

EVALUACION DEL RENDIMIENTO DE COMPUTADORES (II)

Programas basados en código real

1. Objetivo

El objetivo de la presente práctica es el conocimiento de algunos ejemplos de programas de evaluación de rendimiento de computadores (*benchmarks*) basados en código real. Los programas concretos que se utilizarán son el **Linpack**, el **Flops** como ejemplos de programas de tamaño reducido y el conjunto de *benchmark SPEC* como ejemplos de programas reales, estos últimos desarrollados por la organización del mismo nombre (*Standard Performance Evaluation Cooperative*). Todos estos programas tratan de establecer una comparativa realista del rendimiento de los diferentes computadores que van apareciendo en el mercado.

En los Anexos I, II y III de la práctica se resumen las características de los 3 *benchmarks* aludidos. En el caso del conjunto de *benchmarks SPEC*, cada *benchmark* estudia un determinado tipo de rendimiento del sistema computador (CPU, sistema de entrada/salida, aplicaciones cliente/servidor, etc.). Uno de los *benchmarks SPEC* más utilizado es el relativo al *Rendimiento de CPU* del computador, cuya última versión es del año 2006 (*SPEC CPU2006*), y que considera tanto trabajo con enteros (*SPECint*) como trabajo con flotantes (*SPECfp*). Este *benchmark* es el más utilizado, dentro de su categoría, en la comparativa rigurosa de computadores.

2. Análisis a realizar

1. Analizar los *benchmarks Linpack* y *Flops*, comparar máquinas en base a ellos y valorar la precisión de la estimación de rendimiento proporcionada por los mismos.
2. Analizar la documentación relativa al conjunto de *benchmarks SPEC*.
3. Analizar las tablas de métricas correspondientes al *benchmark SPEC CPU* publicadas periódicamente por la organización *SPEC* y realizar comparativas de máquinas en base a ellas.

3. Pasos a seguir en la práctica

- Leer el Anexo I de la práctica. Copiar el archivo fuente del *benchmark Linpack* en las 4 máquinas de la práctica I (.../fran/4atc/linpack/linpack.c). Generar el ejecutable (utilizar optimización), ejecutar el *benchmark*, interpretar los resultados y anotar las correspondientes medidas de rendimiento.
- Leer el Anexo II de la práctica. Copiar el archivo fuente del *benchmark Flops* en las mismas máquinas (.../fran/4atc/flops/flops.c). Generar el ejecutable (utilizar optimización), ejecutar el *benchmark*, interpretar los resultados y elegir de entre ellos el que mejor puede estimar el rendimiento obtenido al trabajar con la aplicación de usuario descrita en la práctica I (*mxvdouble*) ¿Qué criterio de elección es el que has utilizado?. Anotar las correspondientes medidas de rendimiento.
- Leer el Anexo III de la práctica. ¿Qué mide realmente *SPEC CPU95*, solo el rendimiento de la CPU o más cosas? ¿Cuántos programas forman *SPEC CPU95* y de que tipo son? ¿Qué métricas se utilizan y con respecto a que máquina se calculan? ¿Cómo se calculan las métricas en función de los tiempos medidos en los programas? ¿Cuál es la diferencia entre las métricas de velocidad (*speed*) y de productividad (*throughput*)? ¿Cuál es la diferencia entre las métricas base (*baseline*) y las de pico (*peak*)?
- Entrar en la página web de *SPEC* a través del correspondiente enlace en la página web de la asignatura. Analizar el resto de *benchmarks (además del CPU)* desarrollados por *SPEC*.
- Localizar en la página web el enlace a ***SPEC CPU95***. Estudiar la documentación (además de la que está en la página de *SPEC* se puede acceder al archivo *README* de la distribución de *SPEC CPU95* a través del enlace *readme95* de la página de la asignatura) ¿Qué programas se ejecutan en *SPEC CPU95* y que problemas resuelven? ¿Cuáles son sus tiempos de ejecución en la máquina de referencia? ¿Qué información de configuración de las máquinas evaluadas es necesario describir? ¿Cuál es el número mínimo de ejecuciones de los programas para que los resultados sean válidos? ¿Se ejecutan los programas directamente o mediante herramientas que automatizan el proceso?

