



## ARQUITECTURA DE COMPUTADORES II

### AUTORES:

David Expósito Singh

Florin Isaila

Daniel Higuero Alonso-Mardones

Javier García Blas

Borja Bergua Guerra

*Área de Arquitectura y Tecnología de Computadores*

*Departamento de Informática*

*Universidad Carlos III de Madrid*

Julio de 2012

## TEMA 1:

# ***EJECUCIÓN PARALELA: FUNDAMENTOS(I)***

*“A parallel computer is a collection of processing elements that communicate and cooperate to solve large problems fast.”*

*Almasi y Gottlieb, 1989.*

*“A parallel computer is a collection of processing elements that communicate and cooperate to solve large problems fast.”*

*Almasi y Gottlieb, 1989.*

## Concurrencia:

- En computación **distribuida**, *la concurrencia es parte del problema.*
- En computación **paralela**, *la concurrencia es parte de la solución.*

# Índice

1. **¿Por qué se demanda ejecución paralela?**
  - **Demanda de cómputo**
  - **Logros tecnológicos**
  - **Logros en la arquitectura**
  - **Aspectos económicos**
2. Arquitecturas paralelas: clasificación
  - Procesadores vectoriales
  - Multiprocesadores
  - *Clusters* y Cluster Computing
  - Computación GRID
5. Beneficios del paralelismo
6. Paralelización: principios
  - Ley de Amdahl
  - *Overhead* del paralelismo
  - Proximidad y equilibrado
  - Tamaño del ‘grano’
  - Depuración de pp. paralelos
7. Casos reales:
  - Google (Hw)
  - Top500

## A) Demanda de cómputo

- Tradición en Ciencia y en Ingeniería:
  - Hacer teoría o diseño sobre el papel.
  - Realizar experimentos o construir el sistema:
    - difícil
    - lento
    - caro
    - ¡peligroso!
- Enfoque que permite la Informática:
  - Modelar los fenómenos físicos y los sistemas que se experimentan de manera fidedigna a través de métodos numéricos eficientes.

## A) Demanda de cómputo

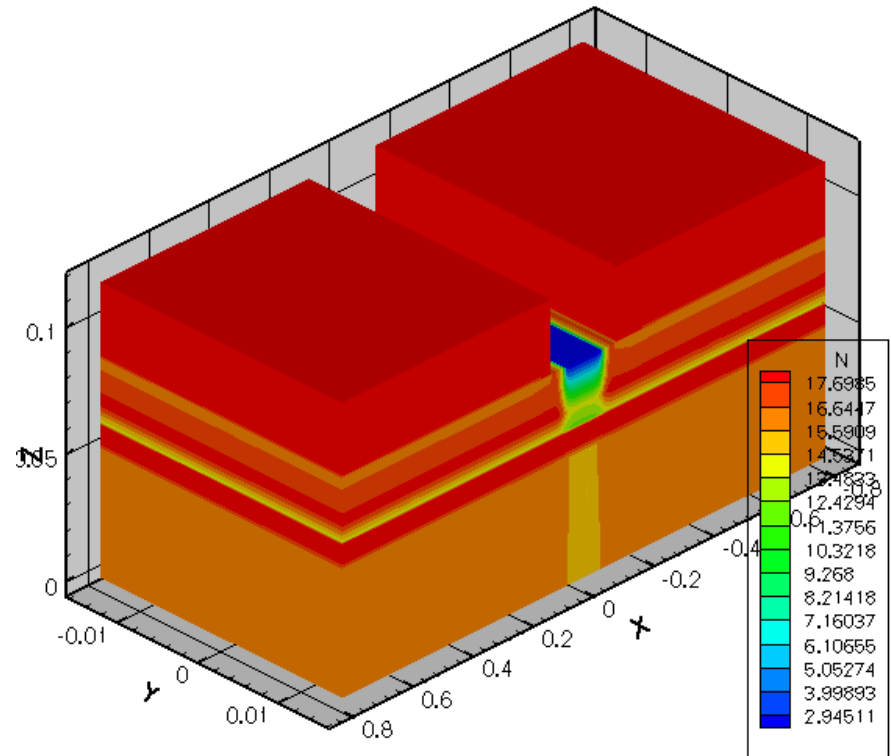
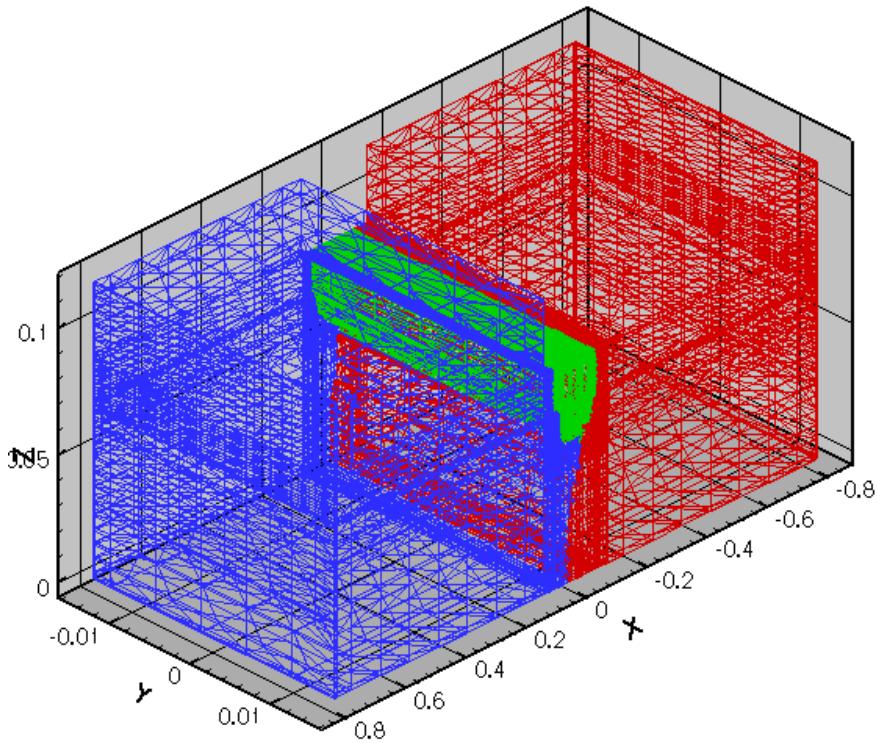
*Necesidad de capacidad de **almacenamiento** y de **potencia de cálculo** suficientes*

