

Caracterización de los sistemas distribuidos

Leandro R. León
Cemisid
Cecalcula

¿Qué es un sistema distribuido?

"A collection of independent computers that appear to the users of the system as a single computer"

-- Tanenbaum

Una colección de computadoras autónomas, débilmente acoplada, conectada en red, y que usa un software que produce un ambiente integrado de computación

--Para nuestro curso

- ◆ Definir SD es una de las actividades más polémicas que existen, pero podemos ayudarnos si:
 - Aclaremos un poco de terminología
 - Considerar bajo cuáles sabores se puede ver la "distribución"
 - Considerar las diferentes plataformas de hardware
 - Observamos ejemplos e intentamos caracterizar SD respecto a los sistemas centralizados

Terminología

Paralelismo

- Alegoría de la informática con la geometría
- Dos líneas son paralelas si jamás se interceptan
- Dos flujos de ejecución son paralelos mientras no se crucen
 - Mientras los flujos son // \Rightarrow flujos independientes
 - **Sistema o aplicación que posea flujos paralelos es "paralelo"**

Distribución

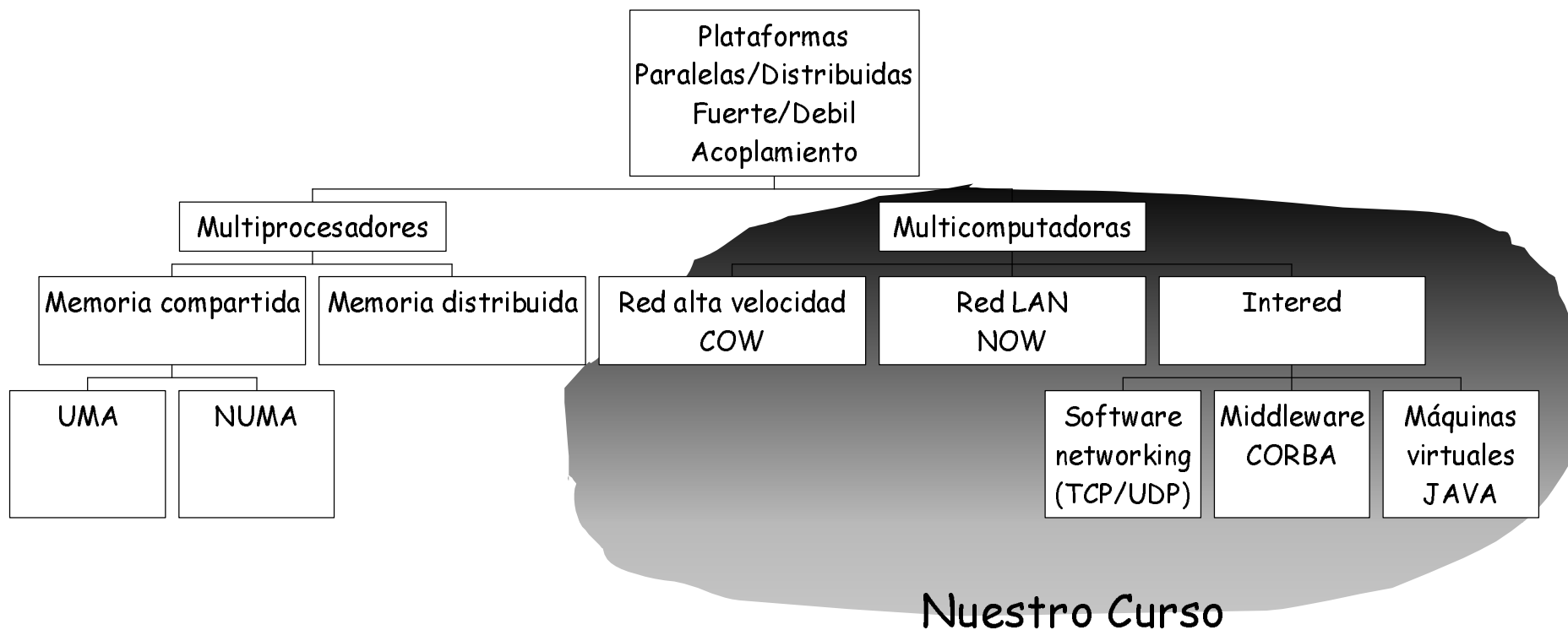
- Alegoría con "*repartir*" cual es sinónimo de "*distribuir*"
- ¿Qué se reparte?
 - control
 - Datos
 - Hardware
- **Sistema o aplicación que "*distribuya*" algo es "*distribuido*"**

Terminología (cont.)

