

# REPLICACIÓN DE DATOS EN AMBIENTES MÓVILES

Marcelo Zanconi      Jorge Ardenghi  
Grupo de Investigación en Sistemas Distribuidos  
Departamento de Ciencias de la Computación  
Universidad Nacional del Sur  
Av. Alem 1253 - 8000 - Bahía Blanca - ARGENTINA  
{cczanc,jra}@criba.edu.ar

## Resumen

La tecnología de comunicaciones ha permitido la concepción de procesos móviles, es decir, de procesos que se conectan a una red de acuerdo a la disponibilidad de transmisión de datos, su costo, la velocidad de transmisión, entre otras restricciones.

Los sistemas móviles son una generalización de los sistemas estáticos y por lo tanto incorporan características que le son propias. Sin embargo también surge una revisión de los algoritmos tradicionales de manejo de datos para adaptarse a esta nueva tecnología. En buena parte de las aplicaciones de sistemas de información, la movilidad no introduce complicaciones, sobre todo en aquellas arquitecturas tipo cliente/servidor, donde el cliente no está permanentemente ligado a un sistema. En aquellas aplicaciones donde el universo de clientes que acceden a un sistema está restringido, por cuestiones de seguridad, servicio, confiabilidad, estrategia, etc, la movilidad introduce cierto overhead en el sistema.

En este trabajo veremos brevemente algunos algoritmos de replicación para bases de datos estables y los extenderemos para soportar movilidad.

## 1 Introducción

La facilidad de replicar datos en un ambiente móvil será fundamental. Pero analicemos previamente cuales son las mejoras que introduce en los sistemas distribuidos típicos, es decir sin unidades móviles.

La copia de datos en bases distribuidas se utiliza como medio para aumentar la disponibilidad y confiabilidad de los datos, debido a que mejora la tolerancia ante fallas del sistema. Además, la replicación de datos reduce el costo de comunicación de red, ya que podemos acceder a la copia que se encuentre más cercana al lugar donde se encuentra el usuario que desee manipular el dato.

La desventaja más aparente de la replicación de datos es la sobrecarga que soporta el sistema para mantener una consistencia entre los datos. El sistema debe asegurar fundamentalmente que múltiples copias de un mismo ítem se comporten como una copia

simple. Esto se asegura a través de un protocolo de control de réplicas que se encarga de ocultar el hecho de que el dato está replicado, además da un conjunto de reglas para regular las lecturas y escrituras sobre las copias de los datos, determina el valor actual del dato y hace la replicación de datos transparente al usuario, [5].

En este trabajo usaremos un concepto estricto de replicación, referido exclusivamente a que cada localidad *conoce* la existencia de un número de réplicas y los algoritmos de control de concurrencia deben lograr un consenso de acceso al dato. En este orden de ideas, descartamos aquellos protocolos donde existe otra copia del dato, - manejada internamente por el administrador de dicho dato-, pues en este caso, el resto de las localidades no conocen a existencia de esta copia, que se provee básicamente con fines de respaldo y tolerancia a la falla del administrador central. Explícitamente, no consideraremos como *sistemas replicados* a aquellos protocolos como copia primaria y nodo central, [?].

El problema de la replicación ha sido extensamente estudiado, buscando fundamentalmente que el mecanismo de control de réplicas no introdujera sobrecarga al sistema, pero aumentara considerablemente la confiabilidad. Las redes más rápidas y confiables han hecho que el manejo de réplicas pueda considerarse viable y que se alcanzaran los objetivos de transparencia.

Los mecanismos tradicionales de replicación pueden ser clasificados como:

- **estáticos:** es decir donde el conjunto de réplicas es fijo y estable y tiene mucho que ver con una decisión de diseño de la base de datos, [4], [6], [7], [15].
- **dinámicos:** es decir donde el conjunto de réplicas es inicialmente fijo pero a medida que el sistema es usado, a través de un análisis de las operaciones realizadas, el conjunto varía ampliándose o contrayéndose, [22].

En un ambiente móvil ambos esquemas son adaptables, bajo ciertas restricciones. Veremos brevemente algunos conceptos de replicación y movilidad, antes de enunciar los protocolos.

