INTRODUCION A LOS SISTEMAS DISTRIBUIDOS

Lenin Hernández
donlino_9012@hotmail.com

Escuela Politécnica Nacional

RESUMEN: *Este documento va a tratar una pequeña Introducción a Sistemas Distribuidos y va a tratar temas como por ejemplo: Técnicas para la sincronización de relojes de varias computadoras referente a lo que son los sistemas distribuidos, así como medidas para prevenir un ataque de denegación de servicios. Se va a tratar además algunos ejemplos de software y hardware que son compartidos en un sistema distribuido.*

1 TÉCNICAS PARA LA SINCRONIZACIÓN DE RELOJES DE VARIAS COMPUTADORAS EN UN SISTEMA DISTRIBUIDO.

Es indispensable entender que a más de la buena comunicación que deben tener las computadoras en un sistema distribuido, es fundamentar entender como funcional los procesos, como cooperan y se sincronizan entre ellos. Además es necesario conocer el tiempo y la manera en que este se mide, ya que el tiempo es un factor primordial en algunos modelos de sincronización, ya que la sincronización es crucial en procesos que son dependientes del tiempo.

Otro tema a considerar es la información que se distribuye entre las distintas maquinas, debido a que los procesos ocurren y toman decisiones respecto la información que estos reciben. Antes de considerar los tipos de técnicas para la sincronización de relojes, vale la pena preguntar es posible sincronizar todos los relojes para generar una sola unidad en común entre todos los computadores. Por ejemplo al trabajar con un solo computador no hay problema si el tiempo de este está desfasado un poco, pero al momento de trabajar con varios computadores, cada uno con su propio reloj la situación llega a cambiar totalmente.

Existen dos tipos de controladores de tiempo por así decirlo entre ellos se encuentran los relojes lógicos y los físicos. Los primeros suelen ser un cristal de cuarzo mecanizado, y estos crsitales oscilan a una determinada frecuencia, cada oscilación disminuye un cantador, cuando este llega a 0 se genera una retención y se vuelve a cargar en el registro de retención. Cada interrupción es un ciclo en el reloj. De esta manera el reloj se mantiene hasta la fecha [1].

Una vez dicho esto es posible sincronizar varios computadores. Lamport menciono que si dos procesos no trabajan juntos no es necesario sincronizar sus relojes, ya que no generarían problemas, es mas a pesar de que trabajen varios procesos juntos necesariamente no importa que todos estos procesos estén exactamente a la misma hora, lo que tienen que saber es el momento en el que va a trabajar cada proceso, para realizar esto Lamport genero un algoritmo. Este al igual que otros tipos de técnicas vamos a analizar en este capítulo.

1.1 ALGORITMO DE LAMPORT

Este algoritmo sincroniza los relojes lógicos de tal forma que cada proceso sepa el momento en el cual va a trabajar. Para ello Lamport define el término suceso anticipado o suceso antes de (happensbefore), a manera de ejemplo seria (a) sucede antes que (b), esto en otras palabras se refiere al hecho de que cada proceso continúa después de otro. Existen dos tipos de situaciones según Lamport tomando en cuenta que a, b y h son procesos:

- a = h solo si b ocurre antes entonces a -> b es verdadero. Un proceso ocurre después de otro.
- Si a es un mensaje y h es un mensaje recibido entonces a -> h. Esto quiere decir que un mensaje no podría ser recibido si antes no se lo envió.

Lo que intenta hacer Lamport es medir el tiempo de forma que cada evento se debe asignar un tiempo, de tal forma que todos los procesos se pongan de acuerdo. En la Figura 1. Veremos de forma más clara el cómo trabaja este algoritmo.

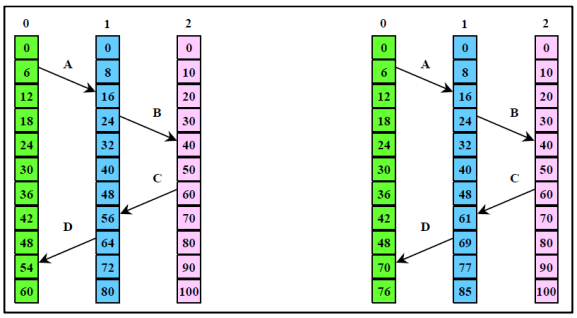

Figura 1. Procesos en funcionamiento con su respectivo reloj corrección usando el algoritmo de Lamport.

