Práctica 5

Servicio de almacenamiento distribuido

Javier Herrer Torres (NIP: 776609)
Javier Fuster Trallero (NIP: 626901)

Sistemas Distribuidos
Grado en Ingeniería Informática

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza
Curso 2020/2021

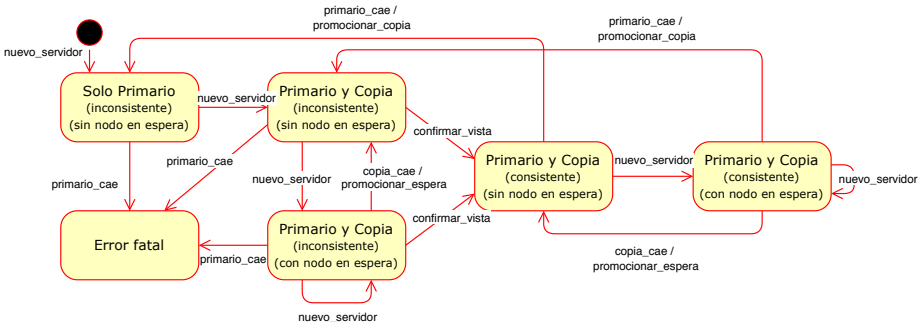

Figura 1: Diagrama de estados del Gestor de Vistas

1. Introducción

El objetivo de esta práctica es diseñar e implementar un servicio de almacenamiento distribuido clave/valor, en memoria RAM, que sea tolerante a fallos mediante el esquema Primario/Copia planteado en clases de teoría, y apoyándose en la implementación del servicio de vistas de la práctica 4.

La figura 1 muestra el diagrama de estados del Gestor de Vistas realizado en la práctica 4.

2. Servicio de almacenamiento

El servidor de almacenamiento podrá recibir los siguientes tipos de mensaje:

- `MsgTickInterno` enviará un latido al Gestor de Vistas para notificar de que «sigue vivo».
- `MsgVistaTentativa` el Gestor de Vistas responde a un latido con la vista tentativa. Si el primario recibe un cambio en la vista tentativa no la enviará como `MsgLatido` hasta que la copia le envíe `MsgConfirmacionTransferencia`.
- `MsgPeticion` enviada por un cliente del servidor de almacenamiento a un primario.
- `MsgPropagacion` enviada por un primario a una copia.
- `MsgRespuestaOperacion` contiene el valor de respuesta de la operación.
- `MsgSolicitudTransferencia` enviada por una nueva copia a un primario.
- `MsgTransferencia` contiene el almacén y el histórico de operaciones.
- `MsgConfirmacionTransferencia` enviada por la copia al primario una vez completada la transferencia.
- `MsgFin` causará la finalización del proceso.

