

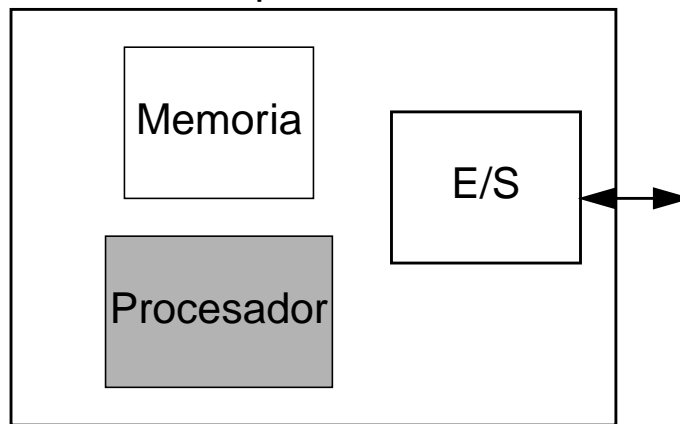
Introducción a los Sistemas Multiprocesadores

- Multiprocesadores estilo Von Neumann
- Modelos de Organización
- Modelos de Programación
- Clasificación de los Multiprocesadores
- ¿Por qué Sistemas Multiprocesadores?
- Problemas y Retos

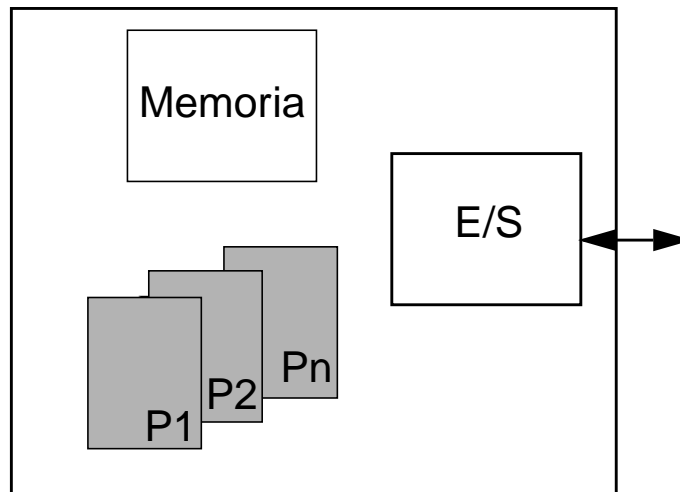


Multiprocesadores estilo Von Neumann

Computador Von Neumann



Multiprocesador Von Neumann



Otros Tipos de Multiprocesadores

- Multiprocesadores de Reducción**
- Multiprocesadores de Flujo de Datos**
- Redes Neuronales**

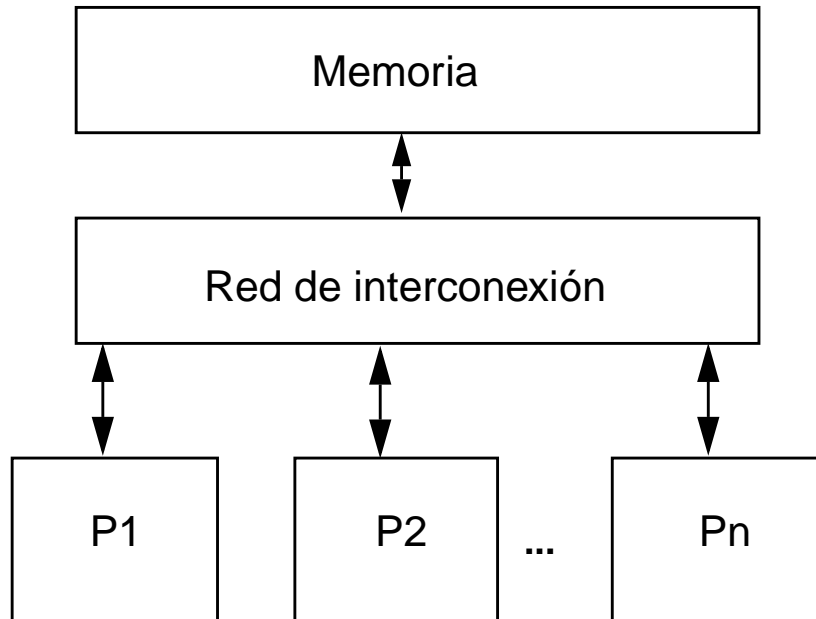


Objetivos de un Multiprocesador

- Acelerar la ejecución de aplicaciones (*orientados a speed-up*). *SUPERCOMPUTACION*.**
- Ejecutar más aplicaciones por unidad de tiempo (*orientados al throughput*).**

