

# MODULO 1. Antecedentes Básicos de los Sistemas Distribuidos



## Introducción

*CONOCER LAS CARACTERÍSTICAS DE LOS  
SISTEMAS DISTRIBUIDOS Y ENTENDER LA  
FORMA EN QUE ESTÁN CONSTITUIDOS.*

ING. CARMEN GUTIÉRREZ

# Introducción



- A mitad de la década de 1980 los avances tecnológicos cambiaron y revolucionaron la era de las computadoras con dos grandes desarrollos.
  - Desarrollo de microprocesadores
  - Desarrollo de redes



# El desarrollo de poderosos microprocesadores



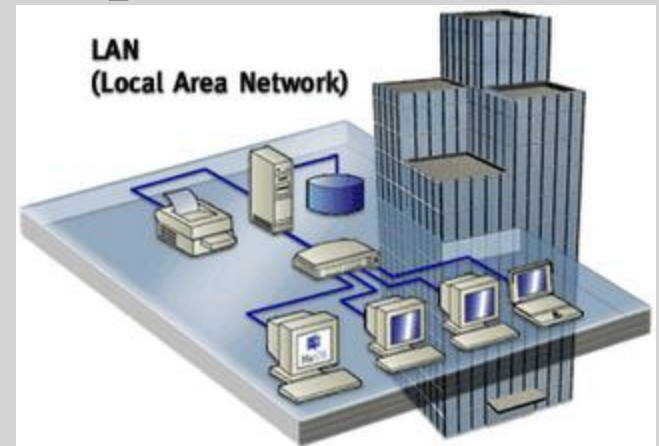
- En principio se disponía de máquinas de 8 bits pero pronto se volvieron comunes las CPU de 16, 32 e incluso 64 bits.



# El segundo desarrollo: la invención de redes



- Las redes de área local (LAN) permiten conectar docenas e incluso cientos de máquinas dentro de un edificio.
- Las redes de área amplia (WAN) permiten que millones de máquinas en toda la Tierra se conecten con velocidades que varían de 64 Kbps





- El resultado de estas tecnologías es que hoy en día no sólo es posible sino fácil reunir sistemas de cómputo compuestos por un gran número de CPU conectados mediante una red de alta velocidad.



- Estos reciben el nombre genérico de sistemas distribuidos en contraste con los sistemas centralizados anteriores (o sistemas con sólo un procesador) que constan de un CPU su memoria sus periféricos y algunas terminales.

# ¿QUÉ ES UN SISTEMA DISTRIBUIDO?



Un sistema distribuido es una colección de computadoras independientes que aparecen ante los usuarios del sistema como una única computadora.



# Aspectos que involucra



- El primero se refiere al hardware: las máquinas son autónomas .
- El segundo se refiere al software: los usuarios piensan que el sistema es como una única computadora.







- Cada estación de trabajo personal podría existir una pila de procesadores en el cuarto de máquinas que no estén asignados a usuarios específicos.
- El sistema podría tener un sistema de archivos único con todos los archivos accesibles desde todas las máquinas de la misma forma y con el mismo nombre de ruta de acceso .
- Cuando el usuario escriba un comando el sistema podría buscar el mejor lugar para ejecutarlo tal vez en la propia estación de trabajo del usuario o en una estación de trabajo inactiva.

# EJEMPLO



- Piense en un enorme banco con cientos de sucursales por todo el mundo.
- Cada oficina tiene una computadora maestra para guardar las cuentas locales y el manejo de las transacciones locales





- Además cada computadora tiene la capacidad de comunicarse con las de otras sucursales y con una computadora central en las oficinas centrales.
- Si las transacciones se pueden realizar sin importar dónde se encuentre el cliente o la cuenta y si los usuarios no observan diferencia alguna entre este sistema y el antiguo centralizado que ha remplazado también se podría considerar como un sistema distribuido

# Sistemas centralizados



## Ventajas:

- Un punto de control. Mayor control de seguridad y protección de la información en un solo punto.
- Fácil de mantener. Empresa con muchos cambios de requerimientos. Fácil despliegue de los cambios. Soporte en un solo punto.
- Tomas de decisiones. Esta arquitectura es primordial en las tomas de decisiones centralizadas, en otro punto llamado de lógica de negocio de la empresa centralizada.

