

**MANEJO DE ARCHIVOS REPLICADOS BAJO TRANSACCIONES DISTRIBUIDAS
EN EL SISTEMA DGDBM**

Orlando CASTRO
María Consuelo FRANKY

Depto. de Ingeniería de Sistemas y Computación
Universidad de los Andes
Apartado Aéreo 4976
Santafé de Bogotá - COLOMBIA
(e-mail: cfranky@uniandes.edu.co)

RESUMEN:

Este artículo describe el diseño y la implantación de los protocolos de replicación de archivos denominados Lectura y Actualización Ponderada (LAVP) y Copia Primaria Copias Secundarias (CPCS) para el sistema DGDBM. El sistema DGDBM es un API que ofrece funciones para el desarrollo de aplicaciones distribuidas en una red de computadores, con ejecución de conjuntos de operaciones en forma transaccional y manipulación de archivos independientemente de su ubicación por parte del programador de aplicaciones. En el artículo se analizan algunas alternativas para implantar protocolos de replicación de archivos, conservando la transparencia a la distribución en los servicios ofrecidos al programador. Además se describen las características y la arquitectura de la implantación final de estos protocolos en el sistema DGDBM.

PALABRAS CLAVE:

Sistemas Distribuidos, Sistemas de Archivos Distribuidos, Sistemas de Transacciones Distribuidas, Protocolos de Replicación de Datos.

1. INTRODUCCIÓN

El uso extendido y cada vez más frecuente de redes de computadores genera la necesidad de herramientas que faciliten la programación de aplicaciones y que exploten las ventajas de contar con una mayor cantidad de dispositivos de almacenamiento y un mayor número de procesadores. Estas herramientas deben buscar aumentar la probabilidad de trabajar con datos locales en lugar de remotos, aumentar la disponibilidad de los datos y asegurar la recuperación en caso de fallas mediante copias de respaldo. Además, se requiere que estas herramientas ofrezcan a los usuarios seguridad respecto a la consistencia y corrección de los datos manejados.

DGDBM es una librería de funciones C que el programador puede incluir en sus programas e invocarlas de manera que la distribución de sus archivos en la red sea transparente para él, y en consecuencia le permita desarrollar su aplicación como si estuviera trabajando con datos centralizados <Franky 95>. DGDBM también ofrece el manejo de transacciones distribuidas asegurando su atomicidad y el control de concurrencia. Adicionalmente, DGDBM busca brindar una mayor disponibilidad de los datos mediante el manejo de archivos replicados.

En este artículo se describe el sistema DGDBM y los protocolos de manejo de replicación de archivos que se implantaron en dicho sistema.

2. DESCRIPCIÓN DEL SISTEMA DGDBM

DGDBM utiliza localmente en cada sitio los servicios del manejador de archivos GDBM <Nelson,93>. Este manejador permite la manipulación de archivos mediante acceso directo utilizando técnicas de hashing extendido. Cada registro de un archivo está formado por dos cadenas de caracteres de longitud variable: una llave y un contenido.

DGDBM ofrece los siguientes servicios <Franky, 95>:

- Comienzo y terminación de transacciones distribuidas : el programador dispone de dos funciones para **marcar** el comienzo y terminación de sus transacciones. Todas las operaciones incluidas entre estas marcas estarán sometidas al control transaccional y por lo tanto, contarán con las propiedades de atomicidad, durabilidad, seriabilidad y aislamiento <Ceri 84> .
- Apertura, Cierre y Eliminación de Archivos: son funciones que permiten la manipulación de archivos completos. De estas funciones solo la de Apertura da la **opción** de especificar el sitio en que se desea crear el archivo, cuando se abre un archivo nuevo; las demás funciones utilizan un catálogo de archivos del sistema para establecer el sitio de ubicación de los mismos.
- Consulta, Actualización y Eliminación de Registros: son funciones que permiten el manejo directo de registros de los archivos. Mediante el manejo de candados jerárquicos a nivel de archivo y de registro <Gray 75>, se puede permitir el acceso de varias transacciones en forma concurrente sobre el mismo archivo y/o registro.
- Consulta Secuencial de Registros: son funciones que permiten la consulta secuencial de los registros de un archivo.

En la figura 1. se muestra la arquitectura de un sistema DGDBM, sus agentes y la relación entre ellos.

