Universidad de Buenos Aires
Facultad de Ingeniería

75.74 Sistemas distribuidos

Trabajo práctico N° 2
Reddit meme analyzer

Autor:
Matías Lafroce

21 de junio 2022

Índice

1 Reddit meme analyzer

1.1 Introducción

Reddit meme analyzer es un sistema distribuido de analisis de posts y comentarios de Reddit acerca de memes. El mismo está compuesto por un cliente, un servidor RabbitMQ, y múltiples servicios que operan sobre el stream de posts y de comentarios, transformando los datos y comunicándose mediante intercambio de mensajes:

1.2 Descripción de los servicios

Existen 3 flujos de datos: cálculo de mediana, posts de tipo "college" y mejor meme (con mejor sentiment promedio).

1.2.1. Cálculo de mediana

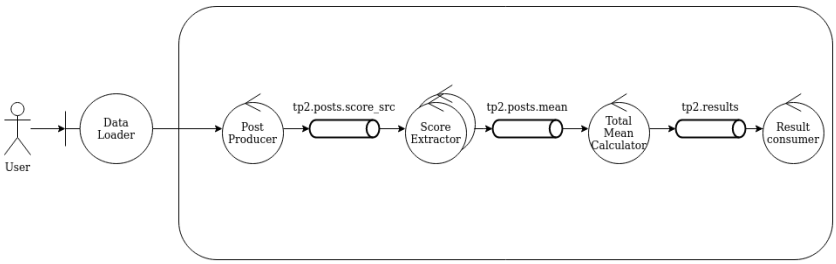

Figura 1: Diagrama de robustez

En el cálculo del score medio total, el proceso **Score extractor** los posts y encola unicamente el score correspondiente. El proceso **Total mean calculator** desencola estos scores y calcula la mediana.

1.2.2. Posts de tipo college

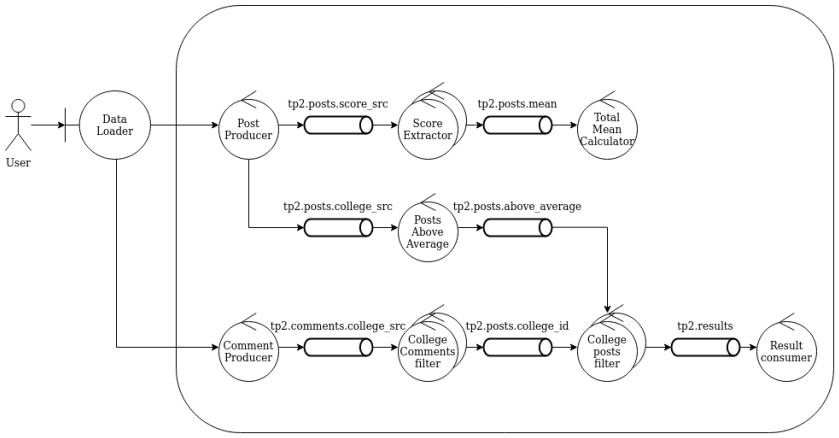

Figura 2: Diagrama de robustez

En la extraccion de posts relacionados con *college*, tenemos por un lado que **College comments filter** nos filtra los comentarios que contienen palabras relacionadas con la temática *College*, y los encola para que el filtro **College posts filter** los desencole. Cuando se termina de filtrar los *post_id* de todos los comentarios empiezo a consumir *Posts con score arriba de la mediana*. Estos son encolados por **Posts above average filter**, que funciona primero recibiendo la mediana de **Total mean calculator** y luego filtrando los posts del *post producer*.

1.2.3. Posts de tipo college

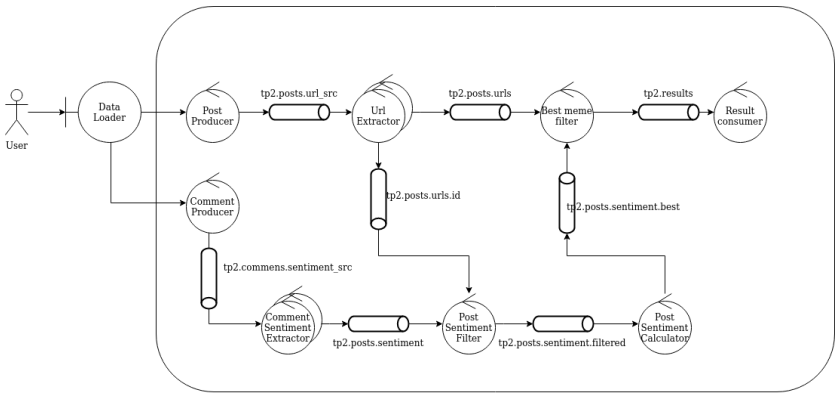

Figura 3: Diagrama de robustez

El último flujo es el de "Best meme". Para este, primero extraemos el sentiment de los comentarios con **Comment sentiment extractor** y enviamos al **Post sentiment filter**. Este filtro se encarga de filtrar los sentimientos de comentarios que pertenezcan a un post **existente**.