# Procesamiento de voz y de imágenes

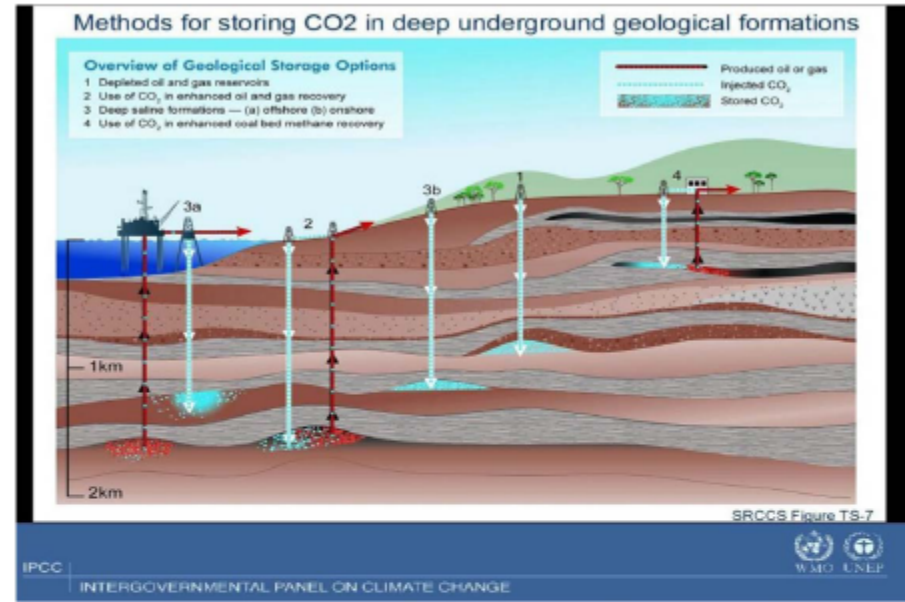
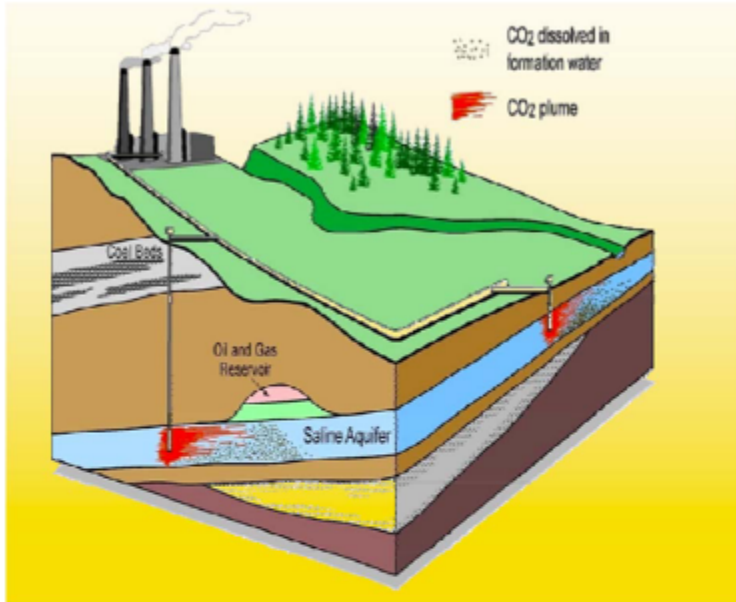




