Administración de Memoria Compartida Distribuida

Diseñar una máquina donde muchos procesadores utilizan la misma memoria en forma simultánea es muy difícil. Los multiprocesadores de buses no pueden usarse con más de una docena de procesadores sin que sufran degradamientos considerables, ya que el bus es un cuello de botella, y los multiprocesadores con conmutador pueden escalarse a grandes sistemas, pero son lentos, caros , complejos y difíciles de mantener.

Sin embargo, las grandes multicomputadoras son fáciles de construir. Se puede tomar un número casi ilimitado de computadoras sencillas y unirlas por medio de una red.

A diferencia de los multiprocesadores, donde una parte importante del diseño se dedica a la sincronización con semáforos, memoria compartida y secciones críticas, en las multicomputadoras la labor de abstracción principal se centra en el **paso de mensajes**, manejados por operaciones de entrada/salida.

Esta transferencia de mensajes trae consigo varios aspectos delicados, como el flujo de control, la pérdida de mensajes, el uso de buffers y el bloqueo. A pesar del trabajo desarrollado, la programación con transferencia de mensajes aún es difícil.

Recordamos algunos esquemas de multiprocesadores con memoria compartida:

- Multiprocesadores basados en bus.- Con y sin ocultamiento. Uso de caché. Memoria centralizada
- Multiprocesadores basados en anillo.- Memnet (p. 298, Tanenbaum). No hay memoria centralizada
- Multiprocesadores con conmutador.- Mejor ancho de banda. Malla. Red Omega. [Ejm.: Dash - Directory Architecture for Shared Memory, p. 302]. Ocultamiento de Hardware.Direcciones virtuales de memoria.
- Multiprocesadores NUMA (Acceso No Uniforme a Memoria).- Como un UMA, la máquina tiene un espacio de direcciones virtuales visibles para todos los CPUs. Cuando cualquier CPU describe un valor en la localidad *a*, una lectura posterior por un procesador diferente regresará el valor recién escrito. [Ejm.: Cm*, BBN Butterfly]
    - Es posible el acceso a memoria remota
    - El acceso a memoria remota es más lento que el de memoria local
    - El tiempo de acceso remoto no se oculta mediante el uso de cachés
- Las máquinas con memoria compartida distribuida (DSM), no pueden hacer llamadas a memoria en forma directa. Cuando un CPU hace referencia a una palabra en un espacio de direcciones que tiene un respaldo de página localizado actualmente en una máquina diferente, ocurre un señalamiento en el sistema operativo y la página solicitada debe ser buscada por el software.

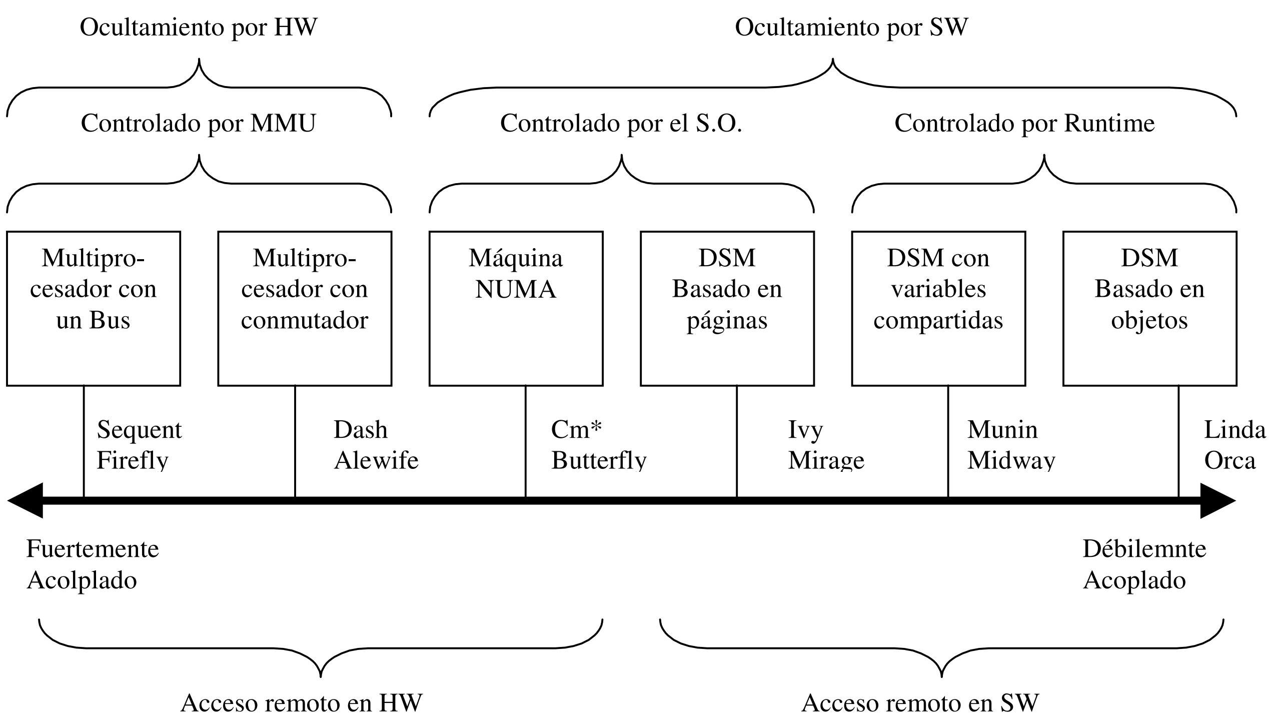
  El S.O. adquiere la página necesaria al enviar un mensaje a la máquina donde reside en la actualidad y solicitarla. Así, tanto la colocación como el acceso se hacen por software. (Ejm: IVY, Mirage).

