



ARA

Projet Peersim

Robin Blottiere-Mayo

Pierre Oumeddour

Professeur : Jonathan Lejeune

26 janvier 2019

Exercice 1

1.1 Question 1 :

L'algorithme de verrouillage utilisé dans la classe Application est un algorithme dérivé de celui de Naimi-Tréhel utilisant la notion de jeton.

Ainsi tous les sites se partagent le même jeton et seul le site possédant le jeton peut entrer en section critique. Dans la classe Application, les constantes `initial_owner` et `nil` servent à discriminer le site possédant le jeton des autres.

Chaque noeud maintient sa propre liste de noeuds en attente de section critique (variable `next`) et son compteur global de sections critiques (variable `global_counter`). Il possède également l'adresse du noeud dont il pense qu'il possède le jeton (variable `last`) ainsi que le nombre de sections critiques qu'il a lui-même effectuées (variable `nb_cs`). Si un noeud veut une section critique, il envoie une demande pour avoir le jeton puis il se met en attente.

Quand un noeud sort de section critique, il envoie le jeton au dernier noeud de la liste des noeuds demandant la section critique avec les informations qu'il possède du nombre de sections critiques réalisées et des noeuds en attente de section critique.

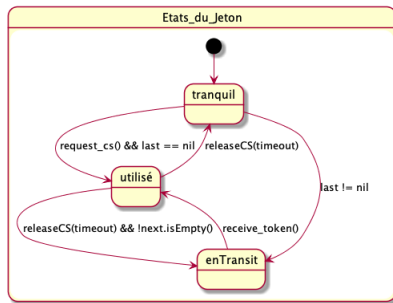
Lorsqu'un noeud reçoit le jeton, il met à jour son compteur de sections critiques avec la valeur envoyée par le dernier noeud ayant été en section critique. Il compare également sa liste de noeuds en attente avec celle qu'il a reçue. Si des noeuds se trouvent dans les deux listes mais pas dans le même ordre, l'ordre envoyé par le dernier noeud en section critique est maintenu. Si des demandes sont absentes de la liste reçue, elles sont ajoutées en fin de liste.

1.2 Question 2 :

On considère trois états du jeton :

1. Un noeud possède le jeton mais ne s'en sert pas (état tranquille).
2. Un noeud possède le jeton et est en section critique (état utilisé).
3. Le jeton est en transit (état enTransit).

tranquille → utilisé : la section critique est demandée et personne ne requiert le jeton.
 tranquille → enTransit : un noeud requiert le jeton.
 utilisé → tranquille : la section critique se termine.
 utilisé → enTransit : `releaseCS(timeout) & !next.isEmpty()`
 enTransit → utilisé : `receive_token()`



1.3 Question 3 :

α étant le temps moyen qu'un processus passe en section critique et β le temps passé entre les sections critiques.

Le ratio $\rho = \alpha / \beta$ représente la charge du réseau.

Plus ce ratio est élevé plus le réseau est ralenti.

1.4 Question 4 :

Messages applicatifs :

Sur cette première courbe, on observe deux valeurs qui varient en fonction de la charge du réseau ρ . Il s'agit du nombre moyen de messages applicatifs par section critique. On différencie le nombre de messages requérants le jeton et le nombre de transmissions du jeton entre les noeuds.

On observe que plus ρ augmente, plus le nombre de messages applicatif diminue. A la fin, il est quasiment de un message et un envoie de jeton par sections critiques. En effet, plus le temps passé en section critique est grand par rapport au temps d'attente plus les noeuds savent qui est celui possédant le jeton.

Cela s'explique par le fait qu'un noeud ayant fini sa section critique va s'endormir pour un temps qui sera court étant donné les paramètres de l'expérience qui incluent un temps de transmission et un temps de repot court entre les sections critiques. En se réveillant, le

noeud enverra une requête de section critique au noeud dont il pense qu'il possède le jeton. Ce noeud aura en effet toujours le jeton en sa possession car il n'aura probablement pas eu le temps de terminer sa section critique. Ainsi, une seul requête aura suffi au noeud pour être repertorié comme demandant l'accès à la section critique. C'est pourquoi on observe quasiment un envoie de requête par section critique.

Cette explication est valable pour un temps d'expérience suffisamment grand pour que les noeuds aient le temps de s'ordonner mutuellement.

Temps passé dans l'état requesting :

Sur cette courbe, on observe le temps moyen qu'un noeud passe dans l'état requesting en fonction de la charge du réseau.

On peut voir que lorsque que le temps β passé entre les sections critiques est beaucoup plus grand que le temps α passé en section critique, le temps moyen par sections critique est très faible. Lorsque ρ s'approche de un, le temps passé en section critique augmente très rapidement. Après cela, l'augmentation redevient proche voir inférieur à un sur une échelle logarithmique.

Cette brusque augmentation peut s'expliquer par le fait qu'un temps α très court peut faire qu'à un certain moment, aucun des cinquante noeud du réseau ne requiert la section critique. Lorsque la charge du réseau sera suffisamment grande pour qu'il y ait en permanence un noeud demandant l'accès à la section critique, le temps passé dans l'état requesting augmentera linéairement en fonction du temps ou le jeton est utilisé par un processus en section critique.

Utilisation du jeton :

Sur cette troisième courbe, on observe le temps que le jeton passe dans chacun des états décrits à la question 2 en fonction de la charge du réseau. L'expérience s'appuyant sur un temps de transport négligable, le temps que le jeton passe dans l'état transmission est proche de 1 quelque soit la charge du réseau.

Les temps que le jeton passe en étant utilisé ou inutilisé suivent des courbes inverse. On observe que lorsque $\rho = 1$, $\alpha = \beta$.

1.5 Question 5 :

Le fait que le temps de transmission moyen soit supérieur au temps moyen passé en section critique augmente la charge du réseau. Le temps de transport augmentant, le temps

ou le jeton est posséd   par un noeud est plus faible.

Cela ne change pas fondamentalement les r  sultats en dehors de rendre le syst  me plus lent et de diminuer le nombre de section critique pour un temps d'ex  cution   gale.

Dans le d  tail, on voit que les premiers instants du fonctionnement du syst  me le noeud ayant le jeton peut ex  cuter plusieurs fois sa section critique avant de recevoir une requ  te et que la file des noeuds demandant l'acc  s    la section critique mettra plus de temps    se remplir. Cependant apr  s que tous les noeuds aient ex  cut   au moins une fois leur section critique, le temps pass   en section critique et le temps de transmission du message s'additionnant rendront la suite de l'ex  cution semblable    ce qu'elle aurait   t   avec un temps de transmission moyen plus faible que le temps pass   en section critique.

Exercise 2

Exercise 3