# Desventajas:



- Interfaz de usuario poco llamativo. Por el uso de la red amplia, se evita tener pantalla con imágenes. Debe controlar el uso de ancho de banda de la red.
- Velocidad de repuestas lenta, dependiendo la conexión de la red a la central.
- Debe haber mecanismo de respaldo o copia del sistema centralizada en caso de contingencia muy estricta, muere el sistema central, muere el sistema a nivel general.
- Crecimiento depende de los equipos que lo soporta. Si la empresa crece en forma exponencial por ejemplo, el equipo debe ser cambiado al no tener estos estudios de crecimiento o la actualización de equipo debe ser constante.

# Sistemas distribuidos



## Ventajas:

- Aumenta la confiabilidad al sistema. Esta arquitectura tiene redundancia, al fallar uno de los sistemas, las demás siguen funcionando.
- Crecimiento de la empresa es soportable. Se realiza tantas copias del sistema en diferentes nuevos lugares geográficos.
- Tomas de decisiones locales. La lógica de negocio y las tomas de decisiones en cada lugar es independiente uno del otro.



- Distribución de datos. No está centralizada los datos de la empresa. Si la lógica de negocio en tener datos locales en las tomas de decisiones es factible, si no, irse a un sistema centralizado.
- Uso de ancho de banda local. Permite tener interfaz muy amigable o vistosa. Solo se consume el ancho de la banda de una red local.
- Velocidad de respuesta rápida, si los datos están en la red local.

## Desventajas:



- Soporte local de la tecnología. Cada lugar geográfico debe tener su personal que soporte tecnológicamente esta arquitectura.
- Una mala distribución de los datos, es peor que un sistema centralizado, uso en exceso de la red amplia.
- Costo y complejidad del software.
- Costo en llevar los cambios del software a cada lugar, en caso de mantenimiento.
- Integridad de los datos es más difícil de controlar.
- Uso de otra área de la tecnología, software de seguridad, protección y de redes.





- Se distinguen entre los *sistemas distribuidos* *diseñados para* que muchos usuarios trabajen en forma conjunta y los *sistemas paralelos* cuya meta es lograr la máxima rapidez en un problema como lo haría nuestra máquina de 500 000 MIPS (millones de instrucciones por segundo)



- Otro sistema inherente distribuido es lo que se denomina con frecuencia un trabajo cooperativo apoyado por computadora en donde un grupo de personas localizados a cierta distancia entre sí trabajan juntos.
- Los juegos cooperativos apoyados por computadora en donde los jugadores de diversos lugares juegan entre ellos en tiempo real.

# CONCEPTOS DE HARDWARE



# SISD



- (Single Instruction Single Data).
- Todas las computadoras tradicionales de un procesador caen dentro de esta categoría desde las computadoras personales hasta las grandes mainframes.

# SIMD



- Single Instruction Multiple Data
- Estas máquinas son útiles para los cálculos que repiten los mismos cálculos en varios conjuntos de datos.

# MISD



- Multiple Instruction Single Data
- Ninguna de las computadoras conocidas se ajusta a este modelo

# MIMD



- Multiple Instruction Multiple Data
- Significa un grupo de computadoras independientes cada una con su propio contador del programa y datos.
- Todos los sistemas distribuidos son MIMD.



- Todas las computadoras MIMD que tienen memoria compartida que por lo general se llaman multiprocesadores y aquellas que no que a veces reciben el nombre de multicomputadoras.
- En ciertos sistemas las máquinas están fuertemente acopladas y en otras están débilmente acopladas.





- Sistema fuertemente acoplado
- Sistema débilmente acoplado

# Sistema fuertemente acoplado



- El retraso al enviar un mensaje de una computadora a otra es corto.
- La tasa de transmisión de los datos que se pueden transferir es alta.

# Sistema débilmente acoplado



- El retraso al enviar un mensaje de una computadora a otra es largo.
- La tasa de transmisión de los datos que se pueden transferir es baja.



- Los sistemas fuertemente acoplados tienden a utilizarse más como sistemas paralelos.
- Los débilmente acoplados tienden a utilizarse como sistemas distribuidos.

# Multiprocesadores con base en buses



- Tiene cierta cantidad de CPU conectados a un bus común junto con un módulo de memoria.
- Un bus típico tiene:
  - 32 o 64 Líneas de direcciones
  - 32 o 64 líneas de datos
  - 32 o más líneas de control



- **Para leer una palabra de memoria un CPU:**
  1. Coloca la dirección de la palabra deseada en las líneas de direcciones del bus.
  2. Coloca una señal en las líneas de control adecuadas para indicar que desea leer.
  3. La memoria responde y coloca el valor de la palabra en las líneas de datos para permitir.
  4. La lectura de ésta por parte del CPU solicitante.
  
