Coffee Shop Analysis
TP Escalabilidad

Carolina Lucia Mauro	108294	cmauro@fi.uba.ar
Esteban Frascarelli	105965	efrascarelli@fi.uba.ar
Nahuel Tomas Benitez	106841	ntbenitez@fi.uba.ar

Alcance

El proyecto Coffee Shop Analysis consiste en un sistema distribuido multi computadora que procesa datos provistos por diferentes cafeterías. Los mismos cuentan con información de las transacciones, productos incluidos dentro de cada transacción, catálogo de productos, clientes y las diferentes sucursales. Para cada cliente, se desea obtener los resultados de las 4 queries explicadas a continuación

Arquitectura

Escenarios

Esta vista también se la conoce como “casos de uso”. Los mismos describen secuencias de interacciones entre los objetos y entre procesos. A continuación detallaremos los casos de uso identificados.

Query 1 (q1)

Descripción: Dado un cliente con el dataset cargado, cuando ejecuta la consulta obtendrá la lista de **(transaction_id, monto)** de transacciones de **2024–2025**, realizadas entre **06:00–23:00**, con **monto ≥ 75**.

Query 2 (q2)

Descripción: Dado un cliente con el dataset cargado, cuando ejecuta la consulta podrá ver, para cada mes en 2024 y 2025:

- Nombre y cantidad de los productos mas vendidos
- Nombre y monto de los productos que mas ganancias han generado

Query 3 (q3)

Descripción: Dado un cliente con el dataset cargado, cuando ejecuta la consulta podrá obtener el TPV (Total Payment Value) por cada semestre en 2024 y 2025, para cada sucursal, para transacciones realizadas entre las 06:00 AM y las 11:00 PM.

Query 4 (q4)

Descripción: Dado un cliente con el dataset cargado, cuando ejecuta la consulta podrá obtener la fecha de cumpleaños de los 3 clientes que han hecho más compras durante 2024 y 2025, para cada sucursal.

Vista Física

Se detallan en esta vista todos los componentes físicos del sistema así como las conexiones físicas entre esos componentes que conforman la solución.

