

Sistemas Distribuidos. Temario.



- 1. Introducción**
- 2. Comunicación**
- 3. Procesos**
- 4. Nombrado y localización**
- 5. Sincronización**
- 6. Consistencia y replicación**
- 7. Tolerancia a fallos**

Tema 7.- Tolerancia a Fallos



1. Introducción
2. Fallos en entornos cliente/servidor
3. Grupos de procesos
4. Compromiso distribuido

Bibliografía: Capítulo 7 de Tanenbaum

1.1.- Terminología

- **Confiabilidad (Dependability)**
 - **Disponibilidad:** porcentaje de tiempo que el sistema funciona sin fallos.
 - **Fiabilidad (Reliability):** intervalo de tiempo sin fallos en el sistema.
 - **Seguridad (Safety):** si el sistema deja de funcionar correctamente, no ocurren catástrofes.
 - **Mantenibilidad:** reemplazo sencillo de elementos con fallos.
- Un sistema falla si no puede lograr sus objetivos
 - Un sistema falla porque ocurren **fallos** que producen **errores**
- **Tolerancia a fallos: que un sistema no falle aunque ocurran fallos.**

1.1.- Terminología



- **Tipos de fallos por su duración:**
 - **Transitorios:** El elemento solo falla una vez y después vuelve a funcionar correctamente
 - **Intermitentes:** El fallo en el elemento aparece de forma intermitente.
 - **Permanentes:** Una vez el elemento falla, ya no se recupera.

1.2.- Modelos de fallos

En general puede ocurrir cualquier tipo de fallo, pero se simplifica para poder diseñar sistemas y protocolos.

- El tipo de fallos que se consideran relevantes, constituyen el modelo de fallos.
- Tipos de fallos a considerar en los modelos:
 - **Fallos de parada:** El elemento que falla, simplemente deja de funcionar y no interfiere con el resto del sistema una vez falla.
 - **Fallos de omisión:** El elemento que falla no hace cierta parte de su cometido: p.ej: un canal de comunicación puede presentar fallos de omisión de envío o de respuesta.
 - **Fallos de temporización:** El elemento que falla no lo hace en el tiempo previsto
 - **Fallos de respuesta:** El elemento responde incorrectamente a las peticiones que se le realizan
 - **Fallos arbitrarios:** El componente que falla funciona de forma descontrolada.

1.3.- Enmascaramiento de fallos por redundancia

- **Enmarcar fallos:** los fallos ocurren, pero el sistema no los muestra hacia su entorno.
- **Técnica fundamental: redundancia**
 - **Información redundante:** información que solo es útil para comprobar que no hay fallos. p.ej: los CRC's
 - **Redundancia en el tiempo** (el número de veces que se realiza una operación)
 - **Redundancia de elementos:** Utilizar varios elementos iguales para intentar disminuir la probabilidad de que no se proporcione servicio:
replicación

Tema 7.- Tolerancia a Fallos



1. Introducción
2. Fallos en entornos cliente/servidor
3. Grupos de procesos
4. Compromiso distribuido

Bibliografía: Capítulo 7 de Tanenbaum

2.- Fallos en entornos cliente/servidor

- En la interacción entre un cliente y un servidor pueden ocurrir muchos tipos de fallos: en el servidor, en el canal y en el cliente:
 1. **El cliente no encuentra al servidor:** sin solución. Simplemente se lanza una excepción
 2. **La petición del cliente al servidor se pierde:** retransmisión después de cierto tiempo. Se deben detectar peticiones duplicadas en el servidor, para que las descarte.
 3. **El servidor falla:**
 1. Al recibir la petición, antes de procesarla (como 2)
 2. A mitad de procesar la petición: diferentes semánticas (próxima transparencia...)
 3. Al terminar el procesamiento, antes de responder (como 4)
 4. **La respuesta del servidor se pierde:** Difícil solución: retransmisión del mensaje del cliente, que el servidor debe identificar para responderle con los resultados previos. Aparece problema de recolección de residuos!
 5. **El cliente falla antes de recibir la respuesta:** Problema de peticiones huérfanas. Difícil solución.

2.- Fallos en entornos cliente/servidor

Falla el servidor a mitad de procesar la petición.

Semánticas posibles:

- **Semántica como mínimo una vez:** se debe garantizar que el servidor ejecuta el servicio como mínimo una vez. Como ha fallado, se debe esperar a que recupere, para reintentar la misma petición.
- **Semántica como máximo una vez.** No se reintentará la operación
- **Semántica exactamente una vez:** En general imposible de lograr. P.ej: la operación implica imprimir un documento. Cuándo sabe el cliente que fue impreso? En este ejemplo, solo posible si la impresora “colabora”.

Tema 7.- Tolerancia a Fallos



1. Introducción
2. Fallos en entornos cliente/servidor
3. Grupos de procesos
4. Compromiso distribuido

Bibliografía: Capítulo 7 de Tanenbaum

3.- Grupos de procesos

- Efectiva técnica de enmascaramiento de fallos: se replican los procesos.
- Un conjunto de procesos “idénticos” recibe y procesa cada petición. Opciones:
 - El cliente se queda con una de las respuestas. Se toleran fallos de parada.
 - El cliente se queda con la respuesta que proporcione la mayoría: Votación. Se toleran fallos arbitrarios.
- Son necesarios protocolos de comunicación a grupos que garanticen que los mensajes lleguen a todos los miembros del grupo o a ninguno: **Comunicación a grupos atómica.**

Tema 7.- Tolerancia a Fallos



1. Introducción
2. Fallos en entornos cliente/servidor
3. Grupos de procesos
4. Compromiso distribuido

Bibliografía: Capítulo 7 de Tanenbaum

4.- Compromiso distribuido

- Transacciones para agrupar un conjunto de peticiones efectuadas sobre un conjunto de elementos.
 - O todas tienen éxito: commit
 - Si alguna falla: rollback de todas las operaciones.
- Cada operación debe estar preparada a que se le ordene deshacer los cambios: No es posible en general si algún elemento no colabora (p.ej: una impresora)
- **Protocolos de compromiso:** Un proceso coordina la transacción y los demás participan con sus votos:
- **Compromiso en dos fases:** el coordinador envía PREPARAR y los demás responden OK o NOK. Si alguno respondió NOK, el coordinador envía ABORT. Si todos respondieron OK, el coordinador envía COMMIT.
 - Problema si el coordinador y un participante fallan antes de que el coordinador envíe COMMIT/ABORT. Se bloquea a los demás participantes, que no saben qué dijo el elemento que falló.
 - Solución: compromiso en 3 fases: Preparar, precommit y commit.