# Simulación dispositivos estado sólido



## CO<sub>2</sub> Sequestration - Storage Options - Model

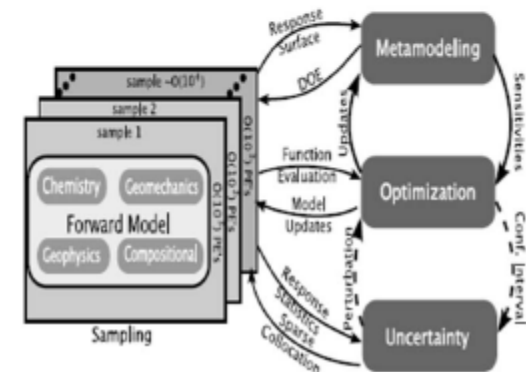


Computational complexity of a 3-D, 3-phase flow and transport in porous media problem on a moderately large subsurface model:

- $N = 10^7$  Discretization elements – 3 unknowns/grid element
- $N_t = 1000$  timesteps
- $N_{MCS} = 1000$  Monte Carlo simulation for uncertainty assessment

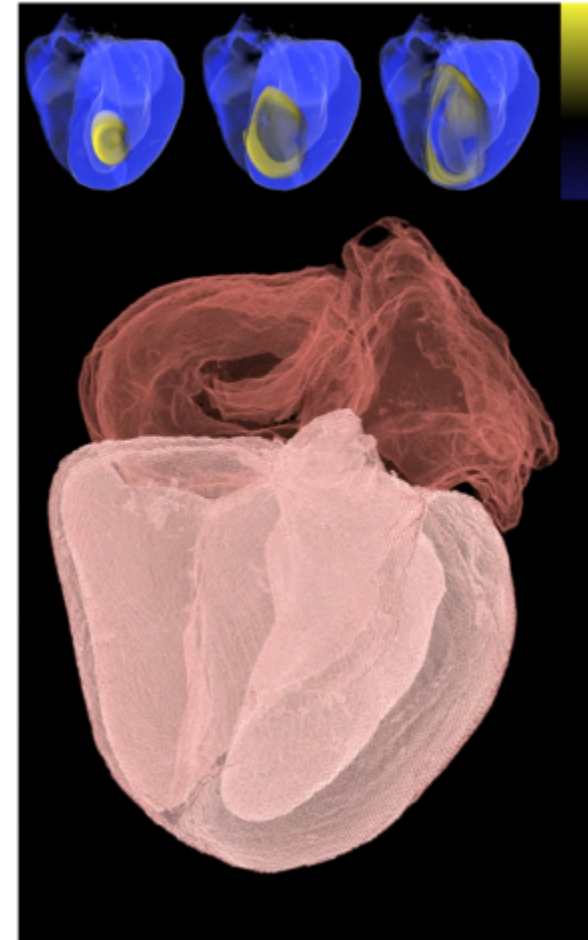
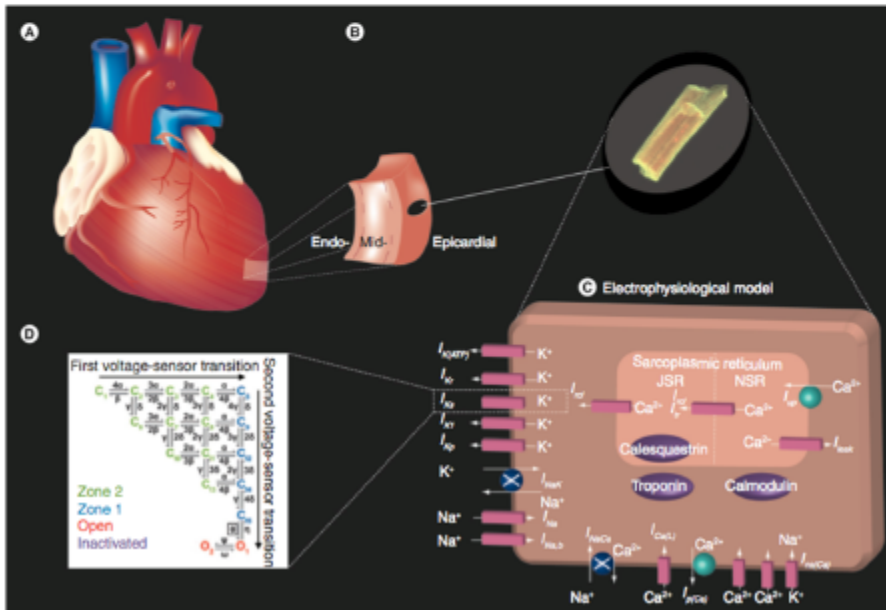
$$\Rightarrow \text{FLOPS} = N_t \cdot N_{MCS} (3N)^2 \sim 10^{21} \text{ (1000 ExaFlops)}$$