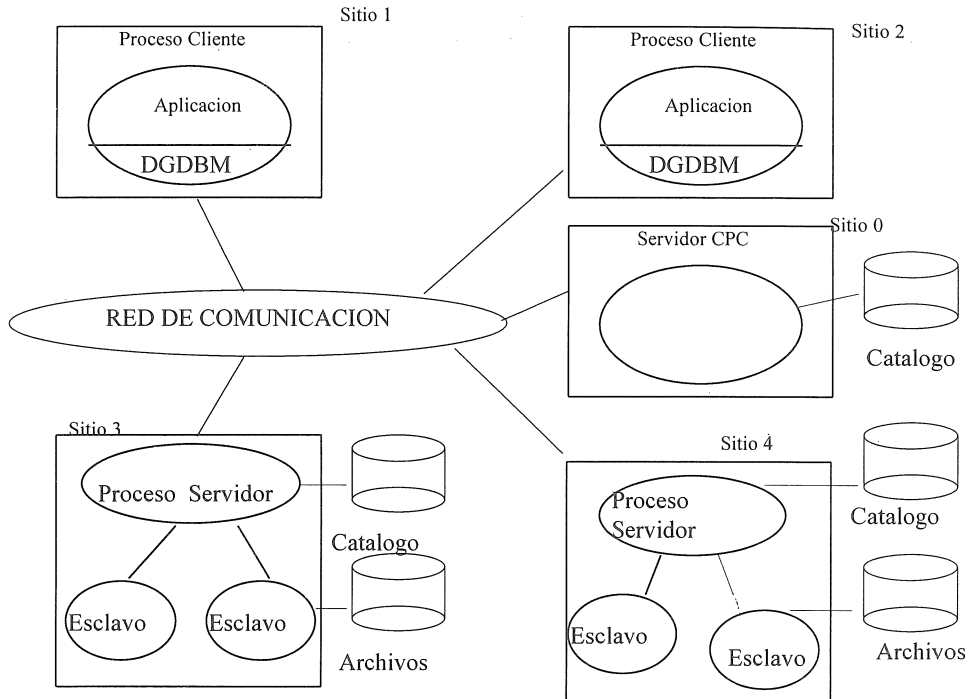


Figura 1. Arquitectura del sistema DGDBM

Una aplicación que utiliza DGDBM se convierte en un **Agente Cliente** en el sistema. Este agente cliente está constituido por el código propio de la aplicación más las funciones del API DGDBM incluido.

En cada sitio componente del sistema se crea, al arrancar el sistema, un **Agente Servidor** encargado de atender las solicitudes de los agentes clientes. Con el fin de mejorar el desempeño del sistema, los agentes servidores generan **Agentes Esclavos** que realizan físicamente las solicitudes de un cliente sobre los archivos, es decir, por cada cliente que esté atendiendo un agente servidor, existe un agente esclavo.

Se maneja un **Catálogo de Archivos**, el cual está replicado en todos los sitios mediante un esquema de copia primaria copias secundarias que es administrado por un **Agente Servidor de la Copia Primaria del Catálogo** <Rangel 94>.

3. PROTOCOLOS DE REPLICACIÓN DE DATOS

Bastantes son los protocolos de replicación que se encuentran descritos y analizados en la literatura. Sin embargo, la mayoría de ellos puede clasificarse en un de los siguientes grupos o familias <Abásolo 94>, <Ahamad 90> <Castro 95>:

- Protocolos de Consenso
- Protocolos de Copias Disponibles
- Protocolos de Copia Primaria-Copias Secundaria

Las características que permiten clasificar un protocolo en uno de estos grupos tienen que ver con la forma en que ofrecen consistencia mutua y corrección de los datos (seriabilidad de copia única, <Mullender 89>).

Los protocolos de consenso establecen un protocolo que exige la participación de varias réplicas del dato, entre ellas al menos una actualizada, para poder realizar una operación. Los de copias disponibles trabajan con las copias que estén activas, pero no siempre garantizan a los usuarios la consistencia de los datos. El protocolo de Copia Primaria-copias secundarias exige la actualización de una misma réplica y permite la consulta de todas; en este caso el usuario decide si la consulta debe hacerse sobre datos actualizados en la copia primaria, o sobre datos no necesariamente actualizados en una de las copias secundarias.

En el sistema DGDBM se implantaron dos protocolos de replicación de datos: Lectura y Actualización por Votación Ponderada y el de Copia Primaria-Copias Secundarias.

