CMCP – Conceptos y Métodos de la Computación Paralela

Master en Computación en la Nube y de Altas Prestaciones

T1. Introducción a la Computación Paralela

J. E. Roman

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València

Curso 2024-2025

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Contenido

Apartado 1

Introducción

Computación Paralela

¿Qué es?

- Se realiza en un computador paralelo
- Permite ejecutar varias operaciones de forma simultánea para la resolución de un único problema

¿Qué no es?

- Hilos de ejecución sin una estrecha relación entre sí
- Procesos poco acoplados o dispersos geográficamente (sistemas distribuidos, cloud, grid)

Motivación

- Microprocesadores cada vez con más núcleos
- A menudo la computación secuencial no es suficiente
  - Problemas de gran dimensión
  - Restricciones de tiempo real

Simulación Numérica

Simulación: emular un sistema físico por computador

En ingeniería:

- Prototipado virtual
- Reducción de costes y ciclo del producto

En ciencia:

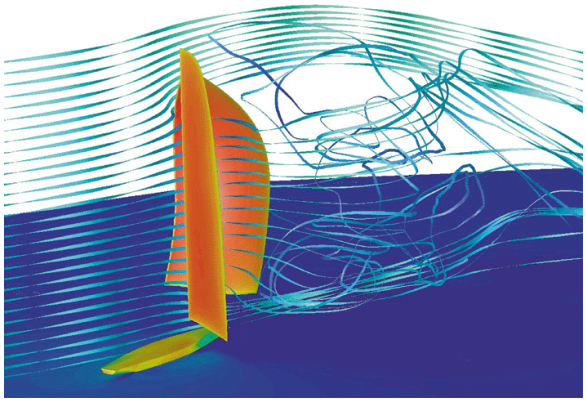
- Fundamental para el avance científico
- Sistemas complejos: geometría compleja, modelos multi-física, no linealidad

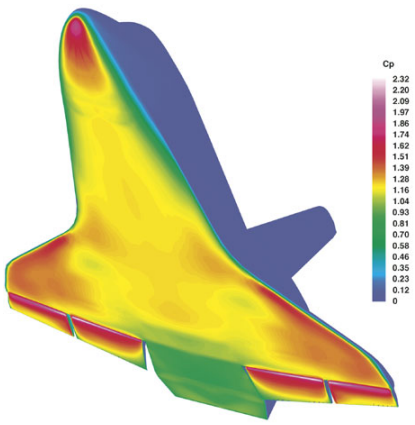
Casi siempre, la simulación exige gran capacidad de cómputo

Simulación: Aplicaciones

Áreas de aplicación:

- Meteorología
- Mecánica de estructuras
- Dinámica de fluidos
- Ciencia de materiales
- Electromagnetismo
- Acústica
- Astrofísica
- Bioingeniería



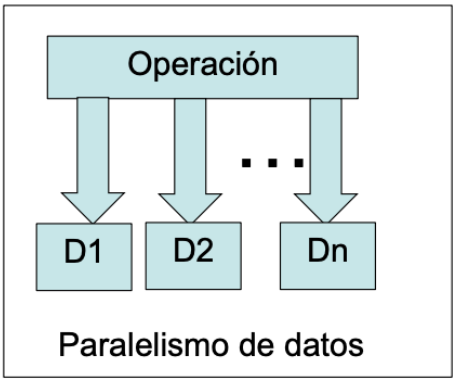


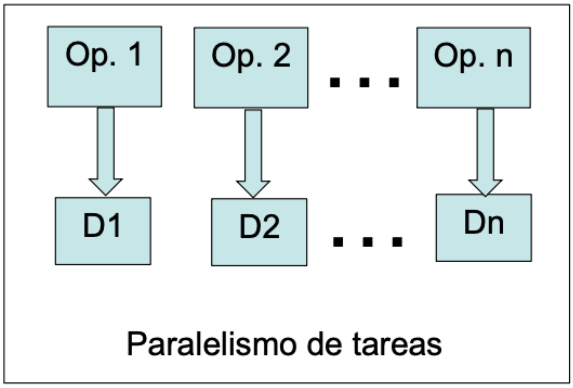
¿Qué es el Paralelismo?

