TEMA 9 :"SINCRONIZACION EN SISTEMAS DISTRIBUIDOS"

INTRODUCCION A LA SINCRONIZACION EN SISTEMAS DISTRIBUIDOS.
Además de la comunicación también es importante la forma en que los procesos cooperen y se sincronizan entre si (Ej. forma de implementar las regiones criticas ; forma de asignar recursos en sistemas distribuidos).
Los problemas de las regiones criticas, exclusión mutua y la sincronización, generalmente se resuelven en sistemas de una sola CPU con métodos como los semáforos y los monitores se basan en la memoria compartida, que no se aplican en sistemas distribuidos.
Otro problema es el tiempo y la forma de medirlo, en los modelos de sincronización.

SINCRONIZACION DE RELOJES: para los algoritmos distribuidos, la información se distribuyen entre varias maquinas, los procesos toman las decisiones en forma local .debe evitarse un único punto de fallo en el sistema. No existe un reloj común del tiempo global, es inaceptable reunir toda la información en un solo lugar para su procesamiento
Lograr sincronización sin centralización. En un sistema distribuido no es importante poner de acuerdo a todos las maquinas en la hora.

- La falta de sincronización en los relojes puede ser problemáticas en procesos dependientes del tiempo
- se requieren un acuerdo global del tiempo
-

RELOJES LOGICOS: las computadoras poseen un circuito para el registro del tiempo llamado "DISPOSITIVO RELOJ", en un cronometro consistente en un cristal de cuarzo de precisión sometida a una tensión eléctrica que oscila con una frecuencia definida. Se puede programar un cronometro para que genere una interrupción x veces por segundo, a cada interrupción se llama "MARCA DE RELOJ"
Para varias computadoras cada una con sus relojes es imposible garantizar que los cristales de maquinas distintas oscilan a igual frecuencia.
La diferencia entre los valores del tiempo se llama "DISTORSION DE RELOJ"; y que puede generar fallos en los programas dependientes del tiempo.
Si dos procesos no interactúan no es necesario que los sus relojes este sincronizados lo importante no es que los procesos estén de acuerdo en la hora; si no que coincidan en el orden en que ocurren los eventos.
Para los relojes lógicos lo que importa es la consistencia interna de los relojes, no su cercanía el tiempo real (oficial).

RELOJES FISICOS: Los relojes físicos deben ser iguales (están sincronizados) y no deben desviarse del tiempo real (Ej. en los sistemas de tiempo real) y es importante la hora real del reloj.
En ciertos sistemas se precisa mas de un reloj físico externo y se deben sincronizar entre si y con los relojes del mundo real.
La medición del tiempo real con alta precisión no es sencilla.
Antiguamente el tiempo se media astronómicamente; el día solar es el intervalo entre dos tránsitos consecutivos del sol El segundo 1/86400 de un día solar. el segundo es el tiempo que tarda el átomo de cesio para hacer 9000 millones de transacciones aproximadamente .La oficina internacional de la hora en Paris recibe indicaciones de 50 relojes atómicos en el mundo y calculo el tiempo atómico internacional (TAI) y de esto surge el tiempo coordinado universal (UTC).

El instituto nacional del tiempo estándar de E.E.U.U. y otros países , operan estaciones de radio de onda corta o satélites de comunicaciones, transmiten pulsos UTC con regularidad (Cada segundo) y conociendo la posición del emisor y del receptor se debe compensar el retraso de la propagación de la señal , si se recibe la señal , si se recibe la señal por moden se compensa y se obtiene el tiempo con precisión.

