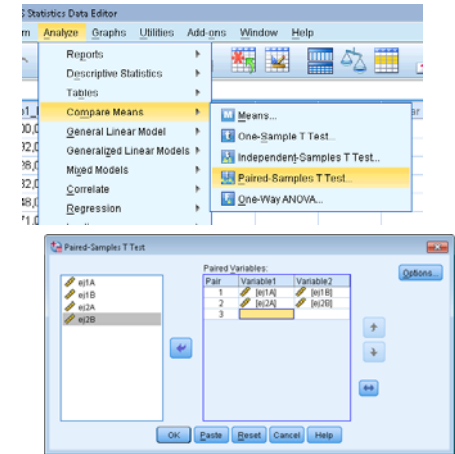
Intervalos de confianza del 95,0% para la media: 24,3333 +/- 20,9117 [3,42166; 45,245]



57

# Test T con SPSS

	ej1A	ej1B	ej2A	ej2B
1	142,00	100,00	23,00	15,00
2	139,00	92,00	28,00	22,00
3	152,00	128,00	19,00	20,00
4	112,00	82,00	29,00	27,00
5	156,00	148,00	36,00	39,00
6	166,00	171,00		
7				



Paired Samples Test									
		Paired Differences					$t_{exp}$ ↓ t	df	Sig. (2-tailed) ↓ P-value
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	ej1A - ej1B	24,33333	19,92653	8,13497	3,42172	45,24495	2,991	5	,030
Pair 2	ej2A - ej2B	2,40000	4,61519	2,06398	-3,33052	8,13052	1,163	4	,316

58

## Intervalos de confianza de medias experimentales

**Hipótesis:** Realizamos n medidas  $d_i$  de un mismo fenómeno (p.ej. tiempos de ejecución de un programa, temperaturas CPU, tiempos acceso disco duro, productividades red,...). Estas pueden diferir debido a efectos aleatorios. Supondremos que  $\{d_i\}$  se distribuye como una distribución normal de media  $\bar{d}_{real}$ . En ese caso, sabemos que

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

se distribuye según 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:

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

**Utilidad:** Podemos usar esta información para saber si hace falta realizar más pruebas experimentales para determinar  $\bar{d}_{real}$  con mejor precisión.

59

## Ejemplo

Queremos determinar un intervalo de confianza para el tiempo medio de escritura de un determinado fichero en un disco duro. Para ello, se han realizado varias medidas experimentales:

#exp	$t_w$ (ms)
1	835
2	798
3	823
4	803
5	834
6	825
7	813
8	829

$$\bar{t}_w = \frac{\sum_{i=1}^n t_{wi}}{n} = 820ms \quad s = \sqrt{\frac{\sum_{i=1}^n (t_{wi} - \bar{t}_w)^2}{n-1}} = 14ms$$

**Por tanto**, hay un 90% ( $\alpha=0.1$ ) de probabilidad de que el tiempo medio de escritura real de ese fichero se encuentre en el intervalo:

$$820 \pm \frac{14}{\sqrt{8}} t_{0.1/2, 8-1} = [811, 829]ms$$

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

$$820 \pm \frac{14}{\sqrt{8}} t_{0.05/2, 8-1} = [808, 832]ms$$

60

## 4.4 Introducción al diseño de experimentos

### Planteamiento del problema

- Supongamos que queremos determinar cuáles de los siguientes factores afectan significativamente al rendimiento de un determinado equipo para un determinado benchmark:
  1. Sistema Operativo: Windows Server, CentOS, Debian, Ubuntu.
  2. Memoria RAM: 32GB, 64GB, 128GB.
  3. Discos duros: SATA, IDE, SAS.
- ¿Qué experimentos debemos diseñar para ello y cómo debemos analizar los resultados?



62

### 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., 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.
- **Repetición:** El número de veces que se repite cada experimento.
- **Interacción:** El efecto de un determinado nivel de un factor sobre la variable respuesta puede ser diferente para cada nivel 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 RAM.

63

### 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.

64



## 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 del mismo. Para ello, se ha escogido tres tipos de discos duros: **SAS, SATA e IDE** y se ha realizado un experimento que consiste en ejecutar un conjunto de programas de prueba usados habitualmente por el servidor y medir el **tiempo de ejecución**. Todos los experimentos se han 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 ( $\varepsilon_j$ )	-10.9	-5.1	16.1

$m_{\text{global}} = 119.9s$

- ¿Tiene influencia el factor disco duro sobre el rendimiento? ¿Son las diferencias entre los discos duros significativas? **Test ANOVA**

65

## 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).

$m_{\text{global}}$ : Media global de todas las observaciones:

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

$$\varepsilon_j: \text{Efecto debido al nivel } j\text{-ésimo: } \varepsilon_j = \frac{1}{n_{\text{rep}}} \sum_{i=1}^{n_{\text{rep}}} y_{ij} - m_{\text{global}}$$

$$\text{Se cumple que } \sum_{j=1}^{n_{\text{niv}}} \varepsilon_j = 0$$

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

- Que tengan varianza constante, independiente del nivel.
- Que su distribución sea normal.

