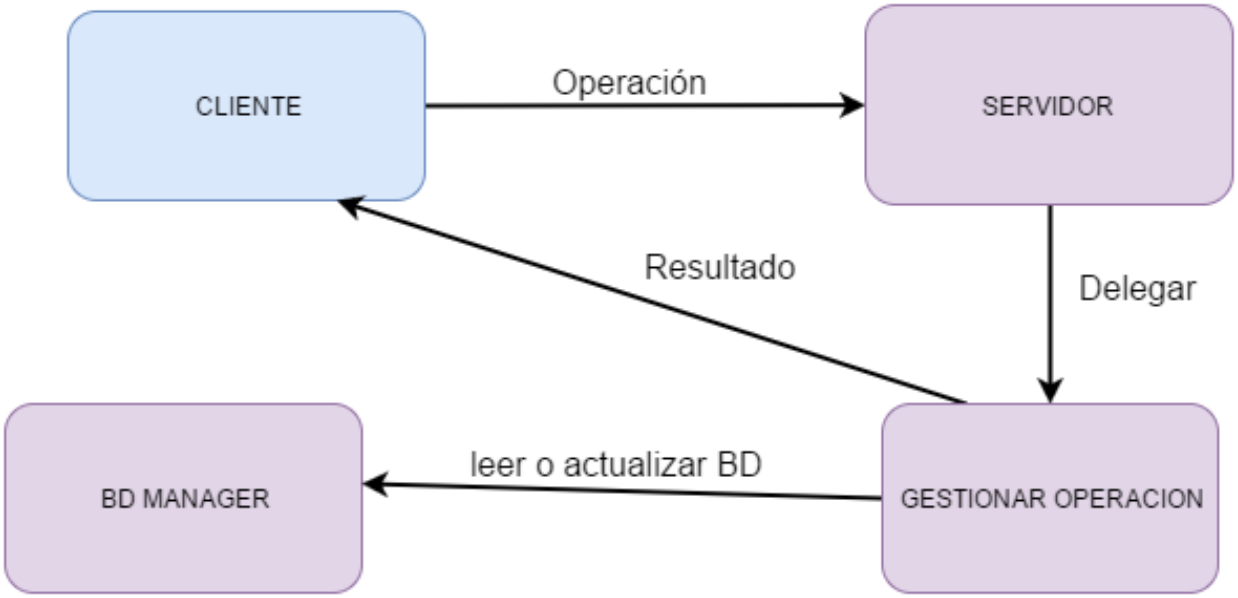
Algoritmo de Paxos

Trabajo de redes y sistemas distribuidos. Parte 2

Autor: Íñigo Alonso

Esquema general de entidades distribuidas



Los agentes que intervienen en este esquema son:

- El **cliente**: su API permite realizar operaciones de lectura y escritura. Este se relaciona con los servidores que conozca, a los cuales les pedirá que realicen su operación, buscando alguno que le responda y pueda llevarla a cabo.
- El **servidor**: Recibe peticiones de lectura y de escritura. Una vez a comprobado que la operación se puede llevar a cabo (Paxos no está caído), delega la operación a un proceso nuevo.
- Proceso que **gestiona** las operaciones: Lleva a cabo peticiones a Paxos para manejar las instancias necesarias para llevar a cabo la operación. Comprueba que cuando Paxos a aceptado la instancia que ha propuesto, contenga realmente el valor y la referencia de la operación que ha querido proponer.
- BD **manager**: Gestiona la BD de datos y la BD de referencias de operaciones llevadas a consenso, pero aun no respondidas. Se encarga de eliminar duplicados al actualizar la BD de datos, y de revivir peticiones de lectura de BD de datos y actualizaciones.

