

# Clase 9- multiprocesadores

miércoles, 28 de mayo de 2025 18:06

## GPU procesadores específicos

Fueron haciendo más chiquitas las cosas electrónicas y se mejoró un montón pero no se puede mejorar mucho más.

Podría hacer más grande el proce pero tardaría más la señal en ir de un lado al otro.

Tengo muchos transistores juntos en muy poquito espacio.

Con muchas cpus que operen rápido, llego a la velocidad que necesito

Muchos procesos están bueno pero hay que darle bola a la coordinación

Con varias cpu, se puede curear todo. Yo podría asignarle a cada tarea una cpu específicamente. No es muy eficiente

Coordinador para repartir tareas

## Varios esquemas

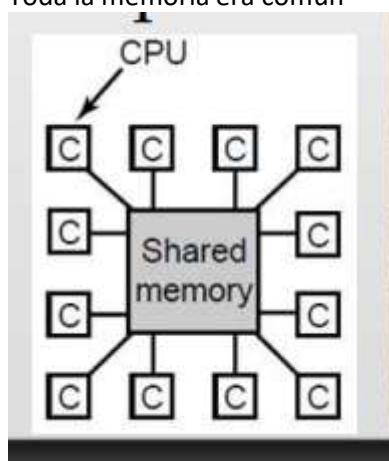
### 1- multiprocesadores con memoria compartida:

Tienen un único bus físico, para acceder entran a un solo espacio lógico de direcciones,

Con un solo bus, solo una puede leer o escribir.

Si metemos otro bus, hay que sincronizar eso

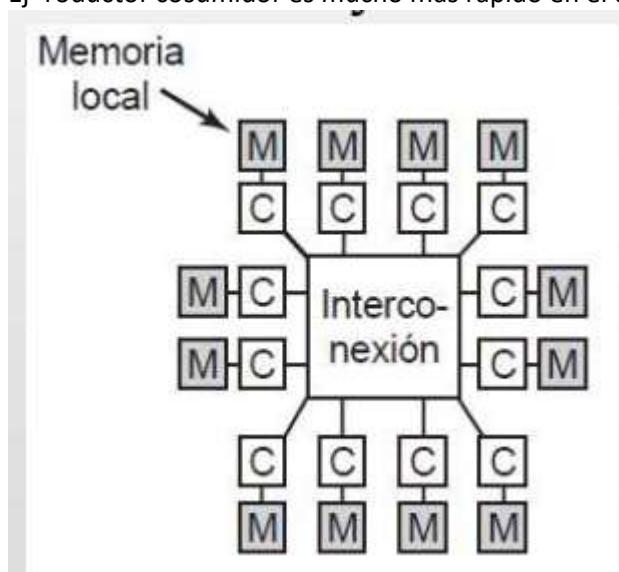
Toda la memoria era común



### 2- multicomputadora con memoria independiente/pasaje de mensajes:

Tienen que hacer pasaje de mensajes si quieren dato de otra cpu,

Ej `roductor consumidor es mucho más rápido en el de arriba.



Los planificadores tienen que adaptarse

Fuertemente acoplado: la idea es que una tarea que inicia termine de ejecutar en el mismo cpu

### 3- sistemas distribuidos

Conecta sistemas mediante una red

Cuando queda chica la arquitectura anterior. Quiero más de 18tb de ram, no me da.

No tengo un fierro con muchas cpus y ram, muchos fierros 😊

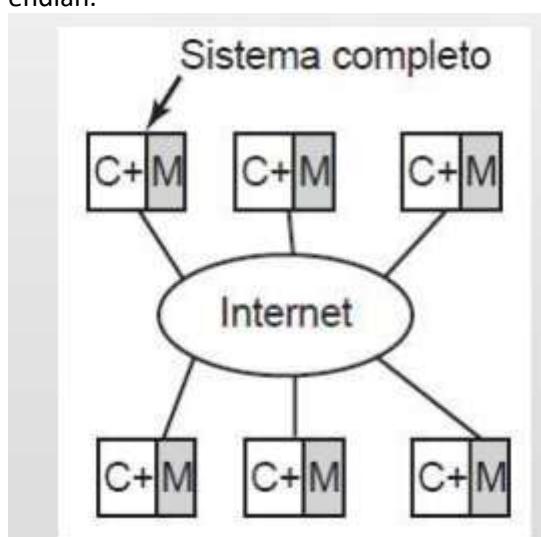
Conecto computadoras con velocidad rápida.

Cada una con memoria propia, cpu propia. Pasaje de mensajes.

