Seminar Report: Chatty

Marc Guinovart, Carles Salvador, Pol Valls

March 4, 2019

1 Introduction

L'objectiu d'aquesta pràctica és implementar en diferents nodes la comunicació entre ells creant una aplicació senzilla de chat. Crearem dues versions:

- Un sistema basat en un servidor i diferents clients que seran els clients connectats al xat.
- Un sistema descentralitzat amb diversos servidors que permet al client connectar-se a diferents servidors.

2 Experiments

2.1 Chatting with buddies

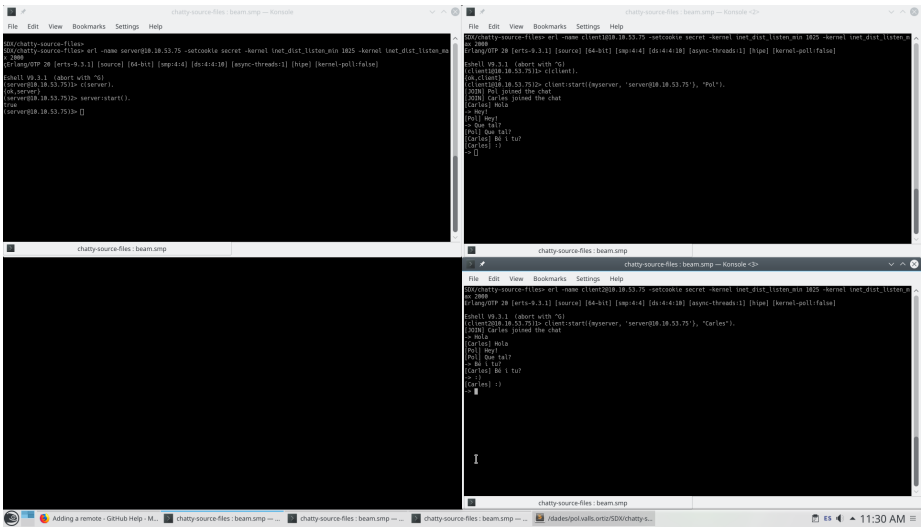
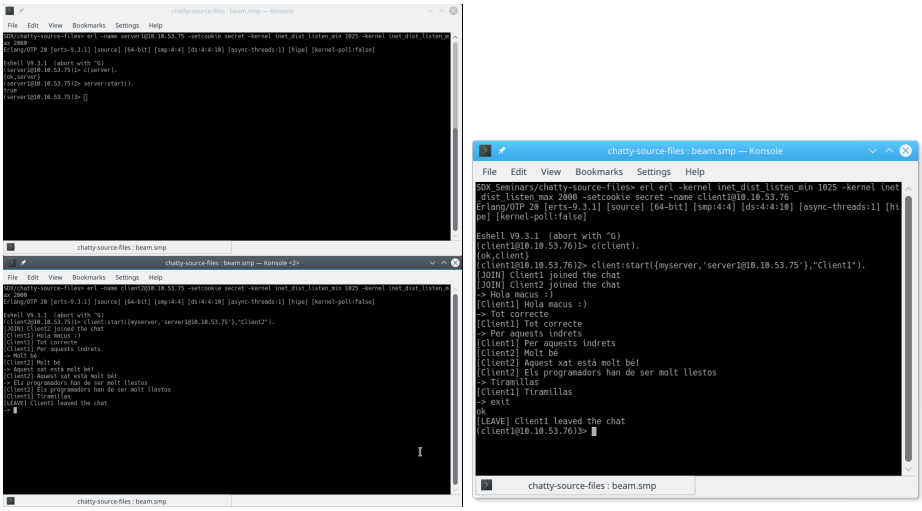


Figure 1: Comunicació de 2 clients (en la mateixa màquina) mitjançant 1 servidor

Com podeu veure a la captura de pantalla anterior, hem creat un servidor que fa d'intercomunicador entre els dos clients. Tots els clients s'han de connectar al mateix servidor i totes les comunicacions les faran mitjançant aquests, és el

servidor l'encarregat de reenviar tots els missatges als seus clients. El servidor manté un registre de tots els clients que estan connectats a ell.
Per fer-ho més complex, hem creat un client en una altre maquina connectada en local. La dificultat resideix alhora de crear els nodes, els hem de crear amb la direcció IP de cada màquina. Després al connectar els clients amb el servidor hem de posar la direcció completa del servidor (amb IP inclosa).



(a) Client i servidor en l'adreça 10.10.53.75 (b) Client en l'adreça 10.10.53.76

Figure 2: Comunicació remota entre dos clients

2.2 Making it robust

Al disseny anterior li faltava robustesa, ja que en cas que el únic servidor que teníem queia, tot el sistema deixava de funcionar. Això ho podem intentar solucionar afegint un set de servidors per tal de que si algun cau, en tinguem d'altres funcionant i enviant missatges. Tant sols hem de tocar el codi del costat del servidor, els clients no seran coneixedors d'aquest canvi.

