

TD algo répartis - Nicolas ollinger

Alexandre Masson

14 Janvier 2013

Table des matières

1	TD1	3
1.1	Election sur anneau	3
1.1.1	Exercice 1	3
1.1.2	Exercice 2 : sur un anneau de taille n	4

1 TD1

1.1 Election sur anneau

Compositon d'un noeud

- state : état du processus
- start : que faire à l'initialisation
- message : Comment à partir de l'état des processus pour envoyer des message, on utilise aussi \perp pour ne pas envoyer de message ou message vide.
- trans : calcul à partir des messages reçus

On rappelle que tous les processus sont indistinguables (même code dedans) mais on connait quand même leur UID(identifiant unique par processus). ET les UID sont toujours comparables.

State : Variables :

- U : uid du processus
- send : message a envoyer, UID ou \perp
- status : leader ou ?

Start : send = u ; |

Message : transmettre send à son voisin si send $\neq \perp$

trans : v = message reçu

si v $\neq \perp$

- case :
- v > u : send := v ;
- v = u : status = leader ;
- endcase

1.1.1 Exercice 1

ajouter l'arrêt des noeuds maintenant :

State : Variables :

- U : uid du processus
- send : message a envoyer, UID ou \perp
- status : leader ou ? ou sbire
- chief : UID ou ?

Start : send = u ; chief = status = ? |

Message : transmettre send à son voisin si $\text{send} \neq \perp$: electionUID U reçu
pUID

trans : v = message reçu

si $v \neq \perp$

- case :
- $v = \text{electionw}$ et $w > u$: $\text{send} := v$; // ici on send election de W du coup.
- $v = u$: $\text{statut} = \text{leader}$; $\text{send} := \text{reçu } u$;
- $v = \text{reçu } u$; arrêt ; $v = \text{reçu } w$ (sous entendu $w \neq u$) ; $\text{chef} = w$, $\text{status} = \text{sbire}$
- endcase

prouvons l'invariant de boucle au temps $2n$ tous les processus sont arrêtés
et

$\text{stratus maxUID} = \text{chef}$

$\text{statusAUTRE} = \text{sbire}$

$\text{chef} = \text{maxUID}$.

Invariant : au temps $1 \leq t \leq n$, le processus $\text{maxUID} + t$ reçoit election maxuid.
au temps $n-1 \leq t \leq 2n$, le processus $\text{maxuid} + t$ reçoit reçuMaxuid

1.1.2 Exercice 2 : sur un anneau de taille n

nombre total de message des n! scénaris : $n! * n$: pour le leader

$\alpha(n) = n(\text{procs}) * n!(\text{scénaris}) * (1 + 1/2 + 1/3 + \dots + 1/n)$ (somme des probabilité que
les procs n soit le plus grand parmi ces K voisins, pour k de 1 à n)

$\text{nbMessage} : \alpha(n)/n! = n * \sum 1/k = \Theta(n \ln(n))$

Construisons un exemple nprocs, avec uid de 0 à n-1 tels que LCR \rightarrow
 $\Theta(n \ln(n))$ messages.

ou plutot $2\exp(n)$ procesuus et $\Theta(n 2^{\exp(n)})$ messages avec l'UI $2^n - 1$