

Arquitectura y Organización de Computadoras II

Sistemas Paralelos

MSc. Ing. Ticiano J. Torres Peralta
Ing. Pablo Toledo

2023 – Segundo Cuatrimestre



UNIVERSIDAD NACIONAL DE TUCUMÁN
facet
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍA

Paralelismo a Nivel Sistema

En el mundo de la computación paralela, las CPU (procesos/hilos) trabajan juntas en diferentes partes del mismo trabajo. Mientras hacen esto, hay mucha comunicación entre las partes para el intercambio de información. Cómo se logra esta comunicación de gran debate en la comunidad de arquitectura. Hay dos diseños distintos:

- Multiprocesadores
- Multicomputadoras

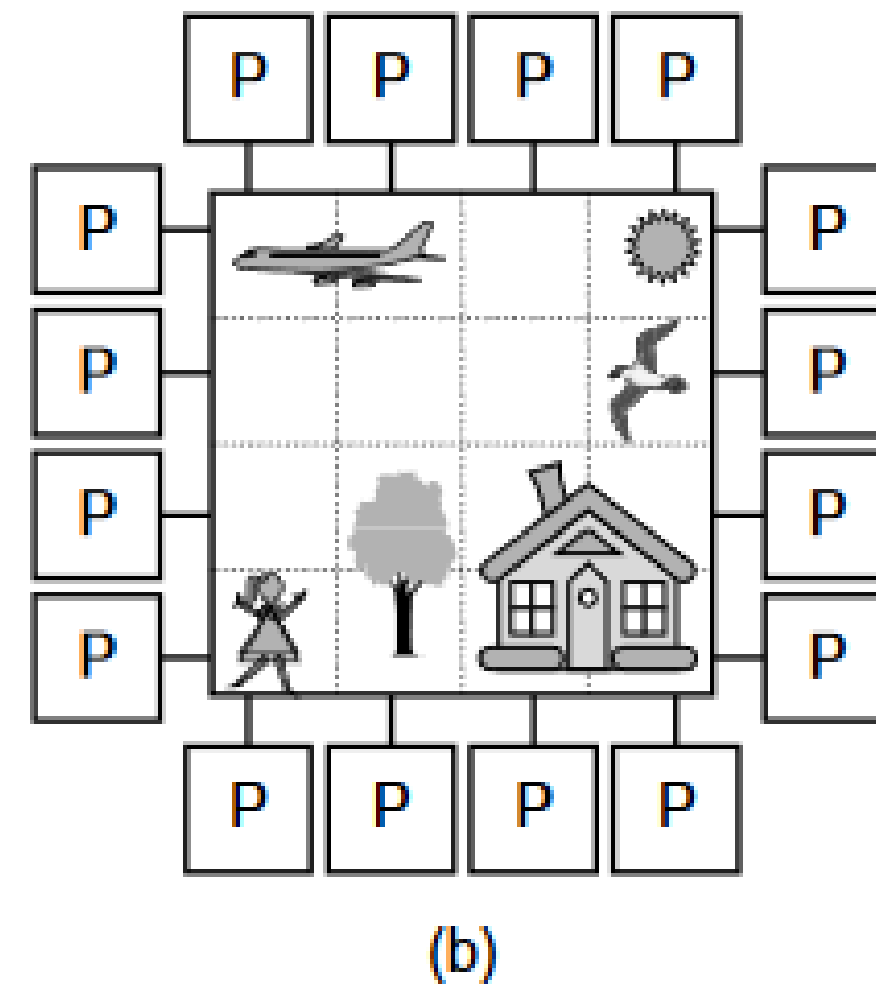
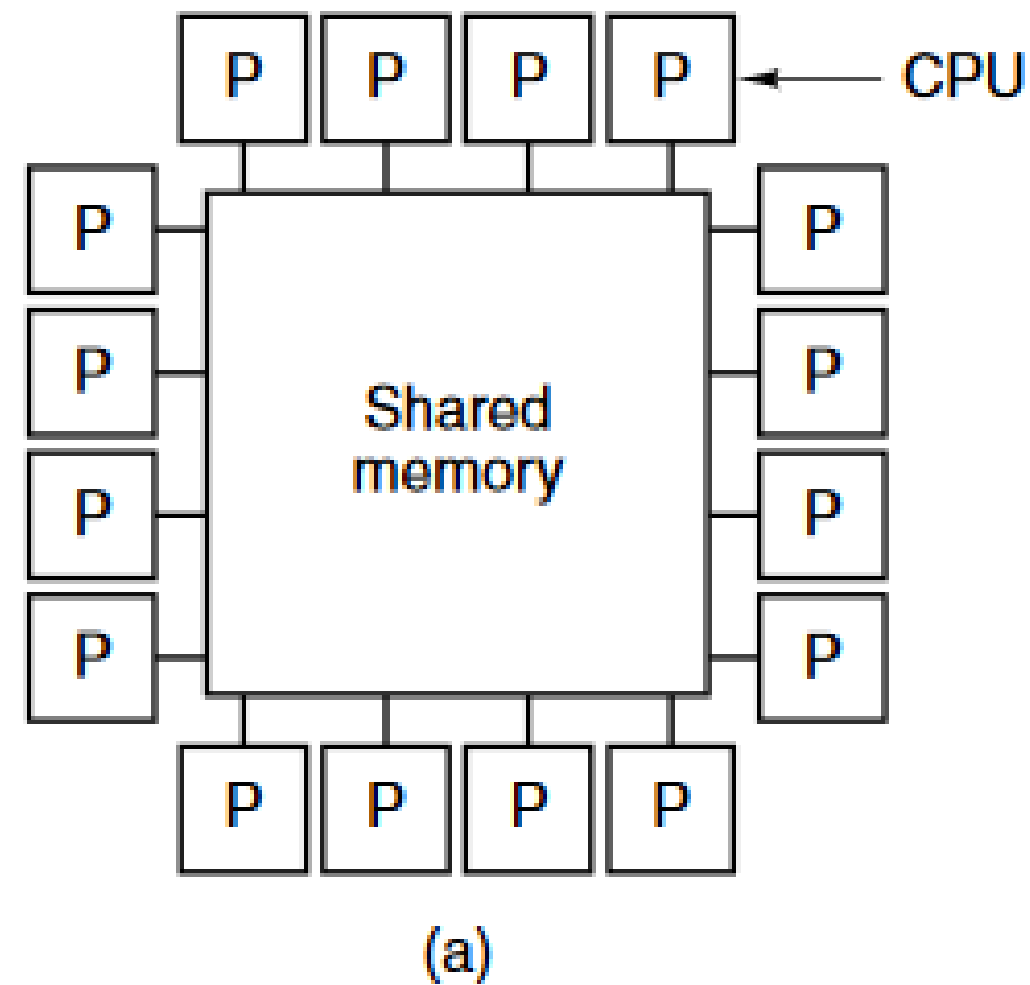
Las diferencias entre los dos diseños determina cómo se diseña, construye y programa el sistema.

