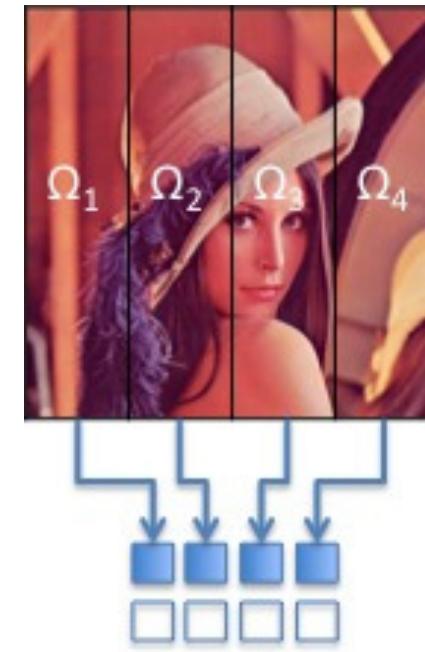
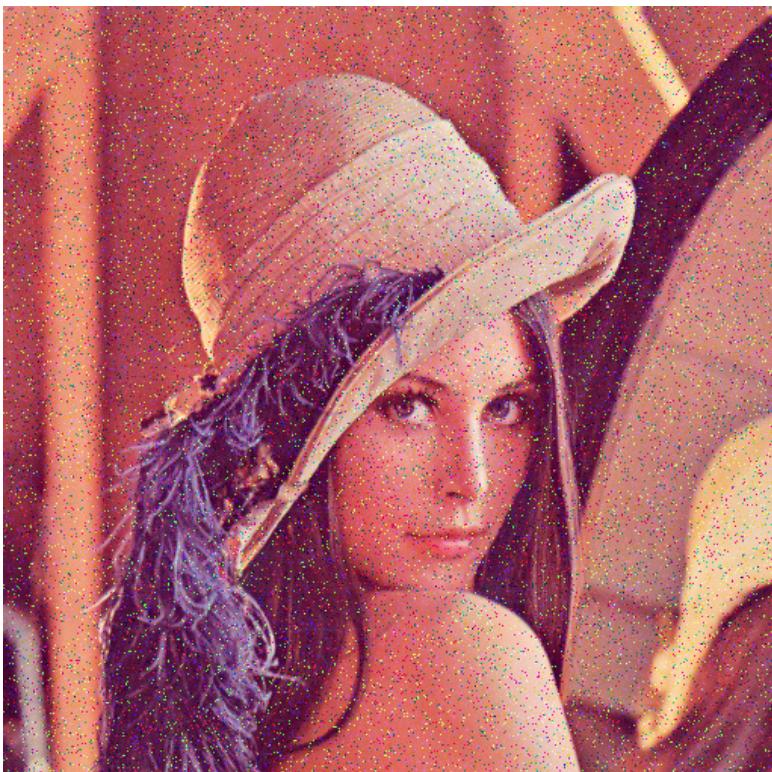


DESARROLLO DE SOFTWARE EN ARQUITECTURAS PARALELAS

1. Motivación y aspectos de la programación paralela.
2. Tipos de sistemas paralelos. Paradigmas de programación paralela.
 - Tipos de arquitecturas paralelas.
 - Paradigmas de programación paralela.
3. Conceptos básicos y medidas de paralelismo.
4. Diseño de programas paralelos.
5. La interface de paso de mensaje: el estándar MPI.
6. Paralelización de algoritmos: ejemplos y aplicaciones.

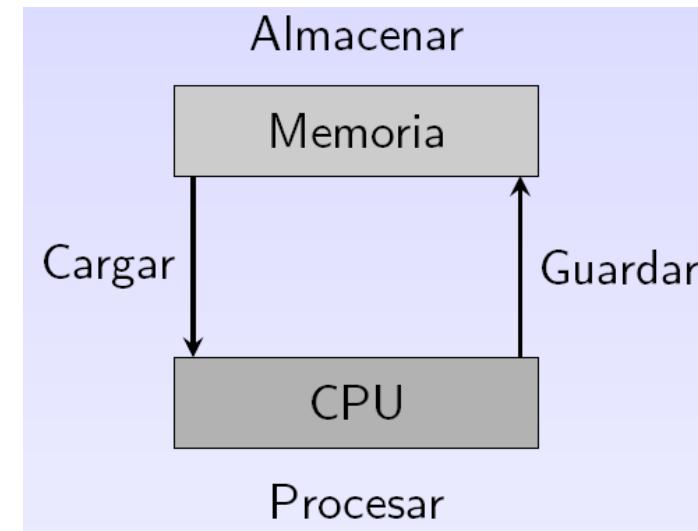
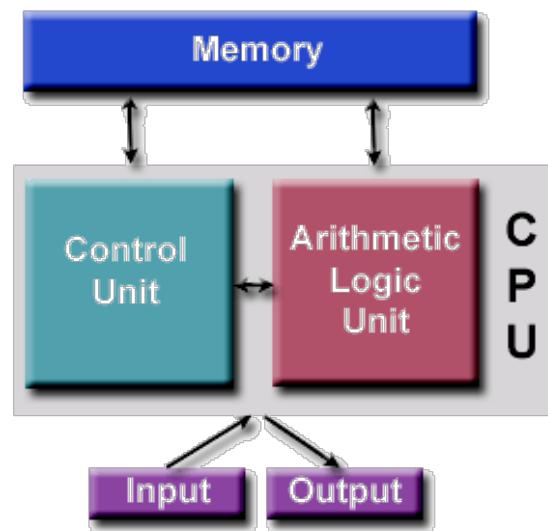
Desarrollo de Software en Arquitecturas Paralelas