Modelos de Organización

❑ Multiprocesadores con Memoria Compartida



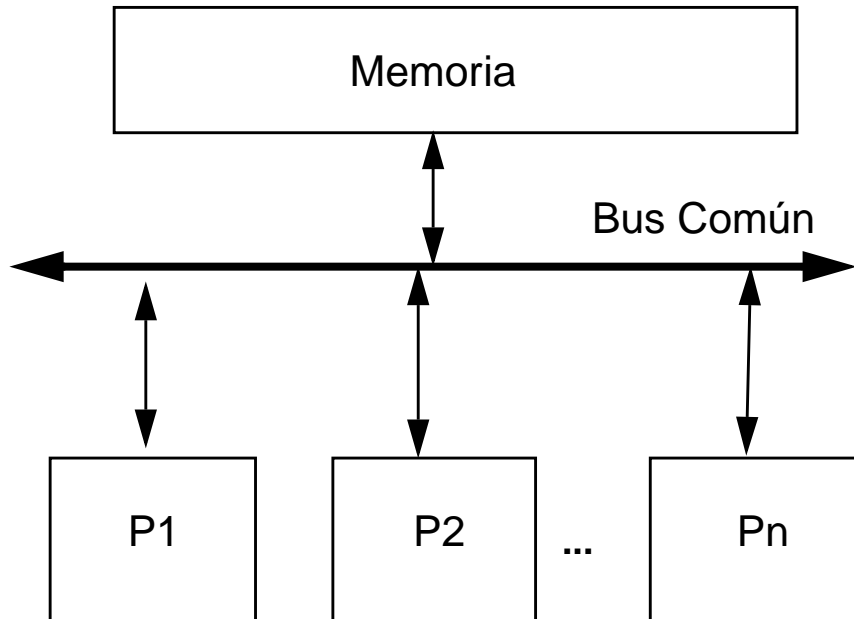
La red de interconexión permite a cualquier procesador acceder a cualquier posición de memoria.

Términos equivalentes

- ✓ Multiprocesador con Memoria Global
- ✓ Multiprocesador Fuertemente Acoplado
- ✓ Multiprocesador

Modelos de Organización

❑ Multiprocesadores con Memoria Compartida

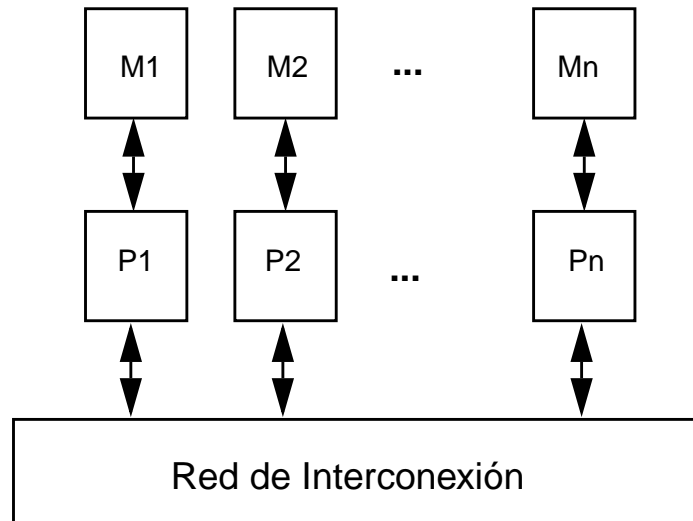


Problema

La red de interconexión representa un cuello de botella que impide que esta organización sea escalable.

Modelos de Organización

❑ Multiprocesadores con Memoria Distribuida



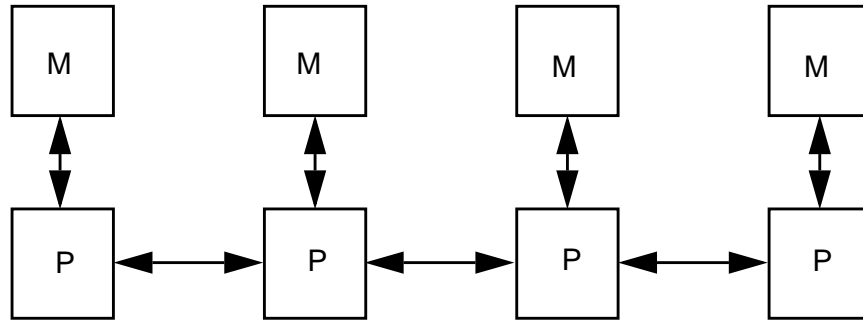
La red de interconexión permite a cualquier procesador comunicarse con cualquiera de los procesadores del sistema.