Multiprocesadores

Una computadora paralela donde todas las CPU comparten una memoria común se llama multiprocesador. Aquí todos los procesos comparten un único espacio de direcciones virtuales, asignado a la memoria física común. En otras palabras, todo lo que necesitan hacer para acceder a la memoria es ejecutar una instrucción LOAD/STORE.

En este escenario, dos procesadores pueden comunicarse escribiendo datos en la memoria y leyéndolos de la misma. Los multiprocesadores son una arquitectura popular debido a su diseño sencillo. Es un modelo fácil de entender para los programadores y es aplicable a una amplia gama de problemas.

Multiprocesadores

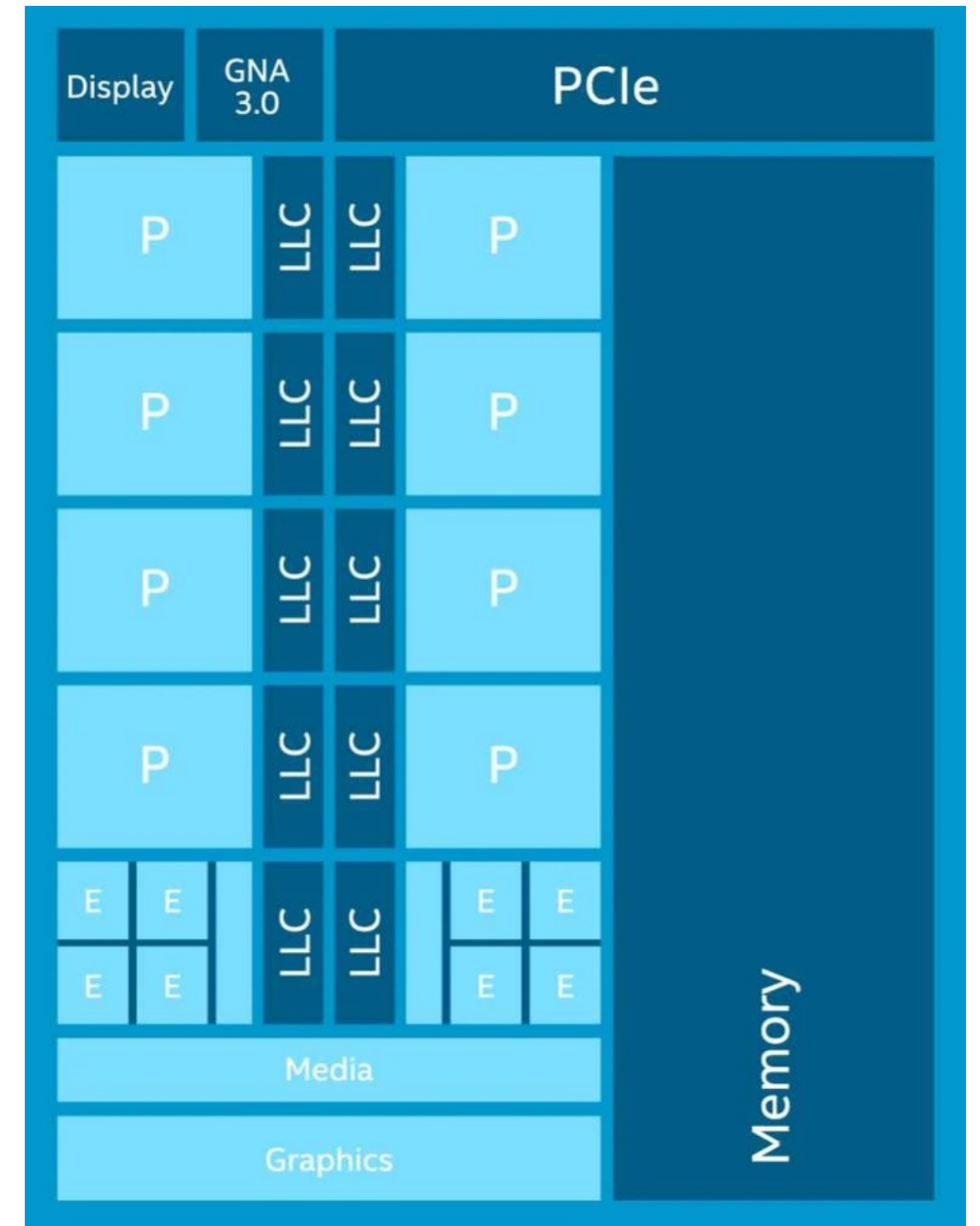
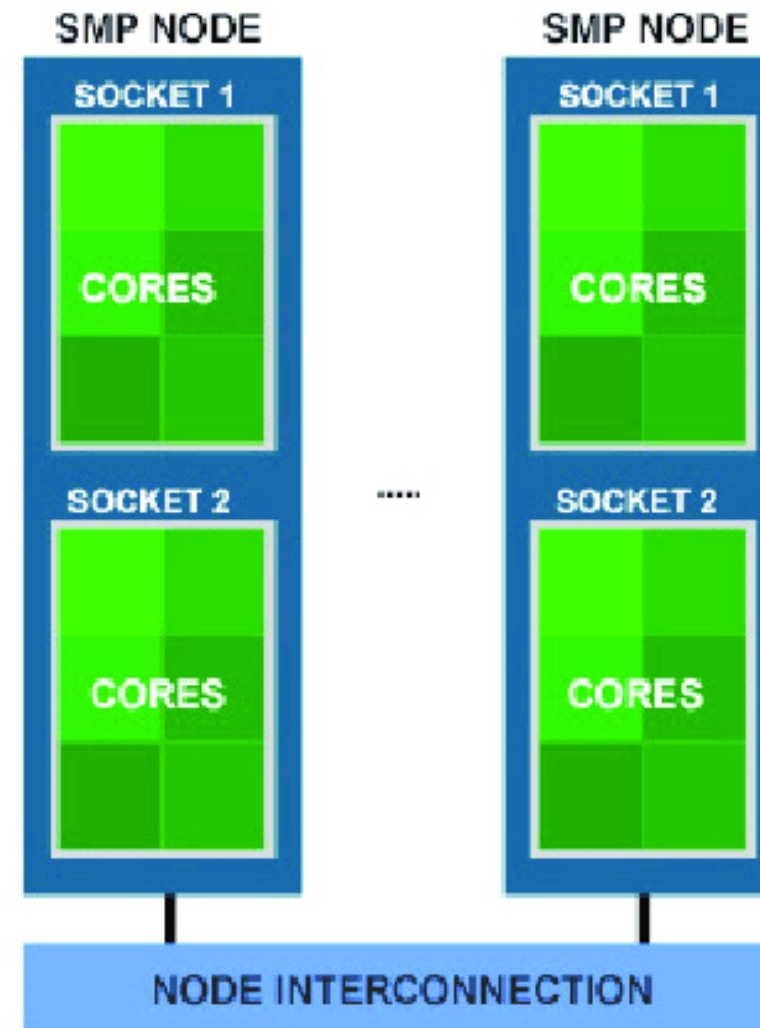
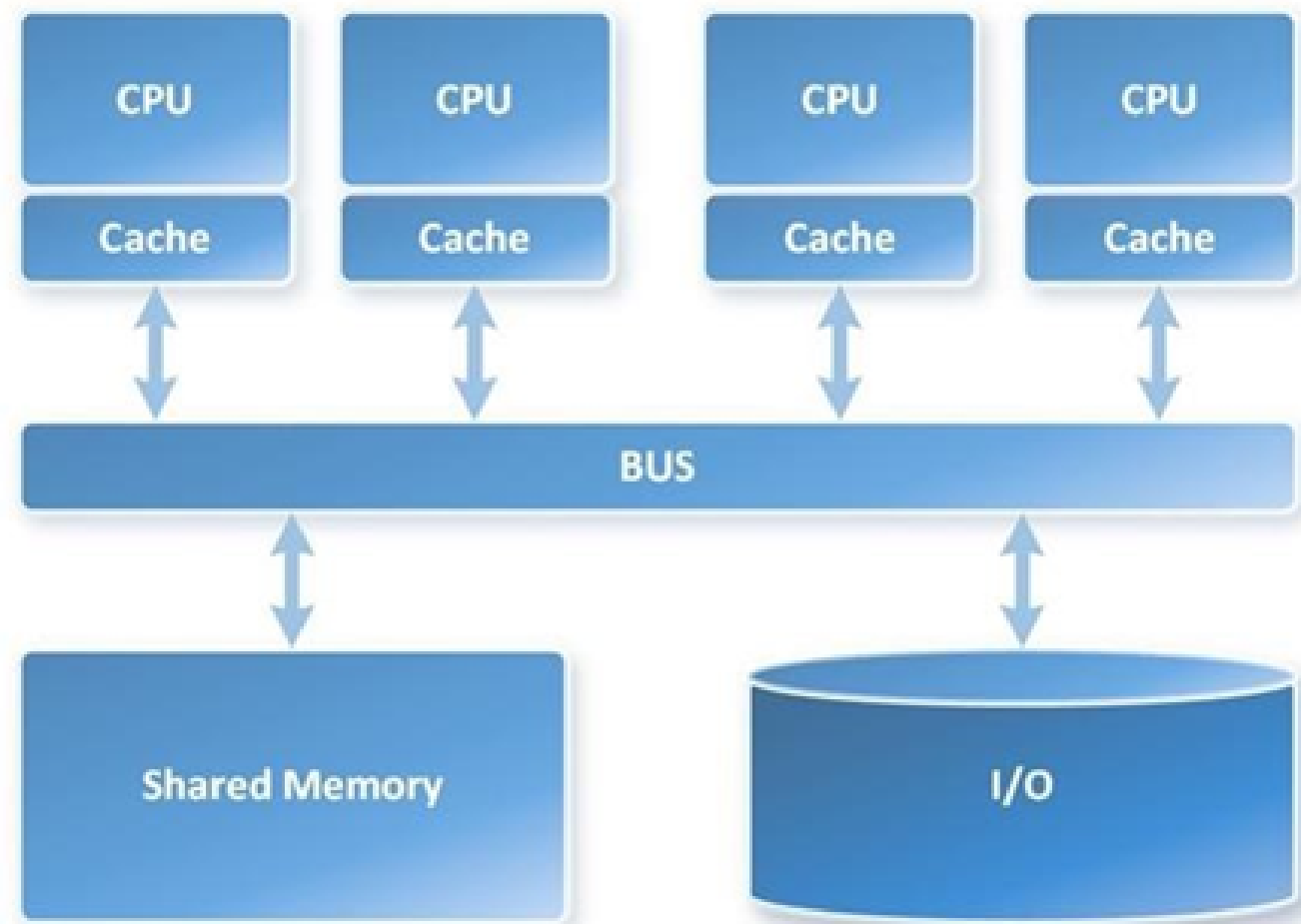


