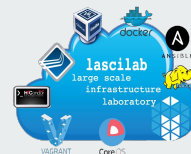




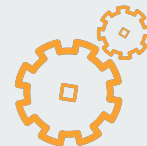
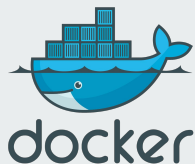
2 Seminario de Plataformas Computacionales

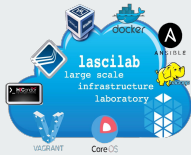
Computación de Alto Rendimiento y Reproducibilidad Experimental



Introducción a la Computación Paralela

Aurelio Vivas - aurelio.vivas@correounivalle.edu.co





Lo que aprenderás hoy

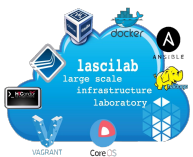
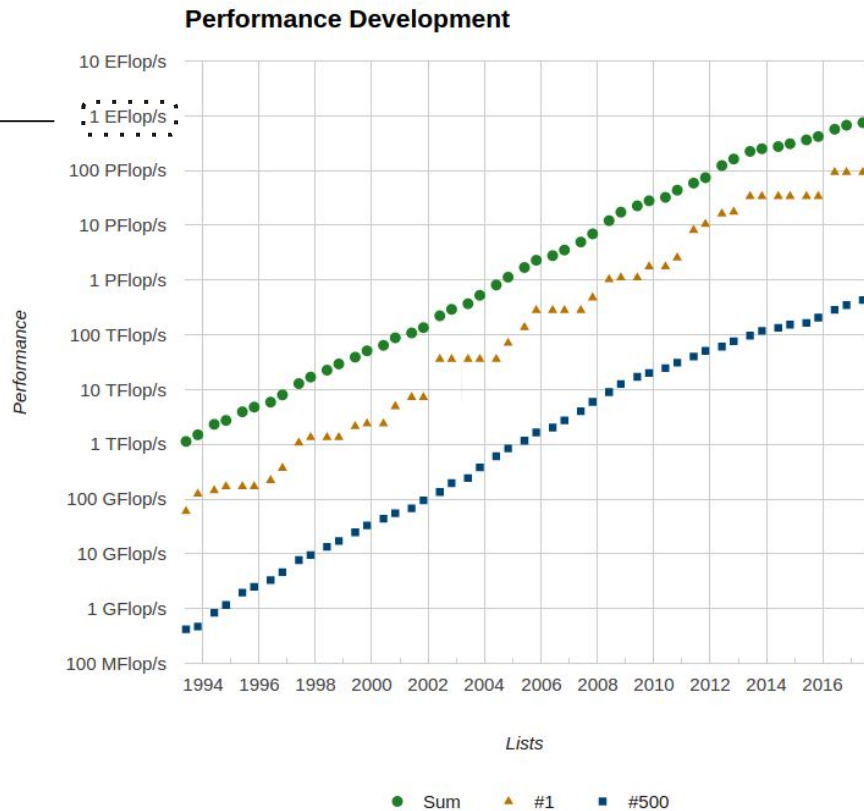
¿Cómo es llevado a cabo el paralelismo un computador?

- ¿Por qué la Computación Paralela es importante?
- ¿Qué es la Computación Paralela?
- ¿Por qué surge?
- Posible solución “Computación Paralela”
- Perspectiva de Paralelismo en la Historia.
- Aplicaciones
- Conclusiones
- Resumen

Exa Float Operations Second

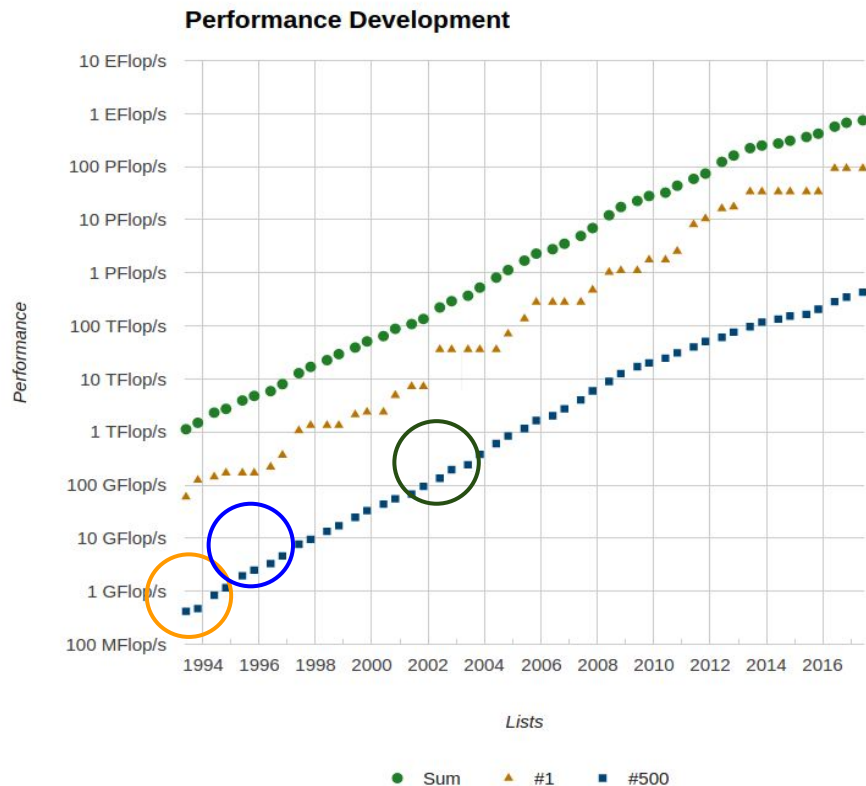
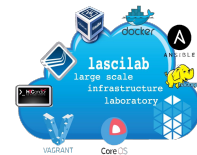
1.000.000.000.000.000.000
Float Operations
per Second

1 EFlop/s



¿Por qué la computación paralela es importante?

¿Por qué la computación paralela es importante?



~2.79 GFLOPs/core
Secuencial

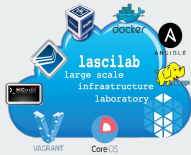
~13.69 GFLOPs/computador cpus
Paralelismo

~641.3 GFLOPs/computador gpus
Paralelismo

Mi Computadora vs Las SuperComputadoras

¿Por qué la Computación Paralela es importante?

Permite a nuestras aplicaciones tener acceso al poder computacional disponible en las computadoras paralelas (PCs, Clusters, Supercomputadores).

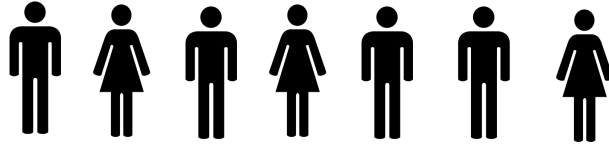
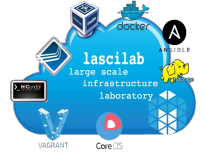


Lo que aprenderás hoy

¿Cómo es llevado a cabo el paralelismo un computador?

- ¿Por qué la Computación Paralela es importante?
- ¿Qué es la Computación Paralela?
- ¿Por qué surge?
- Posible solución “Computación Paralela”
- Perspectiva de Paralelismo en la Historia.
- Aplicaciones
- Conclusiones
- Resumen

¿Qué es la Computación Paralela?



Baño



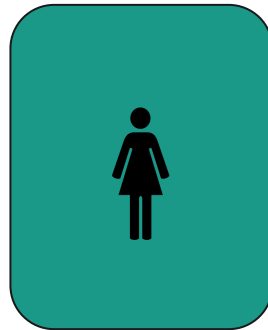
Secuencial

¿Qué es la Computación Paralela?

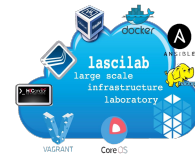


Secuencial

Baño

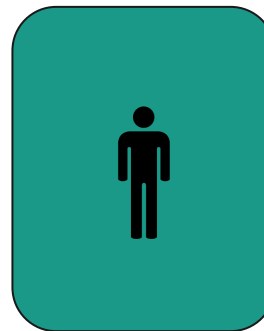


¿Qué es la Computación Paralela?



Secuencial

Baño

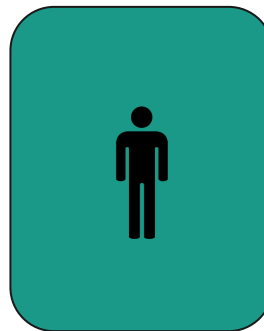


¿Qué es la Computación Paralela?

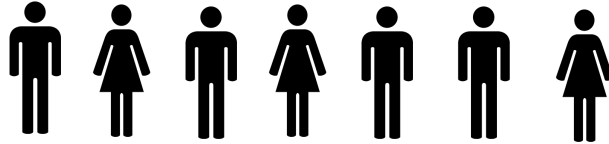
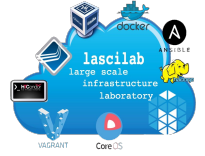


Secuencial ...

Baño

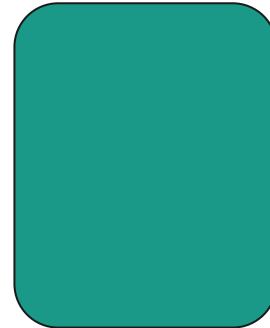


¿Qué es la Computación Paralela?

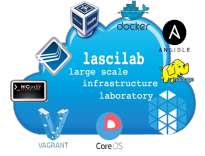


Paralelo

Baños



¿Qué es la Computación Paralela?

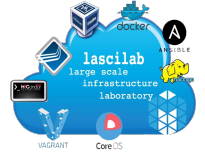


Paralelo

Baños



¿Qué es la Computación Paralela?

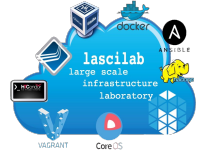


Paralelo

Baños



¿Qué es la Computación Paralela?



Baños



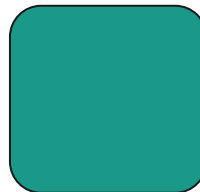
Paralelo ...

¿Qué es la Computación Paralela?