3.1 El Protocolo de Lectura y Actualización por Votación Ponderada (LAVP)

Este es un protocolo de consenso en el que a cada réplica del dato se le asigna un número de votos V_i , existiendo un total de votos T para el sistema, un quórum de lectura R , y un quórum de escritura W . Además debe cumplirse que: $W > T/2$ y $W + R > T$. Estas condiciones permiten garantizar que no hay dos operaciones simultáneas de escritura sobre un dato, y que una lectura y una modificación tampoco se pueden dar simultáneamente sobre el mismo dato. <Ceri 84>.

Cada vez que se quiera ejecutar una operación sobre un dato replicado debe lograrse la participación de tantas réplicas como se requieran para cumplir con las siguientes condiciones:

- si la operación es una consulta, la suma de los votos de las réplicas participantes debe ser mayor o igual al quórum de lectura.
- si la operación es de modificación, la suma de los votos de las réplicas participantes debe ser mayor o igual al quórum de escritura.

3.2 El Protocolo de Copia Primaria-Copias Secundarias (CPCS)

En este protocolo todas las operaciones de modificación se realizan siempre sobre la misma réplica: la copia primaria, y luego son transmitidas en forma diferida (asincrónica) a las demás réplicas: las copias secundarias. Las operaciones de consulta por otra parte, se pueden realizar sobre la copia primaria, o sobre una cualquiera de las copias secundarias las cuales pueden o no estar actualizadas.

Para garantizar la consistencia de los datos, aún en la ocurrencia de transacciones simultáneas sobre un mismo dato, se debe asegurar que las transacciones que accedan copias secundarias lo hagan en modo de consulta y que todas las transacciones de actualización utilicen la copia primaria. Se establece así sobre la copia primaria un control de concurrencia similar al que se realiza en datos con copia única.

4. CARACTERÍSTICAS DESEABLES DEL MANEJO DE REPLICACIÓN

- Además de ofrecer consistencia mutua entre las réplicas y corrección de los datos replicados, la replicación debe tener otras características que la hagan más práctica y que permitan aprovechar el máximo de sus ventajas. Entre otras se pueden mencionar las siguientes que fueron implantadas en el sistema DGDBM:
- **Transparencia al Usuario:** El usuario de los datos (i.e. el programador de aplicaciones en DGDBM), no tiene por qué entenderse con aspectos relacionados con la replicación. Es decir, en la aplicación no debe aparecer código para control y manipulación de la replicación que nada tiene que ver con su propósito original.
- **Coexistencia de Varios Protocolos:** Las ventajas de la replicación están ligadas en gran parte a la forma en que se usen los datos, siendo más apropiado un protocolo que otro según el caso. Como los datos involucrados en una aplicación pueden ser usados en distintas formas, es conveniente que se cuente con más de un protocolo de replicación y además que puedan manejarse indiscriminadamente datos no replicados, con datos replicados bajo uno u otro protocolo.
- **Administración Independiente de la Replicación:** La replicación de datos exige administración, es decir, requiere de mecanismos que permitan eliminar una réplica, aumentar el número de réplicas, cambiar el número de votos asignados, el quórum de lectura o escritura, cambiar el sitio de la copia primaria, etc. Estas labores no pueden dejarse a la aplicación que usa los datos, pero son indispensables para el uso práctico de los protocolos de replicación.

5. ALTERNATIVAS DE DISEÑO DE LOS PROTOCOLOS DE REPLICACIÓN

El control de la replicación de archivos puede implantarse de dos maneras: como parte del mecanismo del control transaccional, o como un nivel inferior, independiente del control transaccional sin perjuicio de las propiedades de las transacciones. A continuación se describen las dos alternativas, sus ventajas y desventajas:

5.1 Control de Replicación como Parte del Control Transaccional

En DGDBM el control transaccional lo efectúa el agente cliente, quien debe registrar los sitios participantes y los cambios de estado de la transacción que originó.

En la figura 2, se muestran dos archivos, el archivo A replicado en los sitios 1, 2 y 3, y el archivo B replicado en los sitios 3, 4 y 5. Además, se muestra un cliente C que ejecuta una transacción sobre estos dos archivos.

Como se puede apreciar, el cliente debe establecer comunicación con los servidores de todos los sitios donde haya copias de los archivos involucrados, y que por la lógica del protocolo, deban participar en la operación. Además, si el protocolo de replicación utilizado es LAVP, debe recibir los votos de cada sitio, establecer el momento en que se alcanza el quórum, etc.

Durante el transcurso de la transacción, las operaciones que involucren estos archivos se deben enviar a todos los sitios con réplicas, y al terminar, los cambios de estado (inicial, prevalidar, validar, o anular) debe hacerse en cada sitio de réplica.

