Seminar Report: Chatty

Víctor Sendino García, Marc Mèndez Roca, Adrián Manco Sánchez

March 6, 2018

# 1 Introduction

Aquest laboratori tenia com a objectiu implementar un sistema de xat entre dos nodes, en el qual un server actuava com a intermediari. D’altra banda, també servia com a introducció pràctica a Erlang.

# 2 Work done

Amb l’enviament s’adjunta el codi font complet de la sessió. Pel que fa a decisions de disseny, aquesta sessió no en requeria cap, ja que el codi proveït ja oferia una proposta clara i funcional del sistema.

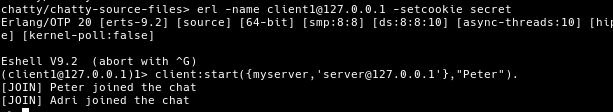
# 3 Experiments

Provide evidence of the experiments you did (e.g., use screenshots) and dis-

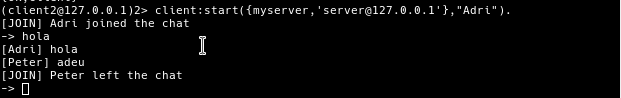
cuss the results you got. In addition, you may provide gures or tables with

experimental results of the system evaluation. For each seminar, we will

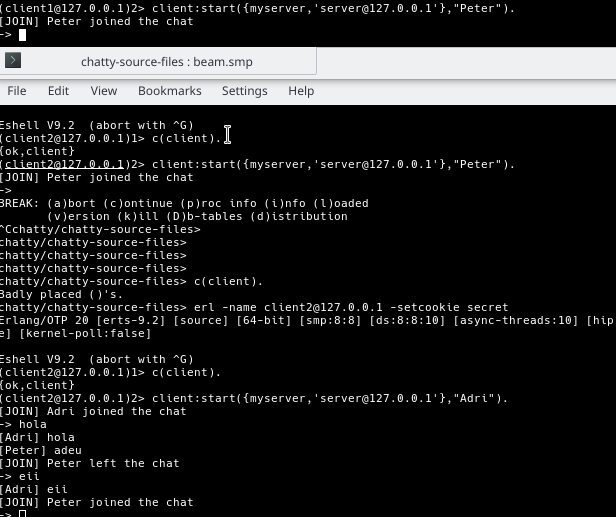
provide you with some guidance on which kind of evaluation you should do.



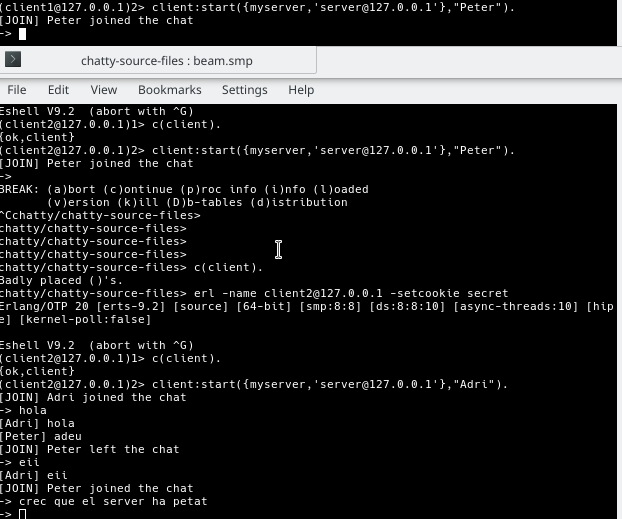
L’usuari Peter veu com ha iniciat sessió, i com l’ usuari Adri també.



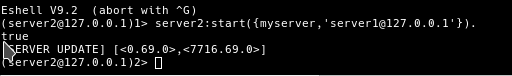
Missatges i notificació de sortida del xat.



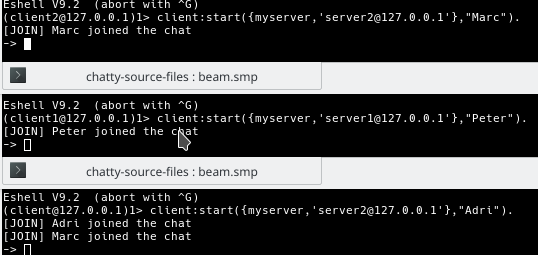
Vista de l’inici de sessió de l’usuari Peter des del seu propi terminal i des del de l’usuari Adri.