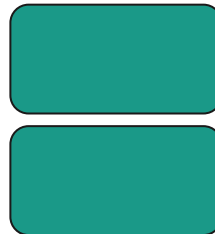
Secuencial

Baño



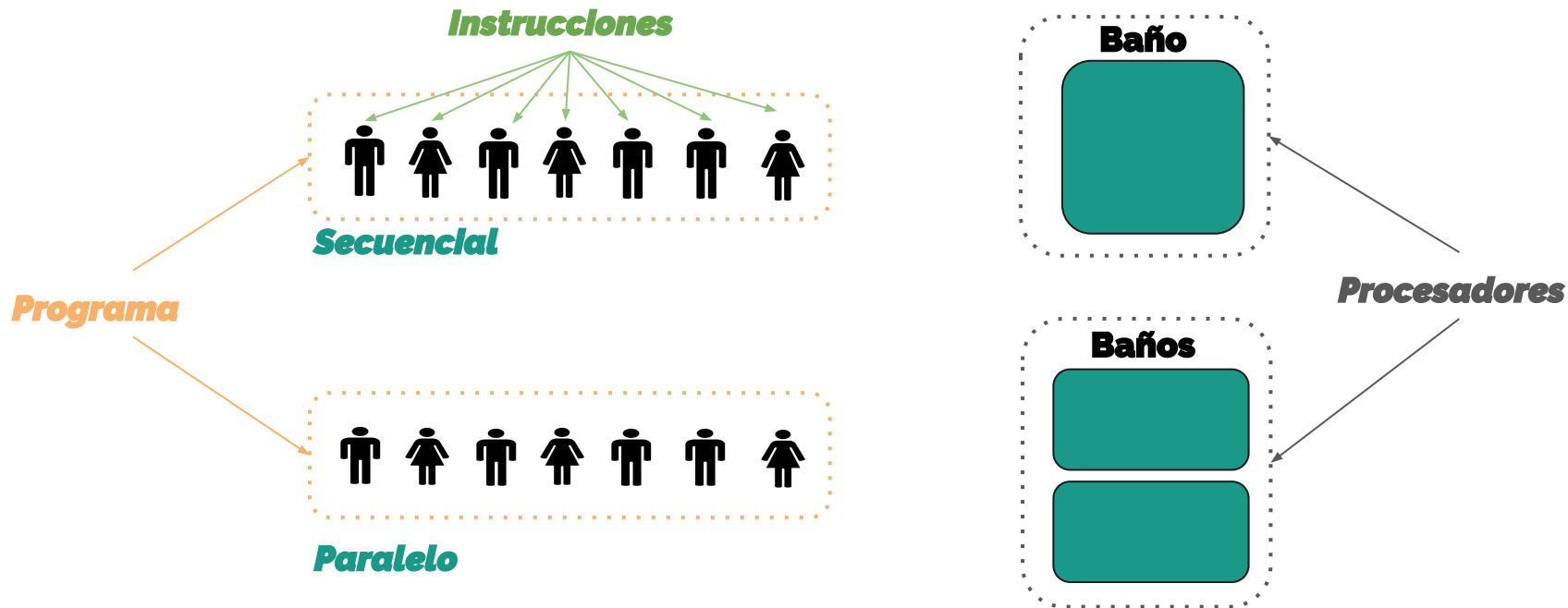
Paralelo

Baños

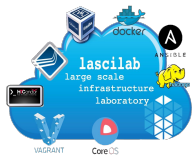


¿Cuántos slides debemos pasar para que todas las personas pasen por el baño?

¿Qué es la Computación Paralela?



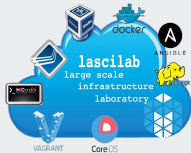
¿Cuántos slides debemos pasar para que todas las personas pasen por el baño?



¿Qué es la Computación Paralela?

“Es una forma de cómputo en la que muchas instrucciones se ejecutan **simultáneamente.**”

“Opera sobre el principio de que **problemas grandes, a menudo se pueden dividir en unos más pequeños**, que luego son resueltos simultáneamente.”



Lo que aprenderás hoy

¿Cómo es llevado a cabo el paralelismo un computador?

- ¿Por qué la Computación Paralela es importante?
- ¿Qué es la Computación Paralela?
- ¿Por qué surge?
- Posible solución “Computación Paralela”
- Perspectiva de Paralelismo en la Historia.
- Aplicaciones
- Conclusiones
- Resumen

GitHub

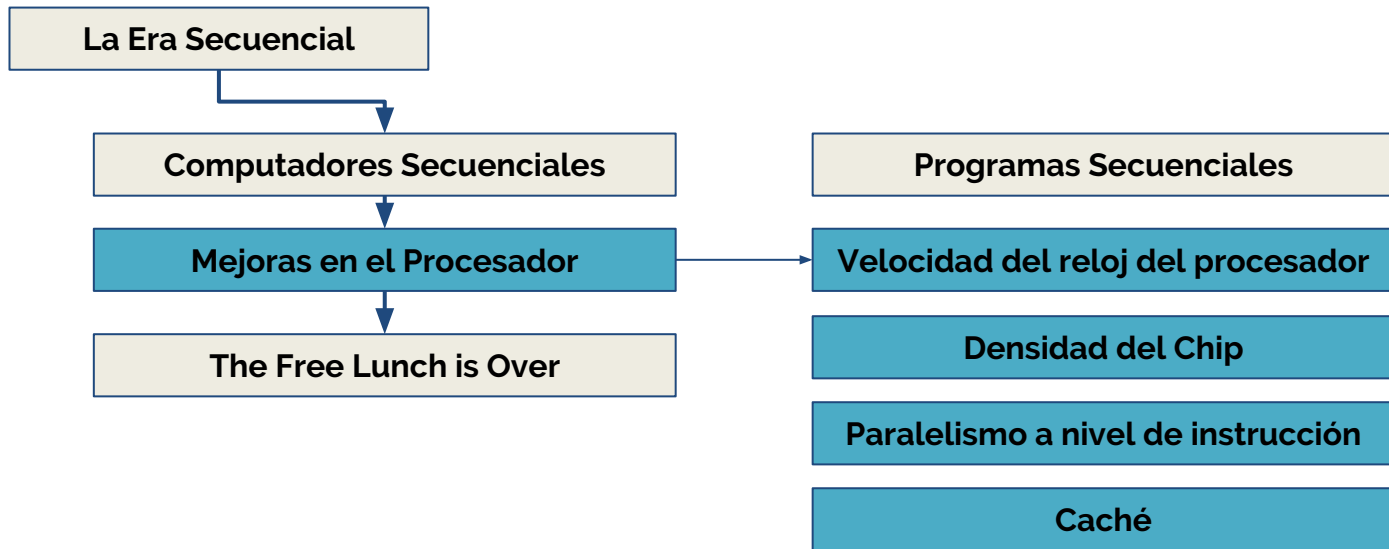


Veamos un ejemplo de un programa secuencial

```
git clone git@github.com:DonAurelio/openmp-workshop.git
```

```
cd openmp-workshop/1.problem
```

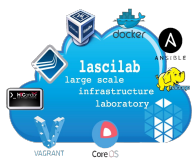






Secuencial

Baño



Aumento en la velocidad del reloj del procesador

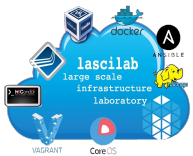
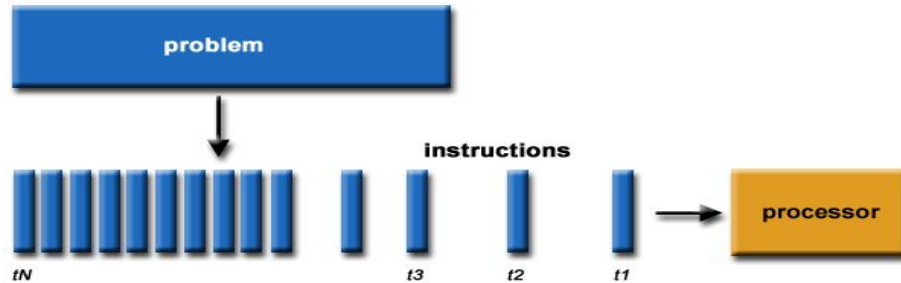


Secuencial

Baño



500 MHz, 1 GHz, 2 GHz, 3 GHz, 3.8 GHz,....., ? GHz

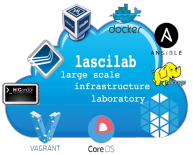


Aumento en la velocidad del reloj del procesador ([ejemplo](#))

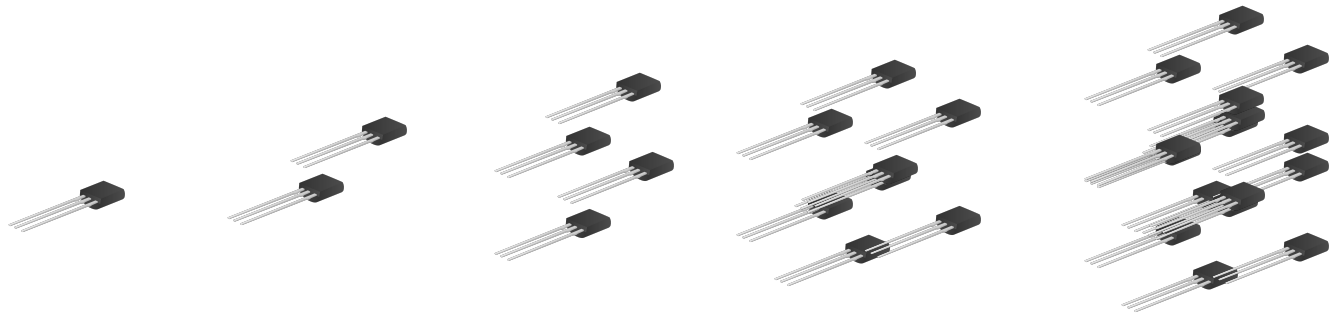
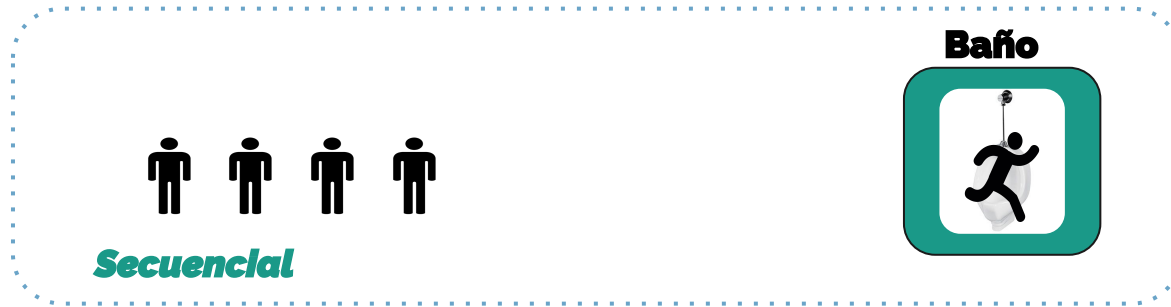


Secuencial

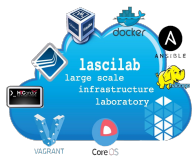
Baño



Aumento en el número de transistores en el chip del procesador



Calor Generado

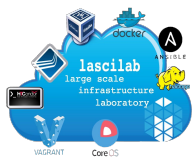
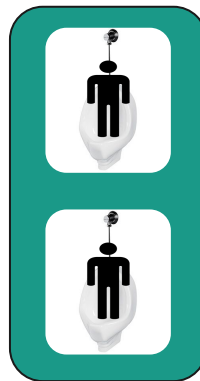