Ejecución simultánea de diferentes partes de un proceso

- Procesos paralelos (o paralelizables): son susceptibles de división en partes independientes
- Procesos secuenciales: no se pueden dividir en partes independientes, existen dependencias intrínsecas que obligan a ejecutar una parte después de otra

Formas básicas de paralelismo lógico:





Elementos de la Computación Paralela

Para poder aprovechar el paralelismo de un problema necesitamos:

- Computadores paralelos
- Software de base: sistema operativo, compiladores, etc.
- Entornos de programación paralela: MPI, OpenMP, CUDA, Parallel Matlab Toolbox, ...
- Herramientas de análisis de prestaciones
- Herramientas software reutilizables (librerías) para facilitar el diseño de aplicaciones

Tendencias en Diseño de Computadores Paralelos

- Eficiencia: capaces de resolver problemas de forma rápida y de beneficiarse de los avances tecnológicos
- Economía: se tiende a huir de los diseños hardware específicos (*custom chips*) para utilizar componentes disponibles en el mercado (*commodity chips*)
- Portabilidad: la rápida evolución de los computadores paralelos ha obligado a definir estándares que permitan no tener que reescribir el software al cambiar de sistema
- Heterogeneidad: especialmente en los últimos años con el uso de GPUs
- Consumo reducido: se pretende conseguir igual rendimiento con menos consumo (operaciones por vatio)

Limitaciones Físicas

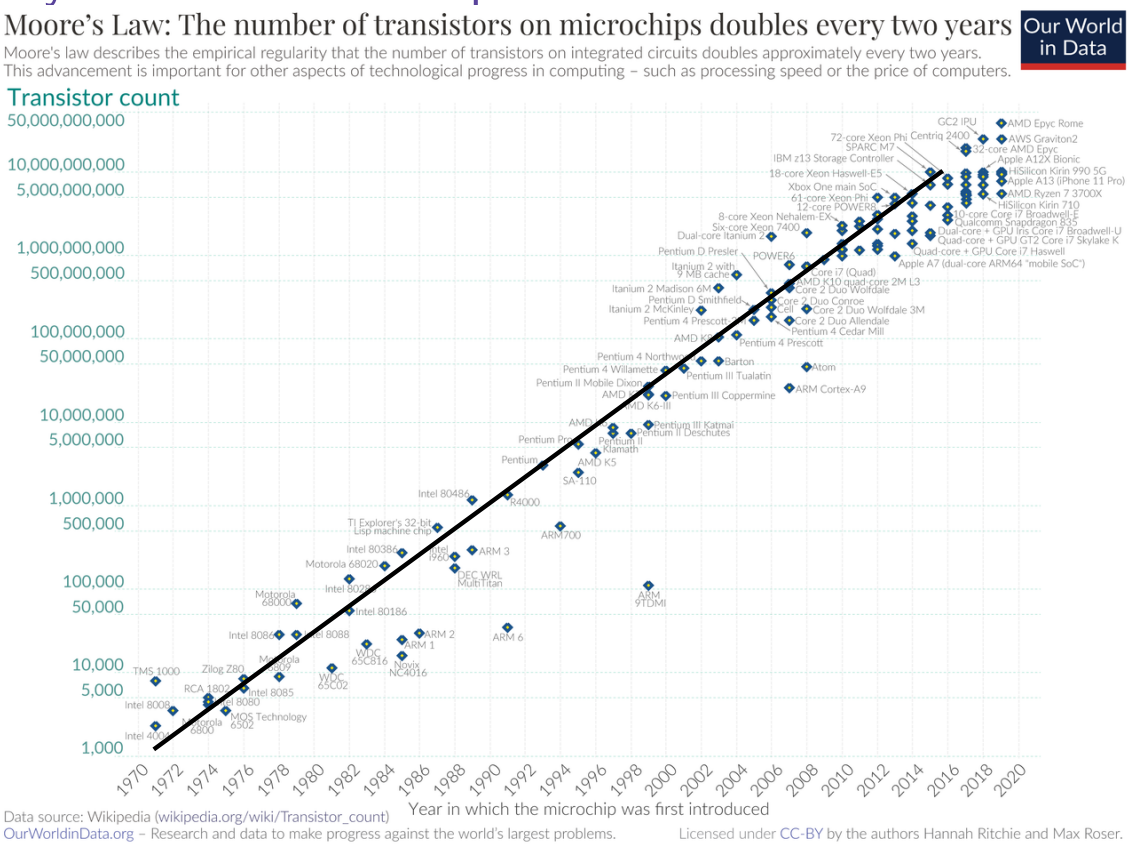
Limitaciones físicas en el diseño de chips:

- Límite de la velocidad de la luz: 30 cm/ns
- Límite en la escala de integración: actualmente cerca del máximo
- Límite de diseño plano: $\mathcal{O}(n^2)$

