



# Programación Paralela - CC332

2021-I

---

José Fiestas PhD

21/04/21

Universidad Nacional de Ingeniería  
[jose.fiestas@uni.edu.pe](mailto:jose.fiestas@uni.edu.pe)

# Introducción al curso

El estudiante conocerá las técnicas y paradigmas del paralelismo y dominará las herramientas para lograr demostrar la escalabilidad de software, optimizando su precisión y minimizando su complejidad.

La competencias logradas por el estudiante al finalizar el curso son:

- Construye algoritmos y diseña soluciones en paralelo de carácter interdisciplinario
- Justifica el impacto de las soluciones encontradas a nivel social y profesional
- Desarrolla el aprendizaje autónomo en el desarrollo de los puntos anteriores

**Consola en Linux / Mac OS**  
**MPICH / OMP / CUDA**

1. Fundamentos de paralelismo y arquitecturas paralelas
2. Metodos de paralelismo
3. Descomposición en paralelo
4. Comunicación y sincronización
5. Análisis y programación de algoritmos paralelos
6. Desempeño en paralelo

# Unidad 1: Fundamentos de paralelismo y arquitecturas paralelas

Al finalizar la unidad, los alumnos conocen:

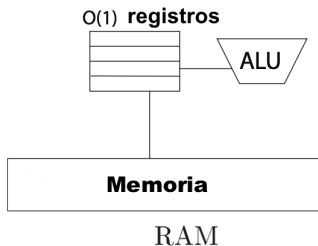
1. Transición del procesamiento secuencial al paralelo
2. Procesadores mutlinúcleo. Taxonomía de Flynn
3. Metas del Paralelismo: velocidad y precisión
4. Memoria compartida vs memoria distribuida.

# **1. Del procesamiento secuencial al paralelo**

---

# Random Access Machine - modelo Von Neumann

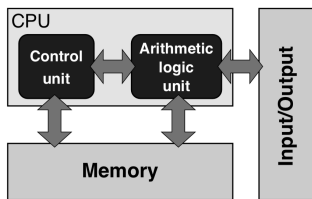
Consiste en  
un procesador con acceso a una  
memoria suficientemente grande y  
un número constante de registros  
que pueden ser procesados  
en unidades aritméticas / lógicas  
La velocidad limitada  
de acceso a memoria constituye  
el **von Neumann bottleneck**



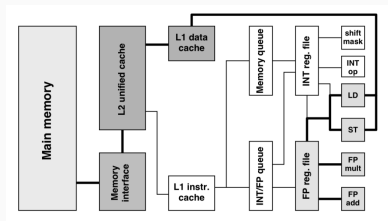


# Stored Programm Computer Architecture

La primera implementación de una arquitectura de computadoras (**EDVAC**, 1949) consiste en la lectura de datos a través de una unidad de control, siendo el cálculo propiamente dicho derivado a una unidad aritmética y lógica. Estas constituyen el **CPU** (Unidad Central de Procesamiento), conectadas a la memoria y a la unidad de input/output,



La estructura de un microprocesador es compleja. Contiene unidades aritméticas enteras (**INT**) y de coma flotante (**FP**). La información es guardada en registros (INT y FP) que acceden a las instrucciones sin mayor retraso. Unidades de carga (**LD**) y almacenamiento (**ST**) transfieren información entre registros. **Caches**, almacenan data para su reutilización

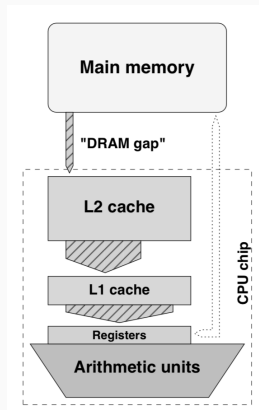


- Conteo de ciclos  $T(I)$  por instancia de un problema. Peor caso  $T(n) = \max_{|I|=n} T(I)$
- Caso promedio:  $T_{avg} = \frac{\sum_{|I|=n} T(I)}{|\{I: |I|=n\}|}$ . E.g. Quicksort es  $O(n \log n)$
- en algoritmos randomizados  $T(n)$  es una variable estocástica y se necesita determinar su valor esperado. E.g. Quicksort con pivot aleatorio posee peor caso  $E[T(n)] = O(n \log n)$
- la complejidad puede depender de varios parámetros. E.g. Algoritmo de Dijkstra de caminos mas cortos necesita  $O(m+n \log n)$ , donde  $n$  es el número de nodos y  $m$  de aristas.