Arquitectura paralela

Tipos de arquitecturas paralelas

- En la **arquitectura de von Neumann** todos los ordenadores siguen el mismo patrón:
 - La memoria almacena el programa y los datos.
 - El programa son instrucciones a seguir por la CPU.
 - Los datos son información a utilizar por el programa.
 - La CPU carga los datos, los procesa según el programa y luego los guarda otra vez en memoria.



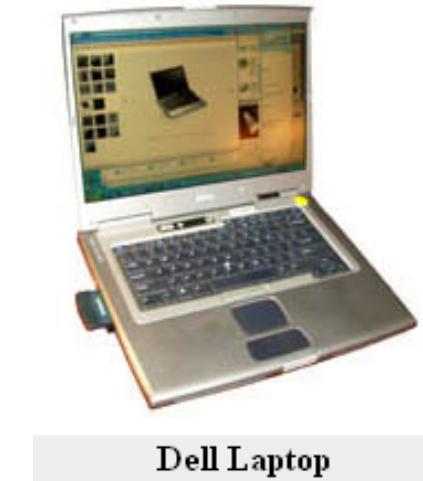
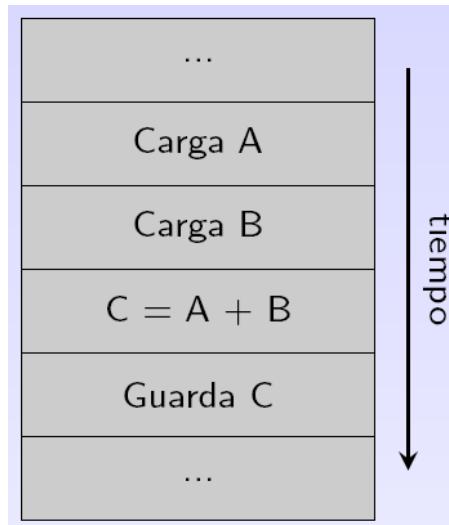
Tipos de arquitecturas paralelas

- La Taxonomía de Flynn es la clasificación más extendida del paralelismo:
 - Distingue entre instrucciones y datos.
 - Estos pueden ser simples o múltiples.