Heterogéneos. No necesito que todos los nodos sean iguales.

Sacrificas el tiempo que tardas para mandar los datos por más procesamiento

Capa de transporte se ocupa de traducir a big endian si el sistema usa little porque la red usa big endian.



### Chips multinucleo:

Meter en una misma pastilla más de un núcleo.

Lo que andes hacia con vairas pastillas distintas hacerlo con una sola.

El software tiene que adaptarse al hardware nuevo.

### Sistema Operativo:

Adaptar al sistema operativo cuando tengo muchas cpu

Sincronizar bien para que no escriban lo mismo al mismo tiempo.

Dos procesos no tienen que acceder al mismo dispositivo de entrada salida

### 1- cada sp con un SO

Dividimos la memoria para cada cpu y una copia de cpu privada.

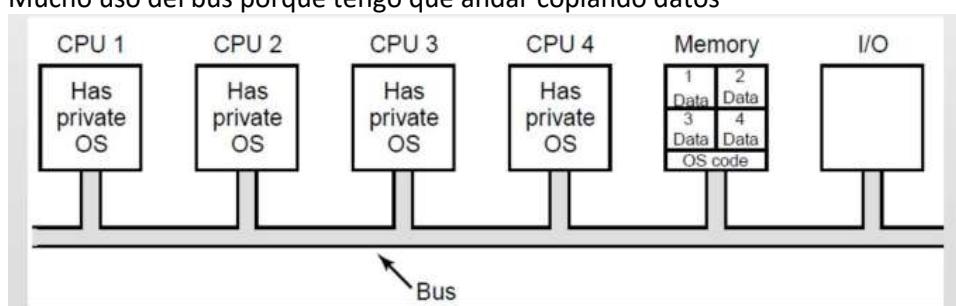
Cada uno tiene su cola de listos y demás

Mucho desbalance

NO comparten páginas de memoria, cache de disco, etc.

Inconsistencia de disco: protocolo de coherencia de cache. Chances de que una cpu intente leer algo que está viejo.

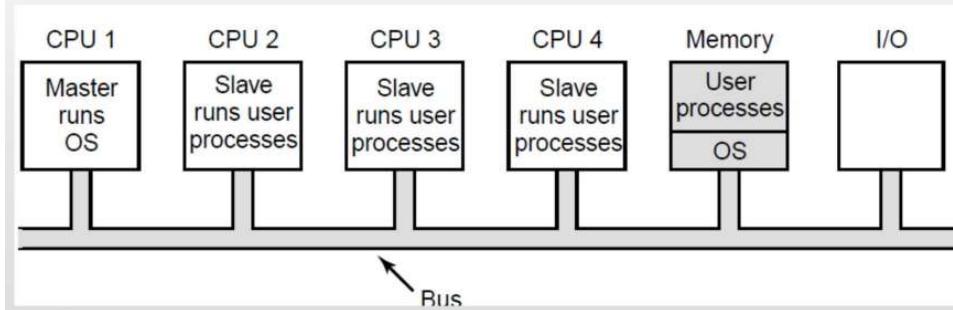
Mucho uso del bus porque tengo que andar copiando datos



### Maestro esclavo:

Hay una cpu en la que solo se ejecuta el SO.

En el resto procesos de usuario



Medio una cagada si hay muchas interrupciones ej pq el cpu del proce queda ocuado

### SMP- multiprocesadores simétricos

Una copia del so en memoria, cualquier cpu lo puede ejecutar.

CUando se hace una sys call, ejecuta la CPU que la invocó

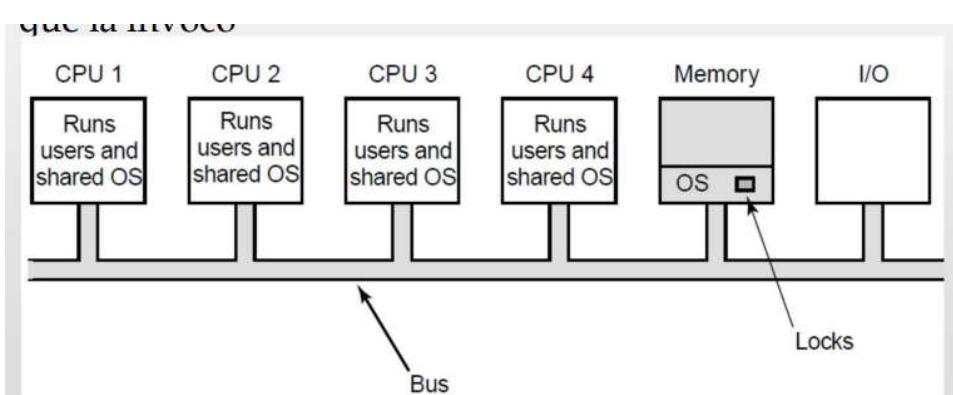
Cada cpu ejecuta a medida que necesita

Volves a tener temas de sincronización. Necesitas que no lean dos instancias del so el mismo espacio.

