Université Jopseph Fourier – UFR-IMA
MST ESI2 : Examen d'algorithmique distribuée
Durée 3h

Tous documents autorisés – Le barême est indicatif

23 Avril 1999

1 Cliché sur anneau(7pts)

Il s'agit de concevoir un algorithme de prise de cliché sur un anneau où les sites sont reliés par des liaisons FIFO fiables monodirectionnelles. Tout site peut émettre sans contrainte sur sa liaison de sortie (pas de jeton d'émission). Tout message porte le numéro du site destinataire. Chaque site est donc responsable du routage sur l'anneau. D'où le comportement suivant :

```
repeter
      lin?<MSG, dest......> && dest!=ICI ==> lout!<MSG, dest......>
      lin?<MSG, dest......> && dest==ICI ==>  traitement local....
```

Vagues sur un anneau Compléter le comportement des sites pour gérer des vagues isues de n'importe quel initiateur. One conserve que la dernière vague par initiateur.

Cliché sur un anneau Dans un premier temps, on ne considère que le cas où un seul site (par exemple le site 0) peut initier une vague.

1. Que doit enregistrer l'initiateur après émission de la commande *cliché* ?
2. Que doit enregistrer un site relayant la commande sur l'anneau ? En quoi consiste le relai de commande ?
3. Compléter le comportement des sites pour gérer des vagues isues de l'initiateur 0 .

Etat global Par rapport à un graphe quelconque, une structure d'anneau à liaison FIFO induit sur les communications un ordre plus fort. Cela permet-t-il de garantir que le cliché calculé est un état atteint par le système (et non plus simplement un état atteignable).

Justifier votre réponse par un exemple de système où deux sites reliés en anneau exécutent chacun un processus client. De façon itérative, chaque client émet vers son voisin puis reçoit de son voisin

1. développez la partie du graphe des états accessibles nécessaires à votre exemple,
2. donnez un entrelacement de l'exécution des deux clients et de la prise du cliché justifiant votre point de vue.

Cliches multiples Tout site peut initier une prise personnelle de cliché à tout moment et sans attendre la fin de la prise précédente. Seul la dernière demande d'un site est valide. Compléter le comportement des sites pour gérer des demandes de cliché issues de n'importe quel initiateur.

2 Cliché sur échéance (7 pts)

Le problème posé consiste à définir un algorithme qui calcule l'état global à une date **D0** donnée **connue** de tous les sites. Chaque site prendra la partie du cliché (état local et état de liaisons) qui le concerne de façon à ce que l'ensemble de ces parties constitue un état cohérent. On ne s'intéressera pas à collecter ces états partiels sur un site particulier.

Le système distribué dispose donc d'une horloge globale consultable par tous les sites (procédure `int Date()`). Sa finesse est suffisante pour dater uniquement les transitions d'état des processus. Pour prélever un cliché de l'état du système à une date **D0** connue de tous les sites, tous les sites enregistrent une partie du cliché à la date **D0**.

Comme on ne s'intérese pas à la structure de données décrivant l'état global, on se donne les procédures d'enregistrement suivantes :

- `RegisterNodeState (Snapshot)` sauve l'état interne du site dans le cliché `Snapshot` ,
- `RegisterLinkState(Snapshot, LinkState[])` enregiste dans le cliché `Snapshot` le tableau `LinkState[]`. Ce tableau devra donc contenir l'état (calculé par votre algorithme) des liaisons au moment du cliché. C'est un tableau des listes (initialement vides) de messages qui circulaient sur chacune des **L** liaisons du site au moment du cliché.

Liaisons fiables avec déséquencement On suppose que toutes les liaisons de communication sont fiables mais peuvent déséquencer les messages.

1. Rappeler sous quelle condition un message reçu sur site doit être enregistré dans le cliché.(4 lignes)
2. Comment un site détecte-t-il que l'enregistrement de l'état d'une liaison est terminé ?(4 lignes)
3. Quelle informations supplémentaires sur les communications un site doit il gérer ? (4 lignes)
4. Que doit contenir un message circulant sur une liaison ? (2 lignes)
5. Compléter le code suivant décrivant le comportement d'un site à la réception d'un message sur une liaison :

```
repeter
    Quelque soit l de L  l?<......> && Date()<D0 ==> .......
    Quelque soit l de L  l?<......> && Date()>D0 ==> .......
                     Date()==D0                  ==>  .....
```

3 Exclusion mutuelle(7pts)

L'algorithme d'exclusion mutuelle sur un réseau complètement maillé (Clique) de Ricart et Agrawala vu en cours est basé sur la synchronisation de vague par horloge logique. C'est un algorithme où un site demande (REQ) aux autres l'autorisation d'accès(ACK) à la section critique.

```
répeter
    arrivée  LOCK   && état== dehors  ==> état = requete_en_cours;........
    arrivée UNLOCK  && état== dedans  ==> état = dehors;........
    Quelque soit l de L  l?<REQ,...> ==> .......
    Quelque soit l de L  l?<ACK,...> ==> .......
    état = requete_en_cours && ....  ==>  .......
```

1. Compléter le comportement précédent des sites en supposant (pour simplifier) qu'un site ne peut demander qu'un accès exclusif à la fois et que toute demande ultérieure doit être faite après la fin de l'accès en cours. On ne précisera pas les interactions avec les processus clients d'un site (arrivée LOCK ou UNLOCK).
2. Le recalage des horloges logiques est fait à la réception d'une requête par `HL=max(HL, date reçue avec la requete)`. Une horloge locale de site peut être incrémentée soit avant soit après l'émission des demandes d'autorisation aux autres sites. Dans ces deux cas, quel est l'intervalle dans lequel peuvent se distribuer les numéros de requêtes en conflit ? Justifier par des traces temporelles (3 sites sont suffisants).
3. quel est la meilleure solution ? pourquoi ? (5 lignes)

Amélioration Un inconvénient de cet algorithme est que les permissions d'accès sont redemandées à tous les autres sites à chaque nouvelle tentative d'accès exclusif. Une amélioration possible est la suivante : un site conserve les permissions acquises tant que c'est possible et ne redemande donc que les permissions manquantes.

1. Modifier l'algorithme précédent en conséquence. On prendra garde à la situation où une permission considérée comme acquise au début d'une requête peut être remise en cause par la réception d'une requête issue du site qui avait précédemment donné sa permission.