Las ventajas de esta forma de controlar la replicación radican en que el mismo control transaccional se puede utilizar para garantizar la consistencia mutua y la corrección de las réplicas, puesto que si se anula la transacción, o se termina con éxito, todas las réplicas necesarias del archivo quedarán en el estado apropiado. Sin embargo, este esquema aumenta la probabilidad de fracaso de una transacción, puesto que exige la disponibilidad de la mayoría (quórum) de los sitios de réplica involucrados, hasta su terminación. Por ejemplo, si una transacción está en estado inicial y uno de los sitios deja de estar disponible, la transacción debe anularse.

De igual manera el tiempo de respuesta al usuario final se aumenta, puesto que las operaciones no pueden considerarse terminadas hasta completar el intercambio de mensajes con todos los servidores involucrados.

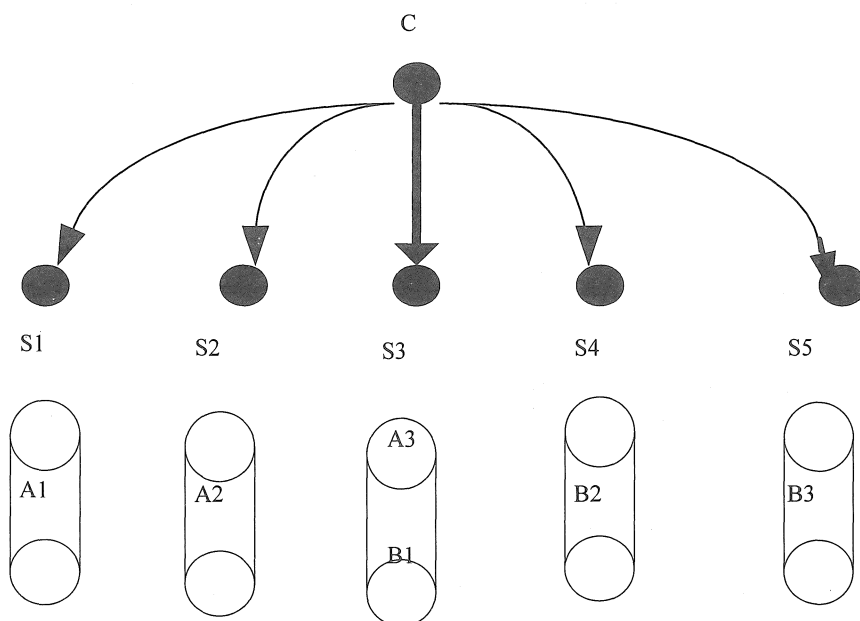


Figura 2. Control de la Replicación dentro del Control Transaccional

5.2 Control de Replicación Independiente del Control Transaccional

El ejemplo que se muestra en la figura 3 corresponde al mismo mostrado en la figura 2, pero como se puede apreciar, el agente cliente que controla la transacción, no se percata de la existencia de replicación de los archivos y tampoco considera la actualización de estas réplicas dentro de la lógica de la transacción. Este agente trata todos los archivos, replicados o no replicados, de la misma forma.

En cambio, aparece un nuevo rol para el agente servidor de un sitio: para cada archivo existe un agente servidor que coordina la actualización de sus réplicas (servidores sitio 1 y sitio 5, en el ejemplo). Este

agente coordinador es el que interactúa con el cliente para todo lo que tenga que ver con el archivo, y, una vez terminada la transacción, es el encargado de actualizar las réplicas y garantizar la consistencia mutua y la corrección de las copias (mediante la lógica del protocolo respectivo).

De esta forma no se requiere la disponibilidad de la mayoría de los sitios de réplicas hasta el final de la transacción sino solo la del coordinador de cada archivo, lo que aumenta la probabilidad de éxito de la transacción y permite, además, disminuir el tiempo de respuesta al usuario final, puesto que disminuye el número de mensajes intercambiados con el cliente.

