

TD sur les horloges logiques.

Exercice : Datation des événements

L'objectif est de comparer deux événements e_1 et e_2 qui se passent dans deux sites différents. Nous supposons que l'évènement e_1 (resp. e_2) est un évènement local du site 1 (resp. 2).

Question : Supposons que les deux sites ont une horloge de Lamport comme système de datation. Comparer les deux événements e_1 et e_2 dans les deux situations suivantes :

1. la date de l'évènement e_1 est 4 et celle de e_2 est 3.
2. la date de l'évènement e_1 est 4 et celle de e_2 est 4.

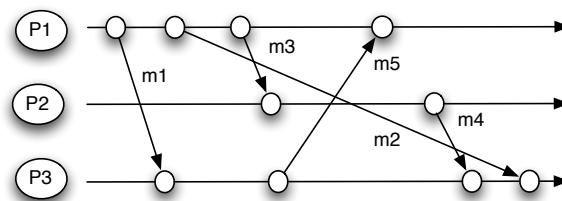
Question : Supposons que les deux sites ont une horloge vectorielle comme système de datation et que le système distribué est composé de trois sites. Comparer les deux événements e_1 et e_2 dans les deux situations : suivantes

1. la date de l'évènement e_1 est $\begin{pmatrix} 4 \\ 3 \\ 2 \end{pmatrix}$ et celle de e_2 est $\begin{pmatrix} 5 \\ 2 \\ 1 \end{pmatrix}$.
2. la date de l'évènement e_1 est $\begin{pmatrix} 5 \\ 2 \\ 4 \end{pmatrix}$ et celle de e_2 est $\begin{pmatrix} 4 \\ 3 \\ 6 \end{pmatrix}$.

Exercice : Horloge scalaire et horloge vectorielle

Complétez le chronogramme suivant

1. avec les horloges scalaires.
2. avec les horloges vectorielles correspondantes.



Exercice : Diffusion ne respectant pas l'ordre FIFO des messages

La diffusion est l'opération qui consiste pour un site donné à envoyer un même message à tous les autres sites d'un système. Nous considérons un système constitué de n sites P_0, P_1, \dots, P_{n-1} tous interconnectés.

On suppose maintenant que les sites sont tous fiables, mais que le réseau de communications peut ne pas respecter l'ordre FIFO. On suppose maintenant que P_0 diffuse plusieurs messages vers les autres sites. Proposer une méthode pour garantir que ces sites traitent les messages en respectant l'ordre d'envoi par P_0 .

Exercice : diffusions respectant l'ordre total.

Construire un algorithme de diffusion garantissant que les messages sont délivrés en respectant un ordre total. Le protocole ABCAST utilise un mécanisme d'estampillage scalaire. C'est un protocole en deux phases.

1. Chaque message contient une estampille qui peut être provisoire ou définitive (choisit par l'émetteur de la diffusion).
2. A la réception d'un message, chacun des sites du groupe lui attribue une estampille provisoire (à l'aide de son horloge logique). Le message reçu est rangé dans une file d'attente dans le site et son estampille est renvoyé à l'émetteur.
3. L'émetteur choisit l'estampille définitive une fois qu'il a reçu toutes les estampilles proposées par chacun du membre du groupe. puis la transmet à tous les membres du groupe.
4. Lors de la réception d'un message ayant une estampille définitive,...

Question 1 Considérons le scénario suivant : le système est composé de 4 sites. Les horloges ont, au début du scénario présenté, respectivement 4, 5, 7 et 6 comme valeurs. Les 4 messages A, B, C et D sont diffusés vers les 4 sites (peu importe la provenance de ces différents messages).

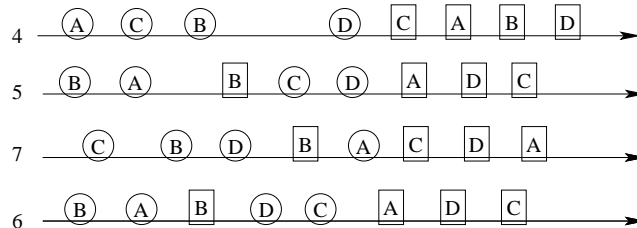


FIGURE 1 – Les événements ronds correspondent à la réception du message. Les événements rectangulaires représentent la réception de l'estampille définitive de ce message.