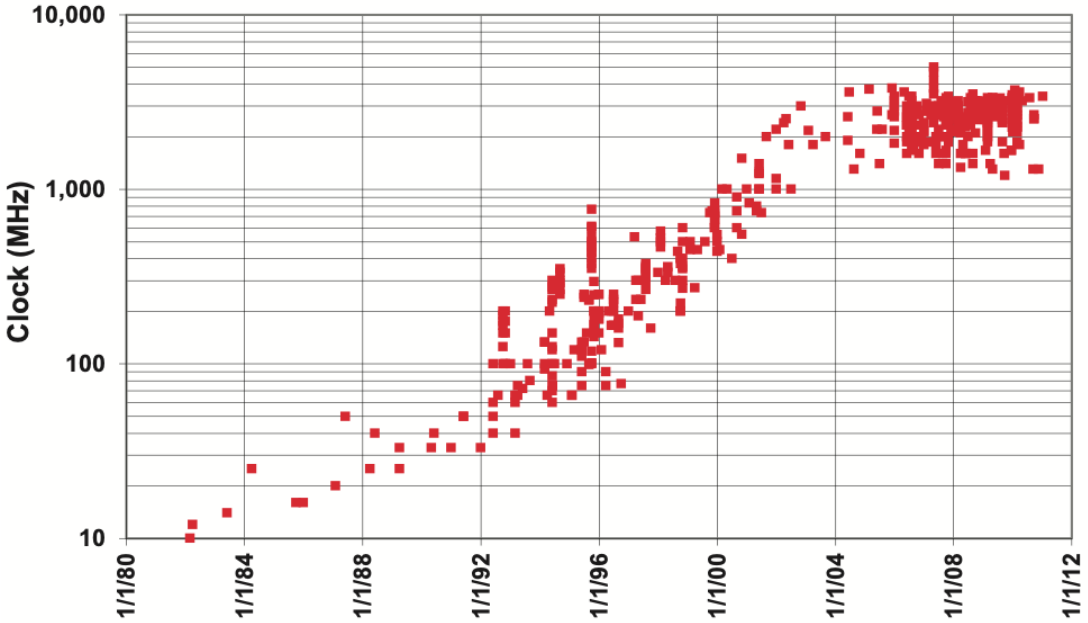
Consecuencias:

- Estancamiento en los incrementos de la frecuencia de reloj
- Más frecuencia $\rightarrow$ más consumo + temperatura elevada
- Miniaturización $\rightarrow$ problemas de interferencia en circuitos próximos y problemas de disipación de calor

La Ley de Moore se Cumple



Frecuencia de Reloj se Estanca



Fuente: Bill Gropp (`www.cs.illinois.edu/~wgropp`)

Explicación: Escalado de Dennard

Dennard (1974): voltaje y corriente deben ser proporcionales a la dimensión lineal del transistor

$$\text{Potencia} = \alpha C F V^2$$

α=fracción de tiempo activo, C=capacidad, F=frecuencia, V=voltaje

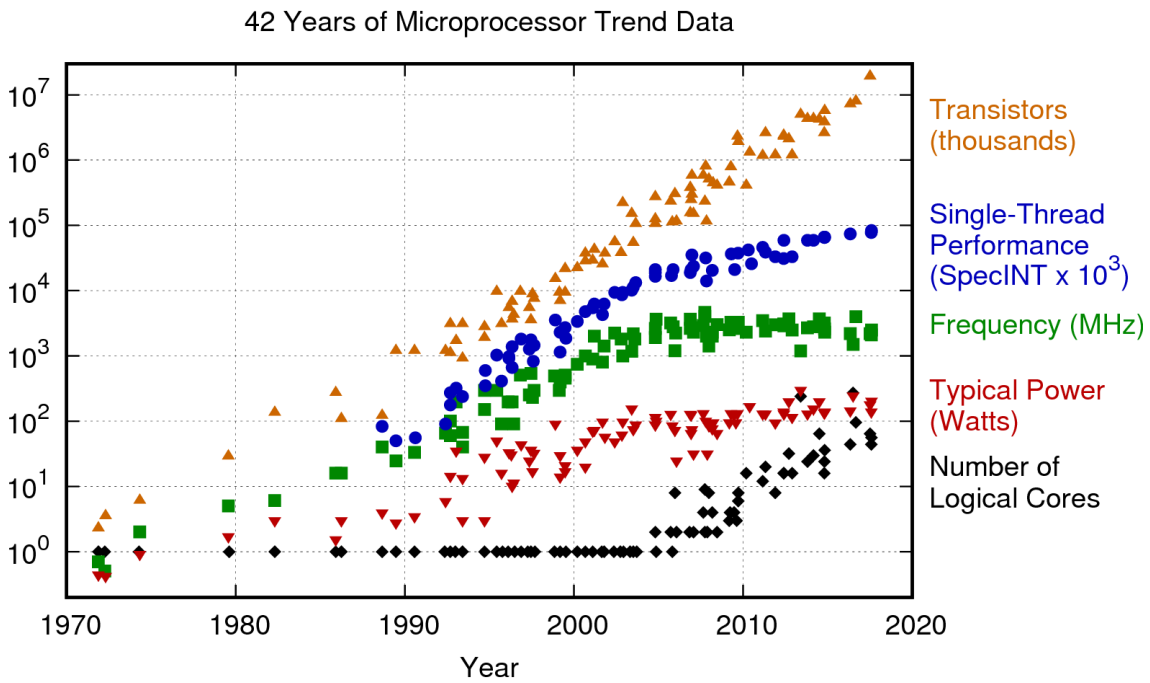
- La capacidad C está relacionada con el área $\rightarrow$ la potencia es proporcional al área
- A menor tamaño de transistor, los circuitos pueden operar a mayor frecuencia con la misma potencia