En el cas que un servidor es desconnecti, tots els clients que estaven connectats aquests quedaran desconnectats. Com podem veure a la *Figura 5*, tenim 3 clients i 2 servidors. Client1 i Client2 es connecten al servidor 1, i Client3 es connecta al servidor 2. Els dos servidors estan connectats entre ells. Al desconnectar el servidor 2 veiem que els missatges del Client3 no arriben als altres clients, però que entre Client1 i Client2 segueix havent connexió.

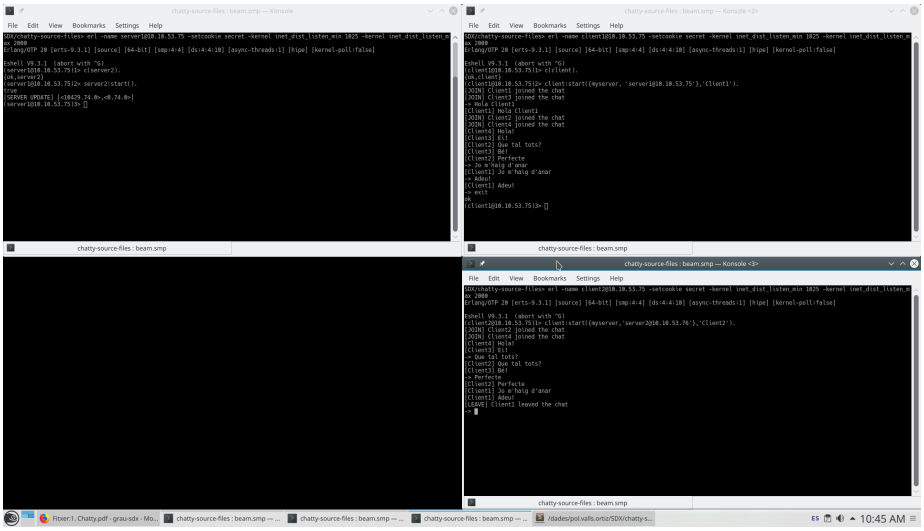


Figure 3: Client1 i Client2 en l'adreça 10.10.53.75

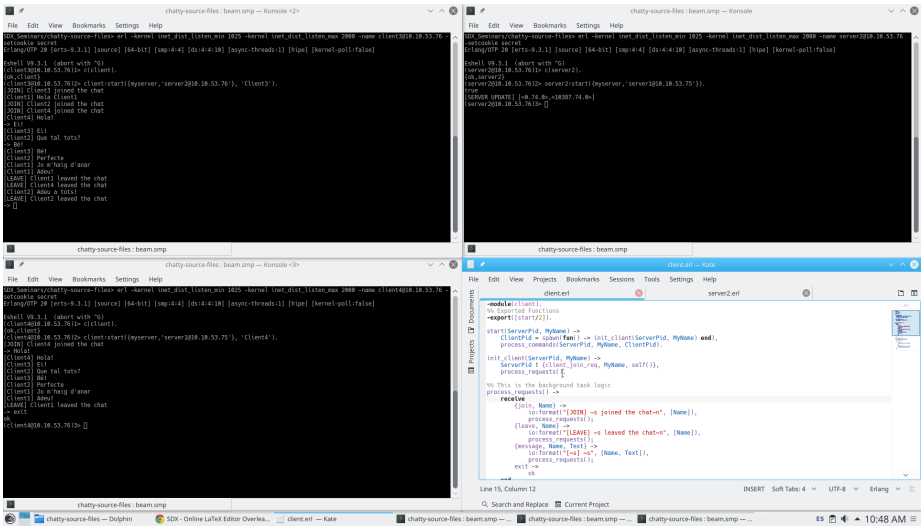
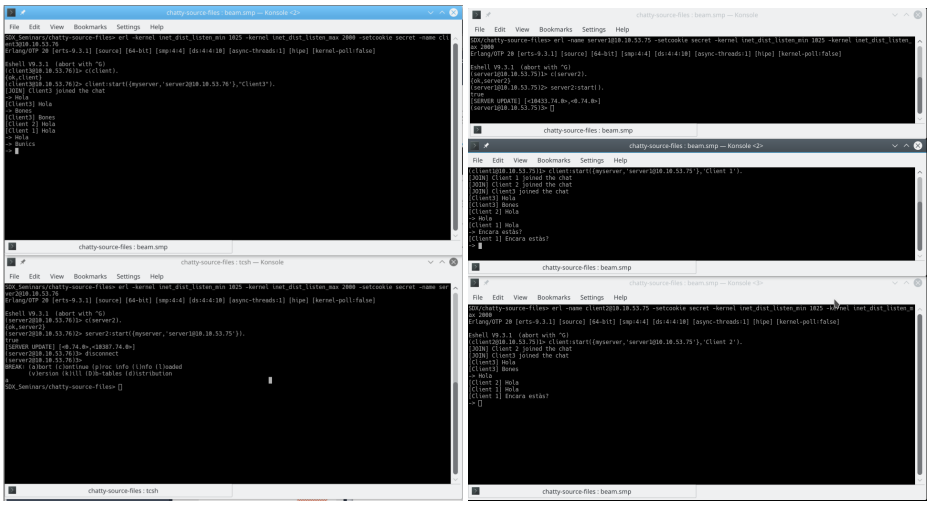