Source: M.F. Wheeler, et. al., NSF PetaApps Proposal 2008

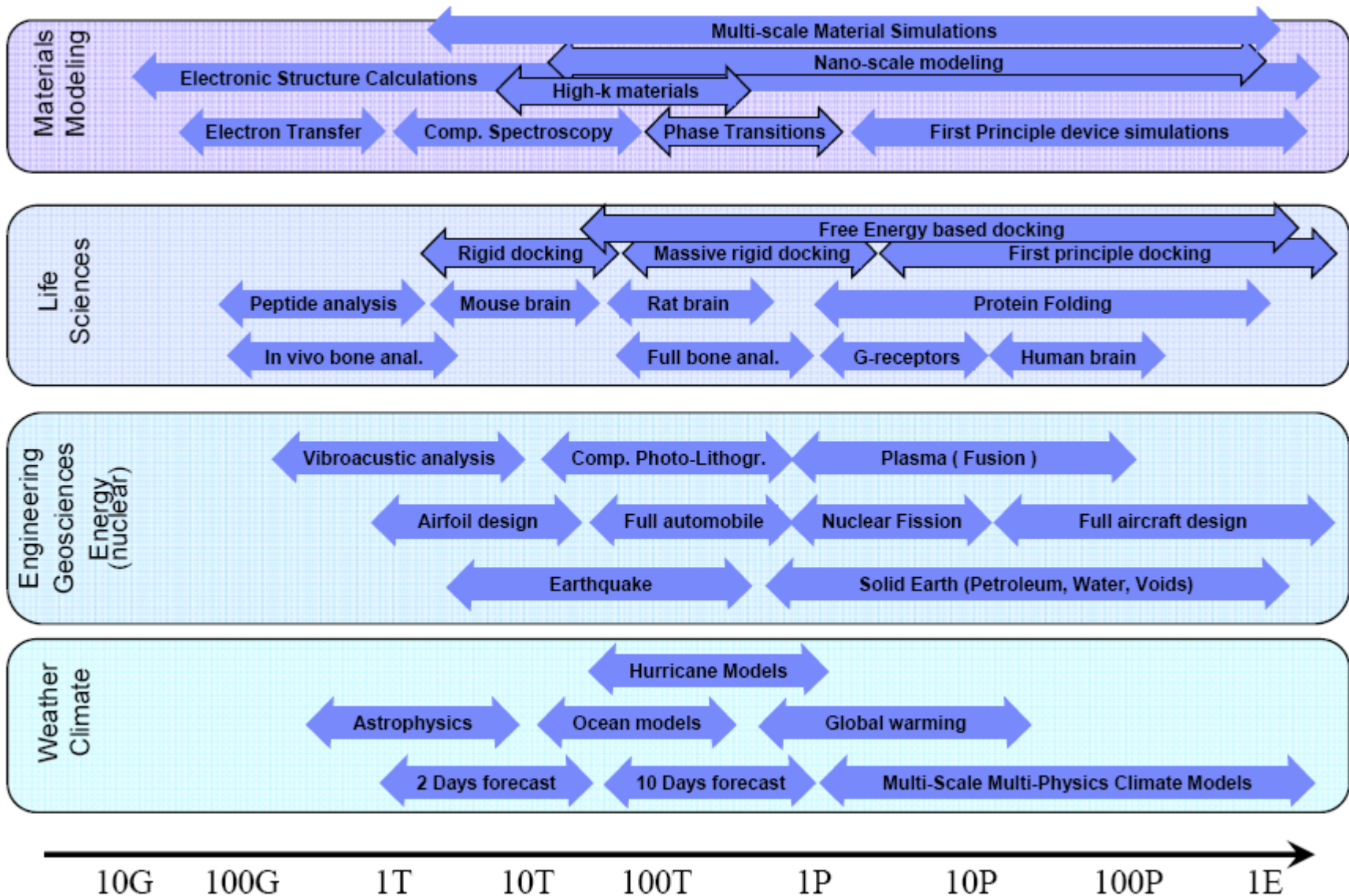


# Computational Medicine: Whole Organ Simulation

- **Predictive Toxicology**
- **Multiscale Model of Organs**
  - from protein function through to cell function through to tissue function through to macroscale organ modeling.
- **Multiple model components and scales require Petascale to Exascale compute capability**
  - Usefulness requires “turnkey” modeling environment where many variations and scenarios can be attempted by the medical or pharmaceutical researcher quickly and accurately
  - Further increases the computational requirements



Fuente: IBM Research



Fuente: IBM Research



## Ejemplos

- Algunos ejemplos:
  - Modelización y predicción **meteorológica** y **climática**.
  - Test de seguridad en **automoción**
  - Modelización y procesamiento en **astrofísica** (IAC).
  - Modelización molecular en la industria **farmacéutica**.
  - Diseño de estructuras en **arquitectura** e **ingeniería**.
  - **Genoma** humano.
  - Modelización en **finanzas** y **economía**.
  - Diseño y validación de **semiconductores**.