Términos Equivalentes

- ✓ Multiprocesador con Memoria Local
- ✓ Multiprocesador Débilmente Acoplado
- ✓ Multicomputador

Modelos de Organización

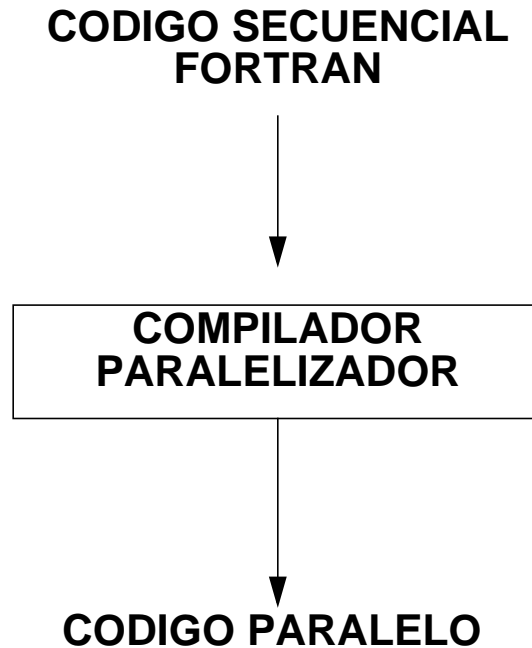
❑ Multiprocesadores con Memoria Distribuida



La organización puede ser escalable en la medida en que los programas realicen comunicaciones locales.

Modelos de Programación

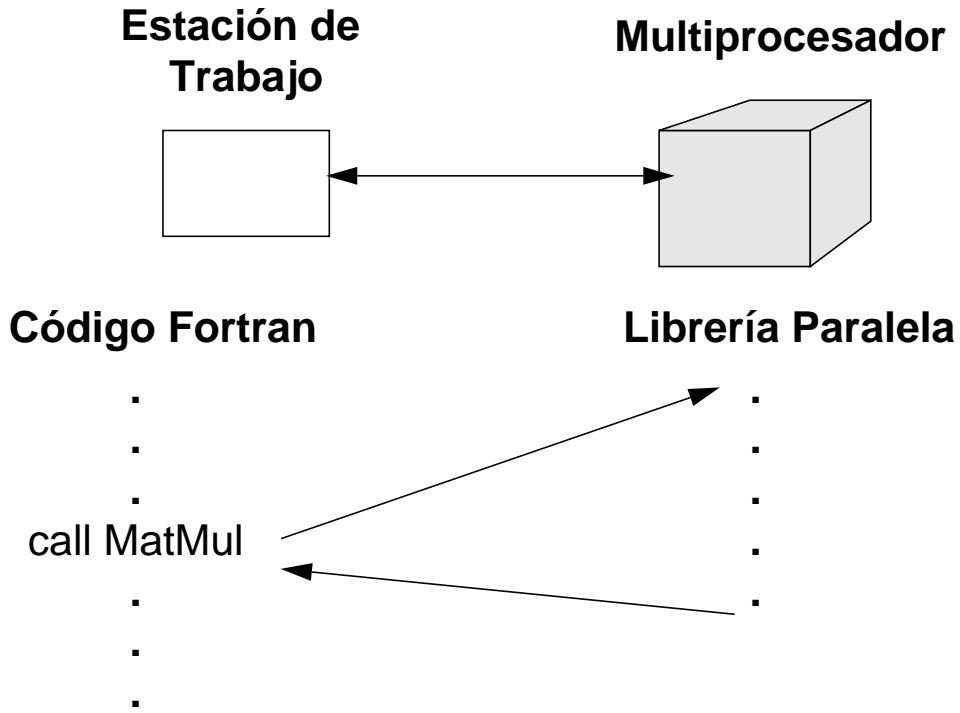
□ Programación Secuencial



Este es el modelo de programación ideal. Sin embargo, la tecnología de compilación no permite todavía obtener un buen rendimiento de los sistemas multiprocesador (en particular, de los que tienen memoria distribuida).

