

SISTEMAS OPERATIVOS II

INTRODUCCION

El método antiguo de tener un computador que realiza todas las tareas computacionales de una empresa ha sido reemplazado por otro, un gran número de computadoras separadas pero interconectadas, hace el trabajo estos sistemas de redes de computadoras.

Este término se adapta para definir un conjunto de computadoras autónomas interconectadas, la interconexión puede ser mediante cable de cobre, fibra óptica, microondas, rayos infrarrojos y los satélites de comunicación.

Los conceptos de red de computadoras y sistemas distribuidos son diferentes, la diferencia está en que un sistema distribuido es un conjunto de computadoras independientes que aparecen ante sus usuarios como un sistema de consistencia, los usuarios están expuestos a las máquinas reales y el sistema no hace ningún intento para que las máquinas se vea y actúen de manera similar, de hecho un sistema distribuido es un sistema de software construido sobre una red, ese software le da un alto grado de consistencia y transparencia por eso la diferencia entre una red de computadoras y un sistema distribuido está en el software, especialmente en el sistema operativo.

ESTRUCTURAS DE SISTEMAS DISTRIBUIDOS

Se define como un sistema de procesadores que se comunican entre sí a través de líneas de alta velocidad y no comparten ni la memoria ni el reloj (débilmente acoplados) estos procesadores pueden ser:

- Microprocesadores.
- Estaciones de trabajo.
- Minicomputadoras.
- Sistemas de propósito general.

Los nombres con los que se le conocen son:

- Sitios.
- Nodos.
- Computadores
- Máquinas (host), anfitriones, etc.

Un sistema de archivo ofrece servicios de archivos cuyos usuarios, servidores y dispositivos de almacenamiento están dispersos entre los sitios de un sistema distribuido.

En un sistema distribuido es el Sistema Operativo el encargado de proporcionar a los usuarios accesos a los diferentes recursos que el sistema ofrece. Estos recursos pueden ser tanto hardware o software.

SISTEMAS OPERATIVOS DE RED.

Cada computador posee su propio sistema operativo, los usuarios conocen la diversidad de las máquinas que están en la red y para acceder a los recursos deben hacer un acceso en forma explícita a la máquina correspondiente remota o haciendo transferencia de los datos requeridos a la propia máquina.

SISTEMAS DISTRIBUIDOS

En un sistema común compartido por una red de computadoras el usuario no necesita saber de la existencia de las máquinas en la red, accede a los recursos remotos de la misma forma que lo hace para los recursos locales.

Motivación

Las principales razones de construir un sistema operativo son:

- **Compartimiento de recursos.-** Los diferentes sitios de la red en general puede tener diferentes capacidades por ejemplo el usuario de sitio A podrá usar la impresora láser disponible en el sitio B y a su vez el usuario del sitio B puede acceder a un archivo ubicado en el sitio C.
- **Agilización de cálculos.-** Cuando es factible dividir un proceso en subproceso que puedan ser ejecutados en forma concurrente, se pueden distribuir los procesos en diferentes sitios disponibles y realizarlos simultáneamente, de la misma manera cuando un sitio está sobrecargado algunos trabajos podrían transferirse a otros sitios que estén disponibles.
- **Confiabilidad.-** En el caso de un sitio de un sistema distribuido falle hay la posibilidad de que otros sitios sigan funcionando, esto depende del grado de autonomía de las instalaciones o nivel de redundancia en el sistema (hardware o software o datos).
- **Comunicación.-** Además de la transferencia de mensajes un sistema distribuido permite funciones tales como transferencia remota, ingreso, correo y llamadas a procedimientos remotos, estos sin límites de distancias.

TOPOLOGÍA

Se refiere a la disposición física de 2 dispositivos y a las líneas de comunicación para elegir alguna configuración en particular, estos son los siguientes criterios:

- **Costo básico.-** Se refiere al costo de vincular los sitios del sistema.
- **Costo de comunicación.-** El tiempo que tarda un mensaje entre A y B.
- **Confiabilidad.-** Si algún enlace del sistema o nodo falla afecta a los restantes.

