Réplica bidireccional basada en control de cambios

Bidirectional data replication based on data change control

Jorge Landrian García

Universidad de las Ciencias Informáticas

jlandrian@uci.cu

Resumen

El mecanismo de réplica de datos descrito en este documento, se basa en el control de cambios en las bases de datos relacionales y la sincronización de las mismas. Esta solución permite la implementación de sistemas de base de datos distribuidos, aislando la complejidad del sistema del mecanismo de replicación de los datos. Dos de los objetivos principales de esta solución son simplificar y optimizar el proceso de sincronización de la información entre servidores de base de datos.

Palabras clave: Actualización de datos, base de datos, réplica, sincronización, sistemas distribuidos.

Abstract

The data replication mechanism described in this document, is based on the data change control in relational databases and the data synchronization between databases. This solution allows the implementation of distributed database systems, isolating the complexity of the system from the data replication mechanism. Two of the main objectives of this solution are to simplify and optimize the process of synchronizing information between database servers.

Key words*: Data updating, data base, replication, synchronization, distributed systems.*

Introducción

En sistemas de gestión y enrolamiento de tamaño mediano o grande es muy común encontrarse varias bases de datos distribuidas geográficamente, sistemas bancarios, de venta, gubernamentales e incluso sistemas que tienen un alcance global y poseen servicios en distintos países. Muchos de estos sistemas con el objetivo de garantizar la mayor disponibilidad posible, funcionan como un conjunto de silos de información, y uno de los puntos mas importantes es la sincronización de la información entre estos silos de datos. La forma más común de sincronización de la información es mediante un mecanismo de réplica.

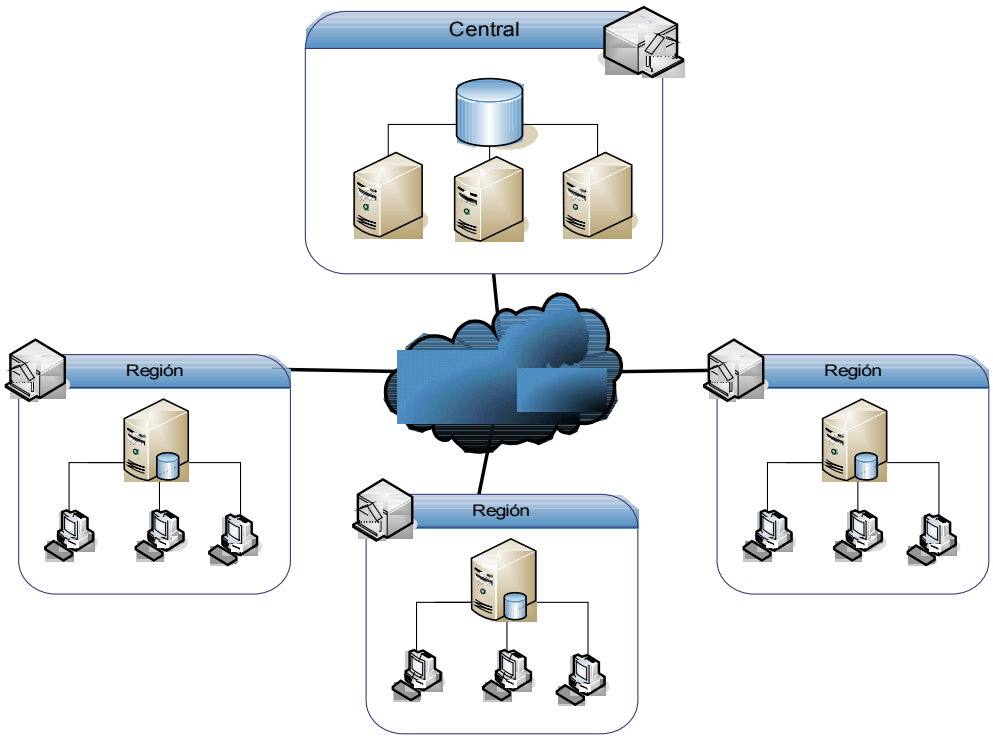


Fig. 1. Ejemplo de sistema distribuido

En el momento de seleccionar el mecanismo de réplica, es muy importante tener en cuenta varios requisitos no funcionales del sistema como son: concurrencia, volumen de información, disponibilidad de la información, gestor de base de datos, conectividad y rendimiento. Una de las principales características que debe poseer un mecanismo de réplica es que debe permitir la mayor independencia posible del negocio del sistema en el cual se utilice, la réplica debe poder ser vista como un componente independiente dentro del sistema.