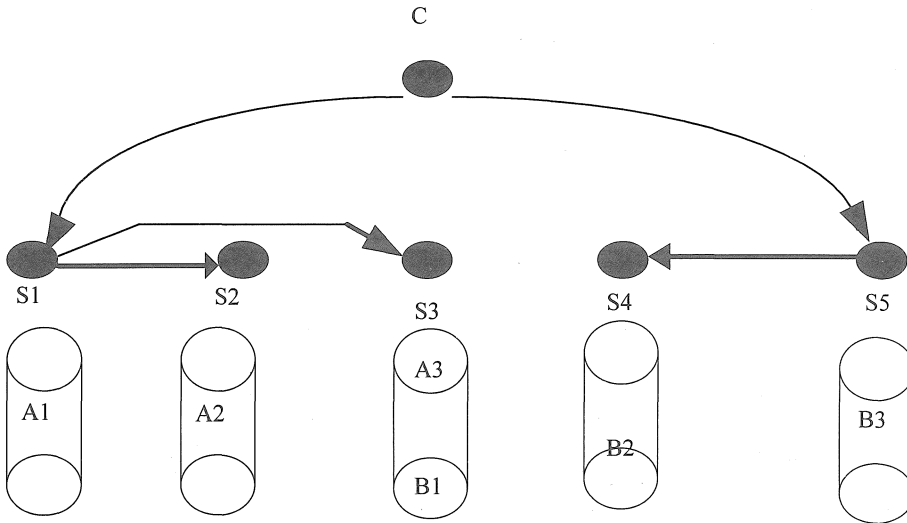


Figura 3: Control de la Replicación Independiente del Control Transaccional

5.3 Seriabilidad y Aislamiento

Un aspecto importante que se debe tener en cuenta al implementar un protocolo de replicación, es el de conservar la seriabilidad de las transacciones que se ejecutan concurrentemente <Coulouris 88>, ya que la existencia de varias copias del mismo dato podría facilitar que varias transacciones accedieran al dato simultáneamente en forma conflictiva.

Si bien esta es una propiedad garantizada por los mecanismos de control transaccional, el diseño de la implantación del protocolo de replicación, debe conservarla.

Para el caso específico en que el control transaccional es independiente del control de replicación, esto se logra mediante el uso de los candados apropiados en cada una de las réplicas. Se garantiza así que para un dato (registro) replicado nunca podrán existir simultáneamente dos coordinadores de operaciones conflictivas pertenecientes a transacciones diferentes.

5.4 Actualización de Réplicas en el Protocolo LAVP

En el protocolo LAVP, no siempre participan todas las réplicas de un dato en las operaciones que lo actualizan, si el quórum de escritura así lo permite. Esto implica que debe contarse con algún mecanismo que ponga al día esas copias que no participaron en un quórum.

En general existen dos aproximaciones para resolver este problema. En una de ellas el proceso encargado de modificar las réplicas, detecta la situación y la registra de manera que la actualización se intente en forma diferida el número de veces que sea necesario o que el sitio de la réplica al recuperarse (si estaba caído) se pueda poner al día haciendo la consulta respectiva.

En la segunda aproximación, la puesta al día se hace en el momento en que una réplica participa en un nuevo quórum, aprovechando el hecho que en dicho quórum siempre debe participar por lo menos una réplica actualizada. El agente que esté controlando el quórum (el coordinador), al detectar que hay réplicas desactualizadas, procede a ponerlas al día.

Para determinar si una réplica está o no actualizada, se maneja un número de versión para cada dato replicado que se va incrementado cada vez que es modificado. El dato más actualizado es el que tiene mayor número de versión.

6. IMPLANTACIÓN DE LOS PROTOCOLOS LAVP Y CPCS EN EL SISTEMA DGDBM

Se escogió el esquema de control de replicación independiente del control transaccional para implantar los protocolos LAVP y CPCS en el sistema DGDBM. A continuación se describe el diseño final implantado.

6.1 El Agente Administrador

Se creó un agente cliente especial con funciones de administración en el sistema DGDBM que entre otras cosas se encarga de la administración de la replicación de archivos, liberando así a los agentes clientes de cualquier función que se relacione con la replicación de archivos.

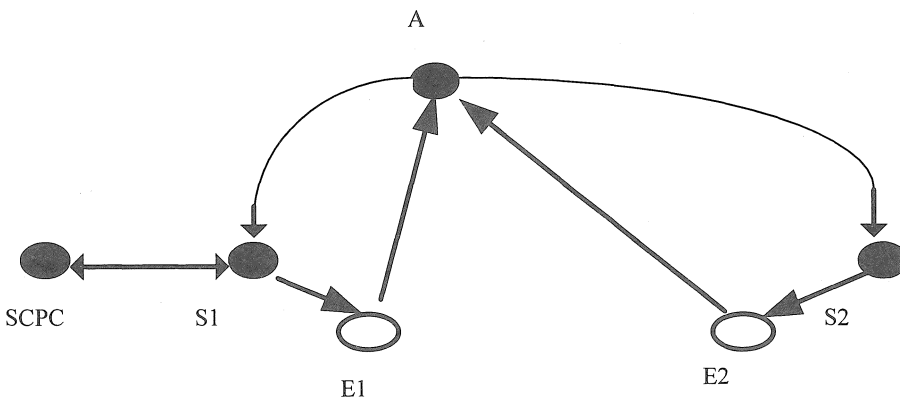


Figura 4: Reserva de Datos y Generación de Esclavos para la Replicación de un Archivo