CONFIGURACION MÁS CONOCIDAS O MÁS COMUNES

- **Redes totalmente conectadas.-** En esta configuración cada sitio se encuentra conectado directamente con los demás sitios del sistema, esta configuración es la que ofrece mayor velocidad y confiabilidad pero el costo básico crece en proporción al cuadrado del número de sitios.
- **Redes parcialmente conectados.-** No todos los sitios de red un enlace directo entre sí el costo básico de esta configuración es menor, la comunicación sería más lenta porque un mensaje de un sitio a otro podría que tener que enviarse a través de varios sitios intermedios, la confiabilidad es menor, el fallo de un enlace podría dividir la red.
- **Redes jerárquicas.-** La organización es de forma de árbol, este es el modulo común en la redes corporativas cada sitio excepto la raíz tiene un padre único y un cierto números de hijos, el costo es generalmente menor que del esquema parcialmente conectado.
- **Redes Estrellas.-** Unos de los sitios del sistema se conecta a todos los demás sitios y ninguno de los demás sitios se conecta con otro, el costo básico es directamente proporcional a números de sitios, aunque el costo de comunicación es bajo, la rapidez puede afectarse cuando el nodo central se convierte en un cuello de botella.
- **Redes Anillos.-** Cada anillo se conecta físicamente a otros dos sitios, el costo básico tiene una relación lineal con número de sitios pero el costo de comunicación puede ser alto (tiempo).
- **Redes de bus multiacceso.-** Hay un solo enlace compartido que es el bus. Todos los sitios se conecta a través de este enlace, el costo de comunicación es muy bajo excepto si el enlace se convierte en un cuello de botella con un sitio central dedicado, esta configuración es similar a estrella, este modelo utiliza Ethernet.

- **Redes Híbridas.-** Es la conexión en sí de redes de distintos tipos por eje: en los sitios se podrían usar usarse un bus multiacceso y una red jerárquica.

TIPOS DE REDES

Los dos tipos de redes son:

- **Redes de área local LAN.-** Aparecieron como sustituto a los sistemas de macro computación grandes (Mainframes). Las redes LAN generalmente cubre áreas geográficamente pequeñas, sus enlaces de comunicación son más rápidos y con menos tasas de errores, se utiliza cable de alta calidad (fibra óptica, cable coaxial, par trenzado), la configuraciones más comunes son bus multiacceso, anillo, estrella, la velocidad de comunicación tiene un rango de 1Mbps hasta 1Gbps, una LAN podría estar compuesta de varios computadores diferentes desde macro computadores hasta laptops, tienen diferentes dispositivos periféricos y puertos para accesos a otras redes.
- **Redes de área extensa WAN.-** Permite una amplia comunidad de usuarios, compartir diferentes hardware y software de forma cómoda y económica. Internet es una red de redes que está compuesta de miles de sistemas de computación, otras redes comerciales son: TELNET (EEUU), DATAPAC (CANADA) por su alta aplicación, los enlaces son relativamente lentos y poco confiables, pueden ser líneas telefónicas, microondas y canales de satélites, se controla mediante canales que son procesadores de comunicación los cuales definen la interfaz, los computadores anfitriones(host) generalmente son diferentes en cuanto a tipo, rapidez palabra, sistema operativos, etc. El uso de modem es imprescindible por lo que estas redes la velocidad de transmisión varia 1200 bps a 1Mbps.

COMUNICACIÓN

Son cinco los aspectos básicos para el diseño de una red de comunicación

1. **Asignación de nombres.-** El método que se utiliza actualmente es el servicio de nombres de dominios DNS el cual especifica la estructura de nombres de los anfitriones así como la relación entre nombres y direcciones por ejemplo: pep.tec.uagrm.edu, en este caso se refiere al anfitrión pepe de la facultad de tecnología de la uagrm, el sitio traduce las direcciones examinando los componentes de nombre del anfitrión en el orden inverso. Cada componente tiene un servidor de aquellos nombres que acepta devolviendo la dirección de este.
2. **Encaminamiento (routing).-** Cada sitio tiene una tabla de encaminamiento que indica las rutas alternativas que pueden usarse para enviar un

mensaje, en esta tabla también hay información de la velocidad y el costo de las distintas rutas, hay tres esquemas de encaminamiento (fijo, virtual y dinámico), un en-caminador puede ser un computador anfitrión o un dispositivo de propósito especial, el cual se encarga de encaminar los mensajes.

3. **Estrategias de paquetes.-** Consiste en el uso de mensajes de longitud fija llamado paquetes (tramas, datagramas).
4. **Estrategias de conexión.-** Se refiere a formas de conexión de pares de procesos en la red, los tres pares de esquemas más comunes son:
 - **Conmutación de circuitos.-** Consiste en el establecimiento de un enlace físico y permanente para la comunicación exclusiva de dos procesos.
 - **Conmutación de mensajes.-** Consiste en la asignación temporal del enlace físico durante el tiempo de transferencia de mensajes entre dos procesos, cada mensaje es un bloque de datos con información del sistema (fuente, destino, códigos de corrección de errores). Es posible el envío de muchos mensajes por el mismo enlace (de diferentes usuarios).
 - **Conmutación de paquetes.-** Cuando un mensaje lógico es grande es posible dividirlo en varios paquetes y enviarlos por separado de acuerdo a las posibilidades, por lo cual es necesario que cada paquete incluya dirección tanto de origen como de destino, además de los datos al llegar a destino deben re ensamblarse.
 - **Conflictos.-** La optimización de paquetes de la misma área por varios usuarios en forma simultánea puede causar conflictos sobre todos en las redes de anillos o los multi-accesos, las formas de evitar colisiones son las siguientes.
 - × **CSMA/CD Carrier sense multiple access with collision detection.-** Para transmitir los mensajes por un enlace un sitio deberá escuchar para comprobar si hay otra transmisión, si el enlace está desocupado puede ceder con la transmisión, en caso contrario debe esperar y continuar escuchando hasta que el enlace esté libre.
 - × **Paso de testigo.-** Es un tipo de mensaje llamado testigo (token) el cual circula continuamente por el sistema (estructura anillo), cuando un sitio desea transmitir información debe esperar la llegada del testigo, el sitio saca al testigo del anillo y transmite su mensaje y cuando finaliza