		Datos	
		Simples	Múltiples
Instrucciones	Simples	SISD	SIMD
	Múltiples	MISD	MIMD

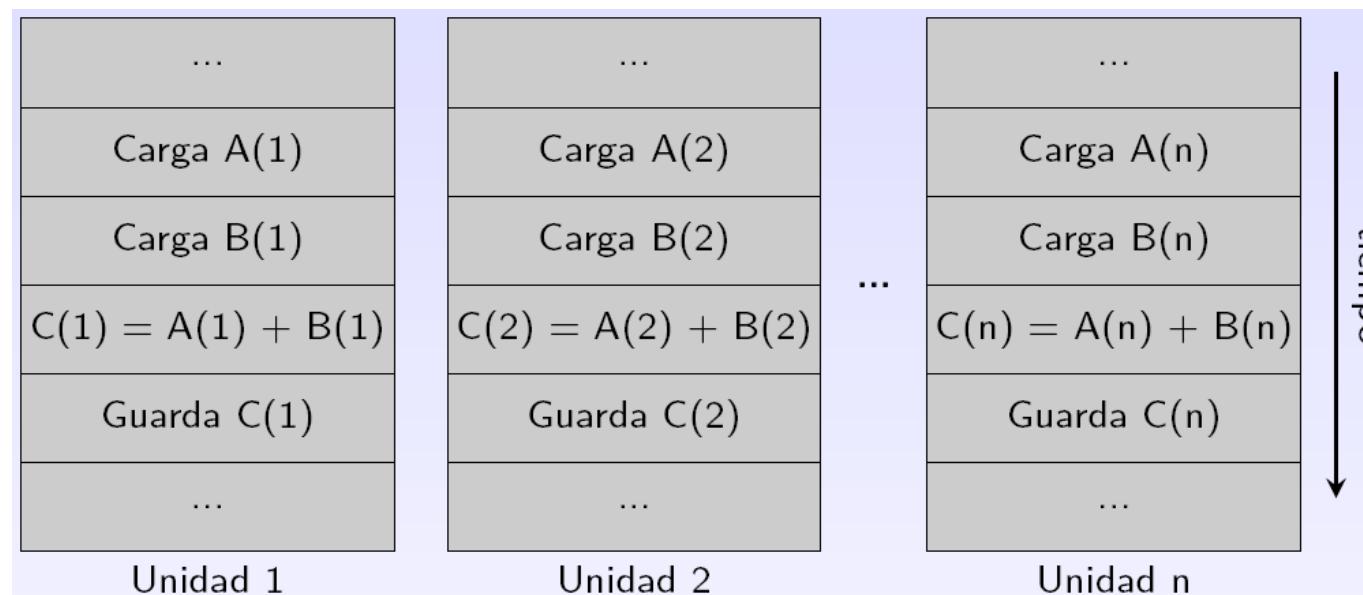
Tipos de arquitecturas paralelas

- SISD (**S**ingle **I**nstruction, **S**ingle **D**ata):
 - La CPU procesa únicamente una instrucción por cada ciclo de reloj.
 - Únicamente un dato se procesa en cada ciclo de reloj.
 - Es el modelo más antiguo de ordenador.
 - Ejemplo: Antiguas generaciones de ordenadores centrales, la primeros PCs.



Tipos de arquitecturas paralelas

- SIMD (**S**ingle **I**nstruction, **M**ultiple **D**ata):
 - Todas las unidades ejecutan la misma instrucción.
 - Cada unidad procesa un dato distinto.
 - Todas las unidades operan simultáneamente.
 - Varios procesadores con una única unidad de control.



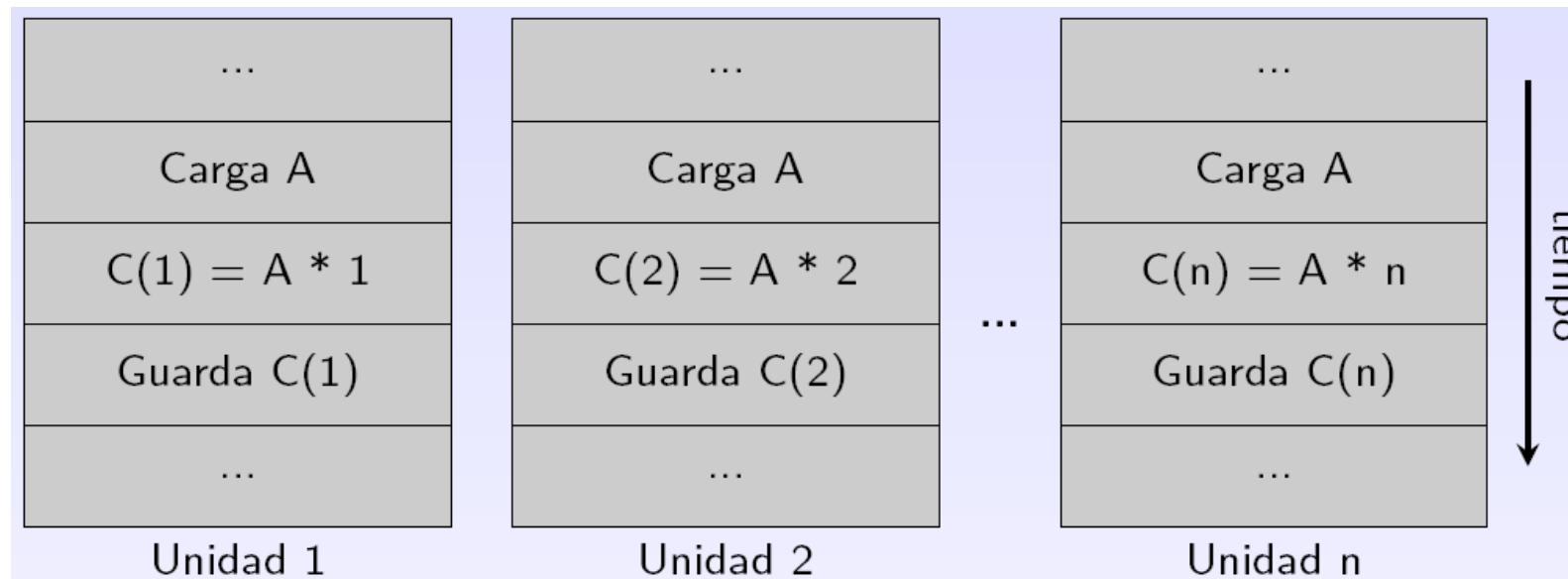
Tipos de arquitecturas paralelas

- SIMD (Single Instruction, Multiple Data):
 - Ejemplos:
 - Processor Arrays: Connection Machine CM-2, MasPar MP-1 & MP-2, ILLIAC IV.
 - Vector Pipelines: IBM 9000, Cray X-MP, Y-MP & C90, Fujitsu VP, NEC SX-2, Hitachi S820, ETA10.
 - Computadores actuales, particularmente los que usan GPUs.