6.2 Registro del Protocolo LAVP o CPCS

El registro de un protocolo de replicación para un archivo, lo hace el Agente Administrador del sistema, mediante una transacción especial que puede ejecutarse simultáneamente con las transacciones de los usuarios. El proceso consiste de dos etapas: en la primera el Agente Administrador transmite la información relacionada con el protocolo al servidor del sitio actual del archivo. Además se asignan candados sobre la copia primaria del catálogo y sobre el archivo a replicar, y se generan los esclavos que atenderán la transacción. Finalmente el Agente Administrador recibe los mensajes de los esclavos indicándole que en todos los sitios se puede iniciar el registro de la replicación. En la figura 4, se ilustra el intercambio de mensajes que se lleva a cabo en esta etapa, suponiendo que se está replicando un archivo cuyo sitio original es el 1 y que se quiere una réplica adicional en el sitio 2.

En la segunda etapa, el Agente Administrador ordena al servidor del sitio actual del archivo que inicie la replicación del archivo enviando su contenido a los demás sitios, orden que es retransmitida al esclavo respectivo. El esclavo a su vez inicia el envío efectivo de los registros del archivo a los servidores de los sitios en donde haya réplicas. En cada uno de estos sitios se retransmite el registro al esclavo respectivo.

Una vez terminada la retransmisión del contenido del archivo, los servidores involucrados informan el hecho al Agente Administrador, quien continúa con las fases de prevalidación y validación de la transacción.

6.3 Acceso de un Archivo Replicado con LAVP

El uso de un archivo replicado con LAVP dentro de una transacción, tiene tres etapas: iniciación de la operación, consecución de quórum y terminación de la operación.

En la *etapa de iniciación* de la operación el agente cliente ejecuta la lógica propia de cada aplicación y envía la solicitud de ejecución de la operación al servidor coordinador de operaciones para el archivo. Si la operación es **apertura** del archivo, en esta etapa el agente cliente establece el sitio que tendrá el rol de coordinador, es decir, aquel sitio que primero responda a la solicitud de apertura del archivo, siendo éste el único servidor con el que se comunicará para los operaciones con el archivo a manipular.

En la *etapa de consecución de quórum* el servidor coordinador solicita a los sitios en que hay réplicas del archivo, el voto que indica que han colocado el candado apropiado sobre la réplica respectiva. En cada sitio de réplica, la asignación del candado se hace en la misma forma en que se haría la asignación de un candado para un archivo no replicado.

Una vez alcanzado el quórum, se realiza la *etapa de terminación* de la operación en la cual la solicitud de operación se retransmite al esclavo respectivo en el sitio del servidor coordinador, como si se tratara de un archivo no replicado. El esclavo responde directamente al cliente.

Estas tres etapas se repiten para cada operación que se ejecute sobre el archivo. Algunos aspectos importantes para resaltar en este proceso son los siguientes:

- Si no responde un sitio en el que hay una réplica de un archivo que se quiere abrir, el agente cliente intenta con otro en forma recurrente hasta agotar la lista de sitios con réplicas.
- Cada vez que se recibe un voto se registra en una **lista de quórums pendientes** y se verifica si se alcanzó o no el quórum.

- Cuando existe un candado incompatible con el solicitado en un sitio determinado, la transacción se pone en lista de espera en ese sitio. Esto puede implicar que toda la transacción quede en espera si no se alcanza el quórum. El tratamiento de transacciones en espera se hace en forma local como si se tratara de un archivo no replicado.

Cuando termina la transacción dentro de la cual se están ejecutando estas operaciones, el servidor coordinador retransmite a todos los servidores de réplicas que participaron en el quórum las modificaciones registradas.