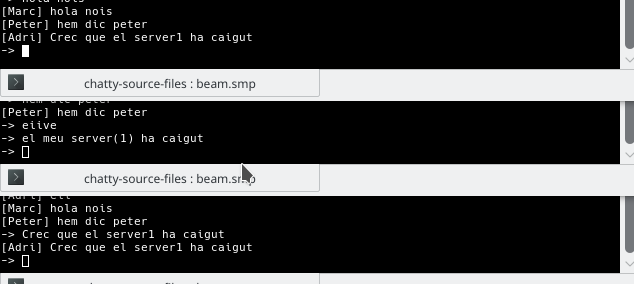
L’usuari Peter es connecta al server restant després de que l’altre hagi caigut.



Inicialització de server2.

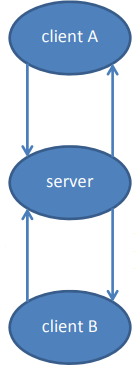


Inicialització de diversos usuaris en diversos servers.



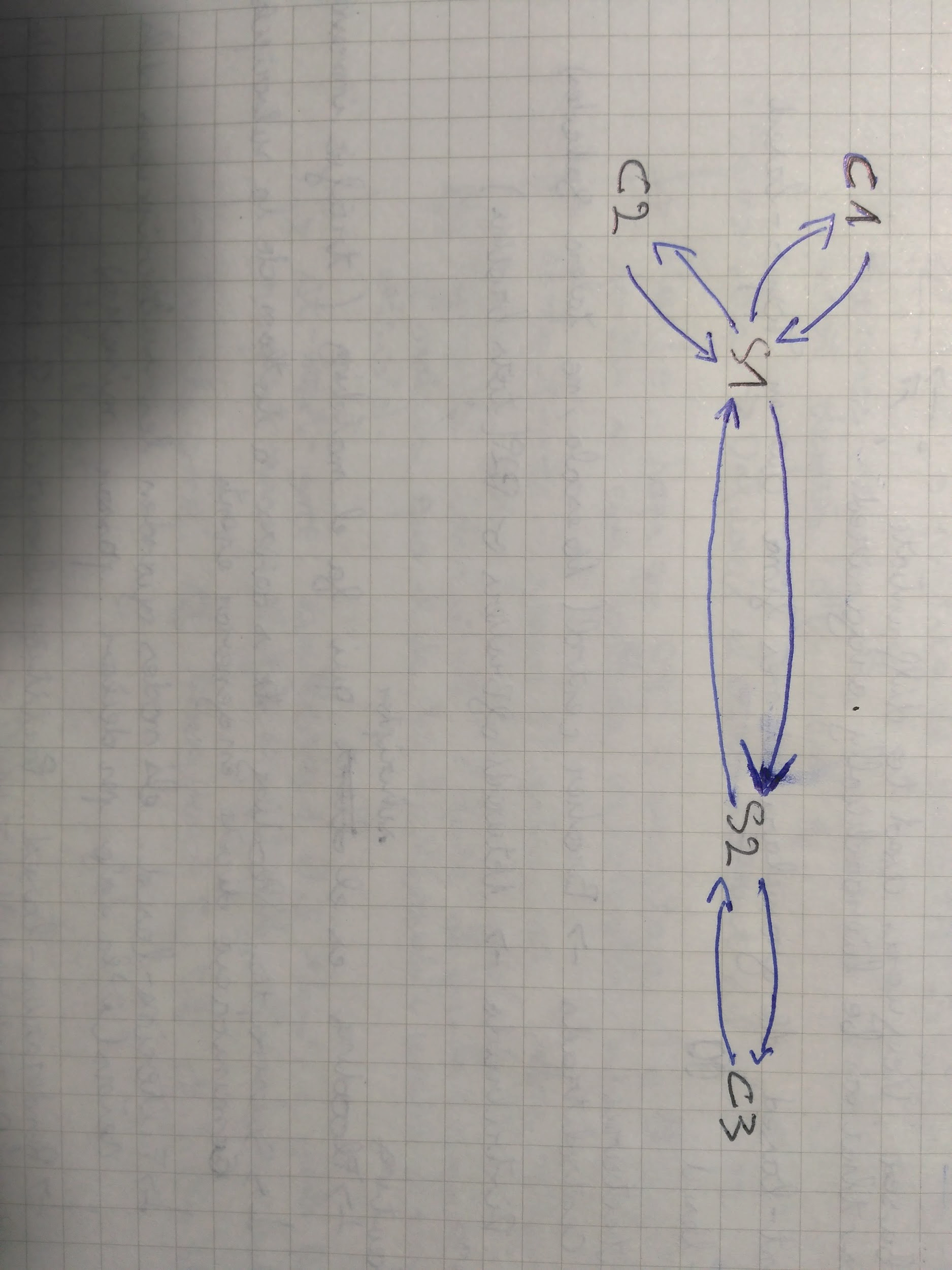
El server 1 cau, però els usuaris que eren connectats als altres servers poden seguir comunicant-se.El que estava al server que ha caigut perd tota comunicació.

