Conception des systèmes répartis

3ième Année Informatique et Mathématiques Appliquées
Durée : 2 heures
Documents autorisés : Précis et notes de cours

14 octobre 2003

Remarque préliminaire : Toutes les questions valent 2 points.

Horloge et causalité directe (d'après Vilay K. Garg)

On considère un schéma de datation utilisant des horloges vectorielles. Une date est représentée par un vecteur de dimension N où N est le nombre de processus communicants. Chaque processus P_i possède une horloge vectorielle locale H_i dotée des opérations classiques d'incrémentation et de recalage. Initialement, l'horloge H_i du processus P_i a la valeur $H_i[i] = 1 \ \wedge \ \forall j \neq i \ :: \ H_i[j] = 0$. La classe Horloge suivante décrit l'algorithme des opérations Top et Recaler.

```
class Horloge {
   int[] cpt ; int loc = 0 ;
   // Lecture et incrémentation de l'horloge
   int[] Top()  { int[] hr = cpt.clone() ; cpt[loc]++ ; return hr ; }
   // Recalage de l'horloge
   void Recaler( int j, int vj ) { /* message issu du processus d'indice j */
      cpt[loc]  = Math.max(cpt[loc],vj + 1) ;
      cpt[j]    = Math.max(cpt[j],vj) ;
      }
   // Constructeur : "loc" localise le site de l'horloge
   Horloge(int où, int N) {
      loc = où; cpt = new int[N] ;
      for (int i = 0; i<N; i++) cpt[i]=0; cpt[loc] = 1 ;
      }
}
```

La sémantique de Recaler et Top sur une horloge H_i s'exprime sous la forme de triplets de Hoare par :

– L'opération Top renvoie la valeur courante de l'horloge et incrémente de 1 l'élément d'indice i.

$$\{H_i = h\} \ \ \mathsf{Top()} \ \ \{H_i[i] = h[i] + 1 \wedge \forall j \neq i :: \ H_i[j] = h[j]\}$$

– L'opération Recaler(j, vj) comporte deux paramètres d'entrée :
  – d'une part, le paramètre j représente l'indice du processus P_j émetteur du message conduisant à l'appel de l'opération Recaler ;
  – d'autre part, le paramètre vj représente la valeur de l'élément $h_e[j]$ datant l'événement d'émission e du message émis par P_j.

$$\{H_i = h\} \ \mathsf{Recaler}(j, vj) \ \{H_i[i] = Max(h[i], vj{+}1) \wedge H_i[j] = Max(h[j], vj) \wedge \forall k \neq i, j \ :: \ H_i[k] = h[k]\}$$

Enfin, les actions de mise à jour de ces horloges lors les différents types d'événements sont les suivantes, où h_e dénote la date de l'événement d'émission d'un message par P_i et h_r la date de l'événement de réception par P_i d'un message m émis par le processus P_j :

Type d'événement du processus P_i	Action mettant en jeu l'horloge H_i
Événement interne sur i	int[] $h \ = \ H_i.Top()$
Émission sur i de m	int[] $h_e \ = \ H_i.Top()$;
	envoi de $< i, h_e[i], m >$;
Réception sur i de $< j, vj, m >$	$H_i.Recaler(j, vj)$;
émis par P_j	int[] $h_r \ = \ H_i.Top()$;

FIG. 1 – Tableau des actions de mise-à-jour de l'horloge vectorielle H_i d'un processus P_i

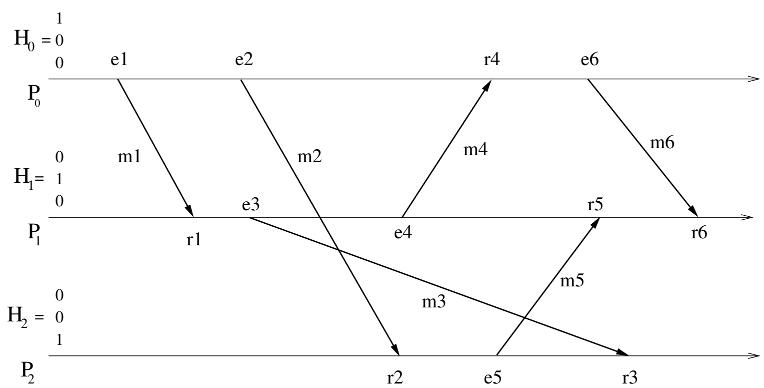

FIG. 2 – Chronogramme exemple

Notations

- Pour tout événement e, on note $e.p$ l'indice du processus ayant provoqué cet événement.
- La relation classique de causalité est notée $\prec$.

Remarque préliminaire Pour la plupart des questions, utiliser la sémantique par triplet de Hoare et le tableau des actions de la figure pour donner une schéma de preuve de la propriété demandée. En particulier, pour les invariants, montrer que si l'invariant est vrai avant une opération Top ou Recaler, celui-ci est encore vrai après l'exécution de l'opération.