Ej si dos procesos en cpu terminan al mismo tiempo y tiene que ir a la cola de listos que no lean lo mismo

Necesitas ayuda del hardware pq con software no podes asegurarte que esté bien.

El hardware bloque que no puedas leer en simultáneo lo mismo.



### Planificando que proceso sigue:

Si es uniprocesador es ez, solo tengo que decir en que hlo o proceso se ejecuta.

Si es multiprocesador sumo en que cpu se ejecuta.

Puede haber hilos que conviene que se ejecuten en conjunto, tareas que dependen de las otras. Ej compilar con 4 hilos. Quiero que se ejecuten en conjunto y que no interfieran otras cosas

### Planificando hilos independientes:

Llega y ejecuto

No me importa cuál cae en qué cpu.

Lo único importante es que si lo salgo y lo vuelta a meter vuelva a la misma cpu para aprovechar cache,

Hilos que trabajan en conjunto:

En varias cpus en conjunto. El grupo se planifica si hay cpus libres para cada hilo del grupo, sino espera.

Si tengo un solo cpu disponible le meto cualquier hilo independiente.

Cómo planifica para que se ejecute en la menor cantidad de tiempo posible, a veces no da lo mismo

Planificación por pandilla:

Todos los hilos relacionados son una pandilla.

Todos se ejecutan a la vez, arrancan y terminan sus intervalos en conjunto

## Sistema multicamputadora: Cluster

Cuando me quedé corta de cpu

No comparten memoria

Idealmente homogeneas.

Usan pasaje de mensajes

Send y receive puede tener problema si son heterogéneos, la idea es que sean iguales pero no pasa

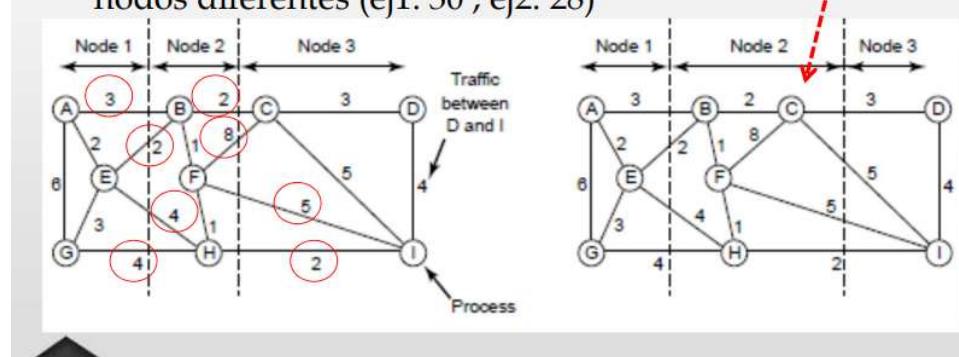
Planificación: un nodo no toma procesos de otro

Puede haber desbalance de carga, pasar de un nodo a otro no es tan fácil como pasar de una cpu a otra.

Hay que tener cuidado al repartir cargas:

Me conviene que sin dos procesos estén relacionados se ejecuten en el mismo nodo.

Se arma un grafo pesado de cuánto me cuesta comunicar cada cosa y  
nodos diferentes (ej. 30, 20)



Me importa el peso de la comunicación entre nodos.

En la de la derecha, moviendo C al nodo 2 y me da 28 en lugar de 30 la comunicación.

Se le suma ocuparse de eso a la planificación.

## Sistemas distribuidos

Colección de computadoras independientes que se muestran como si fueran 1 solo.//

Computadoras en espacios separados se comunican para coordinar sus acciones por pasaje de mensajes.

Parecido a multicamputadoras, menos acoplados, pueden estar en lugares distintos.

Cada nodo puede ejecutar un SO y hardware distinto. Puede ser heterogéneo.

Se tienen que poder comunicar entre sí para mostrarse como uno solo.

**Middleware:** capa de software que está encima del SO y permite uniformidad entre los sistemas -- normaliza el formato de mensajes. Cuando el dato venga a la app, llegue siempre de la misma manera. Ej http: da las reglas para que cualquiera pueda mostrar cosas.

Conjunto de reglas que se encargan de hablar un lenguaje común