Multiprocesadores

Como todas las CPU ven la misma imagen de memoria, solo hay una copia del sistema operativo. En consecuencia, sólo hay un mapa de páginas y una tabla de procesos. Es esta imagen única del sistema es lo que principalmente distingue un multiprocesador de una multicomputadora, donde cada computadora tiene su propia copia del sistema operativo.

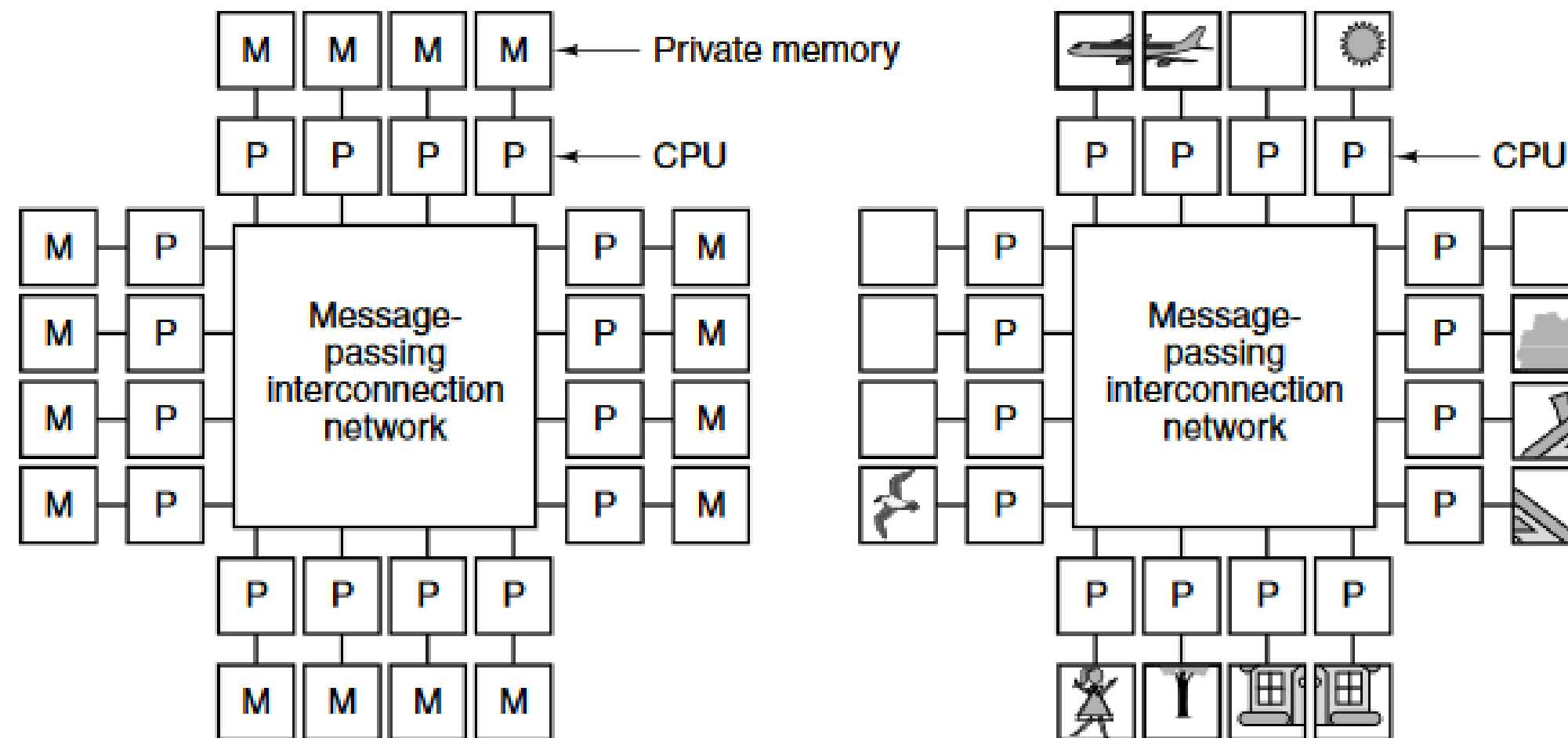
Computadoras de multiprocesadores, como toda computadora, tienen dispositivos de E/S de todo tipo pero sucede que en algunos sistemas, sólo ciertas CPU tienen acceso a los dispositivos de E/S, con funciones especiales para tal capacidad. Mientras que en otros diseños, cada CPU tiene igual acceso a todos los recursos. En este caso, el sistema operativo trata cada CPU como intercambiable con las demás y el sistema se denomina SMP (Symmetric MultiProcessor - multiprocesador simétrico).

