

SOBRE MOVILIDAD DE RÉPLICAS

Leticia A. De Luca Yanina E. Fernández Jorge R. Ardenghi

Laboratorio de Investigación en Sistemas Distribuidos
Departamento de Ciencias de la Computación
Universidad Nacional del Sur

Resumen

El objetivo de este trabajo es desarrollar un protocolo que permita la movilidad de una réplica. El protocolo no permite más de tres migraciones dinámica de una réplica, y obliga una permanencia estable de ésta en su nueva locación por un tiempo determinado.

En cada nodo del sistema residen varias réplicas de diferentes items de datos. Para cada réplica, cada uno de estos nodos registra la suma total de todas las referencias. Esta información es utilizada para decidir si migrar una réplica o no en la Política de Decisión y para determinar la nueva locación de la réplica migrada en la Política de Ubicación. Una vez que se decide migrar la réplica, se puede ejecutar el Algoritmo para la Migración, cuya tarea principal consiste en la transferencia de la réplica al nodo destino.

Palabras Claves: réplica, migración, manejador de réplicas, replicación.

1. Introducción

Cuando uno o más nodos referencian frecuentemente a una réplica remota, puede resultar conveniente moverla o copiarla a un nodo más cercano.

Para desarrollar un protocolo que permita la movilidad de una réplica se debe determinar si es posible mover o copiar una réplica y establecer el mecanismo para llevar a cabo la transferencia de la misma al sitio destino.

Los nodos que hacen referencia a la réplica son vecinos, y pueden encontrarse a varios hops de distancia de dicha réplica. Se considera a dos nodos vecinos cuando están a un hop de distancia, formando una vecindad.

La replicación consiste en la reproducción de copias de datos. Está basada en enviar sólo los cambios de datos a todos los nodos en la red donde residen las copias, garantizando así consistencia de datos de tiempo cercano al real.

La replicación es utilizada para mejorar la disponibilidad y el rendimiento de una aplicación.

Si los datos están replicados en dos o más servidores y alguno de ellos falla, se puede acceder a un servidor alternativo aumentando la disponibilidad de los datos.

La replicación puede mejorar los tiempos de respuesta de servicio buscando la réplica más cercana, logrando así una buena performance.

La replicación es buena opción cuando se distribuyen datos estratégicamente a través del sistema. Permite repartir la carga de trabajo entre varios servidores, evitando cuellos de botella y optimizando así recursos de red, HW y técnicas administrativas.

Se considera una réplica como la copia física de un objeto, item de dato o proceso.

En un sistema se pueden encontrar copias múltiples de un objeto en diferentes nodos. Para lograr un sistema tolerante a fallas se crean al menos tres réplicas de objetos. [NDS Cramsession]

Además de considerar tres réplicas, el protocolo permite no más de tres migraciones dinámicas de una réplica, y obliga una permanencia estable de ésta en su nueva ubicación por un tiempo determinado, para evitar sucesivas migraciones de una misma réplica.

La migración de una réplica trata con la realocación transparente de una réplica de un nodo a otro en un sistema distribuido.

La política de migración se basa en decidir migrar una réplica o no, y en elegir la granularidad de la réplica a ser migrada, así como el nodo destino al cual la réplica será movida siguiendo la Política de Ubicación.

El mecanismo de migración consiste en la transferencia propiamente dicha de la réplica al nodo destino y en la transmisión y manejo de los mensajes durante y después de la migración.

Si la réplica elegida además es accedida por nodos que pertenecen a otra vecindad, parecería más conveniente hacer una copia en lugar de moverla. La migración implicaría actualizar en su espacio de dirección la nueva ubicación de la réplica, o realizar una búsqueda cada vez que quiera accederla.

En principio, la copia sería de lectura solamente en el nodo destino, para evitar la existencia de varias copias inconsistentes.

Sin embargo, si no se limita la cantidad de estas copias, en algún momento todos los nodos terminarían requiriendo copia de todo, desaprovechando las ventajas de estar en un sistema distribuido. Por esta razón resulta conveniente mover la réplica de lugar.

