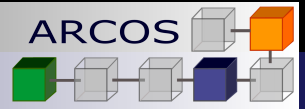
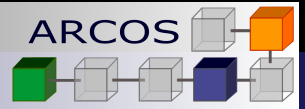


Tendencias y Evaluación

Arquitectura de Computadores

J. Daniel García Sánchez (coordinador)
David Expósito Singh
Javier García Blas
J. Manuel Pérez Lobato

Grupo ARCOS
Departamento de Informática
Universidad Carlos III de Madrid



Impacto de la tecnología

- Los cambios tecnológicos tienen impacto en los mecanismos de implementación de la ISA.

- **Tecnologías**:
  - Lógica de circuitos integrados.
  - DRAM.
  - Flash.
  - Discos magnéticos.
  - Redes.

Tendencias

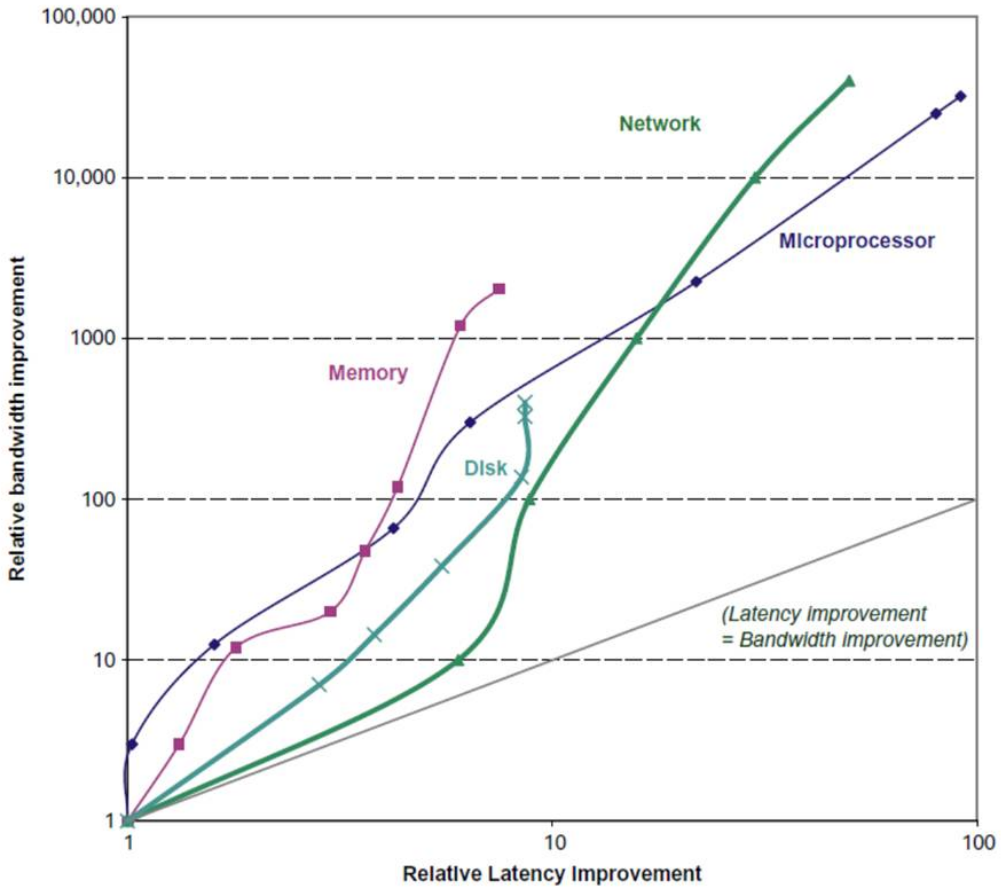
- **Tecnologías de circuitos integrados**.
  - Densidad de transistores: 35% anual.
  - Tamaño del dado: 10%-20% anual.
  - Efecto combinado: 40%-55% anual (Ley de Moore).
    - **¡Dejó de cumplirse!**.
- **Capacidad DRAM**.
  - 25%-40% anual (reduciéndose).
- **Capacidad Flash**.
  - 50%-60% anual.
  - 8-10 veces más barato por bit que DRAM.
- **Capacidad de discos magnéticos**.
  - 5% anual (últimos años).
  - 8-10 veces más barato por bit que Flash.
  - 200-300 veces más barato que DRAM.