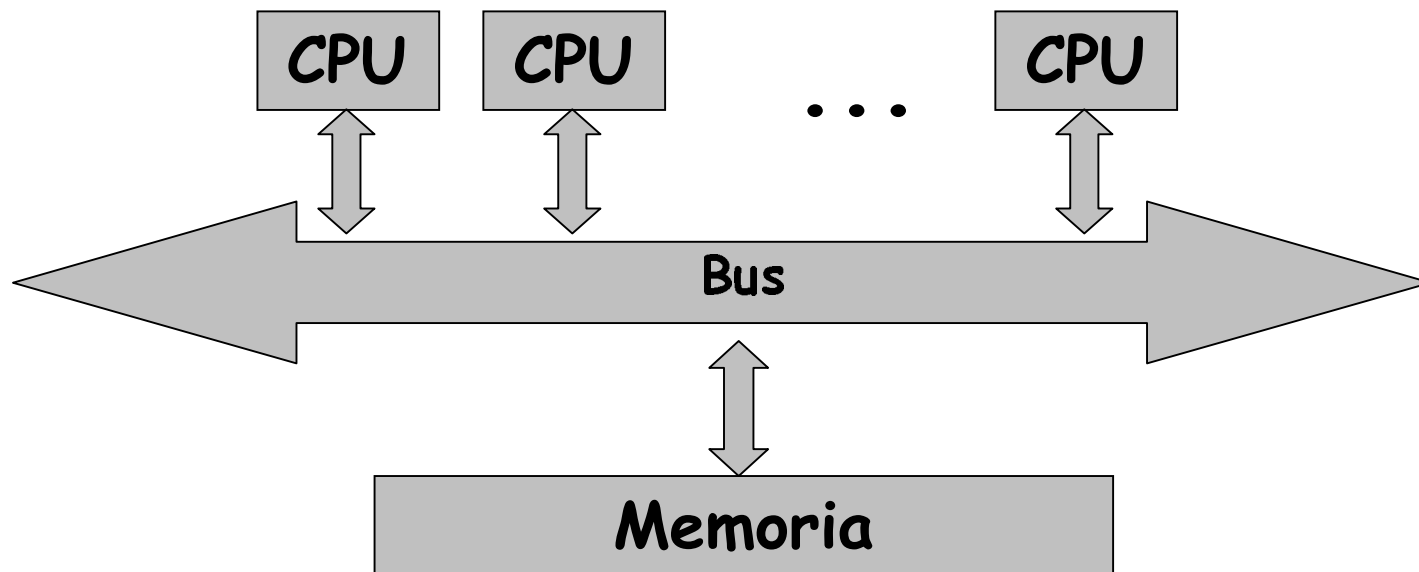
Concurrente

- Alegoría de "*concurrir*": ir hacia el mismo sitio
- Orden de llegada no se considera
- ¿Qué cosas pueden concurrir?
 - Control
 - Datos
- ¿Dónde concurren? Hacia los recurso (sitios, recursos)

