Sistemas Operativos Distribuidos

Práctica miniAFS

Objetivo

- Desarrollar un SFD similar a AFS:
  - Semántica de sesión, cache en disco de cliente, ficheros completos en cache de cliente, validación iniciada por el servidor, *callbacks*, etc.
- Pero muy simplificado:
  - No hay directorios
  - Monousuario
  - Sólo una aplicación/nodo en cada momento
- Tecnología utilizada: RPC de ONC
- IMPORTANTE:
  - Debe poder transferir ficheros binarios, no sólo de texto
    - Pruebe la práctica con ficheros que tengan al menos un byte a 0.

Arquitectura de servicio

- En el servidor: vice
  - Ficheros almacenados en subdirectorio *Ficheros/*
  - Debe desarrollarse IDL (*vice.x*) y servidor (*vice.c*)
- En el cliente: venus
  - Cada "máquina" cliente en un directorio (*cliente1*, *cliente2*, ...)
  - Cache de cliente en subdirectorio *Cache/*
  - Funcionalidad repartida entre biblioteca de servicio y servidor venus
  - Biblioteca de servicio (*venusbib.c*): Debe desarrollarse
    - Se enlaza con aplicaciones y actúa como cliente de servicio vice
  - Servidor (*venus.c*): Debe desarrollarse
    - Se encarga de invalidaciones de ficheros en cache
    - Actúa como servidor para vice
    - Su interfaz de servicio (*venus.x*) ya está definida
      - Servicios *revocar_callback* y *activar_callback*

Interfaz a las aplicaciones

- int OpenFile(char *fichero, int flags)
  - Servicio de venusbib; Retorna descriptor de fichero o error
  - *Flags* permitidos: O_RDONLY, O_RDWR, O_CREAT
    - **Cuidado**: recuerde que en POSIX O_CREAT no trunca el fichero
- int CloseFile(char *fichero, int desc)
  - Servicio de venusbib; Retorna éxito o error
- read, write, lseek, ftruncate, ...
  - Misma estrategia que en AFS:
    - No son servicios de venusbib
    - Son directamente los servicios POSIX

Arranque del sistema

1. Arranque de vice en servidor
2. En cada cliente:
   1. Definir variable PROGNUM
      - Identifica qué nº de programa usará venus en ese nodo cliente
   2. Definir variable SERVIDOR
      - Indica a venusbib en qué máquina está vice
   3. cd al directorio del cliente
   4. Arrancar venus
   5. Ir ejecutando las distintas aplicaciones

Fase 1: Sin coherencia

- No es necesario arrancar venus (ni definir PROGNUM)
- Modelo carga/descarga igual que en AFS
- En OpenFile:
  - Si no está en cache, se descarga
  - A criterio del alumno, enviar el fichero con una sola RPC o múltiples
- En CloseFile:
  - Si modificada copia en cache, se copia a servidor
  - ¿Cómo detectar si modificada copia? Una posibilidad:
    - en descarga poner tiempo de modificación a 0 (usando *utime*)
    - al cerrarlo comprobar si es distinto de 0 (usando *stat*)

Fase 2: Coherencia restringida

- Hay coherencia pero para situaciones simplificadas:
  - Aplicación no puede abrir varias veces un fichero simultáneamente
  - No sesiones de escritura simultáneas sobre el mismo fichero
- Modificaciones sobre versión anterior:
  - Programar servicio *revocar_callback* de venus (no *activar_callback*)
    - Debe borrar la copia en cache
  - Desarrollar algoritmo de coherencia:
    - Expandir RPCs de vice para incluir identificación de cliente
      - *Host* + nº de programa de venus
    - Crear estructuras de datos: qué clientes tienen copias de cada fichero
    - Nueva versión de fichero: vice *revocar_callback* de clientes afectados
      - Todos los que tienen copia excepto escritor

Fase 3: Coherencia mejorada

- Complicaciones al desaparecer algunas simplificaciones
  - Aplicación abre 2 veces un fichero: una RDONLY y otra RDWR
    - Sólo debe volcar copia modificada si sesión que se cierra es RDWR
  - Aplicación abre fichero F, llega invalidación, y abre por segunda vez
    - Segunda apertura requiere copia en cache con última versión de fichero
      - Si no nueva invalidación, nuevas aperturas de la aplicación usan esa copia
    - Pero primera apertura debe seguir trabajando con su propia copia
      - Que podrá ser la nueva versión en el sistema cuando la cierre
    - Si hay otra invalidación, se repite la situación:
      - nuevas aperturas verán última versión y previas se quedan con copia propia
  - Sesiones de escritura a la vez: posibles condiciones de carrera
    - Una borra copia en cache de la otra

Fase 3: Propuesta de solución

- En OpenFile crear dos nombres (F y .F_desc)
- Al volcar fichero modificado al servidor (CloseFile):
  - venusbib debe indicar el descriptor de fichero de la copia modificada
  - vice invoca *revocar_callback* de clientes afectados (incluido escritor)
    - Nombre F desaparece de cache de escritor; Se mantiene .F_desc
  - vice invoca *activar_callback* de escritor pasando nombre y descriptor
    - venus crea nombre F asociado a .F_desc