14.3. Servicios tolerantes a fallos

Cada gestor de réplicas se comporta de acuerdo a la especificación asociada a la semántica de los objetos que gestiona.
Un servicio que esté basado en la replicación se considera correcto si es capaz de responder a pesar de los fallos y si los clientes no pueden percibir la diferencia entre el servicio que obtienen de una implementación con datos replicados y otra proporcionada por un único gestor de réplicas correcto.

Linealizabilidad y consistencia secuencial

Un servicio de objetos compartidos replicados se dice linealizable si para cualquier ejecución existe algún entrelazamiento de las series de operaciones emprendidas por cada cliente que satisface los siguientes criterios:

- La secuencia entrelazada de operaciones cumple la especificación de una única copia correcta de los objetos.
- El orden de las operaciones de entrelazamiento es consistente con los tiempos reales en los cuales ocurrieron las operaciones en la ejecución real.

Para cualquier conjunto de operaciones de los clientes existe una ejecución canónica virtual sobre una única imagen virtual de los objetos compartidos. La linealizabilidad tiene que ver solamente con el entrelazamiento de operaciones individuales y que no está pensado que sea transaccional. Una ejecución linealizable podría romper las nociones de consistencia específicas de una aplicación si no se aplica un control de concurrencia.

Un servicio de objetos replicados compartidos se dice que es secuencialmente consistente si para cualquier ejecución existe algún entrelazamiento de las series de operaciones emprendidas por los clientes que satisfaga los siguientes criterios:

- La secuencia de operaciones intercaladas cumple la especificación de una única copia correcta de los objetos.
- El orden de las operaciones en el entrelazamiento es consistente con el orden en el programa con el cual cada cliente individual ejecutó dichas operaciones.

El orden de cada cliente no debe ser violado y el resultado de cada operación debe ser consistente con las operaciones que la han precedido. Cada servicio linealizable es secuencialmente consistente, ya que la ordenación por tiempo real refleja el orden dentro del programa de cada cliente. Lo contrario no se cumple.

14.3.1. Replicación pasiva (Primario - Respaldo)

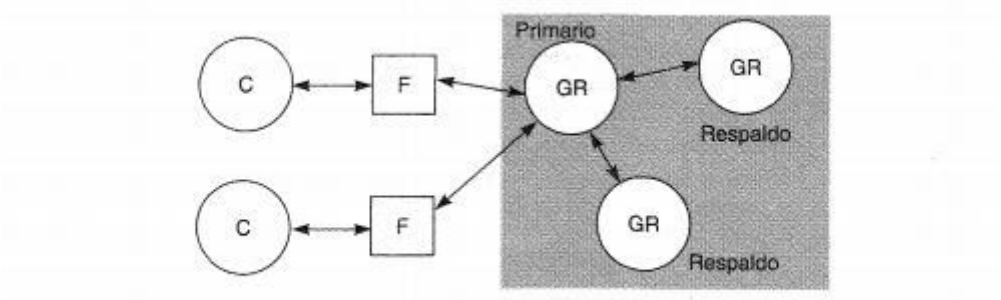

Figura 14.4. El modelo pasivo (primario-respaldo) para tolerancia a los fallos.

En cada instante existe un único gestor de réplicas primario y uno o más gestores de réplicas secundarios (respaldos o esclavos). Los frontales se comunican exclusivamente con el gestor de réplicas primario. El gestor de réplicas primario ejecuta las operaciones y

envía copias de los datos actualizados a los gestores de respaldo. Si el primario falla, uno de los secundarios toma su lugar.

1. *Petición:* el frontal lanza una petición con un identificador único al gestor de réplicas primario.
2. *Coordinación:* el primario acepta cada petición de forma atómica y en el orden en que las recibe. Comprueba el identificador y, en caso de que ya hubiera ejecutado la petición, simplemente reenvía la respuesta.
3. *Ejecución:* el primario ejecuta la petición y guarda la respuesta.
4. *Acuerdo:* el primario envía a todos los respaldos el estado actualizado, la respuesta y el identificador. Los respaldos envían un acuse de recibo.
5. *Respuesta:* el primario responde al frontal, que proporciona la respuesta al cliente.

Si el primario falla:

- El primario es reemplazado por un único respaldo
- Los gestores de réplicas que sobreviven se ponen de acuerdo en qué operaciones se habían realizado en el momento en el que el reemplazo del primario tomó control.

Los gestores de réplicas (primario y respaldos) están organizados como un grupo y el primario utiliza comunicación con vistas síncronas para enviar las actualizaciones a los respaldos.

Discusión sobre la replicación pasiva

El modelo primario - respaldo puede ser usado donde los gestores de réplicas primarios se comporten de una forma no determinista, debido a operaciones multihilo. En lugar de especificar las operaciones en sí mismas, los respaldos registran sumisamente el estado determinado únicamente por las acciones del primario.

Un sistema de replicación pasiva requiere f + 1 gestores de réplica. El frontal necesita poca funcionalidad para lograr la tolerancia a fallos. Necesita ser capaz de encontrar al nuevo primario cuando no responda el actual.

La replicación pasiva tiene la desventaja de producir sobrecargas grandes. La comunicación mediante vistas síncronas necesita varias rondas de comunicación por multidifusión, si falla el primario se introduce mayor latencia mientras el sistema de comunicación en grupo se pone de acuerdo sobre una nueva vista y la entrega.

Los clientes pueden enviar peticiones de lectura a las copias de seguridad, descargando así de trabajo al gestor primario. Se pierde la garantía de linealizabilidad, pero los clientes reciben un servicio secuencialmente consistente.