En la Figura 1, se puede apreciar determinados procesos que ocurren a un determinado tiempo, si observamos en el lado derecho los procesos pueden tener un percance debido a que los tiempos no concuerdan, por ejemplo el proceso dos envía un mensaje C al proceso 1, si inspeccionamos un poco se puede visualizar que hay un retroceso del tiempo lo cual es imposible, lo mismo ocurre al enviar el mensaje D del proceso 1 a 2.

Lamport resolvió este inconveniente debido a que si un proceso ocurre en un determinado tiempo otro

proceso no puede ocurrir al mismo tiempo, sino que ocurre en un tiempo posterior. Cuando un mensaje es enviado y el reloj del receptor adelanta su reloj de tal forma que se tenga mayor tiempo para el envió.

Gracias al algoritmo de Lamport se logra dar un determinado tiempo a cada uno de los procesos en un sistema distribuido y cumple las siguientes condiciones Ref. [2]:

- Si a ocurre antes de b en el mismo proceso, "C(a) < C (b)
- SI a y b son envío y recepción, C(a) < C(b).
- Para todos los eventos a y b, C(a) es distinto de C(b).

1.2 ALGORTIMO DE CRISTIAN

El objetivo principal de este algoritmo es sincronizar todas las maquinas a una máquina que posee un receptor UTC (tiempo coordenado universal). UTC es la base de toda la relojería moderna, posee la antigua norma de la hora de Greenwich refiriéndonos al tiempo astronómico. Cada computadora que interviene envía periódicamente mensajes para solicitar el tiempo actual, esta máquina que posee un receptor UTC devuelve el mensaje que contiene la hora actual [1].

Por lo tanto lo que intenta este algoritmo es mantener un conjunto de relojes o computadores sincronizados, pero no importa la hora que tengan lo importante es todos tengan la misma hora, pero basándose en una hora confiable mediante un receptor UTC, en otras palabras esta máquina esta sincronizada externamente y el resto de máquinas esta sincronizada internamente con esta máquina que posee un receptor UTC.

Este algoritmo presenta algunos inconvenientes [3]:

- Al solicitar la hora a través de un servidor, el problema ocurriría si este servidor falla, el resto de maquinas se quedarían sin referencia, por lo tanto soluciona este problema solicitando el servicio de hora a varios servidores y se queden con el que les responde primero.
- Si la hora de referencia recibida desde el servidor es menor a la que se desea ajustar, habría un pequeño retraso lo que resultaría en inconvenientes sobre todo al realizar modificaciones en documentos.

1.3 ALGORTIMO DE BERKELEY

En este algoritmo existe un servidor activo que pregunta a cada cliente por su hora, al mismo tiempo este calcula el promedio de llegada y informa al cliente como debe efectuar el cambio de hora en caso de necesitar modificarla. Pueden existir varios servidores, entre los clientes realizan un algoritmo de lección que les permite saber cuál es el más factible en el caso de que caiga un servidor.

La diferencia entre el algoritmo de Cristian es que el algoritmo de Cristian es pasivo en otras palabras este espera que los clientes realicen las peticiones, el algoritmo de Berkeley, el servidor se toma el tiempo para preguntar el resto de máquinas que hora tienen, sobre la base de respuestas en todas las maquinas se realiza un promedio de tiempo y les avisa al a las maquinas si deben avanzar sus relojes o retrasarlos hasta que todas tengan la misma hora.

En la Figura 2. se puede apreciar el funcionamiento de este algoritmo [2].

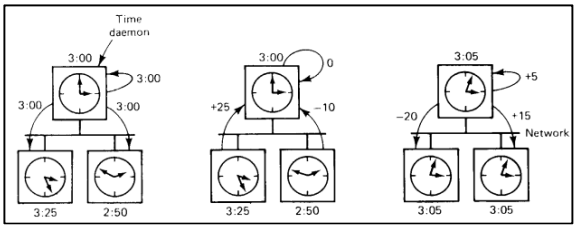

Figura 2. Proceso de peticiones de hora y sincronización, utilizando el algoritmo de Barkeley.

En la Figura 2, en la imagen de la izquierda se puede apreciar como el servidor realiza una petición de la hora al resto de máquinas, en la imagen del medio las maquinas responden a la petición del servidor y dan sus respectivas horas en sus relojes, finalmente en la figura de la derecha el servidor pide al resto de las maquinas retrasar o adelantar sus relojes hasta que todas tengan la misma hora.

