

Tema 1

Aspectos fundamentales en el diseño de
computadores:
Rendimiento, consumo y coste

Contenido

- ¿Qué se entiende por arquitectura de computadores?
- Evolución del rendimiento de los computadores
- Tipos de computadores
- Medidas de rendimiento
- Consumo de potencia
- Coste
- Factores a tener en cuenta en el diseño de un computador

¿Qué se entiende por arquitectura?

- En el pasado se refería a la arquitectura del repertorio de instrucciones (ISA)
- Consideraremos 3 aspectos
 - el hardware sobre el que se ejecutan las instrucciones
 - el repertorio de instrucciones
 - la organización del computador

Evolución del rendimiento

- Un computador de \$500 actual rinde más que el que se podía obtener en 1985 por 1 millón de dólares
- 3 grandes etapas
 - 1978 – 1986 : mejora del 25 % por año
 - 1986 – 2002 : mejora 52% por año
 - 2002 – 2005 : mejora 20% por año

Evolución del rendimiento

los procesadores serían 7 veces más lentos si sólo se produjeran mejoras tecnológicas

nuevas aportaciones en la organización

mejoras tecnológicas y arquitecturales



cada vez es más difícil sacar más rendimiento de un solo procesador. Comienza la era de los multiprocesadores

Rendimiento: Conceptos Básicos

- 2 puntos de vista en la medida:
 - **tiempo de respuesta o latencia** (tiempo de ejecución):
Tiempo requerido para completar una tarea
 - **Productividad** (*throughput*): número de tareas realizadas por unidad de tiempo
- Según Hennessy-Patterson debe medirse el tiempo de ejecución

Tiempo de ejecución

- El tiempo de ejecución depende de varios factores
 - en sistemas Unix es el conocido como *real time*
 - en realidad se desglosa en:
 - tiempo de CPU en modo usuario
 - tiempo de CPU en modo sistema
 - tiempo de I/O
 - por ejemplo: `time gcc -v`

```
gcc versión 3.4.5 20050821 (prerelease) (Debian 3.4.4-8)
```

```
real    0m0.597s
user    0m0.006s
sys     0m0.003s
```

Benchmarks

- Programas de prueba para la evaluación del rendimiento
- Tipos de programas
 - **aplicaciones** reales o modificadas
 - **kernels**: extracción de características concretas
 - **toy benchmarks**: programas muy pequeños: p.e. *quicksort*
 - **synthetic benchmarks**: generados de forma automática con un porcentaje determinado de cada tipo de instrucción. Los más famosos *Whetstone* y *Dhrystone*.

Otras medidas de rendimiento

- Muy usadas pero muy poco fiables
- **MIPS (GIPS, TIPS)**: millones de instrucciones por segundo
 - dependen del juego de instrucciones máquina y de los *benchmarks* utilizados
- **MFLOPS (GFLOPS, TFLOPS, PFLOPS)**: millones de operaciones en punto flotante por segundo
 - Se usan en computadores de orientación científica
 - Sólo puede ser orientativo el valor sostenido, nunca el de pico

Juegos de Benchmarks

- Colecciones de programas variados que tratan de recoger diferentes tipos de características
- Los más conocidos son los **SPEC** (www.spec.org)
 - SPEC CPU2000:
 - Aplicaciones modificadas para mejorar la portabilidad y reducir el impacto de la I/O
 - Mide el rendimiento de la CPU, la arquitectura de memoria y la calidad del compilador
 - 12 programas de aritmética entera
 - 14 programas de aritmética en punto flotante
 - SPECSFS: para servidores de archivos
 - SPECWeb: para servidores web

Juegos de Benchmarks

- TPC (www.tpc.org).
 - Miden la capacidad para procesar transacciones (accesos y actualizaciones de bases de datos)

Los informes SPEC

- Deben proporcionar toda la información que permita reproducir fielmente el test
- requiere al menos
 - Descripción completa del Hw y Sw utilizados
 - Uso de un compilador determinado
 - Dos tipos de resultados
 - *baseline performance*: restringe el tipo de optimizaciones del compilador
 - *peak performance*: optimizados al máximo