Aumento en el número de transistores en el chip del procesador



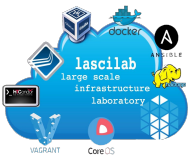
Secuencial

Baño

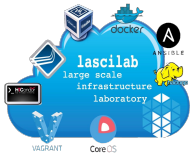


Paralelismo a nivel de instrucción

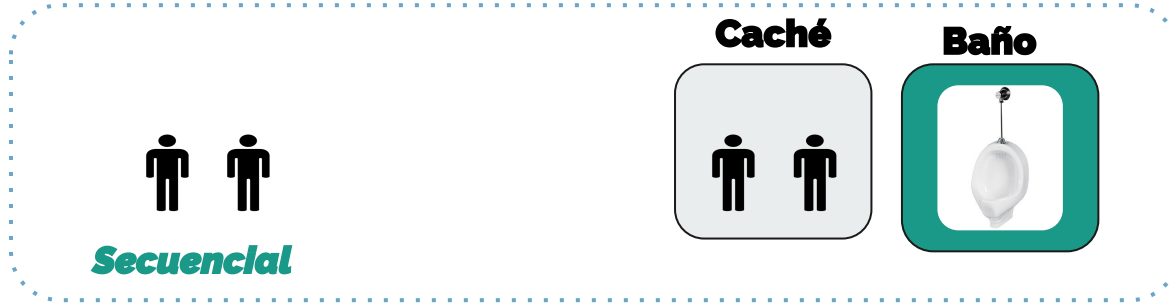
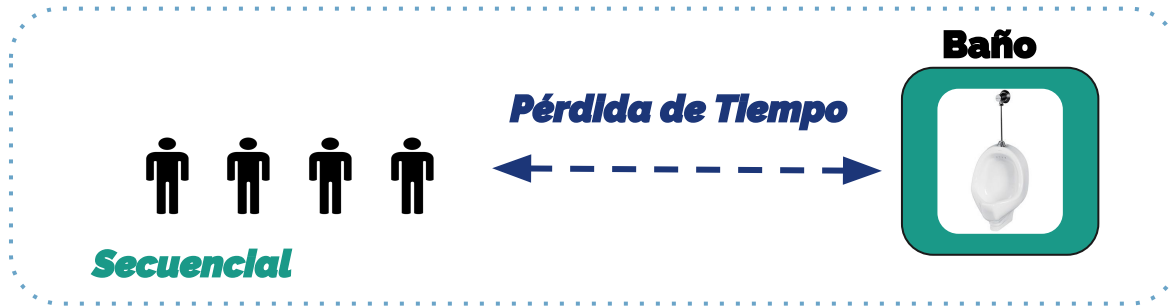
Sequential Execution	Instruction-Level Parallelism
1. $a = 10 + 5$ 2. $b = 12 + 7$ 3. $c = a + b$	1.A. $a = 10 + 5$ 1.B. $b = 12 + 7$ 2. $c = a + b$
Instructions: 3 Cycles: 3	Instructions: 3 Cycles: 2 (-33%)



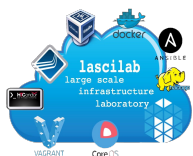
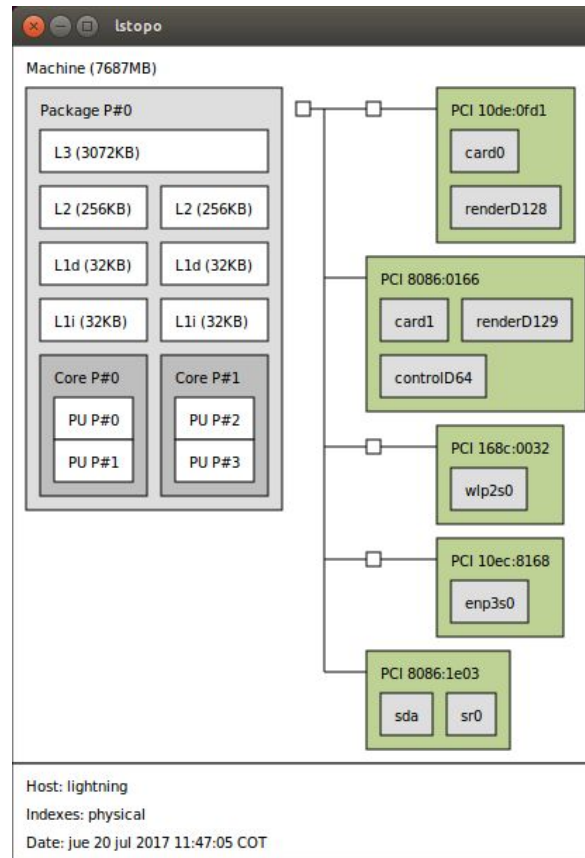
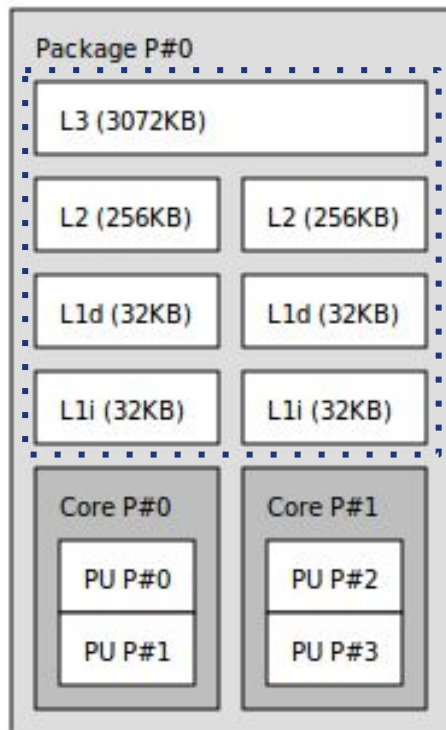
Paralelismo a nivel de instrucción



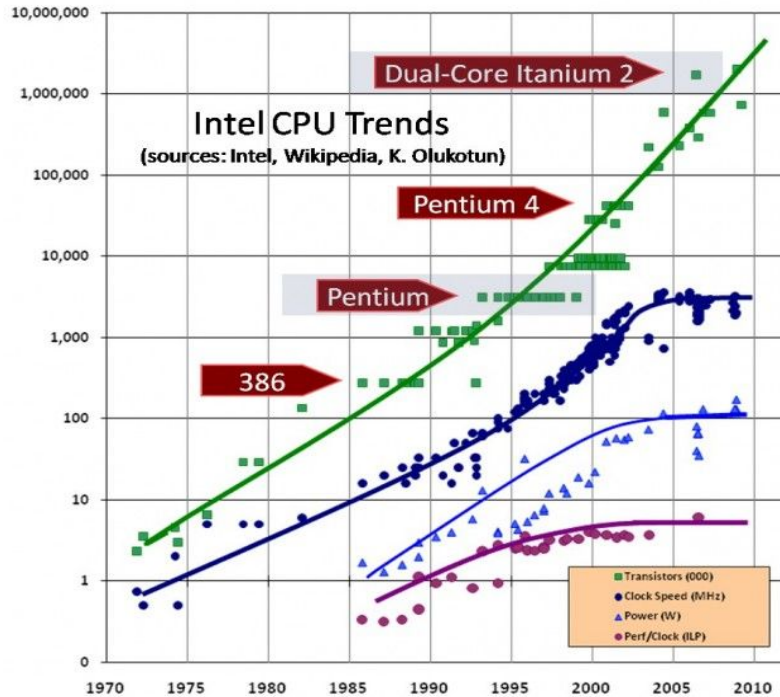
Memoría cache



Machine (7687MB)



Memoria cache

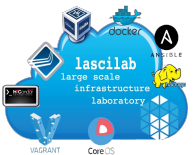


Número de transistores

Velocidad del reloj

Energía en Vatios consumida

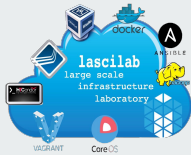
Paralelismo a nivel de instrucción



Tendencia en el diseño de procesadores.

**Chips simples, muchos de
estos ...**

—

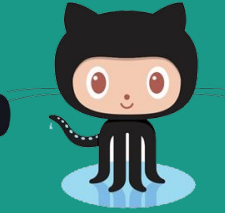


Lo que aprenderás hoy

¿Cómo es llevado a cabo el paralelismo un computador?

- ¿Por qué la Computación Paralela es importante?
- ¿Qué es la Computación Paralela?
- ¿Por qué surge?
- **Posible solución “Computación Paralela”**
- Perspectiva de Paralelismo en la Historia.
- Aplicaciones
- Conclusiones
- Resumen

GitHub



Veamos un ejemplo de un programa paralelizado

```
git clone git@github.com:DonAurelio/openmp-workshop.git
```

```
cd openmp-workshop/3.solution
```



La Era del Paralelismo



Computadores Paralelos



SISD

SIMD

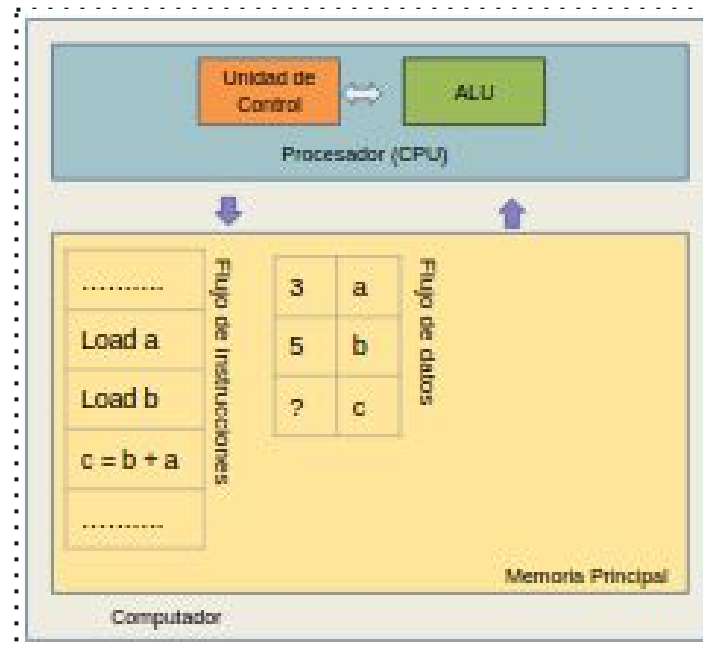
MISD

MIMD

Programas Paralelos

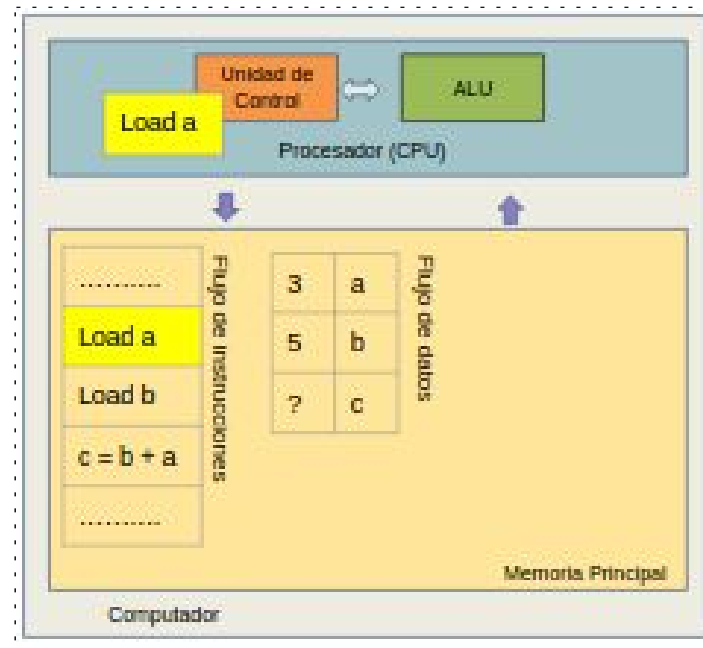
SISD (Single Instruction Stream, Single Data Stream)

- Es un computador no paralelo.
- Se caracteriza por operar sobre un flujo de instrucciones y un flujo de datos en el tiempo.