El **Post sentiment filter** debe primero alimentarse de los ids obtenidos del UrlExtractor, servicio que también se usa para extraer Urls de los memes que serán utilizados al final de la cadena. Una vez que se obtienen todos los ids, se pueden filtrar los comentarios inválidos.

Con la salida del *post sentiment filter* calculamos el sentiment promedio por post con **Post sentiment calculator**. Finalmente enviamos el Id del post con sentiment más alto. Este id lo usamos para filtrar los memes provenientes de **Url extractor**.

Los 3 flujos encolan el resultado en una cola de resultados para que Result consumer los baje a disco.

1.3 Modelo de comunicación

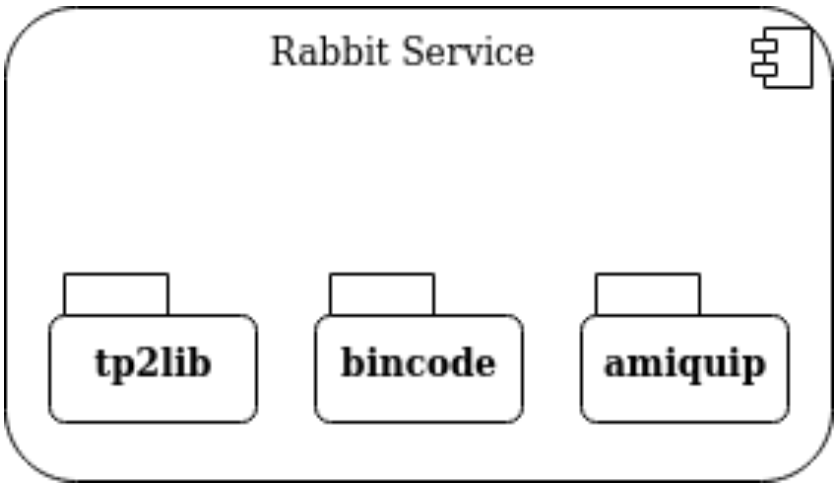

Figura 4: Diagrama de paquetes

Cada servicio está compilado junto al modelo de la aplicación (Estructuras como `Post`, `Comment`, y un enumerado compuesto `Message` con las distintas variantes de mensajes entre servicios).

Además, se hace uso de la biblioteca **bincode**, que serializa estas estructuras en código binario. Primero envía un entero de 64 bits para indicar el tipo de mensaje, y luego envía los atributos del enumerado compuesto que sean necesarios. Los strings se envían con largo preconcatenado.

Por último, para conectarse al servidor RabbitMQ se hace uso de la biblioteca **amiquip**. La misma es de caracter sincrónico, lo que facilita el manejo de errores en comparación a variantes *async*.

1.4 Carga de datos

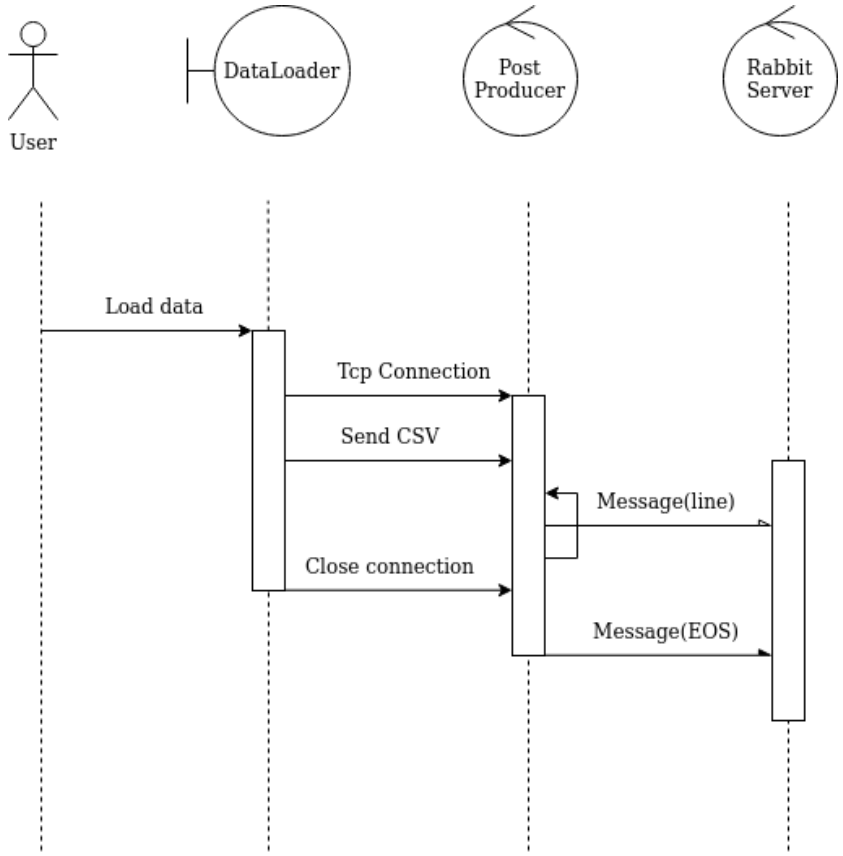

Figura 5: Diagrama de secuencia

Para cargar los datos, el cliente inicia el proceso Data loader. Este proceso se conecta mediante protocolo TCP al Post producer y envía el CSV crudo al servidor. El proceso Post producer **transforma** estas lineas de CSV en mensajes para el servidor RabbitMQ, y lee del stream hasta agotar el stream de datos. Una vez agotado el stream, se envía el mensaje `EndOfStream` Una vez enviados los posts, el proceso Data Loader se conecta con el Comment producer y realiza la misma acción con los comments.

