Laboratorio 3: Programación asistida para Frameworks sobre Cálculo Distribuido

Paradigmas 2023 - FAMAF

En este laboratorio vamos a re-implementar el lector automático de feeds del Laboratorio 2 sobre una arquitectura distribuida, a través del framework Spark, utilizando como lenguaje de programación Java[1]. Además, vamos a aplicar diferentes aproximaciones para hacerlo: programación asistida por inteligencia artificial, ejemplos de proyectos tipo o consultas a sitios de referencia.

Organización del laboratorio

En este laboratorio, además de producir código que efectivamente implemente el lector automático en una arquitectura distribuida, vamos a documentar diferentes procesos para llegar a obtener los detalles técnicos necesarios para implementar en Spark.

En una **primera etapa individual** (entrega el día jueves **1 de junio**), cada miembro del equipo documentará un proceso diferente de desarrollo del proyecto, usando alguna de las siguientes estrategias:

- chatGPT u otra inteligencia artificial generativa orientada a diálogo general, no específicamente código (Claude, Flan-T5, u otras que puedan encontrar en el chatbot arena o en el timeline de modelos disponibles de Omar Sanseviero) en castellano, en inglés u en otro idioma que manejen con suficiente precisión.
- una inteligencia artificial específicamente orientada a código, como coPilot, StarChat, u otras.

[1] Si alguien siente una fuerte necesidad de hacerlo en otro lenguaje, sepan que Spark tiene soporte integrado para Scala, Java, SQL, R y Python. Pueden negociar adaptaciones con su profesor responsable.

- algún traductor de código, como aiTrans
- código de ejemplo obtenido de algún proyecto de ejemplo en github, alguna notebook de colab, kaggle o alguna referencia del mismo proyecto Spark.
- búsquedas en Stack Overflow, o la internet en general.

En una **segunda etapa grupal** (entrega el día jueves **15 de junio**), el conjunto del equipo producirá un documento en el que compararán los tres procesos, sus ventajas y desventajas, compararán los tres códigos obtenidos y justificarán la elección de uno de los tres códigos como mejor opción para el proyecto. A partir de ahí le

Objetivos del programa

El programa a desarrollar tiene que ofrecer las mismas funcionalidades que el programa del Laboratorio 2, pero ahora implementado sobre Spark. Adicionalmente, tiene que ofrecer la funcionalidad de recuperar documentos por palabras clave (como lo haría un motor de búsqueda en la web).

Si no disponen de una versión operativa del Laboratorio 2, podrán pedir a su profesor responsable que les asigne la versión de un grupo que se preste voluntariamente a compartir su código.

Funcionalidades de la entrega individual

- Obtener diferentes feeds de noticias, conformando una colección de documentos, cada uno con una noticia recibida de algún feed.
- Parsear cada uno de los documentos de la colección, asignando el tipo de parser adecuado para cada uno.
- Imprimir las diferentes noticias por consola, mostrando en un formato legible el texto, la fecha, etc.
- Imprimir el listado de entidades nombradas encontradas en toda la colección de documentos

- Imprimir el número de ocurrencias de cada entidad nombrada, y de cada clase y subclase de entidad nombrada, incluyendo las subclases por tema.

Funcionalidades de la entrega grupal

- Recuperar los documentos que contengan una determinada palabra o entidad nombrada, ordenados desde el documento que tiene la mayor cantidad de ocurrencias de la misma al documento que tiene la menor cantidad de ocurrencias. Para hacerlo deberán crear un índice invertido de la colección de documentos.

Requisitos propios de paradigma

Es imprescindible que usen las funciones **`map`** y **`reduce`** provistas por el framework para procesar en paralelo diferentes documentos (identificar las entidades nombradas en el lab individual, contar las ocurrencias de palabras en el lab grupal) y después integrar los resultados parciales obtenidos para cada uno de los documentos (sumar el número de ocurrencias de diferentes entidades nombradas en diferentes documentos en el lab individual, crear el índice invertido en el lab grupal).

Es imprescindible que usen las funcionalidades propias de Spark para las consultas interactivas.

Estructura de los informes

Estructura del informe individual

El objetivo del informe individual es que describan las diferentes etapas para pasar del código de lectura de feeds y conteo de entidades nombradas al código en el que esta funcionalidad está implementada sobre Spark.

Para ello, se recomienda que el informe conteste las siguientes preguntas (posiblemente, con una sección para cada una de ellas):

- ¿Cómo se instala Spark en una computadora personal?
- ¿Qué estructura tiene un programa en Spark?

- ¿Qué estructura tiene un programa de conteo de palabras en diferentes documentos en Spark?
- ¿Cómo adaptar el código del Laboratorio 2 a la estructura del programa objetivo en Spark?
- ¿Cómo se integra una estructura orientada a objetos con la estructura funcional de map-reduce?

En los casos en los que se basen en una implementación de ejemplo (e.g. una notebook o un tutorial) deberán identificar las partes del código de ejemplo que cubren las respuestas a cada una de las preguntas, y pegarlas como parte de la respuesta. En los casos en los que interactúen con un sistema predictivo o inteligencia artificial, deberán pegar como parte de la respuesta capturas de pantalla de sus interacciones con el sistema predictivo o inteligencia artificial.

Estructura del informe grupal

El objetivo del informe grupal es que describan las ventajas y desventajas de las diferentes soluciones individuales, y los motivos por los que eligen una de ellas. También deberán describir la forma en que se incorpora la funcionalidad de recuperar documentos por palabras clave, si hicieron consultas a una inteligencia artificial, si se basaron en proyectos de ejemplo, etc.

Puntos Extras

- Integrar en estructura map-reduce también la obtención de feeds.
- Usar una librería específica para hacer un histograma de la frecuencia de todas las palabras de los feeds, incluyendo las entidades nombradas.

Entrega y Defensa

Fecha límite de **entrega individual**: Jueves **1 de Junio** de 2023 a las 23:59:59 hs. Entrega del código fuente + informe mediante push en bitbucket en el repositorio asignado por la cátedra al estudiante. Día y horario de la defensa a coordinar con el docente asignado a su grupo.

Fecha límite de **entrega grupal**: Jueves **15 de Junio** de 2023 a las 23:59:59 hs. Entrega del código fuente + informe mediante push en bitbucket en el repositorio asignado por la cátedra al grupo. Día y horario de la defensa a coordinar con el docente asignado a su grupo.