vuelve a colocar al testigo, la ventaja de una red con paso de testigo es que el desempeño es constante.

- × **Ranuras de mensaje.-** para la ranura de mensaje de longitud fija circulan continuamente en el sistema, cada ranura puede obtener un mensaje de tamaño fijo además de información de control, un sitio que requiera transmitir un mensaje debe esperar la llegada de una ranura vacía, luego inserta su mensaje en la ranura con la información de control correspondiente, como cada ranura solo puede enviar mensaje de longitud fija es posible utilizar varias ranuras.

5. Proceso clientes servidor.- En general un anfitrión de algún sitio (servidor) tiene un recurso que otro anfitrión en otro sitio (cliente) quiere usar, el propósito de un sitio distribuido es proporcionar un ambiente eficiente y cómodo. Para compartir los recursos existen diferentes definiciones para el proceso cliente/servidor y una de ellas es la siguiente:

- Un modelo de interacción entre proceso de software que ejecutan concurrentemente estos procesos clientes envían solicitudes a un proceso servidor que le responde con los resultados respectivos, los procesos servidores ofrecen servicios a sus clientes generalmente mediante un proceso específico que solo ellos pueden realizar, el proceso cliente libre de la complejidad y la sobrecarga de procesar la transacción puede realizar otras tareas, la interacción entre el procesos cliente y el servidor es un intercambio operativo y transaccional en el que el cliente es activo y el servidor es receptivo.

PROCESO CLIENTE SERVIDOR

La diferencia entre una configuración cliente servidor y la de cualquier otra distribución al sistema distribuido son gran confianza en depositar aplicaciones que son fáciles de usar en los sistemas de los usuarios, dando a los usuarios un alto grado de control respecto a tiempo y uso del computador.

- **Así como las aplicaciones se dispersan.-** se centraliza la base de datos corporativos, así como las muchas funciones de utilidad y de gestión de red.
- **Compromiso.-** Tanto de las organizaciones de usuarios como la de fabricantes hacia los sistemas abiertos y modulares.
- **El trabajo en red es fundamental para la operación.-** De este modo la función y la seguridad de la red tiene una prioridad alta en la organización y operación de los sistemas.

- **Proceso cliente servidor.-** gran confianza (control), dispersión de aplicaciones (Actualizar la base de datos).

CLASES DE APLICACIONES CLIENTE SERVIDOR

Existe una gran gama de posibles implementaciones que dividen el trabajo entre cliente servidor:

- **Proceso basado en el host.-** Es una maquina central (host), no es realmente un proceso cliente/servidor se refiere más bien a entorno tradicional de grandes sistemas en el que todo o casi todo el tratamiento se efectúa en un computador central, la interfaz del usuario generalmente es un terminal tonto.
- **Proceso basado en el servidor.-** La forma básica de configuración cliente/servidor es cuando el servidor es responsable de brindar una interfaz de usuario gráfica y casi todo tratamiento se efectúa en el servidor.
- **Proceso basado en el cliente.-** En el otro extremo casi todo cliente y unas de las más sofisticadas funciones de la base de datos reside el cliente.
- **Proceso cooperativo.-** El proceso de aplicación se hace en forma óptima, utilizando la potencia de la maquina cliente servidora y la distribución de los datos, esta configuración es más compleja de instalar y de mantener pero se obtiene una mayor eficacia.

Clases de aplicaicones Cliente/Servidor

- > **Proceso basado en el host** .- Grandes sistemas.
- > **Proceso basado en el servidor** .- Interfaz
- > **Proceso basado en el cliente** .- Proceso en el cliente.
- > **Proceso cooperativo.-** Potencia y datos distribuidos, mas complejos.