- **La escritura funciona de manera similar.**

- El problema con este esquema es que si sólo se dispone de 4 o 5 CPU el bus estará por lo general sobrecargado y el rendimiento disminuirá en forma drástica.
- La solución es añadir una memoria caché de alta velocidad entre el CPU y el bus.



Figura 7.1: Multiprocesadores con base en un bus.

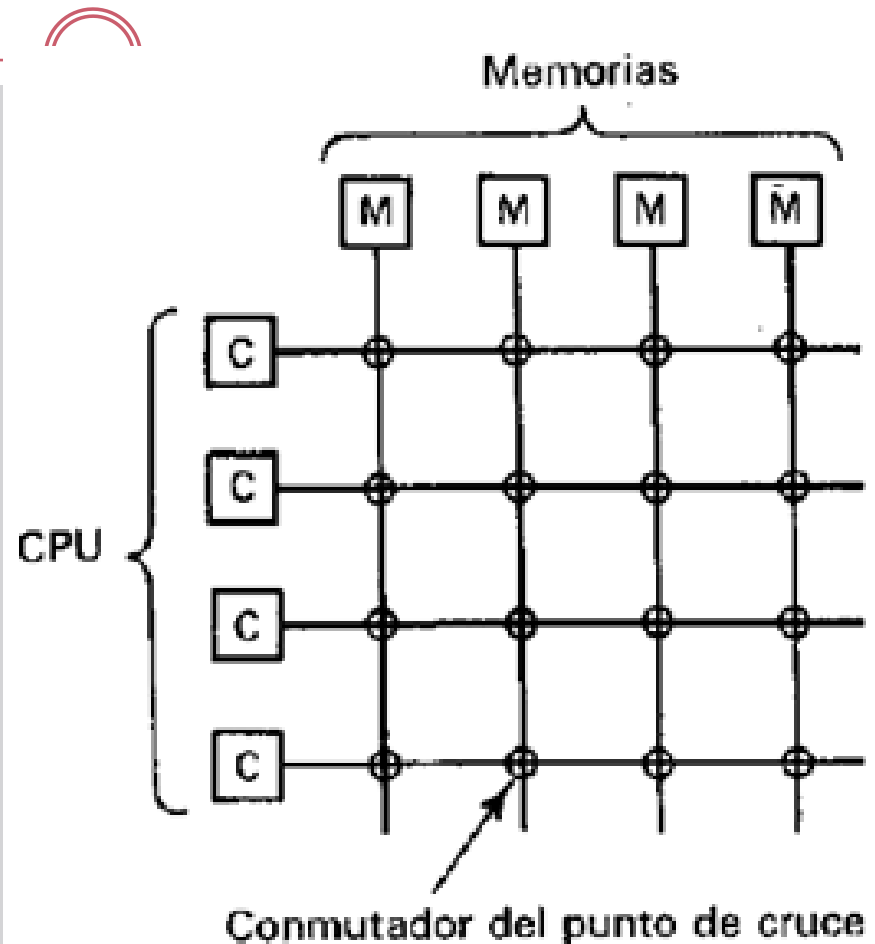
# Multiprocesadores con conmutador



- Para construir un multiprocesador con más de 64 procesadores, es necesario un método distinto para conectar cada CPU con la memoria.
- Una posibilidad es dividir la memoria en módulos y conectarlos a las CPU con un conmutador de cruceta.



- Cada CPU y cada memoria tiene una conexión que sale de él.
- En cada intersección está un delgado conmutador de punto de cruce electrónico que el hardware puede abrir y cerrar.



