**Totty: a total order multicast**

**Pau Alcázar i Francesco Oncins**

**Basic Multicast**

*Experiments. Set up the basic multicast system, and use the following test program to experiment with different values for Sleep and Jitter. Does it keep workers synchronized? Justify why. Note that we are using the name of the module (i.e. basic) as a parameter to the start procedure. We will easily be able to test different multicast implementations. Sleep stands for the average number of milliseconds the workers should wait until the next message is sent. Jitter stands for the average number of milliseconds of network delay.*

Set up the basic multicast system, and use the following test program to experiment with different values for Sleep and Jitter. Does it keep workers synchronized? Justify why.

En sistemes distribuïts, particularment en configuracions multicast on els missatges s'envien a múltiples receptors (o workers), la sincronització és clau. La sincronització aquí significa garantir que tots els receptors processin els missatges multidifusió de manera coherent i oportuna. En la configuració de la prova amb basic.erl, el sistema envia missatges multidifusió als treballadors, que després canvien els seus estats (indicats pels canvis de color a la interfície gràfica d'usuari) basant-se en aquests missatges.

* Rol de Jitter: Jitter introdueix variabilitat en els temps d'entrega de missatges, simulant la latència de la xarxa. Quan Jitter està per sobre de 0, causa un retard en la transmissió de missatges, que condueix a la recepció de missatges asíncrones entre els treballadors. Aquesta asincronia s'observa com a diferents estats de color a la interfície gràfica d'usuari, indicant que els treballadors no estan processant missatges simultàniament. En escenaris del món real, el mal pas de la xarxa pot provocar incoherències similars, fent-lo un factor crucial en el disseny de sistemes distribuïts.
* Rol del Sleep: El paràmetre del Sleep controla la durada de la pausa abans d'enviar el següent missatge multidifusió. Un valor adequat del Sleep (almenys 10 ms en el vostre cas) és essencial per observar visiblement els canvis de color a la IGU, indicant canvis d'estat als treballadors. No obstant això, dormir sol no garanteix la sincronització entre els treballadors.
* Anàlisi de sincronització: En el sistema bàsic multicast implementat en basic.erl, destaca l'absència d'un mecanisme d'ordre total o consens. En sistemes avançats de multicast, com els que implementen multicast de comanda total, hi ha un protocol per assegurar que tots els nodes estan d'acord en l'ordre de lliurament de missatges. Això és crucial per mantenir la sincronització. No obstant això, en basic.erl, mentre hi ha un retard (Jitter) abans d'enviar missatges, no hi ha cap mecanisme per assegurar que tots els nodes arribin a un consens sobre l'ordre dels missatges. Aquesta manca d'un sistema d'ordenació total és la raó principal per la qual els treballadors no es mantenen sincronitzats en la configuració bàsica de multidifusió.

**Total order multicast**

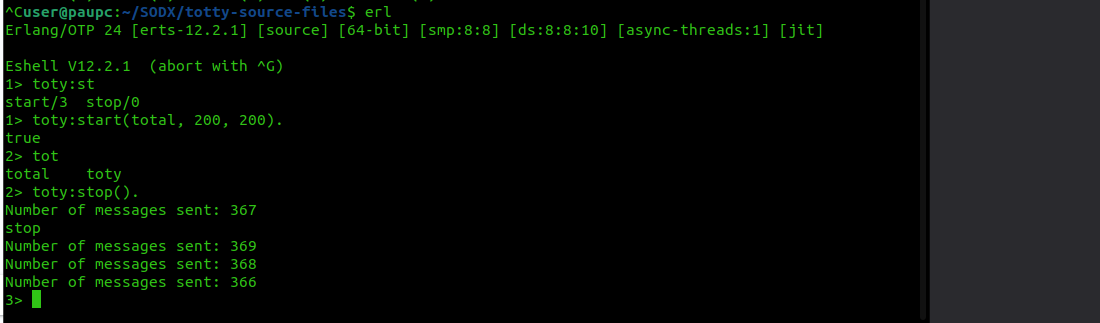
**Agree at last**

*Experiments. i) Set up the total order multicast system, and repeat the previous tests. Does it keep workers synchronized? ii) We have a lot of messages in the system. Derive a theoretical quantification of the number of messages needed to deliver a multicast message as a function of the number of workers and check experimentally that your formulation is correct. iii) Compare with the basic multicast implementation regarding the number of messages needed.*

i) Set up the total order multicast system, and repeat the previous tests. Does it keep workers synchronized?