---

Fin del escalado de Dennard

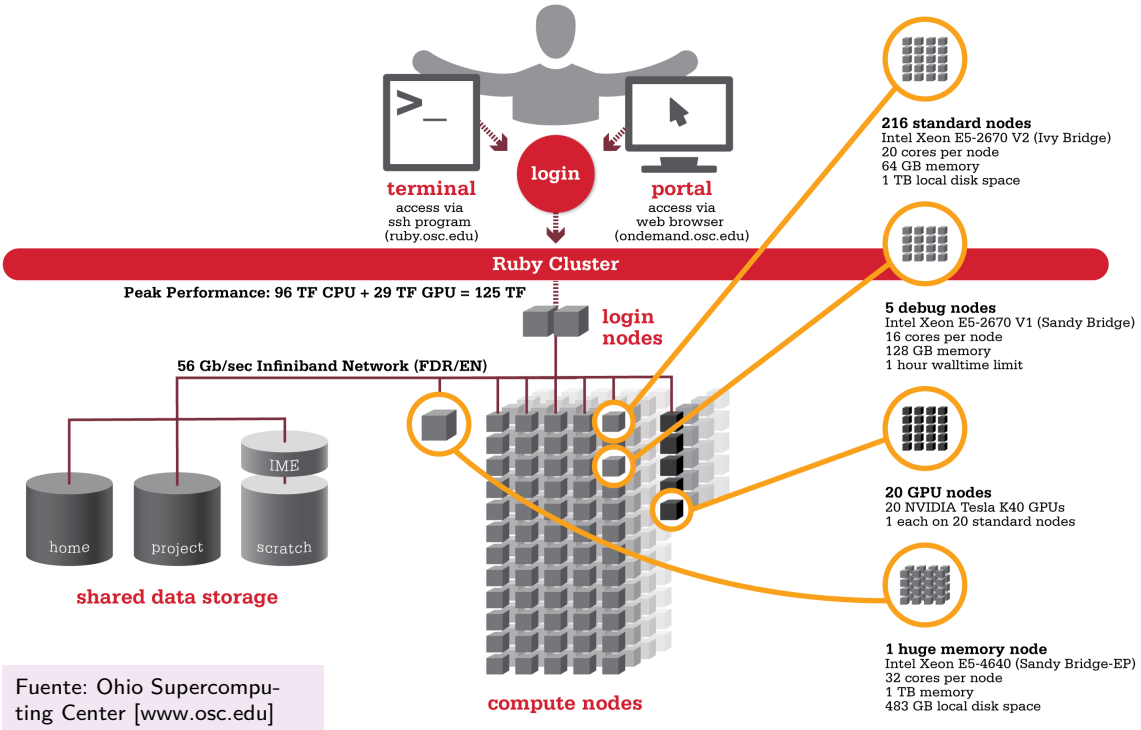
- No tiene en cuenta las pérdidas ni el voltaje umbral, que implica una potencia mínima por transistor
- Esto ha creado la "Power Wall" que limita la frecuencia del procesador a unos 4GHz

Tendencia en Microprocesadores



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Arquitectura de Computadores



Fuente: Ohio Supercomputing Center [www.osc.edu]

TOP500

En la lista TOP500 aparecen los 500 computadores más potentes del mundo
https://www.top500.org

- Se actualiza 2 veces al año
- Los computadores se clasifican según su potencia sostenida medida en operaciones en coma flotante por segundo (Flop/s)

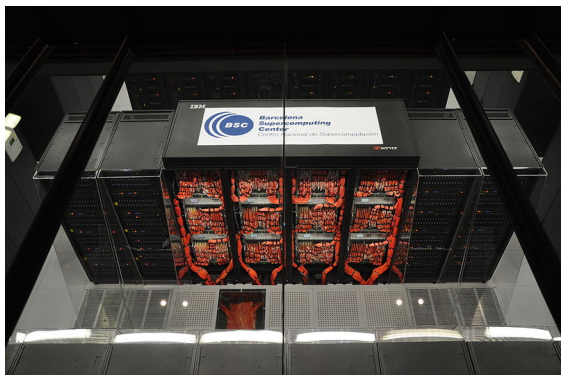
#	Centro	Equipo	Fab.	Cores	RMax
1	ORNL (EEUU)	Frontier - AMD EPYC 64C 2GHz + AMD Instinct MI250X, Slingshot-11	HPE	8.730.112	1102
2	RIKEN (Japón)	Fugaku - A64FX 48C 2.2GHz, Tofu interconnect D	Fujitsu	7.630.848	442
3	CSC (Finlandia)	LUMI - AMD EPYC 64C 2GHz + AMD Instinct MI250X, Slingshot-11	HPE	1.110.144	152
4	CINECA (Italia)	Leonardo - BullSequana, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100, Infiniband	Atos	1.824.768	239
5	ORNL (EEUU)	Summit - Power AC922, Power9 22C 3.07 GHz + Volta GV100, Infiniband	IBM	2.414.592	149