ESTRATEGIA DE DISEÑO

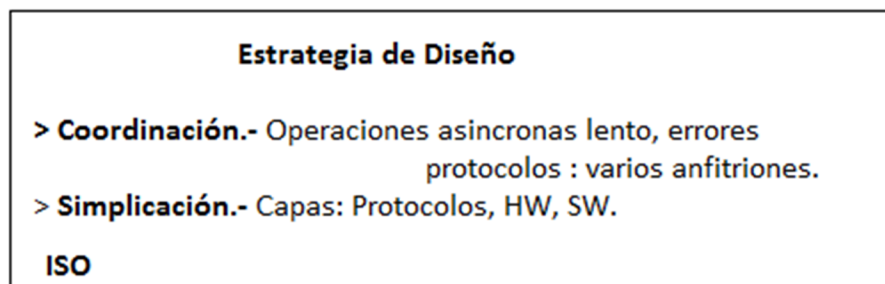
La complejidad en el diseño de una red de comunicaciones es la base de la coordinación de operaciones asíncronas que se comunican en un entorno potencialmente lento y propenso a errores, también se requiere protocolos para determinar nombres y anfitriones localizados, establecer conexiones, etc.

Se puede simplificar el diseño estableciendo varias capas, cada capa de un sistema se comunica en la capa equivalente de un sistema, cada capa puede tener sus propios protocolos y estos pueden implementarse en el hardware y software.

Las tres capas de nivel más bajo se implementa en el hardware y de acuerdo a la organización internacional de normas (ISO, OSI, IEEE).

Las diferencias de las capas son las siguientes:

1. **Capa Física.-** Maneja los detalles tanto mecánicos como eléctricos de transmisión física.
2. **Capa de enlace de datos.-** Se encarga de manejar las tramas en partes de longitud fija de los paquetes.
3. **Capa de red.-** Proporciona conexiones y encamina paquetes en la red de comunicaciones.
4. **Capa de transporte.-** Se refiere al acceso bajo nivel de la red y de transferencia de mensaje de cliente.
5. **Capa de sesión.-** Transporta sesiones o protocolo de comunicación.
6. **Capa de Presentación.-** Resuelve la diferencia de formato entre los diversos sitios de la red, tales como corrección de caracteres y de modo semiduplex y dúplex.
7. **Capa de Aplicación.-** Es la que interactúa directamente con los usuarios como ejemplo transferencia de archivos, protocolo de inicio de sesión remoto y correo electrónico.



El MODELO OSI

Formaliza uno de los trabajos anteriores de protocolo de red, pero se desarrolló a finales de 1970 y su aplicación en la actualidad ha disminuido, parte de la base del modelo OSI es la parte de la pila de protocolos.

1970 IP (Internet protocol).

Los servicios se implementan encima de IP, mediante protocolo de diagrama de usuario

UDP (Uses Datagrama Protocol).

Protocolo de control de transmisión

TCP (Transmisión control protocol)

La pila de Protocolos TCP/IP tiene menos capa que el modelo OSI y en teoría puesto que este modelo combina varias funciones una cada capa es más difícil de implementar pero más eficiente en el trabajo de redes OSI.

ESTRUCTURAS DE SISTEMAS DISTRUIDOS

- **Sistemas operativos de Red.-** Proporciona un entorno donde los usuarios se dan cuenta de la multiplicidad de máquina y pueden acceder a recursos remotos ya sea iniciando una sesión en la maquina remota o transfiriendo dato a la maquina remota.
- **Sistemas operativos Distribuidos.-** En este ambiente los usuarios acceden a recursos remotos tal como lo hacen para recursos locales, la migración de datos y procesos de un sitio a otro está bajo el control del sistema operativo distribuido, para la migración de datos hay dos estrategias.
 - a. Transferir todo el archivo desde el sitio remoto al sitio local y desde ese momento todos los accesos a los archivos serán locales. Cuando el usuario ya no necesita acceder al archivo, se enviara de regreso una copia de archivo (si es que se modifica).
 - b. Transferir solo aquellas porciones de archivo que se necesita, si se requiere otra porción, después se efectúa otra transferencia, cuando el usuario ya no necesita más el archivo, en el que una parte de este quede modificado se lo devuelve al sitio remoto (paginación por demanda).

MIGRACION DE PROCESO

- a) **Balanceo de carga.-** Los procesos o sub-procesos pueden distribuirse en la red con el objeto de equilibrar la carga de trabajo.
- b) **Agilización del cómputo.-** Cuando un proceso puede dividirse en varios sub-procesos los cuales pueden ser ejecutado simultáneamente, podría ser posible el tiempo de retorno total del proceso.
- c) **Preferencia de Hardware.-** El proceso podría tener características que lo hacen más apropiado ejecutarse en un procesador agilizado como por ejemplo la conversión de matrices en procesador de arreglo en lugar de un microprocesador.
- d) **Diferencia de Software.-** El proceso podría requerir software que está disponible en cierto sitio, y no puede ser transferido o resulta más económico mover el proceso.
- e) **Acceso a dato.-** De la misma forma que la migración de cómputos, si los datos que se usan en el cómputo, podría ser más eficiente ejecutar el proceso de forma remota en vez de transferir todos los datos a la maquina local.

