Patrones de Diseño: MapReduce

Pablo Peñaranda - 201922871

El proyecto 'avro-hadoop-starter' es una iniciativa que proporciona ejemplos de trabajos MapReduce en Java, Hadoop Streaming, Pig y Hive que permiten la lectura y/o escritura de datos en formato Avro. Su propósito principal es mostrar cómo trabajar con Avro en el contexto de Hadoop, al tiempo que ofrece ejemplos de trabajos MapReduce en diferentes lenguajes y tecnologías. El proyecto se estructura en directorios correspondientes a cada tecnología, lo cual facilita la ubicación de ejemplos específicos para cada una de ellas. Puede consultar el proyecto en este enlace: https://github.com/miguno/avro-hadoop-starter.

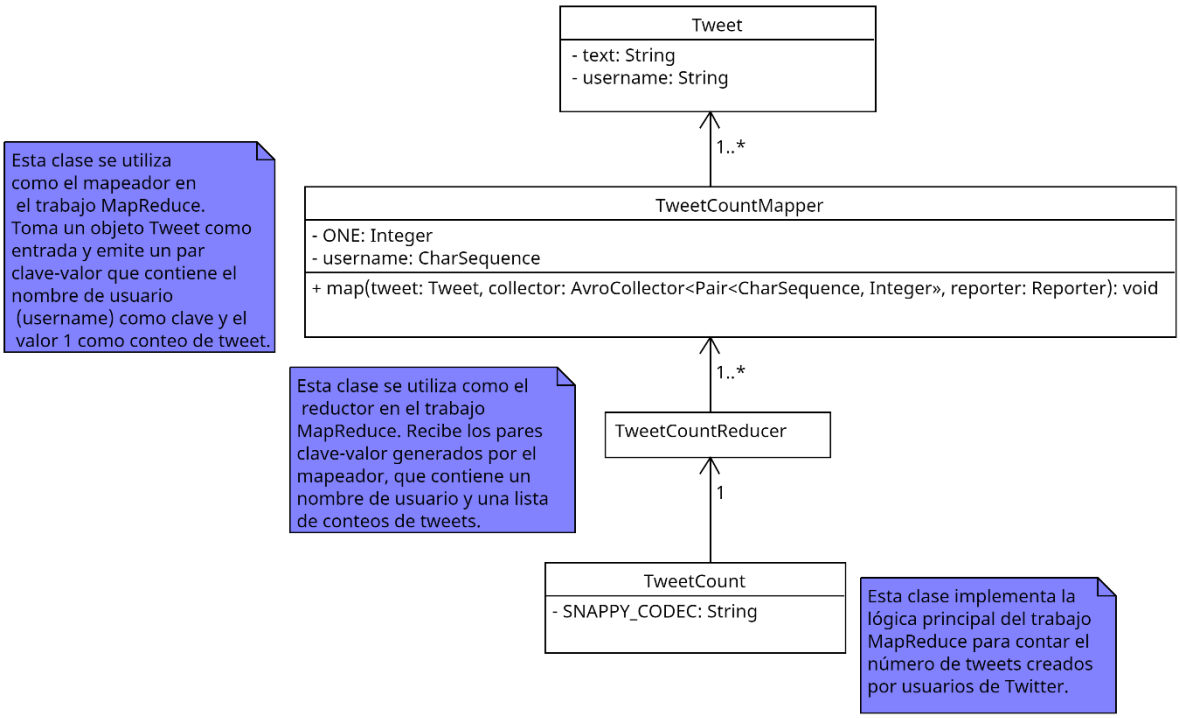
Al trabajar con Avro en un entorno distribuido como Hadoop, se presentan diversos desafíos de diseño que deben abordarse. Estos incluyen la configuración adecuada de las dependencias de Avro para su integración fluida con Hadoop, la implementación eficiente de la serialización y deserialización de datos Avro en un entorno distribuido, y la correcta integración con el ecosistema de Hadoop, que implica la interacción con componentes adicionales.

El patrón MapReduce, ampliamente utilizado en el procesamiento distribuido de datos, se aplica en el proyecto 'avro-hadoop-starter' mediante los ejemplos de trabajos MapReduce proporcionados en diferentes lenguajes y tecnologías compatibles. Estos ejemplos ilustran cómo utilizar el patrón MapReduce para leer y/o escribir datos en formato Avro en un entorno distribuido. En la fase de 'map', se procesan y asignan los pares clave-valor a partir de los datos de entrada, mientras que en la fase de 'reduce', se realiza la agregación o combinación de los resultados intermedios generados por la fase de 'map', para obtener el resultado final deseado.

