Tema 5: Sincronización

Jose Luis Díaz
Curso 2011-2012

Contenidos

Esquema

Introducción

Algunos algoritmos dependen de alguna forma del paso del tiempo para su funcionamiento:

- Pueden requerir la hora exacta
- O pueden necesitar sólo conocer en qué orden ocurrieron ciertos eventos

Problema

En un sistema distribuido no hay un reloj único, sino uno en cada máquina y los relojes de los computadores no son exactos.

¿Cómo funciona el reloj?

En un computador la hora se mantiene gracias a un cristal de cuarzo que oscila con una frecuencia constante.

- Cada oscilación genera un 1 seguido de un 0.
- Un circuito cuenta cuántos "unos" se han generado.
- Cuando la cuenta alcance un valor prefijado, se pondrá de nuevo a cero y se generará una interrupción.
- Cada vez que el sistema operativo recibe una interrupción, avanza un contador.
- Este contador del sistema operativo permite saber cuánto tiempo transcurrió desde que el computador arrancó, y por tanto la hora.

Problema

La frecuencia de oscilación del cuarzo puede ser ligeramente diferente entre máquinas, causando que el reloj *adelante* o *atrase*

Esquema

Reloj lógico

Observar que

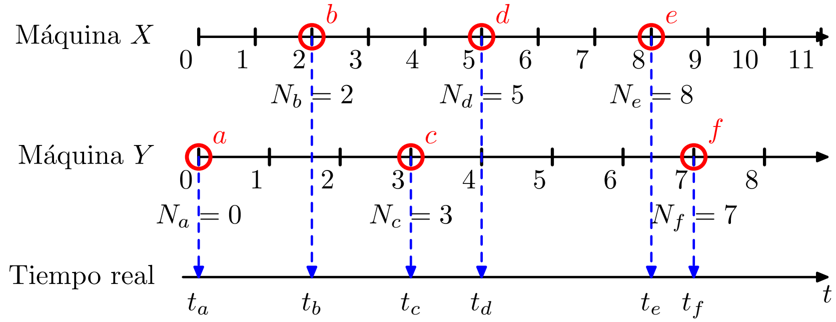
- Si el cuarzo no oscila a la frecuencia correcta, la hora que el computador "cree" que es es falsa
- Pero incluso en ese caso, existe una *consistencia lógica* entre las horas que el computador genera, ya que

Si un evento a tiene un valor de cuenta, N_a, mayor que el de otro evento b, N_b, es que ha ocurrido después. Es decir:

$$N_a > N_b \qquad \Longrightarrow \qquad t_a > t_b$$

- En cambio, si a y b son eventos en diferentes máquinas, la implicación anterior no es necesariamente cierta

Ejemplo



$N_a = 0$	$N_d = 5$
$N_b = 2$	$N_e = 8$
$N_c = 3$	$N_f = 7$

Observar

En los eventos e y f se da que $t_f > t_e$ ¡y sin embargo $N_f < N_e$!

Observación [Lamport]

La hora real, t_a, a la que ocurre un evento a es desconocida. Sólo podemos conocer lo que marca el reloj de esa máquina en ese instante, N_a.

Definimos

La relación $a \to b$ significa **El evento a ocurrió antes que el evento** b, por tanto $t_a < t_b$

Sin saber t_a ni t_b ¿Cuándo podemos afirmar que $a \to b$?

1. Cuando a y b ocurren en la misma máquina y $N_a < N_b$
2. Cuando a representa el envío de un mensaje y b su recepción (aunque sea en diferentes máquinas)

Idea [Lamport]

¿Será posible asignar a cada evento e un número entero $C(e)$ que cumpla

$$C(a) < C(b) \quad \Longrightarrow \quad a \to b \quad ?$$

Es decir:

1. Si a ocurre antes que b en la misma máquina, debe cumplir $C(a) < C(b)$.
   - Esto lo cumple el contador de interrupciones de la máquina
2. Si a es el envío de un mensaje y b es su recepción, debe cumplirse $C(a) < C(b)$
   - Esto lo cumpliría el contador de interrupciones si al enviar un mensaje se incluyera en el mismo el valor de $C(a)$ y al llegar nos aseguramos de que $C(b)$ sea mayor