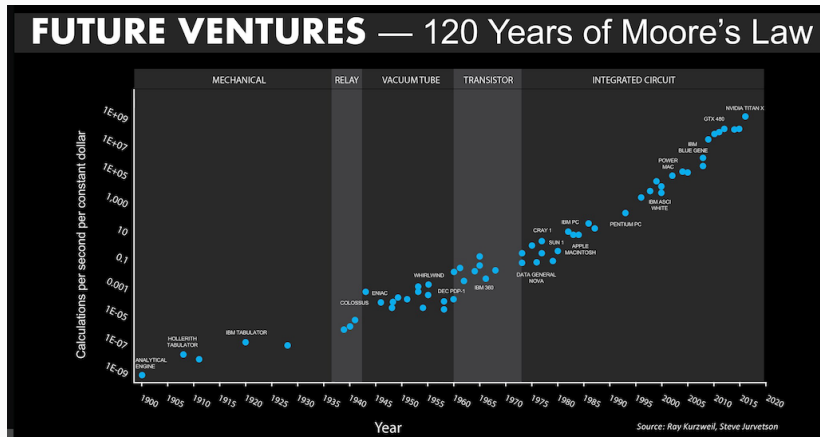
# Métricas de Performance y Benchmarks

## El Peak Performance

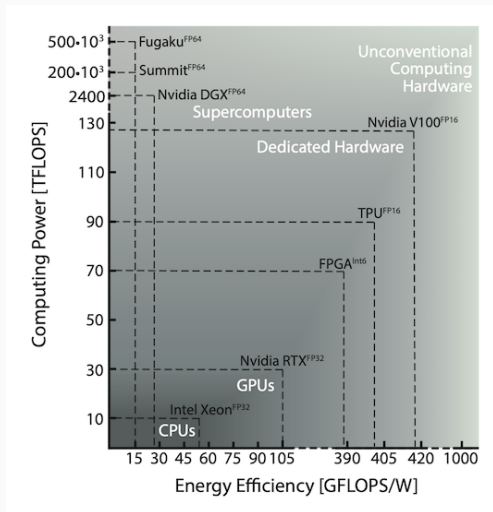
depende de muchos factores, y se centra en la velocidad de las operaciones de suma y multiplicación de coma flotante, medida en **FLOPs** (Floating Point Operations per second). Aritmética más compleja (división, raíz, trigonometría, logaritmos, etc) es mucho mas lenta y deben evitarse en HPC. Microprocesadores que generan dos o cuatro Double Precisión (FPs) por clock cycle, siendo la frecuencia entre 2-3 GHz, obtienen una velocidad de 4-12 GFlops por core. El camino de acceso a Caches y Memoria de mide en **GBytes/sec**, que junto a los **GFlops/sec** constituyen las métricas relevantes de performance para microprocesadores.



El número de transistores que puede contener un circuito integrado a un costo razonable, se duplica cada dos años.



# Eficiencia energética (Greencomputing)



Kirill P. Kalinin and Natalia G. Berloff, 2021,  
<https://arxiv.org/pdf/2104.02553.pdf>

## **Problemas debido a que:**

- Poder de procesadores crece mas rápido que mejoras en acceso a memoria
- Latencias de memoria decrecen lentamente
- Número de núcleos por módulo de memoria crece

## **Por ello:**

La distancia entre velocidad de memoria y la eficiencia teórica de los núcleos crece

Procesadores pierden tiempo (ciclos) mientras esperan la información  
Esto hace difícil explotar al máximo eficiencia teórica de procesadores

## **Soluciones: ...**

Incrementar la frecuencia de reloj (clock count) de un transistor a través de técnicas , tales como:

- Pipelining
- Escalabilidad
- Instrucciones SIMD (Single Instruction Multiple Data)
- incremento de la memoria de Cache
- Procesadores multicore

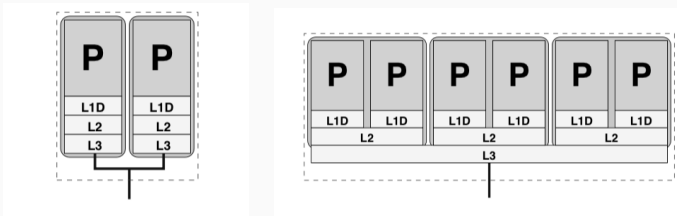


# Procesadores multicore

Microprocesadores sufren de:

- disipación de calor  $W + \delta W = (1 + f)^3 W$
- necesidad de mayores frecuencias de reloj

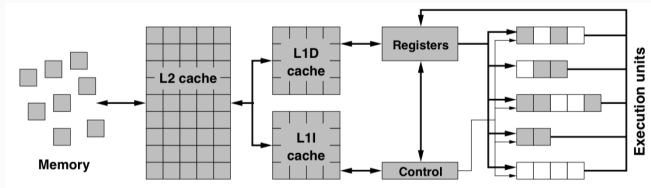
Lo que ha derivado en el desarrollo de procesadores de múltiples núcleos

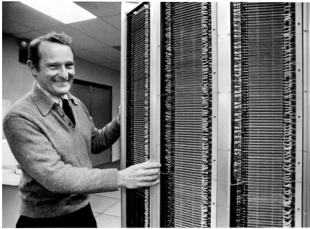


E.g.: IBM POWER hasta 19 núcleos, Intel Xeon-phi hasta 72 núcleos  
Arquitecturas híbridas(GPU o XeonPhi con standard CPUs)

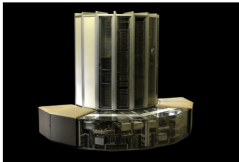
# Procesadores multithreading

Se trata de la implementación de multitareas como duplicados de la arquitectura del CPU (Hyperthreading o Simultaneous Multithreading). Estos son los procesadores lógicos del CPU que ejecutan tareas en paralelo





**Seymour Cray** (Padre de la Supercomputación) fundó CDC (Control Data Corporation) y construyó el supercomputador más rápido de la época



**(a)** Cray-1, 80 MFlops,  
\$ 8.8 M



**(b)** Apple A13, 690  
GFlops, \$ 400



Photo: RIKEN

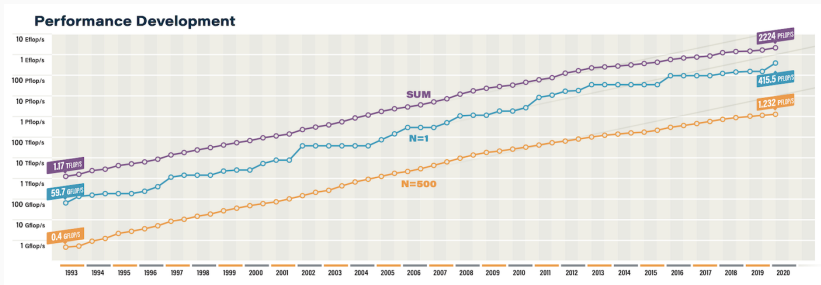
Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	<b>Supercomputer Fugaku -</b> Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,299,072	415,530.0	513,854.7	28,335



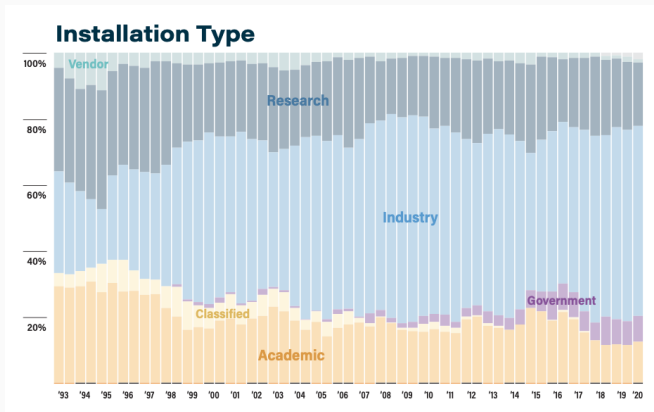
2	<b>Summit</b> - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
---	---	-----------	-----------	-----------	--------

