

La forma en que UNIX utiliza los descriptores de archivo y los desplazamientos implícitos es, inherentemente, un mecanismo con memoria del estado. Los servidores deben mantener tablas para establecer la correspondencia entre los descriptores de archivo y los inodos y deben almacenar el desplazamiento actual dentro de cada archivo. Este requisito es la razón por la que NFS, que emplea un servicio sin memoria del estado, no emplea descriptores de archivo e incluye un desplazamiento explícito en cada acceso.

17.5 Replicación de archivos

La replicación de archivos en diferentes máquinas dentro de un sistema de archivos distribuido constituye un útil mecanismo de redundancia para mejorar la disponibilidad. La replicación múltiple puede aumentar también las prestaciones: seleccionar una réplica cercana para dar servicio a una solicitud de acceso da como resultado un tiempo de servicio más corto.

El requisito básico de un esquema de replicación es que las diferentes réplicas del mismo archivo residan en máquinas que sean independientes en lo respecta a los fallos, es decir, que la disponibilidad de una réplica no se vea afectada por la disponibilidad del resto de las réplicas. Este requisito obvio implica que la gestión de la replicación es, inherentemente, una actividad opaca en lo que respecta a la ubicación. Será necesario proporcionar mecanismos para poder colocar una réplica en una máquina concreta.

Resulta deseable ocultar los detalles de la replicación a ojos de los usuarios. El establecimiento de la correspondencia entre un nombre de archivo replicado y una réplica concreta es tarea del esquema de nombrado. La existencia de réplicas debe ser invisible para los niveles superiores. Sin embargo, en los niveles inferiores, es necesario distinguir unas réplicas de otras utilizando nombres de bajo nivel distintos. Otro requisito de transparencia es que se debe proporcionar mecanismos de control de la replicación en los niveles superiores. Esos mecanismos de control de la replicación incluyen poder determinar el grado de replicación y la colocación de las réplicas. En determinadas circunstancias, puede que convenga proporcionar estos detalles a los usuarios. Locus, por ejemplo, proporciona a los usuarios y a los administradores del sistema mecanismos para control del esquema de replicación. El problema principal asociado con las réplicas es su actualización. Desde el punto de vista de un usuario, todas las réplicas de un archivo denotan la misma entidad lógica, por lo que cualquier actualización en una réplica debe reflejarse en todas las réplicas restantes. De manera más precisa, será necesario preservar la semántica de coherencia relevante cuando los accesos a las réplicas se contemplen como accesos virtuales a los archivos lógicos de las réplicas. Si la coherencia no tiene una importancia fundamental, puede sacrificarse en aras de la disponibilidad y de las prestaciones. En este compromiso fundamental dentro del área de la tolerancia a fallos, la elección que hay que hacer es entre preservar la coherencia a toda costa, creando así el potencial de un bloqueo indefinido, o sacrificar la coherencia en algunas (en teoría, pocas) circunstancias de fallo catastrófico, para poder garantizar que las operaciones progresen de manera continua. Locus, por ejemplo, emplea de manera intensiva los mecanismos de replicación y sacrifica la coherencia en caso de que se produzca una partición de la red, con el fin de garantizar la disponibilidad de los accesos de lectura y escritura. Ibis utiliza una variante de la técnica de copia principal. El dominio de las correspondencias de nombres es una pareja *<identificador-réplica-principal, identificador-réplica-local>*. Si no existe ninguna réplica local, se utiliza un valor especial. Por tanto, estas correspondencias son relativas a cada máquina concreta. Si la réplica local es la principal, la pareja contendrá dos identificadores idénticos. Ibis soporta un mecanismo de replicación bajo demanda, que es una política de control de replicación automático similar a los mecanismos de almacenamiento en caché de archivos completos. Con una replicación bajo demanda, la lectura de una réplica no local, hace que esa réplica se almacene localmente en la caché, generando así una nueva réplica no principal. Las actualizaciones se realizan únicamente en la réplica principal y hacen que las demás réplicas queden invalidadas, enviándose los oportunos mensajes. No se garantiza la invalidación atómica y serializada de todas las réplicas no principales. Por ello, puede darse el caso de que una réplica obsoleta se considere válida. Para satisfacer los accesos de escritura remotos, lo que se hace es migrar la copia principal hasta la máquina solicitante.