- Consultar a través de la página *web* los resultados correspondientes a *SPECint95* y *SPECfp95* y localizar el modelo de máquina correspondiente a **Centauro** (*Digital AlphaServer 2100A 5/300*). Una vez encontrado y chequeadas las métricas de velocidad de base y de pico, visualizar el fichero de resultados correspondiente (p.e. en formato *HTML*) mediante el enlace habilitado para ello. Observar detalladamente los resultados de cada uno de los programas y analizar los diferentes factores que han influido en los resultados obtenidos en dicha máquina, tales como frecuencia del procesador, memoria instalada, compilador utilizado, optimizaciones, etc.
- Localizar en la página *web* principal el enlace a **SPEC CPU2000**. Estudiar la documentación (además de la que está en la página de SPEC se puede acceder al archivo *README* de la distribución de *SPEC CPU2000* a través del enlace *readme2000* de la página de la asignatura) y analizar los cambios más destacables con respecto a la versión anterior. Localizar los modelos más similares a **Orion** (*Intel D815EEA2 motherboard, 1.0 GHz Pentium III processor*) y **Sirio** (*Sun Ultra 25*) y repetir el procedimiento descrito más arriba.
- Localizar en la página *web* principal el enlace a **SPEC CPU2006**. Estudiar la documentación (además de la que está en la página de SPEC se puede acceder al archivo *README* de la distribución de *SPEC CPU2006* a través del enlace *readme2006* de la página de la asignatura) y analizar los cambios más destacables con respecto a la versión anterior. Localizar el modelo de máquina correspondiente a **Lisa** (*Dell PowerEdge 2950 / 2,66GHz.*) y repetir el procedimiento descrito más arriba.
- Explorar los resultados publicados para *SPEC CPU2006* y comparar resultados de los modelos de los diferentes fabricantes tanto para enteros como para flotantes.

4. Trabajo a entregar

- Tabla de datos de rendimiento, *ranking* de velocidad, comparativa de máquinas y diferencia media entre estimaciones de *benchmark* y aplicación (*mxvdouble*) tanto para *Linpack* como para *Flops*, según modelo del anexo IV ¿Qué *benchmark* de los tres analizados (incluyendo el *whetstone* de la práctica anterior) estima de forma más precisa el rendimiento de las máquinas con *mxvdouble*?
- Métricas de velocidad (de base) del modelo de máquina correspondiente a **Centauro** para *SPECint95* y *SPECfp95*, según modelo del anexo V (tabla 1) ¿Varía la métrica al pasar de enteros a flotantes?
- Tabla de métricas de velocidad (de base) de **Centauro** para cada una de las aplicaciones *SPECfp95* y gráfico de barras resumen, según modelo del anexo V (tabla 2) ¿Varía el rendimiento de la máquina con las diferentes aplicaciones?
- *Rankings* de velocidad de los modelos de máquina correspondientes a **Sirio** y **Orion** para *SPECint2000* y *SPECfp2000*, utilizando las métricas de velocidad (de base), según modelo del anexo V (tabla 1) ¿Cómo influyen los diferentes parámetros de las configuraciones (*hardware + software*) sobre el *ranking*? ¿Varía el *ranking* al pasar de enteros a flotantes?
- *Rankings* de velocidad de **4 modelos de computadores** para *SPECint2006* y *SPECfp2006*, utilizando las métricas de velocidad (de base), según modelo del anexo V (tabla 3), considerando dos casos:
 - a) computadores de diferentes fabricantes con 1 solo chip procesador e idéntica frecuencia
 - b) computadores del mismo fabricante (modelos similares) con 1 chip procesador y diferentes frecuencias

¿Cómo influyen los diferentes parámetros de las configuraciones (*hardware + software*) sobre los *rankings*?
¿Varía los *rankings* al pasar de enteros a flotantes?
- *Rankings* de velocidad de **4 modelos de computadores** para *SPECint2006* y *SPECfp2006*, utilizando las métricas de productividad (de base), según modelo del anexo V (tabla 3), considerando modelos similares del mismo fabricante, frecuencias de procesador similares y con 2, 4, 8 y 16 núcleos respectivamente

¿Cómo influyen los diferentes parámetros de las configuraciones (*hardware + software*) sobre los *rankings*?
¿Varía el *ranking* al pasar de enteros a flotantes? ¿Crece el valor de las métricas con el número de núcleos?
- Actualización del documento *Excel* con una nueva hoja (nombrar la nueva pestaña como *práctica 2*).

Anexo I: El benchmark Linpack

The LINPACK benchmark, developed by Jack Dongarra, reflects the performance of computers ranging from the home-used PC to the most powerful supercomputers when solving a dense system of linear equations. Since this operation is a component of many technical applications, the LINPACK benchmark provides some insight into the levels of performance that can be achieved in real applications.

We use that version of the benchmark that allows the user to scale the size of the problem and to optimize the software in order to achieve the best performance for a given machine. This performance does not reflect the overall performance of a given system, as no single number ever can. It does, however, reflect the performance of a dedicated system for solving a dense system of linear equations. Since the problem is very regular, the performance achieved is quite high, and the performance numbers give a good correction of peak performance.

