PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
DEPARTAMENTO INGENIERIA DE SISTEMAS

Introducción a Sistemas Distribuidos – Proyecto 2
Período Académico 2018-3

Recuperación de la Memoria Histórica

Objetivos

El objetivo general del proyecto final del curso ISD se centra en la apropiación práctica de los conceptos relacionados con el desarrollo de aplicaciones y la utilización de middlewares en ambientes de cómputo distribuido. Este propósito se logra mediante la creación de un sistema distribuido que brinde una solución a una problemática del mundo real. Los objetivos específicos son:

- Poner en práctica los conceptos de arquitecturas distribuidas, haciendo énfasis en algunas de las problemáticas clave de gestión de la concurrencia: gestión de archivos, control de concurrencia, manejo transaccional y recuperación ante fallas.
- Implementar, mediante el uso de RMI, un sistema no centralizado que logre la cooperación entre procesos mediante la definición de protocolos de interacción bien definidos.

Para este período académico, el propósito específico es desarrollar una solución basada en el modelo arquitectónico P2P. En particular, el proyecto se focalizará en la gestión de archivos aplicada al manejo de la reconstrucción de la memoria histórica de las víctimas del conflicto armado. El mecanismo para compartir archivos debe funcionar en forma eficiente no centralizada.

Descripción del Sistema a Desarrollar

En el marco de este proyecto, para la reconstrucción de la memoria histórica de las víctimas del conflicto armado, se requiere tener un sistema que compile archivos provenientes de diversas fuentes con información relevante sobre un tópico y genere un nuevo archivo *recapitulativo*. El sistema de gestión de los archivos, tanto de fuentes como generados, se debe implementar mediante una aplicación distribuida inspirada en el protocolo BitTorrent [1]; una descripción sencilla de este protocolo se encuentra en [2][3][4]. En esencia, los archivos son divididos en bloques pequeños, los cuales pueden ser descargados en forma concurrente desde diferentes *peers*.

El proyecto consiste en hacer una implementación de una aplicación inspirada de BitTorrent que permita a un usuario generar un archivo que recapitula información relacionada con el conflicto armado a partir de un conjunto de archivos fuente compartidos ubicados en diferentes máquinas. El usuario puede especificar qué archivos fuente utilizar mediante un archivo descriptor que incluye en su primera línea el nombre del archivo a generar y en las siguientes los nombres de los archivos fuente deseados. Se debe tener en cuenta que:

- el paralelismo de la descarga se debe dar entre múltiples *peers*; la solicitud de bloques a cada uno de éstos debe hacerse buscando balancear el número de bloques descargado de cada máquina que posee el archivo fuente requerido.
- se pueden estar realizando simultáneamente descargas de varios archivos; evidentemente, todas las máquinas pueden ser a la vez fuentes y receptoras de archivos.
- cuando un archivo fuente aún no esté disponible, la solicitud debe aplazarse sin bloquear la continuidad de otras descargas.

- un nuevo archivo recapitulativo, una vez ha sido creado, queda disponible para descarga por cualquiera de los peers.
- cuando un peer descarga un archivo, inmediatamente, puede ser parte del pool de máquinas que pueden enviar bloques en el proceso de descarga del mismo.
- cuando una máquina que está haciendo las veces de fuente se desconecta o elimina sus contenidos, la descarga debe continuar a partir de otro peer hasta completar la operación.

Para facilitar el desarrollo del proyecto, se van a realizar las siguientes simplificaciones:

- se manejarán solo archivos tipo texto con un tamaño de 8 líneas; en cada línea de un archivo se pone el nombre del archivo seguido de un guion y del número de la línea; se sugiere usar nombres cortos que faciliten el proceso de pruebas y la validación del funcionamiento del sistema.
- la división por bloques se realizará a nivel de líneas, de forma tal que el bloque i corresponderá entonces con la línea i del archivo.
- los archivos generados también son de 8 líneas; para simular su generación, se debe tomar al menos una línea de cada uno de los archivos fuente, las líneas a incluir se seleccionan en forma aleatoria.
- se cuenta con un proceso servidor único de nombres, que permite localizar en que máquinas se encuentra un archivo determinado; éste es el único componente centralizado que está permitido en el desarrollo del proyecto.

Al iniciar la operación, todos los *peers* cuentan con al menos una docena de archivos, cada uno de los archivos debe estar replicado en tres máquinas. De igual forma, el peer debe tener al menos una veintena de archivos descriptores con solicitudes de generación de archivos a partir de los ya existentes o incluso de los que van a ser generados durante la operación. Para el arranque del sistema, cada peer debe registrar sus archivos ante el servidor de nombres. Una vez todos los *peers* hayan terminado el registro, el servidor de nombres envía una señal de inicio a todos. No se requiere dar ninguna entrada manual al sistema. La interface gráfica, tanto de los *peers* como del servidor de nombres debe permitir observar fácilmente la traza de lo que realiza cada máquina. Por supuesto, al final se podrán abrir los archivos generados y transferidos para verificar el correcto funcionamiento del sistema.

Comunicación Distribuida

Deben evitarse mecanismos centralizados que puedan constituirse en cuellos de botella. La única excepción es el servidor de nombres. Dados los objetivos de aprendizaje del proyecto, no se pueden utilizar middlewares existentes que ya implementen estos esquemas de comunicación y transferencia de archivos. Todo el desarrollo se debe realizar utilizando directamente los mecanismos proporcionados por RMI de Java.

Equipos de Trabajo

El proyecto se realizará en grupos de trabajo de mínimo 2 personas y máximo 3 personas.

Entrega y Condiciones

La entrega se realizará en la fecha informada por el profesor en el cronograma detallado del curso. Ésta se realizará el mismo día de la sustentación, antes de las 9am. La entrega se debe hacer a través de UVirtual, siguiendo las directrices de entrega dadas por el profesor para tal fin. Además del código, se debe adicionar la documentación correspondiente con el esquema de procesos y sus interacciones, el diagrama UML para identificar las clases utilizadas y la descripción general de los componentes principales. Igualmente, se debe presentar el esquema de pruebas y el análisis de los resultados obtenidos.

No se considera documentación al código fuente, pero si se espera que esté bien estructurado y sea de fácil comprensión. No se requiere hacer interfaces gráficas sofisticadas, pero sí de fácil uso y que permitan verificar el correcto funcionamiento del sistema distribuido corriendo en al menos 3 máquinas de forma ágil y flexible. Se debe hacer la sustentación con el código que se entrega y deben estar presentes todos los integrantes del grupo.

No puede existir replicación de código entre grupos, lo cual se consideraría plagio.

Referencias

[1] Cohen, Bram. “BitTorrent - a new P2P app”, 2001.

[2] Wikipedia, “BitTorrent”, https://es.wikipedia.org/wiki/BitTorrent#cite_note-BC-1, consultado Oct/2018.

[3] Carmack, Carmen. “How BitTorrent Works”, https://computer.howstuffworks.com/bittorrent.htm, consultado Oct/2018.

[4] Bell, Jordan. “How BitTorrent Works”, video en https://www.youtube.com/watch?v=MMnsBJeb0IQ, consultado Oct/2018.