## 1.1 Protocolos de replicación

Los protocolos de replicación de datos se basan en la idea de repetir la copia de un dato  $x$  en distintos nodos y garantizar que las operaciones de lectura y escritura sean servidas de modo de lograr consistencia y alta disponibilidad. Esta última característica busca que la lectura sea una operación que nunca se niegue y que sea atendida a un bajo costo. La consistencia busca que todas las copias tengan, en todo momento, el mismo valor.

La concurrencia es un aspecto fundamental de estos protocolos, estableciendo en general que las operaciones conflictivas no pueden ser atendidas simultáneamente, [8], [9].

Las variantes de los protocolos de replicación se basan en mecanismos de quorum impuestos para cada una de las operaciones:

- Quorum estricto: dadas  $n$  réplicas de un dato, se necesitará acceder a  $k$  copias para una operación de lectura y a  $l$  copias para una operación de escritura, con la restricción  $k + l > n$ , de modo de garantizar consistencia. Para un amplio desarrollo de este tema ver [5], [4], [18].

- Quorum variable: dadas  $n$  réplicas de un dato, se necesita definir un quorum de acceso que depende de la disponibilidad en el sistema, la partición de una red, la posibilidad de comunicación rápida. Estos protocolos buscan eficiencia de acceso, atacando el problema de comunicación en la red, [8], [1], [12].
- Quorum dinámico: originalmente se plantean  $n$  réplicas para un dato, que irán borrándose o agregándose al sistema, de acuerdo al número de operaciones y al origen de las mismas; se busca de este modo, corregir el diseño original de las réplicas respetando la evolución del sistema, manteniendo en todo momento el número de réplicas que mejor se adapta de acuerdo a métricas de performance, [9], [21], [22].

## 1.2 Movilidad

El concepto de movilidad ha surgido como extensión natural de usuarios que trabajan fundamentalmente en nodos operando sobre computadoras portátiles, conectándose con los sistemas cuando la oportunidad lo favorezca, [14].

¿Cuál es el desafío que surge ante un cambio tan simple como introducir la operación sobre computadores portátiles? Fundamentalmente la interacción. En sistemas fijos, un nodo está disponible o no al sistema. Si está disponible es confiable y el diseño de mecanismos de control aseguran consistencia. Si no está disponible, los nodos activos le aseguran una correcta recuperación al momento de despertar, fundamentalmente a través de bitácoras que se mantienen a tal propósito. En sistemas móviles se agrega un tercer estado que podríamos llamar de *baja conectividad*. Es decir, el sistema esta conectado pero la comunicación es débil, -posiblemente a través de un ancho de banda inapropiado para el tipo de información que se desea transmitir-, llega con demasiado retardo, -por ejemplo, en la transmisión de un video, la voz y la imagen llegan a destiempo-, o es escatimada por el usuario móvil por razones de costo y servicio <sup>1</sup>.

Bajo esas condiciones el modo de operación puede ser caracterizado como *conectado* o *desconectado*. Para operar en modo desconectado, el cliente realiza sus operaciones de lectura y escritura sobre una memoria caché local, (algunos autores también llaman a este situación *modo emulado*). Para proveer transparencia a los datos, el sistema debe propagar las modificaciones cuando el cliente opera en modo conectado. En este caso, pueden ocurrir conflictos. Para operar en modo conectado, el cliente lee los datos del servidor, pero escribe en su caché. Dado que el sistema debe estar preparado para la desconectividad, este estado también se llama *expectante*, porque el sistema debe prever mantenimiento de la caché para uso futuro. La idea general puede resumirse en la figura, 1. Una profundización de este tema puede encontrarse en [13], [14], y [3].

## 2 Administración de Réplicas en Ambientes Móviles

¿En que afecta la replicación de datos a los ambientes móviles? ¿Cómo se administran los datos replicados en un ambiente móvil? Para responder estas cuestiones, se debe tener en cuenta que tanto la disponibilidad como la confiabilidad de datos en un sistema donde se interactúa con unidades móviles es fundamental.