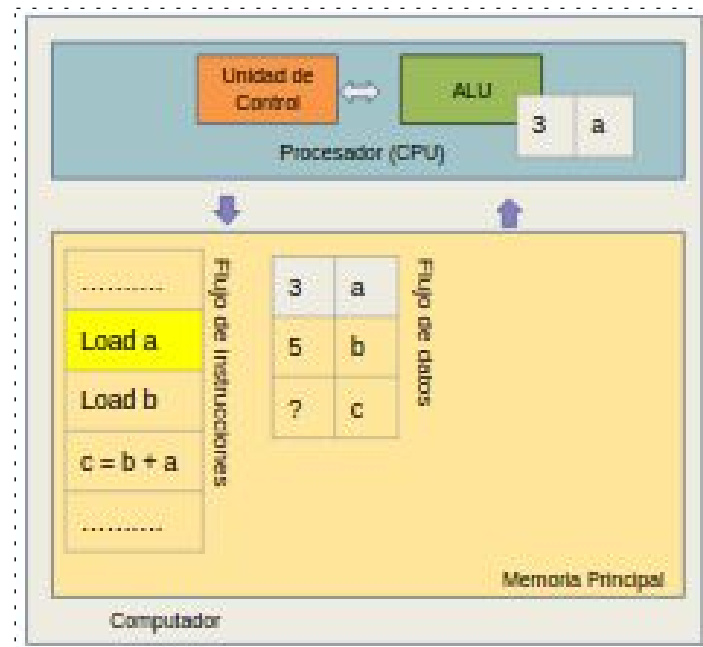
SISD (Single Instruction Stream, Single Data Stream)

- Es un computador no paralelo.
- Se caracteriza por operar sobre un flujo de instrucciones y un flujo de datos en el tiempo.



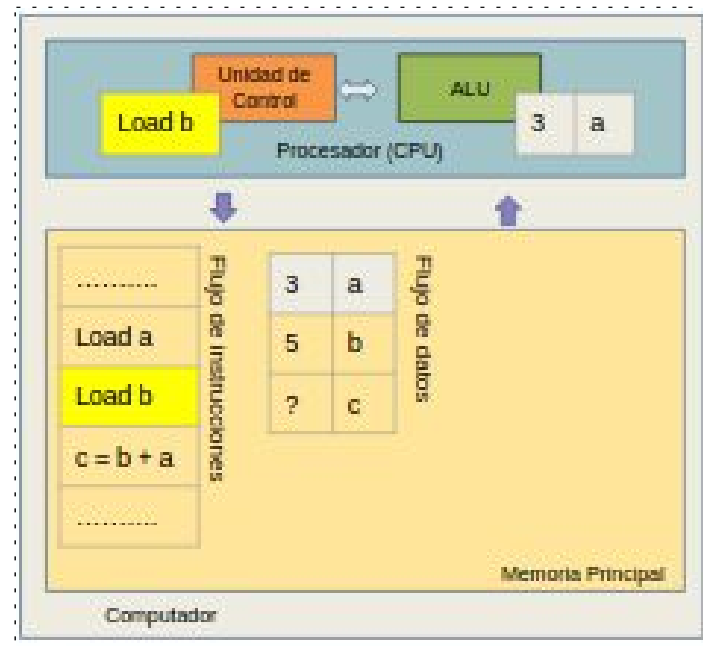
SISD (Single Instruction Stream, Single Data Stream)

- Es un computador no paralelo.
- Se caracteriza por operar sobre un flujo de instrucciones y un flujo de datos en el tiempo.



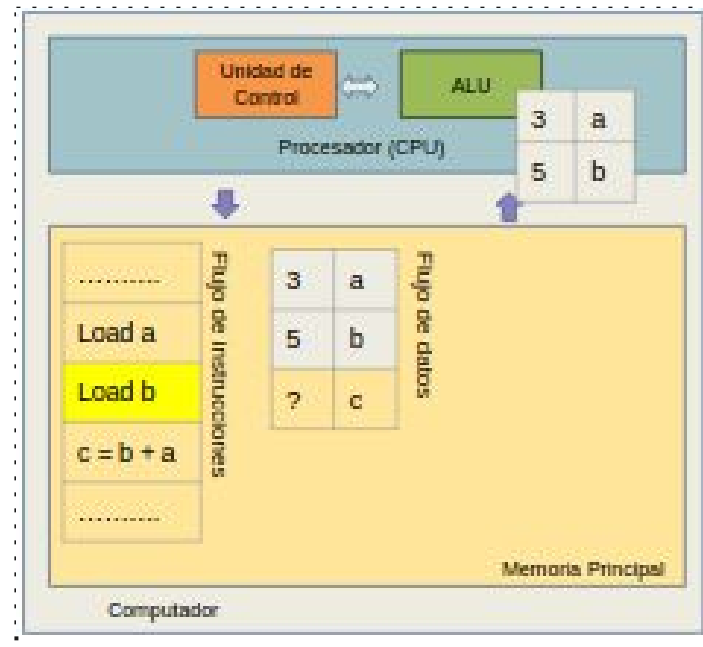
SISD (Single Instruction Stream, Single Data Stream)

- Es un computador no paralelo.
- Se caracteriza por operar sobre un flujo de instrucciones y un flujo de datos en el tiempo.



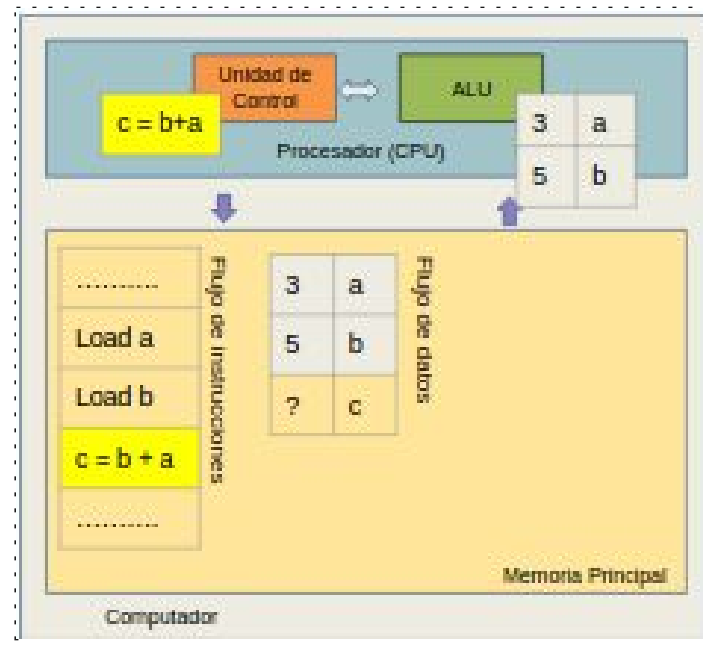
SISD (Single Instruction Stream, Single Data Stream)

- Es un computador no paralelo.
- Se caracteriza por operar sobre un flujo de instrucciones y un flujo de datos en el tiempo.



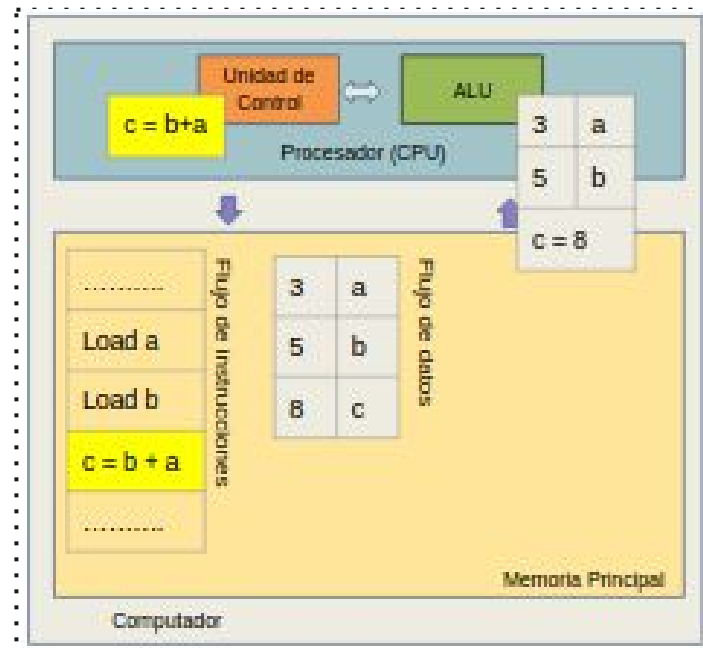
SISD (Single Instruction Stream, Single Data Stream)

- Es un computador no paralelo.
- Se caracteriza por operar sobre un flujo de instrucciones y un flujo de datos en el tiempo.