Anexo II: El benchmark Flops

Flops.c is a 'c' program which attempts to estimate your systems floating-point 'MFLOPS' rating for the FADD, FSUB, FMUL, and FDIV operations based on specific 'instruction mixes' (discussed below). The program provides an estimate of PEAK MFLOPS performance by making maximal use of register variables with minimal interaction with main memory. The execution loops are all small so that they will fit in any cache. Flops.c can be used along with Linpack and the Livermore kernels (which exercise memory much more extensively) to gain further insight into the limits of system performance. The flops.c execution modules also include various percent weightings of FDIV's (from 0% to 25% FDIV's) so that the range of performance can be obtained when using FDIV's. FDIV's, being computationally more intensive than FADD's or FMUL's, can impact performance considerably on some systems.

Flops.c consists of 8 independent modules (routines) which, except for module 2, conduct numerical integration of various functions. Module 2 estimates the value of pi based upon the Maclaurin series expansion of atan(1). MFLOPS ratings are provided for each module, but the programs overall results are summarized by the MFLOPS(1), MFLOPS(2), MFLOPS(3), and MFLOPS(4) outputs.

The MFLOPS(1) result is identical to the result provided by all previous versions of flops.c. It is based only upon the results from modules 2 and 3. Two problems surfaced in using MFLOPS(1). First, it was difficult to completely 'vectorize' the result due to the recurrence of the 's' variable in module 2. This problem is addressed in the MFLOPS(2) result which does not use module 2, but maintains nearly the same weighting of FDIV's (9.2%) as in MFLOPS(1) (9.6%). The second problem with MFLOPS(1) centers around the percentage of FDIV's (9.6%) which was viewed as too high for an important class of problems. This concern is addressed in the MFLOPS(4) result where NO FDIV's are conducted at all.

The number of floating-point instructions per iteration (loop) is given below for each module executed:

MODULE	FADD	FSUB	FMUL	FDIV	TOTAL	Comment
1	7	0	6	1	14	7.1% FDIV's
2	3	2	1	1	7	difficult to vectorize.
3	6	2	9	0	17	0.0% FDIV's
4	7	0	8	0	15	0.0% FDIV's
5	13	0	15	1	29	3.4% FDIV's
6	13	0	16	0	29	0.0% FDIV's
7	3	3	3	3	12	25.0% FDIV's
8	13	0	17	0	30	0.0% FDIV's
A*2+3	21 40.4%	12 23.1%	14 26.9%	5 9.6%	52	A=5, MFLOPS (1) , Same as previous versions of the flops.c program. Includes only Modules 2 and 3, does 9.6% FDIV's, and is not easily vectorizable.
1+3+4 +5+6+ A*7	58 38.2%	14 9.2%	66 43.4%	14 9.2%	152	A=4, MFLOPS (2) , New output does not include Module 2, but does 9.2% FDIV's.
1+3+4 +5+6+ 7+8	62 42.9%	5 3.4%	74 50.7%	5 3.4%	146	A=0, MFLOPS (3) , New output does not include Module 2, but does 3.4% FDIV's.
3+4+6 +8	39 42.9%	2 2.2%	50 54.9%	0 0.0%	91	A=0, MFLOPS (4) , New output does not include Module 2, and does NO FDIV's.

Anexo III: El conjunto de benchmarks SPEC

The SPEC benchmark family is developed by a non-profit corporation called *Standard Performance Evaluation Corporation*. SPEC emphasizes developing real applications benchmarks that closely reflect the actual workload, instead of small, synthetic kernels. For each benchmark and benchmark suite, SPEC defines a few (two in many cases) metrics that measure the overall performance of the entire system.

SPEC started with benchmarks that measure CPU performance, but has extended to client/server computing, commercial applications, I/O subsystems, etc.

SPEC CPU95

These are CPU benchmarks measuring the CPU speed, the cache/memory system, and the compiler as a whole. The time spent in operating system and I/O functions is negligible. SPEC CPU95 consists of CINT95, a set of 8 integer programs; and CFP95, a set of 10 floating-point programs; both CPU-intensive applications.

SPEC periodically publishes performance results of various machines, both in hardcopy and on the Web. The table below shows the SPEC CPU95 results of Digital AlphaStation 500/5000 which uses a 500 MHz. Alpha 21164 processor with 8 MB cache and 128 MB memory. All SPEC95 results of a given system are expressed as ratios compared to a Sun SPARC station 10 (40MHz.), the reference machine. For instance, a value of 5 means that the measured machine achieves a performance that is 5 times that of the Sun SPARCstation 10, or 4 times faster.

Each metric is aggregated over all benchmarks of a suite by taking the geometric mean of the ratios of the individual benchmarks. The speed metrics measure the ratios to execute a single copy of the benchmark, while the throughput (rate) metrics measure the ratios to execute multiple copies of the benchmark. Results are obtained with conservative or aggressive optimizations and are reported as “baseline” and “peak” performance numbers respectively. The peak speeds (boldface in table) are frequently quoted SPEC CPU95 numbers.