---

<sup>1</sup>Imaginemos, por ejemplo, un usuario accediendo mediante su laptop y telefonía satelital a un sistema

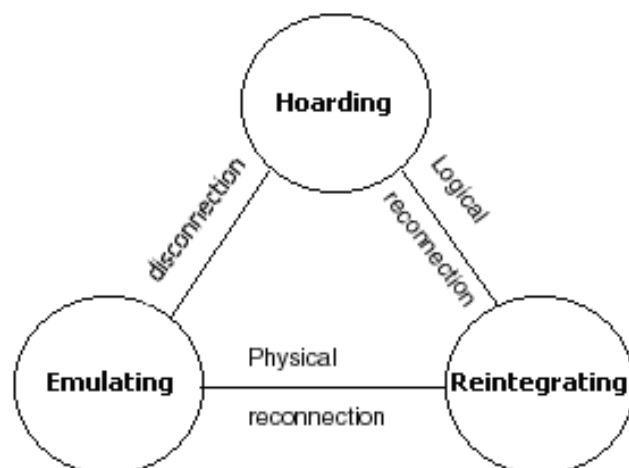


Figura 1: Arquitectura en Sistemas Móviles

Un ambiente móvil es más susceptible a fallas que un ambiente fijo; supongamos que un dato  $x$  está almacenado en una unidad móvil. Debido a las frecuentes desconexiones que ésta puede sufrir<sup>2</sup> el acceso al dato puede ser impedido hasta que la unidad sea reconectada, retardando otras aplicaciones o incluso abortándolas por razones de timeout.

Es claro que tenemos dos opciones: no almacenar datos en unidades móviles (algo que restringe en gran forma al sistema) o almacenamos réplicas de ese dato en otras unidades, sean móviles o fijas. De esta manera siempre tendremos alguna de las copias disponibles para el uso. Puede verse que esto incrementa además en forma directa la accesibilidad del sistema, ya que mantiene al sistema operacional en caso de fallas. En este caso, los sistemas de control de réplicas de quorum variable o dinámico son los que mejor se adaptan.

El otro punto que debemos tener en cuenta es el costo de la comunicación, no solo por el hecho de que, al replicar el dato, podemos acceder a la copia que encontremos más cercana, sino también que debemos recordar que el costo de la comunicación inalámbrica es mayor al de la comunicación por cable. Por lo tanto en el momento de acceder a una copia del dato debe tenerse en cuenta no sólo la proximidad sino el medio por el cual se produce la comunicación.

Además el costo de comunicación entre dos unidades móviles no solo depende de la distancia entre ellas sino del costo incurrido para determinar la locación exacta de ambas.

El costo de la comunicación es asimétrico. El costo de ubicar un dato replicado entonces, también lo será. Si una unidad fija desea acceder a un dato replicado en una unidad móvil, incurrirá no solo en el costo de la comunicación sino también en el costo de búsqueda, para poder localizar esa unidad. En el caso inverso, esto es, que la unidad móvil intente acceder a un ítem de datos que se encuentra replicado en la unidad fija,

<sup>2</sup>Por ejemplo por problemas de comunicación por su conexión inalámbrica o al recargo de una batería

el costo de ubicar la unidad fija es obviamente nulo: la locación de una unidad fija es estática.

La decisión de donde ubicar una copia del dato dependerá fuertemente del volumen de transacciones que se generaran en ese sitio para manipular el dato. Si un gran porcentaje de esas transacciones es generado sobre la unidad móvil, es lógico que prefiramos almacenar la copia del dato en esta unidad, ya que gran cantidad de las transacciones será satisfecha en forma inmediata, mientras que el menor porcentaje incurrirá en el costo adicional de localizar la unidad móvil. Adicionalmente, por seguridad, se pueden realizar frecuentes respaldos de la copia del dato en la estación soporte “Home” de la unidad móvil, previendo las frecuentes desconexiones o caídas accidentales de la unidad móvil, evitando así la pérdida del dato. Así, cuando la copia residente en la unidad móvil es inalcanzable, la estación soporte “Home” la reemplaza. En el momento en que se recupera la unidad móvil, se efectúan las correspondientes actualizaciones y el sistema sigue trabajando normalmente.

Una opción viable utilizada en los esquemas administración de datos replicado típicos adaptable a transacciones móviles es mantener una copia “cache” del dato. Es decir, si un cliente hace uso continuo de un ítem dado, este cliente guarda en su memoria local una copia del dato, que puede acceder mientras no le sea enviado, por el servidor de la copia, un mensaje, advirtiéndole que esa copia “cache” está desactualizada. Esta técnica es muy útil en caso de que se tenga una actividad mayor de lecturas que de escrituras. Y hace un importante ahorro tanto en el costo de la comunicación como en el costo de búsqueda.

