Tema3.pdf

Bigbounze

Desarrollo de Sistemas Distribuidos (Especialidad Ingeniería del Software)

3º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

Tema 3: Coordinación

. Tiempo lógico

Las cuestiones de tiempo importantes en Sistemas Distribuidos por:

- Medida que deseamos obtener con precisión para:
  - Sincronización externa: cuando ocurrió un evento concreto. Para ello es necesario sincronizar la hora de la máquina donde ocurrió el evento con algún reloj o fuente externa autorizada.
  - Sincronización interna: se trata de obtener las mismas referencias de tiempo o intervalo entre dos eventos ocurriendo en dos computadoras diferentes conociendo precisión, para un instante dado.
- Problemas lógicos debidos a la distribución.

Un evento es una acción que parece ocurrir indivisiblemente (p.ej. envío de mensaje). El orden de la ocurrencia de eventos puede ser crítico en aplicaciones distribuidas.

Requisitos y tipos de aplicaciones:

- Centralizadas: sólo necesitan conocer el orden de los eventos, con lo que basta asociar un reloj o contador a cada evento.
- Distribuidas:
  - Conocer el desplazamiento relativo del tiempo de una máquina con respecto a otra, e idéntica en velocidad del pulso (casi imposible).
  - Otra opción es que exista un reloj físico compartido (sistemas síncronos).
  - Servidor de tiempo sobre peticiones (sistemas asíncronos).

Relación de orden:

- Esquema de ordenación de eventos basados en dos puntos:
  - Si dos eventos ocurren en el mismo proceso, entonces ocurren en el orden que se observan.
  - Si se envía un mensaje, entonces el evento asociado al envío ocurre antes que el evento de recepción de dicho mensaje.
- Lamport generalizó estas dos relaciones en una relación de orden causal denominada ocurrió-antes (-->):

1. Si $\exists p: x \xrightarrow{p} y$ (en p) entonces $x \rightarrow y$
2. $\forall m \in Mensajes,\ send(m) \rightarrow receive(m)$
3. Siendo $x, y, z \in Eventos$: $x \rightarrow y$ e $y \rightarrow z$ entonces $x \rightarrow z$

Relojes lógicos:

- Mecanismo simple que propuso Lamport para capturar numéricamente la relación Ocurrió Antes.
- Un reloj lógico es un contador software que se incrementa monótonamente:

- C_p : nota el reloj lógico C del proceso p
- $C_p(a)$: nota la marca de tiempo del evento a en el proceso p
- $C(b)$: nota la marca de tiempo del evento b en cualquier proceso donde haya ocurrido

- Para captar la relación, los procesos actualizan sus relojes lógicos y transmiten sus valores en los mensajes:

1. C_p se incrementa antes de cada evento que ocurre en p
2. Cuando un proceso p envía un mensaje le añade el valor $t = C_p$
3. Cuando un proceso q recibe un mensaje entonces:
   - computar $C_q = max\ (C_q, t)$ y
   - aplicar *acción 1* antes de marcar el evento *receive(m,t)*

- Es fácil demostrar que si a --> b entonces C(a) < C(b)
- Extensión a relación de orden total
  - $C(a) < C(b) \Leftrightarrow Cp(a) < Cq(b) \vee (Cp(a) = Cq(b) \wedge p < q)$

. Algoritmos distribuidos

Exclusión Mutua:

- Cuando no existe núcleo central local para basar la exclusión mutua en variables u otras facilidades compartidas. Un ejemplo de esto sería la existencia de servidores que no tienen mecanismos de sincronización incorporados.

Elección:

- Método para escoger un único proceso que realice un rol concreto.

. Exclusión mutua

Requisitos básicos:

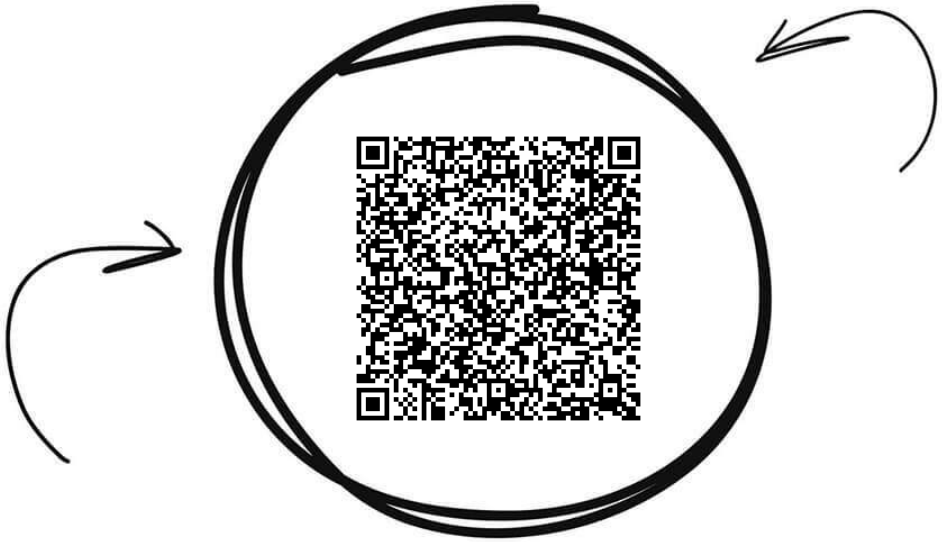
- Propiedades de seguridad y vivacidad
- Orden causal en la entrada a la sección crítica

Soluciones:

- Servidor centralizado:
  - Para entrar en la sección crítica se envía una petición al servidor y se espera la respuesta.
  - El servidor encola peticiones cuando no dispone del testigo.


