Replicación y consistencia

Tema 1

Gestión de fallos. Conceptos básicos

Índice

1. Modelo de sistema distribuido
2. Fallos: concepto y tipos
3. Fiabilidad
4. Disponibilidad
5. Recuperación
6. Confiabilidad

Bibliografía

[CKV01] Gregory Chockler, Idit Keidar, Roman Vitenberg: “Group Communication Specifications: A Comprehensive Study”, ACM Computing Surveys, 33(4):427-469, 2001.

[Cri91] Flaviu Cristian: “Understanding Fault-Tolerant Distributed Systems”, Communications of the ACM, 34(2):57-78, febrero 1991.

[HT93] V. Hadzilacos, S. Toueg: “Fault-Tolerant Broadcasts and Related Problems”. En S. J. Mullender, editor, “Distributed Systems”, capítulo 5, págs. 97-145. Addison-Wesley, 2ª edición, 1993.

[KV93] H. Kopetz, P. Veríssimo: “Real Time and Dependability Concepts”. En S. J. Mullender, editor, “Distributed Systems”, capítulo 16, págs. 411-446, Addison-Wesley, 2ª edición, 1993.

[Nel90] Victor P. Nelson: “Fault-Tolerant Computing: Fundamental Concepts”, IEEE Computer, 23(7):19-25, julio 1990.

[Sch93] F. B. Schneider: “What Good Are Models and What Models Are Good?”. En S. J. Mullender, editor, “Distributed Systems”, capítulo 2, págs. 17-26. Addison-Wesley, 2ª edición, 1993.

Índice

1. Modelo de sistema distribuido

- Ciertas características de un sistema distribuido son importantes a la hora de diseñar cualquier algoritmo:
  - Grado de sincronía.
  - Tipos de fallos que puedan ocurrir.
  - Pérdida de la conectividad.
- Definiremos los fallos en la próxima sección.
- Estudiaremos ahora la sincronía y el particionado.

1. Modelo de sistema distribuido

- Un sistema distribuido se considera sincrónico si [Sch93]:
  - Existe una cota sobre la diferencia de velocidad entre cada uno de los procesos que lo compongan.
  - Existe una cota sobre el tiempo de transmisión de los mensajes entre cualquier par de procesos.
- Un sistema se considera asíncrono si no cumple ninguna de estas restricciones, y parcialmente sincrónico si cumpliera sólo una de ellas.
  - Muchos problemas son irresolubles en sistemas distribuidos asíncronos.

1. Modelo de sistema distribuido

- Otra característica importante es la conectividad entre los componentes que definan el sistema.
- En caso de perderse tal conectividad y quedar grupos de procesos aislados aparece el problema de **particionado**.
- En [CKV01] se distinguen dos tipos de gestión del particionado:
  - **Sistema particionable**. La aplicación que se esté desarrollando permite que su ejecución continúe en cada subgrupo que haya quedado aislado. Posteriormente, al recuperarse de este fallo, ya reconciliará las diferencias existentes.
  - **Componente primario** (o **modelo de partición primaria**). Sólo se permite que continúe aquel subgrupo que tenga al menos más de la mitad de los procesos preconfigurados. Esto facilita que la consistencia se mantenga, aunque reduce la disponibilidad.

Índice

1. Modelo de sistema distribuido
2. **Fallos: concepto y tipos**
3. Fiabilidad
4. Disponibilidad
5. Recuperación
6. Confiabilidad

2. Fallos

- **Conceptos básicos** [Nel90]**:**
  - **Fallo (“Failure”):** Incapacidad para que un elemento desarrolle aquellas funciones para las que ha sido diseñado debido a errores en el propio elemento o en su entorno, que han sido causados por diferentes faltas.
  - **Falta / Defecto (“Fault”):** Condición anómala. Ejemplos: Errores en el diseño, problemas en la fabricación, interferencias, entradas no previstas, errores en el uso, corte del fluido eléctrico, etc.
  - **Error:** Manifestación de una falta en un sistema, donde el estado de un componente diferirá del previsto.

2. Fallos

- **Tolerancia a defectos** [Cri91]**:**
  - Un sistema tolera defectos cuando exhibe un comportamiento bien definido en caso de falta o enmascara un defecto en sus componentes a sus usuarios (es decir, continúa facilitando sus servicios estándar a pesar de la ocurrencia de esa falta o defecto).
  - Un servicio tolera defectos si los toleran todos aquellos servicios de los cuales dependa. Es decir, todos aquellos servicios que deba utilizar en algún momento.