SERVICIOS REMOTOS

Cuando un usuario requiere acceder a datos de algún otro sitio, por ejemplo el número total de líneas, palabra y caracteres de un archivo remoto, un servidor remoto del sitio atiende la solicitud, para ello accede al archivo, calcula el resultado y transfiere la información al usuario.

Nota: Migración -> procesos -> lo hace el usuario. Servicio remoto ya está preparado.

SISTEMAS DE ARCHIVOS DISTRIBUIDOS DFS (Distribución file system)

Es una implementación del modelo clásico del sistema de archivo al uso compartido, donde múltiples servicios comparten archivos y recursos de almacenamiento, el objetivo de DFS es agregar el mismo tipo de compartimiento, cuando el archivo se halla disperso físicamente entre los sitios de un sitio distribuido.

Un DFS es un sistema de archivo cuyo cliente servidores y dispositivos de almacenamiento están dispersos entre las máquinas de un sistema distribuido, hay configuraciones en los que los servidores se ejecuta en máquina dedicadas a otras en la que una máquina es servidor o un cliente.

La medida más importante de un DFS es el tiempo que toma satisfacer diversas solicitudes de servicios, el desempeño de un DFS se puede ver como otra dimensión de la transferencia de los DFS es decir:

Que el desempeño ideal de un DFS seria comparable al de un sistema de archivo convencional.

NOMBRES Y TRANSFERENCIAS

Los nombre establece una correspondencia multinivel entre objeto lógico y físico , el usuario usando un nombre textual el cual se transforma en un identificador numérico de más bajo nivel correspondiente a bloques de disco , está correspondencia , proporciona a los usuarios una atracción ocultando los detalles de la ubicación del archivo en disco.

En D.F.S. Se añade otra dimensión a la abstracción que es la de ocultar donde se encuentra el archivo en la red, esto ofrece la posibilidad de replicar archivos en varias ubicaciones.

La abstracción oculta al usuario tanto la existencia con la ubicación de múltiples copias.

Nota.- En un sistema → añade que el usuario no sabe cuántos archivo hay.

ACCESO DE ARCHIVO REMOTO

Una forma de efectuar transferencia de datos es utilizando el mecanismo de servicio remoto, mediante el paradigma llamada al procedimiento remoto (R.P.C.).

Para un mayor desempeño se puede usar cache's , como se hace en un sistema convencional en los D.F.S. , la meta es reducir tanto el tráfico de la red, como en la entrada y salida del disco.

Los aspecto de implementación de los cache's en los D.F.S. son los siguientes :

- **Esquema de uso de Caché básico.-** Este concepto se debe manejar con el concepto del problema de consistencia de caché granularidad de los datos de caché. --> Unidad de colocación de caché, tamaño del bloque.
- **Ubicación del Caché.-** Puede ser caché de disco, o caché en memoria principal (R.A.M.)
- **Política de actualización de caché.-** Hay tres política básica que son:

- 1.-Escritura a través
- 2.-Escritura retardada
- 3.-Escritura al cerrar

- **Consistencia.**-La estrategia usada existe:
 - *Estrategia iniciada por el cliente
 - *Estrategia iniciada por el servidor
- **Comparación del uso de caché y Servicio remoto.-**
 - Contraste de uso local con caché y tráfico del servidor.
 - Gasto extra, total de la vece para volúmenes grandes de dato con caché que con comparación del servicios remoto es menor.
- **Rutina de acceso a disco.-** En el servidor puede optimizarse para solicitudes de segmentos grandes contiguo, en lugares de bloques de disco aleatorios.
- **Desventaja.-** En el caché es la consistencia, cuando hay patrones de acceso a escrituras, es preferible el uso de caché, pero con escritura frecuente, el gasto extra es considerable en termino de desempeño, tráfico de red y carga del servidor.
- **Uso de caché.-** Para que sea preferible, se debe tener discos locales de gran capacidad de memoria principal grandes, de lo contrario, conviene el método de servicio remoto.

SERVICIO CON Y SIN ESTADO

Un servicio conectado, se describe como una conexión, entre el cliente y el servidor entre una sección, la ventana de este servicio es un mejor desempeño, la información del archivo se coloca en un caché de memoria principal, ahorrando de este modo el acceso a disco.

Un servicio conectado mantiene en su memoria principal información acerca de los clientes.

Un servidor de archivo "Sin" estado no tiene la información, por lo que cada solicitud identifica el archivo y la posición del archivo.

No necesita mantener una tabla de archivos abiertos de memoria principal.

La diferencia entre un servicio “Con” y “Sin” estado se hace evidente cuando ocurre una caída, un servidor de estado pierde todo en estado volátil en una caída, en servidor “Sin” estado no tiene estos problemas.

