

# “Iteración 5”

Maria F. Mena Barrera, Susan P. Joven Vasquez, Jorleidis P. Monsalvo Torres,

Rafael A. Forero Alvarado, Camilo Santacruz, David L. Avendaño

Transacciones distribuidas

Universidad de los Andes, Bogotá, Colombia

{mf.mena, sp.joven, jp.monsalvo, ra.forero11, c.santacruz, dl.avendano}@uniandes.edu.co

Fecha de presentación: Diciembre 11 de 2017

## Tabla de contenido

1	Análisis de requerimientos .....	1
1.1	Funcional 18.....	1
1.2	Funcional 19.....	2
1.3	Funcional de consulta 13 .....	3
1.4	Funcional de consulta 14 .....	3
2	Restricciones existentes.....	4
3	Análisis general de RotondAndes .....	4

## 1 Análisis de requerimientos

### 1.1 Funcional 18

- Grupo 1

Two Face Commit: Se puede asegurar la regla de negocio de que no se pueda hacer el pedido si no se pueden confirmar todos los productos de este. Gracias a que mediante este protocolo podemos esperar los votos de las cortes con la respuesta de confirmación o de abortar y con esto se podría deshacer la transacción en todos los restaurantes involucrados.

Colas de mensajes: Con este protocolo se tendrían que confirmar los productos de cada pedido, pero si llegara a faltar algún producto el rollback que se tendría que ejecutar es más difícil, ya que se tendría que llenar la cola de mensajes nuevamente con las instrucciones para deshacer la transacción, no se podría hacer automáticamente en todos los servicios

- Grupo 2

Colas de mensaje: en el requerimiento RF18 se solicita que todas las unidades respondan de forma inmediata a la solicitud en caso contrario esta debe quedar en estado "rechazada". Debido a que una cola de mensajes es asincrónica al hacer una solicitud no se espera una respuesta de forma inmediata, por lo cual si se realiza con colas de mensaje habría que colocar una restricción para al momento en el que la solicitud entre a la cola la respuesta sea de manera inmediata y verificar para cada una de las bases de datos que la respuesta sea coherente.

Two Phase Commit: usar este protocolo para la implementación del requerimiento RF18 sería útil ya que este trabaja de forma sincrónica por lo que al momento de hacer la solicitud la respuesta debe ser de forma inmediata y se tendría la respuesta de las 3 unidades participantes. Sin embargo, el requerimiento RF18 se puede realizar aun cuando una de las unidades no haya dado respuesta positiva por lo que habría que implementar el protocolo de tal manera que realice la transacción cuando al menos 2 de las unidades hayan dado respuesta correcta.

- Grupo 3

Cola de mensajes: Enviaría mensajes a las otras máquinas remotas preguntando si tienen productos (que no están disponibles localmente) disponibles. El problema es que se necesita una respuesta inmediata, de modo que sería necesaria una restricción en el tiempo de respuesta y en la prioridad del método para así cumplir rápidamente con requerimiento.

Two Phase Commit: Funcionaría este proceso, pero se complicaría debido a la concurrencia de varios pedidos mesa ya que el protocolo two phase commit bloquea los recursos hasta que la transacción termine.

## 1.2 Funcional 19

- Grupo 1

Two Face Commit: Con este protocolo se puede tener la confirmación de que en todas las coretes se realiza la búsqueda y el retiro del restaurante, pero si alguna de las cortes se llegara presentar un fallo como no votar, la transacción abortaría para todas las instancias de la rotonda sin retirar ninguno de los restaurantes (ya que puede que en las rotondas estén los mismos restaurantes).

Colas de mensajes: Se pone en la cola la solicitud de retirar el sistema un restaurante y se cada servicio responde a la solicitud buscando el restaurante en su sistema y si encuentra lo retira (protocolo que es independiente de cada una de las rotondas) No es necesario la confirmación de todas las aplicaciones para asegurar los cambios; se puede retirar el restaurante de una de las aplicaciones sin depender de la otra.

- Grupo 2

Colas de mensaje: si este requerimiento se realiza mediante colas no se podría garantizar que se elimine el restaurante solicitado en todas las unidades participantes ya que la solicitud entra en la cola y a medida que lo van leyendo las bases de datos de la cola van realizando la transacción y puede que alguna no pueda realizarlo y las otras sí.

Two Phase Commit: este protocolo sería bueno en la implementación del requerimiento RF19 debido a que si una de las bases de datos envía un abort entonces todas las demás deben abortar la transacción también lo que no permite que el restaurante se elimine en unas bases de datos si y en otras no, dejando las bases de datos en estados coherentes.