RMax en PFlop/s

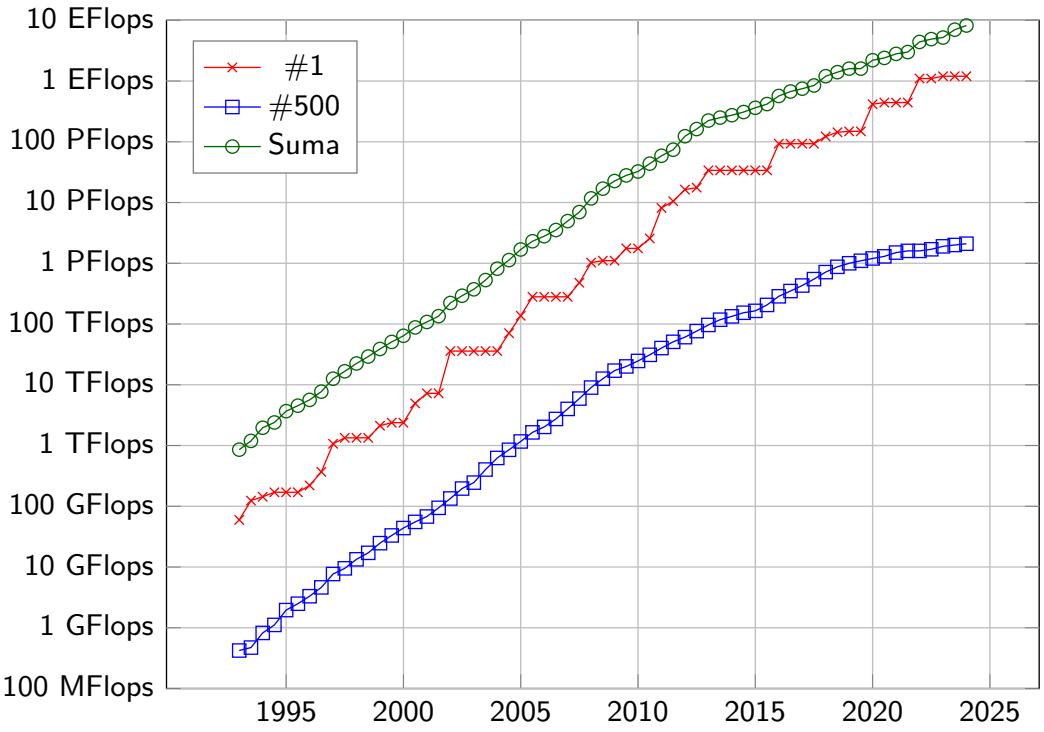
Red Española de Supercomputación (RES)

MareNostrum 5 tiene actualmente dos particiones:

- GPP: 6408 nodos 2x Intel Shappire Rapids 8480+ 56c
- ACC: 1120 nodos 2x Intel Shappire Rapids 8460Y+ 40c, 4x NVIDIA Hopper 64GB HBM
- Potencia de pico: 305.5 PFlop/s
- Red Infiniband NDR200
- 2.6 PB memoria principal, 652 PB disco



Top500: Evolución de las Prestaciones



Fuente: `https://www.top500.org`

Taxonomía de Flynn

SISD	SIMD
Single Instruction, Single Data	Single Instruction, Multiple Data
MISD	**MIMD**
Multiple Instruction, Single Data	Multiple Instruction, Multiple Data

- **SISD** Computador secuencial
- **SIMD** Computadores vectoriales (NEC, Fujitsu, Cray), procesadores con extensiones vectoriales (SSE3, Altivec, AVX-512)
- **MIMD** Multiprocesadores, clusters, multi-core

Apartado 2

Paralelismo Implícito

Paralelismo dentro del Procesador

Incluso en procesadores con un solo núcleo hay paralelismo

- Unidades funcionales segmentadas
- Unidades funcionales replicadas

Hay diversas arquitecturas que emplean estas técnicas

- Procesadores supersegmentados y superescalares
- Procesadores VLIW (*very long instruction word*)
- Capacidades multi-hilo: *hyper-threading*
- Procesadores vectoriales
  - Juego de instrucciones con operaciones vectoriales
  - Registros vectoriales (por ejemplo 16 `float`)