Multiprocesadores



Multicomputadoras

El segundo diseño posible para una arquitectura paralela es en el que cada CPU tiene su propia memoria privada, accesible sólo para ella misma. Este tipo de diseño se denomina multicomputadora o sistema de memoria distribuida.



Multicomputadoras

En un sistema multicomputador, las CPU se comunican pasando mensajes de un lado a otro utilizando la red de interconexión y puede tener una variedad de diseños. Esta red tiene implicaciones para la estructura del software.

Por ejemplo, supongamos que CPU 0 descubre que parte del objeto que tiene se extiende a la memoria de otra CPU, digamos CPU 1. CPU 0 no puede simplemente leer la memoria de CPU 1, tiene que hacer algo diferente para obtener los datos que necesita. En particular, tiene que descubrir (de alguna manera) cual CPU tiene los datos que necesita y consecuentemente enviar un mensaje, al CPU correspondiente, solicitando una copia de los datos. Por lo general, se bloqueará hasta que se responda la solicitud. Cuando el mensaje llega a CPU 1, el software debe analizarlo y devolver los datos necesarios. Cuando el mensaje de respuesta vuelve a CPU 0, el software se desbloquea y puede continuar ejecutándose.

Multicomputadoras

La comunicación entre procesos suele utilizar primitivas de software, usualmente nombradas Enviar y Recibir. Esto le da al software una estructura diferente y mucho más compleja que en un multiprocesador. También significa que dividir correctamente los datos y colocarlos en ubicaciones óptimas es un problema importante.

Programar para aprovechar paralelismo en una multicomputadora es mucho más difícil que lo mismo en un multiprocesador. Si este es el caso, ¿por qué diseñar y construir multicomputadoras en primer lugar?

Multiprocesador vs Multicomputador

La respuesta es sencilla: las multicomputadoras grandes son mucho más sencillas y baratas de construir que los multiprocesadores con la misma cantidad de CPUs. En otras palabras, implementar una memoria compartida para una gran cantidad de CPUs, incluso por unos pocos cientos, es una tarea sustancial, mientras que construir una multicomputadora con 10.000 CPU o más es sencillo.

Entonces se nos presenta un dilema:

- Los multiprocesadores son difíciles de construir pero fáciles de programar.
- Las multicomputadoras son fáciles de construir pero difíciles de programar.

Multiprocesador vs Multicomputador

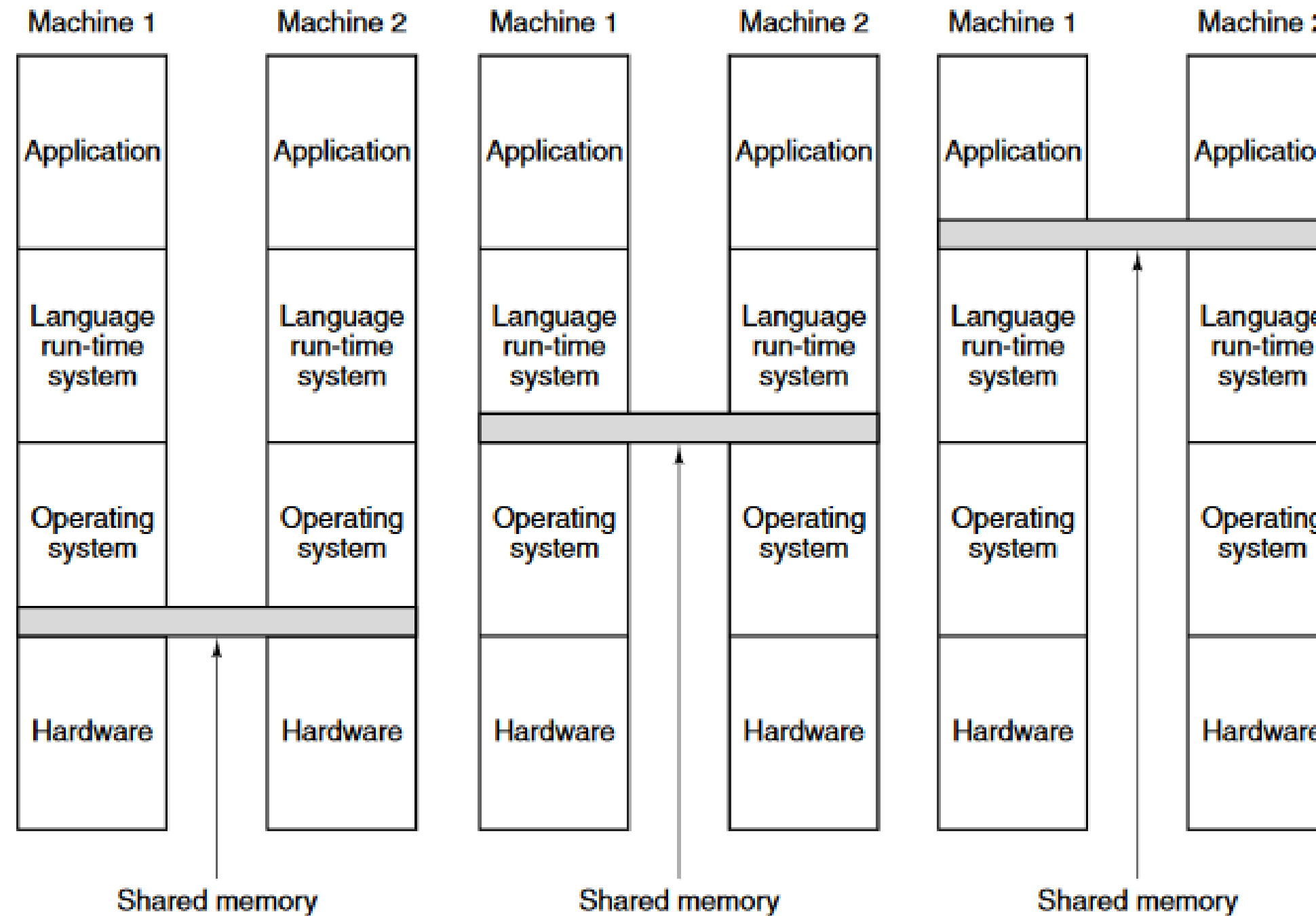
Esta observación ha llevado a un gran esfuerzo para construir sistemas híbridos que sean relativamente fáciles de construir y de programar. Este trabajo también ha llevado a la comprensión de que el concepto de memoria compartida se puede implementar de varias maneras, cada una con su propio conjunto de ventajas y desventajas. El objetivo de todo esto es encontrar diseños que sean escalables, que sigan funcionando bien a medida que se añaden más CPUs, y que sean fáciles de programar.

