



Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Implementación Chatty usando Erlang

CURSO SODX 2023/24 Q1

Khaled Malhal Abbas

Roger Hurtado Manzano

Indice

1. Chatting with buddies	3
1.1. Does this solution scale when the number of users increase?	3
1.2. What happens if the server fails?	3
1.3. Are the messages from a single client guaranteed to be delivered to any other client in the order they were issued?	3
1.4. Are the messages sent concurrently by several clients guaranteed to be delivered to any other client in the order they were issued?	4
1.5. Is it possible that a client receives a response to a message from another client before receiving the original message from a third client?	4
1.6. If a user joins or leaves the chat while the server is broadcasting a message, will he/she receive that message?	4
2. Making it robust	5
2.1. What happens if a server fails?	5
2.2. Do your answers to previous questions 3, 4 and 5 still hold in this implementation?	5
2.3. What happens if there are concurrent requests from servers to join or leave the system?	5
2.4. What are the advantages and disadvantages of this implementation regarding the previous one?	5

1. Chatting with buddies

1.1. Does this solution scale when the number of users increase?

Esta solución no escala eficientemente a medida que aumenta el número de usuarios. Es una escala proporcional a la cantidad de clientes conectados. El servidor mantiene una lista de clientes en la variable "Clients". A medida que el número de clientes aumenta, gestionar esta lista puede convertirse en un cuello de botella. Para cada mensaje de difusión, el servidor necesita iterar sobre la lista de clientes, lo que puede generar problemas de rendimiento a medida que aumenta el número de clientes.

1.2. What happens if the server fails?

Si el proceso del servidor falla todos los procesos del cliente perderán su conexión con el servidor, aunque tengan el chat abierto en el terminal, no sabrán si el servidor ha fallado o no.

1.3. Are the messages from a single client guaranteed to be delivered to any other client in the order they were issued?

En Erlang, en general, se garantiza que los mensajes enviados entre procesos se entreguen en el orden en que se emitieron, pero esta garantía no se aplica si hablamos de varios clientes. Los mensajes enviados por un solo cliente al servidor se recibirán en el orden en que se enviaron. Sin embargo, no existe esta garantía para los mensajes enviados por diferentes clientes. Los procesos de Erlang se comunican mediante el envío de mensajes, y el orden en que se reciben los mensajes depende de la cola de mensajes del proceso receptor. Puede haber intercalación entre los mensajes enviados por diferentes clientes.

1.4. Are the messages sent concurrently by several clients guaranteed to be delivered to any other client in the order they were issued?

La respuesta a esta pregunta puede encontrarse en la pregunta anterior. Como ya se ha dicho previamente, los mensajes enviados concurrentemente por diferentes clientes no están garantizados para entregarse en el orden en que se emitieron a otros clientes. Los procesos de Erlang operan de manera independiente, y el orden en que se entregan los mensajes a otros clientes puede variar según la cola de mensajes de los procesos receptores.

1.5. Is it possible that a client receives a response to a message from another client before receiving the original message from a third client?

Siguiendo con el argumento de las respuestas anteriores, es posible que un cliente reciba una respuesta a un mensaje de otro cliente antes de recibir el mensaje original de un tercer cliente. En Erlang, la entrega de mensajes es asíncrona, y no hay garantía sobre el orden relativo de los mensajes entre diferentes clientes. El orden en que un cliente receptivo procesa los mensajes puede variar y podría no coincidir con el orden en que fueron enviados por diferentes clientes.

1.6. If a user joins or leaves the chat while the server is broadcasting a message, will he/she receive that message?

En el cliente y servidor que hemos hecho, no se incluye un caso concreto en el que ocurra lo que menciona la pregunta. Por lo tanto, depende de que se produzca antes, la conexión/desconexión, o la difusión, por lo que el resultado puede variar.

2. Making it robust

2.1. What happens if a server fails?

Si un servidor falla, los otros servidores pueden continuar funcionando ya que cada servidor tendrá una lista de servidores a quien hacer broadcast de las peticiones, pero los clientes que estaban conectados a este servidor dejarán de tener conexión, puesto que no hemos implementado ningún método de relocalización para los clientes en este caso.

2.2. Do your answers to previous questions 3, 4 and 5 still hold in this implementation?

Si, siguen vigentes, ya que la mejora que hemos implementado en el servidor no afecta directamente al orden de recepción de mensajes, ni a la gestión de los mensajes en caso de una conexión o desconexión simultánea a una difusión. La diferencia es que en este caso cada servidor gestionará la cola de recepción de los mensajes de sus clientes.

2.3. What happens if there are concurrent requests from servers to join or leave the system?

Estos casos no deberían suponer un problema, la implementación realizada maneja solicitudes concurrentes de servidores para unirse o abandonar el sistema de manera segura y eficiente, por lo que asegura que se gestionan los servidores en el orden en el que se ha producido la solicitud, ya que una simultaneidad total, no es posible.

2.4. What are the advantages and disadvantages of this implementation regarding the previous one?

Esta implementación posee la capacidad de mantener viva el servicio de chat con tal de que exista un servidor en alta. No solo eso, sino que los servidores también conocen el estado de los otros servidores y quienes están de alta.

Uno de los inconvenientes que tiene esta implementación es que los clientes, al haber un servidor caído o desconectado, no tendrán acceso al servidor ni sabrán siquiera si el servidor al que estaban conectados está en alta o no.