.

Tecnologías de Procesamiento

Big Data

ÍNDICE

INTRODUCCIÓN

Nuestro proyecto, enfocado en optimizar la estrategia de Trading de Broskis Brokers dentro del sector de Tecnologías de la Información del SP-500, se encuentra en pleno desarrollo bajo la metodología ágil de Scrum. Este segundo sprint marca un momento crucial en nuestro proceso, donde nos enfocamos en objetivos específicos para la integración y análisis de datos en tiempo real, así como en la implementación de herramientas clave para nuestro flujo de trabajo.

SPRINT 1 RETROSPECTIVE

Después de este primer sprint, hemos realizado una reunión al concluir nuestro deadline, hemos revisado todo lo ocurrido durante el sprint (qué se hizo mal y qué se hizo bien, y cuáles fueron las principales dificultades a las que hubo que enfrentarse). Nuestro objetivo para esta reunión fue simplemete adquirir conociemientos para mejorar en futuros sprints y proyectos.

En primer lugar, como aspectos positivos, llegamos a la fecha límite con todo terminado y cumpliendo el objetivo que nos habíamos marcado en la primera reunión; además la coordinación y la división de trabajo ha sido perfecta, nadie se ha encontrado sobrecargado en cuanto horas empleadas al proyecto se refiere.

En segundo lugar, en cuanto aspectos negativos, lo único que se ha recalcado ha sido la posible necesidad de hacer dailys más a menudo, con el objetivo de hacer reuniones más veces y más cortas, y tener más conocimiento sobre el progreso del resto de compañeros, en lugar de enterarse al final o cercanos al final del sprint.

SPRINT 2 PLANNING

Durante este sprint, uno de nuestros principales objetivos es establecer una fuente de datos en tiempo real desde el 01/01/2024 hasta la fecha actual para su análisis. Este

requisito nos lleva a utilizar técnicas de Extract, Transform, Load (ETL) para fusionar los datos históricos con los datos en tiempo real y la información de las compañías.

Para lograr una integración eficiente de los datos en tiempo real, hemos decidido implementar Kafka como parte de nuestra arquitectura. Esto implica enviar cada fila de datos al sistema y garantizar el funcionamiento del consumidor, lo que requerirá una cuidadosa configuración, incluyendo 4 particiones para la distribución de carga.

Además, en este sprint exploraremos y analizaremos Apache NiFi como una posible herramienta para la integración de datos con Kafka. Esto incluirá la configuración de un conector que vincule Apache NiFi con Kafka y un conversor para transformar todos los datos a formato JSON, facilitando su posterior envío a Elasticsearch.

Por último, para asegurar la persistencia y accesibilidad de los datos analizados, configuraremos otro conector desde Apache NiFi a Elasticsearch. Esto garantizará que los datos estén disponibles para su consulta y análisis en todo momento, contribuyendo así a nuestra estrategia de Trading de manera efectiva y eficiente.

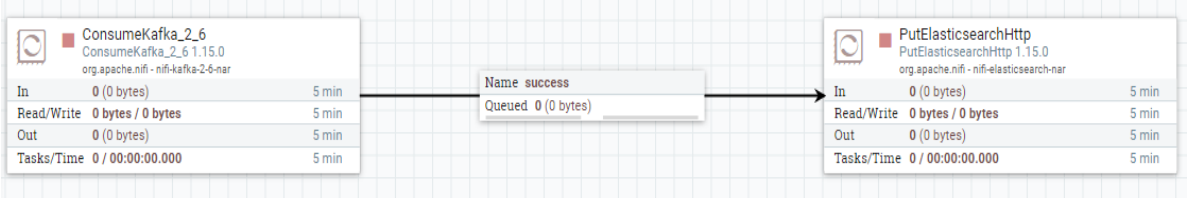
ORGANIZACIÓN DE TAREAS

Se decidió asignarle el puesto de scrum master a Mario Kroll y el de product owner a Antonio Mora. Las funcionalidades del scrum máster son: facilitar el proceso Scrum, guiando al equipo en la adopción y mejora continua de las prácticas ágiles. Sus funciones incluyen eliminar obstáculos, fomentar la colaboración y asegurar que el equipo siga los principios de Scrum para lograr entregas exitosas y eficientes. Además, se enfoca en el desarrollo del equipo y la mejora del proceso. Por otro lado, el product owner será el

encargado de definir y priorizar las características del producto, asegurando el valor empresarial. Colabora estrechamente con el equipo de desarrollo y actúa como representante del cliente para tomar decisiones que optimicen la entrega de un producto exitoso y alineado con los objetivos del negocio en la metodología ágil Scrum. Debido al éxito de este primer sprint no hemos valorado la opción de un cambio de roles, por ello se encargaron de la designación de tareas para este segundo sprint. Quedando el esquema de la suiguiente forma