2. Ventajas y Desventajas de la Migración

Las facilidades de migración de réplica puede ser implementada en un sistema distribuido para proveer una o más de las siguientes ventajas a los usuarios:

- **Mejor tiempo de respuesta** : los mensajes que se transmiten recorren una distancia menor o nula, por lo tanto, las respuestas a los requerimientos llegarán más rápido.
- **Disminución del tráfico de red** : como los mensajes recorren distancias más cortas, la permanencia de los mismos en la red también es menor.
- **Aumento de la disponibilidad** : alta probabilidad de que ante una partición de la red, la réplica permanezca disponible a aquellos nodos que más la referencian.
- **Utilización de recursos eficientemente** : como consecuencia de las ventajas anteriores, los procesos completarán sus tareas más rápidamente, utilizando menos ciclos de CPU y ocupando por menos tiempo la red.

También presenta algunas desventajas:

- **Aumento de Tráfico de Red** cuando el tamaño de la réplica a migrar es grande y también cuando se migran diferentes réplicas muchas veces.
- **Problemas al informar la nueva localidad de la réplica** a los demás nodos que la referencian. Si se realiza broadcast para informar la nueva localidad, resulta en un aumento considerable de mensajes transmitidos por la red. Si en cambio se elige la política perezosa, disminuye la disponibilidad inmediata de la réplica ya que el nodo que la solicitó tendrá que esperar a que la nueva localidad sea encontrada y comunicada.

Suspensión de tarea. El proceso que accedía a la réplica en modo escritura en el momento de ser migrada, deberá suspender su ejecución mientras dure la transferencia.

3. Migración de Réplicas

3.1 Definición

Migración es la realocación de una réplica de su nodo fuente (locación actual donde reside la réplica) en el nodo destino (locación perteneciente a la vecindad que más la referencia).

3.2 Políticas de Migración

En cada nodo del sistema residen varias réplicas de diferentes ítems de datos.

Para cada réplica, cada uno de estos nodos registra la suma total de todas las referencias. Esta información es utilizada para decidir si migrar una réplica o no en la Política de Decisión y para determinar la nueva locación de la réplica migrada en la Política de Ubicación.

Una vez que se decide migrar la réplica, se puede ejecutar el Algoritmo para la Migración, explicado en el ítem 3 y parte central de este trabajo.

3.2.1 Política de Decisión

La idea de esta política es decidir migrar una réplica o no.

Para determinar cuando migrar una réplica se puede seguir la política del Umbral. Éste valor es el límite para saber si una réplica es muy referenciada o poco referenciada.

Pero tener un solo umbral vuelve inestable al sistema. Se aplica una **política Alto-Bajo**

	Más Referenciada
Alto	Normal
Bajo	Menos Referenciada

Se establecen dos umbrales para determinar en qué región se ubica la réplica según la cantidad de referencias a la misma.

Sea S la suma total de todas las referencias a la réplica en un nodo determinado:

- Si S es mayor que el umbral Alto entonces la réplica pertenece a la región Más Referenciada. Luego se debe seguir la *Política de Ubicación* para determinar:
 - ✓ Si la vecindad elegida contiene el nodo propietario de la réplica, no se lleva a cabo la migración.
 - ✓ En caso de haber dos o más vecindades con el mayor número de referencias, se pueden dar dos situaciones :
 1. Si existe alguna vecindad que contiene el nodo propietario de la réplica (nodo con distancia cero), entonces no hay migración;
 2. Si no, se elige aquel nodo cuya distancia a la réplica sea menor.

✓ En otro caso, se transfiere la réplica siguiendo los 5 Pasos para la Migración, descritos más adelante.

- Si S está entre el límite Alto y Bajo, la réplica pertenece a la región Normal. En este caso no hay migración, el beneficio obtenido en migrar una réplica que es normalmente referenciada no es significativo si se desea mejorar el tráfico de red.