Identificar el hardware

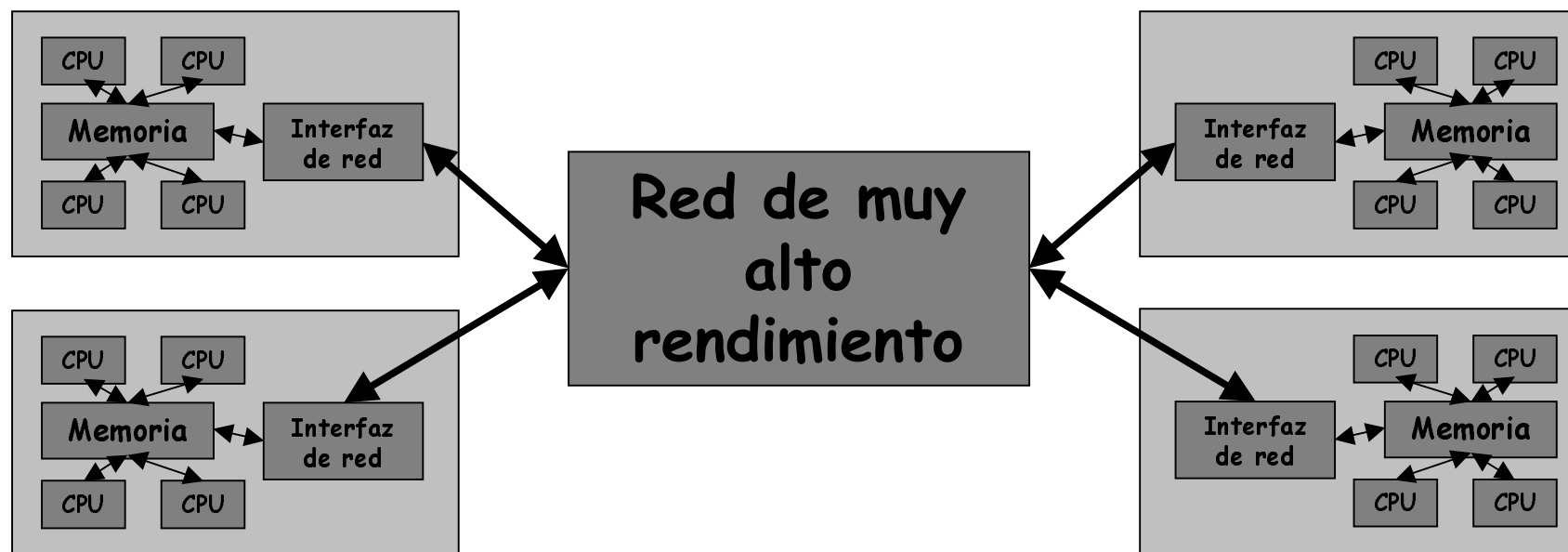


Multiprocesadores UMA



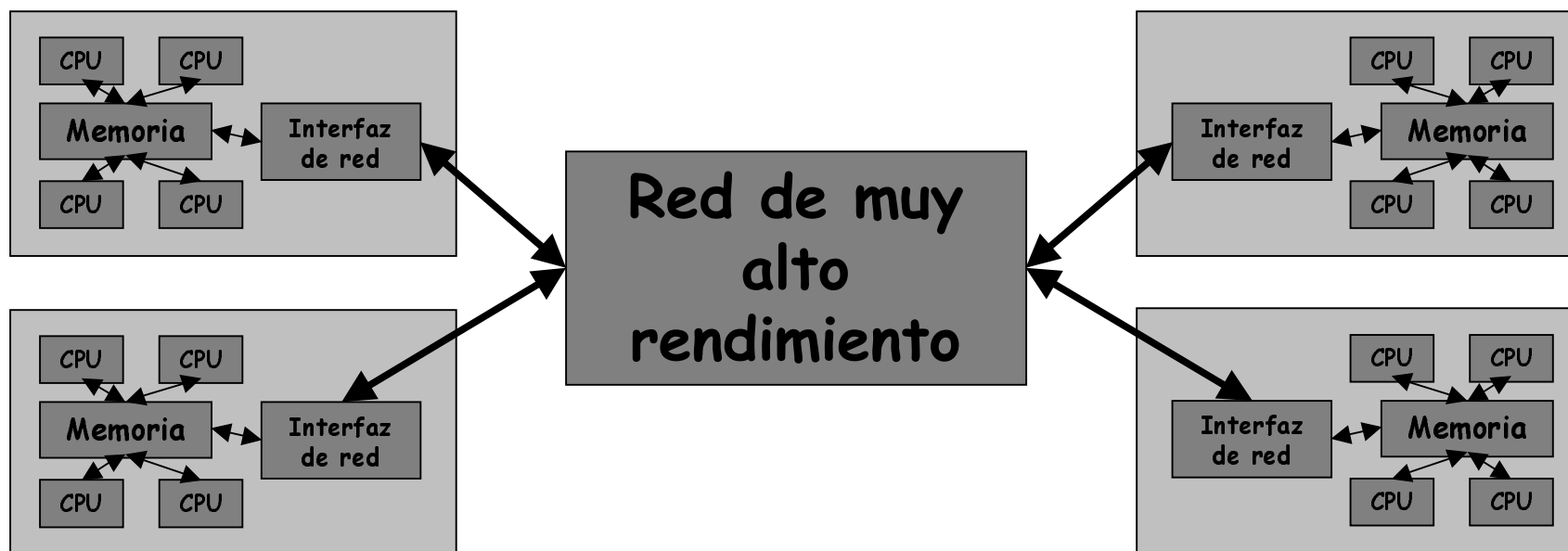
- ◆ Muy rápido
- ◆ Programación no muy alejada de la programación tradicional
- ◆ Bus limita la escalabilidad

NUMA y CC-NUMA



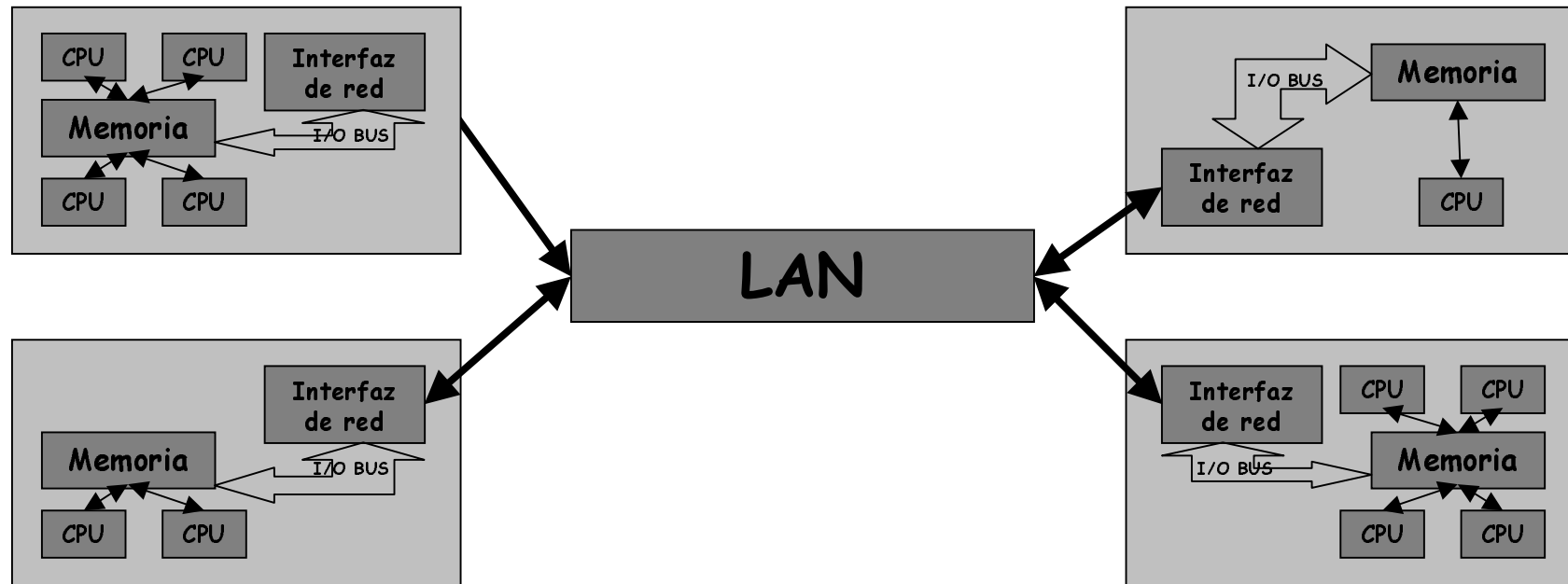
- ◆ Soporte de hardware para memoria compartida y coherencia de cache (CC)
- ◆ Memoria es vista como una sola
- ◆ Acceso remoto y local se hacen de la misma manera
- ◆ Red de muy alto rendimiento basada en conmutadores
- ◆ Memorias remotas son más lentas
- ◆ Programación igual que UMA