Pero el costo de usar el servicio “Sin” estado, es que los mensajes de solicitud son más largos y el procesamiento de sus solicitudes es más lento, puesto que no hay información en el núcleo que lo agilice.

REPLICACIÓN DE ARCHIVOS

Es una redundancia útil para mejorar la disponibilidad, el requisito básico, es que en la réplica que su mismo archivo reciba en maquina independiente en cuanto a fallo, la existencia de réplica debe ser invisible para los niveles superiores, no obstante la réplica se debe distinguir entre sí con nombres con bajo nivel distinto.

El principal problema de la réplica es su actualización.

Hay que conservar la semántica de consistencia pertinente cuando los accesos a replica sí trata de accesos virtuales de archivo.

Si la consistencia no es de primordial consistencia, se puede sacrificar a favor de la disponibilidad y el desempeño.

GESTION Y DISTRIBUCIÓN DE PROCESOS

- Migración de procesos.

Motivación

- a) Compartición de cargas.-** La carga puede equilibrarse y así mejorar el rendimiento global, no obstante se debe tener cuidado en el diseño de algoritmo para el equilibrio de carga. El rendimiento tiende a empeorar con el mayor requerimiento de comunicación que es necesario.
- b) Rendimiento de comunicaciones.-** Una manera de reducir el corte de comunicaciones es moviendo los procesos que interactúan en forma intensiva hacia un mismo nodo, además cuando se procesan archivos muy grandes es más ventajoso llevar los procesos hacia los datos.
- c) Disponibilidad.-** Los procesos que requieren de mucho tiempo deben moverse a otro sistema (nodo). Este en previsión a fallo o puede quedar en el mismo sistema asegurándose que pueden ser reanudados.
- d) Utilización de capacidades especiales.-** Un proceso podría requerir de algunas capacidades que son disponibles en otro nodo.

MECANISMO PARA MIGRACION DE PROCESOS

- a) **Inicio de migración.-** Su objetivo de la migración es equilibrar la carga, entonces será el sistema operativo el responsable de la decisión y si el objetivo es obtener algún recurso necesario la decisión del proceso.
- b) **Qué emigra?**
 - **Transferencia de todo el espacio de direcciones.-** Esta alternativa puede ser costosa cuando el espacio de direcciones es muy grande para luego el proceso solo necesite una parte menor
 - **Transferencia de solo la parte residente de memoria principal.-** En este caso si es necesario otro bloque y adicional se transfiere bajo demanda, esto implica que la máquina de origen siga involucrada en la vida del proceso. (bajo demanda).

NEGOCIACION DEL LA MIGRACIÓN

Transferencia apropiativo y no apropiativo.

- a) **Apropiativo.-** Consiste en transferir un proceso parcialmente ejecutado o al menos cuya creación se haya completado, esto implica transferir el estado del proceso.
- b) **No Apropiativo.-** Involucra a procesos que no comenzaron su ejecución, porque que no se necesitan transferir el estado del proceso.

ESTADOS GLOBALES DISTRIBUIDOS

Todos los problemas de concurrencia de un sistema fuertemente acoplados como la exclusión mutua, el interbloqueo y la inanición también existen en un sistema distribuido y de forma más completa, esto a causa de que no existe un estado global del sistema, o sea que no es posible que el sistema operativo. Cuando un proceso conozca el estado actual de todos los procesos distribuidos

Un proceso puede conocer solo el estado actual de todos los procesos del sistema local mediante el acceso a los lotes de control de los procesos que están en la memoria principal pero de los procesos remotos solo se puede obtener información de estados mediante mensajes. Pero estos mensajes representan el estado de proceso remoto en algún instante del pasado.

Los estados temporales impuestos por la naturaleza de un sistema distribuido complican todas las relaciones relativas a la concurrencia, para comprender las dificultades y darle una solución se debe analizar los términos siguientes.

- **Canal.-** Existe un canal entre dos procesos para el intercambio de mensajes.
- **Estado.-** El estado de un proceso es la secuencia de mensajes enviados y recibidos por los canales.
- **Instantánea.-** Registra el estado de un proceso.
- **Estado global.-** El estado combinado de todos los procesos.
- **Instantánea distribuida.-** Un conjunto de instantáneas una para cada proceso, el objetivo es que en la instantánea distribuida se registre un estado global consistente que se cumple cuando para cada estado de un proceso que registra la recepción de un mensaje, el envío de dicho mensaje ya se ha registrado en el proceso que lo envió.

El algoritmo instantáneo puede utilizarse para adoptar a cualquier algoritmo centralizado de un entorno distribuido para el conocimiento del estado global.

GESTION DISTRIBUIDA DE PROCESO – EXCLUSIÓN MUTUA

Los algoritmos de exclusión mutua pueden ser centralizado o distribuidos.