En la actualidad una de las variantes de almacenamiento más utilizada son los sistemas de base de datos relacionales. Ese tipo de almacenamiento de datos, permite mantener grandes volúmenes de información y acceder a la misma mediante el estándar SQL. La mayoría de los productos de réplicas están orientados a las base de datos relacionales, y casi en su totalidad son software propietarios con un alto costo, además de estar implementados para gestores de base de datos específicos.

Mecanismo de réplica

El mecanismo de réplica descrito en este trabajo está creado para gestores de base de datos relacionales, y está diseñado con el objetivo de ser un mecanismo simple y al mismo tiempo óptimo para la sincronización de datos.

En todos los mecanismos de réplica se pueden observar tres procesos fundamentales:

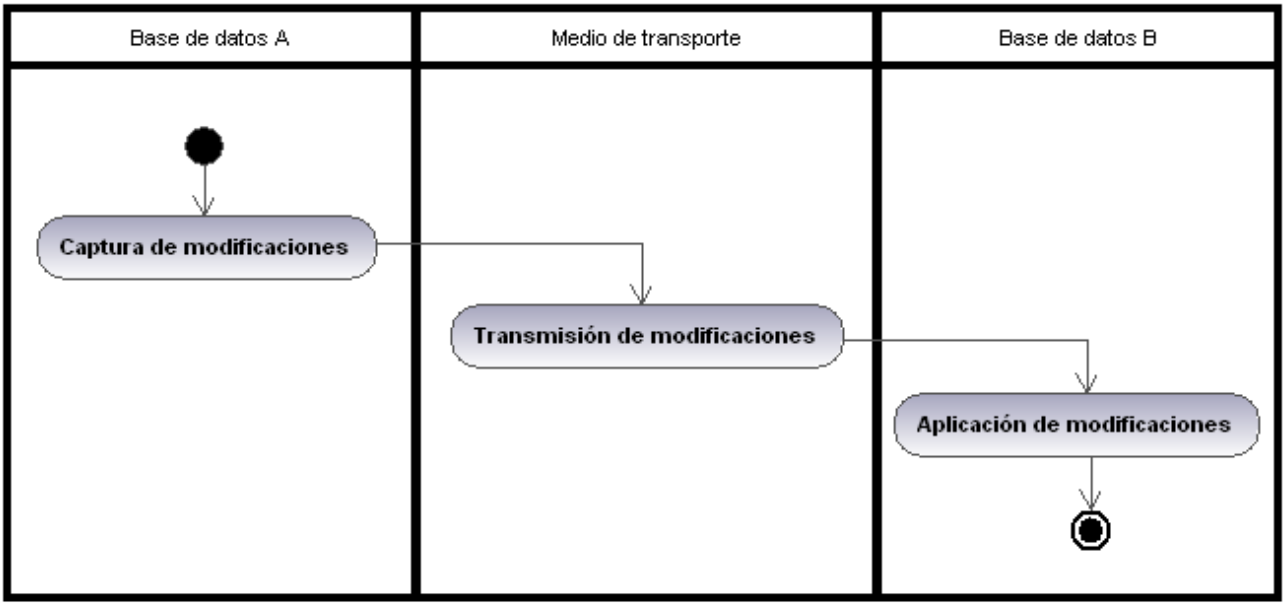
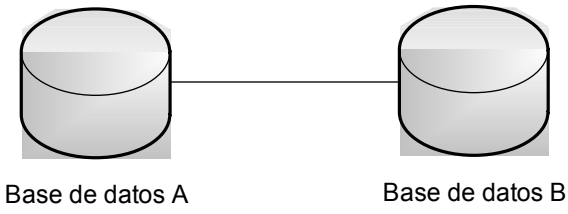


Fig. 2. Procesos de un mecanismo de réplica

Existen dos vertientes fundamentales del proceso de captura de modificaciones en los sistemas de replicación, una es la captura de las operaciones realizadas en la base de datos (utilizada en las réplicas basada en logs), y otra es mantener el control de sobre que datos se han realizado modificaciones.

En el siguiente ejemplo se muestra la diferencia en el proceso de captura de modificaciones entre estos dos tipos de réplica:



En ambas base de datos existe una tabla Persona (id, nombre, fecha_nacimiento).

Sobre la base de datos A, se realizan las siguientes operaciones:

Insert into Persona (id, nombre, fecha_nacimiento) values (1, 'Juan', '10/04/1981');

Update Persona set nombre='Juana' where id=1;

Update Persona set fecha_nacimiento='01/10/1980' where id=1;

En el mecanismo de captura de las réplicas basadas en logs, se almacenarían estas mismas tres operaciones, para luego ser transportadas y aplicadas en este mismo orden que fueron capturadas. A diferencia de esto en el mecanismo de control de cambio de datos, solo se llevaría el control de que la tupla con id=1 de la tabla Persona ha sido modificada, en dependencia de algunas implementaciones puede también o no llevarse el control de qué columnas fueron las modificadas para una tupla.