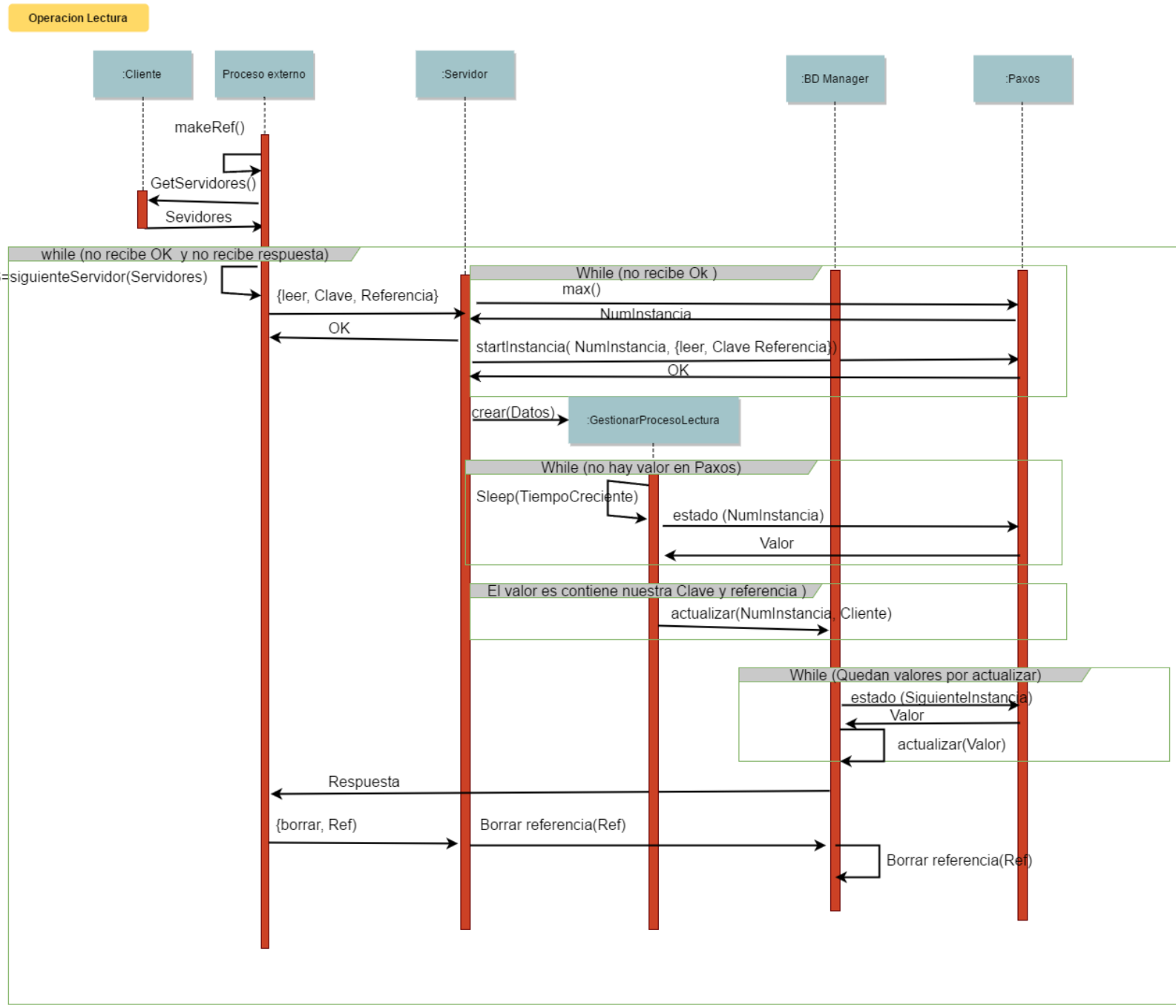
Lo más importante, es entender el siguiente apartado entero.

Funcionamiento e Interacción de procesos

Explicación del funcionamiento y proceso de una operación.

1. Desde la Api se llama a leer o escribir y se genera una referencia única para esta operación.
2. El cliente devuelve los servidores disponibles.
3. Se van realizando peticiones a los distintos servidores hasta que alguno responda con un OK.
4. Ahora, este servidor llevará a cabo la operación. Pero, si en un cierto tiempo no responde con el resultado, se vuelve a realizar peticiones a otros servidores.
5. El servidor comprueba que Paxos está vivo y pide el máximo de número de instancia vista y delega la operación a un nuevo proceso pasándole toda la información del estado y ese número de instancia para la operación.
6. Este proceso entra en un bucle donde, cada vez que entra se queda dormido esperando a que Paxos lleve a consenso su petición. Cada vez que se duerme, será con un tiempo mayor.
7. Una vez se tiene que Paxos ha llegado a consenso con ese número de instancia, se comprueba que tenga el valor de la referencia de la operación, es decir, que sea el que el cliente ha pedido. Si no es así se vuelve al paso número 5.
8. Si coincide, pide al BD manager que actualice BD de datos hasta ese número de instancia.
9. El BD manager pedirá a Paxos todos los valores que le faltan para actualizar la BD de datos (Las lecturas no editan la BD, las escrituras sí), siempre comprobando con la BD de referencias vistas, que se eliminen posibles duplicados de operaciones y solo aplicarlas una vez.
10. El BD manager es el que devuelve el resultado al cliente. Para las lecturas, una vez actualizada la BD de datos, lee el valor de allí y lo devuelve. Para las escrituras de hash, debe actualizar hasta el anterior número de referencia para no perder el dato anterior a la escritura.
11. Una vez el cliente recibe la respuesta, envía al servidor una petición para borrar la referencia de su operación.
12. El servidor llevará a consenso a través de Paxos esta petición de borrar la referencia, para que sea visible y consistente a todos los demás servidores realizándolo de la misma forma que las demás operaciones (pasos 5-7).