2. Fallos

- **Clasificación (i)** [Cri91]**:**
  - **Fallo de omisión.** Cuando un servidor omite su respuesta ante una petición.
  - **Fallo de temporización.** Cuando la respuesta proporcionada es funcionalmente correcta, pero fuera del plazo preestablecido.
  - **Fallo de respuesta.** Cuando se proporciona una respuesta incorrecta. Tipos:
    - **De valor.** El valor proporcionado es incorrecto.
    - **De transición de estado.** Tras proporcionar la respuesta, el estado del servidor no es el correcto.

2. Fallos

- **Clasificación (ii):**
  - **Caída (*crash*).** Cuando tras un primer fallo de omisión, el servidor deja de proporcionar respuestas hasta que es reiniciado.

    Tipos, según el estado del servidor al reiniciar:
    - **Amnesia:** El servidor se reinicia en un estado preestablecido que no depende de las peticiones que atendió antes de fallar.
    - **Amnesia parcial:** El servidor recupera parte del estado que tenía antes de fallar, pero el resto toma valores predeterminados.
    - **Pausa:** El servidor recupera el mismo estado que tenía antes de fallar.
    - **Parada:** El servidor no se reinicia nunca y permanece parado.

2. Fallos

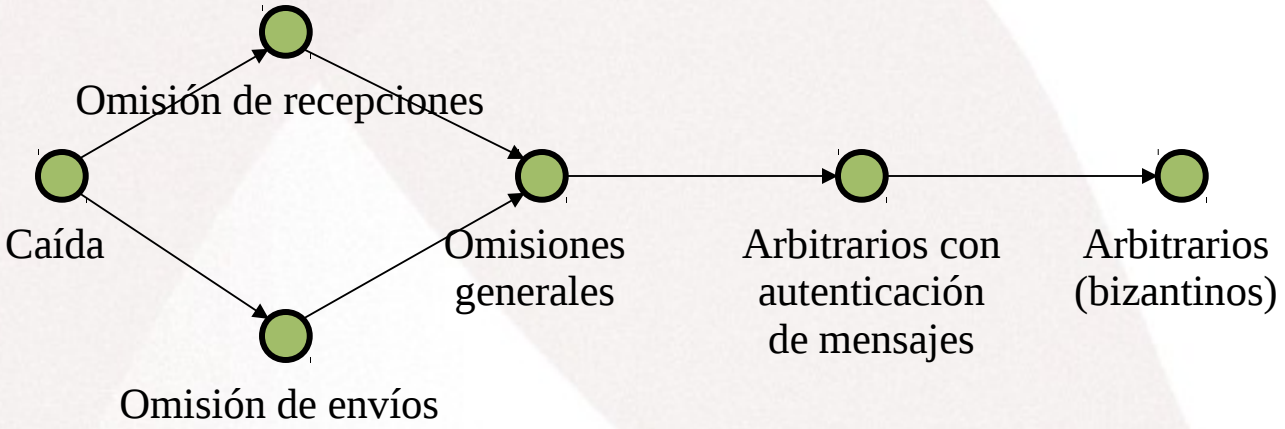
- Otra clasificación (i) [Sch93]:
  - **Fallo parada:** El procesador falla, parando. Una vez parado, permanecerá siempre en ese estado. Este hecho podrá ser detectado por otros procesadores.
  - **Caída (*crash*):** El procesador falla, parando. Una vez parado, permanecerá siempre en ese estado. Este hecho puede que no sea detectado por otros procesadores.
  - **Caída y enlace:** El procesador falla, parando. Una vez parado, permanecerá siempre en ese estado. Un enlace falla perdiendo algunos mensajes, pero no retrasa, duplica ni corrompe mensajes.

2. Fallos

- Otra clasificación (ii):
  - **Omisión de recepciones:** Un procesador falla recibiendo sólo un subconjunto de los mensajes que se le han enviado o parando y permaneciendo parado.
  - **Omisión de envíos:** Un procesador falla enviando sólo un subconjunto de los mensajes que debía enviar o parando y permaneciendo parado.
  - **Omisión general:** Se combina la omisión de recepciones y la omisión de envíos.
  - **Fallos bizantinos:** Un procesador falla exhibiendo un comportamiento arbitrario.

2. Fallos

- Tercera clasificación [HT93]:
  - A--->B, indica que A es menos severo que B.



Índice

1. Modelo de sistema distribuido
2. Fallos: concepto y tipos
3. **Fiabilidad**
4. Disponibilidad
5. Recuperación
6. Confiabilidad

3. Fiabilidad

- **Concepto** [Nel90]**:**
  - La fiabilidad, *F(t),* es la probabilidad condicionada de que un sistema pueda desarrollar sus funciones en el instante *t*, sabiendo que era operativo en el instante *t*=0.
  - La fiabilidad dependerá de:
    - Los defectos que puedan afectar al sistema.
    - Los mecanismos que posea el sistema para evitar que aparezcan fallos cuando se den defectos.
  - Para clasificar a un sistema como fiable, éste debe poseer mecanismos que eviten que se dé ningún fallo: **mecanismos de recuperación automática**.