Una aproximación a un sistema híbrido se basa en el hecho de que los sistemas informáticos modernos no son monolíticos sino que están contruidos como una serie de capas. Esta idea abre la posibilidad de implementar la memoria compartida en cualquiera de ellas.

Memoria Compartida

En la siguiente figura, vemos en la parte más izquierda una memoria compartida implementada a nivel hardware como se vería en un multiprocesador. En este diseño, hay una única copia del sistema operativo con un único conjunto de tablas, en particular, la tabla de asignación de memoria.

Memoria Compartida



Memoria Compartida

Una segunda posibilidad es hacer que el sistema operativo simule la memoria compartida proporcionando un único espacio de direcciones virtuales paginadas compartidas para todos los procesadores de una maquina multiprocesador. Tambien existe la particularidad que este nivel de memoria compartida se lo puede implementar para un entorno multicomputador.

Este enfoque se llama DSM (Distributed Shared Memory - Memoria compartida distribuida). Aquí cada página se ubica en una de las memorias. Cada máquina tiene su propia memoria virtual y sus propias tablas de páginas. Cuando una CPU realiza una LOAD/STORE en una página que no tiene, se produce una trampa en el sistema operativo. El sistema operativo localiza la página y le pide a la CPU correspondiente que desasigne la página y la envíe a través de la red de interconexión. Cuando llega, la página se asigna y se reinicia la instrucción errónea. De ser así, el sistema operativo simplemente satisface los errores de página (pagination errors) desde la memoria remota en lugar de hacerlo desde el disco. Para el usuario, la máquina parece tener memoria compartida.

Memoria Compartida

Una tercera posibilidad es hacer que un sistema en tiempo de ejecución a nivel de usuario implemente una forma de memoria compartida. En este enfoque, el lenguaje de programación (o sus librerías) proporciona algún tipo de abstracción de memoria compartida, que luego es implementada por el compilador y el sistema en tiempo de ejecución.

Por ejemplo, el modelo Linda (para coordinación de comunicación) se basa en la abstracción de un espacio compartido de tuplas. Los procesos en cualquier máquina pueden ingresar una tupla desde el espacio de tuplas compartido o enviar una tupla al espacio de tuplas compartido. El acceso al espacio de tupla se controla completamente mediante software, no se necesita hardware especial ni soporte de sistema operativo.

Memoria Compartida

Operaciones de Linda:

Una de las principales ventajas de Linda es que es simple, con sólo seis operaciones para acceder y trabajar con el espacio de tuplas:

- out: Pone una tupla en el espacio de tuplas.
- in: saca una tupla que coincide con un patrón dado del espacio de tuplas.
- rd: Copia una tupla que coincide con un patrón dado del espacio de tuplas.
- eval: Crea un nuevo proceso para evaluar tuplas.
- inp: una versión sin bloqueo de in.
- rdp: una versión sin bloqueo de rd.

Taxonomía de Computadoras Paralelas

A lo largo de los años se han propuesto y construido muchos tipos de computadoras paralelas, por lo que es natural preguntarse si existe alguna forma de categorizarlas bajo una taxonomía. Un esquema popular es la Taxonomía de Flynn.

La taxonomía de Flynn se basa en dos conceptos, el flujo de instrucciones y el flujo de datos. Un flujo de instrucciones corresponde a un contador de programa. Un sistema con n CPU tiene n contadores de programas y, por tanto, n flujos de instrucciones. El flujo de datos consta de un conjunto de operandos dentro de una operación.