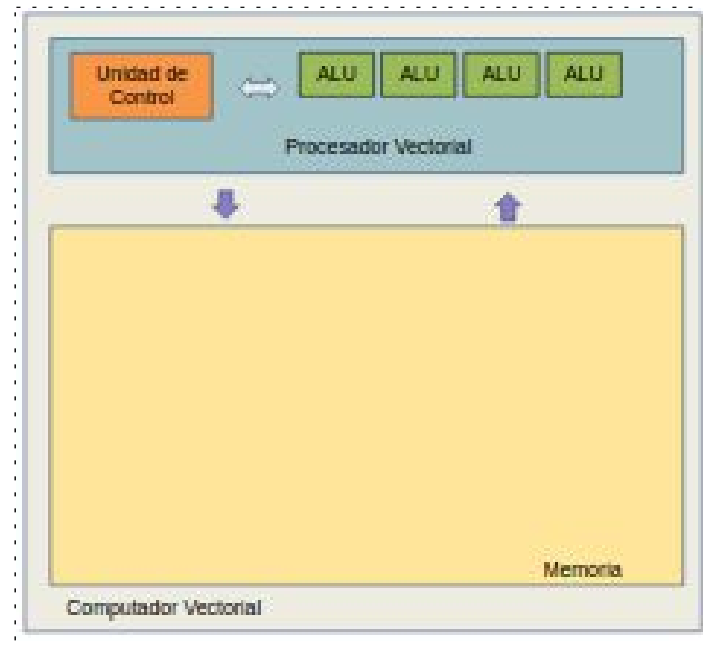
SISD (Single Instruction Stream, Single Data Stream)

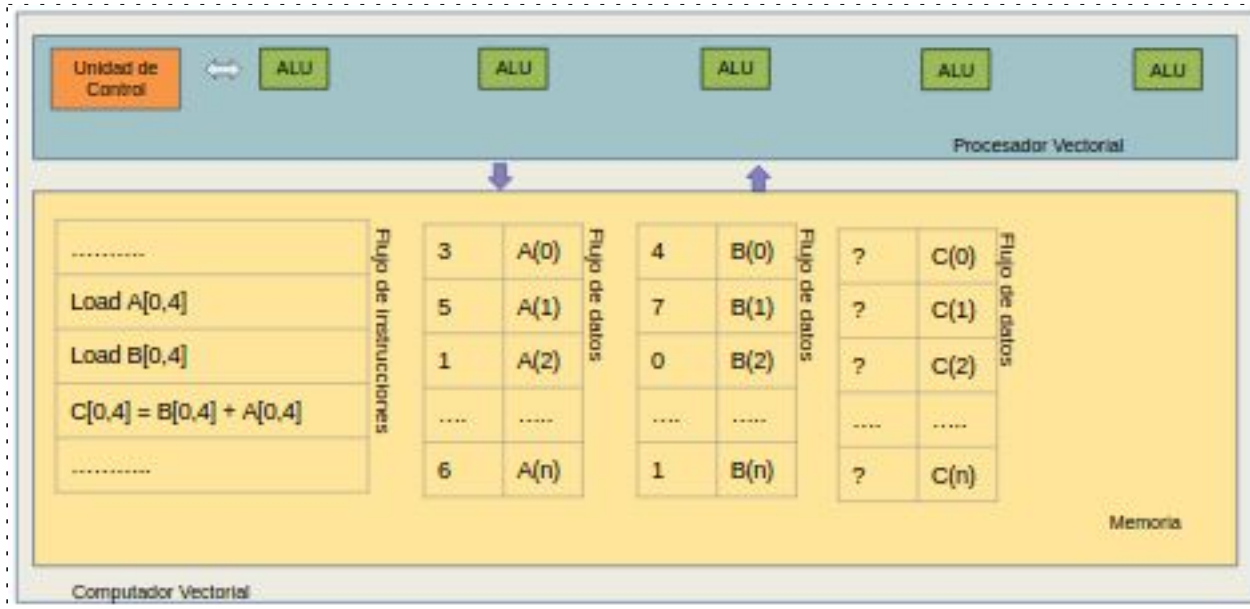
- Fueron diseñados para correr programas **secuenciales**.
- El sistema operativo permite la **conurrencia entre programas**.

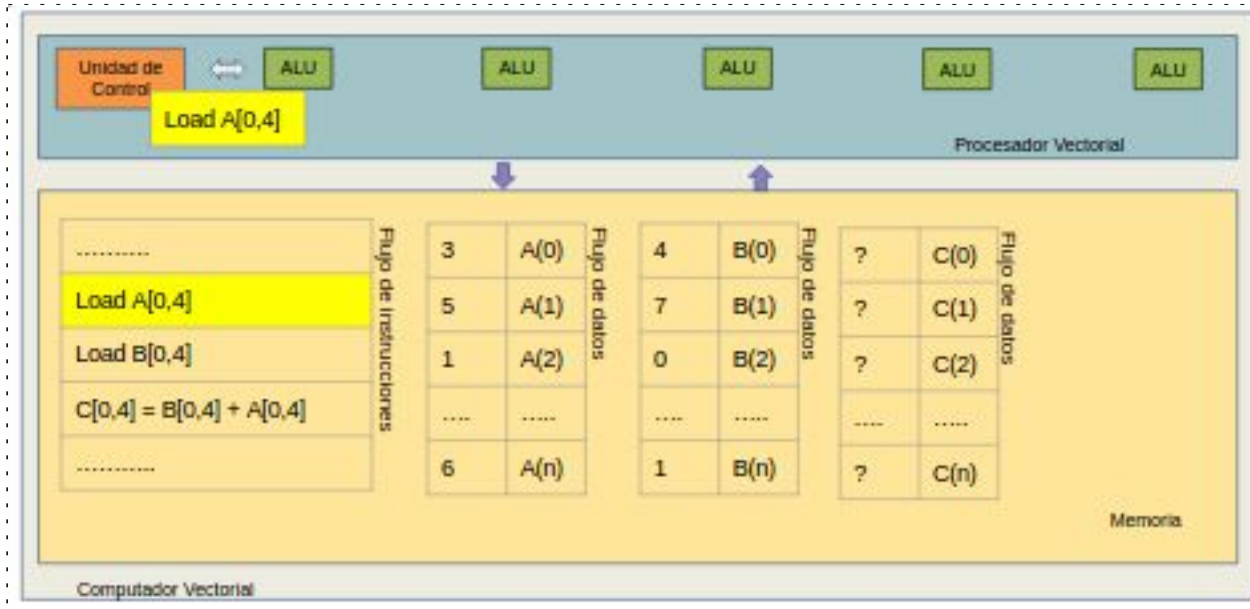


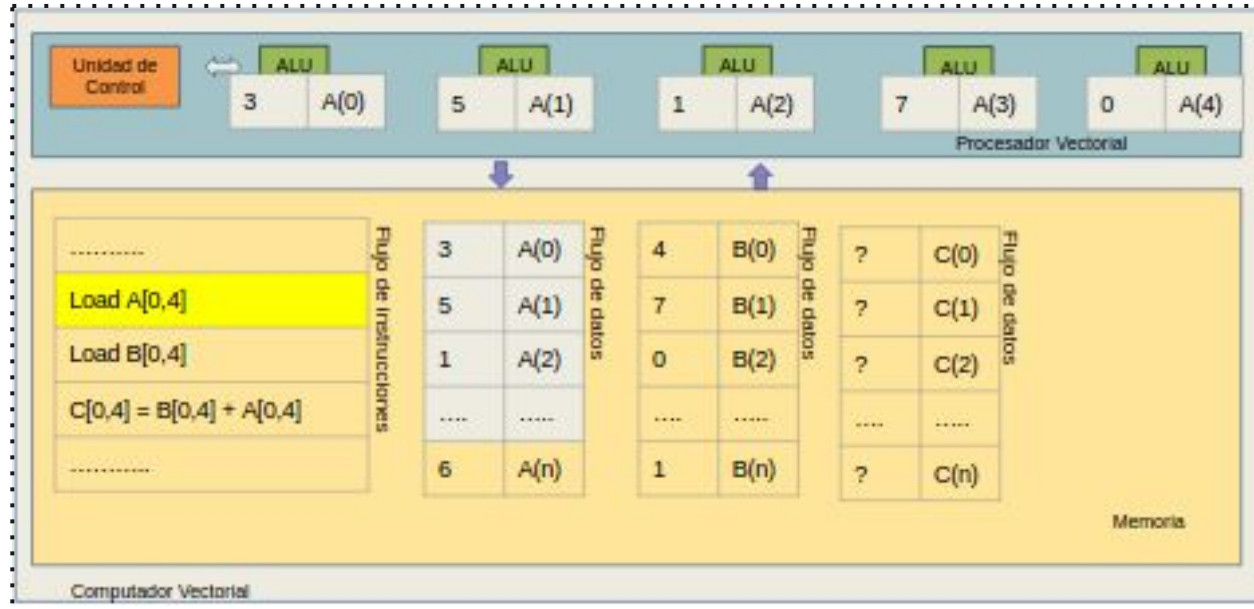
SIMD (Single Instruction Stream, Multiple Data Stream)

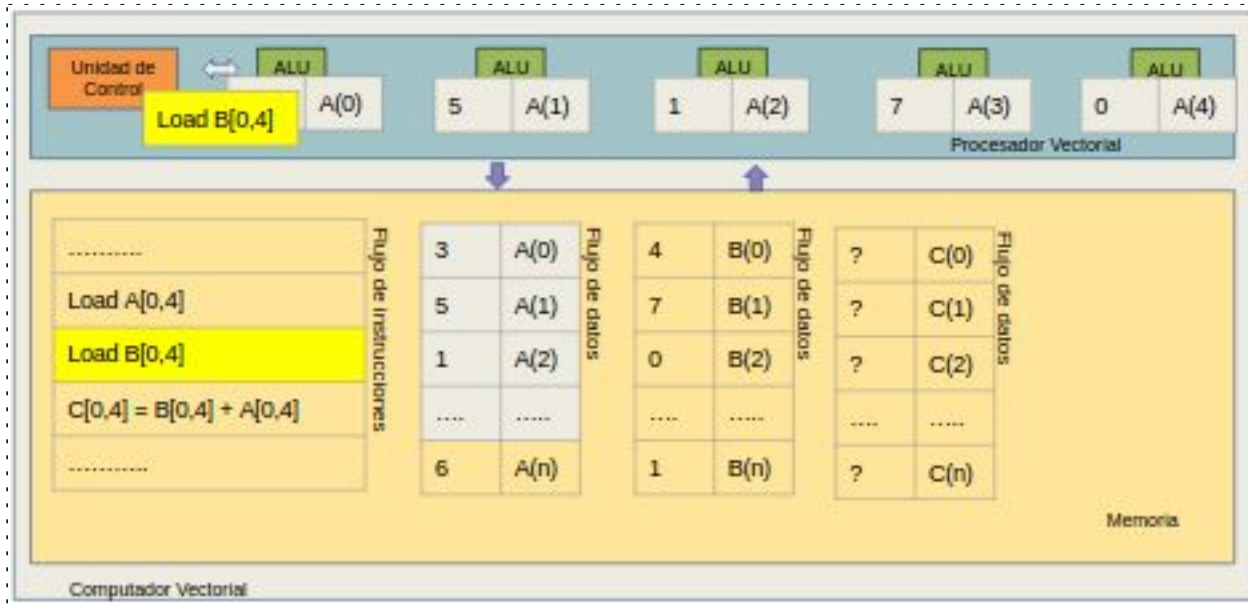
- Es un computador paralelo.
- Se caracteriza por operar sobre un flujo de instrucciones y múltiples flujos de datos en el tiempo.