Modelos de Programación

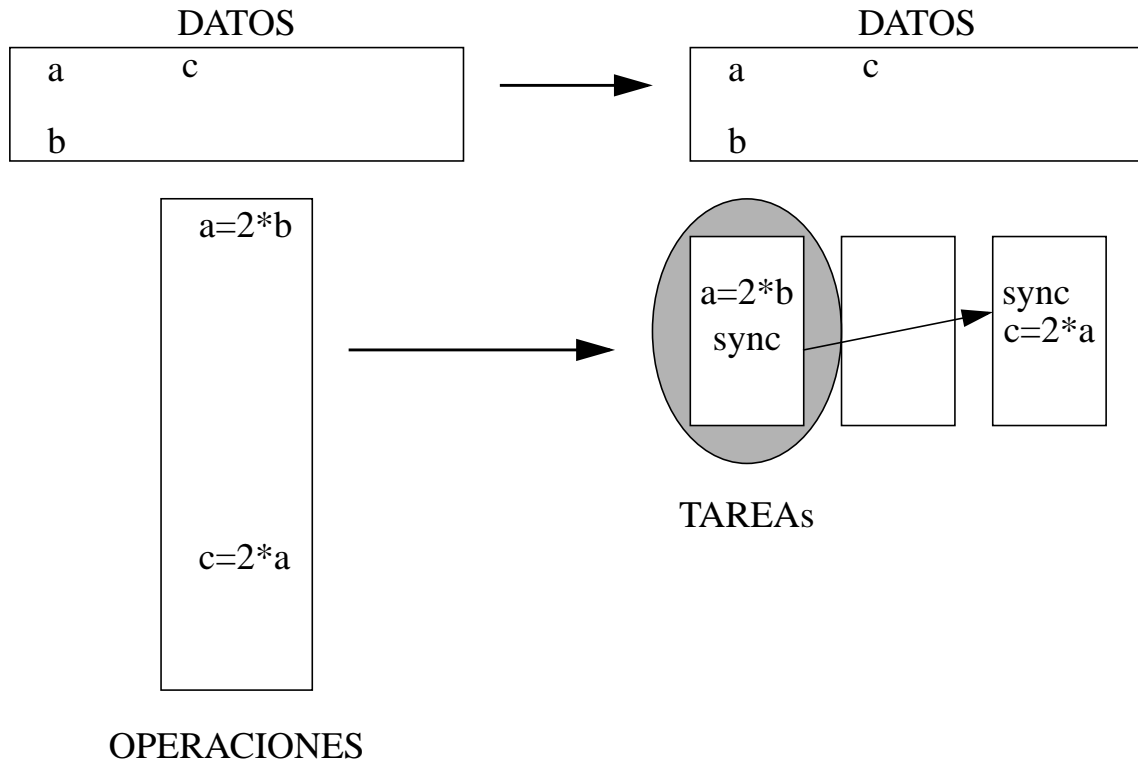
□ Programación Secuencial + Librería de Rutinas Paralelas



Limita el uso del sistema multiprocesador.