En resumen todas aquellas suposiciones implícitas que eran válidas en ambientes distribuidos tradicionales ya no tienen validez en ambientes móviles:

- El costo de comunicación ya no es simétrico.
- Existen restricciones de poder y de almacenamiento en las unidades móviles.
- Existen nuevas fallas, a parte de las típicas de comunicación: ahora tendremos frecuentes conexiones y desconexiones.
- La locación de una unidad que contiene un dato no es estáticamente conocida.

## 2.1 Bases para un algoritmo de replicación y movilidad

A continuación, se detallan las bases de diseño para seleccionar un algoritmo de replicación de datos en un ambiente móvil.

- Tipo de Copia: encontramos dos tipos de copia,
  - Copias actualizables: son aquellas que pueden ser actualizadas por el usuario, y que residen en unidades centrales.
  - Copias de lectura: son copias no modificables en forma local (solo accesibles para lectura) y de rápido acceso.

La importancia de esta distinción es útil para la performance del sistema. Es decir, si se desea hacer una modificación a un dato específico no tiene sentido pedir

autorización a todos los sitios donde ese dato está replicado, en lugar de eso solo pedimos autorización a un subconjunto de esas copias (el subconjunto de las copias centrales) y luego de hecha la modificación a este subconjunto, ésta es propagada al resto de los sitios que contiene las copias del dato. Esta actualización a las copias de lectura puede ser realizada en forma asincrónica ya que se supone que ellas nunca son usadas para generar nuevas actualizaciones y estará reservada a un superusuario (tal como el administrador de la base de datos) o al sistema como una transacción privilegiada o de alta prioridad, donde en tal caso será manejada por administrador de transacciones y sometida a consideración del administrador de datos, quienes determinaran la forma y el momento más oportuno para ejecutar dicha transacción.

- Reconfiguración del conjunto de réplicas: ya se ha establecido que aquellos algoritmos de replicación con quorum variable o dinámico son que mejor se adaptan a un ambiente móvil, ya que las frecuentes fallas pueden crear la necesidad de migrar una copia de un lugar a otro o de agregar una nueva copia en otro sitio.

Hay varias razones para querer mover el sitio central, y si algunos de los sitios se encuentran activos pueden participar en el movimiento, facilitando la migración. Podemos desear cambiar el número de copias o agregar sitios o eliminar sitios donde éstas se encuentran. Aumentar el número de copias incrementa la disponibilidad del sistema, pero se introduce sobrecarga en el costo de actualizar dichas copias, obviamente eliminar copias produce el efecto contrario.

Se debe observar además, que si las desconexiones son frecuentes debe seleccionarse una estrategia dinámica que administre el directorio que permite las reconfiguraciones de los sitios que mantienen las copias.

Por otro lado, una copia central no debería ser ubicada en una unidad móvil que tenga una comunicación con bajo ancho de banda, a menos, claro, que en ese sitio se acceda frecuentemente a la copia o se deba tener acceso a ella en todo momento, es decir, aun cuando la unidad se encuentre desconectada.

Por otro lado, debe cuidarse que dicha unidad no sufra frecuentes desconexiones ya que esto volverá al sistema menos confiable que si se colocan sobre las unidades fijas, con conexión de red cableada.

Una técnica interesante es colocar en las unidades móviles las copias de “solo lectura” y en las unidades fijas las copias centrales, esto hace al sistema más seguro, confiable y accesible.

- Fallas del sistema: las fallas de los sistemas han sido ampliamente estudiadas y clasificadas, para una excelente recopilación ver [10].
  - Fallas de procesamiento y de memoria: como FAIL-STOP, (cuando ocurre una falla el procesador simplemente se detiene (HALT), la memoria principal se pierde, pero el almacenamiento estable continua intacto), bizantina, (cuando no se puede predecir la forma en que se actuará ante una falla).
  - Fallas de red: en general la tecnología nos permite asumir que la red es “segura”, es decir que los mensajes recibidos a través de ella son “creíbles” o bien que no llega. Otro aspect

ERROR: undefined  
OFFENDING COMMAND: J141

STACK:

-dictionary-