Sistemas Distribuidos

Resumen

Dorian Wozniak

Índice

1 COORDINACIÓN BÁSICA **2**
- 1.1 Modelos temporales . . . 2
  - 1.1.1 Modelo asíncrono . . . 2
  - 1.1.2 Modelo síncrono . . . 2
  - 1.1.3 Modelo parcialmente síncrono . . . 2
- 1.2 Relojes . . . 2
  - 1.2.1 Relojes físicos . . . 2
  - 1.2.2 Relojes lógicos . . . 2
- 1.3 Estados globales . . . 3
  - 1.3.1 Historia local . . . 3
  - 1.3.2 Estado del proceso . . . 3
  - 1.3.3 Historia global . . . 3
  - 1.3.4 Corte . . . 3
  - 1.3.5 Algoritmo de Chandy-Lamport (instantáneas) . . . 3

2 COORDINACIÓN DE RECURSOS COMPARTIDOS **4**
- 2.1 Algoritmos distribuidos centralizados . . . 4
- 2.2 Algoritmos distribuidos descentralizados . . . 4
  - 2.2.1 Algoritmo de Lamport . . . 4
  - 2.2.2 Algoritmo de Ricart-Agrawala . . . 4
  - 2.2.3 Algoritmo de Suzuki-Kasami (paso de testigo) . . . 4
  - 2.2.4 Algoritmo de Raymond . . . 4

3 TOLERANCIA A FALLOS **5**
- 3.1 Modelos de fallo . . . 5
- 3.2 Detección de fallos . . . 5
- 3.3 Gestión de fallos . . . 5

Capítulo 1

COORDINACIÓN BÁSICA

1.1 Modelos temporales

1.1.1 Modelo asíncrono

1.1.2 Modelo síncrono

1.1.3 Modelo parcialmente síncrono

1.2 Relojes

1.2.1 Relojes físicos

1.2.2 Relojes lógicos

Los relojes lógicos se basan en **relaciones de causalidad** entre eventos. Si un proceso P_i secuencial observa un evento e antes que e', se conoce como una **relación de orden local** ($e \rightarrow_i e'$). Importan especialmente las relaciones entre envíos y recepciones entre procesos.

1.2.2.1 Relojes escalares (de Lamport)

En los relojes de Lamport se establecen **relaciones de orden global** (*happened before*) entre eventos. Si a ocurre antes que b, ($a \rightarrow b$) y a afecta casualmente a b.

- Los eventos se ordenan causalmente
- La relación es transitiva
- La relación es de orden parcial (no reflexiva), pero puede ser de orden total si se incluye los identificadores de proceso
- Si $a \not\rightarrow b$ y $b \not\rightarrow a$, son eventos **concurrentes** ($a||b$)

Los relojes C_i deben satisfacer las siguientes condiciones:

- Si $a \rightarrow b$ en el proceso P_i, $C_i(a) < C_i(b)$
- Si a es un envío desde P_i, y b es una recepción en P_j, $C_i(a) < C_j(b)$

Se siguen las siguientes **reglas de implementación**:

- Cada evento local de P_i se estampilla con $C_i = C_i + 1$
- Al realizar un envío de un mensaje m de P_i a P_j
  - P_i envía C_i junto a m, siendo $C_i = C_i + 1$
  - P_j guarda el mensaje, siendo $C_j = max(C_i, C_j) + 1$

En eventos no locales, $C(a) < C(b)$ no implica automaticamente que a ocurra antes que b puesto que no hay traza de donde procede el avance del reloj.

1.2.2.2 Relojes vectoriales

En un sistema de N procesos, cada uno contiene un vector $V_i[N]$, donde $V_i[j]$ contiene el mejor valor conocido por el proceso P_i de un proceso P_j, y $V_i[i]$ contiene su reloj.

Se siguen las siguientes **reglas de implementación**:

- Cada evento local de P_i se estampilla con $V_i[i] = V_i[i] + 1$
- Al realizar un envío de un mensaje m de P_i a P_j
  - P_i envía V_i junto a m, siendo $V_i[i] = V_i[i] + 1$
  - P_j guarda el mensaje, siendo V_j:
    - \$V_j[j] $= V_j[j] + 1$
    - $V_j[k] = max(V_i[k], V_j[k]) \quad \forall k \neq j$

Los relojes cumplen las siguientes propiedades:

- Un proceso tiene siempre la versión mas actualizada de su propio reloj
- $a \rightarrow b$ solo si $V[a] < V[b]$
- Dados dos relojes vectoriales:
  - $V = V'$ si $V[i] = V'[i] \quad \forall i \in 1..N$
  - $V \geq V'$ si $V[i] \geq V'[i] \quad \forall i \in 1..N$
  - $V > V'$ si $V \geq V'$ y $V \neq V'$
  - $V || V'$ si $V \not\geq V'$ y $V' \not\geq V$

1.3 Estados globales

1.3.1 Historia local

Cada proceso P_i contiene una historia (secuencia de eventos): $h_i = [e_i^0, e_i^1, e_i^2, ...]$

- La historia puede ser finita o infinita
- Un k-prefijo de h_i es la historia de h_i hasta k: h_i^k
- Cada evento puede ser local o de comunicación

1.3.2 Estado del proceso

s_i^k es el estado de un proceso P_i antes del evento e_i^k

- s_i^k memoriza todos los eventos en $h_i^k - 1$
- s_i^0 es el estado inical de P_i

1.3.3 Historia global

1.3.4 Corte

1.3.5 Algoritmo de Chandy-Lamport (instantáneas)

Capítulo 2

COORDINACIÓN DE RECURSOS COMPARTIDOS

2.1 Algoritmos distribuidos centralizados

2.2 Algoritmos distribuidos descentralizados

2.2.1 Algoritmo de Lamport

2.2.2 Algoritmo de Ricart-Agrawala

2.2.3 Algoritmo de Suzuki-Kasami (paso de testigo)

2.2.4 Algoritmo de Raymond

Capítulo 3

TOLERANCIA A FALLOS

3.1 Modelos de fallo

3.2 Detección de fallos

3.3 Gestión de fallos