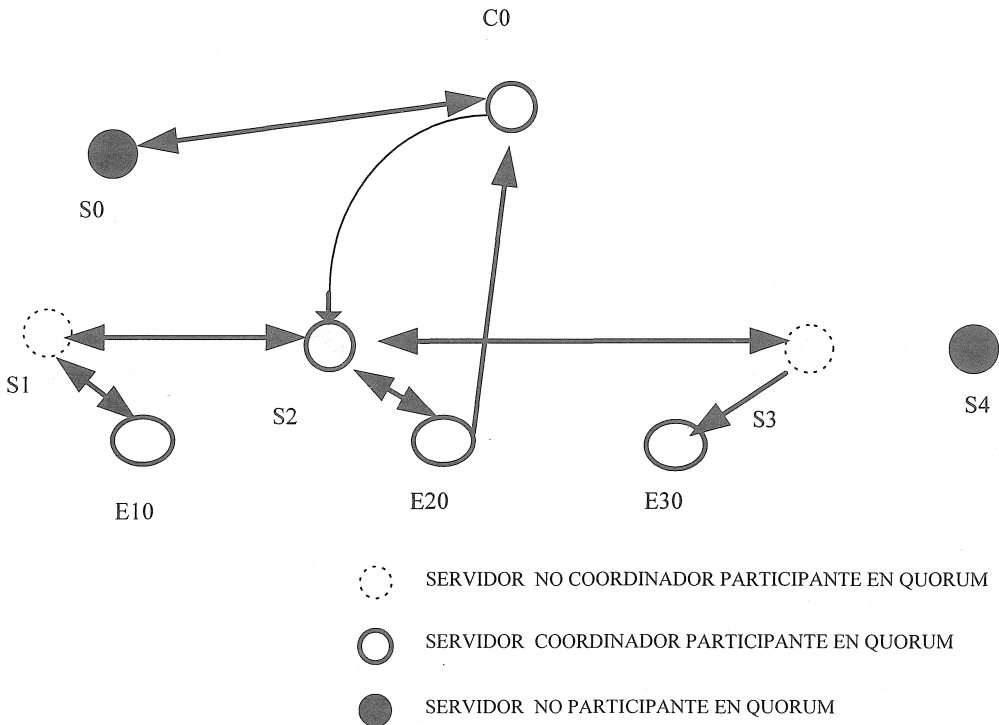


Figura 5: Uso de un Archivo Replicado con Protocolo LAVP

En la figura 5 se ilustra un ejemplo del intercambio de mensajes para el uso de un archivo replicado con LAVP que tiene cuatro réplicas en los sitios 1,2, 3, y 4. Además en esta figura se distinguen los roles de coordinador, participante y no participante en quórum que pueden desempeñar los agentes servidores de los sitios en que hay réplicas de un archivo.

6.4 Recuperación de Sitios con Archivos Replicados con LAVP

Desde el punto de vista de la recuperación por caída de un sitio se presentan tres casos: recuperación de un Servidor no participante en quórum, recuperación de un Servidor participante en un quórum pero no coordinador, y recuperación de un servidor coordinador de quórum.

Un sitio no participante en quórum cuyas réplicas se desactualizan, se recupera al participar en nuevos quóruns, como se mencionó antes. Un sitio participante en un quórum para modificar un archivo, tiene

una tabla de candados en memoria secundaria que revisa al recuperarse. De esta forma puede detectar si tiene candados sobre réplicas de archivos con protocolo LAVP, caso en el cual debe comunicarse con el servidor coordinador de la operación y resolver la situación validando o anulando la transacción.

Un servidor coordinador de operaciones para un archivo, se comporta ante el control transaccional ejercido por el agente cliente como si tuviera un archivo sin replicar; por lo tanto, si se cae durante una transacción, su recuperación es tratada por la lógica usual de recuperación de transacciones.

6.5 Acceso de un Archivo Replicado con CPCS

En este protocolo las modificaciones se hacen siempre en la copia primaria del archivo. Por lo tanto, para esta clase de operaciones, el archivo se comporta como si se tratara de un archivo sin replicación ubicado en el sitio en el que esté la copia primaria. El agente cliente siempre que manipule un archivo replicado con CPCS, determina el sitio de la copia primaria mediante comunicación con su servidor local y dirige a ese sitio todas las operaciones de modificación.

Una vez terminada la transacción que efectuó las modificaciones, el servidor coordinador las difunde a los sitios en que hay réplicas del archivo. Esta difusión no garantiza, sin embargo, que todas las réplicas sean actualizadas, puesto que algunas de ellas pueden estar en sitios aislados o caídos.

En cuanto a las operaciones de consulta, dado que en este protocolo se pueden hacer en la copia primaria o en las copias secundarias a voluntad del usuario, se ha implementado una función que permite las consultas de registros de la copia secundaria, y otra para hacer las consultas sobre la copia primaria.

6.6 Recuperación de Sitios con Archivos Replicados con CPCS

Un sitio que se cae o se aísla y que tiene archivos replicados con CPCS, puede estar en una de dos situaciones: la réplica que tiene es la copia primaria, o la réplica que tiene es la copia secundaria.

