

Sincronización

Importancia de sincronizar los procesos

Es importante que varios procesos no accedan simultáneamente a un recurso compartido, sino que cooperen para garantizar a cada uno el acceso exclusivo temporal al recurso. La sincronización en sistemas distribuidos es con frecuencia mucho más difícil comparada con la sincronización en sistemas de un procesador o de multiprocesadores.

Sincronización de reloj

En un sistema centralizado, el tiempo no es ambiguo. Cuando un proceso quiere saber la hora, realiza una llamada de sistema y el núcleo se la dice. Si un proceso A pregunta la hora, y después un proceso B pregunta la hora, el valor que obtiene B será mayor que (o probablemente igual que) el valor que obtuvo A; con certeza no será menor. En un sistema distribuido, lograr un acuerdo con respecto al tiempo no es algo trivial.

Ejemplo

Pensemos sólo en las implicaciones de carecer de un tiempo global en el programa make de UNIX. En general, en UNIX, los programas grandes se dividen en varios archivos fuente, de modo tal que un cambio en un archivo fuente sólo necesita compilar un archivo, y no todos. Si un programa consta de 100 archivos, el hecho de no tener que recompilar todo, debido a que un archivo se modificó, incrementa enormemente la velocidad a la que los programadores pueden trabajar.

Sincronización de reloj

Por lo general, los cronómetros de computadora son cristales de cuarzo mecanizados con precisión. Cada cristal tiene dos registros, un contador y un registro de soporte. Cada vuelta del cristal reduce el contador en uno. Cuando el contador llega a cero, se genera una interrupción y el contador se reinicia desde el registro de mantenimiento. De esta forma, el temporizador se puede programar para interrumpir 60 veces por segundo o en cualquier otra frecuencia deseada. Cada interrupción se llama un tic de reloj. Con una computadora y un reloj, no importa si el reloj está apagado una pequeña cantidad. Dado que todos los procesos en la máquina usan el mismo reloj, serán internamente consistentes. Por ejemplo, si input.c tiene una hora de 2151 e input.o tiene una hora de 2150, make volverá a compilar los archivos de origen aunque los relojes estén retrasados por 2 cuando las horas reales sean 2153 y 2152, respectivamente. Lo que realmente importa es el momento relativo.

Tiempo atómico internacional

El tiempo se pudo medir con mayor precisión, con precisión e independientemente del movimiento de la Tierra, calculando las transiciones de los átomos de cesio-133. Los físicos, asumiendo lo que hacen los astrónomos para medir el tiempo, definen un segundo como el tiempo que tarda un átomo de cesio-133 en hacer exactamente 9192631770 transiciones. Se eligió 9192631770 para hacer que el segundo atómico fuera igual al segundo solar en el año en que se introdujo. Actualmente hay relojes de cesio-133 en varios laboratorios alrededor del mundo.

Usar TAI para el tiempo significa que a medida que pasa el tiempo, la cena llegará cada vez más temprano, hasta que en algún momento llegue en las primeras horas de la mañana. La gente tomará nota y tendremos la misma situación que en 1582 cuando el Papa Gregorio XIII ordenó quitar 10 días del calendario.

Sincronización de reloj

GPS

El GPS es un sistema distribuido basado en satélites lanzado en 1978. Aunque se usa principalmente en aplicaciones militares, en los últimos años se ha abierto camino en muchas aplicaciones civiles, principalmente en la navegación. Sin embargo, hay muchas más áreas de aplicación. Por ejemplo, los teléfonos GPS ahora permiten a las personas que llaman rastrear la ubicación de los demás, lo que puede ser muy útil si se pierde o tiene problemas.

Hasta ahora hemos asumido que las medidas son completamente precisas, sin peligro ellos no son. Primero, el GPS no tiene en cuenta los segundos vacíos. En otras palabras, hay una desviación sistemática de UTC de 14 segundos cada 1 de enero de 2006. Este error se puede compensar fácilmente en el software. Sin embargo, hay muchas otras fuentes de error, que van desde relojes atómicos satelitales que no siempre están perfectamente sincronizados, posiciones satelitales inexactas, precisión limitada de los relojes receptores y velocidades de propagación de señales erráticas (por ejemplo, cuando la señal ingresa a la ionosfera, la señal cambia más abajo), etcétera. También todos sabemos que la Tierra no es una esfera perfecta y, por lo tanto, necesitamos correcciones adicionales.