ALGORITMO PARA LA SINCRONIZACION DE RELOJES: si la maquina tiene receptor de UTC todas deben sincronizarse con ella .Si no tiene, cada maquina eleva el registro de su tiempo y se debe mantenerse el tiempo de todas lo mas cercano posible.
ALGORITMO DE CRISTIAN: es para sistemas donde una maquina tiene un receptor de UTC .
El objetivo es sincronizar todas la maquinas con ella cada maquina envía un mensaje Del servidor para solicitar el tiempo actual. El despachador del tiempo responde .Para mejorar la precisión se toman varias mediciones y se promedian.
ALGORITMO DE BERKELEY:en el algoritmo de cristian el servidor de tiempo es pasivo acá es activo. Realiza un muestreo periódico de todas las maquinas para preguntarles el tiempo, y con la respuestas calcula el tiempo promedio- Es adecuado cuando no hay un receptor UTC .Indica a las maquinas que avancen o disminuyan
ALGORITMO POR PROMEDIO: Los anteriores algoritmos centralizados, este es descentralizado .Divide el tiempo en intervalos de resincronizacion de longitud fija .Al inicio de cada intervalo cada maquina transmite el tiempo actual según su reloj y los transmisiones no serán simultaneas porque difieren las velocidades de los relojes, se reciben todas y se promedian los valores.
VARIAS FUENTES EXTERNAS DE TIEMPO: los sistemas que requieren una sincronización los distintos intervalos de tiempo de las distintas fuentes de tiempo, y determinan la intersección, se compensan los retrasos y las velocidades de los relojes.

EXCLUSION MUTUA: cuando un proceso debe leer o actualizar ciertas estructuras de datos compartidas entre una región critica para lograr la exclusión mutua y garantizar que ningún otro proceso utilice las estructuras de datos del mismo tiempo .
En sistemas monoprocesadores las regiones criticas se protegen con semáforos, monitores pero en sistemas distribuidos es mas difícil.
ALGORITMO CENTRALIZADO: se logra la exclusión mutua similar a la forma que se lleva a cabo en un sistema monoprocesador .se elige u proceso coordinador y cuando un proceso sea ingresar a la región critica envía un mensaje de solicitud del coordinador para solicitar servicio, si no hay otro otorga el permiso, el proceso entra.
Si un proceso pide permiso para entrar a una región critica ya asignada a otro proceso, el coordinador no otorga el permiso y encola los pedidos.
En su esquema sencillo y justo con pocos mensajes de control pero el coordinador es un cuello de botella y si falla se bloquean los procesos en espera de una respuesta de acceso
ALGORITMO DISTRIBUIDO: el objetivo es no tener un único punto de fallo (el coordinador central) .se ordenan todos los eventos en el sistema para saber cual ocurrió primero. Cuando un proceso quiere entrar en una región critica, construye un mensaje los envía a todos los procesos y estos enviaran el reconocimiento diciendo OK. Si el receptor ya esta en la región critica no responde y encola la solicitud .Si dos procesos quieren entrar al mismo tiempo se comparan las marcas de tiempo y la menor gana. Un proceso necesita el permiso de todos para entrar a la región critica. Si cualquier proceso falla no responderá a las solicitudes y se interceptara como negación de acceso .S e incrementa la probabilidad de fallo y el trafico en la red. Se sobrecarga el sistema

porque todos los procesos participan en la toma de decisiones para el ingreso a las regiones críticas.
ALGORITMO DE ANILLO DE FICHAS (token ring): los procesos forman un anillo lógico y cada proceso tiene una posición en un anillo, cada proceso sabe quien sigue .Al iniciar el anillo se le da a un proceso la ficha que circula en todo el anillo .Cuando un proceso obtiene la ficha de su vecino verifica si intenta ingresar a una región critica, en caso positivo entra, trabaja, sale y para la ficha, sino vuelve a pasarla .En un instante solo un proceso esta en una región critica.

ALGORITMO DE ELECCION: son los algoritmos para la elección de un proceso coordinador, iniciador, secuenciador, etc.
El objetivo es garantizar que luego de la elección todos los procesos están de acuerdo con que sea así la identidad del nuevo coordinador.
ALGORITMO DE GRANDULON GARCIA-MOLINA: un proceso inicia una elección cuando el coordinador no responde a las solicitudes .El proceso envía un mensaje "elección" a los demás procesos con un numero mayor .Si nadie responde asume que gana y se convierte en el nuevo coordinador .Si responde un proceso con un numero mayor toma el control y el otro proceso termina. Se envía un mensaje para anunciarlo.
ALGORITMO DE ANILLO: los procesos tienen un orden físico o lógico y cada proceso conoce a su sucesor .Cuando un proceso nota que el coordinador no funciona se construye un mensaje " elección" con su numero de proceso y se lo envía a su sucesor y así sucesivamente y cada proceso añade su numero de proceso .Cuando llega nuevamente el mensaje al proceso que lo inicio, se hace circular un mensaje "coordinador" para informar a los demás procesos quien es coordinador (el que tenga el numero mayor) y los miembros del anillo.