- Si S es menor que el límite Bajo, entonces la réplica pertenece a la región Menos Referenciada. Sólo hay migración si todas las referencias pertenecen a una misma vecindad y ésta no contiene al propietario de la réplica.

Se observa que los valores Alto y Bajo se deben establecer dependiendo de la cantidad de referencias promedio. Estos valores afectan al rendimiento del sistema, ya que, si se elige un valor muy elevado para el umbral Alto, serán pocas las réplicas más referenciadas. En consecuencia se realizarán pocas migraciones y no se disminuirá el tráfico en la red.

Por otro lado, si se elige un valor pequeño para el umbral Alto, no tendrá sentido tener dos umbrales, y además, se realizarán continuas migraciones de réplicas aumentando el tráfico de red.

3.2.2 Política de Ubicación de Réplicas

La idea de esta política es transferir la réplica a un nodo dentro de la vecindad de los nodos que más la referencia.

Asumiendo que se conoce la cantidad de referencias que realiza el nodo a una réplica, es posible determinar, entre dos vecindades distintas, quién la referencia más.

Supóngase tener dos vecindades V1 y V2, compuestas por los siguientes nodos con sus respectivas cantidades de referencia a la réplica y distancias al nodo que contiene la réplica:

V1

Nodos	Referencias	Distancia
N1	3	1
N2	8	0
N3	5	1
N4	1	1
Total	17	

V2

Nodos	Referencias	Distancia
M1	10	12
M2	2	11
M3	3	10
Total	15	

Observemos que el nodo que presenta la mayor cantidad de referencias es M1 perteneciente a la vecindad V2, con un total de 10 referencias.

Sin embargo, se elige la vecindad V1 ya que la suma de todas las referencias es mayor que la suma de la vecindad V2.

3.2.3 Políticas de Actualización de Réplicas

En este ítem se presentan dos posibles políticas de actualización de réplicas.

El Sistema de Replicación sigue la Política de Invalidación de Escritura.

1. Actualización de Escritura: las actualizaciones hechas por un proceso son locales y se hace multicast a los demás manejadores de réplicas que tienen una copia del ítem

que inmediatamente modifican el dato leído por procesos locales. No necesitan comunicación para leer las copias locales de items. Se usa multicast totalmente ordenado, el cual no retorna hasta que el mensaje de actualización ha sido localmente liberado. Todos los procesos están de acuerdo en el orden de actualización. Las lecturas son baratas. Los protocolos de Multicast son caros en SW (deben ser soportados por HW). Permite múltiples lectores y múltiples escritores.

2. Invalidación de Escritura: los item solo pueden ser accedidos en modo solo lectura por uno o más procesos y escrito solamente por uno. Cuando un proceso intenta escribir, primero envía un multicast a todas las copias para invalidarlas, y debe ser reconocido como único escritor antes de poder escribir, así los otros procesos están prevenidos. El resultado final de la invalidación es que solo un nodo continua con la réplica modificada, por lo que se evitan los problemas de inconsistencia. Si uno de los nodos que tienen una copia inválida intenta utilizarla, será rechazado y es bloqueado si existe un escritor. Las actualizaciones son solo propagadas cuando se accede a un dato solamente para lecturas. Distintas actualizaciones pueden llevarse a cabo antes que sea necesario una comunicación. Previene que datos viejos sean leídos por copias invalidadas de items actualizados. Permite múltiples lectores, pero solamente un escritor.

4. Algoritmo para la Migración.

Consiste de los siguientes pasos:

- ① Comprobación de la posibilidad de migración
- ② Elección de la granularidad de la réplica a ser migrada.
- ③ Elección del sitio destino a donde la réplica debe ser movida.
- ④ **Transferencia de la réplica al nodo destino.**
- ⑤ Hacer estable la réplica por un tiempo determinado.

4.1 Comprobación de la posibilidad de migración

Una réplica puede migrarse si se cumplió el tiempo de permanencia estable, y si no superó la cantidad máxima de migraciones posibles.