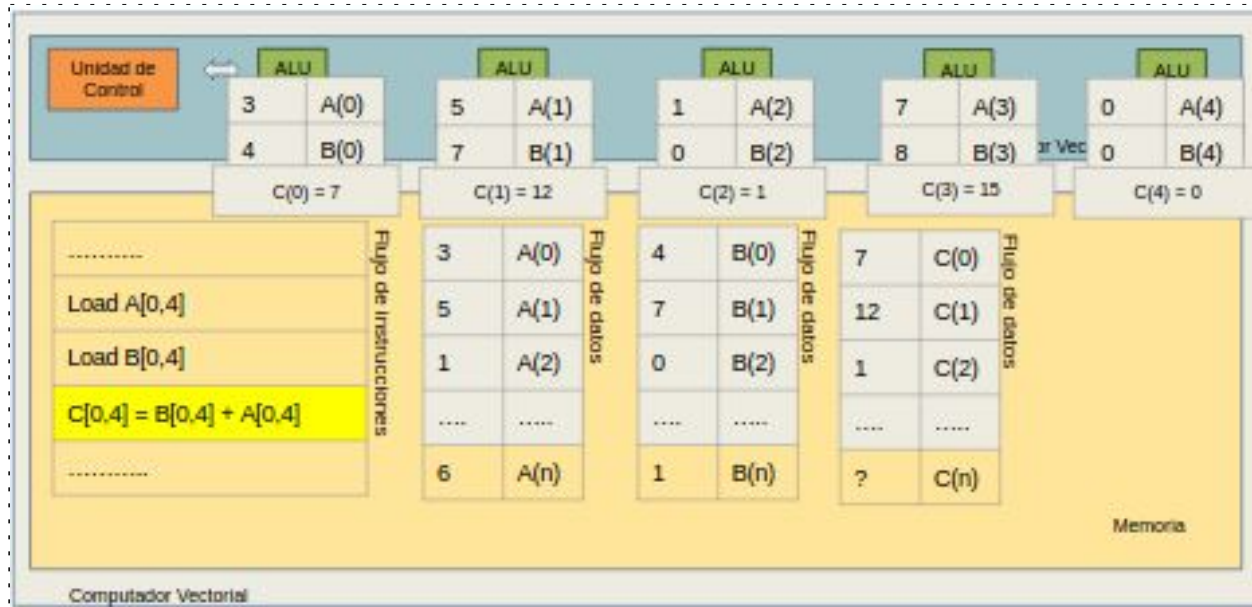






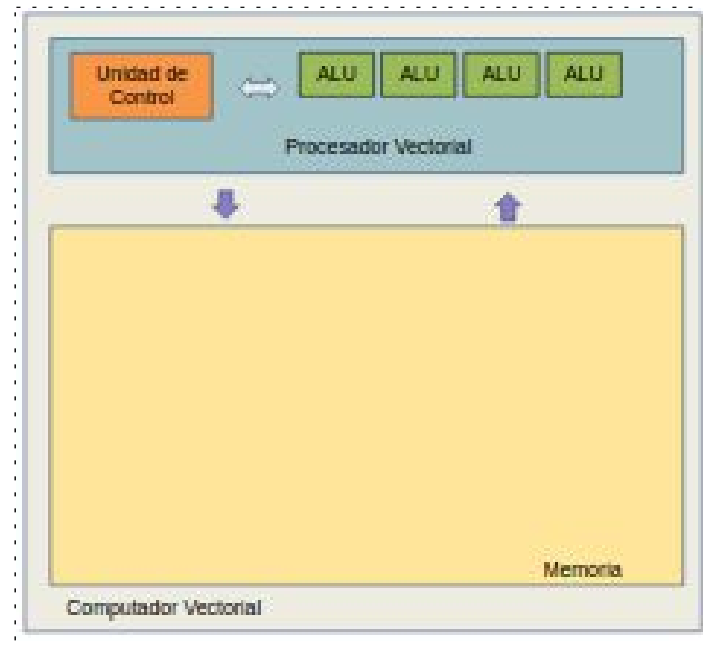






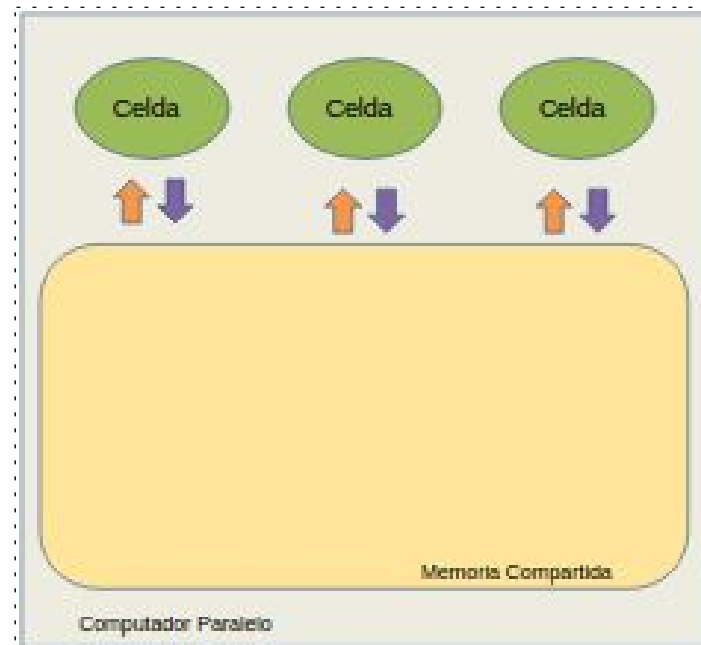
SIMD (Single Instruction Stream, Multiple Data Stream)

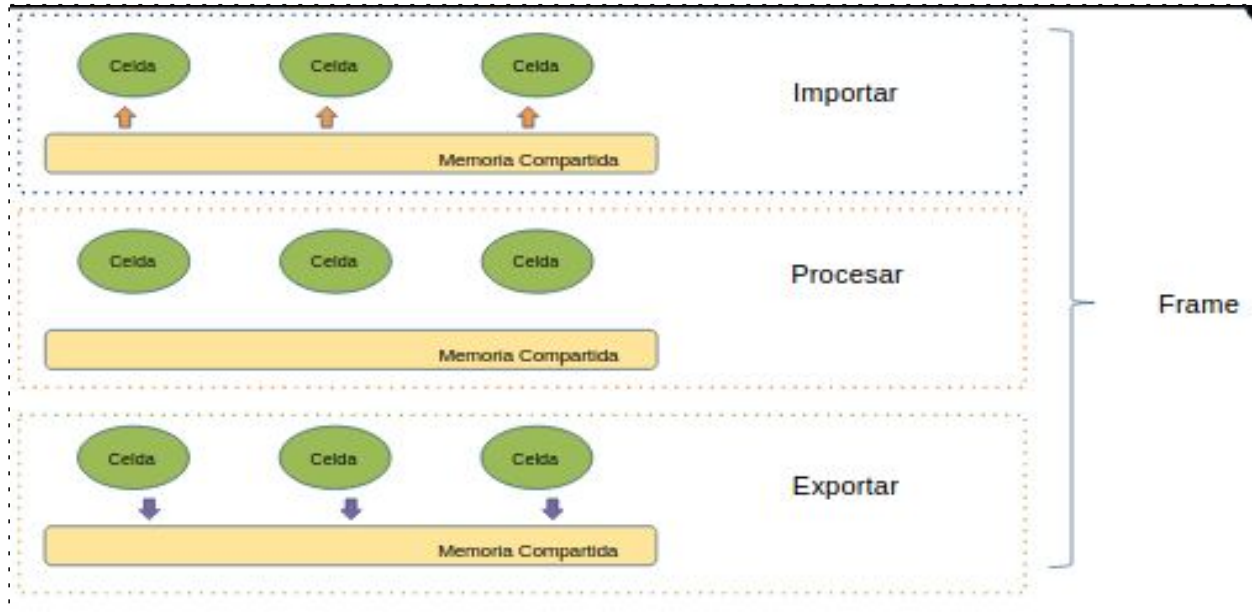
- Fueron diseñados para **acelerar el procesamiento de procesar vectores y matrices**.
- Son esenciales para el procesamiento de imágenes.
- Una sola instrucción implica una gran cantidad de trabajo.

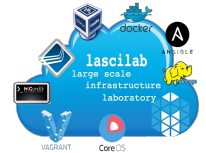


MISD (Multiple Instruction Stream, Single Data Stream)

- Es un computador paralelo.
- Este computador se componen de varias unidades de procesamiento llamadas **celdas**.
- Cada celda ejecuta un programa (**tile**).
- Todos los programas **comparten el mismo flujo de datos**.