# 4 Open questions

Implementació 1:

1. Does this solution scale when the number of users increase?
   1. Si el nombre d’users és massa gran, el server únic podria arribar a saturar-se, i la qualitat de servei podria reduir-se considerablement. No sembla ser una implementació realista per a un sistema distribuït de dimensions majors.
2. What happens if the server fails?
   1. Aquest sistema només té un server central, així que no ofereix cap tipus de redundància, ni cap altra opció que augmenti la resiliència del sistema. SI el server cau, el sistema sencer falla.
3. Are the messages from a single client guaranteed to be delivered to any other client in the order they were issued?
   1. Si, en aquest cas l’ordre d’arribada està garantit per com està dissenyat, ja que hi ha un únic servidor, que reenviarà als clients els missatges en l’ordre que els rebi.
4. Are the messages sent concurrently by several clients guaranteed to be delivered to any other client in the order they were issued?
   1. En situacions distribuïdes, és a dir, on més d’un procés de comunicació s’està realitzant, Erlang no pot guarantir l’ordre dels missatges, ja que això comportaria una sèrie de mecanismes i controls que el llenguatge no adopta.
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?
   1. En aquesta implementació, no seria possible que es donés aquesta situació, ja que els clients són connectats al mateix server, així que independentment de la font del missatge, seguiran camins equivalents fins arribar al destí. Com que no hi ha cap tipus de retenció client-server, els missatges haurien d’arribar en el mateix ordre en el que van ser enviats.
6. If a user joins or leaves the chat while the server is broadcasting a message, will he/she receive that message?
   1. No ho hauria de rebre, ja que el server ha de registrar prèviament els users que s’uneixen al xat, i no pot fer-ho de manera simultània al broadcast.

Implementació 2:

1. What happens if a server fails?
   1. Els usuaris connectats al server caigut deixaran de formar part del xat. Només haurien de connectar-se a l’altre server per seguir comunicant-se amb la resta de nodes.
2. Do your answers to previous questions iii, iv, and v still hold in this implementation?
   1. iii i iv)L’ordre dels missatges entre dos nodes no està garantit en un sistema amb més d’un procés com és aquest cas. v)En quant la resposta a la pregunta 5 de la implementació anterior, en aquest cas sí que es podria donar el cas proposat per l’enunciat. Per exemple en el següent cas:hi ha dos servers, S1 i S2 i tres clients C1, C2 i C3 on C1 i C2 són clients de S1 i C3 és client de S2. En el cas que C3 enviés un missatge i just després C1 enviés un altre podria passar que C2 rebés abans el de C1, perquè aquest missatge seguirà el camí C1-S1-C2 i l’altre seguiria el camí C3-S2-S1-C2, que és lleugerament més llarg i depenent de les latències podria passar el que diu l’enunciat.
3. What happens with the list of servers if there are concurrent requests from servers to join or leave the system?
   1. La llista es va modificant a mesura que rep les peticions de unir-se o marxar de la xarxa. No garanteix l’ordre ja que és un sistema amb més d’un procés. L’únic que s’hauria de tenir en compte es la carga de la xarxa ja que moltes peticions d’unir-se i marxar implica un enviament de dades a través de tots els servidors que pot arribar a saturar la xarxa.
4. What are the advantages and disadvantages of this implementation regarding the previous one? (compare their scalability, fault tolerance, and message latency).
   1. Aquesta implementació amb diversos servers és més tolerant a fallades que la primera, ja que el sistema en la seva totalitat no falla si un dels servers cau. Solament falla pels clients del server afectat. Aquest aspecte fa que també sigui més escalable, ja que els nodes es poden repartir entre els servers disponibles i fer load balancing per a estabilitzar el sistema. Quant a la latència, en el cas de que el sistema arribés a tenir un cert nombre de servers i nodes, el sistema de forwarding de missatges entre servers podria fer baixar la latència efectiva del sistema.

# 5 Personal opinion

Aquest laboratori és una bona manera d'introduir Erlang. El fet que s’hagi d’omplir forats a un codi ja proveït fa que et centris en aspectes concrets del llenguatge, i és satisfactori veure com es succeeixen els events una vegada tot funciona com ho ha de fer.