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	<b>Supercomputer Fugaku</b> - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, <b>Fujitsu</b> RIKEN Center for Computational Science Japan	7,299,072	415,530.0	513,854.7	28,335
2	<b>Summit</b> - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, <b>IBM</b> DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	<b>Sierra</b> - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, <b>IBM / NVIDIA / Mellanox</b> DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	<b>Sunway TaihuLight</b> - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, <b>NRCPC</b> National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	<b>Tianhe-2A</b> - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, <b>NUDT</b> National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482

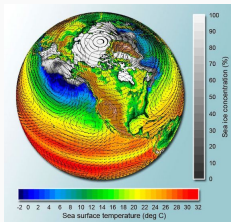
## Velocidad de Supercomputadoras crece exponencialmente



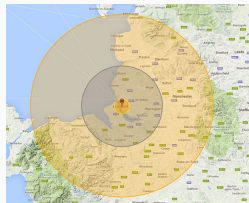
El avance industrial/comercial/académico se logra gracias a la supercomputación







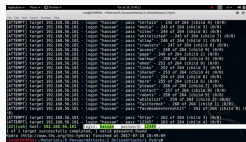
(a) modelos climáticos



(b) pruebas nucleares



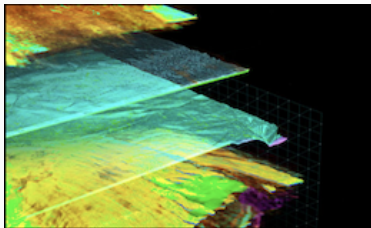
(c) accidentes  
automovilísticos



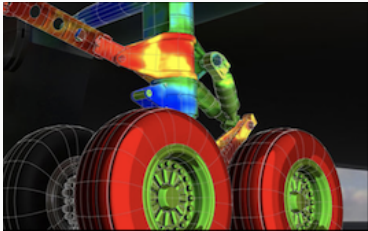
(d) decodificación



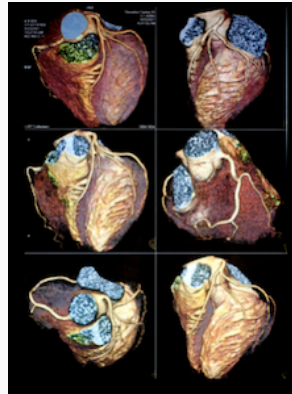
**(a)** Sistemas MultiGPU son usados para procesamiento gráfico de alta performance y en paralelo



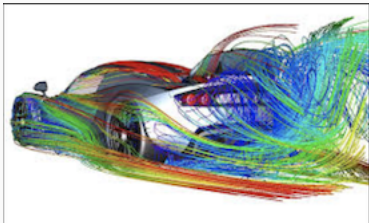
**(b)** Análisis sísmico en sistemas Multi-GPU, para mejorar interpretación de datos en la exploración de gas y petróleo



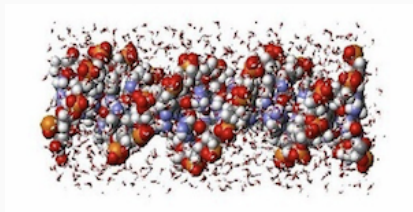
(a) Diseño realístico en sistemas Muti-GPU, acorta tiempos de diseño logrando llegar más rápido al mercado.