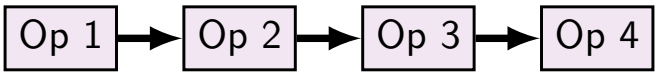
Este tipo de paralelismo es transparente al programador

- Propia lógica de ejecución de instrucciones
- Papel del compilador (opciones de optimización)

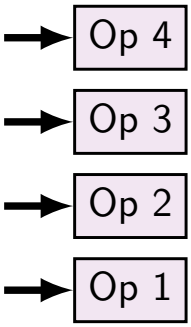
Incremento de Prestaciones en Monoprocesadores

Formas básicas de paralelismo físico:

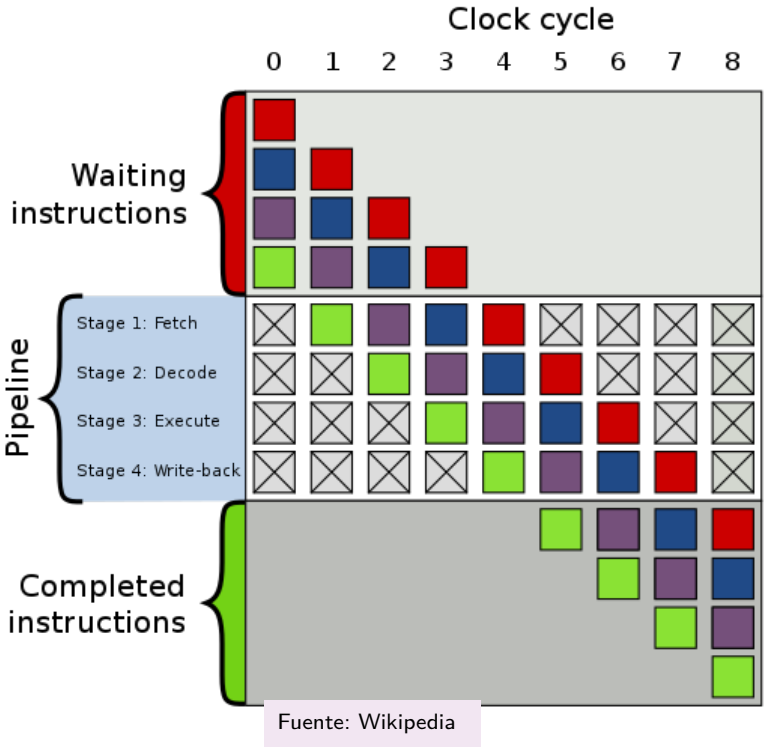
- Segmentación (*pipelining*)