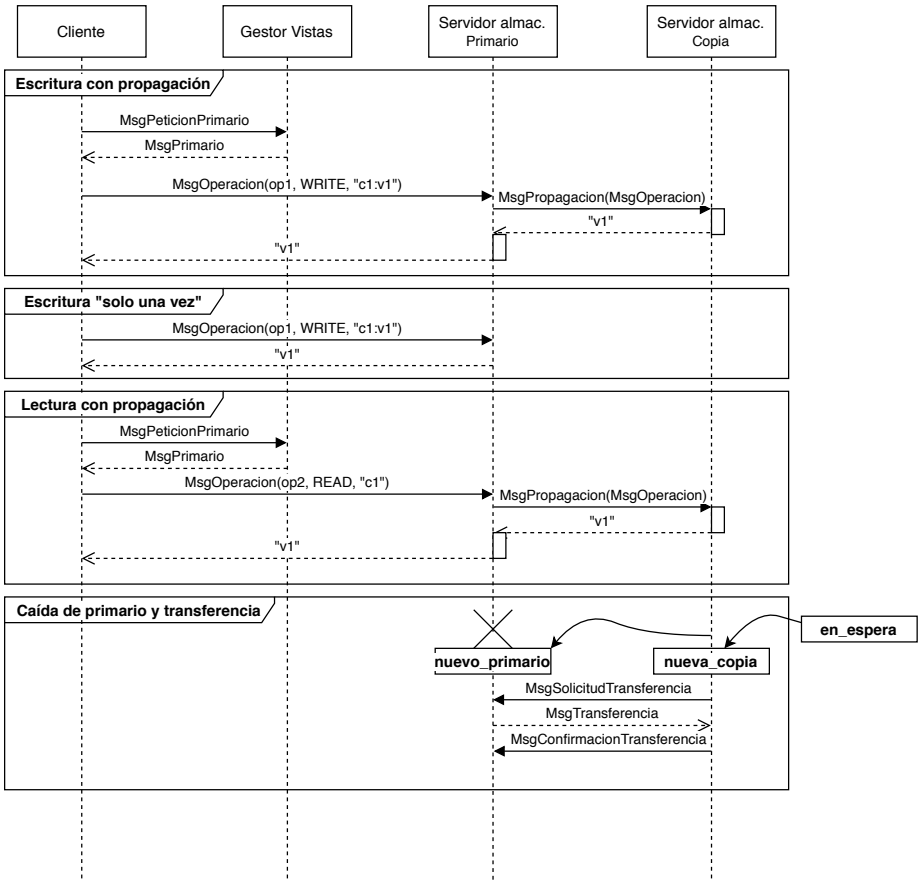

Figura 2: Diagrama de secuencia del Servidor de Almacenamiento

2.1. Procesamiento de peticiones

El primario aceptará peticiones de clientes, rechazando propagaciones. La copia aceptará propagaciones del primario, rechazando peticiones directas de clientes. En la figura 2 se muestra la operativa del procesamiento de una petición.

En primer lugar, el cliente pregunta al Gestor de Vistas el primario. Posteriormente, le envía a ese nodo la petición de operación (lectura o escritura). El primario, propagará sistemáticamente la petición a la copia y, una vez reciba la respuesta de su ejecución, responderá al cliente con el resultado de su ejecución.

2.2. Procesamiento de transferencias

Cuando un nodo es copia en la vista i, pero no lo fue en la vista $i-1$, debe solicitar al primario la transferencia completa del estado. El estado incluye el almacén de datos y el histórico de operaciones. En la figura 2 se muestra en el

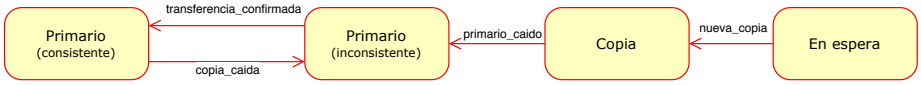

Figura 3: Diagrama de estados del Servidor de Almacenamiento

último caso la operativa de una transferencia.

Cuando un nodo en espera detecta que ha pasado a ser copia, envía una solicitud de transferencia al primario y este le responde con el estado. Hasta que el primario no reciba respuesta de confirmación de la copia, enviará al Gestor de Vista la vista antigua como latido.

2.3. Semántica «exactamente una vez»

Cada operación que realiza un cliente llevará consigo un identificador de operación autoincremental. A su vez, los nodos primario y copia almacenan en su histórico de operaciones el resultado de cada operación (identificador) *para cada nodo*.

Tal y como se aprecia en el segundo caso de la figura 2, cuando un primario detecta que ha recibido una petición anteriormente ejecutada, responde automáticamente al cliente con el valor que contiene el histórico *sin volver a ejecutar*.

3. Pruebas realizadas

3.1. Arranque correcto de máquinas

El fin de este test es comprobar que el lanzamiento de de los distintos servidores en distintas máquinas mediante *SSH* funciona correctamente. Para ello se ejecutan los binarios y se procede a su parada mediante un `MsgFin`.

3.2. La operación de escritura se realiza correctamente

Se busca comprobar que tras una caída de un servidor copia, el sistema funciona correctamente.