Multiprocesadores a memoria distribuida



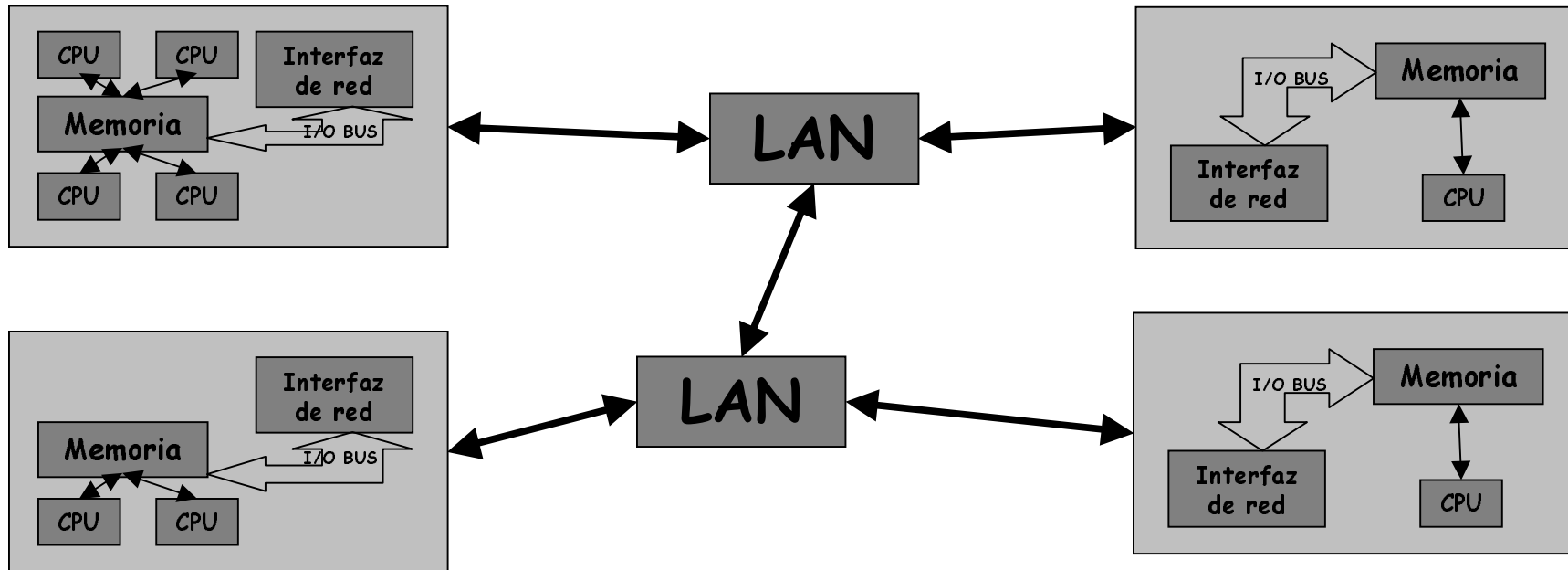
- ◆ No hay soporte hardware para acceso directo a memoria remota y para coherencia de cache
- ◆ Visión de memoria única es implantada en software
- ◆ Pero la ausencia de éste soporte abarata los costes
- ◆ Acceso a memoria remota requiere envío de mensajes
- ◆ Programación es más dificultosa

Redes y/o clusters de WS



- ◆ Nodos independientes; configuraciones diferentes; quizá heterogéneos;
- ◆ Red de área local
 - Mucho más barata que las dedicadas
 - Mucho más lentas
 - Jitter muy bajo
- ◆ Programación por pase de mensajes o software memoria compartida
- ◆ Software de networking de alto rendimiento

Interredes



- ◆ Nodos heterogéneos
- ◆ Red de área metropolitana/Mundial
 - Muy lentas
 - Alto jitter
- ◆ Programación por pase de mensajes