- Existen también máquinas que comparten sólo un a porción de su espacio de direcciones, como variables compartidas y estructuras de datos (Munin, Midway). Se requiere info. adicional para determinar cuáles variables son compartidas y cuáles no.

- Finalmente, se tienen sistemas con DSM basada en objetos. A diferencia de los anteriores, los programas no pueden sólo accesar los datos compartidos; tiene qué recoger métodos protegidos, lo que aumenta el control. Todo aquí se hace en software (Ejm: Linda, Orca)

La diferencia real entre los multiprocesadores y los DSM es que, en los primeros, se puede tener acceso a datos remotos por hardware mediante sus direcciones, y en los últimos siempre se necesita la intervención del software.

Máquinas de Memoria Compartida



Modelos de consistencia de memoria

Un modelo de consistencia es un contrato entre el software y la memoria. Dice que si el SW acepta seguir ciertas reglas, la memoria promete trabajar en forma correcta (!¡). Si el SW viola estas reglas, todo acaba y ya no se garantiza que la operación de memoria sea correcta.

Existen varios modelos básicos de consistencia:

<table>
<tr><td rowspan="5">Sin operaciones de sincronización</td><td>Consistencia estricta</td><td>Ordenamiento absoluto con respecto al tiempo de todo lo relacionado con el acceso a memoria (casi imposible en un sistema distribuido, nunca se utiliza)</td></tr>
<tr><td>Consistencia secuencial</td><td>Todos los procesos ven todos los accesos compartidos en el mismo orden (tiempos lógicos. Desempeño pobre)</td></tr>
<tr><td>Consistencia causal</td><td>Todos los procesos ven todos los accesos compartidos relacionados causalmente en el mismo orden (causalidad)</td></tr>
<tr><td>Consistencia del procesador</td><td>Consistencia PRAM + coherencia de memoria</td></tr>
<tr><td>Consistencia PRAM (Pipelined RAM)</td><td>Todos los procesos ven las escrituras de cada procesador en el orden en que fueron ejecutadas. Las escrituras de procesadores diferentes podrían no ser vistas en el mismo orden</td></tr>
<tr><td rowspan="3">Con operaciones de sincronización</td><td>Consistencia débil</td><td>Los datos compartidos sólo pueden considerarse como consistentes después de una sincronización (uso de variables de sincronización)</td></tr>
<tr><td>Consistencia de liberación</td><td>Los datos compartidos son consistentes al salir de una región crítica</td></tr>
<tr><td>Consistencia de entrada</td><td>Los datos compartidos pertenecientes a una región crítica son consistentes después de salir de una región crítica</td></tr>
</table>

Manejo de Páginas de Memoria Distribuida

En sistemas DSM, no existe memoria global. Cada procesador tiene su memoria local y la memoria se comparte a través de páginas o bloques de tamaño constante.

Un procesador puede poseer una o varias páginas y cuando otro procesador requiere información de una página que no tiene, debe enviarse la página por la red.

Normalmente existe una copia de cada página que es pasada entre los procesadores, pero a veces puede permitirse la generación de copias si sólo se usan para lectura.

Conceptos útiles en manejo de páginas

- Réplica.- Existen o no copias de las páginas de memoria
- Granularidad.- Se transmiten páginas completas o sólo parte de ellas
- Sincronización.- Cuándo copiar o transmitir páginas (modelos de consistencia)