1.4 ALGORITMO PROMEDIADO.

Este tipo de algoritmo es descentralizado, todas las maquinas dan su hora actual de acuerdo a su reloj, a causa de que los relojes son diferentes y en diferentes maquinas no funcionan a la misma velocidad va a tardar un poco de tiempo hasta que todas las maquinas den su hora actual. Hay un intervalo de tiempo hasta que todas las maquinas terminen de enviar su hora actual, finalmente se hace un promedio para calcular un nuevo tiempo, este tiempo de recepciones se lo realiza mediante la topología de red, u realizando un cronometrado hasta que todas las maquinas envíen su hora.

2 ATAQUE DE DENEGACION DE SERVICIOS Y MEDIDAS QUE EXISTEN PARA PREVENIRLOS

Se lo define dentro de los ataque activos realizados para realizar una interrupción. Lo que realiza un ataque de denegación de servicios es evitar la ejecución de una actividad por ejemplo navegación entre las aplicaciones en la web, transferencia de archivos o peticiones. Este tipo de denegaciones se crean enviando un conjunto de mensajes hacia un destinatario de forma que interfieran

el funcionamiento del mismo, teniendo como resultado afectar total o de forma parcial al servicio que esté realizando el destinatario.

Existen dos tipos de ataque: ataque de vulnerabilidad y ataque de inundación. El primero es cuando se realiza infiltraciones a una maquina objetivo, y la segunda es cuando se envían una determinada cantidad de paquetes hasta que el receptor resulte afectado. Los paquetes maliciosos puede provocar en un determinado sistema pueden provocar bucles infinitos, ralentizar ciertos procesos y la velocidad de ejecución de los mismos, consumir grandes cantidades de memoria y finalmente el paro del servicio que se esté realizando.

Otra forma de efectuar este tipo de ataques y afectar un servicio es atacar a la interface de la red, para ello solo se genera flujos de velocidad mayor al máximo admitido por la tarjeta de red, en el caso de no poseer algún tipo de filtrado de paquetes y de dichos tráficos, estos paquetes consumirán todos los recursos de la red y finalmente afectaran los servicios de la víctima.

La ocultación es un método que utilizan los atacantes para que las víctimas no se den cuenta, y para que no puedan rastrear el origen de donde proviene. Esto normalmente consiste en realizar una comunicación con varias máquinas, una atrás de otra antes de ingresar a los gestores, realizando saltos y ocultándose, esto dificulta en gran manera la investigación para llegar al atacante, en ocasiones los saltos entre máquina y maquina hasta la víctima se los efectúan de país en país, incluso puede llegar a otro continente, y termina en una tarea uy complicada.

Hay un tema referente a los ataques de denegación de servicio el denominado ataque distribuido [4]. Y se efectúa juntando varias máquinas que provocan tráfico de forma sincronizada a la víctima. Estas máquinas no son de propiedad del atacante, pueden ser maquinas ubicadas en universidades, escuelas, laboratorios que fueron invadidos y controlados por el atacante. El atacante normalmente ve posibles fallos y logra infiltrarse en las máquinas, obteniendo un acceso ilimitado a sus sistemas. Todo esto lo realizan con ciertas herramientas que buscan vulnerabilidades en ciertos sistemas de forma automática, después el atacante envía paquetes masivos con un simple comando realizando una inundación y afectando a sus víctimas de forma considerable.

2.1 DEFENSA FRENTE A LOS ATAQUES DE DENEGACION DE SERVICIOS

2.1.1 MECANISMOS DE SEGURIDAD EN EL SISTEMA

Este tipo de defensas evita la proliferación de máquinas vulnerables, esto impedirá que los atacantes realicen ataques masivos. Mecanismos de seguridad como por ejemplo firewalls, uso de antivirus, cierre de puertos no utilizados, actualización automática de software, chequeo de aplicaciones en entornos personalizados que no afecten al sistema.

Existen ciertos sistemas denominados honeypot que se construyó con la finalidad de ser atacado, actúan como cebos de tal forma que el atacante piensa que está afectando al sistema principal, pero en realidad no. El funcionamiento principal de estos sistemas es obtener información acerca del atacante, aprender los métodos de ataque y en ocasiones detectar la ubicación del atacante.

2.1.2 MECANISMOS DE SUPERVISION DE RECURSOS

El controlar el acceso de los usuarios, permiten realizar un servicio adecuado a los usuarios que realmente pertenecen al sistema y deniegan y bloquean a los usuarios que no tienen permisos. El proveer contraseñas adecuadas al sistema impide el acceso a los atacantes y lo hacen más seguro.