  - “**Filogénesis**”: estudio de la historia de las especies.
  - Búsqueda en el **Web**.
  - Modelización en **medicina**: p.ej., experimentación de nuevas válvulas cardiacas.
  - Procesamiento de **transacciones**.
  - Procesamiento de **voz** y de **imágenes**.

## Límite de cómputo de una CPU

- ¿puede una sola CPU satisfacer esos niveles de cómputo?

Ejemplo:

- *UltraSparc* 166 MHz (1995).
  - 330 MFLOPS (máximo)
  - 300 ns fallo de cache
- Resultados reales sobre un determinado programa patrón de multiplicación de matrices:
  - ~ 4,5 MFLOPS

## Límite de cómputo de una CPU

- Intel Montecito.
  - ▣ **45 GFLOPs**
  - ▣ Dos núcleos
  - ▣ 2-way coarse-grained [multithreading](#) por núcleo (no simultáneo).
  - ▣ 16 KB Instruction L1 y 16 KB Data L1 cache por núcleo.
  - ▣ 1 MB Instruction L2 y 256 KB Data L2 cache por núcleo.
  - ▣ 12 MB L3 cache por núcleo (24 MB L3 por chip)
  - ▣  $1.72 * 10^9$  transistores por chip:
    - core logic — 57M, or 28.5M per core
    - core caches — 106.5M
    - 24 MB L3 cache — 1550M
    - bus logic and I/O — 6.7M
  - ▣ Tamaño chip 27.72 [mm](#) x 21.5 mm, 596 mm<sup>2</sup>
  - ▣ Consumo de 75-104 [W](#).