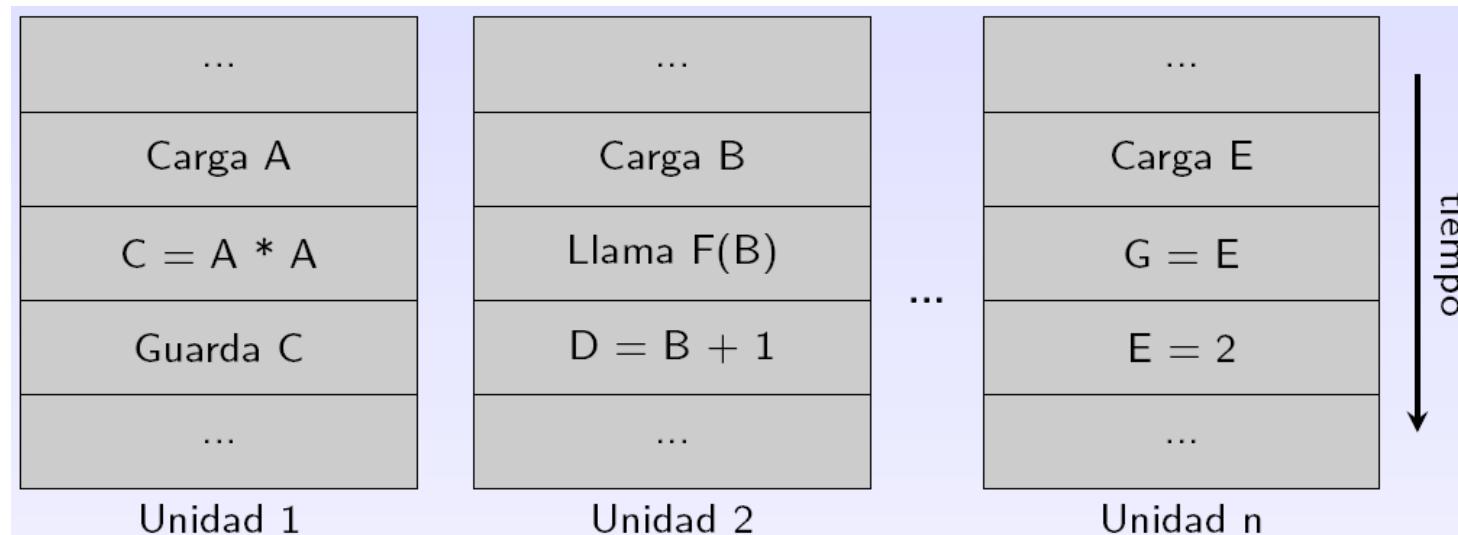
Tipos de arquitecturas paralelas

- MISD (**M**ultiple **I**nstruction, **S**ingle **D**ata):
 - Cada unidad ejecuta una instrucción distinta.
 - Cada unidad procesa el mismo dato.
 - Aplicación muy limitada en la vida real.



Tipos de arquitecturas paralelas

- **MIMD (Multiple Instruction, Multiple Data):**
 - Cada unidad ejecuta una instrucción distinta.
 - Cada unidad procesa un dato distinto.
 - Todas las unidades operan simultáneamente.
 - Varios elementos de proceso y cada uno con su propia unidad de control. También denominados *Multiprocesadores*.



Tipos de arquitecturas paralelas

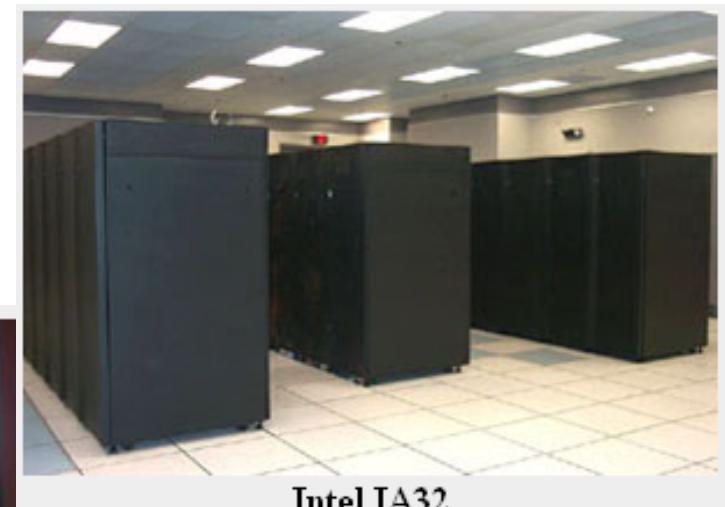
- MIMD (**M**ultiple **I**nstruction, **M**ultiple **D**ata):
 - Ejemplos: la mayoría de los actuales supercomputadores, clusters de computadores, multicore.