- La pregunta que intenta contestar el test ANOVA es: ¿Tiene influencia el factor sobre la variable respuesta (algún  $\varepsilon_j$  es distinto de cero)? ¿Son los efectos de cada nivel ( $\varepsilon_j$ ) significativamente diferentes unos de otros?

66

## Análisis de la Varianza (ANOVA) de un factor (II)

El método 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 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 de que el factor no influye sobre los resultados ( $\varepsilon_j \approx 0 \forall j = 1 \dots n_{\text{niv}}$ ). Si esto es cierto, resulta que 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.

67

## Análisis de la Varianza (ANOVA) de un factor (III)

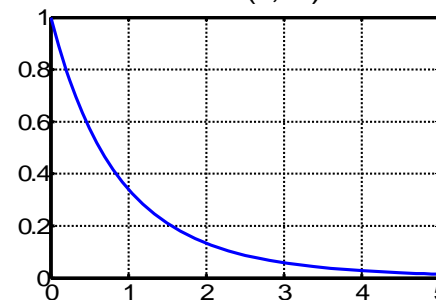
En nuestro ejemplo:

$$\begin{aligned} \text{SST} &= 2809 \\ \text{SSA} &= 2020 \\ \text{SSE} &= 789 \end{aligned} \quad F_{\text{exp}} \equiv \frac{\frac{\text{SSA}}{n_{\text{niv}} - 1}}{\frac{\text{SSE}}{(n_{\text{niv}} \times (n_{\text{rep}} - 1))}} = \frac{\frac{2020}{3-1}}{\frac{789}{3 \times (5-1)}} = 15.37$$

¿Qué probabilidad hay de que la muestra 15.37 se haya extraído de una distribución  $F_{2,12}$ ?  $P - \text{value} = P(F \geq 15.37; 2,12) = 5 \cdot 10^{-4}$ .

Matlab: 1-fcdf(15.37,2,12); Excel y Calc: DISTR.F(15.37;2;12).

FPDF(2,12)



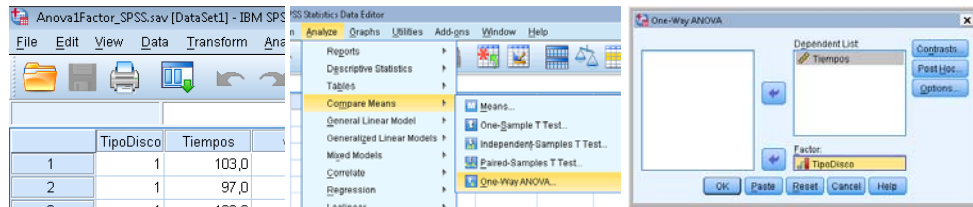
Igual que con los ejemplos anteriores, si la probabilidad es menor que  $\alpha=0.05$  diremos que *descartamos la hipótesis de que el factor no influya* a un  $(1-\alpha) \times 100\% = 95\%$  de confianza.

Si el factor influye, a continuación (post-hoc) comparamos las medias de cada nivel unas con otras usando esencialmente el *test t* visto anteriormente (**prueba de múltiples rangos o de comparaciones múltiples**).

68

# Diseños con un solo factor con SPSS

1: SAS; 2: SATA; 3: IDE



**ANOVA**

Dependent Variable	Tiempos	Sum of Squares	df	Mean Square	F	Sig.
Between Groups		2020,133	2	1010,067	15,366	,000
Within Groups		788,800	12	65,733		
Total		2808,933	14			

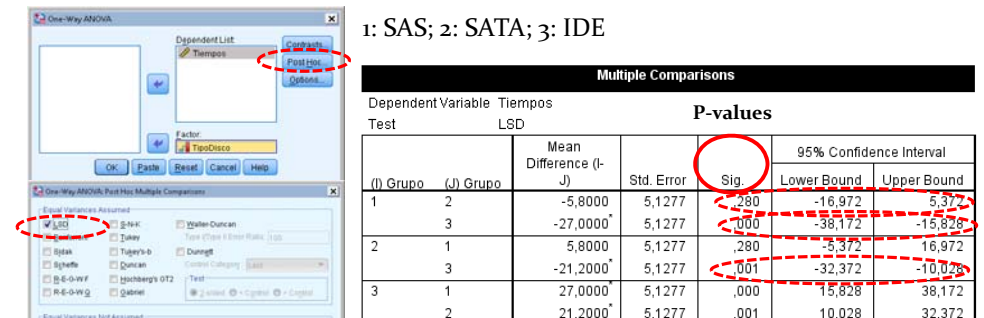
Esto demuestra que el tipo de disco duro afecta significativamente al rendimiento del equipo casi para cualquier nivel de significatividad que usemos.

69

# Diseños con un solo factor con SPSS (II)

Ahora queremos hacer un contraste por parejas para comparar el efecto de cada tipo de disco duro unos con otros.

