Trabajo Práctico 2 - YPF Ruta

Programación Concurrente

108397 -	**Alejo Ordoñez**	github.com/alejoordonez02
105666 -	**Francisco Pereyra**	github.com/fapereyra
107863 -	**Lorenzo Minervino**	github.com/lminervino18
103376 -	**Alejandro Paff**	github.com/AlePaff

2c2025

Índice

Aplicaciones

Server

El servidor consiste de un sistema distribuido en el que la información con respecto a las cuentas y tarjetas se encuentra centralizada en un clúster compuesto de un *nodo líder* y *nodos réplica*. La entidad por default para un nodo del sistema es *estación*; esto es, todos los nodos cumplen con el rol de ser una estación y además pueden ser líder o réplica.
A nivel estación, los *surtidores* que residen en ella se intercomunican para mantener la funcionalidad de la estación, y lograr así una abstracción de los surtidores a nivel sistema global.

Arquitectura del servidor

Como ya se mencionó, el servidor está implementado de manera distribuida. El foco principal del diseño de la arquitectura está en reducir la cantidad de mensajes entre nodos que tienen viajar en la red.

Dado que la información se encuentra centralizada en el *clúster de consenso*, se vuelve necesario que las estaciones consulten el estado de la información de la cuenta a la cuál pertenece la tarjeta que quiere realizar el pago en ellas.
Para ésto, las estaciones conocen incialmente *quién* es el nodo líder. Cualquier consulta que precisen hacer se la envían al mismo. Si el líder dejara de funcionar, se ejecutaría entonces un algoritmo de elección de líder como *bully-algorithm* para elegir un nuevo de entre las réplicas, para luego actualizar a todas las estaciones con el resultado de la elección.

De esta manera todos los nodos necesitan un único socket para comunicarse con el nodo líder, salvo éste último que necesita tantos sockets como estaciones existan además de él. Esto ocurre a nivel lógico, ya que estaciones poco concurridas no necesitan estar constantemente conectadas con el nodo líder, por lo que el mismo tiene la posibilidad de mantener sólo un top N conexiones con el resto de las estaciones. Cuando pasa un tiempo sin que se envíen mensajes del sistema, la conexión se cierra para ahorrar recursos.

Clúster de surtidores.

Los surtidores en una estaicón se conectan directamente al servidor que ejecuta la funcionalidad de nodo en el sistema global-los surtidores no son computadoras, son hardware que envía I/O al servidor de la estación-, por lo que la concurrencia en éste clúster es a nivel memoria. Para prevenir las race conditions que surgen del acceso concurrente de lecto escritura a memoria, se utiliza el modelo de actores.

Clúster de consenso

El clúster de consenso está conformado por un único nodo líder y N réplicas de la información que éste contiene. Si el líder deja de funcionar, las réplicas lo detectan y inician la re-elección mediante un *bully-algorithm*.
Para evitar desincronización en casos falla de alguno de los nodos del clúster de consenso, se utiliza algoritmo de sincronización de transacciones *two-phase commit*.

Clúster de estaciones (*sistema global*)

El **clúster de estaciones** se refiere a todos los nodos que se ejecutan en las estaciones de YPF. Todos las estaciones deben cumplir con éste mínimo rol: poder realizar el cobro de cargarle nafta a un conductor de YPF Ruta.

1. *Un conductor usa su tarjeta por primera vez en el surtidor de una estación.*

A nivel estación, quien recibe la responsabilidad de realizar el cobro a una tarjeta es un surtidor. Como la tarjeta no se encuentra aún cargada en el sistema, cuando el nodo estación envía la consulta sobre la disponibilidad de saldo de la misma (o de su cuenta), el nodo líder genera el registro de la tarjeta, así como también de la cuenta a la que esta pertenece si no existiera aún; y envía el nuevo registro a los nodos réplica.
Si la estación es el nodo líder entonces el checkeo se realiza en memoria en vez de mediante un paquete de red. Los nodos réplica no tienen ningún comportamiento especial fuera del flujo que se sigue al registro de la tarjeta-envían la consulta por red al líder.

Política de cobro en estaciones sin conexión

Si una estación se encuentra sin conexión, no hay nada que hacer si se trata de un nodo que está fuera del clúster de consenso. Se puede o bien realizar el cobro o no.
Tampoco hay mucho más que hacer cuando cuando se trata de un nodo réplica o líder, más que tomar una política de asumir que la información que se tiene está actualizada o no. En el primer caso, el nodo réplica o líder, revisa el saldo restante de la tarjeta (y de la cuenta a la que pertenece) y realiza el cobro en base a esa información-si no hay saldo suficiente niega la operación. En el segundo caso, la operación se lleva a cabo sin revisar el registro de la tarjeta (ni el de la cuenta a la que pertenece).

Sin importar el nodo o la política que se le aplique, en caso de que el cobro finalmente se efectúe, la actualización del registro de la tarjeta debe ser encolada para poder ser enviada al líder del clúster de consenso una vez recuperada la conexión.

Si fuera el nodo líder el que perdió la conexión, entonces cuando la recuperase, ya se habría elegido a otro y por tanto sería a ese nuevo líder al que se enviarían las actualizaciones encoladas si así las hubiera.