IBM POWER5



Cray XT3



Intel IA32

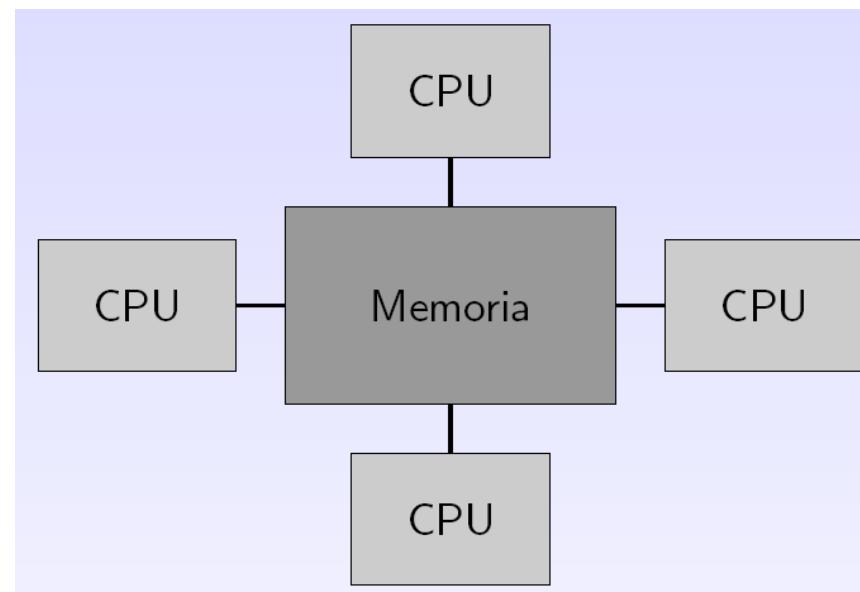
Tipos de arquitecturas paralelas

La clasificación más habitual de los multiprocesadores es atendiendo a la distribución de la memoria:

- Multiprocesadores de memoria compartida.
- Multiprocesadores de memoria distribuida.
- Multiprocesadores híbridos.

Tipos de arquitecturas paralelas

- Multiprocs. de memoria compartida (características):
 - Todos los elementos de proceso tienen acceso a una memoria común.
 - Cambios en la memoria son visibles por todas las CPUs.
 - Cada uno puede tener una memoria local.
 - La comunicación se realiza a través de la memoria común.



Tipos de arquitecturas paralelas

- Multiprocesadores de memoria compartida:

- Ventajas:

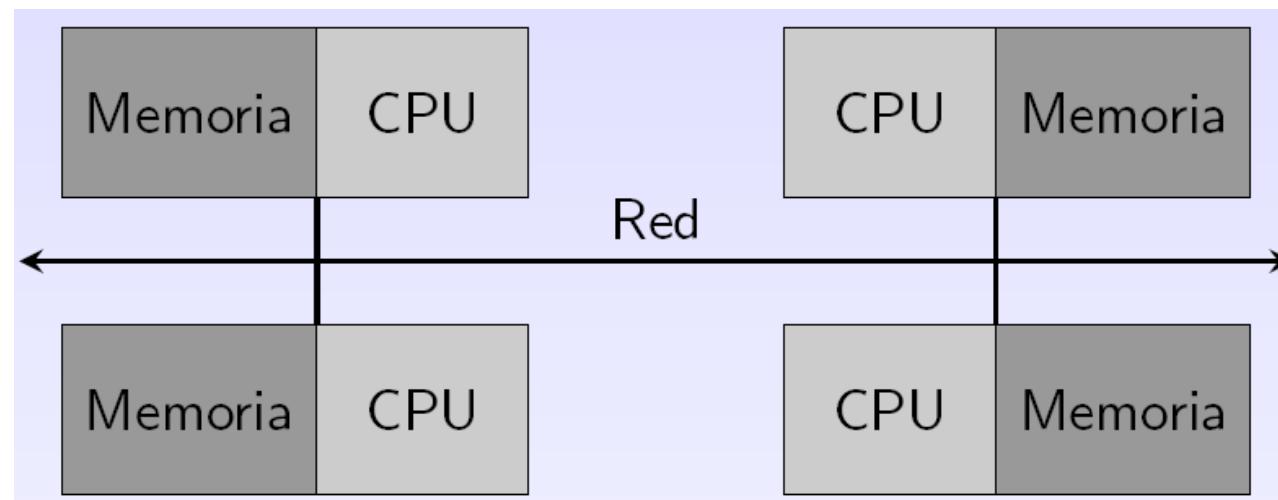
- Conceptualmente fáciles de programar al disponer de un espacio global de direcciones de memoria.
 - Compartir datos entre tareas es muy fácil y rápido debido a la proximidad de memoria y CPU.

- Desventajas:

- La escalabilidad entre CPUs y memoria es mala.
 - El programador es responsable de la sincronización para asegurar un correcto acceso a la memoria global.
 - Es muy costoso hacer ordenadores con muchas CPUs.

Tipos de arquitecturas paralelas

- Multiprocs. de memoria distribuida (características):
 - Cada procesador tiene su propia memoria, inaccesible a los demás procesadores.
 - No existe memoria común.
 - La comunicación se efectúa mediante paso de mensajes.