Ancho de banda y latencia

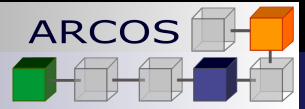
- Ancho de banda o tasa de procesamiento (*throughput*).
  - Cantidad de trabajo realizado por unidad de tiempo.
  - **Procesadores**: Incremento entre 32.000 y 40.000 veces.
  - **Memoria y discos**: Incremento entre 300 y 1.200 veces.

- Latencia y tiempo de respuesta.
  - Tiempo entre inicio y fin de un evento.
  - **Procesadores**: Incremento entre 50 y 90 veces.
  - **Memorias y discos**: Incremento entre 6 y 8 veces.

Ancho de banda frente a latencia



Fuente: Computer Architecture: A Quantitative Approach. 6 Ed Hennessy and Patterson. Morgan Kaufmann. 2017.



Tendencias en potencia y energía

- Se dispone de dos sistemas (**A** y **B**).
  - **A** consume un 20% más de potencia eléctrica que **B**.
  - **A** ejecuta una tarea en el 70% de tiempo que **B**.
  - ¿Cuál tiene menor coste?

- La métrica adecuada para la comparación es la **Energía**.
  - $E(B) = P(B) \cdot t(B)$
  - $E(A) = 1.2 \cdot P(B) \cdot 0.7 \cdot t(B) = 0.84 \cdot E(B)$
  - El sistema **A** consume el 84% de la energía de **B**.

Energía y potencia en micros

- En tecnología CMOS, el consumo de energía se deriva de la conmutación de transistores.
- **Energía dinámica**:
  - Cantidad de energía necesaria para conmutar.
    - $0 \rightarrow 1$ o $1 \rightarrow 0$.
    - $E_d \approx \frac{1}{2} \cdot X_c \cdot V^2$
- **Potencia dinámica**:
  - Depende de frecuencia de conmutación.
    - $P_d \approx \frac{1}{2} \cdot X_c \cdot V^2 \cdot f$

Nota

X_c: Carga capacitiva V: Voltaje f: Frecuencia

Ejemplo

- Si una reducción de voltaje del 15% implica una reducción de frecuencia del 15%:
  - ¿Qué efecto hay sobre la potencia dinámica?

Solución

$$\frac{P_{nueva}}{P_{ant}} = \frac{(V \cdot 0.85)^2 \cdot (f \cdot 0.85)}{V^2 \cdot f} = 0.85^3 = 0.61$$

