

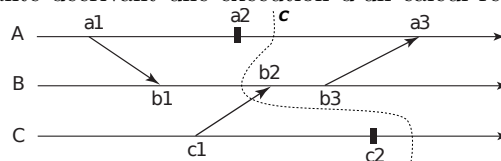
Systemes et algorithmique répartis

ENSEEIH/DIMA, master 2 Informatique et Télécommunication
1h45, documents autorisés

11 décembre 2013

1 Calcul réparti et causalité (8 points)

On considère la figure suivante décrivant une exécution d'un calcul réparti qui comporte 3 processus :



Questions (1 point par question)

1. Donner deux exemples de couples d'événements « parallèles » (c'est-à-dire non causalement liés) présents dans cette exécution répartie ;
2. Quel est le passé causal de b_3 ?
3. La coupure C indiquée en pointillé est-elle une coupure cohérente ? (justifier la réponse)
4. Dans l'exécution présentée, les événements de ce calcul se sont déroulés dans l'ordre temporel suivant : $a_1; b_1; c_1; a_2; b_2; b_3; c_2; a_3$. Donner une autre séquence d'exécution ordonnant temporellement les événements dans un ordre différent (au moins un événement déplacé) mais représentant le même calcul réparti, c'est-à-dire conservant les mêmes relations causales entre ces événements.
5. Ajouter un nombre si possible minimal de messages pour que tous les événements soient causalement ordonnés deux à deux dans ce calcul « augmenté » tout en conservant les relations de causalité initialement présentes. Faire la figure décrivant une exécution possible de ce calcul augmenté en mettant les nouveaux messages en pointillés.
6. Donner l'ordre total induit par les relations causales existant entre les événements de votre calcul augmenté.
7. Montrer qu'il est toujours possible de compléter un calcul réparti de façon à obtenir un calcul réparti « augmenté » c'est-à-dire conservant les relations causales initialement existantes mais introduisant un ordre total entre tous les événements du calcul via des nouveaux messages.
8. Proposer un algorithme général d'introduction des nouveaux messages.

2 Élection et consensus (3 points)

On suppose disponible un service de consensus dont l'interface est : **proposer**(v) pour qu'un site propose une valeur, et **obtenir**() $\rightarrow v$ pour acquérir la valeur décidée par le service de consensus (avec les propriétés d'accord, intégrité, validité, terminaison).

Question Proposer un algorithme d'élection, qui permet de choisir un site parmi un ensemble, en s'appuyant sur le service de consensus. Quelle hypothèse sur les processus participants est nécessaire ?

3 Problème : Diffusion totalement ordonnée (9 points)

L'étude porte sur l'implantation d'un protocole de diffusion totalement ordonnée sur un réseau de type Ethernet. Ce type de réseau offre la diffusion au niveau physique via une interface socket UDP, mais sans garantie de fiabilité de la diffusion (certains sites peuvent ne pas recevoir un message diffusé). On supposera que la diffusion entre deux sites fixés respecte l'ordre FIFO.

Au cours de l'étude, deux propriétés seront recherchées :

- Propriété *Ordre* : tous les messages diffusés sont délivrés dans le même ordre (total) sur l'ensemble des sites ;
- Propriété *Atomicité* : un message est délivré si et seulement si tous les autres sites l'ont reçu.

Hypothèse générales L'ensemble S des N sites participant à la diffusion est fixe. Chaque site est structuré en un code noyau exécutant le protocole de diffusion et un code application utilisant ce protocole. Un code application dispose de deux opérations implantées par le noyau :

- *Broadcast*(in message m) : diffuse le message m à tous les sites y compris lui-même.
- *Accept*(out message m) : délivre à l'appelant un message diffusé.

La primitive *Accept* extrait ses messages d'une file d'attente locale. Cette file est alimentée par la réception des messages diffusés (y compris par le site lui-même) dont la délivrance à l'application doit être contrôlée.

Pour implanter ces primitives (et pour simplifier), le noyau s'appuie sur :

- une diffusion non fiable fifo (fournie par l'interface socket UDP) ;
- une communication point à point fiable et fifo.