3. Fiabilidad

- Numéricamente:

  F = P(sin defectos) + P(funcionamiento correcto/defectos) * P(defectos)

  donde:

  - F: Fiabilidad.
  - P(sin defectos): Probabilidad de que no haya ningún defecto. Dependerá de cómo se haya diseñado el sistema: esto influye en lo que puede considerarse falta o no.
  - P(funcionamiento correcto/defectos): Probabilidad de que el sistema funcione correctamente en caso de que haya habido un defecto. Puede aumentarse utilizando redundancia de componentes.
  - P(defectos): Probabilidad de que se dé algún tipo de defecto.

Índice

4. Disponibilidad

- Cuando el sistema no es totalmente fiable y su reparación costará cierto tiempo en el que no se puede proporcionar servicio.
- **Concepto de disponibilidad** [Nel90]:
  - Probabilidad de que un sistema o servicio esté operativo en un determinado instante.

$$Disp. = \frac{TMEF}{TMEF + TMDR}$$

  - Donde:
    - Disp.: Disponibilidad.
    - TMEF: Tiempo medio entre fallos.
    - TMDR: Tiempo medio de reparación.

4. Disponibilidad

- **Clases:**

Disponibilidad	Indisponibilidad anual	Clase
90 a 99%	entre 4 días y un mes	1
99 a 99.9%	entre 9 horas y 4 días	2
99.9 a 99.99%	entre 1 y 9 horas	3
99.99 a 99.999%	entre 5 y 60 minutos	4
99.999 a 99.9999%	entre 30 y 300 segundos	5
99.9999% a 99.99999%	entre 3 y 30 segundos	6

4. Disponibilidad

- **Alta disponibilidad:**
  - Cuando se exige que el tiempo total de indisponibilidad sea corto. Clases 5 y 6.
  - Además, el tiempo necesario para recuperar el servicio debe ser muy bajo.
  - Para obtener alta disponibilidad necesitamos replicación:
    - El fallo de algún componente no implica que falle el sistema.
    - Habrá otras réplicas que seguirán funcionando.

4. Disponibilidad

- **La indisponibilidad no siempre está causada por fallos:**
  - Pueden darse tareas de mantenimiento ya previstas que dejen momentáneamente sin servicio a los clientes.
  - Ejemplos:
    - Actualización o renovación del software que ofrece el servicio.
    - Adición de réplicas.
    - Reconfiguración.
    - Copias de seguridad.

Índice

5. Recuperación

- Para tener un sistema que tolere defectos, se deben emplear o combinar algunas de las siguientes técnicas:
  - Enmascarado.
  - Detección.
  - Contención.
  - Diagnóstico.
  - Reparación o reconfiguración.
  - Recuperación.

5. Recuperación

- **Enmascarado:** Corrección dinámica de los errores generados. El cliente no apreciará ningún fallo.
- **Detección:** Detección de un error, que es el síntoma de la ocurrencia de una falta.
- **Contención:** Evitar que un error se propague a otros servicios que dependan de aquél donde haya ocurrido.
- **Diagnóstico:** Identificación del módulo responsable de un determinado error que ha podido ser detectado.

5. Recuperación

- **Reparación - reconfiguración:** Eliminación o reemplazo de un componente afectado por una falta. Utilización de mecanismos que permitan continuar el servicio sin utilizar el componente afectado por la falta.
- **Recuperación:** Corrección del sistema para lograr un estado en que sea viable la prestación de servicios.

  Esto implica cierto tiempo de recuperación para llevar a cabo esta corrección. Durante ese tiempo el sistema puede que no esté disponible.

Índice

6. Confiabilidad

- La confiabilidad (“dependability”) fue definida en [KV93] como “un concepto que cubre los atributos no funcionales de un sistema en relación a su calidad de servicio durante un largo intervalo de tiempo”.
- Para medir tal confiabilidad, estos autores se centran en cuatro atributos:
  - Fiabilidad (“Reliability”).
  - Seguridad (“Safety”). Probabilidad de que un sistema no falle de manera catastrófica durante un intervalo de tiempo.
  - Disponibilidad (“Availability”).
  - Mantenibilidad (“Maintainability”). Probabilidad de que el sistema ya esté recuperado en el instante t si había fallado en el instante 0.

6. Confiabilidad

- Además, también reconocen que existe un quinto atributo importante:
  - Seguridad (“Security”): Capacidad de un sistema para evitar accesos no autorizados, tanto al propio sistema como a la información en él contenida.
- Sin embargo, este último tipo de seguridad no se puede medir cuantitativamente. No podemos asignar ninguna probabilidad, al igual que en el resto de atributos.