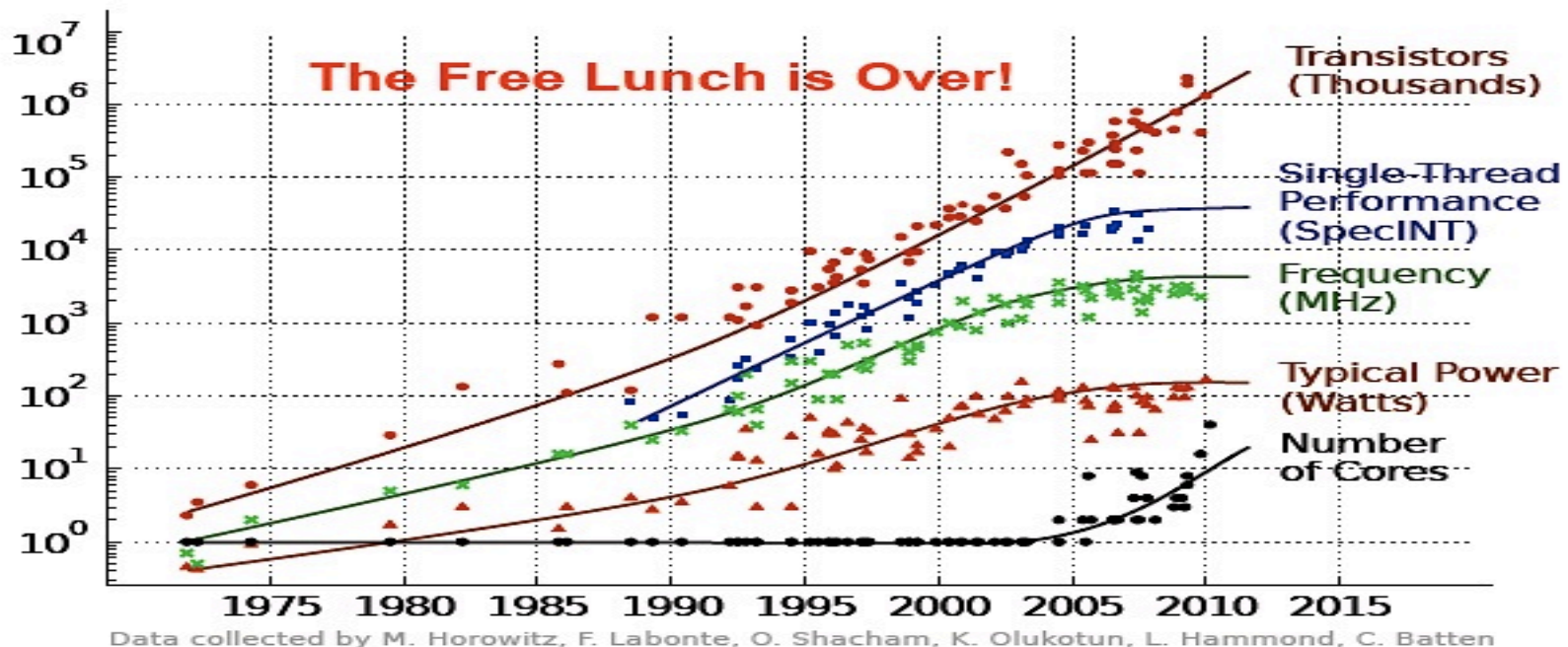
## B) Logros tecnológicos

■ Hasta dónde puede seguir aumentando la capacidad de ejecución de las CPUs?: ¿Limitaciones tecnológicas?: ¿sí?

■ Número de transistores:  $\uparrow$  40% al año

■ Velocidad de reloj:  $\uparrow$  30% al año

[Culler99]