- Réplica



Segmentación de la Ejecución de Instrucciones

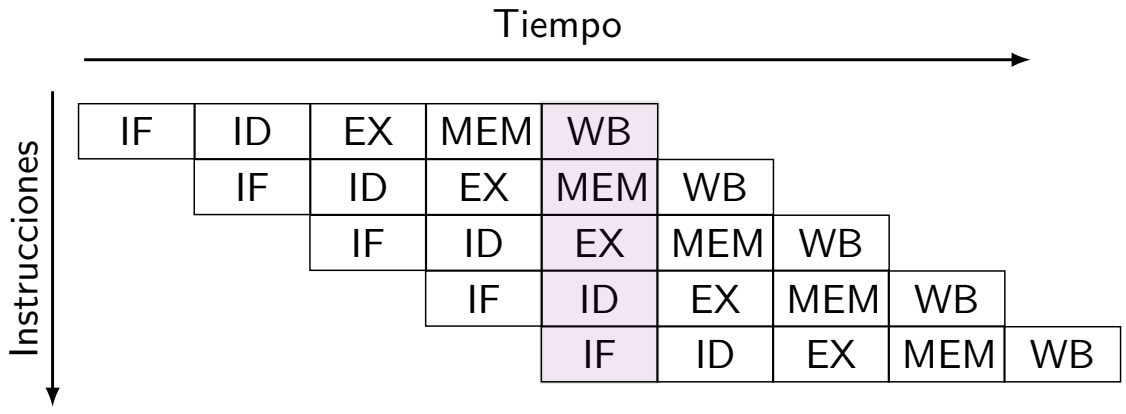


Fuente: Wikipedia

Objetivo: tener ocupadas todas las etapas del pipeline

Se forman huecos en caso de dependencias o latencia de memoria

Segmentación Clásica RISC

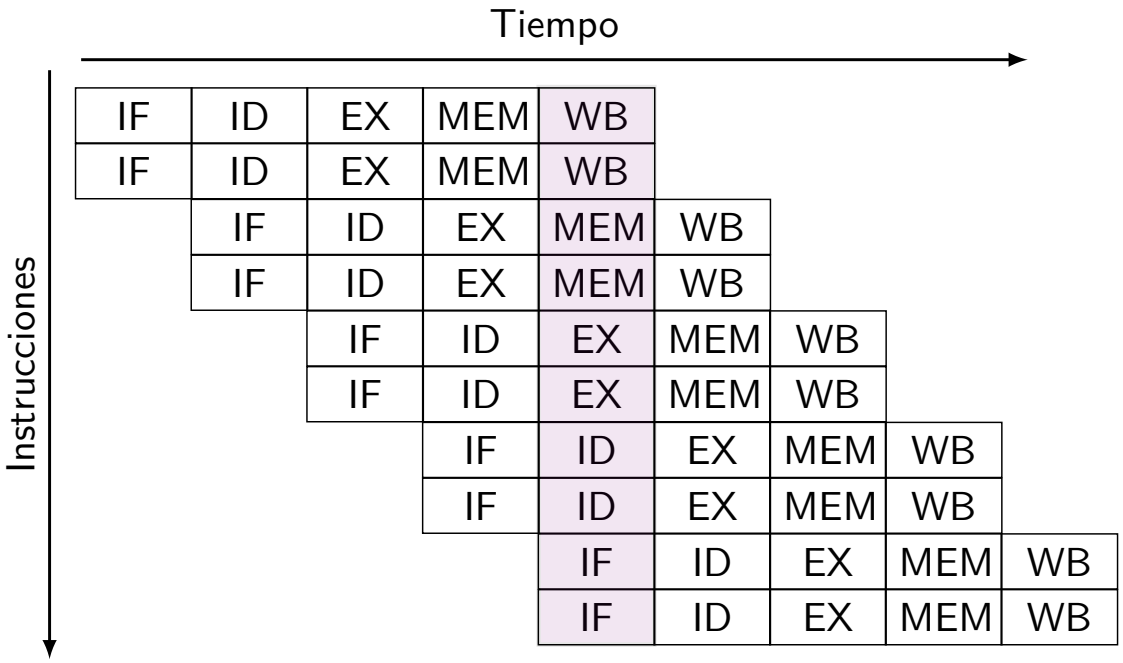


IF=Instruction Fetch, ID=Instruction Decode,
EX=Execute, MEM=Memory access, WB=Register write back

Se consigue ejecutar una instrucción por ciclo de reloj

Procesador supersegmentado: si cada fase se divide aún más

Procesadores Superescalares

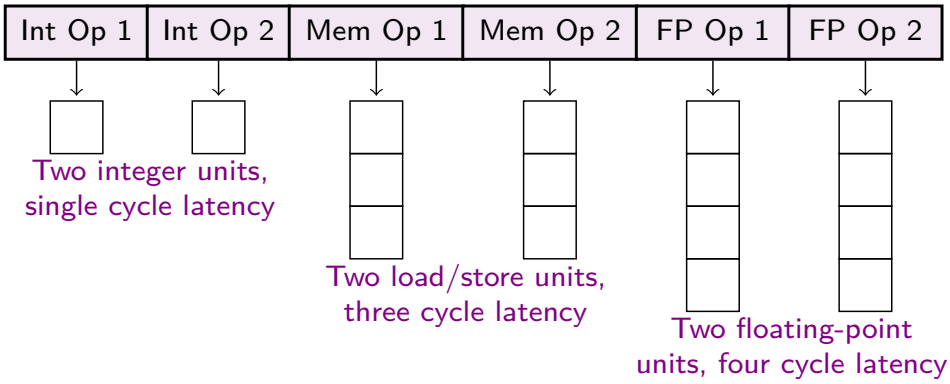


Se consigue completar dos instrucciones por cada ciclo de reloj

Procesadores VLIW

VLIW: *Very Long Instruction Word*

- Multiples operaciones empaquetadas en una instrucción
- Cada slot es para un tipo de operación determinada
- Simplifica diseño hardware, complejidad en compilador



