

Práctica 4 - Servicio de gestión de vistas

Sistemas distribuidos 2019 - 2020

Pedro Tamargo Allué - 758267
Juan José Tambo Tambo - 755742

Introducción

En esta práctica se va a implementar un sistema tolerante a fallos utilizando la estrategia del gestor de vistas visto en la asignatura.

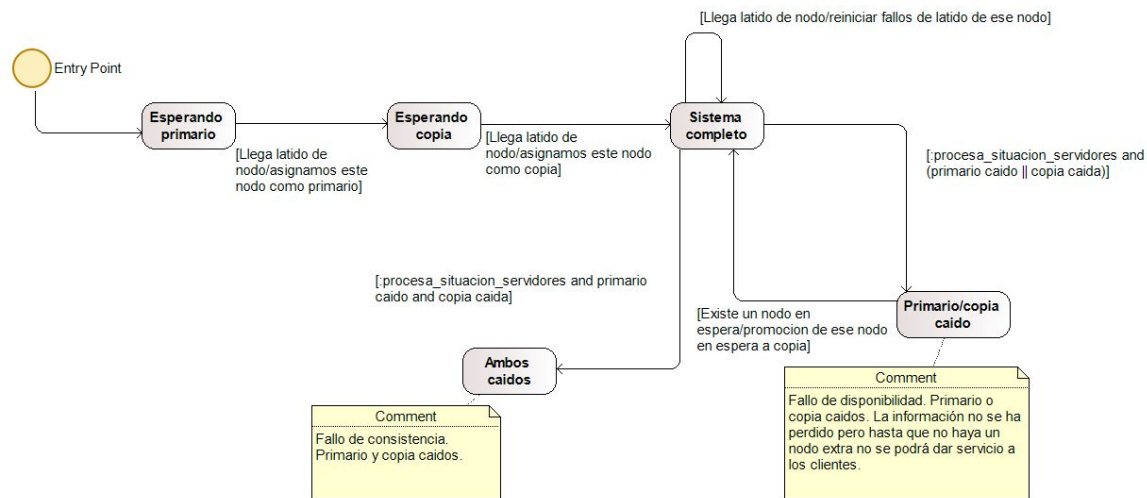
En esta práctica, el sistema solo será tolerante a un fallo ya que sólomente se va a implementar con un nodo copia.

El sistema primario/copia con gestor de vistas, se basa en un nodo primario y en la replicación de sus operaciones a un nodo copia (que mantendrá siempre la misma información que el primario). En este sistema también existe una lista de nodos en espera, es decir, aquellos nodos que no son ni primario ni copia (no intervienen de manera activa en las operaciones del sistema) pero envían latidos periódicamente para indicar que no han caído.

Cuando el nodo primario cae, el gestor de vistas debe crear una nueva vista, y promocionar al nodo que ocupaba la posición de copia a primario, ya que este contiene toda la información que contenía el primario. Si no existiera ningún nodo en espera para promocionarlo a copia, el primario dejaría de responder peticiones ya que efectuaría una parada por un fallo de disponibilidad.

En el caso de que se produzca una caída del primario y de la copia a la vez, el sistema dejaría de responder peticiones debido a un fallo de consistencia por pérdida de toda la información.

Autómata del gestor de vistas



Autómata utilizado para el gestor de vistas.

Al inicio, el gestor de vistas espera un latido $\{0, nodo\}$ para asignar *nodo* como primario. Una vez asignado el primario a la vista tentativa, esperaríamos un latido $\{0, nodo\}$ para asignarlo como copia a la vista tentativa. En el próximo latido del primario, recibirá la vista tentativa, y la validará al estar completa, es decir, tiene asignados primario y copia.

Tras esto, el estado sería “*Sistema completo*” ya que la vista tentativa y la vista válida son completas. En este estado, cuando lleguen latidos del nodo primario o de la copia se reiniciará su contador de latidos fallidos a 0, y en el caso de que llegará un latido $\{0, nodo\}$ siendo *nodo* un nodo que no es ni el primario ni la copia en la vista tentativa, se añadiría como un nodo en espera y se empezarían a monitorizar sus latidos de igual manera que los del primario y la copia.

Este estado se mantendrá hasta que llegue el mensaje del proceso *monitor*, para revisar el estado de los servidores. Aquí se detectan los servidores caídos (han fallado $@latidos_fallidos$ latidos).

Si se detecta un fallo del primario o la copia (uno de ellos), se cambia el estado (a *Primario/copia caído*), invalidando al nodo caído y promocionando (en el caso de que el caído fuera el primario) al nodo copia al nuevo primario. Tras esto, se buscaría un nodo para promocionar de “*nodo en espera*” a “*nodo copia*”. Si no existiese ningún nodo en espera, el primario no respondería peticiones por falta de un nodo copia (fallo de disponibilidad), hasta que llegara un latido $\{0, nodo\}$, de un nuevo nodo y se asignaría como copia.

Si se detectan fallos de primario y de copia a la vez, se cambia de estado a *Ambos caídos*, donde el sistema ya no responderá más peticiones porque ambos nodos (primario y copia) han caído, perdiendo así los datos (fallo de consistencia).