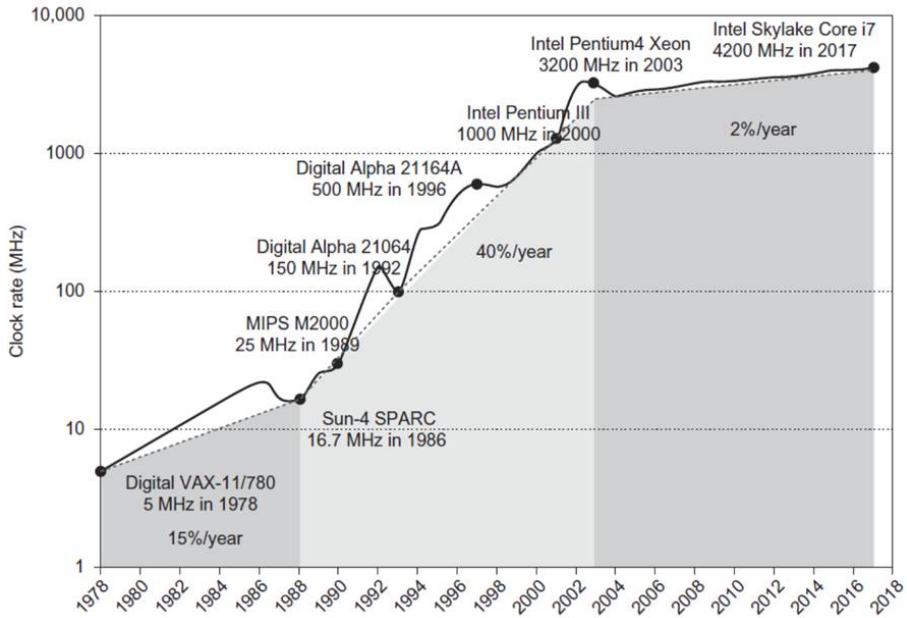
Consecuencias

- **Reducción**:
  - La potencia y energía dinámica se reducen al bajar el voltaje.
    - En 20 años el voltaje ha bajado de 5V a 1V.
  - La carga capacitiva depende de número de transistores conectados a una salida.
    - Mecanismo de control de potencia y energía.

Evolución

- Evolución dominada por incremento de número de transistores e incremento de frecuencia.
  - Incremento de potencia y energía.

- Intel 80386 $\rightarrow$ 2 W
- Intel Core i9-10900K 5.3GHz $\rightarrow$ 95 W.
  - Chip: 1.5 $\times$ 1.5 cm.
  - Límite de enfriamiento por ventilación.



Fuente: Computer Architecture: A Quantitative Approach. 6 Ed Hennessy and Patterson. Morgan Kaufmann. 2017.

Eficiencia energética

- **Técnicas**:
  - Desactivación de reloj de unidades inactivas.
  - Escalado dinámico de voltaje y frecuencia (DVFS).
  - Modos de bajo consumo en memoria y discos.
    - Requiere mecanismo para reactivar.
  - Overclocking automático.
    - Se activa si es seguro.
    - Ej. Core i7 3.3 GHz puede ejecutar ráfagas a 3.6 GHz.

Coste

- El coste de fabricación de un computador se reduce a lo largo del tiempo.
  - Principio de la curva de aprendizaje.
    - Medido por el rendimiento del proceso de fabricación (Porcentaje de dispositivos que sobreviven a la fabricación).
    - Si se dobla el rendimiento se divide a la mitad el coste.
    - **DRAM**: Promedio de caída anual del 40% en coste y precio (Excepto periodos de escasez o superávit).
  - Volumen:
    - Decremento del 10% en coste si se dobla volumen.
    - Reducción de amortización de desarrollo por unidad.
    - Incremento de eficiencia del proceso de fabricación.
  - Venta por múltiples fabricantes de mismo producto.
    - Mayor competencia.

Coste de circuito integrado

- Proceso de fabricación.
  - Oblea $\rightarrow$ Dados.

Coste

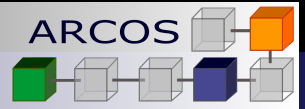
$$Coste_{IC} = \frac{Coste_{dado} + Coste_{pruebas} + Coste_{empaquetado}}{Rendimiento}$$

$$Coste_{dado} = \frac{Coste_{oblea}}{Dados_{oblea} \times Rendimiento}$$

$$Dados_{oblea} = \frac{\pi \times (\frac{diametro}{2})^2}{area} - \frac{\pi \times diametro}{\sqrt{2 \times area}}$$

Ejemplo

- Oblea de 30 cm. de diámetro.
  - Dados de 1.5 cm.
    - **Dados por oblea**: 270.
  - Dados de 1 cm.
    - **Dados por oblea**: 640.



Velocidad de ejecución

- ¿Qué significa que el computador **A** es más rápido que el computador **B**?
  - Desktop.
    - Mi programa se ejecuta en menos tiempo.
    - Quiero reducir el tiempo de ejecución.
  - Administrador de sitio Web.
    - Puedo procesar más transacciones por hora.
    - Quiero aumentar la tasa de procesamiento.

Rendimiento y tiempo de ejecución

- El rendimiento $R(x)$ es una métrica inversa al tiempo de ejecución $T(x)$.

Rendimiento

$$R(x) = \frac{1}{T(x)}$$

- Alto rendimiento $\rightarrow$ Bajo tiempo de ejecución.

- X se ejecuta n veces más rápido que Y.