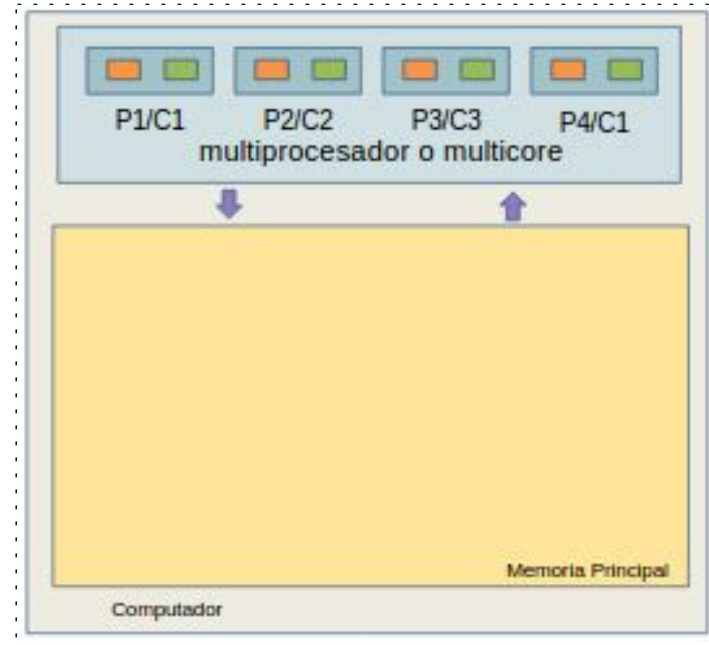


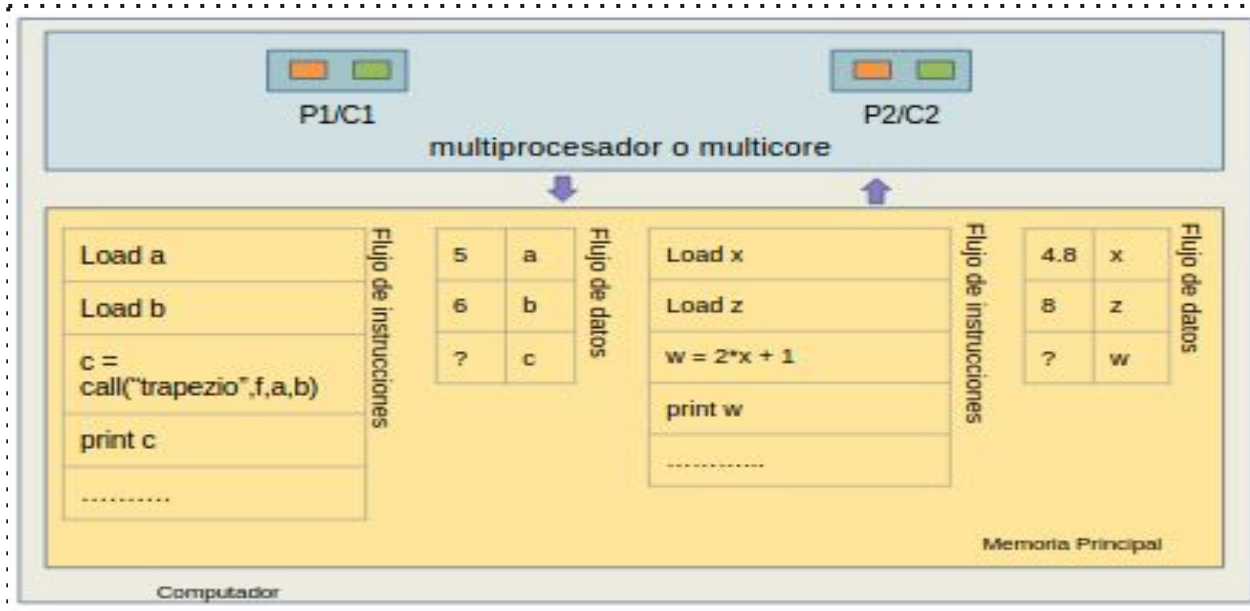
MISD (Multiple Instruction Stream, Single Data Stream)

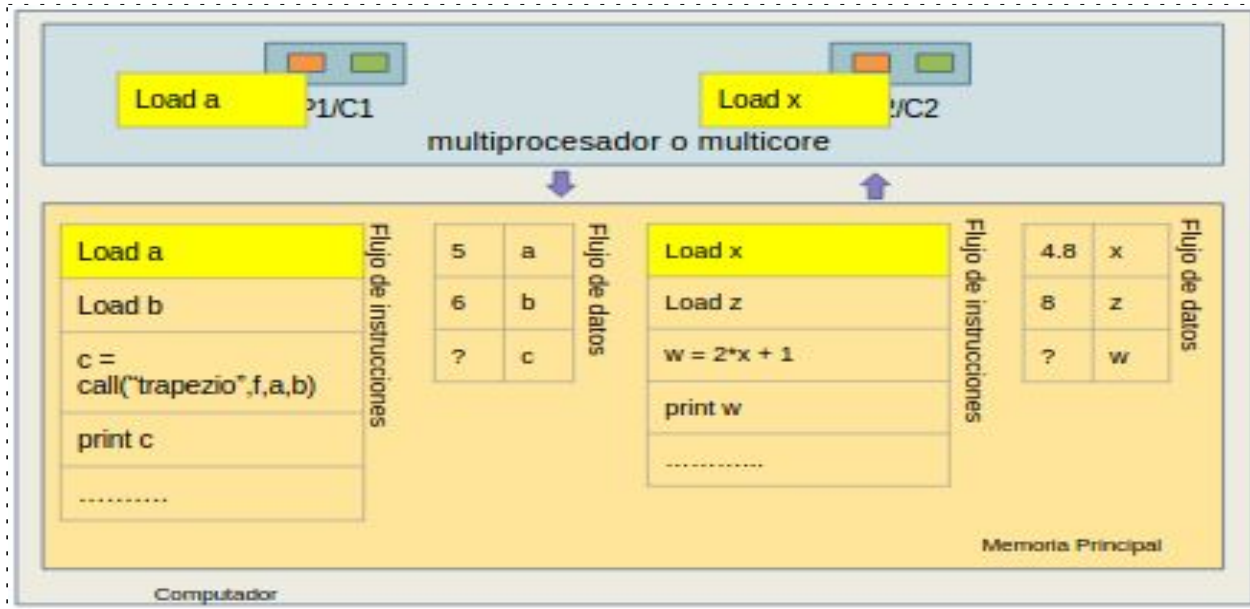
- Se ideó con el objetivo de llevar a cabo simulaciones.
- Puede llevar a cabo múltiples simulaciones en paralelo, donde estas comparten los datos de entrada.
- Puede ser utilizado para descifrar contraseñas usando múltiples algoritmos de descifrado.
- Puede ser usado en ambientes de IA donde el ambiente está determinado por los datos de entrada y cada programa representa un agente cuyo comportamiento se encuentra programado.
- Permite simular líneas de producción (pipelines), por ejemplo de automóviles, donde la salida de una sección (celda) es la entrada de otra sección de la línea de producción.

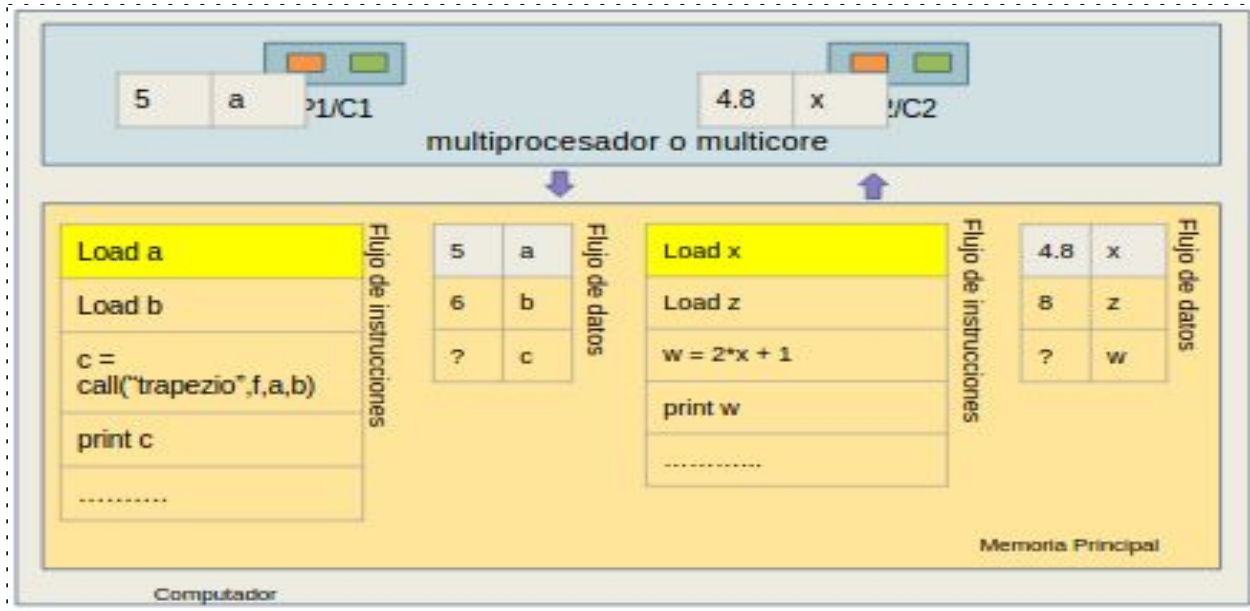
MIMD (Multiple Instruction Stream, Multiple Data Stream)

- Es un computador paralelo.
- Arquitectura de los computadores actuales (Celulares, Workstations, Clusters, etc).
- Varios flujos de datos, y varios flujos de instrucciones son ejecutados en el tiempo.