1: SAS; 2: SATA; 3: IDE



**Multiple Comparisons**

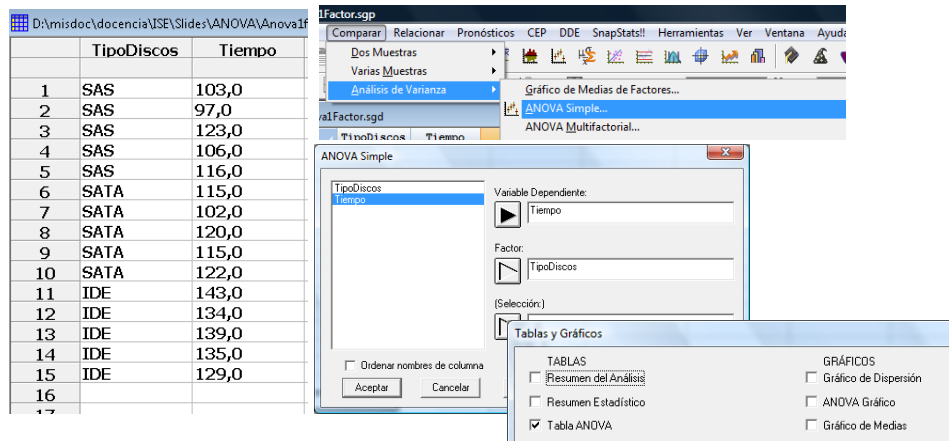
Test	Dependent Variable	Tiempos	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval
1	2	-5,8000*	5,1277	,280	-16,972	5,372
1	3	-27,0000*	5,1277	,000	-38,172	-15,828
2	1	5,8000	5,1277	,280	-5,372	16,972
2	3	-21,2000*	5,1277	,001	-32,372	-10,028
3	1	27,0000*	5,1277	,000	15,828	38,172
3	2	21,2000*	5,1277	,001	10,028	32,372

\*. The mean difference is significant at the 0.05 level.

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 (incluyen al cero), por lo que podríamos decidirnos por el más barato (o hacer más pruebas para estar más seguros).

70

# Diseños con un solo factor con Statgraphics



**ANOVA Simple**

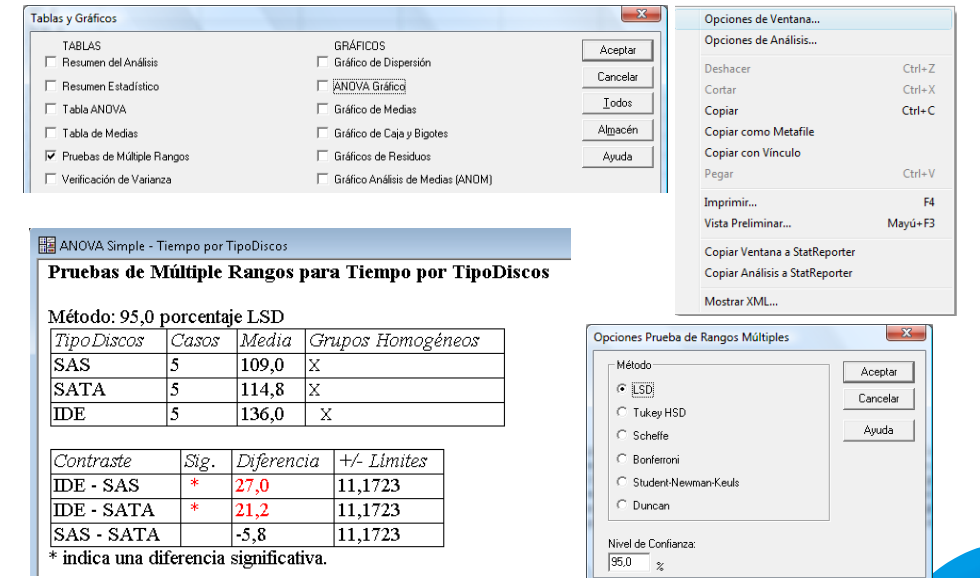
Variable Dependiente:	Tiempo
Factor:	TipoDiscos

**Tabla ANOVA para Tiempos por Tipo Disco**

Fuente	Suma de Cuadrados	Gl	Cuadrado Medio	Razón-F	Valor-P
Entre grupos	2020,13	2	1010,07	15,37	0,0005
Intra grupos	788,8	12	65,7333		
Total (Corr.)	2808,93	14			

71

# Diseños con un solo factor con Statgraphics (II)



**ANOVA Simple - Tiempo por TipoDiscos**

**Pruebas de Múltiple Rangos para Tiempo por TipoDiscos**

Método: 95,0 porcentaje LSD

TipoDiscos	Casos	Media	Grupos Homogéneos
SAS	5	109,0	X
SATA	5	114,8	X
IDE	5	136,0	X

Contraste	Sig.	Diferencia	+/- Limites
IDE - SAS	*	27,0	11,1723
IDE - SATA	*	21,2	11,1723
SAS - SATA		-5,8	11,1723

\* indica una diferencia significativa.

72