Instruction streams	Data streams	Name	Examples
1	1	SISD	Classical Von Neumann machine
1	Multiple	SIMD	Vector supercomputer, array processor
Multiple	1	MISD	Arguably none
Multiple	Multiple	MIMD	Multiprocessor, multicomputer

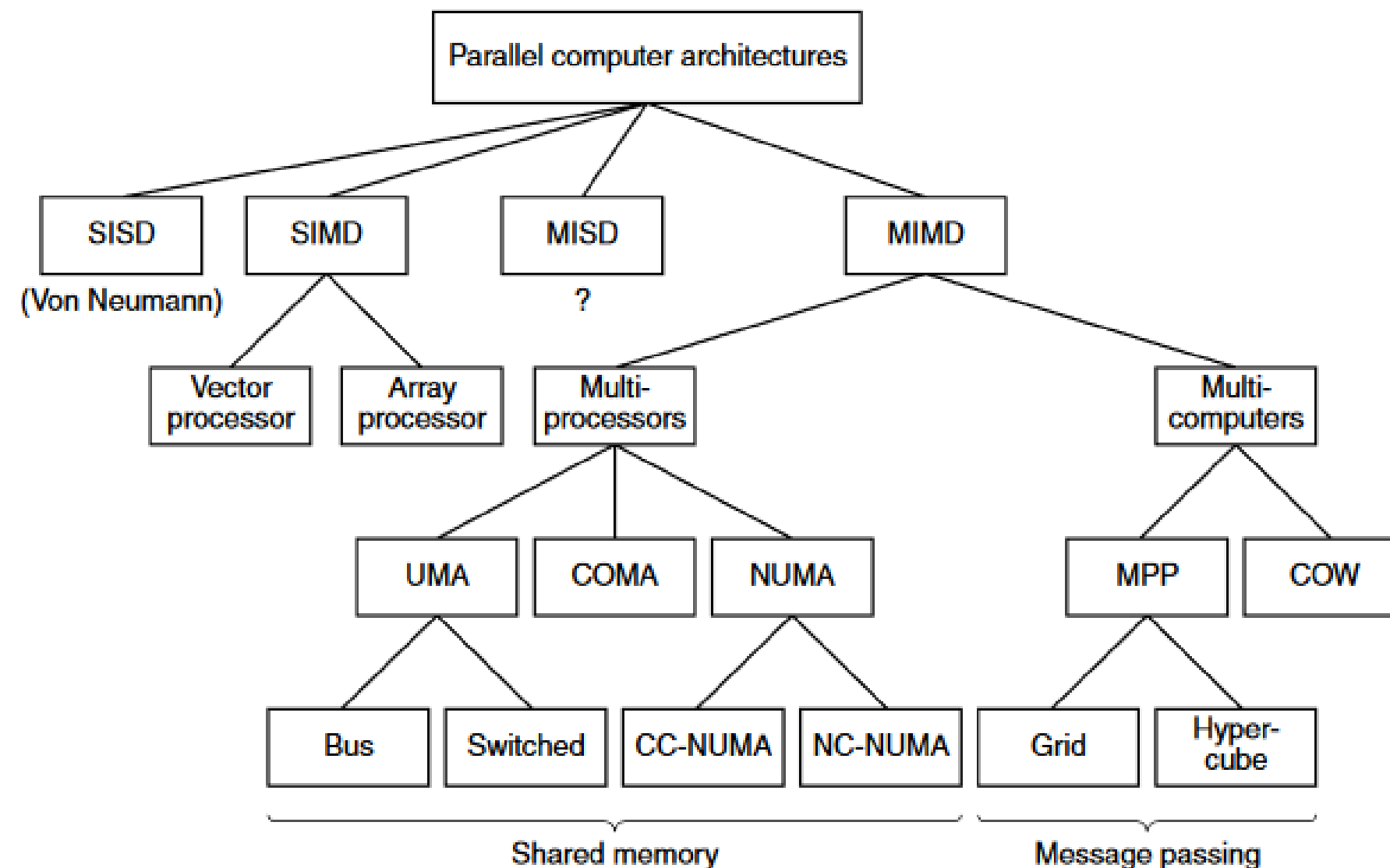
Taxonomía de Computadoras Paralelas

El flujo de instrucciones y el flujo de datos son, hasta cierto punto, independientes, por lo que existen cuatro combinaciones:

- SISD: el clásico ordenador secuencial de von Neumann. Tiene un flujo de instrucciones y se ejecuta con un dato a la vez.
- SIMD: tienen una única unidad de control que emite una instrucción a la vez, al disponer de múltiples ALU para ejecutarla sobre múltiples conjuntos de datos simultáneamente. Las computadoras convencionales a veces tienen algunas instrucciones SIMD especiales. También existen algunos procesadores SIMD especiales, como los procesadores de flujo (stream processors), que están diseñados específicamente para manejar la demanda de renderizado multimedia.
- MISD: son una categoría extraña, con múltiples instrucciones operando sobre el mismo dato. Algunas personas consideran que las máquinas con segmentación son MISD.
- MIMD: son simplemente múltiples CPU independientes que funcionan como parte de un sistema más grande. La mayoría de los procesadores paralelos entran en esta categoría. Tanto los multiprocesadores como las multicomputadoras son máquinas MIMD.

Taxonomía de Computadoras Paralelas

Lo previo, es la esencia de la taxonomía de Flynn, pero en realidad cada grupo se puede subdividir en otros subgrupos. Por ejemplo, SIMD se puede subdividir en dos subgrupos: los procesadores vectoriales para supercomputadoras numéricas y otras máquinas que operan con vectores, realizando la misma operación en cada elemento vectorial; y los procesadores de matriz para máquinas de tipo paralelo, donde una unidad de control maestra despacha instrucciones a muchas ALU independientes.