Aceleración

$$n = \frac{T(x)}{T(y)} = \frac{\frac{1}{R(x)}}{\frac{1}{R(y)}} = \frac{R(y)}{R(x)}$$

Métricas

- La única métrica fiable para comparar el rendimiento de computadores es la ejecución de **programas reales**.
  - Cualquier otra métrica conduce a errores.
  - Cualquier alternativa a programas reales conduce a errores.

- **Tiempo de ejecución**.
  - **Tiempo de respuesta**: Tiempo total transcurrido.
  - **Tiempo de CPU**: Tiempo que la CPU ha estado ocupada.

Carga de trabajo

- El **rendimiento** de un computador depende de la **carga de trabajo** con la que se evalúa.

- Computadores adaptados a cargas específicas:
  - Servidores web.
  - Servidores de bases de datos.
  - Servidores de ficheros.
  - Computadores personales.
  - Multiprocesadores.
  - Multicomputadores.
  - ...

Benchmarks

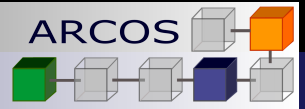
- Aplicación o conjunto de aplicaciones usadas para evaluar el rendimiento.
- **Aproximaciones**:
  - **Kernels**: Partes pequeñas de aplicaciones reales.
    - *Ejemplo*: FFT.
  - **Programas de juguete**: Programas cortos.
    - *Ejemplo*: Quicksort.
  - **Benchmarks sintéticos**: Inventados para representar aplicaciones reales.
    - *Ejemplo*: Dhrystone.
- Todas malas aproximaciones:
  - ¡El arquitecto y el compilador pueden engañar!

Benchmarks

- **Empotrados**:
  - Dhrystone (relevancia discutible).
  - EEMBC (kernels).
- **Desktop**:
  - SPEC2017 (mezcla de programas enteros y coma flotante).
- **Servidores**:
  - SPECWeb, SPECSFS, SPECjbb, SPECvirt_Sc2010.
  - TPC

Ejemplo: SPEC2017

- **SPECrate/SPECspeed 2017 Integer**: Aritmética entera.
  - 20 programas: C (10), C++ (8), Fortran (2).
  - Diversas áreas de aplicación:
    - Lenguajes y compiladores, planficiación de rutas, simulación de eventos, compresión de video, inteligencia artificial, ...
- **SPECrate 2017, SPECspeed 2017 Floating point**: Coma flotante
  - 23 programas. Fortran (6), C (6), Fortran/C (5) C++ (2), C++/C (2), C++/C/Fortran (2).
  - Diversas áreas de aplicación:
    - Física, Química, Biología, Álgebra, *Rendering* de imágenes, ...



Ley de Amdahl

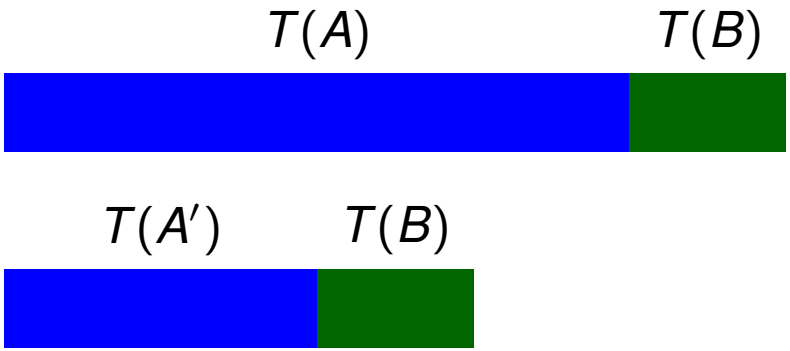
- El incremento de rendimiento obtenido usando un modo de ejecución más rápido está limitado por la fracción de tiempo que se puede usar dicho modo.