Algoritmos de sincronización de relojes

Todos los algoritmos tienen como base el mismo modelo del sistema. Se supone que cada máquina tiene un cronómetro que ocasiona una interrupción H veces por segundo. Cuando este cronómetro se apaga, el manipulador de interrupciones agrega 1 al reloj de software, el cual da seguimiento al número de marcas (interrupciones) a partir de algún momento pasado acordado. Llamemos C al valor de este reloj.

de la máquina p es $C_p(t)$. En un mundo perfecto, tendríamos $C_p(t) = t$ para toda p y toda t . En otras palabras, $C_p(t) = dC/dt$ idealmente debería ser 1. $C_p(t)$ se conoce como la frecuencia del reloj de p en el tiempo t . La **distorsión del reloj** se define como $C_p(t) - 1$, y denota cuánto difiere la frecuencia de la de un reloj perfecto. La **compensación** relativa a un tiempo específico t es $C_p(t) - t$. Los cronómetros reales no interrumpen exactamente H veces por segundo. En teoría, un cronómetro con $H = 60$ debe generar 216000 marcas por hora. En la práctica, el error relativo obtenible con chips cronómetros modernos es de alrededor de 10^{-5} , lo cual significa que una máquina en particular puede obtener un valor situado en el rango de 215998 a 216002 marcas por hora. De manera más precisa, si existe una constante p tal que

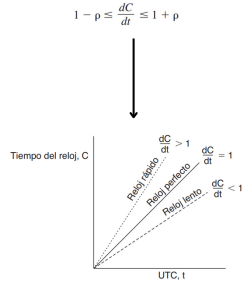


Figura 6-5. Relación entre el tiempo del reloj y el UTC cuando el reloj marca a diferentes velocidades.

Protocolo de tiempo de red

Un método común en muchos protocolos, originalmente propuesto por Cristian (1989), es dejar a los clientes contactar a un servidor de tiempo. El último puede proporcionar exactamente el tiempo actual, por ejemplo, debido a que está equipado con un receptor WWV o un reloj exacto.

El truco reside en encontrar una buena estimación para estos retrasos. Considere la situación esquematizada en la figura 6-6.

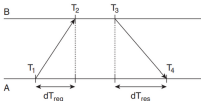


Figura 6-6. Obtención del tiempo actual desde un servidor de tiempo.

En este caso, A enviará una petición a B, con un registro de tiempo T_1 . B, a su vez, registrará el tiempo del receptor, T_2 (tomado de su propio reloj local), y devolverá una respuesta con un registro de tiempo T_3 y encimará el valor T_2 previamente registrado. Por último, A registra el tiempo de la llegada de la respuesta, T_4 . Supongamos que el retraso de la propagación de A a B es casi el mismo que de B a A, ello significa que $T_2 - T_1 = T_4 - T_3$. En ese caso, A puede estimar su compensación con respecto a B como

$$\theta = T_3 - \frac{(T_2 - T_1) + (T_4 - T_3)}{2} = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

En el caso del protocolo de tiempo de red (NTP, por sus siglas en inglés), éste se configura en pares entre servidores. En otras palabras, B también sondeará a A en cuanto a su tiempo actual. La compensación se calcula tal como ya indicamos, junto con la estimación para el retraso:

$$\delta = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$$

Aplicar el NTP simétricamente debe, en principio, permitir también a B ajustar su reloj con el de A. Sin embargo, si se sabe que el reloj de B es más exacto, entonces tal ajuste sería imprudente. Para resolver este problema, el NTP divide a los servidores en estratos. Un servidor con un reloj de referencia, tal como un receptor WWV o un reloj atómico, se conoce como servidor de estrato 1 (se dice que el propio reloj opera en el estrato 0).

Algoritmo de Berkeley

En Berkeley UNIX se aplica exactamente el método opuesto (Gusella y Zatti, 1989). Aquí, el servidor de tiempo (de hecho, un demonio de tiempo) es activo, ya que cada cierto tiempo pregunta a cada máquina sobre la hora ahí registrada. Basado en las respuestas, calcula un tiempo promedio y les indica a todas las máquinas que adelanten o atrasen sus relojes, según la nueva hora. Este método es conveniente para un sistema en el que ninguna máquina tiene un receptor WWV. El operador debe configurar manualmente y de manera periódica la hora del demonio de tiempo. La figura 6-7 ilustra el método.

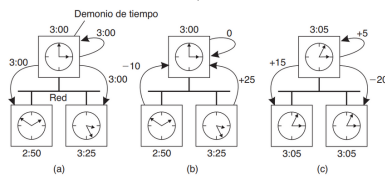


Figura 6-7. (a) El demonio de tiempo pregunta a las otras máquinas los valores de sus relojes. (b) Las máquinas responden. (c) El demonio de tiempo les indica cómo ajustar sus relojes.