Características de plataforma en un SD

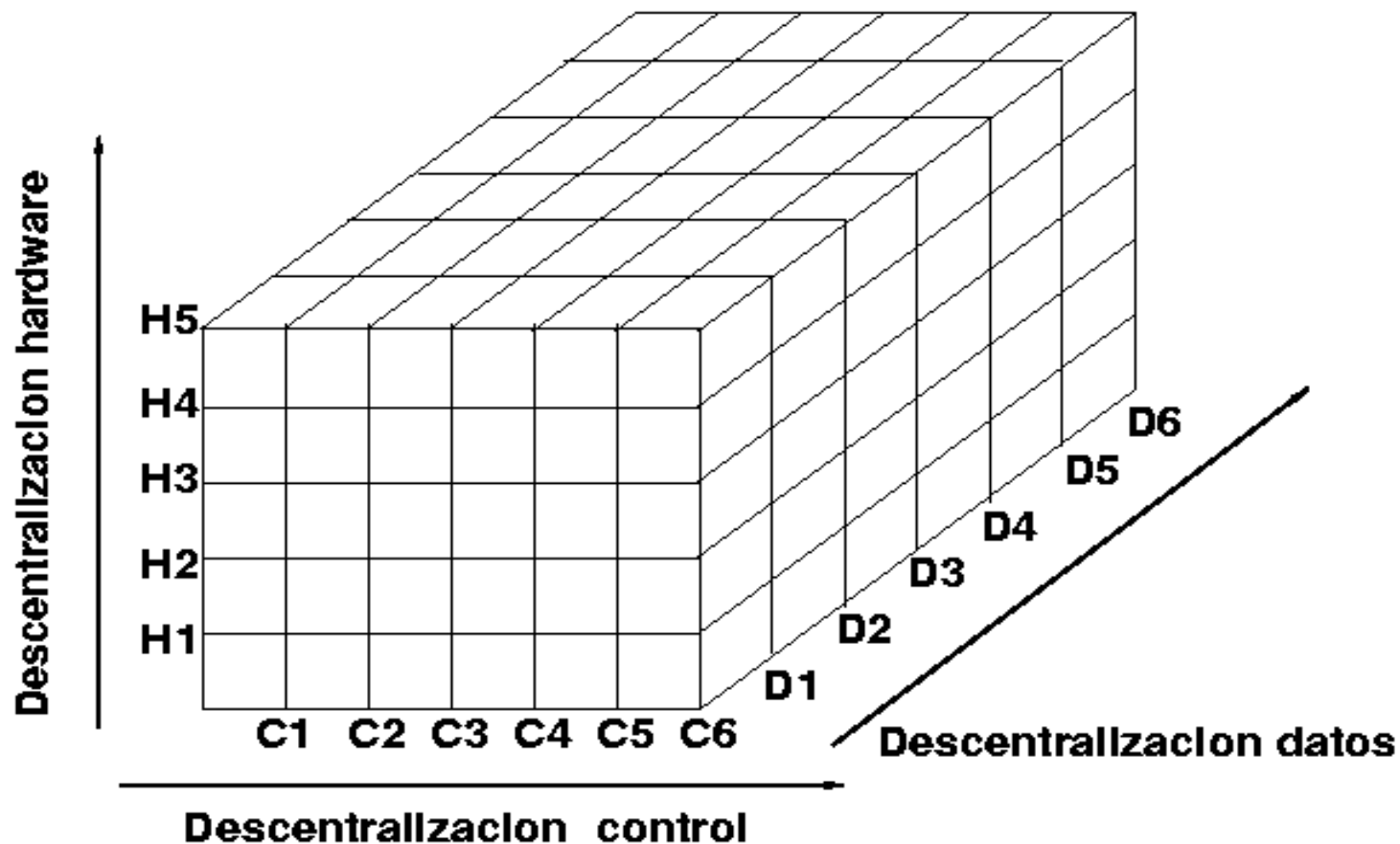
- ◆ Acoplamiento débil \Rightarrow acceso remoto considerablemente más lento que acceso local
- ◆ Autonomía de las computadoras: recursos locales de cada computadora son gestionados localmente (con algún grado de autonomía)
- ◆ Conexión de red por software:
 - Comunicación remota \Rightarrow envío de mensajes
 - Procesamiento de mensajes es parte del programa (pasan por CPU)
 - Protocolos de confianza, detección de fallas, orden, etc. son implantados en software
- ◆ La única forma de comunicar es por pase de mensajes; no hay un mecanismo de señalización en hardware que implante lo que se hace en software

Obstáculos respecto a plataformas centralizadas

- ◆ Mucho más difícil
- ◆ Concurrencia
 - Modelos de programación diferentes
 - No hay soporte de hardware (ej. test & set en simple CPU)
- ◆ Tiempo
 - Imposibilidad de reloj global
 - Orden de eventos es difícil
- ◆ Conexión en red:
 - Comunicación solo por envío de mensajes
 - Fallas son muy difíciles de detectar
 - Imposibilidad de distinguir entre una falla en un nodo o una falla de red
 - Más difícil el control de privacidad y de autenticación

Definición de Enslow (grado de distribución)

Sistema Distribuido = Hardware Distribuido + Control distribuido + Datos Distribuidos



Descentralización del hardware

- ◆ **H1:** Un simple CPU; un sólo flujo interno
 - ◆ **H2:** Un simple CPU con múltiples ALU; una sólo flujo interno
 - ◆ **H3:** Unidades de hardware especializadas
 - Un CPU con coprocesador de punto flotante
 - Un CPU súper escalar
 - ◆ **H4:** Varios CPU fuertemente acoplados
 - Comparten un bus
 - Memoria común
 - I/O común
 - ◆ **H5:** Computadores (H1 .. H4) interconectados en red
 - Memorias locales
 - I/O locales
- ¿A qué nivel pertenecen las plataformas estudiadas?