TRANSACCIONES ATOMICAS: las técnicas de sincronización son de bajo nivel. El programador se enfrenta directa con:

- exclusión mutua
- manejo de regiones criticas
- prevención de bloqueos
- recuperación de fallos

en la transacción atómica se precisan técnicas de abstracción de mayor nivel para ocultar aspectos técnicas y permitan a los programadores concertarse en los algoritmos y como los procesos trabajen juntos en paralelo.
La propiedad de la transacción atómica el "TODO O NADA", o se hace todo lo que hay que hacer como una unidad o no se hace nada (Ej. retirar dinero de una cuenta y depositar en otra).Se agrupa las dos operaciones en una transacción, o se hacen las dos operaciones o no se hacen ninguna, y se puede regresar al estado inicial sino concluye la transacción.

EL MODELO DE TRANSACCION: si en el sistema hay varios procesos independientes que pueden fallar aleatoriamente.
ALMACENAMIENTO ESTABLE: se implanta con una pareja d discos comunes, cada una es una copia exacta (espejo) de otra unidad .Primero se actualiza uno y luego otro, si falla el sistema y solo actualizo la unidad 1, luego de la recuperación se puede actualizar la unidad 2 en función de la 1.Es un esquema adecuado para aplicaciones que requieren de un alto grado de tolerancia a fallos (ej transacciones atómicas)

PRIMITIVAS DE TRANSACCION: Son proporcionados por el S.O. o por el sistema de tiempo de ejecución del lenguaje .Existen comandos que pueden formar una transacción, terminarla o eliminarla.
PROPIEDADES DE LAS TRANSACCIONES:

- *SEÑALIZACION*: las transacciones que ocurren al mismo tiempo no interfieran entre si.
- *ATOMICIDAD:* la transacción no ocurre si ocurre totalmente e instantáneamente.
- PERMANENCIA: si se compromete la transacción los cambios son permanentes.
- *TRASACCIONES ANIDADAS*: Cuando las transacciones pueden contener subtransacciones(procesos hijos) que se ejecuten en paralelo entre si en procesadores distintos y originan nuevas transacciones
- .

IMPLANTACION DEL MODELO DE TRANSACCION.
existen varios metodos:
ESPACIO DE TRABAJO PARTICULAR: cuando un proceso inicia una transacción se le otorga un espacio de trabajo particular que contienen todos los archivos a los cuales tienen acceso. Las lecturas- escrituras van a este espacio hasta que la transacción se completa o aborte.
Hay un alto consumo de recursos por las copias de los objetos al espacio de trabajo.
BITACORA DE ESCRITURA ANTICIPADA: o lista de intenciones .Los archivos realmente se modifican pero antes de cambiarlos, se graba un registro con los datos necesarios en la bitácora, si la transacción aborta se utiliza la bitácora para respaldo del estado original y se deshacen los cambios hechos .La bitácora permite ir hacia delante o hacia atrás.
PROTOCOLO DE COMPROMISO DE DOS FASES. Uno de los procesos que interviene funcionan como coordinador, este escribe una entrada en la bitácora para indicar que inicia el protocolo, se envía un mensaje a cada procesos para que estén preparados para el compromiso, los que los reciben el mensaje verifican si están listos y lo informan, cuando el coordinador recibe todas las respuestas sabe si se establece el compromiso o aborta.