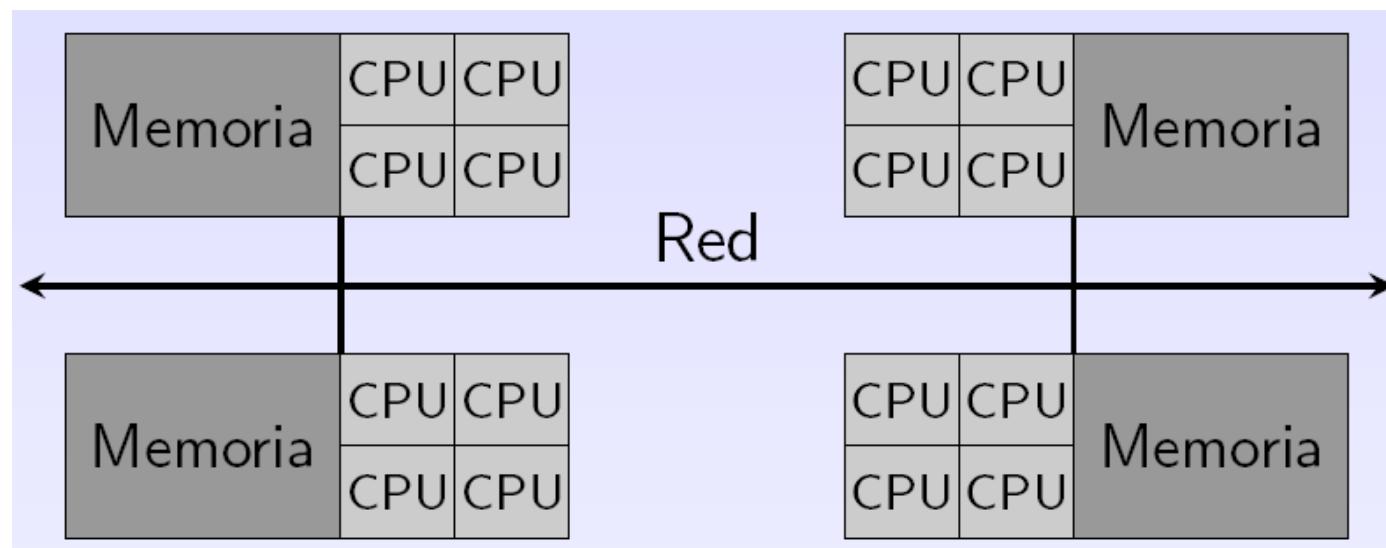


Tipos de arquitecturas paralelas

- Multiprocesadores de memoria distribuida:
 - Ventajas:
 - La escalabilidad entre CPUs y memoria es muy buena.
 - Acceso rápido y en exclusiva a la memoria.
 - El coste es “lineal” con el número de CPUs.
 - Desventajas:
 - El programador es responsable de las comunicaciones.
 - La conversión de programas con acceso a una memoria global puede no ser trivial.
 - La red de comunicaciones suele ser el cuello de botella.

Tipos de arquitecturas paralelas

- Multiprocesadores híbridos (características)
 - Grupos de CPUs comparten una misma memoria.
 - Los grupos de CPUs se comunican por una red.
 - Suelen ser máquinas SMP (Symmetric multiprocessing) interconectadas entre sí.



Tipos de arquitecturas paralelas

- Multiprocesadores híbridos:

- Ventajas:

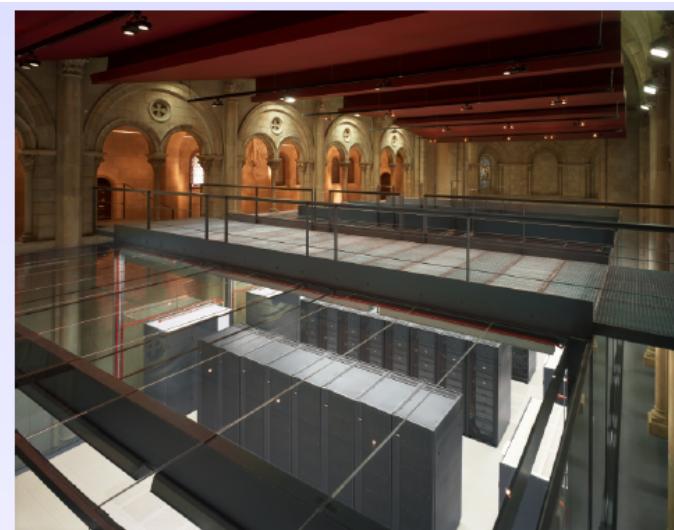
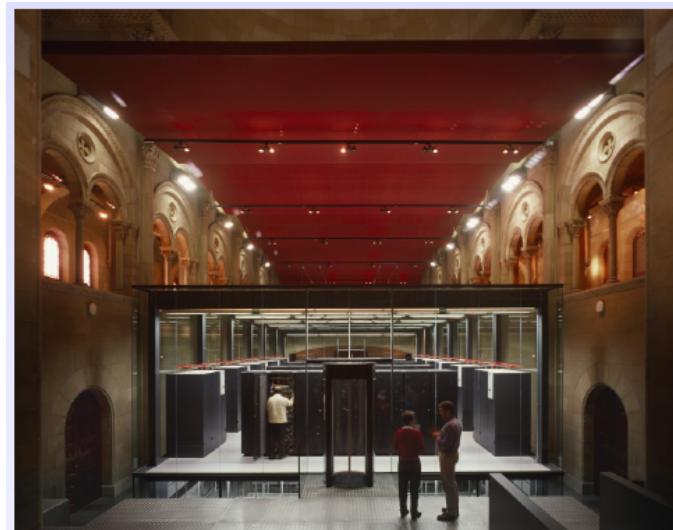
- La escalabilidad entre CPUs y memoria es razonable.
 - El coste es “lineal” con el número de grupos de CPUs.
 - La **red** de comunicaciones es **menos crítica**.