Esto es, si ya se migró la réplica al menos una vez, verificar que pasó un tiempo determinado de permanencia en dicho nodo y que no migró más de tres veces.

4.2 Elección de la granularidad de la réplica a ser migrada

La granularidad se refiere al tamaño de la parte de réplica que se va a mover. Se debe establecer si se migra la totalidad de la réplica o una parte de la misma.

Migrar la totalidad de la réplica requiere mayor tiempo de transferencia, por lo tanto sólo es conveniente esta opción cuando no existe posibilidad de mover la parte necesitada.

Un tamaño pequeño implica granularidad fina. Si se trabaja con este grado, se corre el riesgo de hacer referencia a un dato ausente, lo cual provoca un fallo. Entonces se puede mover una parte más grande.

La granularidad gruesa tiene sus ventajas y desventajas.

Ventajas : - No existe mucha pérdida de tiempo de transferencia.
- Si un nodo hace referencia a un dato de esa parte seguramente hará referencia a otro dato de la misma parte más tarde y se aprovechará mejor la localidad de referencias.

Desventajas : - La red se bloqueará más tiempo con una transferencia mayor.
- Puede introducir “ falsos compartir “.

4.3 Elección del sitio destino a donde la réplica debe ser movida

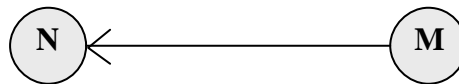
Dentro de la vecindad obtenida con la Política de Ubicación, se elige el nodo con mayor cantidad de referencias.

En caso de haber más de uno, se tomará aquel que presente menos sobrecarga de tareas.

4.4 Transferencia de la réplica al nodo destino

Antes de proceder con la transferencia, se deberá bloquear aquel proceso con permiso de lectura-escritura cuando intente actualizar la réplica, hasta la finalización de la migración. Esto se hace para evitar problemas de inconsistencia.

Asumamos que la réplica que se desea migrar se encuentra originalmente en el nodo M, y el nodo destino es el nodo N.



Luego, la transferencia propiamente dicha, implica tres pasos:

1. Copiar la réplica del nodo M al nodo N.
2. Informar la nueva locación de la réplica.
3. Eliminar la réplica de M.

4.4.1 Copiar la réplica del nodo M al nodo N

Se realiza la copia manteniendo el estado de la réplica del nodo fuente, ya sea solamente de lectura o de lectura-escritura. De esta manera se asegura el mismo comportamiento de la réplica evitando problemas de inconsistencia.

4.4.2 Nueva locación de la réplica

En primer lugar el nodo destino comunica la nueva locación al Manejador de Réplicas.

Luego, el nuevo nodo dueño de la réplica puede notificar dicha locación a aquellos nodos que la accedieron anteriormente en M, mediante un broadcast. Sin embargo, ésta no es la mejor opción porque involucra la transmisión de muchos mensajes, aumentando el tráfico de red.

Se presentan dos propuestas para solucionar este problema:

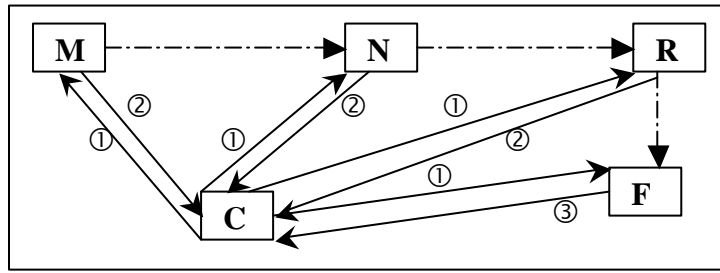
- ✓ Política perezosa con registro:

Luego de copiar la réplica al nodo destino, el nodo M mantiene en un registro la nueva locación para futuros requerimientos.