Descentralización del control

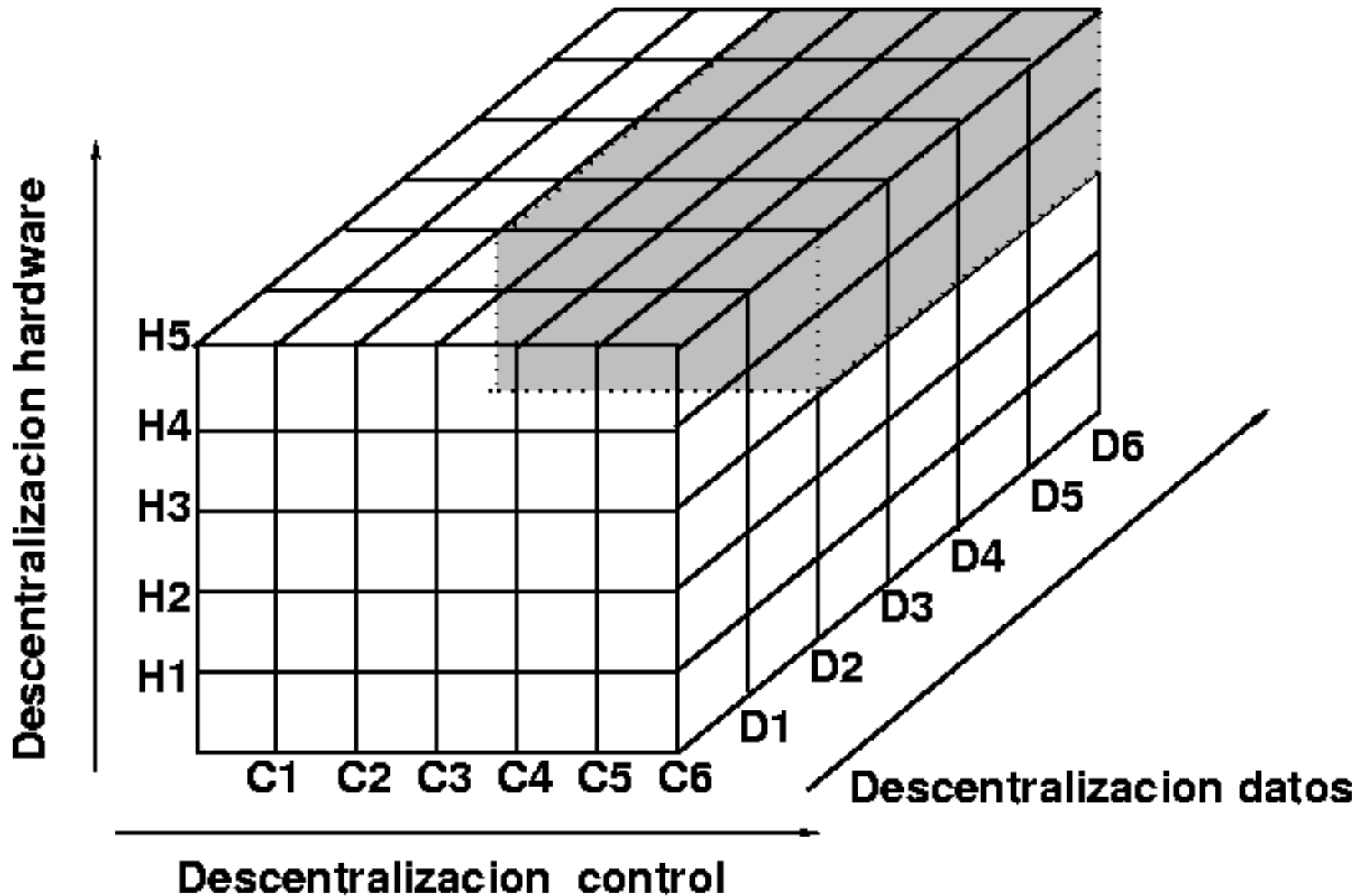
- ◆ **C1:** Un solo flujo de control (independiente del valor H_i)
- ◆ **C2:** Un solo flujo de control que dinámicamente cambia entre unidades de hardware
- ◆ **C3:** Una estructura fija maestro/esclavo
 - Un CPU maestro con un coprocesador esclavo
 - Un Servidor fijo con múltiples clientes
- ◆ **C4:** Una estructura dinámica maestro/esclavo
 - El rol de servidor cambia dinámicamente
 - Roles son modificados por software
- ◆ **C5:** Flujos homogéneos de control
 - Flujos pueden ser paralelos/concurrentes
 - El programa (controlador) es el mismo
- ◆ **C6:** Múltiples flujos de control, heterogéneos
 - Flujos paralelos/concurrentes/independientes
 - Diferentes controladores por flujo