(b) Procesamiento de imágenes médicas, para manipular imágenes en 3D

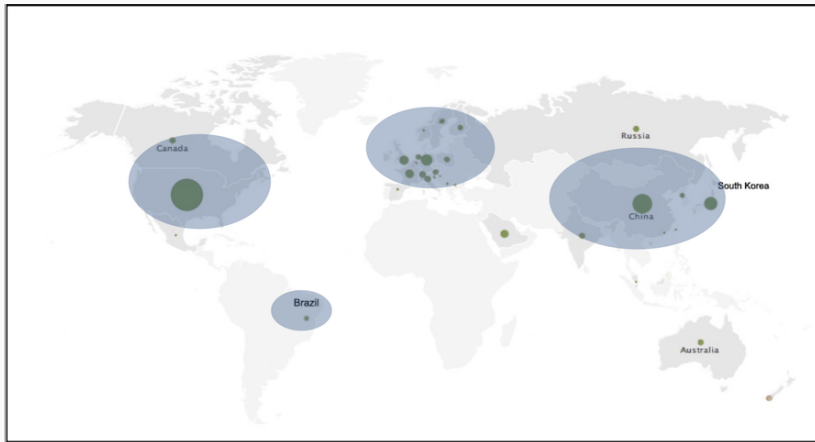


**(a)** Dinámica de fluídos, utiliza software sofisticado que requiere alto poder computacional



**(b)** Dinámica molecular, para mejorar productos que no afecten el medio ambiente

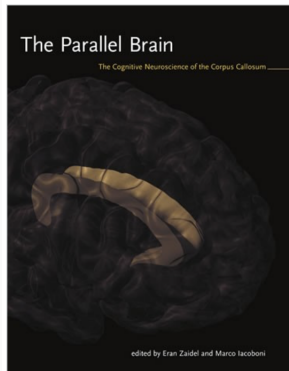
# Performance mundial (2020)



Distribución de performance (FLOPs) a nivel mundial. Elaboración propia

Compare  
el cerebro humano con un computador:  
¿Funciona el cerebro humano en paralelo?  
¿Cuánto podemos recordar?

perspectives to this question. The emerging view is that the corpus callosum consists of many parallel interhemispheric channels for communication and control, and that every transfer channel is context-dependent and modulated by attention.



**¿Qué tan complicado es programar en paralelo?**

**¿Por qué en paralelo?**

Diseñamos **algoritmos paralelos** utilizando un lenguaje formal (modelo de costo), que los hace **abstractos** y **confiables**

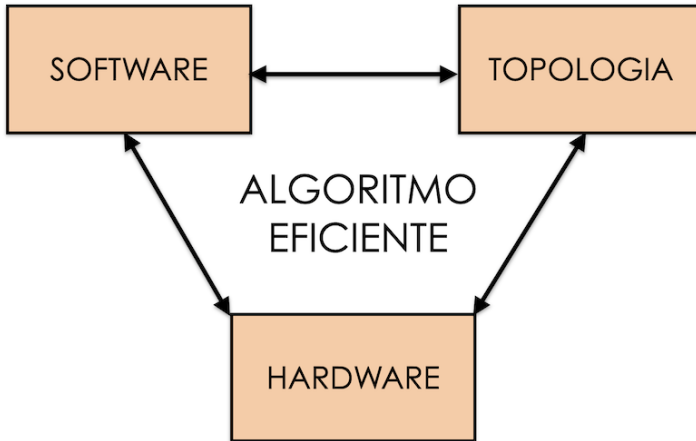
Ejemplo: reservar un tour ...






Generalmente buscamos mejorar el performance de un código secuencial en un solo **core**, ya que:

- el tiempo de ejecución no es 'aceptable'
- la dimensión del problema (resolución, cantidad de objetos, complejidad) exige mayor cantidad de memoria



# Diseño de un algoritmo paralelo



-  David B. Kirk and Wen-mei W. Hwu *Programming Massively Parallel Processors: A Hands-on Approach*. 2nd. Morgan Kaufmann, 2013. isbn: 978-0-12-415992-1.
-  Norm Matloff. *Programming on Parallel Machines*. University of California, Davis, 2014.
-  Peter S. Pacheco. *An Introduction to Parallel Programming*. 1st. Morgan Kaufmann, 2011. isbn: 978-0-12-374260- 5.
-  Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003. isbn: 0071232656.
-  Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Program- ming*. 1st. Addison-Wesley Professional, 2010. isbn: 0131387685, 9780131387683.