En la siguiente figura (Figura 1), se muestra el funcionamiento de esta política. Supóngase que el nodo M es el primer propietario de la réplica. Si el nodo C quiere solicitar la réplica, envía un mensaje a M. M le informa que la réplica migró a una nueva localidad N, C debe reenviar el mensaje de solicitud a N, y así seguir hasta obtener la locación actual de la réplica, en este caso F.

Como la cantidad de migraciones está limitada a no más de tres, en el peor de los casos, serán necesarios ocho mensajes para obtener la réplica deseada

Figura 1



- > Migración de la réplica.
- ① —> Mensajes de Solicitud de réplica.
- ② —> Mensajes de Información de migración indicando nueva localidad.
- ③ —> Mensaje de Obtención de la réplica y su nueva localición.

Esta opción tiene la desventaja de que el nodo que solicita la réplica debe realizar una búsqueda de la nueva localición y en consecuencia no la tendrá disponible inmediatamente.

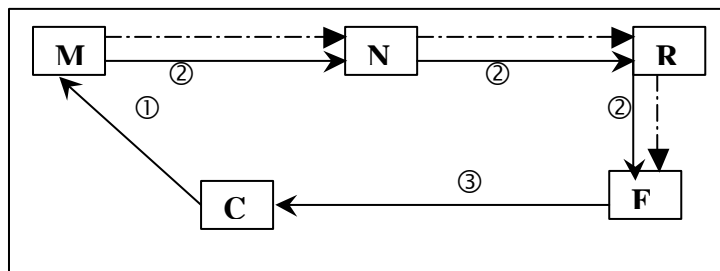
Además, esta alternativa no presenta transparencia en la comunicación entre los nodos. El nodo que requiere la réplica no debería saber que ocurrieron una o varias migraciones. Se opta por una mejor alternativa, en la cual una vez que se envía un requerimiento de acceso a la réplica se debe alcanzar al nodo propietario sin la necesidad de reenviar el mensaje.

En el método anterior el nodo que solicitó la réplica debe realizar la búsqueda accediendo a todos los nodos por donde fue migrando la réplica.

En cambio, en esta técnica, el nodo solicitante no se encarga de realizar la búsqueda, envía el mensaje de solicitud y recibe la réplica junto con la nueva localición, para un uso futuro. (Figura 2).

En el peor de los casos, considerando que solo puede migrar hasta tres veces, se transmitirían cinco mensajes en lugar de ocho, teniendo como ventaja además, la transparencia para el nodo que solicitó la réplica.

Figura 2



- > Migración de la réplica.
- ① —> Mensajes de Solicitud de réplica.
- ② —> Mensajes de Solicitud de réplica indicando quien la requiere.
- ③ —> Mensaje de Obtención de la réplica y su nueva localición.

El nodo C, ignora si la nueva localición de la réplica es la más cercana, para lo cual, en algún momento, si desea optimizar el tiempo de respuesta, puede consultar al Manejador de Réplicas para determinar si existe una réplica con menor costo de acceso.

El costo de acceso a una réplica se establece en función de la distancia entre los nodos involucrados y el tráfico de la red en ese tramo. La réplica mas cercana no asegura el mejor tiempo de respuesta debido a la carga de los canales de comunicación.

Este cálculo se debe realizar debido a que la réplica más cercana no necesariamente dará el tiempo de respuesta óptimo.

✓ Política perezosa sin registro:

Luego de copiar la réplica al nodo destino, el nodo M no mantiene la nueva locación de la réplica.

En futuros requerimientos M solo responderá que no posee dicha réplica y el nodo solicitante deberá reenviar el mensaje al Manejador de Réplicas. Éste, deberá calcular los costos de acceso a la réplica desde el nodo solicitante C, a todos los nodos propietarios y enviar un mensaje conteniendo la locación de la réplica con menor costo de acceso.