Descentralización de los datos

Asumiendo datos en archivos y un directorio

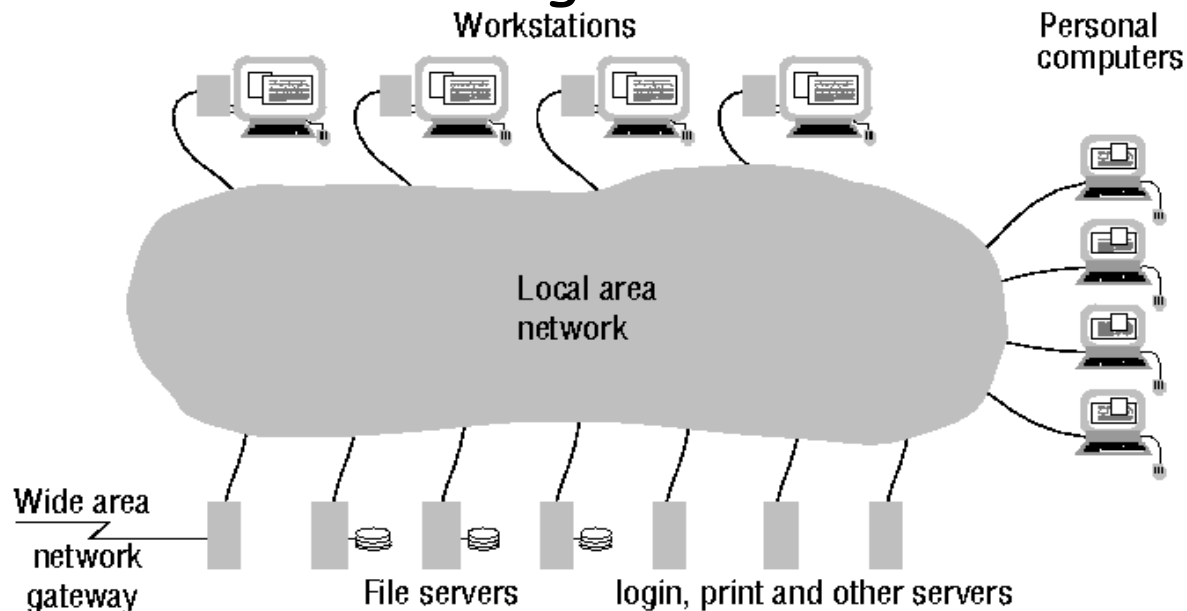
- ◆ **D1:** Datos centralizados; una sólo copia de archivos y de directorios
- ◆ **D2:** Archivos distribuidos a lo largo de los sitios; un directorio central
- ◆ **D3:** Archivos y directorio replicados; una copia de cada archivo y del directorio en cada sitio
- ◆ **D4:** Archivos distribuidos (no necesariamente replicados) en sitios; un sitio maestro mantiene una copia completa de los archivos y del directorio
- ◆ **D5:** Archivos distribuidos en sitios; un sitio maestro mantiene una copia del directorio
- ◆ **D6:** Archivos distribuidos en sitios sin ningún archivo o directorio maestro

¿Dónde están los SD? (grado de distribución)



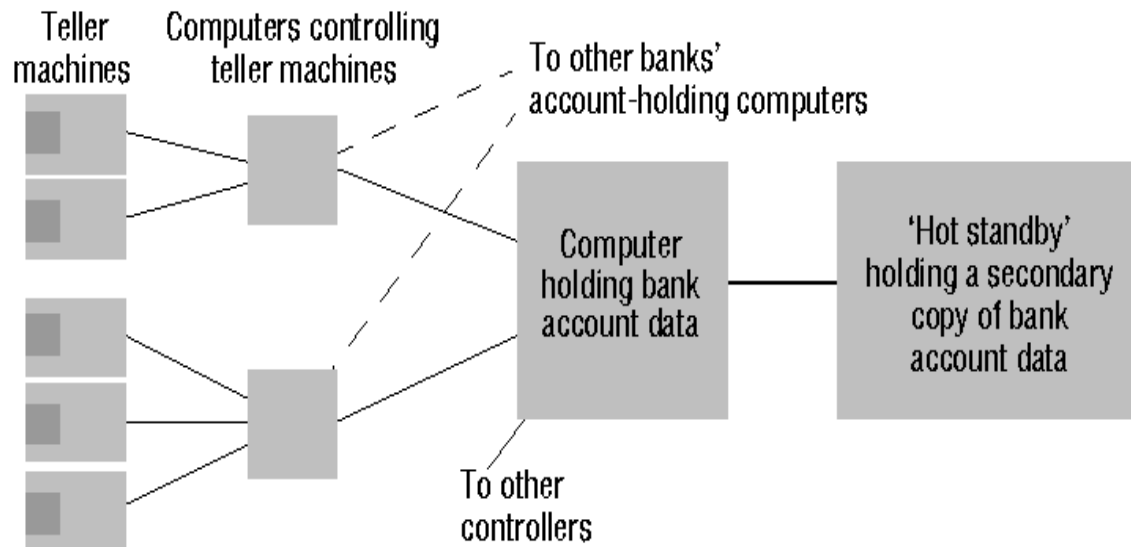
Ejemplo 1: UNIX distribuido

- ◆ Meta fue extender el tiempo compartido a una red
- ◆ Distribución parcial, implantada por servidores. Ej. NFS, X11, email, login.
- ◆ No hay casi gestión de recursos distribuida (procesadores, memoria, etc.)
- ◆ UNIX es +- abierto (apertura)
- ◆ ¿Cuál es la calificación según Enslow?



