

SODX LAB2 - Preguntes

Ixent Cornella, Eric Gonzalez

INTRODUCCIÓ

El Laboratori 2 consisteix en implementar un sistema d'exclusió mútua per a sistemes distribuïts.

1. THE ARCHITECTURE

i) Make tests with different Sleep and Work parameters to analyze how this lock implementation responds to different contention degrees.

Hem provat diversos paràmetres per a fer una comparació entre ells, i els hem anat apuntant en la taula que es pot observar tot seguit

VALORS {SLEEP, WORK}	1000/3000	500/200	300/2000	1000/200	500/100	400/300	750/100	750/200	850/200
W1 (ms)	3626,4	98,11	2998,06	27,61	108,56	264,66	164,87	65,18	61,1
W2 (ms)	3982,6	113,8	2929,25	123,9	191,44	260,09	165,08	64,02	77,04
W3 (ms)	3648,9	98,17	3054,75	140,63	233,37	268,57	15,75	71,95	64,39
W4 (ms)	3829,1	113,09	2731	51,7	24,63	266,87	16,1	81,32	66,7
Whitdrawals	0	0	2	2	6	0	4	0	0
Average (ms)	3771,75	105,7925	2928,265	85,96	139,5	265,0475	90,45	70,6175	67,3075
Efficiency	3771,75	105,7925	8784,795	257,88	976,5	265,0475	452,25	70,6175	67,3075

ii) Split the muty module and make the needed adaptations to enable each worker-lock pair to run in different machines (that is, john and l1 should run in a machine, ringo and l2 in another, and so on). Remember how names registered in remote nodes are referred and how Erlang runtime should be started to run distributed programs.

El que hem fet és reutilitzar el codi ja donat, però ara adaptant-lo per a que es pugui executar en nodes diferents. Ho aconseguim mitjançant el mòdul “global”, que ens

permet registrar processos i enviar missatges en nodes remots. D'aquesta manera, el codi resultant és d'aquest estil

```
start(Lock, Sleep, Work, Node1, Node2, Node3, Node4)
```

Com podem observar, és igual que el codi donat però ara necessitem els nodes amb els que treballarem, per poder registrar-los i enviar-lis missatges (més al codi entregat). Cal iniciar la consola d'erl amb l'opció `-boot start_sasl` per a poder tenir un programa distribuït.

Open Question) What is the behavior of the lock when you increase the risk of a conflict?

Quan s'incrementa el risc a un conflicte, com el lock, no té un identificador propi (se li donarà en el següent punt lock2), no es té certesa de quina instància de lock anirà tot seguit, per tant es fa aleatòriament. Aquest problema causa que el lock es comporti de manera imprevisible quan hi ha més possibilitat de conflicte.

2. RESOLVING DEADLOCK

Aquesta versió del lock no ens ha sortit del tot, però el que hem intentat és que sempre s'envii el ID del lock per a poder mantenir coherència entre workers i locks.

OPEN QUESTION:

i) Justify how your code guarantees that only one process is in the critical section at any time.

Només hi haurà un procés, i aquest és el que hagi aconseguit l'accés al lock. Amb l'ID del lock, podem garantir que no es repetirà el lock entre workers.

ii) What is the main drawback of lock2 implementation?

El principal inconvenient de la millora que s'ha realitzat al lock (lock2) pot ser que en aquest cas s'hagin que s'ha atorgat identificadors a cada lock, ara si una instància de lock està en l'estat d'espera, llavors no enviarà un "ok" al worker demanat pel lock fins que aquest hagi acabat el seu període de work.