Las ventajas fundamentales del mecanismo de captura de modificaciones mediante el control de cambios son:

Transmisión de la menor cantidad de datos posibles para sincronizar las bases de datos.

Independiente del gestor de base de datos que se utilice, a diferencia del basado en logs en el cual el SQL capturado puede ser específico de un gestor determinado no siendo un SQL válido para otro.

El mecanismo de réplica propuesto en este trabajo, es basado en control de cambios y utiliza el siguiente esquema para su funcionamiento:

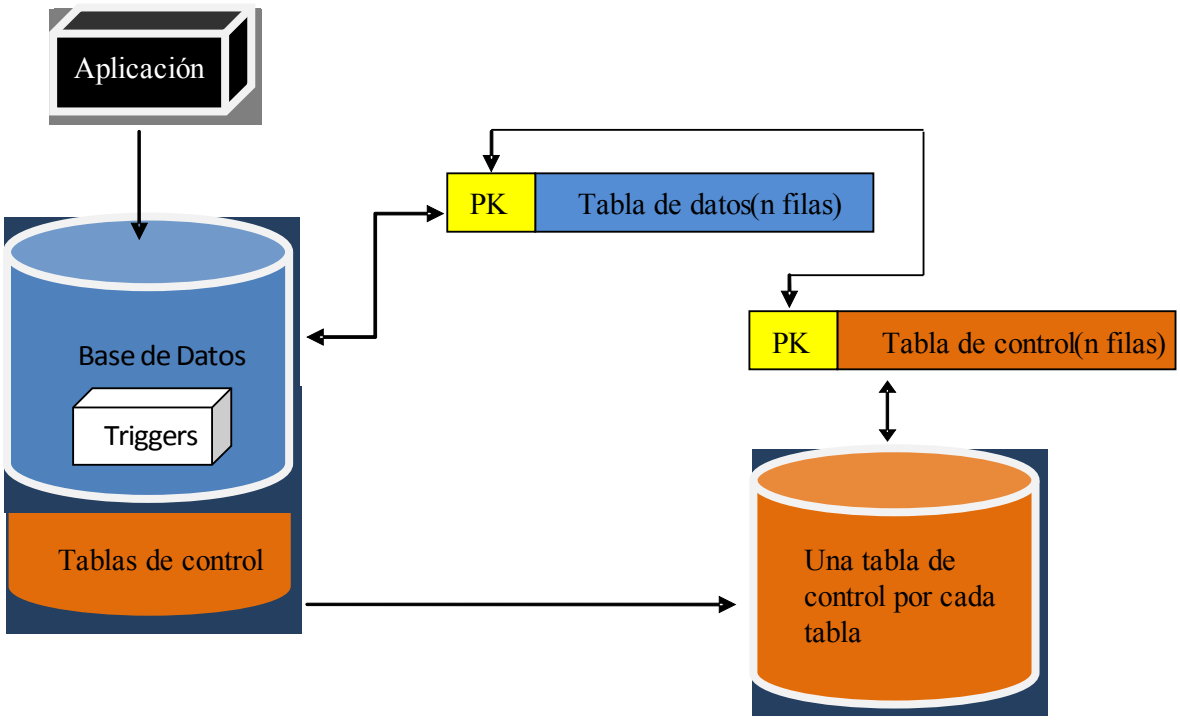


Fig. 3. Esquema de funcionamiento del control de modificaciones

Por cada tabla de la base de datos que se desea replicar, se tiene una tabla de control que contiene la llave primaria de la tabla correspondiente y tres columnas que se utilizan para el control de cambios.

Tabla 1. Estructura de las tablas de control

Columna	Descripción
…	Columnas de la llave primaria de la tabla correspondiente.
Fecha	Fecha en la que fue descubierta este cambio
Cod	0 si se realizó una inserción o una actualización y 1 si se elimina la tupla
Fuente	Un identificador de la base de datos donde se realizó la operación.

Para mantener actualizadas estas tablas de control se utilizan *triggers* sobre las tablas de la base de datos, que en el instante que se inserta, actualiza o elimina una tupla se escribe la tupla de control correspondiente. Es importante tener en cuenta que todas estas tablas de control y *triggers* son generados automáticamente por una herramienta que utiliza el esquema de datos de las tablas que participan en la réplica.

El proceso de captura de modificaciones, se divide en dos subprocesos en este mecanismo y se realizan de forma independiente, el subproceso de control de modificaciones y el subproceso de descubrimiento de estas modificaciones. En el momento que se dispara un *trigger* de control, se actualiza la tupla correspondiente en la tabla de control, pero el campo fecha se le asigna el valor nulo, esto quiere decir que el cambio ha sido registrado, pero que aun no ha sido descubierto. El subproceso de descubrimiento corre en otro proceso independiente y tiene en cuenta la dependencia entre las tablas en el momento de descubrir estos cambios registrados.