Ejemplo 2: Cajero Automático

- ◆ Meta es comercio a lo largo de la geografía
- ◆ Requerimientos críticos:
 - Confianza (Tolerancia a fallas)
 - Seguridad - autenticación
 - Aumento de escala (escalabilidad)
- ◆ Problemas a resolver:
 - Consistencia
 - Concurrencia
 - Atomicidad



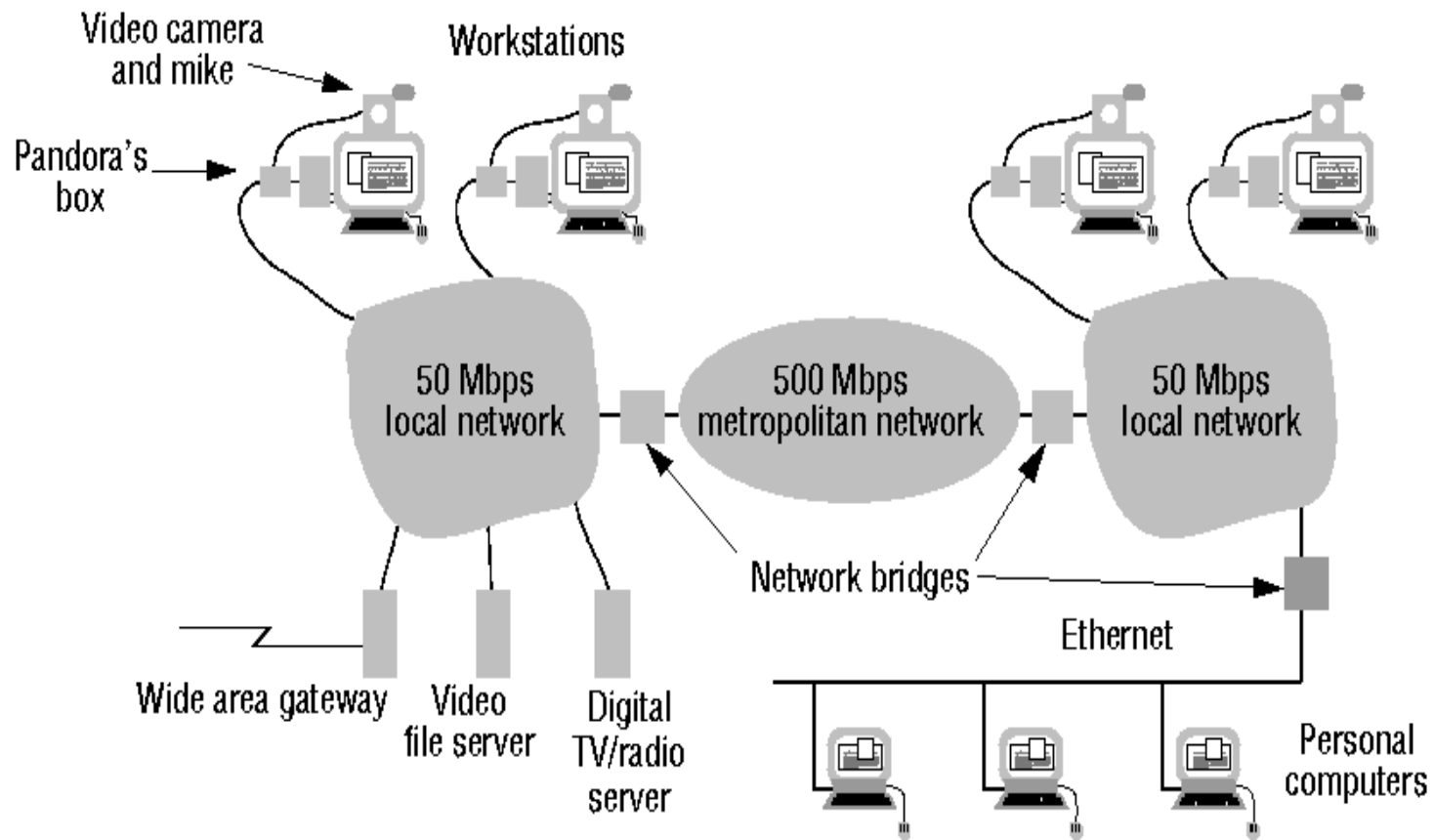
- ◆ ¿Cuál es la calificación según Enslow?

Ejemplo 3: Internet

- ◆ Red mundial con servicios diversos
- ◆ Aplicaciones ejemplos:
 - email
 - WWW
 - News
 - IRC
- ◆ Características relevantes:
 - Escalabilidad
 - Transparencia
- ◆ Problemas a resolver
 - Asociación de nombres con recursos (Nombramiento)
 - Más escala
- ◆ ¿Cuál es la calificación según Enslow?

Ejemplo 4: Sistema multimedia (1)

◆ Sistema Pandora de vídeo conferencia



Ejemplo 4: Sistema multimedia (2)

- ◆ Requerimientos tiempo real:
 - Flujo continuo de datos (vídeo y audio)
 - Retardo de señal limitado (Ej. 100 msec para el audio)
- ◆ Requerimientos materiales:
 - Gran ancho de banda
 - Hardware especial (cámaras vídeo, pto. flotante, etc.)
 - ATM
- ◆ Técnicas interesantes aplicadas:
 - Flujos de red prioritarios (audio - vídeo - otros)
 - Calidad del servicio
- ◆ ¿Cuál es la calificación según Enslow?