Metric	Speed		Throughput	
	95	_base95	_rate95	_rate_base95
CINT	15	12.6	135	113
CFP	20.4	18.3	183	165

The Digital AlphaStation 500/500 has SPECint_rate_base95 = 113 and SPECfp95 = 20.4.
What do these numbers mean?

The SPECint_rate_base95 number is obtained by taking the geometric mean of the rates of the eight benchmarks of the CINT95 suite, where each benchmark is compiled with a low optimization (e.g., -O). The rate of each benchmark is measured by running multiple copies of the benchmark, and the execution time is normalized with respect to Sun SPARCstation 10. The number 113 means that the Digital machine executes 112 times more copies of CINT95 than Sun SPARCstation 10.

The SPECfp95 number is obtained taking the geometric mean of the ratios of the 10 benchmarks of the CFP95 suite, where each benchmark is compiled with aggressive optimization (e.g., -O3). The benchmark is measured by running a single copy with execution time normalized with respect to a Sun SPARCstation 10. The number 20.4 means that the Digital machine is 19.4 times faster than Sun SPARCstation 10 for executing a single copy of CFP95.

SPEC CPU2000

SPEC CPU2000 consists of CINT2000, a set of 11 integer programs; and CFP2000, a set of 14 floating-point programs.

All SPEC2000 results of a given system are expressed as ratios (**x100 for speed metrics**) compared to a Sun Ultra 10 (300MHz.), the reference machine. A value of 500 means that the measured machine achieves a performance that is 5 times that of the Sun Ultra 10, or 4 times faster.

SPEC CPU2006

The SPEC CPU95 benchmarks are the current version of the most famous SPEC benchmarks widely used by vendors and users. SPEC CPU2006 consists of CINT2006, a set of 12 integer programs; and CFP2006, a set of 17 floating-point programs.

All SPEC2006 results of a given system are expressed as ratios compared to a Sun Ultra Enterprise 2 (296MHz.) the reference machine. A value of 5 means that the measured machine achieves a performance that is 5 times that of the Sun Ultra Enterprise 2, or 4 times faster.

It takes about 12 days to do a rule-conforming run of the base metrics for CINT2006 and CFP2006 on the reference machine.

Anexo IV: Tablas de Excel para Linpack y Flops

	Lisa	Sirio	Orion	Centauro
<i>benchmark</i>				
<i>mxvdouble</i>				

1) Ranking

	Máquina	MFLOPS
1		
2		
3		
4		

2) Comparativa

Lisa	-			
	-			
Sirio		-		
		-		
Orion			-	
			-	
Centauro				-
				-
	Lisa	Sirio	Orion	Centauro

3) Precisión

Diferencia media entre estimaciones de *benchmark* y aplicación

%

¿Qué benchmark de los tres analizados (*whetstone*, *flops* y *linpack*) estima de forma más precisa el rendimiento de las máquinas con *mxvdouble*?

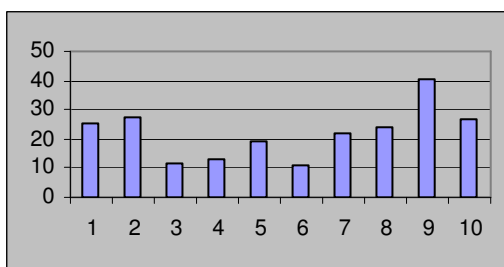
Anexo V: Tablas y gráfica de Excel para SPEC CPU

CINT_base95= CFP_base95=

¿Varía la métrica al pasar de enteros a flotantes? Sí/No

(1) Modelo de tabla para CFP_base95

Máquina	tomcatv	Swim	su2cor	Hydro2d	mgrid	applu	turb3d	apsi	fpppp	wave5



¿Varía el rendimiento de la máquina con las diferentes aplicaciones? Sí/No

(2) Modelo de tabla para CINT_base2000 y CFP_base2000

Rank	Máquina	Métrica	CPU frecuencia	MEM tamaño-tipo	Cache L1-L2-L3	Compilador
1						
2						

¿Cómo influyen los diferentes parámetros de las configuraciones (hardware + software) sobre los rankings?

¿Varía el ranking al pasar de enteros a flotantes? Sí/No

(3) Modelo de tabla para CINT_base2006, CFP_base2006, CINT_rate_base2006 y CFP_rate_base2006

Rank	Máquina	Métrica	CPU frecuencia	CPU Nº chips	CPU Nº núcleos	MEM tamaño-tipo	Cache L1-L2-L3	Compilador
1								
2								
3								
4								

¿Cómo influyen los diferentes parámetros de las configuraciones (hardware + software) sobre los rankings?

¿Varían los rankings al pasar de enteros a flotantes? Sí/No

¿Crece el valor de las métricas de productividad con el número de núcleos? Sí/No