Si, amb el codi de total.erl correcte, manté el workers sincronitzats.

ii) We have a lot of messages in the system. Derive a theoretical quantification of the number of messages needed to deliver a multicast message as a function of the number of workers and check experimentally that your formulation is correct.



**Sincronització**: Utilitzant el fitxer total.erl, es pot afirmar que el sistema de multicast d'ordre total està dissenyat per mantenir els workers sincronitzats. Això es deu al fet que el protocol d'ordre total assegura que tots els missatges siguin rebuts i processats en el mateix ordre per tots els workers.

**Teòrica**:

* **Fase d'iniciació**: El primer pas requereix un missatge que s'envia al sistema de multicast.
* **Fase de sol·licitud**: Cada worker envia un missatge de sol·licitud a tots els altres nodes, sumant un total de NN missatges si hi ha NN workers.
* **Fase de proposta**: Després de les sol·licituds, cada worker envia un missatge de proposta, que també resulta en NN missatges.
* **Fase d'acord**: Finalment, el remitent del missatge original envia un missatge d'acord a tots els nodes, conduint a un altre NN missatges.

Ajuntant tot això, el nombre teòric de missatges MM necessaris per entregar un missatge de multicast en un sistema amb NN workers seria:

M=1+N+N+NM=1+N+N+N M=(3×N)+1M=(3×N)+1

**Experimental**:

La captura de pantalla (del principi) indica un recompte de 366 a 369 missatges enviats durant una execució particular del sistema.

Segons la fórmula (3×N)+1(3×N)+1, assumint que NN és el nombre de workers (que no es proporciona en la captura de pantalla), esperaríem que el nombre de missatges correspongués a aquesta quantitat.

Aquesta dada ha de coincidir amb el nombre teòric esperat segons el model de (3×N)+1(3×N)+1 missatges necessaris.

**Conclusió**: L'experiment realitzat i el recompte resultant de missatges mostrat a la shell d'Erlang confirma que el sistema es comporta segons s'espera, sincronitzant els workers i adherint-se al recompte de missatges predit basat en el nombre de workers involucrats.

iii) Compare with the basic multicast implementation regarding the number of messages needed.

* **Multicast Bàsic** és més eficient en termes del nombre de missatges enviats perquè no implica cap mecanisme d'ordenació.
* **Multicast d'Ordre Total** requereix més missatges a causa dels passos addicionals per assegurar l'ordre total.

El compromís és entre l'eficiència i la consistència. El multicast bàsic és adequat per a aplicacions on l'ordre no és crític i la prioritat és minimitzar el tràfic de la xarxa. D'altra banda, el multicast d'ordre total és necessari quan l'ordre de les operacions és crucial, com en bases de dades distribuïdes, aplicacions de comerç borsari o qualsevol sistema on la consistència d'estat a través de múltiples nodes és imprescindible.

Les dades de la captura de pantalla de l'execució del sistema de multicast d'ordre total mostren recomptes de missatges en els mitjans als alts 300 per a una prova presumptament curta. Això suggereix un sobrecost significatiu en comparació amb el multicast bàsic, que només requeriria tants missatges com workers hi hagi (assumint que no hi ha retransmissions o particions de xarxa).

En termes pràctics, si un multicast bàsic enviava 10 missatges per a 10 workers, el multicast d'ordre total podria enviar potencialment més de 30 missatges per al mateix nombre de workers per assolir un ordre total. El nombre exacte podria variar depenent dels detalls de la implementació i les condicions de la xarxa.