Principe de base Pour assurer le contrôle de la délivrance des messages, un protocole de supervision est mis en œuvre à l'aide d'un jeton circulant sur un anneau logique passant par tous les sites. Pour pouvoir diffuser un message, un site doit posséder le jeton. La circulation du jeton se fait via la communication point à point fiable.

Notation : Pour désigner la valeur d'un champ *item* transportée par le jeton, on préfixe son nom par *Jeton* en notant cette référence *Jeton.item*. Pour désigner une variable d'état x locale au site numéro s , on préfixe par s en notant cette référence $s.x$.

Question (2 points)

1. Montrer que le principe de base énoncé assure que tous les messages diffusés sont causalement liés.

Délivrance simple Dans un premier temps, le jeton transporte comme seule information le numéro dd de la dernière diffusion exécutée. De type entier naturel, *Jeton.dd* vaut 0 initialement. Pour assurer l'ordonnement des messages, la primitive *Broadcast*(m) exécute l'algorithme suivant :

```
Attendre le jeton du site prédécesseur ;
Mettre à jour la valeur du jeton (Jeton.dd := Jeton.dd + 1) ;
Diffuser ( $m$ , Jeton.dd) ;
Passer le jeton au site successeur ;
```

Un site qui n'a pas de *Broadcast* à réaliser transmet simplement le jeton au suivant.

Dans les questions qui suivent, on ne s'intéresse pas à la propriété *Atomicité*, sauf cas explicitement mentionné.

Questions (1 point par question)

2. Sur chaque site, on suppose qu'il existe une variable d'état *rcv* mémorisant le numéro du dernier message délivré sur ce site. Préciser le prédicat autorisant un site s à délivrer le message (m , $nseq$) reçu (présent dans la file locale) en garantissant seulement la propriété *Ordre*.

3. Un site peut-il toujours délivrer immédiatement à son application locale un message que lui-même diffuse (en supposant qu'un appel local *Accept* est en attente) ?
4. Comment un site quelconque s peut-il s'apercevoir de la perte d'un message ? Préciser le prédicat associé à la détection d'une telle perte.

Perte de message La perte d'un message au niveau d'un site s bloque la délivrance des messages sur ce site. Pour éviter ce blocage, le jeton est utilisé pour mettre en œuvre un protocole de retransmission. On introduit un champ supplémentaire dans le jeton. Ce champ *perdu*, de type entier naturel, permet à un site de placer un numéro de message perdu. Dès qu'un site détecte la perte d'un message, il va donc utiliser ce champ pour le faire savoir au site émetteur.

Initialement, la valeur du champ *perdu* est nulle. Elle est remise à zéro par le site émetteur lorsque celui-ci rediffusera le message demandé. Par ailleurs, pour pouvoir réémettre les messages, un site doit mémoriser ses messages.

Questions (1 point par question)

5. Pour pouvoir placer le numéro de son message perdu, un site attend de voir passer le jeton avec une valeur nulle dans le champ *perdu*. Montrer que cette stratégie pose un problème d'équité.
6. Proposer une stratégie équitable en gardant le même champ unique.

Ramasse-miettes Chaque site est amené à conserver les messages qu'il a diffusés pour pouvoir répondre aux éventuelles requêtes de retransmission. Il s'agit donc d'éviter une croissance infinie de cette file. Pour cela, on introduit dans le jeton un champ supplémentaire de type tableau. Ce tableau $pgr[N]$ de N éléments mémorise le plus grand numéro de message reçu **sans trou** pour chaque site. Si $pgr[s] = i$, cela signifie que le site s a reçu tous les messages de numéro inférieur ou égal à la valeur i .

Questions (1 point par question)

7. Préciser le prédicat qui permet à un site s d'éliminer un message $(m, nseq)$ présent dans sa file des messages émis.
8. Montrer que ce même prédicat peut être utilisé pour assurer la propriété *Atomicité* en contrôlant la délivrance des messages grâce à ce prédicat.