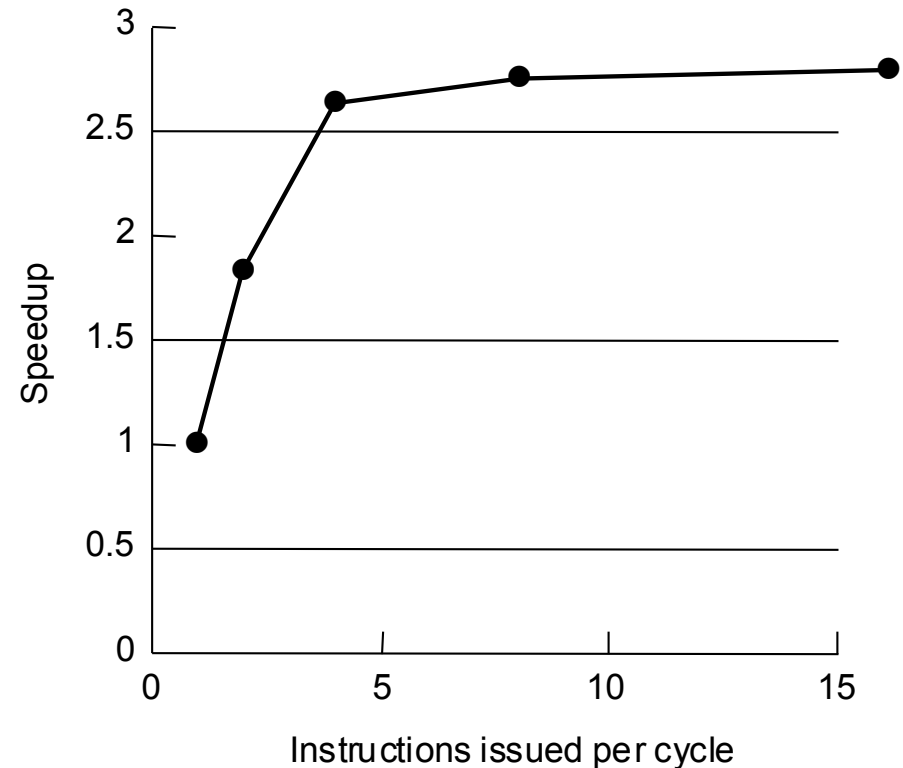
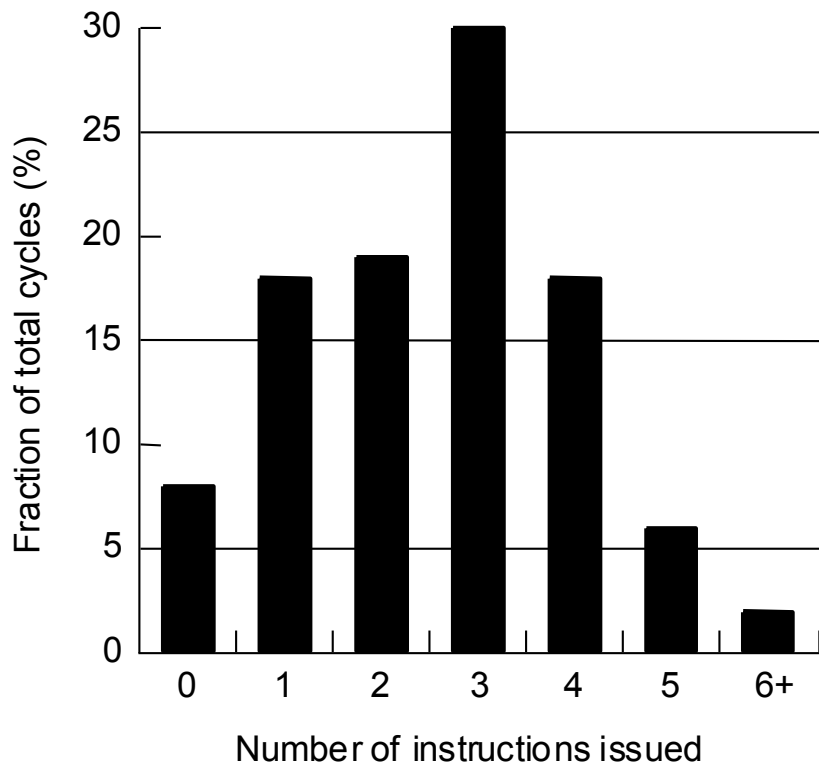


## C) Logros en la arquitectura

- Evolución de los microprocesadores: ligado al desarrollo VLSI: aumento del nivel de paralelismo interno
  - Hasta **1985**: paralelismo entre **bits** (*bit level parallelism*): 4 bits → 8 bits → 16 bits → 32 bits
    - los 32 bits representan un punto de inflexión: inclusión de cache en el chip
    - 64, 128 bits
  - Mediados de los **80s** hasta la **actualidad**: paralelismo entre las **instrucciones** (*instruction level parallelism, ILP*):
    - (*pipelining* + RISC) + mejoras en los compiladores
    - varias unidades funcionales + caches ⇔ ejecución superescalar
    - sofisticación en el control: ejecución fuera de orden, especulativa, predicción de saltos.
  - Paralelismo entre *threads*: *Hyper-Threading technologies*.
  - Paralelismo entre **múltiples unidades de procesamiento**: *multi-core architectures*.