El uso del patrón MapReduce en este proyecto tiene sentido debido a las ventajas que brinda en el contexto de Hadoop. Al dividir el trabajo en fases de 'map' y 'reduce', se puede aprovechar la capacidad de procesamiento distribuido y paralelo de un clúster de Hadoop. Esto ofrece beneficios significativos en términos de escalabilidad, rendimiento y tolerancia a fallos. Al distribuir las tareas en diferentes nodos del clúster, se puede lograr un procesamiento eficiente de grandes volúmenes de datos.

Sin embargo, es importante tener en cuenta que el uso del patrón MapReduce también presenta desventajas. Su implementación puede ser compleja, requiriendo un sólido entendimiento de los conceptos y principios asociados con el patrón. Además, en comparación con enfoques más modernos como Apache Spark, el patrón MapReduce puede tener limitaciones en términos de flexibilidad y velocidad de procesamiento.

Es importante destacar que, aparte del patrón MapReduce, existen otras alternativas para abordar los desafíos que este proyecto busca resolver. Tecnologías como Apache Spark y Apache Flink ofrecen modelos de programación más flexibles y eficientes para el procesamiento distribuido y análisis de datos en entornos de big data. La elección de la tecnología y el enfoque dependerá de los requisitos específicos del proyecto y las necesidades del caso de uso.



UML Resumido que exceptúa algunos detalles por motivos ilustrativos además de evitar términos complejos.

Glosario:

- Hadoop Streaming: Es una utilidad de Hadoop que permite ejecutar trabajos MapReduce utilizando scripts o programas escritos en lenguajes como Python, Ruby o Perl. Permite procesar datos en diferentes formatos sin necesidad de utilizar Java.

- Pig: Pig es una plataforma de alto nivel para analizar grandes conjuntos de datos en Hadoop. Proporciona un lenguaje de scripting llamado Pig Latin, que permite expresar operaciones de transformación y análisis de datos de manera sencilla y eficiente. Pig optimiza y ejecuta estas operaciones en clústeres de Hadoop.

- Hive: Hive es una infraestructura de data warehousing construida sobre Hadoop que proporciona una interfaz SQL-like para consultar y analizar grandes volúmenes de datos almacenados en Hadoop. Hive traduce consultas en Hive Query Language (HiveQL) a trabajos MapReduce, permitiendo a los usuarios trabajar con datos estructurados utilizando un lenguaje similar a SQL.

- Formato Avro: Avro es un formato de datos de alto rendimiento y compacto. Es utilizado para serializar datos en Hadoop y otras tecnologías de big data. Avro proporciona un

esquema flexible y evolutivo que permite almacenar datos estructurados junto con su esquema en un archivo compacto. Esto facilita el intercambio y procesamiento eficiente de datos en diferentes sistemas.

- Apache Spark: Apache Spark es un motor de procesamiento de datos de código abierto y distribuido. Proporciona una plataforma unificada para el procesamiento de datos en tiempo real, procesamiento por lotes, aprendizaje automático y análisis interactivo. Spark ofrece un modelo de programación flexible y eficiente, y puede trabajar en conjunto con Hadoop y otros sistemas de almacenamiento distribuido.

- Apache Flink: Apache Flink es otro motor de procesamiento de datos distribuido y de código abierto. Se centra en el procesamiento de datos en tiempo real y por lotes, y ofrece un modelo de programación de alto nivel y eficiente. Flink proporciona soporte para la ingestión, procesamiento y análisis de datos en tiempo real con baja latencia y alta capacidad de escalabilidad.

Referencias:

- ¿Qué es Hadoop MapReduce?: https://aprenderbigdata.com/hadoop-mapreduce
- Estudio comparativo entre Apache Flink y Apache Spark: http://sedici.unlp.edu.ar/handle/10915/126780
- Hadoop – Streaming: https://www.tutorialspoint.com/es/hadoop/hadoop_streaming.htm#:~:text=Hadoop%20streaming%20es%20una%20utilidad,mapa%20y%2Fo%20el%20reductor.
- ¿Qué es Apache Avro?: https://keepcoding.io/blog/que-es-apache-avro/#:~:text=Apache%20Avro%20es%20un%20formato,un%20bajo%20coste%20de%20tamaño.
- apache pig – Conceptos básicos del chanchito de hadoop..: https://sitiobigdata.com/2016/08/30/apache-pig-conceptos-basicos-hadoop/
- ¿Qué es Hive?: https://keepcoding.io/blog/hive/#:~:text=Hive%20o%20Apache%20Hive%20es,(Hadoop%20Data%20File%20System).