Para ello se han realizado escrituras concurrentes contra el sistema en una configuración completa, para posteriormente forzar la parada del nodo copia, comprobar que un nodo en espera es promocionado y que el sistema sigue permitiendo la escritura.

3.3. Escrituras concurrentes y comprobación de consistencia tras caída de primario

Se busca comprobar que tras una caída de un servidor primario, el sistema funciona correctamente.

Para ello, se han realizado escrituras concurrentes contra el sistema en una configuración completa, para posteriormente forzar la parada del primario, comprobar que la copia es promocionada a primario y que un nodo en espera es promocionado a copia. El sistema tras estos cambios sigue permitiendo la escritura.

3.4. Escrituras concurrentes y comprobación de consistencia tras caída de primario y copia

La finalidad de este test es comprobar que el estado inicial en configuración completa no se pierda tras una caída secuencial de primario y copia (si caen a la vez, el sistema es imposible que se recupere).

Para ello se realizan unas escrituras iniciales contra el sistema de almacenamiento, se envía `MsgFin` a primario, tras un tiempo prudencial para que el gestor de vistas pueda reconfigurar y se realice la transferencia del estado entre el nuevo primario y la nueva copia, se envía `MsgFin` al nuevo primario y se rearranca el nodo primario inicial. Se tienen que observar los mismos valores para las mismas claves que antes de la caída de los servidores y su reconfiguración.

```
{2, N1, N2} N1 recibe MsgFin
{3, N2, N3} N2 copia estado y recibe MsgFin, N1 es rearrancado
{4, N3, N1} Vista tras reconfiguración
```

3.5. Petición de escritura inmediatamente después de la caída de nodo copia

Se busca comprobar que no se responda a un cliente del servidor mientras no se pueda propagar su petición.

Para ello partiendo de una configuración completa, se envía un MsgFin al nodo copia y sin dejar tiempo para una correcta reconfiguración, se intenta realizar una escritura.

El resultado de dicha escritura debe ser un error con dos posibles tipos, error al obtener el primario del gestor de vistas al darse cuenta de la situación de inconsistencia o error al realizar la operación por parte del primario al no tener copia.

3.6. Petición de escritura duplicada por perdida de respuesta

Este test confirmará el correcto funcionamiento de lo expuesto en la sección 2.3. El cliente enviará una petición de escritura y, posteriormente se simulará una pérdida de respuesta volviendo a enviar la operación con el mismo identificador.

Además, para cerciorarse del funcionamiento deseado, se enviará un valor de escritura diferente al anterior que *no deberá ser escrito en el almacén* cuando el primario detecte que ya ha realizado una operación con ese identificador para ese nodo.

3.7. Comprobación de que un antiguo primario no debería servir operaciones de lectura

Se simulará la caída del primario enviando al Gestor de Vistas un latido 0 haciéndose pasar por el primario. Esto hará que el primario no sea conocedor de que ha dejado de ser primario, pero el Gestor de Vistas sí que realizará las acciones oportunas tras la caída de un primario. Es decir, promocionará a la copia y a un nodo en espera.

Ahora, el antiguo primario no podrá realizar operaciones y contestará a los clientes de almacenamiento con un error. Esto se debe a que, cuando intenta propagar la petición a la copia (ahora primario), esta no va a aceptar propagaciones.

4. Conclusiones

Se ha implementado un servidor de almacenamiento distribuido clave/valor basado en una aproximación primario/copia que podrá servir peticiones de clientes y además será tolerante a fallos gracias a la utilización de un gestor de vistas.

Cada servidor de almacenamiento contendrá en su estado el almacén de datos y el histórico de operaciones. Este histórico añadirá una funcionalidad de tolerancia a fallos adicional mediante la semántica «exactamente una vez». También se mantienen todos los otros mecanismos de tolerancia a fallos implementados en la práctica 4.

Por otro lado, mientras se está realizando la transferencia de estado, el sistema de almacenamiento no va a servir peticiones a clientes.