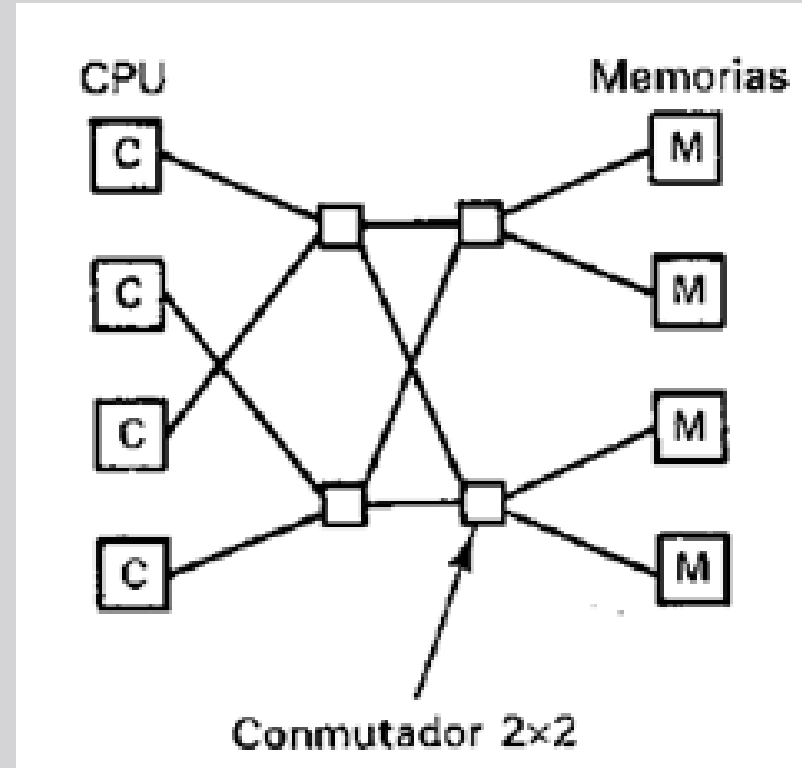


- La desventaja del conmutador de cruceta es que con  $n$  CPU y  $n$  memorias, se necesitan  $n^2$  conmutadores en los puntos de cruce.
- Si  $n$  es grande, este número puede ser prohibido.

# Red Omega



- Utiliza conmutadores 2x2, cada uno de los cuales tiene dos entradas y dos salidas.
- Se necesitan  $(n \log_2 n)/2$  conmutadores.
- Cada conmutador puede dirigir cualquiera de las entradas en cualquiera de las salidas.



# Máquina NUMA

(Non Uniform Memory Access )

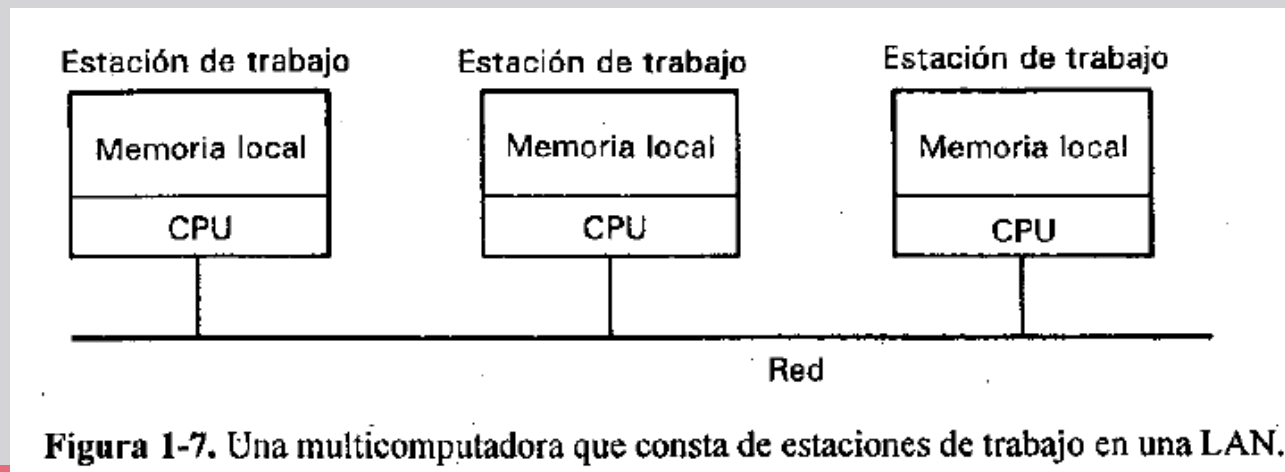


- Diseño de memoria donde la memoria se accede en posiciones relativas de otro proceso o memoria compartida entre procesos.
- Cada CPU tiene asociada cierta memoria.
- Cada CPU puede tener un rápido acceso a su propia memoria local, pero será más lento el acceso a la memoria de los demás.