- Grupo 3

- a. Cola de mensajes: Pone en cola de cada una de las máquinas el mensaje de retiro del restaurante “X”, en donde cada una de las rotondas independientemente cumplen este retiro siguiendo sus propias restricciones.
- b. Two Phase Commit: Boquea el recurso restaurante “X” en todas las rotondas e intenta eliminarlo, si alguna de las rotondas informa que no puede eliminarlo porque está siendo usado u otra razón, se hace un rollback y no elimina el restaurante de ninguna rotonda

### **1.3 Funcional de consulta 13**

- Grupo 1

Two Face Commit: Con este protocolo se garantiza obtener la respuesta de los tres servicios (si no es así no se obtiene ninguna y se aborta la transacción completamente ya que se especifica que se quieren obtener los productos de las 3 rotondas)

Colas de mensajes: Con este protocolo se hace la consulta local y después juntarla con las respuestas obtenidas de la distribuida; si no todos los servicios contestan no se tendría inconveniente si se tiene conciencia de que servicio (s) fue (ron) el(los) que no contesto (contestaron) o si esto no es requerido. Se mostraría a el usuario los productos que se lograron obtener de las otras rotondas.

- Grupo 2

Colas de mensajes: ya que se tiene que tener respuesta de las 3 unidades participantes, si alguna de las unidades no puede contestar se debe saber al menos a quienes pertenecen las respuestas almacenadas en la cola de mensaje. Por ser asíncronas no se garantiza que se vaya a obtener la respuesta a la consulta incluyendo la de las otras unidades de forma inmediata.

Two Phase commit: este protocolo ayuda a garantizar que las tres unidades participantes den respuesta a la consulta, por lo que se estaría cumpliendo que la respuesta final incluya las respuestas de las tres unidades.

- Grupo 3

- c. Cola de mensajes: Envía a las colas la solicitud de retornar todos los productos, ellos retornan los productos que tienen y se van sumando a la respuesta, si alguna de las rotondas no puede devolver dar la información se deberá hacer un rollback puesto que no se cumpliría la consulta.
- d. Two Phase Commit: Se hará la consulta en todas las rotondas, bloqueando la tabla productos en todas las rotondas hasta que termine la transacción, si una no puede dar la información hace rollback de todas las sub-transacciones.

### **1.4 Funcional de consulta 14**

- Grupo 1

Two Face Commit: Con este protocolo se garantiza obtener la respuesta de los tres servicios (si no es así no se obtiene ninguna y se aborta la transacción completamente). Cosa que es buena porque siempre que se pueda realizar la consulta estaría la rentabilidad de toda la super-rotonda, pero si se quiere tener un estimado de solo la información que se consigue no se podría obtener.

Colas de mensajes: Con este protocolo se hace la consulta local y después juntarla con las respuestas obtenidas de la distribuida; si no todos los servicios contestan no se tendría inconveniente si se tiene conciencia de que servicio (s) fue (ron) el(los) que no contesto (contestaron) o si esto no es requerido. Se le podrían presentar resultados parciales a el usuario que realizo la consulta.

- Grupo 2

Colas de mensajes: el requerimiento RFC14 se podría implementar con colas de mensajes se va guardando la respuesta de cada base de datos y luego el cliente toma todas las respuestas disponibles en la cola, pero si no se sabe a quién pertenece cada respuesta podrían presentarse problemas.

Two Phase commit: implementar el requerimiento RFC14 con este protocolo sería una muy buena opción ya que se puede garantizar que en la respuesta final se incluya la individual de cada una de las unidades participantes.

- Grupo 3

- a. Cola de mensajes: Hace la petición en cada rotonda para un restaurante específico por este motivo no puede devolver la respuesta sin tener la información de todas rotondas con ese restaurante.
- b. Two Phase Commit: Valida que todas las rotondas estén disponibles y que puedan enviar la información de la consulta, por esta razón nos garantiza que la respuesta sea completa o que no se haga la consulta.

## **2 Restricciones existentes**

1. En todas las solicitudes, todas las unidades participantes deben dar respuesta.
2. Después de cada transacción la base de datos debe estar en estado coherente.
3. Existe un tiempo límite para que cada sub-transacción pueda retornar la información.
4. Todos los participantes deben estar activos y disponibles para poder realizar las transacciones por el protocolo de Two Face Commit

## **3 Análisis general de RotondAndes**

En general, la mejor opción para cumplir con las reglas de negocio de rotondAndes es implementar los requerimientos con el protocolo Two Phase Commit ya que est garantiza que todas las unidades participantes respondan a la solicitud y si alguna aborta entonces todas abortan, lo que ayuda a garantizar un estado coherente de las bases de datos después de cada transacción.