Taxonomía de Computadoras Paralelas

MIMD, como se indicó anteriormente, se puede subdividir en multiprocesadores y multicomputadoras, donde cada uno se puede subdividir en más subgrupos. Existen tres tipos de multiprocesadores, que se distinguen por cómo se implementa la memoria compartida:

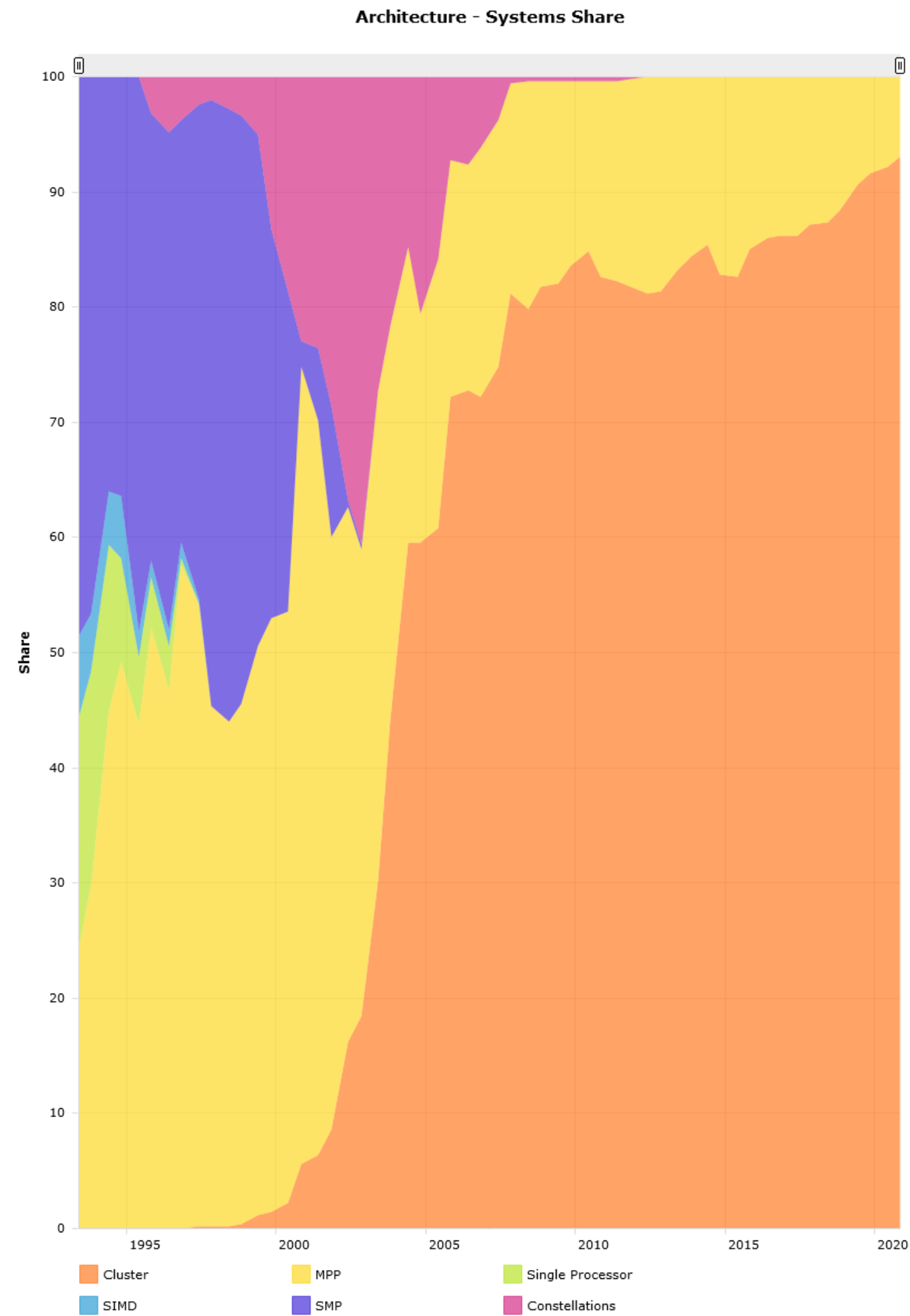
- UMA (Acceso Uniforme a Memoria): cada CPU tiene el mismo tiempo de acceso a cada módulo de memoria. En otras palabras, cada palabra de memoria se puede leer tan rápido como cualquier otra palabra de memoria. Si esto es físicamente imposible, las referencias más rápidas se ralentizan a la más lenta, por lo que los programadores no ven la diferencia. Esto es lo que significa uniforme en su nombre. Esto hace que el rendimiento sea predecible, un factor importante para escribir código eficiente.
- NUMA (Acceso a Memoria No Uniforme): a diferencia de UMA, en NUMA el acceso a memoria no es uniforme. El resultado es que, por motivos de rendimiento, es importante saber dónde se colocan el código y los datos.
- COMA (Cache Only Memory Access): son similares a NUMA, en el sentido de que es un modelo no uniforme. A diferencia de NUMA (y UMA) que usan la memoria principal, como memoria; en COMA, la memoria principal se la utiliza como cache.

Taxonomía de Computadoras Paralelas

Las multicomputadoras, debido a que no tienen acceso directo a la memoria remota, a veces se las denomina máquinas NORMA (NO Remote Memory Access). Hay dos tipos de multicomputadoras:

- MPP (Massively Parallel Processors - procesadores masivamente paralelos): son supercomputadoras costosas que constan de muchas CPU estrechamente acopladas por una red de interconexión patentada de alta velocidad.
- NOW (Network of Workstations - red de estaciones de trabajo)/COW (Cluster of Workstations - Clúster de estaciones de trabajo): consta de PCs, estaciones de trabajo o servidores, conectados mediante tecnología de interconexión comercial disponible en el mercado.

Lógicamente, no hay mucha diferencia entre MPP y COW, pero las computadoras enormes que cuestan muchos millones de dólares se usan de manera diferente a las redes de PC que cuestan una fracción del precio.



2023
Segundo Cuatrimestre