- **Algoritmos totalmente centralizados.-** Se designa un nodo como el nodo central de acceso a todos los objetos compartidos estos algoritmos centralizado tiene dos propiedades básicas:
 - Solo el nodo central toma decisiones para la asignación de recursos.
 - Toda la información necesaria se encuentra en el nodo central.

ALGORITMO DISTRIBUIDO

- Todos los nodos disponen de una cantidad igual de información por término medio.
- Cada nodo dispone solo de una representación parcial del sistema total y debe tomar decisiones basándose en esta información.
- Todos los nodos tienen igual responsabilidad en la decisión final.
- Todos los nodos dedican igual esfuerzo por término medio en llevar a cabo una decisión final.
- **Fallo.-** El fallo de un nodo no provocará en general el colapso total del sistema.

- No existe un reloj común a todo el sistema para regular el ritmo de los sucesos.

La carencia de un reloj común o de un medio de sincronizar los relojes locales es la principal restricción de los sistemas distribuidos.

Para superar estas dificultades hay un método llamado registros de tiempo o marcas de tiempo, este método ordena los sucesos de un sistema distribuidos sin utilizar relojes físicos.

El esquema de marcas de tiempo tiene la finalidad de ordenar los sucesos consistentes en la transmisión de los mensajes, la ordenación impuesta por este esquema no tiene por qué coincidir con la secuencia real en el tiempo, para los algoritmos basados en este esquema de marca de tiempo no es importante que el suceso ocurra primero, en realidad solo importa que todo los procesos que implementen el algoritmo estén de acuerdo con la ordenación que se impone sobre los sucesos.

COLAS DISTRIBUIDAS

La exclusión mutua excluida está basada en el concepto de cola distribuida, entre los algoritmos para mantener la exclusión mutua distribuida está el enfoque de paso de testigo, consiste en pasar un testigo entre los procesos participantes, el testigo es una entidad que en un momento dado es retenido por un proceso, el proceso que posee el testigo puede entrar en su sesión crítica sin pedir permiso. Cuando el proceso abandona su sesión crítica pasa el testigo a otro proceso.

INTERBLOQUEO DISTRIBUIDO

El bloqueo permanente de un conjunto de procesos que compiten por los recursos del sistema o se comunican unos con otros. Esta definición es válida tanto en un sistema sencillo como en uno distribuido del mismo modo que la exclusión mutua el interbloqueo introduce problemas más complejos en un sistema distribuido en comparación con un sistema de memoria compartida.

DETECCION DEL INTERBLOQUEO

La dificultad de la detección distribuida de los interbloqueos es que cada nodo solo conoce sus propios recursos mientras que el interbloqueo puede involucrar a recursos distribuidos. Los enfoques de solución son:

- **Control Centralizado.-** Un nodo es responsable de la detección de interbloqueo, todos los mensajes de petición y liberación de recursos se envían al proceso central así como el proceso que controla al recurso específico, este método necesita muchos mensajes y es vulnerable a fallos en el nodo central. Además de que pueden haber interbloques fantasmas.
- **Control Jerárquico.-** Los nodos se originan en una estructura de árbol con un nodo actuando como raíz del árbol, en cada nodo que no sea una hoja se reúne información sobre la asignación de recursos de todos los nodos dependientes, esto permite detectar interbloques entre niveles inferiores al nodo raíz.
- **Control Distribuido.-** Todos los procesos cooperan en la detección de interbloques, en general esto significa que deberá intercambiarse una información considerable con marcas de tiempo, esto hace que el coste sea significativo.

SISTEMAS DISTRIBUIDO DE EJECUCIÓN DISTRIBUIDA

a. Soporte de software distribuido en Middleware

El middleware puede proporcionar abstracciones de nivel más alto que son más sencilla de utilizar por los programadores para implementar aplicaciones distribuidas.

Las abstracciones generalmente se implementa como alguna forma de paquete de biblioteca por ejemplo los hilos se introdujeron inicialmente como extensiones de biblioteca para el lenguaje y el sistema operativo; otras veces resulta conveniente para los diseñadores de lenguajes que se implemente la abstracción como parte del sistemas de ejecución usado en la forma compilada del lenguaje

Un sistema de ejecución es una biblioteca que proporciona una colección de funciones específica por ejemplo todos los sistemas de programación del lenguaje C depende de la asignación dinámica proporcionada de memoria; en los sistemas actuales la mayoría de las cuestiones abiertas sobre que pertenecen a middleware y que pertenece al S.O. están relacionadas con la programación distribuida hasta antes que aparezca la WEB casi toda la atención en la computación distribuida estaba delicado a soportar el SW que resolviera problemas científicos los cuales exijan las máximas particiones posibles del HW y que normalmente estaban limitadas por computo.