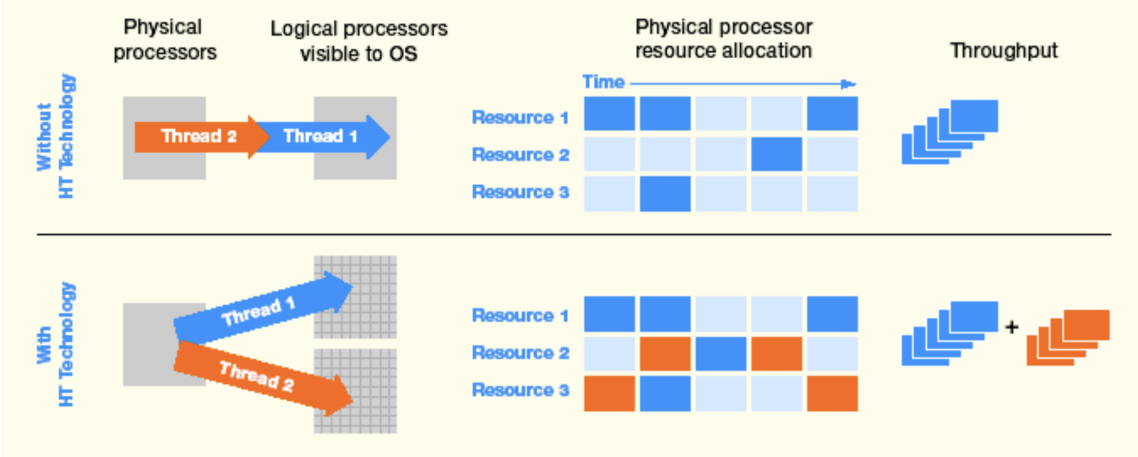
EPIC: *Explicitly Parallel Instruction Computing* (Intel Itanium)

- *Bundle* de 3 instrucciones de 41 bits más un *template* (5 bits) que indica qué unidad de ejecución procesa cada una

Hyper-Threading

Una técnica de *multi-threading* simultáneo, propietaria de Intel

- Mejora la eficiencia de procesadores superescalares cuando hay varios hilos de ejecución
- Por cada núcleo físico, el sistema operativo ve dos lógicos



Mejora de prestaciones (hasta 30 %) dependiendo de aplicación

Procesadores Vectoriales

Incluye instrucciones que operan simultáneamente sobre múltiples datos (SIMD)

- Alcanzan su cénit en 1980-1990 (Cray, Fujitsu, NEC)
- Utilizan un gran número de unidades segmentadas especializadas
- Se caracterizan por su gran potencia en operaciones en coma flotante (sumas y multiplicaciones)

Actualmente la mayoría de procesadores ofrecen instrucciones vectoriales:

- Intel: MMX, SSE, AVX
- AVX-512 permite operar con registros vectoriales de 512 bits (8 `double` o 16 `float`)

Paralelismo SIMD: SSE

Ejemplo: Operación `sscal`, versión SSE

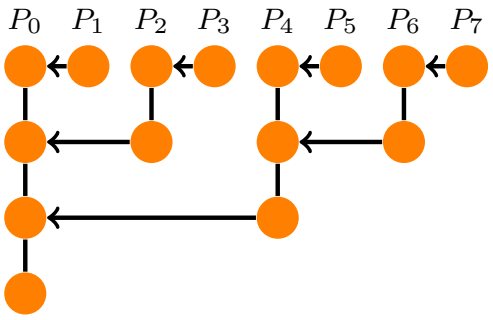
```
void sscal(int n, float alpha, float *x) {
  int i, ns;
  float alpha_vec[4];
  __m128 tmm0, tmm1;

  for (i=0; i<4; i++) {
    alpha_vec[i]=alpha;       /* 4 copias de alpha */
  }
  tmm0 = _mm_load_ps(alpha_vec);

  ns = n/4;
  for (i=0; i<ns; i++) {
    tmm1 = _mm_load_ps(&x[4*i]);     /* carga 4 x's */
    tmm1 = _mm_mul_ps(tmm1,tmm0);    /* alpha*x's   */
    _mm_store_ps(&x[4*i],tmm1);      /* guarda x's  */
  }
}
```


Procesadores Matriciales: Reducción

Algunas operaciones como la reducción 'horizontal' requieren varios ciclos



Ventajas:

- Fácil programación, paralelismo de datos

Inconvenientes:

- Dificultad para uso eficiente de recursos (programas con una gran fracción escalar)

Resumen:

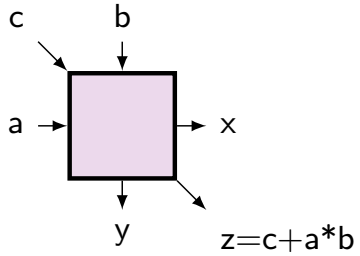
- Poca eficiencia
- Tendencias: GPUs, procesadores de propósito específico

Procesadores Sistólicos (o Matriciales)

Red homogénea de celdas sencillas (programables), calculan un resultado parcial a partir de los datos recibidos de los vecinos

- Paralelismo de grano fino
- Paralelismo síncrono
- Memoria limitada a unos pocos registros
- Evitan cuello de botella de accesos a memoria

Elemento de proceso



Procesadores de propósito específico

- Procesado de señal (FFT, filtrado), Google TPU, ...
- Actualmente se implementan con FPGAs, ASIC