# Multicomputadoras con base en buses



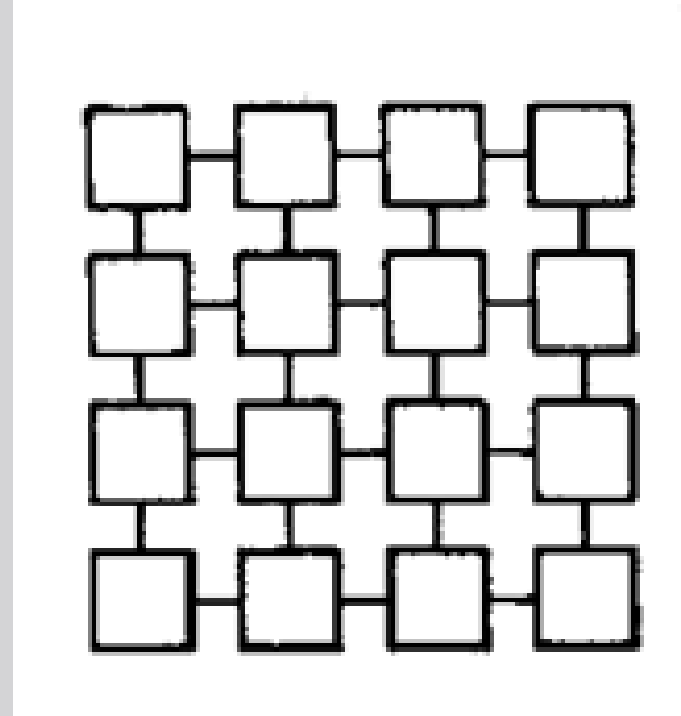
- Cada CPU tiene conexión directa con su propia memoria local. El problema es la forma en que los CPU se comunicarán entre sí. Se necesita cierto esquema de interconexión.

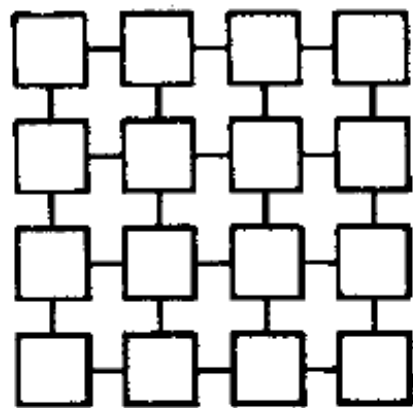


# Multicomputadoras con conmutador

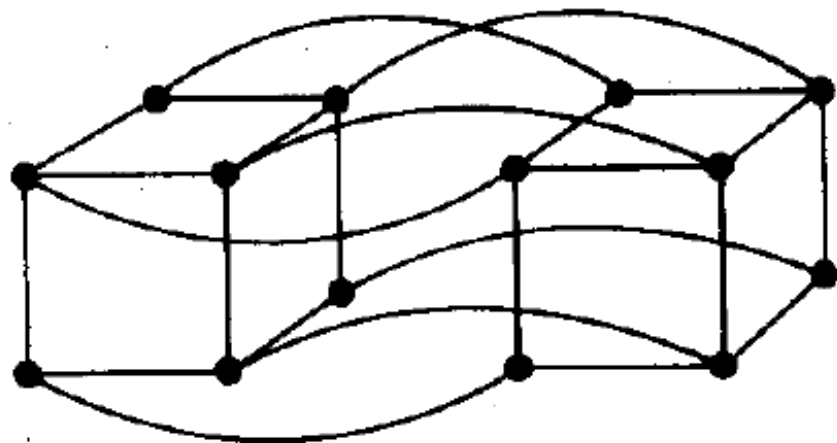


- Las retículas son fáciles de comprender y se basan en las tarjetas de circuitos impresos.
- Se adecuan mejor a problemas con naturaleza bidimensional inherente, como la teoría de gráficas o la visión (por ejemplo, los ojos de un robot o el análisis de fotografías).





(a)



(b)

**Figura 1-8.** (a) Retícula. (b) Hiper cubo.



- Un hipercubo es un cubo  $n$ -dimensional
- Cada vértice es un CPU. Cada arista es una conexión entre dos CPU.
- Se conectan los vértices correspondientes de cada uno de los cubos.
- Para extender el cubo a 5 dimensiones, podríamos añadir a la figura otro conjunto de dos cubos conectados entre sí y conectar las aristas correspondientes en las dos mitades, y así en lo sucesivo. Para el caso de un hipercubo  $n$ -dimensional, cada CPU tiene  $n$  conexiones con otras CPU.