Questions

1. Décorer le chronogramme de la figure 2 en précisant la date affectée par le mécanisme d'horloge proposé à chacun des événements. (Note : vous pouvez utiliser le chronogramme de la figure 3 en quatrième page comme support de la réponse.)
2. Montrer que toute horloge H_i vérifie l'invariant suivant :

$$\textbf{invariant} \ \ \forall j \neq i \ :: \ H_i[j] < H_i[i]$$

3. Montrer que l'horloge d'un processus est croissante. Autrement dit, deux événements distincts e et e' ayant lieu dans le même processus i et tels que e précède causalement e', sont datés par l'horloge H_i du processus avec des dates croissantes :

$$\textbf{invariant} \ \ \forall e \neq e' \ : \ e.p = e'.p = i \ :: \ e \prec e' \Rightarrow \ h_e < h_{e'}$$

4. On appelle chemin causal $C(e_n)$ conduisant à un événement e_n, une suite d'événements $\{e_1, e_2, \ldots, e_n\}$ vérifiant la propriété suivante : $e_1 \prec e_2 \prec \ldots \prec e_n$. La longueur d'un chemin causal, notée $|C(e_n)|$ est égale au nombre d'événements de la suite. [1]

   Montrez que, pour tout événement e d'un processus $P_{e.p}$, la valeur de l'élément $h_e[e.p]$ du vecteur horloge h_e datant cet événement est égale à la longueur du (ou des) plus long(s) chemin(s) causal(causaux) conduisant à e. Autrement dit, l'invariant suivant est vérifié :

$$\textbf{invariant} \;\; \forall\, e \;:\; h_e[e.p] = Max(|C(e)| : \forall C(e))$$

5. Montrer que ce type d'horloge capte la causalité entre événements. Autrement dit, si deux événements distincts e daté par le vecteur h_e et e' daté par le vecteur $h_{e'}$ sont tels que $e \prec e'$, alors $h_e[e.p] < h_{e'}[e'.p]$:

$$\textbf{invariant} \;\; \forall e \neq e' \;::\; e \prec e' \Rightarrow\; h_e[e.p] < h_{e'}[e'.p]$$

Causalité directe

On introduit la relation de causalité directe, notée $\prec_m$ dans laquelle m dénote un message. Cette relation est vérifiée entre deux événements e et e' ayant lieu dans deux processus **distincts** si et seulement si il existe un chemin causal ne comportant qu'un seul message m entre e et e'. Autrement dit, il existe au moins une communication de $P_{e.p}$ à $P_{e'.p}$ dans l'intervale de temps $[e, e']$. À titre d'exemple, dans le chronogramme de la figure 2, on a les relations :

$$r_1 \prec_{m_4} e_6, \quad e_1 \prec_{m_1} r_6, \quad e_1 \prec_{m_6} r_6, \quad e_1 \prec_{m_2} r_3$$

Questions

6. Donner, à partir du chronogramme de la figure 2, deux autres exemples d'événements vérifiant la relation de causalité directe et deux exemples ne vérifiant pas cette relation (paires d'événements (e, e') tels que $\neg(e \prec_m \; e' \;\vee\; e' \prec_{m'} e)$).
7. Montrez que : $\forall e, e' \;:\; e.p \neq e'.p \;::\; e \prec_m e' \Rightarrow e \prec e'$.
8. Montrer que le système de datation proposé vérifie :

$$\textbf{invariant} \;\; \forall e \neq e' : e.p \neq e'.p \;::\; h_e[e.p] \leq h_{e'}[e.p] \;\Leftrightarrow\; e \prec_m e'$$

Coupe cohérente

On suppose que les processus, au cours de leur exécution, décident (séparément) de prendre un cliché de leur état local $L[i]$ en le datant avec l'horloge locale du processus. Ces clichés locaux datés sont ensuite envoyés à un processus collecteur. Lorsque celui-ci possède une copie de tous les clichés locaux $L[i]$, il dispose d'un cliché global sous la forme du tableau L.

Questions

9. Montrez que le tableau des clichés locaux $L[i]$ constitue un cliché global cohérent si aucun des états locaux n'est causalement lié. Autrement dit si :

$$\forall i \neq j \;:\; \neg(L[i].e \prec L[j].e)$$

   où $L[x].e$ désigne l'événement de prise de cliché local sur P_x.
10. Comment vérifier, à partir des dates $L[i].h$ des clichés locaux collectés, la cohérence du cliché global que ces clichés locaux représentent ? Penser à utiliser (les propriétés de) la causalité directe des questions $(6, 7, 8)$.

[1] Par convention, on admet qu'un chemin causal $C(e)$ réduit à l'événement lui-même ($C(e) = \{e\}$) est donc de longueur 1.

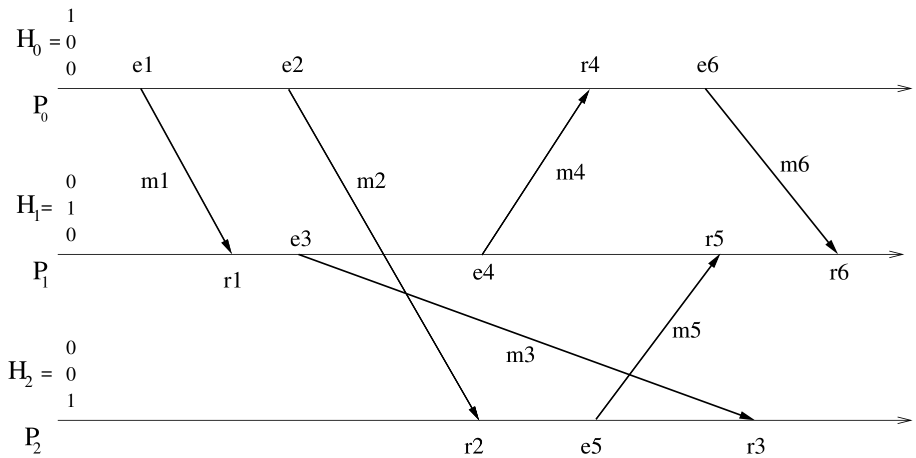

FIG. 3 – Chronogramme : à compléter