Los informes SPEC

Benchmarks	Ultra 5 Time (sec)	Opteron Time (sec)	SPECRatio	Itanium 2 Time (sec)	SPECRatio	Opteron/Itanium Times (sec)	Itanium/Opteron SPECRatios
wupwise	1600	51.5	31.06	56.1	28.53	0.92	0.92
swim	3100	125.0	24.73	70.7	43.85	1.77	1.77
mgrid	1800	98.0	18.37	65.8	27.36	1.49	1.49
applu	2100	94.0	22.34	50.9	41.25	1.85	1.85
mesa	1400	64.6	21.69	108.0	12.99	0.60	0.60
galgel	2900	86.4	33.57	40.0	72.47	2.16	2.16
art	2600	92.4	28.13	21.0	123.67	4.40	4.40
equake	1300	72.6	17.92	36.3	35.78	2.00	2.00
facerec	1900	73.6	25.80	86.9	21.86	0.85	0.85
ammp	2200	136.0	16.14	132.0	16.63	1.03	1.03
lucas	2000	88.8	22.52	107.0	18.76	0.83	0.83
fma3d	2100	120.0	17.48	131.0	16.09	0.92	0.92
sixtrack	1100	123.0	8.95	68.8	15.99	1.79	1.79
apsi	2600	150.0	17.36	231.0	11.27	0.65	0.65
Geometric mean			20.86		27.12	1.30	1.30

comparativa entre una Sun Ultra5 (máquina de referencia en SPEC2000), un AMD Opteron, y un Intel Itanium 2

Medidas de rendimiento

- ¿Cómo resumir rendimiento en una sola cifra?
- Media aritmética del tiempo de ejecución

$$T_{medio} = \frac{1}{n} \sum_{i=1}^n T_i$$

- En el caso de frecuencias (ej. MIPS) se utiliza la media armónica

$$F_{media} = \frac{n}{\sum_{i=1}^n \frac{1}{F_i}}$$

Medidas de rendimiento

- Si la frecuencia de los programas no es la misma debe ponderarse la media

$$T_{medio} = \sum_{i=1}^n P_i T_i \quad \text{donde } P_i \text{ es la frecuencia de ejecución del programa}$$

$$\text{, luego } \sum_{i=1}^n P_i = 1$$

- Para las frecuencias:

$$F_{media} = \frac{1}{\sum_{i=1}^n \frac{P_i}{F_i}}$$

Medidas de rendimiento

- Si el tiempo de ejecución de los programas es muy distinto, la media podría ser normalizada para que el tiempo se distribuya de forma equivalente entre todos los programas

$$P_i = \frac{1}{\text{Tiempo}_i \sum_{j=1}^n \frac{1}{T_j}}$$

el peso de cada programa es inversamente proporcional al tiempo de ejecución.

Cuanto mayor sea su tiempo de ejecución, menor será su peso, y por tanto su influencia sobre programas más cortos.

Medidas de rendimiento

- Otro tipo de medida es la que normaliza respecto a una máquina de referencia y aplica luego la media geométrica

$T_A = \text{tiempo sobre la máquina } A$

$T_B = \text{tiempo sobre la máquina } B$

$R_i = \frac{T_B}{T_A} = \text{tiempo normalizado respecto de } A$

$$R_{medio} = \sqrt[n]{\prod_{i=1}^n R_i} \quad \text{media geométrica}$$

Medidas de rendimiento

- Ejemplo con medias aritméticas

ver transp 16

	Máquinas			Pesos		
	(medido en segundos)			Media	ponderado	ponderado
	A	B	C	arit.	sobre B	sobre A
Programa 1	1,00	10,00	20,00	0,50	0,909	0,999
Programa 2	1000,00	100,00	20,00	0,50	0,091	0,001
Media arit.	500,50	55,00	20,00			
Media B	91,82	18,18	20,00			
Media A	2,00	10,09	20,00			

En la media aritmética el programa más largo tiene mayor influencia

La relación cambia drásticamente en función de cual sea la referencia:

- Según la Media A, la máquina C es unas 5 veces más rápida que la A
- Según la Media B, la máquina C es unas 10 veces más lenta que la A