Desde que internet se ha convertido en soporte de navegadores WEB y los servidores de contenido (el entorno World wide web) ha evolucionado una

nueva clase de software distribuido esta clase esta menos preocupada por los altas prestaciones (procesamiento) y está más dirigida a una distribución de contenido (información) eficiente.

b. Programación de aplicación distribuido clásicos

La red proporciona una oportunidad para que la conmutación sea dividida en unidades lógicas y después ejecutar estas de manera simultánea sobre distintos computadores de la red.

La aceleración es una métrica de prestaciones para la compilación distribuida suponiendo que un cálculo necesita T_1 segundos para ejecutarse en un monoprocesador si pudiera ejecutarse en $t_5 = t_1/5$ segundos.

En una red con 5 o más computadores podíamos decir que se ha acelerado en una razón de 5 por efecto de la sobrecarga debido a la administración la sincronización y las comunicaciones resulta difícil que se consigue una aceleración de N en una conexión de N el programa de aplicaciones debe conocer el comportamiento de algoritmo utilizado de la computación.

El comportamiento del sistema de SW que soporta de distribución y la naturaleza de las plataformas HW hay 2 enfoques generales que un diseñador los podía elegir:

1. Partición de datos
2. Partición de funcionalidad

1. Particiones de datos

Consiste en dividir los datos que se verán leído por el programa original en N flujo diferentes, puede dividir el conjuntos de datos original en N sub archivos y después confiar cada archivo a uno de los diferentes clones del programa secuencial original.

La partición de datos puede ser particularmente eficaz ya que uno existe interacción entre los procesos que ejecutan los registro individuales, pero si un proceso necesita acelerar a datos que están para de su partición entonces la partición de dato es mucho más compleja.

2. Partición de funcionalidad

Se divide un programa en partir luego los datos se pasan entre las distintas partes en algunos casos hay paralelismo haciendo que la etapas funcionales trabajen en paralelo con una caleña de montaje,

la dificultad con esta partición es que debe reescribirse el programa secuencial para ser utilizado además la partición resultante crea partes que deben traer instalado los mecanismos de compartición y sincronización apropiados.

SOPORTE MIDDLEWARE PARA LA PROGRAMACIÓN DISTRIBUIDA

Los programadores que trabajan con programación de aplicación distribuida clásica desean resolver diversos problemas científicos, si un sistema operativo proporciona el protocolo UDP/TCP el programador de la aplicación podrá usar los mecanismos de IPC (Interprocess comunicación) como base para construir la solución. Sin embargo esto no maneja las tareas de gestión de procesos asociados como por ejemplo el de crear un proceso hijo en otra máquina, algunos paquetes middleware que se utilizan para soportar aplicaciones de cálculo científico son:

- **PVM (Parallel virtual machine).**- Cada computador utilizado es una configuración PVM contiene aplicaciones con bibliotecas PVM para el sistema operativo host. Las funciones de biblioteca utilizan las capacidades de gestión de procesos locales del sistema operativo para crear y gestionar procesos PVM, una tarea de PVM es una unidad de conjunto planificable que utiliza una máquina virtual para ejecutarse.
- **Beowulf.**- el hardware Beowulf es estándar implementado mediante tecnología Ethernet, un cluster Beowulf utiliza varios paquetes de software de libre disposición para crear el entorno incluyendo Linux, PVM, MP1 y software GNU.
- **DCE OSF.**- OSF ha creado un paquete middleware unificado para soportar explícitamente la computación distribuida denominado el entorno de computación distribuida. El middleware del DCE está implementado como un conjunto de rutina de biblioteca que corre en el sistema operativo de la máquina cliente proporcionando los servicios que están disponibles en la máquina local así no actuando como cliente intermediario, si los servicios se obtienen en una máquina remota. Los servicios DCE integrados son:
 - Servicios de archivos distribuidos.
 - Servicios para la seguridad de autenticación y la utilización de accesos a recursos.
 - Servicios a directorios para la denominación y localización uniforme de los recursos en el entorno distribuido.

- Servicios de tiempo para sincronizar los relojes de la red.

PROGRAMACIÓN DISTRIBUIDA EN LA RED

La programación distribuida en paralela a estado tradicionalmente por la necesidad de soluciones de altas prestaciones para los problemas científicos, por ese motivo emergió un nuevo y específico modelo de computación basado en la integración de los navegadores y los servidores web. En el caso más simple existe una pequeña compactación distribuida en la recuperación de contenidos en la web, utilizando el enfoque simple. Cada iteración está compuesta por la transferencia de información desde el cliente seguida por la respuesta del servidor.

Se proporciona un marco computacional en el navegador web, para acoger este tipo de cómputo mediante la descarga de un pequeño programa de aplicación applet desde el servidor hacia el cliente al comienzo de la transacción. La idea general de la applet se denomina código móvil que se encarga en el cliente solo cuando es necesario.