Diagrama de secuencia de iniciación del sistema

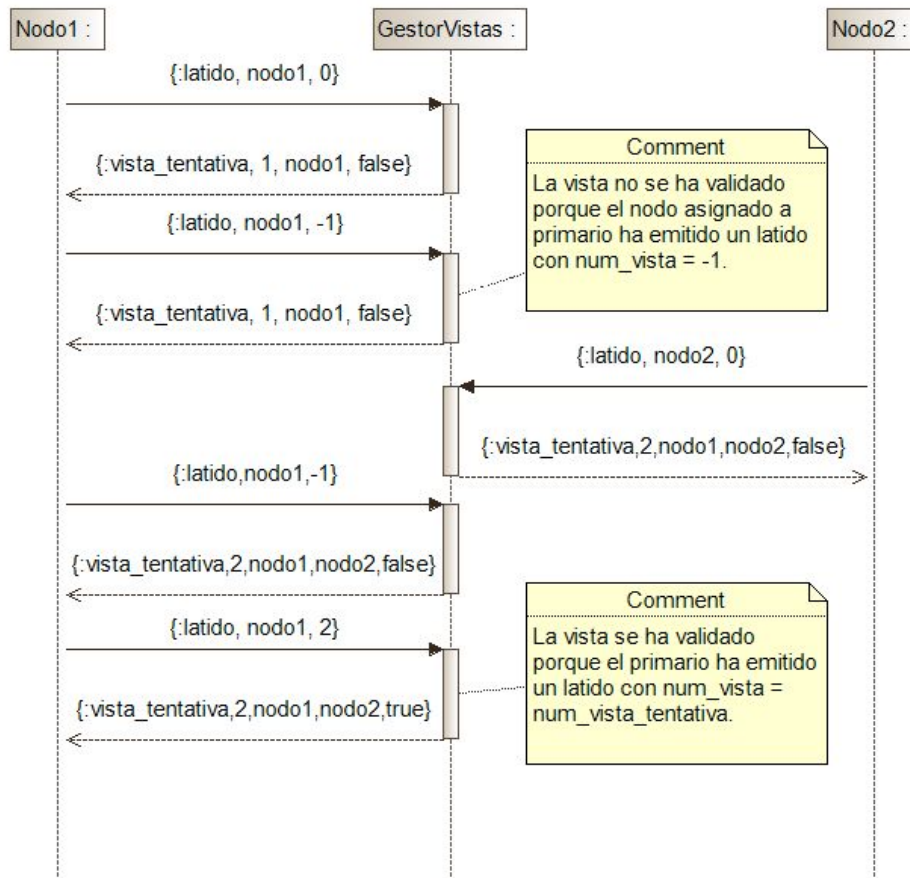


Diagrama de secuencia de inicialización del sistema.

En este diagrama se puede observar como el nodo primario no válida la primera vista hasta que esta no tiene asignado el primario y la copia.

Diagrama de secuencia de caída de nodo primario

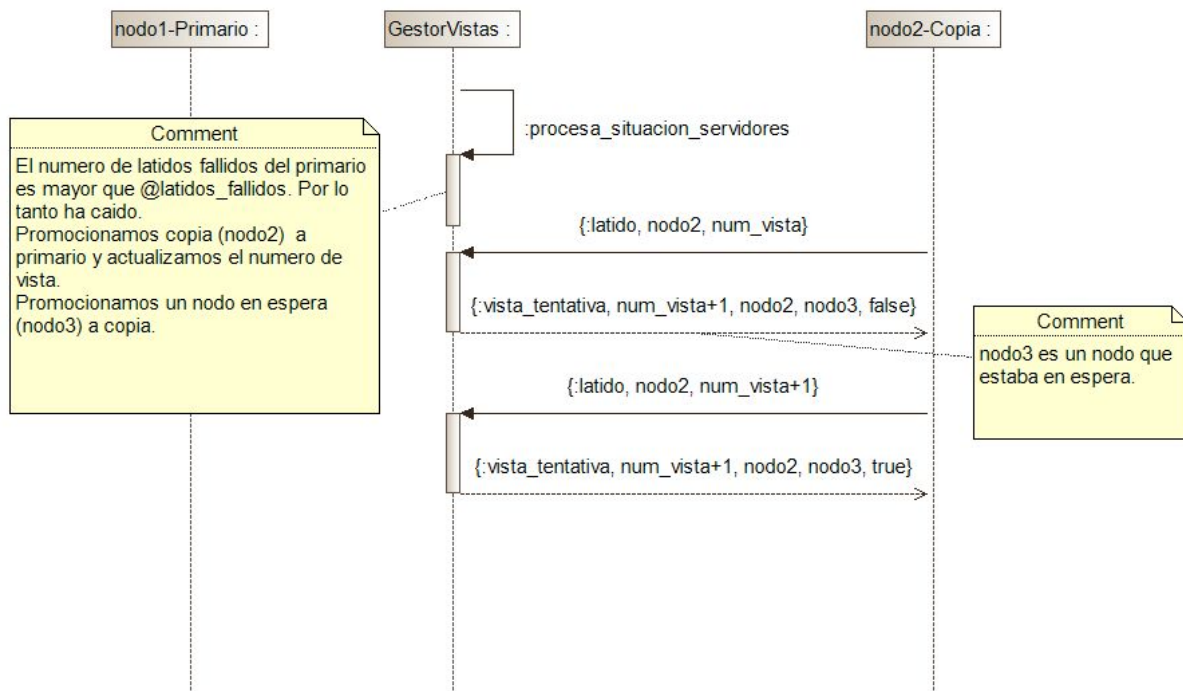


Diagrama de secuencia de la caída del nodo primario con `num_vista` en la vista válida y `tentativa = num_vista`.

En este diagrama de secuencia se puede apreciar cómo tras la llegada del mensaje `:procesa_situación_servidores` (enviado por el proceso monitor) se ha detectado la caída del nodo primario (*nodo1*). Tras esto el gestor de vistas se promociona a la copia (*nodo2*) y se utiliza un nodo en espera (*nodo3*) como nueva copia, consiguiendo así, la nueva vista.