Medidas de rendimiento

- Ejemplo con medias geométricas

	normalizado sobre A			normalizado sobre B			normalizado sobre C		
	A	B	C	A	B	C	A	B	C
P1	1,00	10,00	20,00	0,10	1,00	2,00	0,05	0,50	1
P2	1,00	0,10	0,02	10,00	1,00	0,20	50,00	5,00	1
Arit.	1,00	5,05	10,01	5,05	1,00	1,10	25,03	2,75	1
Geom.	1,00	1,00	0,63	1,00	1,00	0,63	1,58	1,58	1
Total	1,00	0,11	0,04	0,11	1,00	0,36	25,03	2,75	1

La media geométrica es independiente de la máquina utilizada como referente

Medidas de rendimiento

- Problemática de la media geométrica
 - Al ser mutiplicativa, una mejora de $x\%$ en cualquiera de los programas del benchmark $\Rightarrow$ mejorar en $x\%$ el rendimiento
 - Pero:
 - No es lo mismo mejorar un programa de 1s en 0,5s que uno de 100s en 50s
 - En ambos casos la media geométrica es idéntica
 - Difícilmente la mejora arquitectural del primer programa conducirá a reducir el tiempo de ejecución a la mitad en el segundo

Principios cuantitativos del diseño

- Sistema equilibrado
 - ¿Cuáles son los cuellos de botella (bottleneck) del sistema?
 - Mejorar un sistema pasa por eliminar los cuellos de botella.
 - La aceleración debe concentrarse en el caso típico de funcionamiento
 - Ejemplos de cuellos de botella:
 - sistema multiprocesador con único bus de memoria
 - CPU rápida con memoria lenta
 - CPU rápida con periféricos lentos

Ley de Amdahl

- ¿Cómo afecta al rendimiento la mejora de una parte de la máquina?
- Definiciones:
 - **F**: fracción de tiempo del sistema original que va a ser mejorada
 - **G_p**: Ganancia en velocidad de la parte mejorada
 - **T_o**: tiempo de ejecución del sistema original
 - **T_m**: tiempo de ejecución mejorado

Ley de Amdahl

$$T_m = T_o(1 - F) + \frac{F * T_o}{G_p}$$

si $G_g = \frac{T_o}{T_m}$ entonces $G_g = \frac{1}{(1 - F) + \frac{F}{G_p}}$

a mayor Frecuencia de utilización
de la mejora mayor ganancia

El límite de la ganancia es $1/1-F$

Rendimiento de la CPU

- El rendimiento de la CPU es inversamente proporcional al tiempo de CPU $R_{CPU} = \frac{1}{T_{CPU}}$

$$T_{CPU} = Ciclos * T_{ciclo} \quad \text{tiempo de ejecución en la CPU}$$

$$CPI = \frac{N^{\circ} Ciclos}{Intruc. Ejecutadas} \quad \text{ciclos por instrucción}$$

$$T_{CPU} = CPI * IE * T_{ciclo}$$

y como $Freq = \frac{1}{T_{ciclo}}$

$$T_{CPU} = \frac{CPI * IE}{Freq}$$

Rendimiento de la CPU

$$T_{CPU} = \frac{\text{Intruc.}}{\text{Programa}} * \frac{N^{\circ} \text{ Ciclos}}{\text{Instruc.}} * \frac{\text{Segundos}}{\text{Ciclo}}$$

	Instruc.	CPI	T _{ciclo}
compilador	+		
ISA	+	+	
organización		+	+
tecnología			+

Rendimiento de la CPU

- No todas las instrucciones requieren el mismo número de ciclos

$$CPI = \frac{N^{\circ} \text{ Ciclos}}{\text{Intruc. Ejecutadas}} \quad \text{luego}$$

$$N^{\circ} \text{ Ciclos} = CPI * \text{Intruc. Ejecutadas}$$

y si distinguimos entre diferentes tipos de instrucciones

$$N^{\circ} \text{ Ciclos} = \sum_{i=1}^n (N^{\circ} \text{ Ciclos})_i = \sum_{i=1}^n CPI_i * IE_i \quad y$$

$$CPI = \sum_{i=1}^n CPI_i * \left(\frac{IE_i}{\text{Instrucciones Ejecutadas}} \right)$$

Consumo de potencia

- A medida que aumenta la complejidad de los procesadores, también lo hace el consumo
 - Los primeros procesadores consumían décimas de vatio
 - Un PIV a 3.2GHz consume 135W
- El consumo presenta dos problemas:
 - mayor calor => necesidad de disiparlo
 - puntos calientes (hotspots) que reducen la vida del chip
 - menor vida de las baterías (en disp. móviles)