- *Speedup* o aceleración:
  - Ratio entre el rendimiento mejorado y el rendimiento original.

$$S = \frac{R(M)}{R(O)}$$

$$S = \frac{T(O)}{T(M)}$$

Tiempo de ejecución



$$F = \frac{T(A)}{T(A) + T(B)}$$

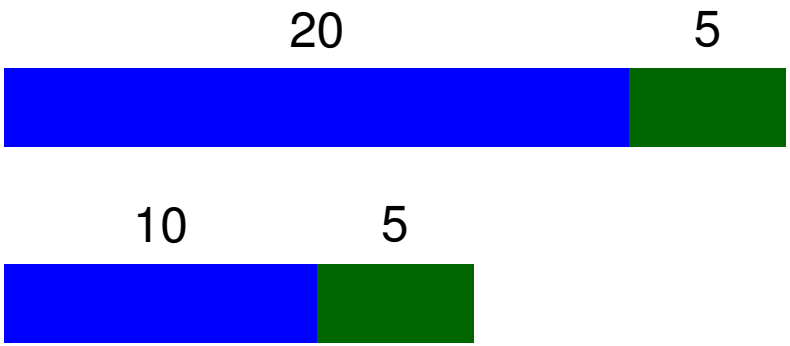
$$S(m) = \frac{T(A)}{T(A')}$$

$$T' = T(A') + T(B) = \frac{T(A)}{S(m)} + (1 - F) \times T$$

$$T' = \frac{F \times T}{S(m)} + (1 - F) \times T$$

$$T' = T \times \left[(1 - F) + \frac{F}{S(m)}\right]$$

Ejemplo



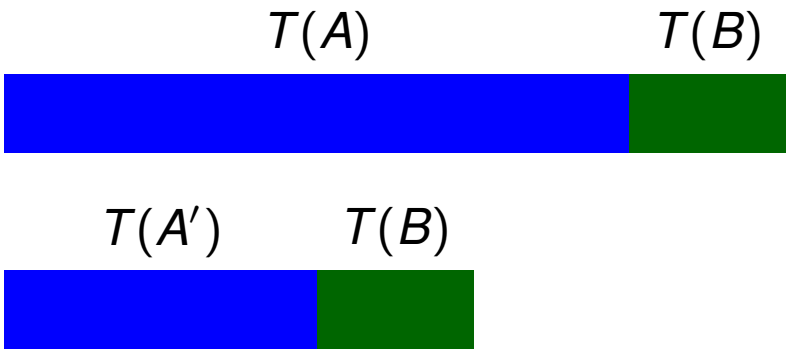
$$F = \frac{20}{20+5} = 0.8$$

$$S(m) = \frac{20}{10} = 2$$

$$T' = T \times \left[(1-F) + \frac{F}{S(m)}\right] = 25 \times \left[(1-0.8) + \frac{0.8}{2}\right] = 15$$

- ¡Esto ya lo sabíamos!

Ley de Amdahl



$$T' = T \times \left[(1 - F) + \frac{F}{S(m)}\right]$$

$$S = \frac{T}{T'} = \frac{T}{T \times \left[(1 - F) + \frac{F}{S(m)}\right]} = \frac{1}{(1 - F) + \frac{F}{S(m)}}$$

- El ***speedup*** depende **exclusivamente** de la **fracción de mejora** y el **speedup de la mejora**.

Caso 1

- Un servidor Web distribuye su tiempo en:
  - **Cómputo**: 40
  - **E/S**: 60
- Si se sustituye por otra máquina que puede hacer el cómputo 10 veces más rápido, ¿Cuál es el *speedup* global?