Modelos de Programación

❑ El Modelo de Variables Compartidas

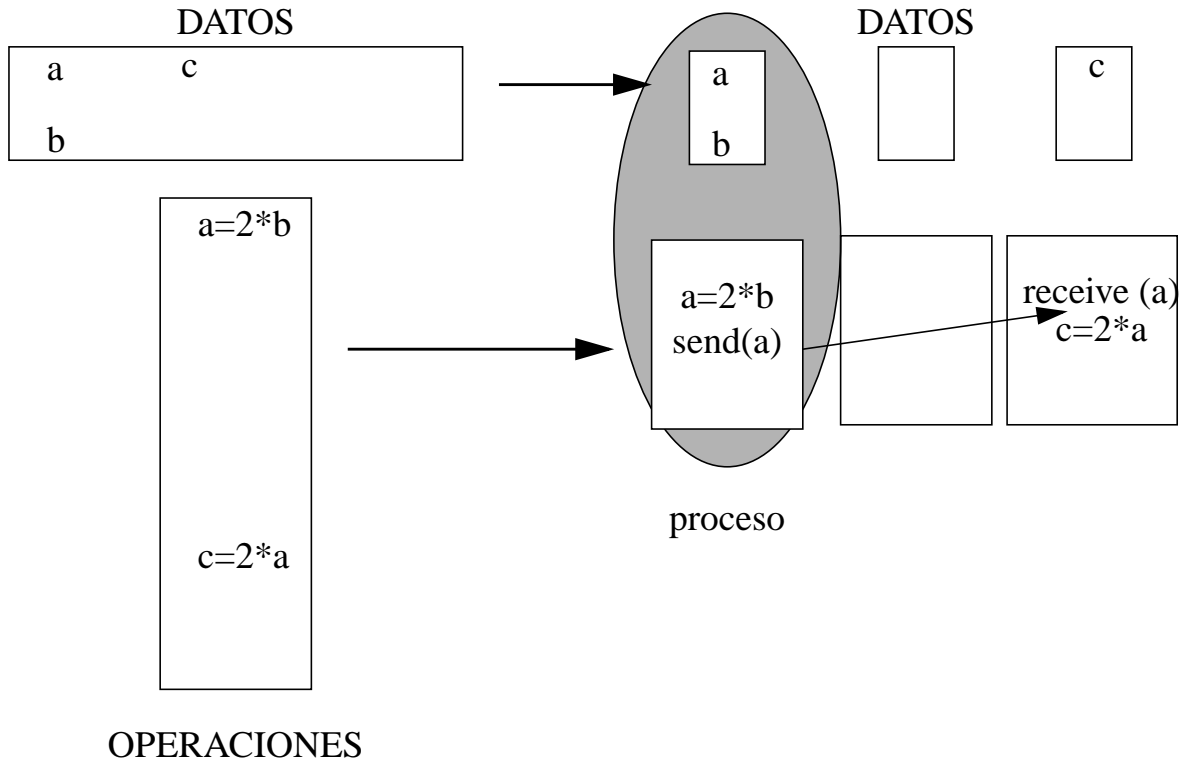


Las operaciones se descomponen en tareas.
Los datos son compartidos por todas las tareas.
Se requieren primitivas de sincronización para:

- ✓ Señalización
- ✓ Acceso Exclusivo

Modelos de Programación

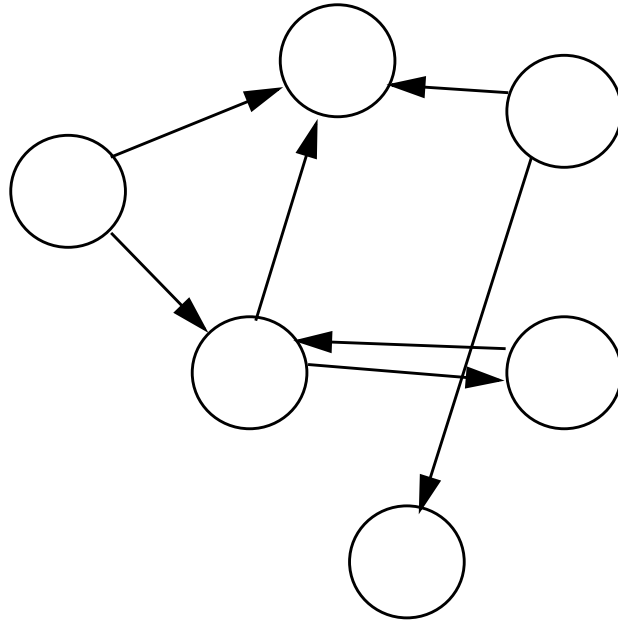
□ El Modelo de Paso de Mensajes



Las operaciones Y LOS DATOS se descomponen en procesos.
Los procesos sólo tienen acceso directo a los datos privados (locales).
Los datos no locales se acceden mediante intercambio de mensajes entre los procesos.

Modelos de Programación

□ El Modelo de Paso de Mensajes



En el modelo de paso de mensajes, un programa paralelo se ve como un conjunto de procesos que se intercambian información a través de canales.

Clasificación de los Multiprocesadores

		PROGRAMACIÓN	
		Variables Compartidas	Paso de Mensajes
ORGANIZACIÓN	Memoria Compartida	<p><i>SMP</i> <i>(Symmetric Multiprocessors)</i></p> <p>Combinación natural Poco escalable Fácil de programar</p>	<p>?</p> <p>Poco escalable Programación difícil</p>
	Memoria Distribuida	<p><i>DSM</i> <i>(Distributed Shared Memory)</i></p> <p>Programación fácil Escalable Implementación difícil</p>	<p><i>Multicomputer</i></p> <p>Combinación natural Programación difícil Escalable</p>

¿Por qué Sistemas Multiprocesadores?

Está ampliamente aceptado que los sistemas multiprocesadores (en particular, los de memoria distribuida) representan uno de los caminos más razonables para seguir aumentando la potencia de cálculo de los computadores

Problemas y Retos

Hardware

Buscar organizaciones del sistema que permitan al software obtener una fracción significativa de la velocidad máxima del sistema.

Aplicaciones y Algoritmos

Identificar aplicaciones críticas que pueden beneficiarse del uso de los sistemas multiprocesadores.

Encontrar algoritmos paralelos eficientes para resolver los núcleos computacionales más habituales.

Herramientas

Desarrollar herramientas (modelos de programación, compiladores, depuradores, monitorizadores de rendimiento, etc) que faciliten el uso del sistema.