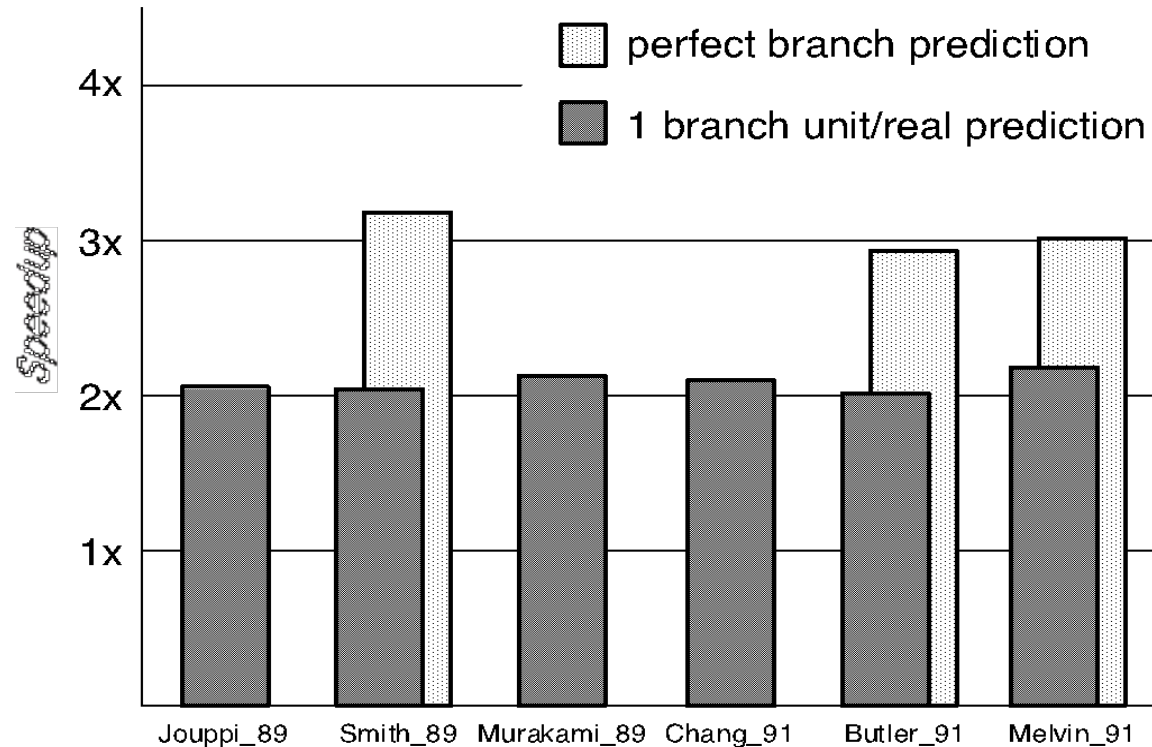
## ¿Límite del *ILP*?

- ¿Hasta qué punto es *ILP* la solución?: ¿está limitado el *speedup*? (I)
  - Resumen resultados experimentales: n° de recursos y ancho de banda de instrucciones infinitos + predicción de salto y renombrado de registros.



## ¿Límite del *ILP*?

- Resumen de resultados para superescalares de 4 “vías” (4-*issue*):
  - Difícil superar un *speedup* de 2.
  - Es necesario el paralelismo entre *threads*.



## D) Aspectos económicos

- Los microprocesadores no sólo son rápidos, sino que también son *baratos*
  - Coste de desarrollo (5-100 millones de dólares) incluso mayor que el de los **supercomputadores**, *pero* se venden muchos más.
  - Las familias de microprocesadores facilitan la construcción de máquinas basadas en una arquitectura.
  - Las arquitecturas añaden facilidades para la construcción de Multiprocesadores (MP).
  
- Los constructores de computadores (COMPAQ, DEC, IBM, SGI, Sun, etc.) fabrican versiones MP de sus máquinas con un sobreprecio muy pequeño
  
- Cray fabricó también MP (T3D, T3E)

## Tendencias actuales

- ILP valioso pero limitado.
- Desarrollo de lenguajes de programación.
- Desarrollo de paralelizadores automáticos.
- Hardware para realizar la cooperación.