Figure 4: Client3 i Client4 en l'adreça 10.10.53.76



(a) Desconnexió server2 i incomunicació client3

(b) Veiem com Client1 i 2 no reben missatges de Client3

Figure 5: Fallada d'un servidor

3 Open questions

Try to answer all the open questions in the documentation. When possible, do experiments to support your answers.

3.1 Chatting with buddies

i) Does this solution scale when the number of users increase?

No, ja que només tenim una instància de servidor fixa, és a dir, el número de servidors no augmentarà proporcionalment a mesura que el nombre d'usuaris connectats vagi creixent.

ii) What happens if the server fails?

El que passarà serà que tots els usuaris es quedaran sense connectivitat, obligant a reiniciar totes les connexions d'aquests quan el servidor es torni a arrencar.

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

Si, ja que només hi ha un sol procés distribuint els missatges d'un client a la resta, i per tant, aquest enviarà els missatges als clients en l'ordre apropiat.

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

No està garantit, doncs dependrà de la latència de la xarxa respecte cada client connectat, ja que es pot donar que un client A envii abans un missatge que un

client B, però arribi abans al servidor el missatge de B (i per tant a tots als altres clients) degut als diferents temps de latencia entre un client i el servidor.

v) Is it possible that a client receives a response to a message (A) from another client before receiving the original message (B) from a third client?

No, ja que el servidor rebrà sempre primer B i després A, i ho enviarà a tots els seus clients en aquest ordre, i per tant el client veurà sempre primer el missatge original abans que la seva resposta.

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

Si el usuari s'uneix no el rebrà doncs a posteriori el servidor rebrà un client_join_req i l'afegirà a la llista de clients, en canvi si es desconnecta si el rebrà doncs al fer el broadcast el client si estarà a la llista de clients al que enviar el missatge i a posteriori atendrà la client_leave_req.

3.2 Making it robust

i) What happens if a server fails?

Si un servidor falla, l'únic que passarà és que els clients que estiguin conectats al mateix fallaran, i es desconectaran. Això fa que el sistema sigui més robust, ja que si falla un servidor no afecta a la totalitat del servei, sinó que només a uns quants.

ii) Do your answers to previous questions iii, iv, and v still hold in this implementation?

Per a la pregunta iii) passa exactament el mateix, depen sempre de l'ordre de recepció, a sobre s'afegeix un delay, per el temps d'intercanvi entre servidors. El mateix passa amb la pregunta iv). Passarà també el comportament descrit a la pregunta v).

iii) What might happen with the list of server if there are concurrent requests from servers to join or leave the system?

El que podria arribar a passar, és que la latència de registre dels servidors augmentaria, però amb la tecnologia de buffers que implementa Erlang, mai es perdria ninguna request.

iv) What are the advantages and disadvantages of this implementation regarding the previous one? (compare their scalability, fault tolerance, and message latency)

El primer punt, l'escalabilitat, augmenta sense dubtes. Podem anar afegint servidors alhora que augmenten els clients, i anar derivant els clients per a que es registrin de forma correcta balancejant la càrrega. En quant a la tolerància a fallades, el sistema millora també, ja que les repercusions de que un servidor deixi de funcionar són molt menors que pel primer cas. Finalment, tota millora

té un efecte col·lateral i és la latència dels missatges, la qual incrementa, ja que ha de passar per molts més nodes per arribar a tots els destinacions.

4 Personal opinion

Provide your personal opinion of the seminar, indicating whether it should be included in next year's course or not.

Aquest seminari, ens ha aportat als integrants del grup una introducció detallada del funcionament real d'Erlang. Això fa que ja adquirim una base per a poder programar sistemes més complexes. La interconexió amb la teoria fa que l'experiència amb aquest seminari sigui més enriquidora. Poder veure com realment funciona i com s'intercanvien missatges, aporta més