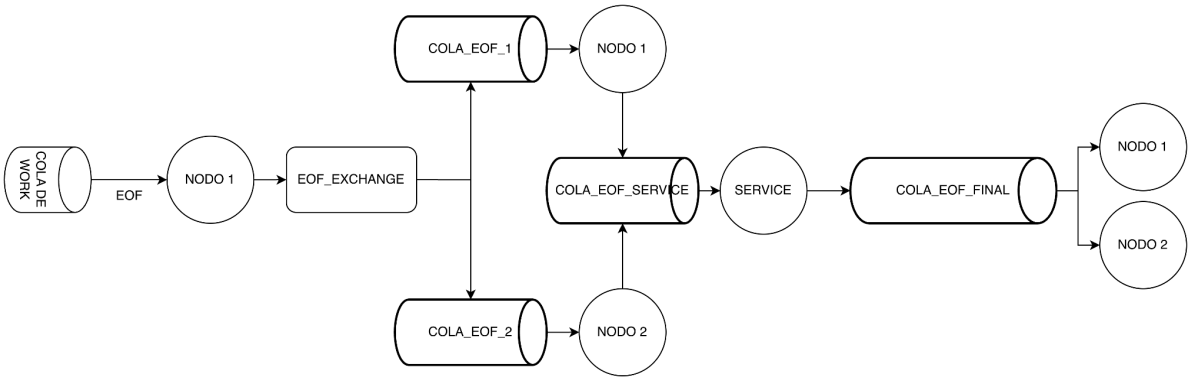
Diagrama de robustez

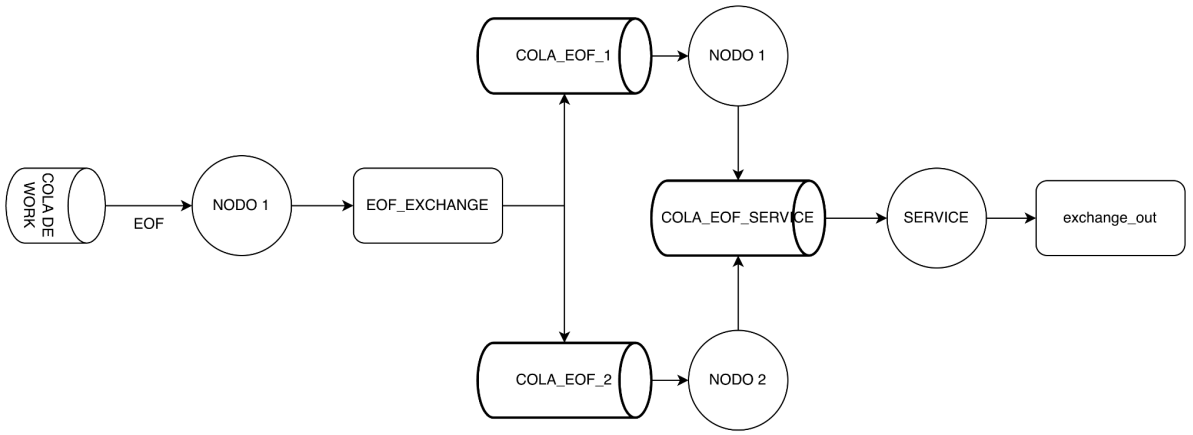
Para esta entrega los principales cambios radican en el manejo de los End of Files. Al posibilitar que se conecten muchos clientes en simultáneo, se complejiza la forma en que los nodos procesan la información. Hay dos reglas generales que aplican para describir los cambios en la mayoría de componentes: diccionarios para guardar la información por cliente dentro de los joiner o aggregators, y servicio de EOFs para manejar correctamente el fin de archivo para los nodos con réplica.

- **Diccionario de datos**: al recibir información de distintos clientes, para poder 'aplanar' los distintos datasets, hay que incluir la variable "request_id" dentro de los diccionarios de resultados, y de esta forma saber que siempre estamos haciendo la carga dentro del cliente correcto. Para lograrlo, en vez de guardar diccionarios con una variable como key, se crean diccionarios con una tupla como key, teniendo en ella los valores de la variable inicial, como producto o fecha, acompañada del request_id. De esta forma los diccionarios quedan con el formato: (variable, request_id) : valor. Por ejemplo:
  users_by_store[(store_id, message.request_id)][user_id]

- **EOF Service:** al permitir multicliente los EOF pierden confianza, ya que podríamos estar recibiendo EOF de cualquiera de los clientes y datasets, y gestionar esto entre varios nodos genera muchas discrepancias entre procesos. Para eso se creó un servicio que ayuda a gestionarlos entre nodos de una misma etapa. El EOF service se implementa como un servicio que tiene una cola de entrada de la cuál todos los nodos del stage tienen referencia. Cuando un nodo pullea el EOF lo reenvía a un exchange que tiene todas las colas de los nodos hermanos, incluyéndose a sí mismo. Todos los nodos pullearán ese EOF de la cola de EOFs propia y lo enviarán a la cola del Service. El Service sabe cuántos EOFs debería esperar, y una vez que el servicio detectó que todos los nodos recibieron el EOF, se pasa el mensaje a los siguientes nodos. Para reenviarlo hay dos técnicas dependiendo del nodo:
  Para Amount e Year filter, lo que se hace es pushear a una cola de EOF Final, de la cual cualquiera de los nodos obtendrá el EOF y lo enviará directamente a la siguiente cola.
  Para el resto de los nodos con réplica, el Servicio pushea el EOF directamente en la siguiente la cola de batches de la siguiente etapa.

Finalmente para aquellos nodos que no tengan réplicas, lo más simple es utilizar un diccionario de EOFs por cliente, de esta forma dentro de un mismo nodo sabremos qué información del diccionario enviar al siguiente nodo, diferenciandolo por request_id.


COLA_EOF_1
NODO 1
COLA DE WORK
EOF
NODO 1
EOF_EXCHANGE
COLA_EOF_SERVICE
SERVICE
COLA_EOF_FINAL
NODO 1
NODO 2
COLA_EOF_2
NODO 2


COLA_EOF_1
NODO 1
COLA DE WORK
EOF
NODO 1
EOF_EXCHANGE
COLA_EOF_SERVICE
SERVICE
exchange_out
COLA_EOF_2
NODO 2