Por último, el nodo solicitante C, deberá reenviar el mensaje de solicitud y así obtener la réplica. (Figura 3)

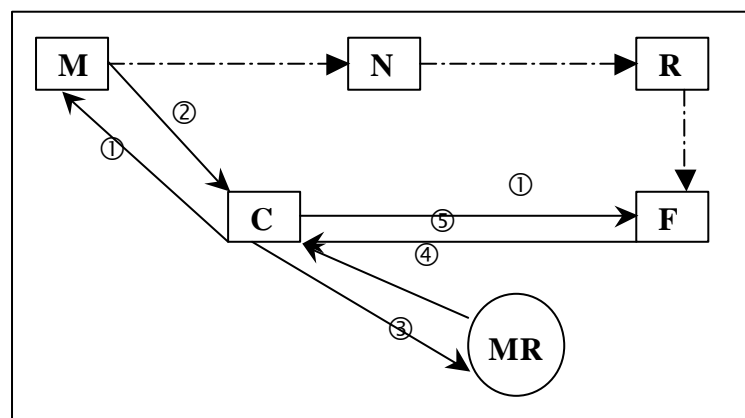


Figura 3

- > Migración de la réplica.
 - ① —> Mensaje de Solicitud de réplica.
 - ② —> Mensaje de Migración sin nueva locación.
 - ③ —> Mensaje de Solicitud de nueva locación.
 - ④ —> Mensaje de Obtención de nueva locación
 - ⑤ —> Mensaje de Obtención de réplica.
- MR** Manejador de Réplicas

En esta política, serán necesarios seis mensajes para obtener la réplica deseada. A diferencia de la política anterior, no depende de la cantidad de migraciones.

Además, no se logra la transparencia ya que el nodo solicitante sabe que ocurrieron migraciones.

Por las razones anteriores, el algoritmo para la migración sigue una política perezosa con registro, aunque no resulte natural que los nodos realicen la búsqueda de la nueva locación de la réplica migrada.

4.4.3 Eliminar la réplica de M

Ahora M no contendrá más la réplica sino la nueva dirección de la misma, N, para comunicársela a aquellos que intenten accederla en un futuro.

Una vez finalizada la transferencia de la réplica, se reanuda la ejecución del proceso suspendido.

4.5 Estabilidad de la réplica.

Para lograr que el sistema sea estable hay que evitar la migración de una réplica que no ha cumplido el tiempo de permanencia en el nuevo nodo. Para lo cual se establece una latencia.

5. Conclusión

En este trabajo se ha presentado una política de ubicación de réplicas que las “regionaliza” reduciendo considerablemente el número de posibles ubicaciones.

También se propuso un protocolo de transferencia se ajusta en forma adecuada a la política de ubicación presentada.

En el futuro debería ponerse énfasis en refinar las métricas sobre las cuales se deben tomar las decisiones de migración.

4. Bibliografía

[BUR 97] BURETTA, M.: *Data Replication. Tools and Techniques for Managing Distributed Information*, Ed. John Wiley & Sons, Inc., 1997.

[COUL 94] COULOURIS, G.F., DOLLIMORE, J., y KINDBERG, T.: *Distributed Systems Concepts and Design*, segunda edición, Ed. Addison-Wesley, 1994.

[MULL 93] MULLENDER, S.J.(ed.): *Distributed Systems*, segunda edición, ACM Press, Ed. Addison-Wesley, 1993.

[TAN 95] TANENBAUM, A.S.: *Distributed Operating System*, Ed. Prentice-Hall, Inc, 1995.

[GSYC 98] GRUPOS DE SISTEMAS Y COMUNICACIONES: Grupos de Procesos Replicados, http://gsync.escet.urjc.es/simple_com/phd-thesis-es/node13.html, Universidad Carlos III, 1998.

[DSGCH06 2000] Chapter 6 - Active Directory Replication
<http://www.microsoft.com/TechNet/win2000/win2ksrv/reskit/dsgch06.asp> ,2000.

[NDS Cramsession] NDS Design and Implementation, Cramsession , 2000,
<http://Cramsession.brainbuzz.conferencesession/novell/nds/guide.asp>