2.1.3 AUMENTO DE RECURSOS

El proveer a las maquinas con recursos necesarios permite contrarrestar los ataques de paquetes masivos impidiendo el agotamiento de los mismos. Por ejemplo poseer un ancho de banda elevado, poseer un buen número de servidores de tal forma que puedan compartir la carga, o incluso tener servidores de respaldo los cuales se activan cuando los servidores principales están en aprietos [4].

2.1.4 DETECCION DE ATAQUES

Hay algunos modos de detectar los ataques ya sea por el tipo de actividad que se esté realizando y identificando cómo se comporta el trafico habitualmente, e incluso por un reconocimiento por experiencia, a este tipo de reconocimiento se lo denomina basado en firmas. Al entender el comportamiento que han tenido ataque efectuados anteriormente ya se entiende su funcionamiento y por lo tanto es sencillo de identificar y establecer medidas de seguridad. Este tipo de mecanismos permite detectar ataques debido a que ta se tiene almacenado patrones de ataques en una base de datos, lo que realiza simplemente es un bloqueo. La desventaja es que es vulnerable el sistema ante ataques nuevos.

Otra manera de detectar ataques es mediante anomalías, basándose en comportamientos anormales del sistema, esto se lo realiza mediante comparaciones efectuadas en distintos periodos de tiempo con el funcionamiento correcto del sistema. Para ello es necesario establecer un marco de funcionamiento normal y un marco de funcionamiento anormal.

3 DE 5 EJEMPLOS DE RECURSOS HARDWARE Y 5 EJEMPLOS DE RECURSOS SOFTWARE QUE

PUEDEN SER COMPARTIDOS EN UN SISTEMA DISTRIBUIDO.

Entre los recursos hardware tenemos:

- Discos duros. Normalmente en unos sistemas distribuidos se pueden repartir el consumo de almacenamiento para un mejor funcionamiento y como seguridad en el caso de pérdida de información.
- Impresoras. El compartir recursos como impresoras en un sistema distribuido pequeño evita mayores costos dentro de la empresa.
- Procesadores. Mejora el funcionamiento de los procesos al repartirlos equitativamente.
- Servidores. Este es el caso de los servidores de correo, juegan un roll importante en los sistemas distribuidos
- Redes de comunicación. Esto permite ahorrar espacio al interconectar varias computadoras.

Entre los recursos software tenemos:

- Aplicaciones. El uso de aplicaciones es común encontrarlas en un sistema distribuido, esto sucede en el caso de los juegos.
- Base de datos. Normalmente el tener repartida la información tiene como objetivo en primer lugar seguridad y en segundo lugar el acceso.
- Sistemas operativos. Este tipo de sistemas trabajan en multiprocesamiento.
- Ficheros, archivos. Permiten distribuir la información de tal forma que todo se encuentra distribuido. Este trabaja a manera de nodos de tal forma que se comportan como iguales entre si. Tienen como objetivo realizar un intercambio directo de la información.
- Navegación web. Permite a las computadoras estar interconectadas entre sí.

4 EJERCICIO

Un programa servidor escrito en C++ provee la implementación de un objeto BLOB que se supone puede ser accedido por diferentes clientes que pueden estar escritos en diferentes lenguajes como Java. Los computadores cliente y servidor pueden tener diferente hardware pero todos están conectados al Internet. Describa los problemas ocasionados por cada uno de los 5 aspectos de heterogeneidad mencionados que deben ser resueltos para que un cliente pueda invocar un método en el objeto BLOB del servidor.

- Redes de comunicación.
- Hardware entre computadoras pueden ocasionar inconvenientes.
- Sistemas Operativos.
- Los lenguajes de programación.
- Implementación por distintos desarrolladores.

5 FORMATO

6 REFERENCIAS

[1] Andrews S. Tanenbaum, Distributed Operating Systems, 1ra Edition, Prentice Hall, pp. 120-134, 1994.

[2] David L. Martinez. (2001, Noviembre). Sistemas Operativos. [En linea]. Disponible en: http://sistop.gwolf.org/biblio/Sistemas_Operativos_-_Luis_La_Red_Martinez.pdf.

[3] Fernando L. Romero. (2009, Febrero).Sincronizacion de relojes en ambientes distribuidos. [En Linea]. Disponible en: http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Redes_de_Datos/Tesis/Romero_Fernando.pdf

[4] Gabriel M. Fernández. (2007, Mayo). Ataque de denegación de servicios a baja tasa contra servidores. [En linea]. Disponible en: http://0-hera.ugr.es.adrastea.ugr.es/tesisugr/16714763.pdf