Entre operaciones de lectura y escritura tan solo se diferencia en cómo se actualizan en la Base de datos, y cómo se responde.


Operacion Lectura
:Cliente
Proceso externo
:Servidor
:BD Manager
:Paxos
makeRef()
GetServidores()
Sevidores
while (no recibe OK y no recibe respuesta)
S=siguienteServidor(Servidores)
{leer, Clave, Referencia}
While (no recibe Ok)
max()
NumInstancia
OK
startInstancia(NumInstancia, {leer, Clave Referencia})
OK
crear(Datos)
:GestionarProcesoLectura
While (no hay valor en Paxos)
Sleep(TiempoCreciente)
estado (NumInstancia)
Valor
El valor es contiene nuestra Clave y referencia)
actualizar(NumInstancia, Cliente)
While (Quedan valores por actualizar)
estado (SiguienteInstancia)
Valor
actualizar(Valor)
Respuesta
{borrar, Ref)
Borrar referencia(Ref)
Borrar referencia(Ref)

Funciones API

Funciones utilizables de forma externa del cliente:

Star(Nodo)t: Pone en marcha un cliente.

Stop(Nodo): Para en marcha un cliente.

Lee(Nodo, Clave): Devuelve el valor de dicha clave en la base de datos Clave-Valor del servidor.

Escribe(Nodo, Clave, Valor): Escribe el valor en dicha clave en la base de datos Clave-Valor del servidor.

Escribe_hash(Nodo, Clave, Valor):: Escribe el valor en dicha clave en la base de datos Clave-Valor del servidor y devuelve el valor en la base de datos previa escritura.

Funciones utilizables de forma externa del servidor:

Sordo(Nodo)t: Pone sordo al servidor.

Escucha(Nodo): Hace que el servidor vuelva a escuchar

Fiable(Nodo): Hace que los mensajes que recibe el servidor sean fiables.

No_fiable(Nod): Hace que los mensajes que recibe el servidor no sean fiables.

Protocolos de interacción

Existen muy pocos mensajes en esta implementación, la clave está en la gestión e implementación interna (apartado Funcionamiento e Interacción de procesos).

Procesos externos (API) <-> Cliente:

-> {get servidores, From}: Se piden los servidores disponibles, el cliente contesta con <- {servidores cliente, Servidores}

Cliente <-> Servidor:

El cliente envía una petición de una operación (leer, escribir o escribir hash}: ->{Operación, Clave, IPCliente, Referencia}, y el servidor le responde con <- Ok o con <- caído, dependiendo del estado de Paxos.

El cliente, una vez realizada la operación, le avisa al servidor que puede borrar su referencia -> {borrar ref, Referencia, IPCliente}

Servidor <-> GestionarOperación:

La única relación que existe entre estos dos procesos es la de creación y delegación desde el servidor al proceso que gestiona la operación donde el pasa todos los datos necesarios y, cuando la instancia a llevar a consenso, no tiene el valor esperado, el proceso que gestiona la operación vuelve a llamar al servidor de igual forma que el cliente operación {leer, escribir o escribir_hash}: ->{Operación, Clave, IPCliente, Referencia}

GestionarOperación <-> BD manager:

->{actualizar_bd, NumInstancia, PidGestionador}: Pide al gestor de la BD que la actualice hasta el número de instancia correspondiente, una vez lo hace, devuelve la BD de datos y de referencias <- {actualizado, BD}

Cliente <- GestionarOperación:

Una vez actualiza la BD por una petición de una operación, le devuelve el resultado correspondiente al cliente -> {respuesta_escritura, Value} o {respuesta_lectura, Value}.

Además, todos los procesos utilizan la API de Paxos.

Tests realizados

- Untest que comprueba que las operaciones básicas funcionan de forma coherente y consistente.
- Un test que comprueba que el servicio funciona con mensajes no fiables
- Un test que comprueba que funciona con servidores sordos durante ciertos intervalos
- Untest que comprueba que el servicio funciona de forma correcta con clientes concurrentes.