Cuando se presentan fallas en un sitio cuya réplica es la copia primaria del archivo, la recuperación del sitio y del archivo se hace como si se tratara de un archivo sin replicar, es decir, se actúa de acuerdo a la lógica de la recuperación de transacciones. De esta forma se garantiza que la copia primaria del archivo siempre estará actualizada y en un estado correcto y consistente.

Si se trata de copias secundarias cuyos sitios se aíslan o se caen mientras se ejecuta una modificación en la copia primaria, éstas se desactualizan y solo se vuelven a ponerse al día cuando, periódicamente y de acuerdo con un criterio preestablecido, el Agente Administrador ejecute la función diseñada con tal fin.

7. CONCLUSIONES

El diseño e implantación de protocolos de replicación de archivos en el sistema DGDBM nos lleva a formular las siguientes conclusiones:

- El manejo de archivos replicados en combinación con archivos no replicados, debe hacerse en forma transparente para el usuario.
- Siempre que un sistema ofrezca protocolos de replicación, también debe ofrecer un esquema de administración de esa replicación.

- La administración de la replicación debe ser independiente de las aplicaciones que manejan los archivos replicados.
- La implementación de un protocolo de replicación debe prevenir la alteración de las propiedades transaccionales del sistema.
- Los aspectos que los protocolos de replicación no contemplan, pero que deben implantarse, requieren de atención especial para que funcionen coordinadamente con el resto del sistema.

En el sistema DGDBM fue posible implantar simultáneamente múltiples protocolos de replicación de archivos, ofreciendo a los usuarios del sistema la flexibilidad de poder decidir el número de copias de cada archivo y el protocolo de replicación asociado. En el futuro, se planea extender estas facilidades llevando estadísticas de uso de los archivos para que el usuario tenga información que le permita decidir dinámicamente el grado de replicación adecuado para un archivo y el protocolo que lo maneja de manera más eficiente.

BIBLIOGRAFÍA

- <Abásolo 94> Abásolo, J., Blanco R., Mendieta, F., "Metodología de Diseño de Bases de Datos y su Implantación sobre Oracle 7", Reporte de Investigación Facultad de Ingeniería, Universidad de los Andes, Bogotá-Colombia, 1994
- <Ahamad 90> Ahamad, M., Ammar, M., Cheung, S., "Replicated Data Management in Distributed Systems", Readings in Distributed Systems, Casavant and Singhal, editores, IEEE Computer Society Press, 1990.
- <Barbara 91> Barbara, L., Gruber, R., Johnson, P., Shrira, L., "A Replicated Unix File System", Operating Systems Review ACM press, vol 25, No. 1, Enero 1991.
- <Birman 93> Birman, K., "The Process Group Approach to Reliable Distributed Computing", CACM, vol 36, No. 12, Diciembre 1993.
- <Castro 95> O. Castro, "Diseño e implantación de protocolos de replicación de archivos en el sistema ", Tesis de Magister, Departamento de Sistemas y Computación, Universidad de los Andes, Bogotá-Colombia 1995.
- <Ceri 84> Ceri, S., Pelagatti, G., "Distributed Databases: Principles and Systems", McGraw Hill Company, 1984.
- <Coulouris 88> Coulouris, G., Dollimore, J., "Distributed Systems", Addison-Wesley, 1988.
- <Franky 95> Franky, M., DGDBM: Programming Support for Distributed Transactions over Replicated Files", ACM Operating Systems Review, Vol 29 No. 3, July 1995, pp 64-74.
- <Gray 75> Gray N., Lorie R. A., Putzolu G. R., Traiger I. L., " Granularity of Locks and Degrees of Consistency in a Shared Data Base", IBM Research, 1975.
- <Hu 92> Hu, P., Wilbur, S., "Low Storage Cost, Partition-Tolerant Dynamic Algorithms for Replicated File Systems", IEEE, marzo 1992.
- <Nelson 93> P.A. Nelson, "GNU dbm (GDBM) versión 1.7", Software libre GNU, disponible mediante ftp anónimo al directorio /pub/gnu del nodo *prep.ai.mit.edu*, Septiembre 1993.
- <Rangel 94> E. Rangel, "DGDBM: Ambiente de programación de aplicaciones transaccionales distribuidas sobre archivos de acceso directo", Tesis de Magister, Departamento de Sistemas y Computación, Universidad de los Andes, Bogotá-Colombia 1994.