

Distributed Systems, Assignment 1



Aleix Carrillo Solà

Ivan Arenal Fernández

16/06/2024

Índex

| | | |
|------|---|---|
| 1. | Abstract..... | 3 |
| 2. | System design and discusión | 4 |
| 3. | Questions..... | 5 |
| 3.1. | Q1 Are private chats persistent? If not, how could we give them persistency?..... | 5 |
| 3.2. | Q2 Are there stateful communication patterns in your system? | 5 |
| 3.3. | Q3 What kind of pattern do group chats rely on? In terms of functionality, compare transient and persistent communication in group chats using RabbitMQ. | 5 |
| 3.4. | Q4 Redis can also implement Queues and pubsub patterns. Would you prefer to use Redis than RabbitMQ for events? Could you have transient and persistent group chats? Compare both approaches..... | 5 |

1. Abstract

Esta práctica trata en desarrollar una aplicación de chat online que se centra en entender y trabajar los diferentes patrones de comunicación presentes en un sistema distribuido. Esta practica tiene diversas funcionalidades como pueden ser chats privados entre dos usuarios, chats grupales entre multitud de usuarios o el poder descubrir chats.

Para realizarla se utiliza Python y sus bibliotecas como pueden ser gRPC para las llamadas de los procedimientos remotos definidos en el servidor. Se utiliza Redis para guardar los usuarios y RabbitMQ para poder encolar los mensajes.

2. System design and discusión

En nuestra practica se ha utilizado gRPC para la comunicación entre el cliente y servidor. Redis para guardar y registrar los usuarios y RabbitMQ para encolar los mensajes.

Primeramente, se pide el nombre del usuario que va a utilizar el programa. Seguidamente se le muestra un menú con diferentes opciones a escoger, creemos que es la mejor manera de distribuir las funcionalidades. Una de las opciones es el chat privado entre dos usuarios, cuando los dos usuarios quieren hablar entre si se pide los nombres mutuamente, y se verifica que la dirección de los dos sea la correcta especificada.

Por decisión de diseño hemos considerado separar los chats grupales y el chat de insultos aun y así siendo un chat grupal más. Al seleccionar los grupales, se muestra un submenú con diversas opciones como son: crear un chat grupal, suscribirse a un chat grupal, enviar mensajes al chat grupal, y desuscribirse de un chat grupal. Primeramente, el usuario deberá crear el chat grupal en caso de que este no exista, seguidamente deberá suscribirse al chat creado para luego enviar mensajes en el. El programa antes de enviar mensajes comprobará si el usuario realmente esta suscrito al chat, en caso de que no lo este, se le impedirá el envío de los mensajes por ese grupo.

3. Questions

3.1. Q1 Are private chats persistent? If not, how could we give them persistency?

Los chats privados no son persistentes, los mensajes solo se guardan en memoria. Para conseguir que fuesen persistentes necesitaríamos almacenar los mensajes en una base de datos por ejemplo. Podríamos utilizar Redis como ejemplo.

3.2. Q2 Are there stateful communication patterns in your system?

Si, mi sistema presenta patrones de comunicación stateful. Por ejemplo, se almacena la información de los usuarios durante la sesión del chat.

3.3. Q3 What kind of pattern do group chats rely on? In terms of functionality, compare transient and persistent communication in group chats using RabbitMQ.

Mi sistema de chats grupales esta basado en el sistema de pub/sub de RabbitMQ. Cuando la comunicación es transient, una vez enviados los mensajes solo se guardan en la cola hasta que se consumen o hasta que se elimina la cola, utiliza menos recursos y es tiene mayor velocidad de la transmisión de los mensajes. Cuando la comunicación es persistente, los mensajes se guardan en disco, lo que asegura que no se pierdan incluso si se reinicia el servidor RabbitMQ. Para hacer esta implementación tendríamos que declarar la cola y los mensajes como persistentes.

3.4. Q4 Redis can also implement Queues and pubsub patterns. Would you prefer to use Redis than RabbitMQ for events? Could you have transient and persistent group chats? Compare both approaches.

Por un lado, si preferimos la velocidad podríamos usar Redis ya que es mas rápido, por otro lado si preferimos la persistencia usaríamos RabbitMQ ya que es más robusto. Supongo que en nuestro caso preferiríamos utilizar Redis porque creemos que es más fácil de programar y configurar. En ambos enfoques se podrían implementar tanto chats grupales transitorios como persistentes.

Link github: <https://github.com/IvanArFe/Online-chat-application>