GRANDATA

Ejercicio de Data Engineer

Contexto

Para desenvolverse de forma ágil en el puesto, poder manejarse con Spark y estar familiarizado con la administración de recursos de un cluster on premise Hadoop.
Se proponen una serie de ejercicios que nos permitan entender un poco más de qué manera se encararía la solución de algunos requerimientos que pueden surgir en el día a día.

Restricciones

- Hacer lo más simple que pueda funcionar.
- Escribir la mejor solución posible.

Ejercicio 1 - Spark + Docker

El dataset **eventos.csv.gz** refleja las interacciones entre los usuarios de un servicio de comunicaciones. Cada registro muestra los eventos ocurridos durante una hora entera entre un par de usuarios (origen y destino).

Columna	Descripción
id_source	ID del usuario origen (el que inicia los eventos)
id_destination	ID del usuario destino
region	Región del país donde se registaron los eventos
date	Fecha (YYYYMMDD)
hour	Hora del día (0, ..., 23)
calls	Cantidad de llamadas entre los usuarios durante esa hora
seconds	Cantidad total de segundos que duraron las llamadas realizadas durante esa hora (0 si calls=0)
sms	Cantidad de sms entre los usuarios durante esa hora

Ejemplo: si el día 20/01/2021 entre las 16:00 y las 16:59 hs

1. el ID A realizó 2 llamadas de 10 segundos cada una al ID B
2. el ID B envió 3 sms al ID A

Entonces en el dataset de eventos se observarán los siguientes registros:

1. id_source=A, id_destination=B, date=20210120, hour=16, calls=2, seconds=20, sms=0, ...
2. id_source=B, id_destination=A, date=20210120, hour=16, calls=0, seconds=0, sms=3, ...

Aquellos registros con id_source o id_destination nulo deben ser descartados.

El dataset **free_sms_destinations.csv.gz** contiene los ID de los usuarios hacia quienes se puede enviar sms de manera gratuita (destinos gratuitos).

Los sms se facturan siempre al usuario origen, de la siguiente manera:

- $0.0, si el destino es gratuito
- $1.5, si el evento se registra en las regiones 1 a 5
- $2.0, si el evento se registra en las regiones 6 a 9

Se pide:

1. Calcular el monto total que facturará el proveedor del servicio por envíos de sms.
2. Generar un dataset que contenga los ID de los 100 usuarios con mayor facturación por envío de sms y el monto total a facturar a cada uno. Además del ID, incluir el ID hasheado mediante el algoritmo MD5. Escribir el dataset en formato parquet con compresión gzip.
3. Graficar un histograma de cantidad de llamadas que se realizan por hora del día.

Requerimientos técnicos:

- Notebook: Zeppelin o Jupyter
- Framework de procesamiento: Spark v2.3
- Lenguajes de programación: Scala 2.11 o Python 3.6
- Docker y Docker Compose
- Documentación README

Adjuntar el proyecto desarrollado, detallando en el README adicionalmente el monto total del punto 1, e incluyendo el dataset generado en el punto 2 y el histograma en formato PNG del punto 3.

Ejercicio 2 - Preguntas generales

1. La empresa cuenta con un cluster on premise de Hadoop en el cual se ejecuta, tanto el data pipeline principal de los datos, como los análisis exploratorios de los equipos de Data Science y Data Engineering. Teniendo en cuenta que cada proceso compite por un número específico de recursos del cluster:
    - ¿Cómo priorizaría los procesos productivos sobre los procesos de análisis exploratorios?
    - Debido a que los procesos productivos del pipeline poseen un uso intensivo tanto de CPU como de memoria, ¿qué estrategia utilizaría para administrar su ejecución durante el día? ¿qué herramientas de scheduling conoce para tal fin?
2. Existe una tabla del Data Lake con alta transaccionalidad, que es actualizada diariamente con un gran volumen de datos. Consultas que cruzan información con esta tabla ven afectada su performance en tiempos de respuesta.
   Según su criterio, ¿cuáles serían las posibles causas de este problema? Dada la respuesta anterior, qué sugeriría para solucionarlo.
3. Imagine un clúster Hadoop de 3 nodos, con 50 GB de memoria y 12 cores por nodo. Necesita ejecutar un proceso de Spark que utilizará la mitad de los recursos del clúster, dejando la otra mitad disponible para otros jobs que se lanzarán posteriormente.
   ¿Qué configuraciones en la sesión de Spark implementaría para garantizar que la mitad del clúster esté disponible para los jobs restantes?
   Proporcione detalles sobre la asignación de recursos, configuraciones de Spark, y cualquier otra configuración relevante.

Incluir las respuestas a las preguntas dentro del README.