En el mecanismo propuesto, el proceso de transporte y aplicación de las operaciones se realiza en un solo proceso con el objetivo de simplificar el componente de réplica, para el perfecto funcionamiento de esta variante se requiere de una conectividad relativamente estable, si la red utilizada no cumple con estos requerimientos, se puede implementar un mecanismo de transporte alternativo sin afectar esto la solución de réplica.

El control de los datos a sincronizar, se realiza mediante la fecha de sincronización que se tenga con respecto a la base de datos a la cual se va a actualizar. Al momento de extraer estos datos se obtienen todos aquellos que hayan sido descubiertos posteriormente a la última fecha de sincronización con el servidor al cual se van a transmitir. Este proceso se realiza en los dos sentidos y paralelamente, teniéndose de esta forma una réplica bidireccional.

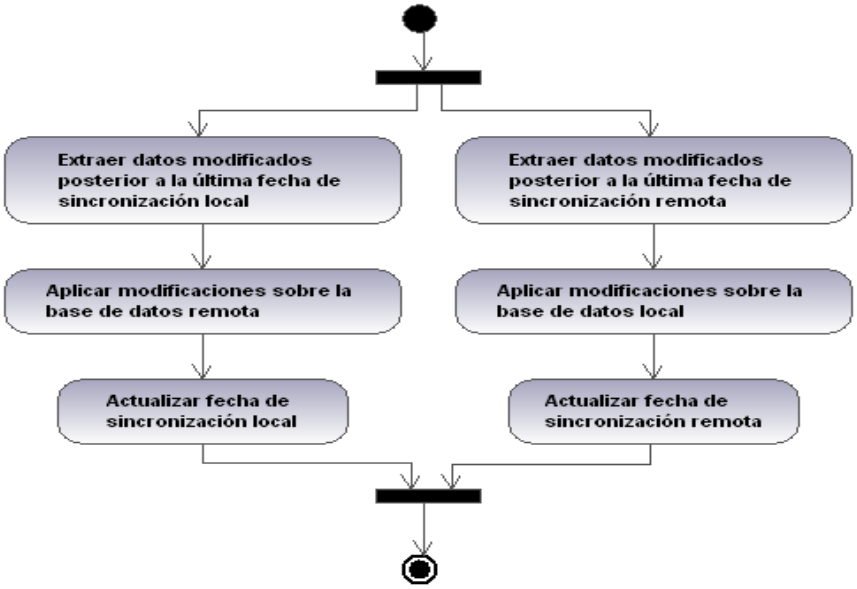


Fig. 4. Proceso de sincronización de las bases de datos

Para el control de las fechas de sincronización con los servidores se utiliza una tabla que posee la siguiente estructura:

Tabla 2. Estructura de la tabla de control de la fecha de sincronización

Columna	Descripción
Server	Nombre que identifica el servidor con el cual se sincroniza
Fecha_Sinc _Local	Última fecha en la que se sincronizaron los datos del servidor local al servidor remote
Fecha_Sinc_Remoto	Última fecha en la que se sincronizaron los datos del servidor remoto al servidor local
Estado	Campo que se puede utilizar para detener o habilitar la sincronización.

El proceso de aplicación de las modificaciones se realiza mediante llamadas a procedimientos almacenados que reciben como parámetro las columnas de las tablas a replicar, además del código de la modificación y el identificador del servidor donde se originó la modificación, y según el estado actual de esa tupla en la base de datos destino y el código de la modificación se realiza la operación necesaria para igualar la tupla a los datos enviados.

Una de las posibilidades que brinda este mecanismo de réplica es que permite definir reglas para el filtrado de las tuplas a replicar, se pueden definir expresiones booleanas SQL para filtrar las tuplas que pueden ser replicadas.

Esta solución permite implementar diferentes arquitecturas de base de datos distribuidas, tanto arquitectura con servidores que centralizan toda la información, o redes de replicación con servidores que según reglas se mantienen actualizados entre sí.

Conclusiones

La implementación de esta solución de réplica, disminuye la complejidad de un sistema distribuido, abstrayendo a los desarrolladores de la arquitectura de base de datos utilizada, además que posibilita la configuración de distintas arquitecturas de

base de datos distribuidas sin tener que realizar cambio en el modelo de negocio. El producto DBSynchronize implementa este mecanismo de sincronización y permite diferentes configuraciones según el objetivo que se quiera cumplir.

Bibliografía Consultada

Doug Tidwell. XSLT. Canada. O'REILLY. 2001. 395.

Progress Software Corporation. Progress DataExted Remote Edition Release Notes. 2006.

http://www.progress.com/realtime/techsupport/documentation