

Proyecto (20%)

1. Sinopsis

Se desea que Ud. diseñe el protocolo de una aplicación cliente/servidor con un servidor central y tres servidores de descarga desde los cuales se bajarán videos. Específicamente cada usuario recuperará el video en tres partes, en principio cada trozo desde cada uno de los servidores de descarga.

El sistema no podrá soportar más de 5 clientes de videos simultáneamente. En ese caso, debe definirse una lista de espera para contener las solicitudes de descargas

Todos los actores: servidor central, servidores de descarga y clientes dispondrán de una consola o ventana de interacción que indicará explícitamente cada uno de los mensajes que intercambia y las acciones que realizan.

En caso de caída de uno o dos servidores de descarga, el usuario debe poder continuar bajando el video, siempre en tres partes.

2. Servidor Central

Inicialmente es bien conocido el puerto y el IP del servidor central. La BD del servidor central está vacía por lo que tanto los clientes y como los servidores de descarga deben inscribirse ante el servidor central. Los servidores de descarga indicarán al servidor central los videos que ofrecen, su IP y puerto de escucha (esto último depende de como definan el protocolo). El servidor central coordinará que cada servidor de descarga tenga los videos que los demás tienen. Este intercambio debe verse explícitamente en la ventana de interacción

La inscripción de los usuarios deberá ser previa al proceso de descarga así como también la sincronización de todos los servidores de descarga para que cada uno tenga los videos de los demás.

Es responsabilidad del servidor central manejar la lista de espera en caso de llegar al tope de 5 cliente descargando. Al terminar un cliente con sus tres partes comienza la descarga del primero de la lista de espera. En caso de caída del servidor central debe enterarse el cliente al hacer alguna solicitud sin dar errores fatales que impliquen la caída del cliente.

Por último el servidor central debe llevar estadísticas de descargas y clientes atendidos que podrán solicitarse por su consola. Los comandos son:

NUMERO_DESCARGAS_VIDEO

VIDEOS_CLIENTE

Para poder responder a estas estadísticas el servidor central debe ser informado por los servidores de descarga de su actividad para poder actualizar las estadísticas.

No deben olvidar que el propio servidor central y la consola deben ser concurrente para poder atender clientes, servidores secundarios y tener la consola siempre a disposición para ejecutar cualquiera de los dos comandos de estadísticas.

3. Servidores Secundarios

Una vez inscritos, los servidores de descarga estarán disponibles para los clientes que soliciten videos. En caso de caída de un servidor el video continuará descargándose en tres partes. En ese caso uno de servidores debe asumir dos partes de la descarga. Si se cae un segundo servidor, el que resta asume las tres partes. Si se caen los tres servidores, deben enterarse el servidor central y los clientes sin dar errores fatales.

Al levantarse el servidor de descarga caído debe asumir automáticamente las partes que delegó si aún no ha terminado las descargas que tenía antes de la caída.

Los servidores de descarga ofrecen como estadísticas:

VIDEOS_DESCARGANDO

VIDEOS_DESCARGADOS

Al igual que en el servidor central, debe manejarse la concurrencia tanto del propio servidor de descarga como de su consola, es decir, puede ejecutar los comandos de estadísticas sin importar la cantidad de clientes que estén conectados.

4. Cliente

Los clientes solicitan el video al servidor central quien comunica a los servidores de descarga, la IP y el puerto del cliente. También puede ser comunicando al servidor de descarga el puerto del cliente ... que hacer depende de como definan su protocolo. En caso de caídas, el cliente debe recibir las notificaciones de error para ser informado del estado de su solicitud de descarga.

Los comandos del cliente son:

INSCRIBIR *nombre*

LISTA_VIDEOS

VIDEO *nombre*

El *status* de descarga debe verse en tiempo real lo cual no impide que el cliente pueda escribir otros comandos. Es decir, al igual que con el servidor central y los servidores secundarios, la consola siempre debe estar disponible mientras se estén descargando videos.

5. Consideraciones generales

- Además de todos los servidores (un central y tres de descarga) la corrida debe realizarse con al menos tres clientes.
- Los archivos de datos deben ser permanentes, es decir, la información debe perdurar aún cuando los servidores o computadores estén caídos. Los datos se guardarán en el formato que más les convenga. Puede ser XML o un manejador de BD como `mysql` o `postgres`.
- La concurrencia debe manejarse en varias instancias a nivel de respuestas de solicitudes y comandos en consola.
- Debe indicarse explícitamente el protocolo a nivel de capa de aplicación mediante un diagrama de secuencia o máquina de estado finito. En el RFC959 del protocolo `ftp`, sección 6, se muestran los diagramas de estado o máquinas de estado finito del protocolo `ftp` que pueden usar a modo de ejemplo para definir su protocolo. Pueden tener un diagrama por cada orden o comando o pueden integrarlo en un sólo gran diagrama. También pueden usar diagramas de secuencia como se usan en UML.
- Use videos lo suficientemente grandes para poder apreciar toda la interacción del sistema. Además mientras más *feedback* coloque en la consola más fácil será la corrección del proyecto y la propia depuración por parte de Uds. porque se sabrá en todo momento que realiza cada programa.
- La implementación puede ser en Java, C o Python y la concurrencia se manejará con procesos livianos (hilos). Las librerías que pueden utilizar son *sockets* (la cual existe en casi cualquier lenguaje de programación) o procedimientos remotos (RMI para JAVA y RPC para C).
- No es necesario implementar cualquier tipo de interfaz gráfica
- Aunque no está previsto el manejo de tolerancia a fallas, deben manejar los errores de caídas de enlace o servidores sin dar errores fatales.

6. Informe

- Debe tener una breve descripción de la estructura del código (si le parece conveniente puede colocar un diagrama de clases), tener los diagramas de secuencia o máquinas de estado finito de los protocolos entre cada uno de los actores. Es importante que estos diagramas sean lo más detallado posible indicando claramente la semántica y que se ejecuta una vez llegado el mensaje. La clara definición de cada protocolo es importante ya que será evaluado y revisado los aspectos que lo diferencien de los otros proyectos. Finalmente deben tener las conclusiones donde indiquen los criterios usados para la selección del lenguaje y la librería de aplicaciones cliente/servidor.
- Deben entregar el código con al menos la documentación de cada método y quizás las líneas relevantes porque tendrá una calificación asignada. Por tanto documentar el código será un punto a evaluar. La corrida es presencial.

- En caso de usar código que esté en Internet debe indicarlo explícitamente en el informe para excluir cualquier duda sobre plagio.