- Desventajas:

- El programador es responsable de las comunicaciones.
 - El programador es responsable de la sincronía.
 - La conversión de programas seriales puede no ser trivial.

Tipos de arquitecturas paralelas

- Multiprocesadores híbridos (ejemplo):
 - **Mare Nostrum I (España 2004): primer gran supercomputador en España y el más potente de Europa cuando se adquirió**
 - 2282 nodos SMP con 2 CPUs por nodo.
 - 4GB de memoria por nodo.
 - Mare Nostrum IV: 34 millones de euros.
 - Instalación 2017.



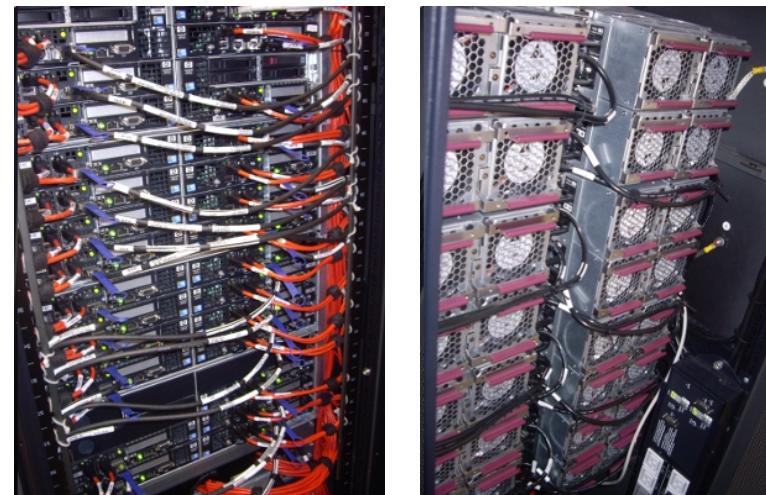
Tipos de arquitecturas paralelas

- Multiprocesadores híbridos (ejemplo):
 - **Mare Nostrum IV (Datos Noviembre 2019):**
 - 30 en Top 500
 - 153.216 cores



Tipos de arquitecturas paralelas

- Multiprocesadores híbridos (ejemplo):
- **Cluster de Computación – IUII** (Universidad de Alicante, España):
 - 26 nodos de cálculo:
 - 2 procesadores Intel XEON X5660 hexacore.
 - 48 GB de memoria por nodo.
 - Total: 312 CPUs y 1248 GB de memoria.
 - 3 GPU HP Tesla M2050 (3 GB) + 1 GPU NVidia Tesla 2070 (6 GB).



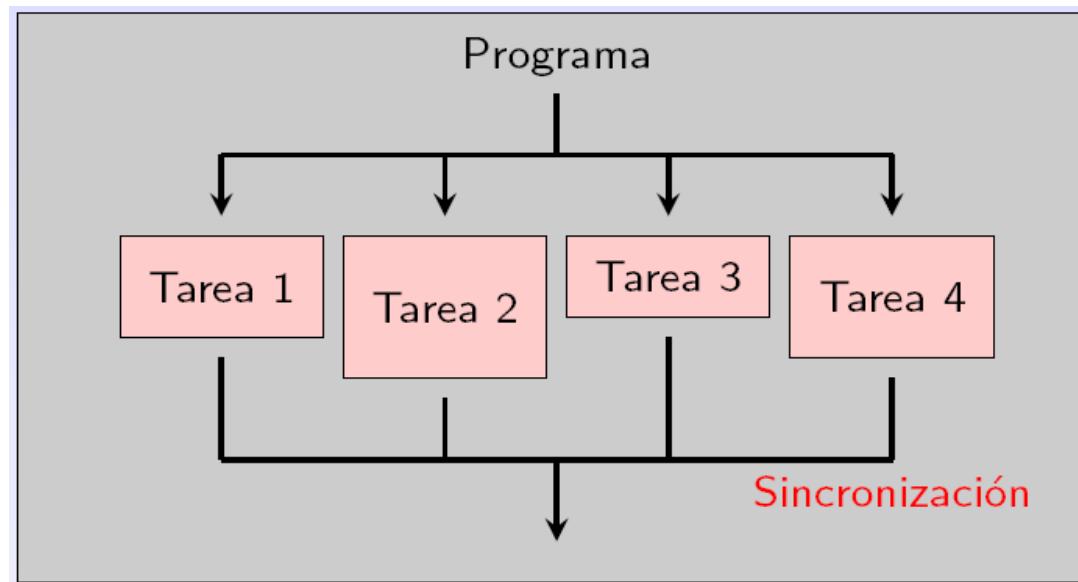
Paradigmas de programación paralela

- Existen varias formas de programar en paralelo. Las más importantes son:
 - Modelo de tareas (OpenMP).
 - Paralelismo de datos.
 - Paso de mensajes (MPI).