14.3.2. Replicación Activa

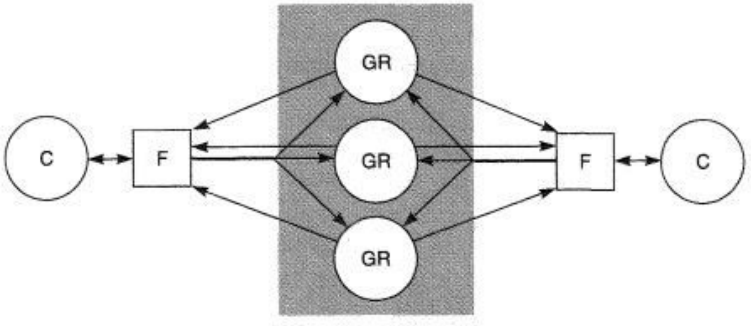

Figura 14.5. Replicación activa.

Los gestores de réplicas son máquinas de estado que desempeñan papeles equivalentes y se organizan como un grupo. Los frontales multi difunden sus peticiones al grupo de gestores de replicas y todos procesan la petición de forma independiente, pero idéntica, y responden. Si cae cualquier gestor, no tiene impacto en las prestaciones del servicio, puesto que los restantes gestores continúan respondiendo de forma habitual. Puede tolerar fallos bizantinos (extraños), puesto que el frontal recompila y compara las respuestas que recibe.

1. *Petición:* el frontal adjunta un identificador único a la petición y la multi difunde al grupo de gestores de réplicas, usando multidifusión fiable y ordenado. No emite la siguiente petición hasta que haya recibido respuesta.
2. *Coordinación:* el sistema de comunicación en grupo reparte la petición a cada gestor correcto en el mismo orden (total).
3. *Ejecución:* cada gestor de réplicas ejecuta la petición. Todos los gestores de réplicas correctos procesan idénticamente la petición. La respuesta contiene el identificador de la petición.
4. *Acuerdo:* no se necesita.
5. *Respuesta:* cada gestor de réplicas envía su respuesta al frontal. El número de respuestas que recoge el frontal depende de las suposiciones de fallo y del algoritmo de multidifusión.

Este sistema consigue la consistencia secuencial. La fiabilidad de la multidifusión asegura que todo gestor de réplicas correcto procesa el mismo conjunto de peticiones, y el orden total asegura que las procesan en el mismo orden. Todos finalizan en el mismo estado tras cada petición. Las peticiones de cada frontal son servidas en orden FIFO, que es el mismo que el orden del programa.

Si los clientes no se comunican con otros clientes mientras esperan por respuestas a sus peticiones, entonces éstas se procesan en orden *sucedió antes*. Si los clientes disponen de multi-hilo y pueden comunicarse con otros, para garantizar el procesado de peticiones ordenación *sucedió antes* debería reemplazarse la multidifusión por una que esté ordenada y a la vez total y causalmente.

No consigue linealizabilidad. El orden local en el cual los gestores de réplicas procesan las peticiones no es necesariamente el mismo que el orden impuesto por el tiempo real en el que los clientes hicieron sus peticiones. Schneider (1990) describe como el orden total en el que los gestores procesan las peticiones puede basarse en el orden las marcas temporales físicas que proporcionarán los frontales junto a sus peticiones.

Discusión sobre la replicación activa

Resolver la multidifusión fiable y totalmente ordenada es equivalente a resolver el problema del consenso. Esto requiere que el sistema sea síncrono o que se utilice una técnica como los detectores de fallos en un sistema asíncrono.

El sistema de replicación activa puede enmascarar hasta f fallos extraños, siempre que el servicio incorpore como mínimo 2f + 1 gestores de réplicas. Cada frontal espera a recoger f + 1 respuestas idénticas y entrega la respuesta al cliente. Descarta otras respuestas a la misma petición. Para estar totalmente seguros de qué respuesta está realmente asociada con qué petición se requiere que los gestores firmen digitalmente sus respuestas.

Todas las actualizaciones sobre los objetos replicados compartidos deben suceder en el mismo orden. En la práctica, algunas operaciones pueden conmutarse.

Un sistema de replicación activa debe ser capaz de explotar el conocimiento sobre conmutatividad para evitar el gasto de ordenar todas las peticiones. Los frontales pueden enviar peticiones de *sólo lectura* únicamente a gestores individuales. Pierden la tolerancia a fallos que proporciona el multi difundir las peticiones, pero el servicio mantiene la consistencia secuencial. El frontal puede enmascarar el fallo de un gestor de réplicas, simplemente enviando la petición de *sólo lectura* a otro gestor.

PREGUNTAS

1.- ¿Cuáles son los dos métodos de replicación de objetos compartidos?
A. Pasiva y activa.
B. Primario y respaldo.
C. Síncrono y asíncrono.
D. Linealizable y consistente.

2.- ¿Cuántos gestores de réplica necesita un sistema de replicación pasiva?
A. f + 1
B. f
C. 1
D. 2f + 1

3.- ¿Cuántos gestores de réplica necesita como mínimo un sistema de replicación activa?
A. 2f + 1
B. f
C. 1
D. f + 1

4.- ¿Cuál es una desventaja de la replicación pasiva?
A. Producir sobrecargas relativamente grandes.
B. Algunas operaciones pueden conmutarse.
C. Se pierde la linealizabilidad.
D. Se pierde la consistencia secuencial.

5.- ¿Qué implica el que un servicio sea linealizable?
A. Que es secuencialmente consistente.
B. El entrelazamiento de operaciones individuales es transaccional.
C. No se necesita un control de concurrencia.
D. El orden de las peticiones de cada cliente no importa.