CONTROL DE CONCURRENCIA EN EL MODELO DE TRANSACCION: se necesitan algoritmos de control de concurrencia cuando se ejecutan varias transacciones simultáneamente: en distintos procesos, en distintos procesadores.
Los principales algoritmos son.
CERRADURA (LOCKING): cuando un proceso debe leer o escribir en un archivo como parte de una transacción, primero cierra el archivo .La cerradura se hace mediante un único manejador centralizado o manejador local en cada maquina las cerradura de lectura se comparten, las cerraduras d escrituras son exclusivas para una transacción .
CONTROL OPTIMISTA DE LA CONCURRENCIA: es sencilla, se sigue adelante y hace todo lo que tiene que hacer, sin prestar atención que tiene que hacer, sin presta atención a lo que hacen los demás se actúa luego si hay algún problema. Se registran los archivos leídos o grabados y el momento del compromiso se verifican las transacciones para ver si algún archivo se modifico desde el inicio de la transacción si ocurre esto se aborta la transacción, sino se completa.
MARCAS DE TIEMPO: a cada transacción se le asocia a una marca d tiempo al iniciarla, las marcas so únicas, cada archivo tendrá una marca para lectura y otra para escritura, que indican la ultima transacción y cuando un proceso intente acceder a un

archivo lo hará si las marcas de tiempo son menores (mas viejas) que la marca de la transacción actual.

BLOQUEOS EN SISTEMAS DISTRIBUIDOS: son peores que en los sistemas monoprocesadores (mas difícil de evitar, prevenir, detectar y recuperan).toda la información relevante esta dispersa en muchas maquinas .Son criticas en sistemas de base de datos distribuidos .Estrategias usuales para el manejo de bloqueos.
ALGORITMO DE AVESTRUZ. Ignora el problema igual que el uso de un solos procesador.
EVITACION. Evitar los bloqueos mediante la asignación cuidadosa de los recursos, pero en los sistemas distribuidos es difícil implementarlos, se requiere información de antemano (proporción de recursos que necesitara cada proceso)
Las más aplicables son:
DETECCION: se permiten que ocurran los bloqueos, se los detecta e intenta recuperarse de ellos.
PREVENCION: se hace que los bloqueos sean imposibles desde el punto de vista estructural.

DETECCION DISTRIBUIDA DE BLOQUEOS: cuando se detecta un bloqueo en un S.O. convencional se eliminan uno o más procesos. Pero en un sistema basado en transacciones se abortan una o mas transacciones y el sistema restaura el estado que tenia antes de iniciar la transacción, pero también la transacción puede volver a comenzar .Las consecuencias de eliminar un proceso son menos severas si se utilizan transacciones.
DETECCION CENTRALIZADA DE BLOQUEOS. Cada maquina tiene la grafica de recursos de sus propios procesos y recursos .Un coordinador central tiene la grafica de recursos todos el sistema, cuando el coordinador detecta un ciclo elimina un proceso para romper el ciclo.
DETECCION DISTRIBUIDA DE BLOQUEOS: los procesos pueden solicita varios recursos al mimo tiempo, en vez de uno cada vez. Un proceso espera uno o mas recursos simultáneamente y pueden ser recursos locales o remotos .Si un proceso se bloquea por otro proceso se envía un mensaje de exploración al procesos que detiene los recursos necesarios, si el mensaje, recorre todo y regresa del emisor, significa que existe un ciclo y el sistema esta bloqueado .Se lo rompe eliminado al proceso que inicio la exploración, o el que tiene el numero mas alto.
PREVENCION DISTRIBUIDA DE BLOQUEOS: consiste en el diseño cuidadoso del sistema para que los bloqueos sean imposibles estructuralmente.
Hay diversas técnicas:
* permitir a los procesos que solo conserven un recurso a la vez
* Exigirle a los procesos que soliciten todos sus recursos al principio.
* que los procesos liberen sus recursos cuando soliciten uno nuevo.
En sistemas distribuidos con el tiempo global y transacciones atómicas : se asocia a cada transacción una marca d tiempo global al indicar la transacción (no hay parejas de transacciones con igual marca).cuando un proceso se va a bloquear en espera de un recurso que esta usando otro proceso , se verifica cual tiene la marca mayor (o mas joven) y se permite que espere solo si el proceso es espera (mas vieja) tiene menor que el otro .al seguir una cadena de procesos en espera, las marca aparecen de forma creciente y los ciclos son imposibles.