Algoritmo de Lamport

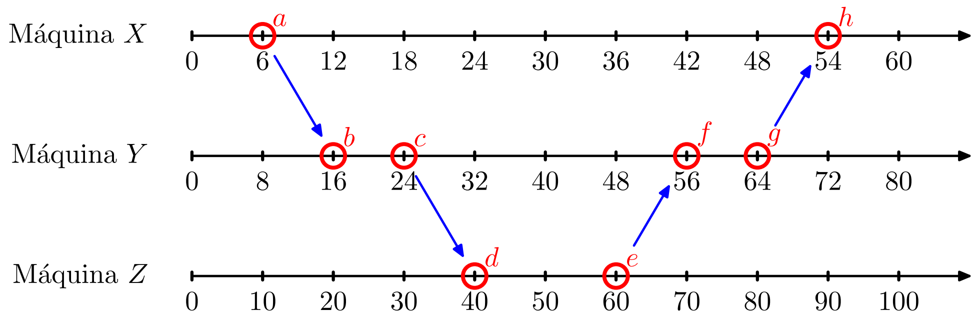
- Se usa como $C(a)$ de un evento a el valor del contador de interrupciones de la máquina en que ocurre el evento.
- Al enviar un mensaje, se incluye en el mismo el contador de interrupciones en el momento del envío, $C(a)$.
- Al recibir un mensaje, se compara el contador de interrupciones local, $C(b)$, con el valor que viene en el mensaje, $C(a)$
  - Si $C(a) < C(b)$ no hay contradicción lógica. No se hace nada.
  - Si $C(a) \geq C(b)$ se adelanta el contador local, haciendo $C(b) = C(a) + 1$.
- Este algoritmo garantiza además que, si $a \rightarrow b$ y $b \rightarrow c$, entonces $a \rightarrow c$
- Observar que pueden existir pares de eventos a, b tales que no sea posible afirmar $a \rightarrow b$, ni tampoco $b \rightarrow a$. Se dice en este caso que a y b son concurrentes, y se denota por $a||b$.

Ej

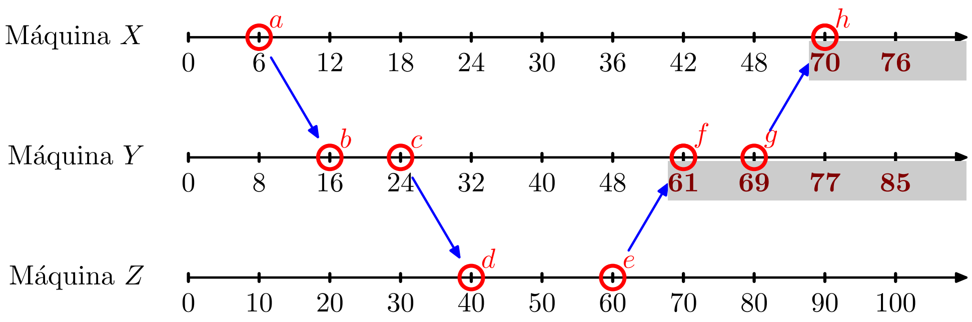
Ejemplo de aplicación del algoritmo

Supongamos tres máquinas, X, Y y Z cuyos osciladores tienen diferentes frecuencias de modo que mientras en Z se producen 10 interrupciones, en Y se producen 8 y en X sólo 6.

La máquina X envía un mensaje a Y, poco después, ésta envía un mensaje a Z. Más tarde Z responde a Y y seguidamente Y responde a X. La situación se muestra en la figura:



Ejemplo de aplicación del algoritmo



Cuando se detecta una contradicción al recibir un mensaje se adelanta el reloj en el receptor

Esquema

¿Qué hora es?

Denominaremos *reloj físico* a uno que marca la hora exacta, y no una mera ordenación de eventos como en el caso de los relojes lógicos.

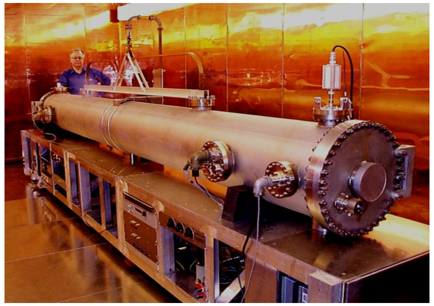
La definición de "hora exacta" es más compleja de lo que parece a simple vista.

- Hasta 1948:
  - El segundo estaba definido como 1/86400 del día
  - Pero resulta que la longitud del día sufre pequeñas variaciones
  - Se define el **segundo solar medio**, como 1/86400 del día promedio
  - La rotación de la tierra sobre si misma se va ralentizando con el tiempo. La definición de segundo solar por tanto no es constante. . .

Definición moderna de segundo

- En lugar de las vueltas de la tierra, se cuentan las transiciones de un átomo de cesio-133.
- Un reloj atómico de cesio puede contar con precisión cuántas transiciones hace un átomo de cesio-133
- Se define el **TAI** (International Atomic Time), como el número de transiciones que ha hecho un átomo de cesio-133 desde el 1 de Enero de 1958, dividido entre 9 192 631 770

El *Bureau International de l'Heure* (BIH) en Paris recibe información de relojes de cesio, y calcula el **International Atomic Time** (TAI).



UTC

Medir el tiempo usando TAI presenta un problema:

- El "segundo de cesio" (TAI) es constante de un año a otro
- El "segundo solar" en cambio va creciendo con el tiempo, debido a la ralentización de la rotación terrestre.

Estas dos medidas, por tanto, se desincronizan.