Solución

$$S = \frac{1}{0.6 + \frac{0.4}{10}} = \frac{1}{0.64} = 1.5625$$

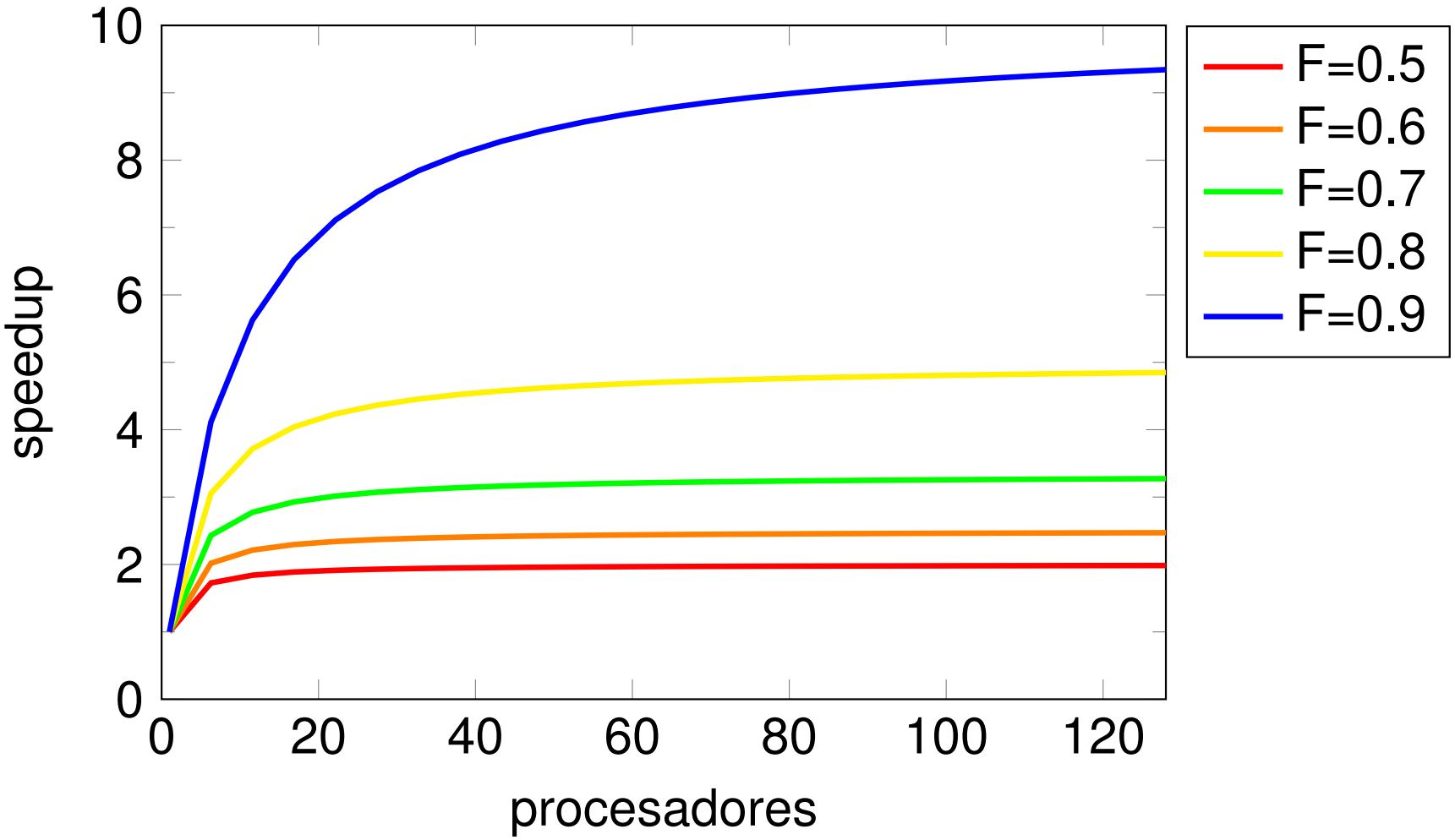
Caso 2

- Una aplicación tiene una parte paralelizable que consume el 50% del tiempo.
  - Si se asume que se puede paralelizar esta parte completamente con 32 procesadores, ¿cuál será el máximo speedup?

Solución

$$S = \frac{1}{0.5 + \frac{0.5}{32}} = \frac{1}{0.515625} = 1.9393$$

Evolución de speedup



Reflexiones sobre Ley de Amdahl

- Una mejora es más efectiva cuanto más grande es la fracción de tiempo en que ésta se aplica
- Para mejorar un sistema complejo hay que optimizar los elementos que se utilicen durante la mayor parte del tiempo (caso más común).
- Campos de aplicación de las optimizaciones:
  - **Dentro del procesador**: la ruta de datos (data path)
  - **En el juego de instrucciones**: la ejecución de las instrucciones más frecuentes
  - **En el diseño de la jerarquía de memoria, la programación y la compilación**: hay que explotar la localidad de las referencias.
    - El 90% del tiempo se está ejecutando el 10% del código.