Cabe aclara que el orden de los posts y comments es indistinto, pudiéndose hacer en paralelo.

1.5 Comunicación entre servicios

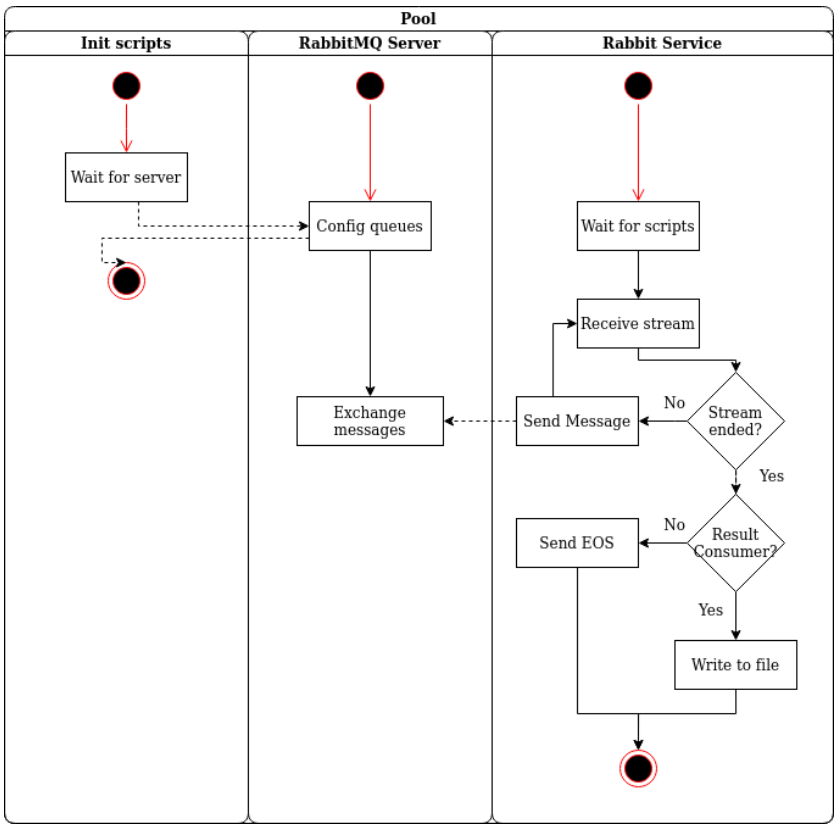

Figura 6: Diagrama de actividades

1.6 Despliegue de servicios

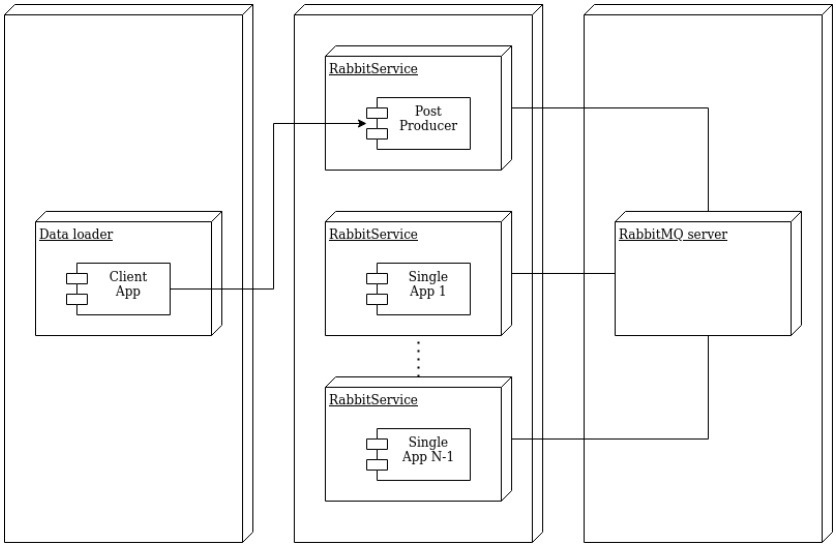

Figura 7: Diagrama de despliegue

Se propone que la aplicación data loader se ejecute en el equipo del cliente que quiere extraer métricas. Este debe poder conectarse de forma directa tanto a *PostProducer* como a *CommentProducer*, que son los únicos servicios con puertos expuestos al usuario.

Por otra parte, debe haber un servidor de RabbitMQ presente en el sistema. Todos los servicios deben poder comunicarse con este servidor de RabbitMQ.

No es necesario que los servicios estén ubicados en el mismo nodo, pero facilita la sincronización a la hora de activarlos.