Paradigmas de programación paralela

○ Modelo de tareas (características):

- Un programa secuencial define un conjunto de tareas.
- Cada tarea dispone de su memoria local.
- Cada tarea tiene acceso a una memoria conjunta.
- Las tareas se ejecutan simultáneamente.
- El programador es responsable de la sincronización.



Paradigmas de programación paralela

- Modelo de tareas (El estándar OpenMP):
 - **OpenMP** (Open specifications for Multi Processing) es una API (Application Programming Interface) para la paralelización de programas en memoria compartida.
 - Lo incorporan la mayoría de compiladores actuales.
 - Paralelización de:
 - Bucles: paralelismo de “grano fino”.
 - Regiones paralelas: paralelismo de “grano grueso”



Paradigmas de programación paralela

- Modelo de tareas (Ejemplo):

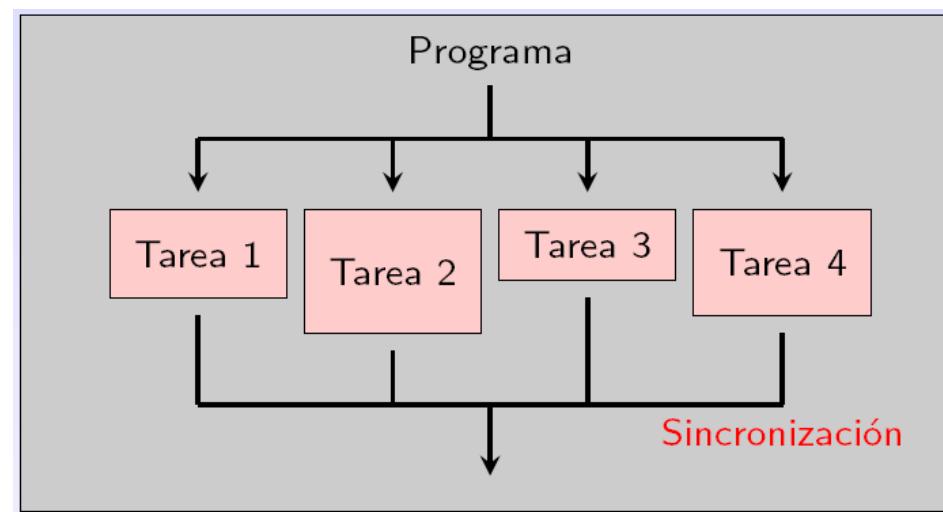
Realizar en paralelo el bucle para $i=1$ hasta 25

Tarea 1: $i=1$ hasta 5

Tarea 2: $i=6$ hasta 12

Tarea 3: $i=13$ hasta 20

Tarea 4: $i=21$ hasta 25

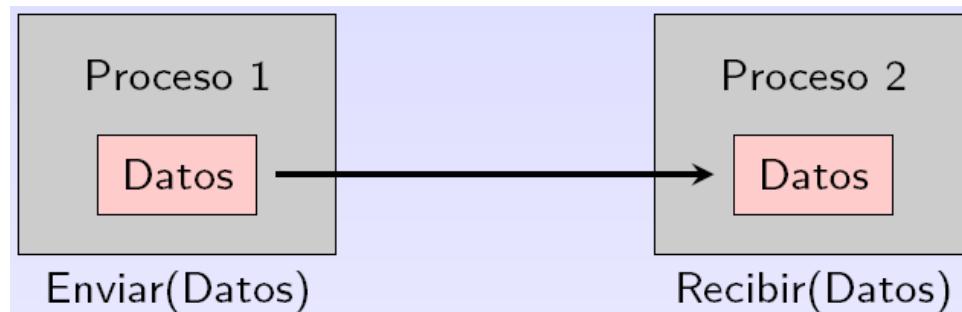


Paradigmas de programación paralela

- Paralelismo de datos (características):
 - Distintos procesos aplican las **mismas operaciones a distintos datos.**
 - Fácil de programar. Modificación del algoritmo secuencial.
 - El programador se encarga de indicar la **distribución de los datos.**
 - Menos eficiente que el paso de mensajes.
 - Aplicable a Memoria Compartida y Distribuida.
 - Técnicas (programación paralela SPMD):
 - **High Performance Fortran (HPF)**: Lenguajes de paralelismos de datos.
 - **Coarray Fortran**: un conjunto de extensiones de Fortran 95, <http://www.co-array.org/>.
 - **Unified Parallel C (UPC)**: extensión de C, <http://upc.lbl.gov/>.

Paradigmas de programación paralela

- Paso de mensajes (características):
 - Cada proceso tiene acceso sólo a sus datos locales.
 - Intercambio de información mediante el paso explícito de mensajes.
 - Difícil de programar. Muy eficiente.
 - Adecuado para Multiprocesadores con Memoria Distribuida.
 - El programador es responsable del envío y la recepción de mensajes (típicamente mediante llamadas a una librería).
 - Técnicas:
 - Librerías de paso de mensajes (p.e., PVM, MPI).



Paradigmas de programación paralela

- Paso de mensajes (Ejemplo):

ID = quien_soy_yo

Si ID=1 entonces
envio datos a 2 y espero confirmación
Si ID=2 entonces
recibo datos de 1 y envío confirmación