- Cada vez que la diferencia entre los "segundos TAI" y los "segundos solares" se hace mayor de 800ms, el BIH inserta un "segundo intercalar" (*leap second*), de modo que queden sincronizados de nuevo.

El contador de segundos (incluyendo intercalares) suministrado por el BIH, se denomina **UTC** (Universal Coordinated Time), y es la referencia mundial de "hora exacta".

Cómo obtener el valor de UTC

WWV

Es una emisora de radio de onda corta que emite un pulso cada vez que transcurre un segundo UTC.

- Un computador puede equiparse con un receptor WWV

GPS

El sistema de localización global (GPS) basado en satélites, también suministra el valor de UTC con precisón de 10ms

- Un computador puede equiparse con un receptor GPS

Sincronización de relojes

Caso más simple: una de las máquinas tiene un receptor WWV o GPS, y las demás se deben sincronizar con ella.

Idea básica

Cada cierto tiempo, las restantes máquinas piden la hora a la que tiene el receptor, y usan la respuesta para poner en hora sus relojes.

Problemas

- ¿Cada cuánto le preguntan la hora?
- ¿Cómo se tiene en cuenta el retardo de la red?
- ¿Cómo poner en hora el reloj local una vez recibida la respuesta?

Frecuencia de sincronización

Llamemos $C(t)$ a la hora que tiene una máquina cuendo la hora real es t.

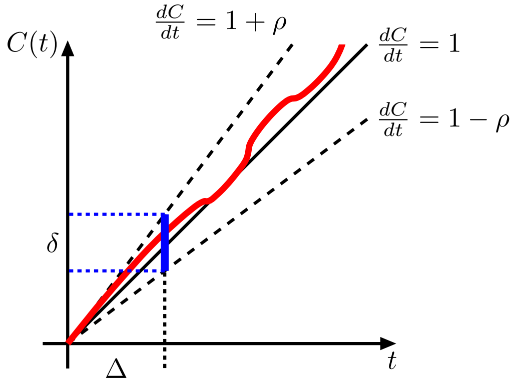
- Si su reloj fuera perfecto, cumpliría $C(t) = t$ todo el tiempo
- Dicho de otra forma, en un reloj exacto $\frac{dC}{dt} = 1$
- Mientras el reloj adelante, $\frac{dC}{dt} > 1$
- Mientras el reloj atrase, $\frac{dC}{dt} < 1$

El fabricante de un reloj debe garantizar que $\frac{dC}{dt}$ no se aleja de 1 sin control, sino dentro de unos límites. Es decir, debe cumplir

$$\frac{dC}{dt} = 1 \pm \rho$$

siendo ρ el **ratio máximo de deriva** del reloj. Este dato lo dará el fabricante.

Frecuencia de sincronización

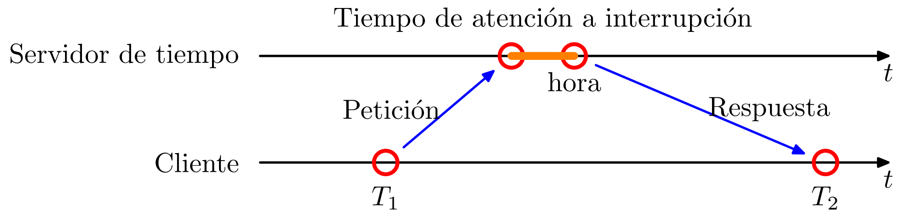


Cuando haya transcurrido un tiempo Δ, dos relojes con el mismo ρ estarán desincronizados como máximo $\delta = 2\rho\Delta$

Si no queremos que la discrepancia sea mayor de δ, hay que resincronizarlos cada $\delta/2\rho$ segundos.

El retardo de la red

La hora que recibimos no es la que tiene en ese momento el servidor, sino la que tenía cuando respondió.



En ausencia de más datos podemos aproximar el retardo de la respuesta por $(T_2 - T_1)/2$, siendo T_1 la hora local cuando preguntamos la hora, y T_2 la hora local cuando la recibimos. [Algoritmo de Christian]

Poner en hora el reloj local

- Si la hora real es posterior a la que marca nuestro reloj, basta adelantarlo, pero si es anterior no podemos atrasarlo. ¡El reloj nunca debe retroceder!
- En ese caso se puede hacer que el reloj atrase, hasta que alcance la hora correcta.

Ejemplo

Si en nuestra máquina se producen 100 interrupciones por segundo, el operativo normalmente añadirá 10ms a su hora cada vez que ocurra una interrupción. Si tenemos que retrasar la hora 0.02 seg, el operativo puede hacer que cada interrupción incremente la hora en solo 9ms. Tras 20 interrupciones habremos acumulado un retraso de 20ms, con lo que el reloj ya estará de nuevo en hora.

- Una técnica similar a la anterior, pero haciendo que el reloj adelante, se puede usar también para avanzar la hora del reloj de forma suave en vez de brusca.