Coste de un computador

- Factores que influyen en el coste:
 - *yield* : % de componentes fabricados que funcionan
 - volumen
 - competencia
- El yield mejora con el paso del tiempo sin necesidad de mejoras tecnológicas (learning curve)
 - ej. el precio por MB de DRAM se reduce un 40% x año
 - en los microprocesadores el impacto no es tan directo
- Doblar el volumen reduce el coste un 10%
 - Los gastos fijos se amortizan entre un mayor número de unidades

Coste de un computador

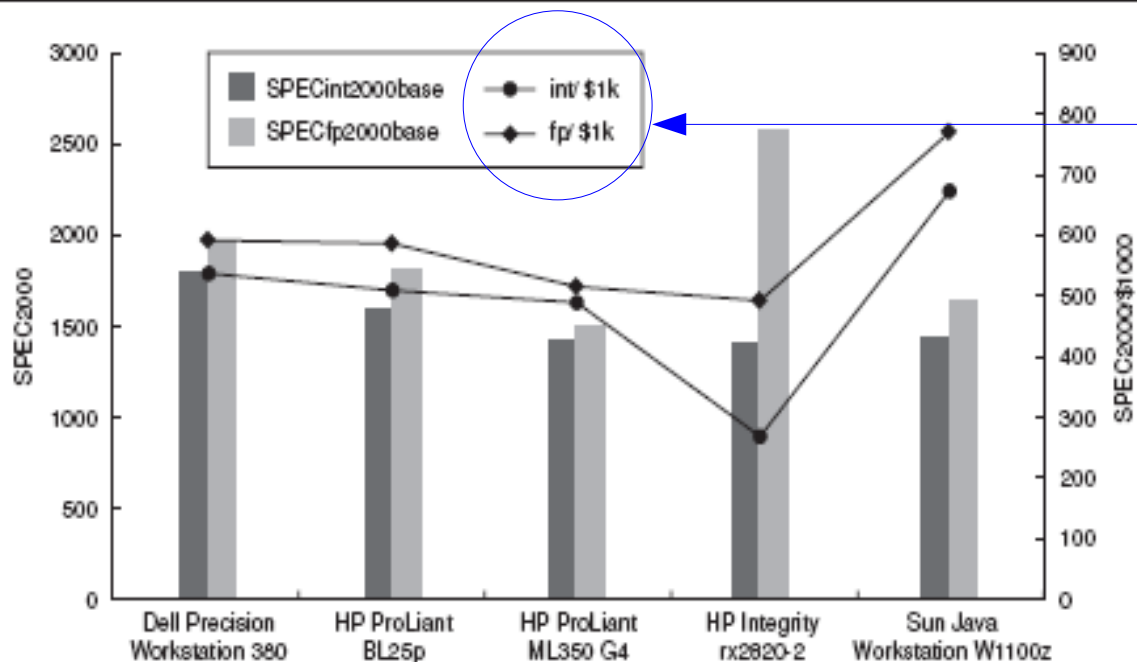
- Desde el punto de vista técnico importa la relación coste/rendimiento
- Desde el punto de vista comercial la relación a considerar es de precio/rendimiento
- Coste $\neq$ Precio
 - R&D, marketing, ventas, mantenimiento de equipos, impuestos, margen de venta, ...
- el coste también puede influir en el volumen de ventas
- actualmente en los PCs medios el coste en R&D oscila entre el 4% y 12% (incluyendo la ingeniería)

Coste de un computador

- Rendimiento vs. precio

Vendor/model	Processor	Clock rate	L2 cache	Type	Price
Dell Precision Workstation 380	Intel Pentium 4 Xeon	3.8 GHz	2 MB	Desk	\$3346
HP ProLiant BL25p	AMD Opteron 252	2.6 GHz	1 MB	Rack	\$3099
HP ProLiant ML350 G4	Intel Pentium 4 Xeon	3.4 GHz	1 MB	Desk	\$2907
HP Integrity rx2620-2	Itanium 2	1.6 GHz	3 MB	Rack	\$5201
Sun Java Workstation W1100z	AMD Opteron 150	2.4 GHz	1 MB	Desk	\$2145

precios de Agosto 2005



rendimiento por cada \$1000