InfoJobs
x VISA
¡Gana 1.000€ en un día como
Probador/a de pago con el móvil!

Desarrollo de Sistemas Distr...



Banco de apuntes de la

WUOLAH

Comparte estos flyers en tu clase y consigue más dinero y recompensas

1. Imprime esta hoja
2. Recorta por la mitad
3. Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes
4. Llévate dinero por cada descarga de los documentos descargados a través de tu QR

    - Cuando un proceso sale de la sección crítica envía un mensaje de liberación (devuelve el testigo), si el servidor tiene mensajes de petición encolados le envía el testigo al primero de ellos y lo saca de la cola.
    - El servidor se puede convertir en un cuello de botella y el punto crítico de fallo.
    - Hay que regenerar el testigo si el cliente que lo tiene falla.
- Algoritmo distribuido en relojes lógicos:
    - Los procesos que desean entrar en la sección crítica envían un mensaje multicast a los otros n-1 procesos. Un proceso puede entrar si todos los demás le responden, es decir, la obtención del testigo requiere n mensajes.
    - Suposiciones:
        - Los procesos conocen las direcciones de los demás.
        - Paso de mensajes fiable.
        - Cada proceso mantiene su reloj lógico.
    - Como obtener el testigo requiere n mensajes se vuelve un proceso muy costoso. Cualquier proceso es un punto crítico de fallo. Cada proceso recibe peticiones y envía respuestas, por tanto, el cuello de botella puede ocurrir.
- Algoritmo basado en anillo:
    - Los procesos se configuran en un anillo lógico (cada proceso conoce la dirección de sus vecinos).
    - El testigo circula en una sola dirección y el proceso que lo tienen puede acceder a la sección crítica, en caso contrario ha de esperar.
    - Suposiciones:
        - Cada proceso conoce la dirección de su vecino por la derecha
        - Paso de mensaje fiable
    - Obtener el testigo requiere n mensajes.
    - El testigo está continuamente circulando.
    - Si un proceso falla se ha de reconfigurar el anillo.
    - Se debe regenerar el testigo si se pierde.
    - No es posible asegurar el cumplimiento de la relación Ocurrió-Antes.

. Elección

Se trata de escoger un único proceso de un conjunto de ellos. Su principal requisito es que el proceso sea único incluso si varios solicitan la elección simultáneamente.

Soluciones:

- Algoritmo del valentón:
    - Requisitos:
        - Los miembros del grupo se conocen.
        - Paso de mensajes fiable.
    - Tres tipos de mensaje:
        - Elección: para anunciar una elección.
        - Respuesta: se envía como respuesta a un mensaje de elección.
        - Coordinador: se envía para anunciar el id del nuevo proceso coordinador.
    - Es costoso pues se requieren hasta n mensajes.
    - Los pasos que sigue el algoritmo del valentón son:

        - Un proceso comienza una elección cuando detecta que un proceso ha fallado. Envía un mensaje de elección a los procesos con id más alto que él mismo.
        - Espera un mensaje de respuesta.
        - Si no llega el mensaje, se proclama coordinador y envía un mensaje coordinador a todos los procesos con id más bajo que él, en otro caso, espera un tiempo a que llegue un mensaje de coordinador del proceso elegido, si éste no llega comienza una nueva elección.
        - Si un proceso recibe un mensaje de coordinador graba el identificador contenido en dicho mensaje.
        - Si un proceso recibe un mensaje de elección devuelve un mensaje de respuesta y comienza una elección, al menos que haya iniciado una.
        - Cuando se restablece un proceso que había fallado, comienza una nueva elección. Si tiene el id más alto será el nuevo coordinador junto con el actual hasta que éste reciba el mensaje coordinador.
- Algoritmo basado en anillo:
    - Requisitos:
        - Los procesos están organizados en un anillo lógico, aunque sin coincidir el orden con su id.
        - Paso de mensajes fiable y los procesos no fallan durante la elección.
    - Los procesos indican su participación con dos tipos de mensajes:
        - Elección: para anunciar que hay una elección en curso.
        - Coordinador: se envía para anunciar el id del nuevo proceso coordinador.
    - Costoso al requerir hasta n mensajes.
    - Los pasos que sigue son:
        - Inicialmente, cada proceso está marcado como no-participante. Cualquiera puede comenzar la elección, se marca como participante y envía un mensaje de elección con su id de proceso a su vecino por la derecha.
        - Si recibe un mensaje de elección, compara su id con el del mensaje
            - Si es mayor que el del mensaje:
                - Si está marcado como no participante, sustituye el id del mensaje de elección por el suyo y lo envía.
                - Si está marcado como participante, no envía el mensaje.
            - Si es menor pasa el mensaje de elección recibido a su vecino.
            - En cualquier caso, si no lo estaba, se marca como participante.
        - Si el id recibido es el del propio receptor es el proceso con mayor id y se convierte en coordinador. Se marca como no-participante y envía un mensaje de coordinador a su vecino con su id.
        - Si se recibe un mensaje de coordinador, se marca como no-participante y reenvía dicho mensaje a su vecino.