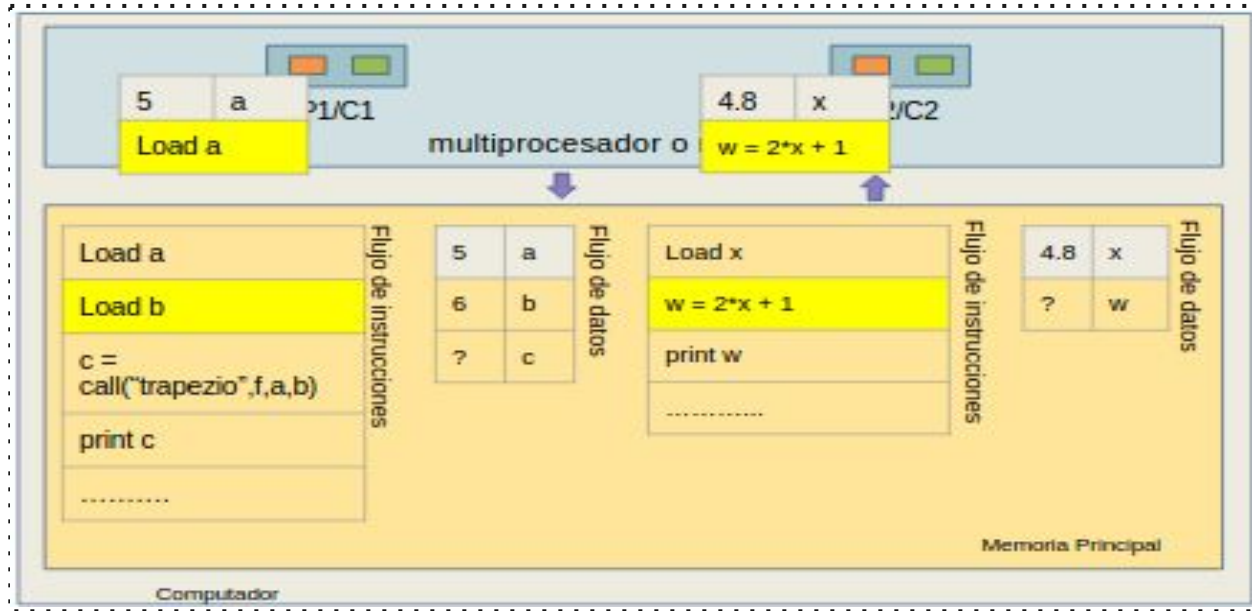




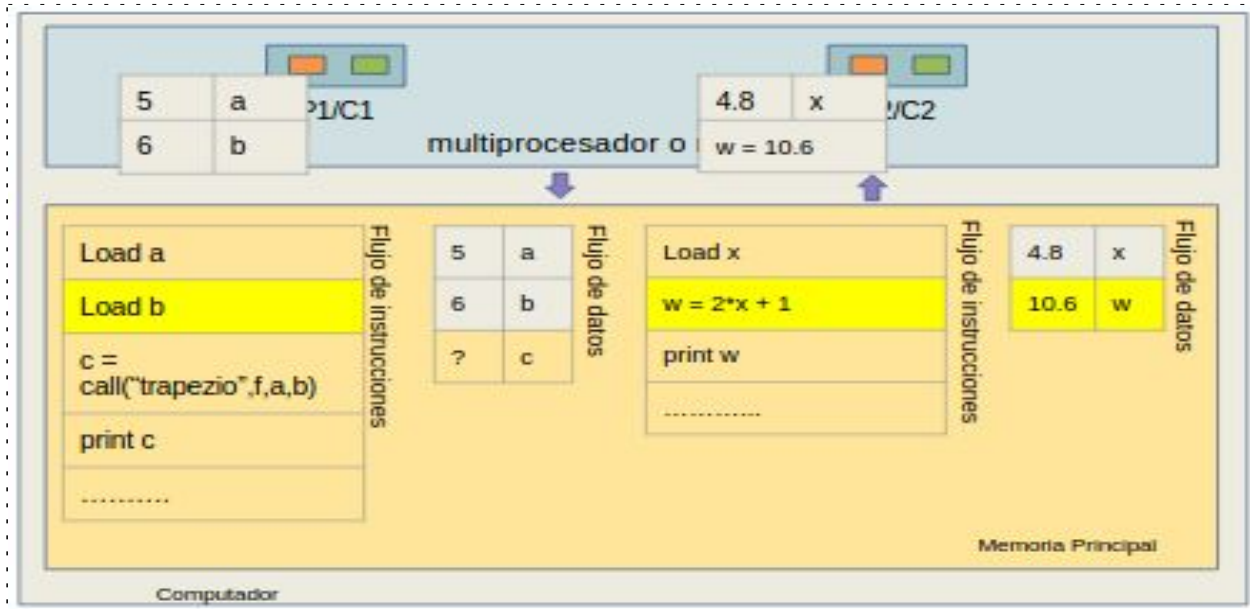




MIMD (Multiple Instruction Stream, Multiple Data Stream)



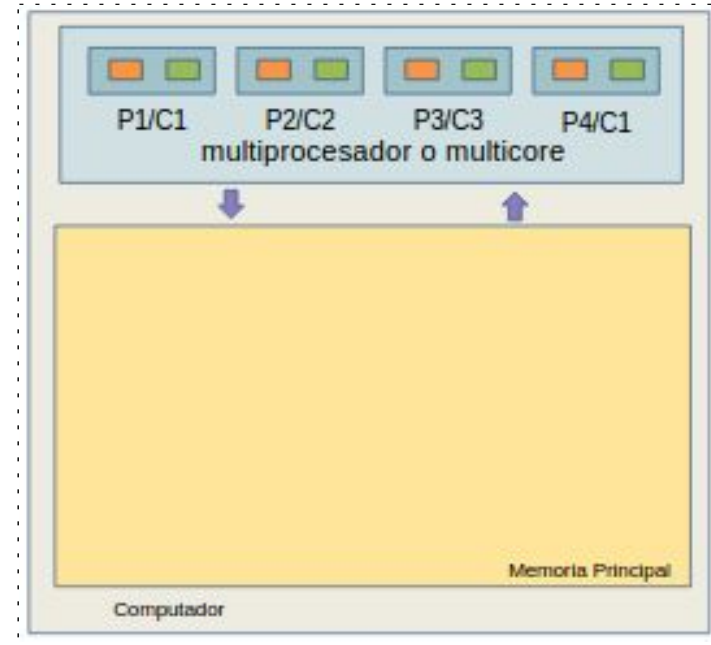
MIMD (Multiple Instruction Stream, Multiple Data Stream)

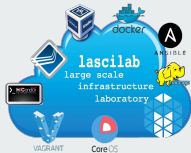


MIMD (Multiple Instruction Stream, Multiple Data Stream)

MIMD (Multiple Instruction Stream, Multiple Data Stream)

- Son computadores de propósito general.
- Su amplia gama de acción ha permitido su estabilidad en el mercado.
- Aplican funcionalidades de los **SISD**, **SIMD**, **MISD**.

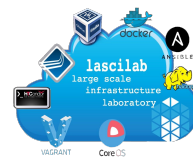




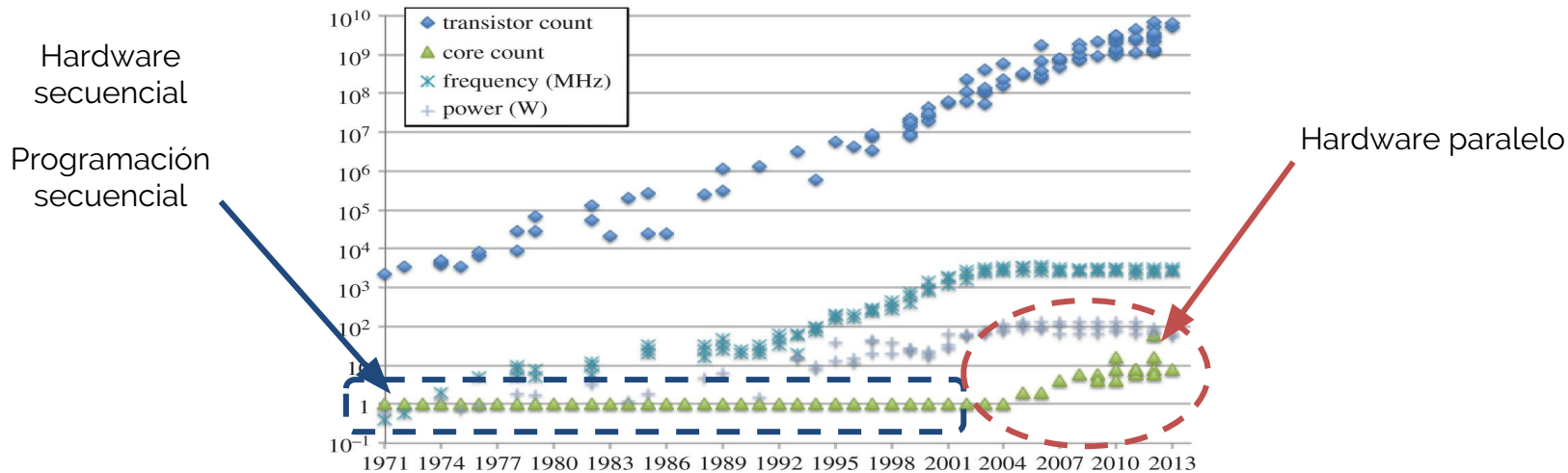
Lo que aprenderás hoy

¿Cómo es llevado a cabo el paralelismo un computador?

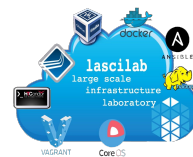
- ¿Por qué la Computación Paralela es importante?
- ¿Qué es la Computación Paralela?
- ¿Por qué surge?
- Posible solución “Computación Paralela”
- Perspectiva de Paralelismo en la Historia.
- Aplicaciones
- Conclusiones
- Resumen



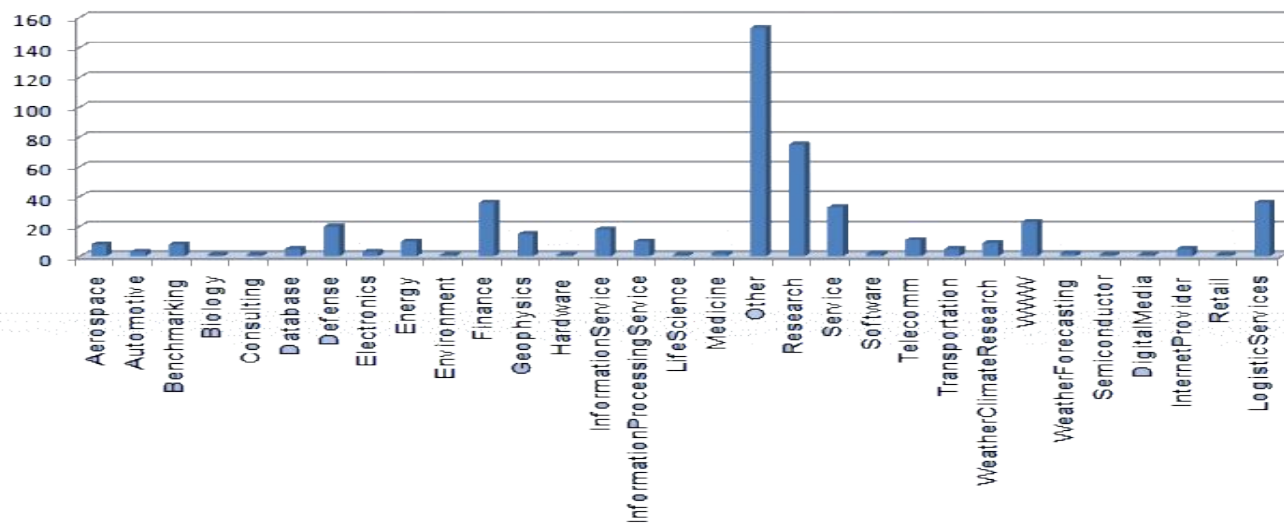
Perspectiva de Paralelismo en la Historia

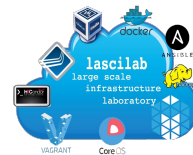


¿Dónde estamos?



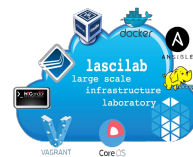
Aplicaciones





Conclusiones

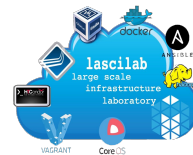
- Los Clusters computacionales hoy en día se componen de computadores:
 - SIMD
 - MIMD
 - Computadores híbridos CPU/GPU
- Las estaciones de trabajo también poseen en su interior procesadores SIMD, MIMD, y son llamados computadores híbridos CPU/GPU.
- Para programar computadores paralelos es necesario usar lenguajes que soporten el paralelismo.
 - Python - Threads, OpenMP, OpenMPI
 - Fortran - OpenMP
 - C/C++ - Pthreads, OpenMPI, OpenMP
 - R ...



Resumen

- Computación Paralela:
 - Es una forma de cómputo en la que muchas instrucciones se llevan a cabo simultáneamente.
- ¿Por qué surge?
 - Era de la Computación Secuencial.
 - Velocidad del reloj del procesador.
 - Aumento en el número de transistores.
 - Paralelismo a nivel de instrucción.
 - Memoria caché.
- Tendencia en el diseño de los procesadores





Resumen

- Posible Solución
 - Era de la Computación Paralela
 - SISD
 - SIMD
 - MISD
 - MIMD
- Perspectiva del Paralelismo en la Historia
- Aplicaciones
- Conclusiones



A continuación ...



Aurelio Vivas

Introducción a la Programación
Paralela en OpenMP y
OpenACC



```
#include <omp.h>
```

```
int main(void){
    #pragma omp parallel // or pragma acc parallel
    { // Start Parallel Region
        // some code .. to be executed in parallel
    } // End Parallel Region
    return 0;
}
```