TO DO	RESPONSIBLE	DONE
Kafka	Antonio Mora	✔
Nifi-Elastic Search	Mario Kroll y Nicolas Villagrán	✔
DOCUMENTACIÓN	Pablo Díaz	✔

CONEXIÓN Y WORKFLOW CON NIFI



Esta imagen nos sirve para ver la conexión que existe entre Kafka, Apache Nifi y Elastic Search. Para subir los datos a Kafka, hemos redactado una serie de scripts de Python que nos permiten establecer una comunicación entre la primera parte de esta conexión, es decir, de Kafka a Apache Nifi.

Para ello, hemos creado primero un Topic con 4 particiones y un factor de replicación de 2. Esto nos permite más espacio para almacenar el histórico de datos y poder

salvarlos en caso de fallo. Luego, hemos pasado la referencia de dicho Topic a Nifi y probamos primero a ver que leía los datos enviando un mensaje simple. Una vez asegurada la conexión, nos pusimos a investigar sobre Elastic Search y cómo conectarlo al consumidor de Kafka.

OBTENCIÓN DE DATOS A KAFKA

En el marco del proyecto desarrollado, se implementó un sistema de procesamiento de datos en tiempo real utilizando Apache Kafka, una plataforma de streaming distribuido que permite manejar flujos de datos de manera eficiente. El objetivo era procesar y analizar datos históricos almacenados en un DataFrame, enviándolos a un sistema de Kafka configurado previamente para asegurar la integración y el análisis en tiempo real de la información. La implementación se llevó a cabo en dos fases principales.

La integración de datos en Kafka medeiante un script de Python. Este script estaba diseñado para leer cada fila del DataFrame que contenía los datos históricos a procesar. Por cada fila leída, el script ejecutaba una operación de envío de datos, donde cada registro del DataFrame se convertía en un mensaje que se enviaba a un topic específico de Kafka. Este topic estaba previamente configurado con 4 particiones, lo cual permitía una distribución equitativa de los mensajes y una mayor paralelización en el procesamiento de los datos.

Cuando se enviaban datos al sistema de Kafka, se procedió a la implementación y configuración de consumidores de Kafka. Estos consumidores estaban encargados de leer los mensajes del topic al que se enviaban los datos históricos, procesándolos según los requerimientos del proyecto.

Para poder pasar los datos a Kafka, hemos tenido que hacer un pequeño reajuste de los datos de salida de nuestra clase ETL. Tras obtener el DataFrame con todos los datos de las empresas de todos los años correspondientes, veíamos que el formato de tipo Datetime no era compatible con la carga de datos en Kafka, por lo tanto, lo sustituimos por un string que contuviera la fecha en formato “%Y-%m-%d”.

El último reajuste de los datos que hicimos tenía que ver con la forma en la que Kafka produce y consume los datos. Debido a que esta plataforma trabaja con los datos en forma récords/mensajes, al pasar directamente del DataFrame a JSON, teníamos problemas de compatibilidad. Por ello, hemos hecho una reestructuración del JSON de para que Kafka pueda procesar los datos sin inconvenientes.

CONEXIÓN NIFI CON ELASTICSEARCH


```
{
  "_index" : "technology_information",
  "_id" : "gd4l040BUDDssV8DkVOn",
  "_score" : 4.179168,
  "_source" : {
    "Date" : "2018-02-13",
    "CIK" : "0000877212",
    "Symbol" : "ZBRA",
    "Open" : 115.69999694824219,
    "High" : 116.94000244140625,
    "Low" : 115.37999725341797,
    "Close" : 116.70999908447266,
    "Volume" : 199200.0,
    "Dividends" : 0.0,
    "Stock Splits" : 0.0
  }
```

```
"settings" : {
  "index" : {
    "routing" : {
      "allocation" : {
        "include" : {
          "_tier_preference" : "data_content"
        }
      }
    },
    "number_of_shards" : "1",
    "provided_name" : "technology_information",
    "creation_date" : "1708455383781",
    "number_of_replicas" : "1",
    "uuid" : "x9LSzVVMT_WCrb7do8Oi3Q",
    "version" : {
      "created" : "8500003"
    }
```


Para poder realizar la conexión con Elastic Search, como podemos ver en la imagen de Nifi del apartado anterior, hemos creado un procesador de Elastic Search para ello. Este procesador especifica el índice y la URL a la que se conectará.