4 Evaluación del rendimiento

- Métricas de rendimiento
- Benchmarks
- Ley de Amdahl
- Rendimiento del procesador

Tiempo de ejecución

- Un procesador ejecuta cada instrucción en varios ciclos de reloj.

Tiempo consumido por CPU

$$tiempo_{CPU} = \frac{ciclos_{CPU}}{\text{frecuencia de reloj}}$$

CPI: Ciclos por instrucción

- Se puede expresar la velocidad media como ciclos por instrucción (CPI) a partir de:
  - El número total de ciclos consumidos y,
  - el número de instrucciones ejecutadas (IC).

CPI

$$CPI = \frac{ciclos_{CPU}}{IC}$$

Factores en tiempo de ejecución

CPI y tiempo de CPU

$$CPI = \frac{ciclos_{CPU}}{IC}$$

$$tiempo_{CPU} = \frac{ciclos_{CPU}}{f} = \frac{CPI \times IC}{f} = CPI \times IC \times T$$

- Si se reducen un 10% cualquiera de los 3 factores se reduce un 10% el tiempo de ejecución.
  - Pero los 3 factores están interrelacionados.

Clases de instrucciones

- Distintas clases de instrucciones tienen distinto IC y CPI.

CPI global

$$ciclos_{CPU} = \sum_{i=1}^{n} IC_i \times CPI$$

$$tiempo_{CPU} = (\sum_{i=1}^{n} IC_i \times CPI_i) \times T$$

$$CPI_{global} = \frac{\sum_{i=1}^{n} IC_i \times CPI_i}{IC} = \sum_{i=1}^{n} \frac{IC_i}{IC} \times CPI_i$$

Impacto de la frecuencia relativa de instrucciones en ejecución de programa.

Ejemplo

- En la ejecución de un programa se ha visto que:
  - **Operaciones coma flotante**: 25% (4.0 CPI en promedio).
  - **Operación FPSQR** (*raíz cuadrada*): 2% (20 CPI).
    - Incluida en coma flotante.
  - **Resto de instrucciones**: 1.33 CPI.

- Elegir entre alternativas de diseño:
  a. Reducir CPI de FPSQR a 2.
  b. Reducir CPI de todas las operaciones de coma flotante a 2.5.

Solución

$$CPI = 0.25 \times 4 + 1.33 \times 0.75 = \mathbf{1.9975}$$

$$0.25 \times CPI_{FP} = 0.23 \times CPI_{otrasFP} + 0.02 \times CPI_{FPSQR}$$
$$0.25 \times 4 = 0.23 \times CPI_{otrasFP} + 0.02 \times 20$$
$$CPI_{otrasFP} = \frac{0.24 \times 4 - 0.02 \times 20}{0.23} = 2.6087$$

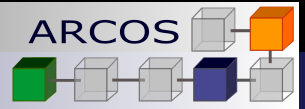
$$CPI_{nuevoFPSQR} = 0.23 \times 2.6087 + 0.02 \times 2 + 0.75 \times 1.33 = \mathbf{1.6375}$$
$$CPI_{nuevoFP} = 0.25 \times 2.5 + 0.75 \times 1.33 = \mathbf{1.6225}$$

Resumen

- El ancho de banda ha mejorado mucho más que la latencia en los últimos 20 años.
- El crecimiento de potencia consumida limita la frecuencia de reloj.
- Reducción del coste de fabricación a lo largo del tiempo.
- La única métrica fiable para comparar el rendimiento de computadores es la ejecución de programas reales.
- La ley de Amdahl establece un límite sobre la mejora del rendimiento con múltiples aplicaciones.
- La frecuencia relativa de las instrucciones tiene impacto en la velocidad de ejecución de programas.

Referencias

- **Computer Architecture. A Quantitative Approach**
  5th Ed.
  Hennessy and Patterson.
  Secciones 1.4 a 1.9.



Tendencias y Evaluación

Arquitectura de Computadores

J. Daniel García Sánchez (coordinador)
David Expósito Singh
Javier García Blas
J. Manuel Pérez Lobato

Grupo ARCOS
Departamento de Informática
Universidad Carlos III de Madrid