1. Donner les estampille de la première réception du message C sur chaque site la première fois. En déduire son estampille définitive.
2. Faire de même pour les messages A, B, D .

Question 2 Donnez l'ordre dans lequel les messages doivent être délivrés (en utilisant les définitives de ces messages).

Question 3 Donnez le comportement du premier site.

Question 4 Donnez le comportement du second site.

Question 5 : Terminer d'écrire l'algorithme

Exercice : diffusions respectant l'ordre causal (CB-CAST).

L'objectif de cet exercice est de construire un algorithme de diffusions garantissant que les messages sont délivrés en respectant l'ordre causal.

Définition : Si m_1 et m_2 sont deux messages, on dira que m_1 précède immédiatement m_2 , noté $m_1 \rightsquigarrow m_2$ si,

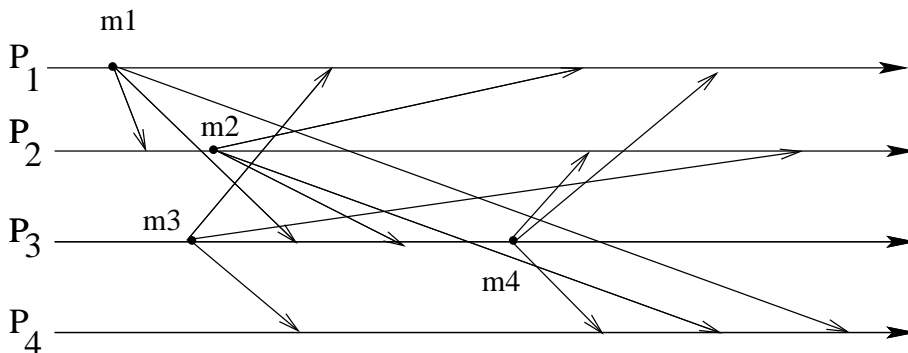
- $Envoyer(m_1) \rightarrow Envoyer(m_2)$,
- il n'existe pas de message m tel que $Envoyer(m_1) \rightarrow Envoyer(m)$ et $Envoyer(m) \rightarrow Envoyer(m_2)$.

Objectif : définir un protocole de diffusion respectant l'ordre causal

le but est le suivant :

Si $Envoyer(m_1) \rightarrow Envoyer(m)$ (délivrer en i), alors on veut qu'en site i , Délivrer (m_1) avant Délivrer(m)

Question 1 : Sur l'exemple suivant, définissez les dépendances causales entre les différentes diffusions. Définissez dans quel ordre les messages peuvent être délivrés ?



Nous allons définir un protocole de diffusion respectant l'ordre causal basé sur la notion d'horloge vectorielle.

Question 2 : Que peut-on dire sur les "dates" des messages m_1, m_2 tels que $Envoyer(m_1) \rightarrow Envoyer(m_2)$?

Question 3 : Lorsqu'un site P_i reçoit un message M , sous quelles conditions peut-il délivrer M en prenant en compte les horloges vectorielles ?

Question 4 : Écrire l'algorithme et appliquer l'algorithme sur l'exemple précédent.

Exercice : Horloge matricielle

Considérons un système contenant 3 sites. Tous les sites possèdent des horloges logiques matricielles. Chaque site i gère une horloge (notée HM_i) matricielle ($n \times n$) avec n le nombre de sites. Chaque message m envoyé est estampillé (daté) (EM_m) par la valeur courante de HM_i . L'élément $HM_i[j, k]$ signifie

- $HM_i[j, k]$ est le nombre de messages issus de p_j vers p_k dont p_i a connaissance.
- $HM_i[j, j]$ correspond au nombre d'évènements locaux du site j .

Question 1 : Comment HM_i est-elle modifiée lors de l'évènement local ou lors d'une émission d'un message ? *Indication : voir la figure suivante.*

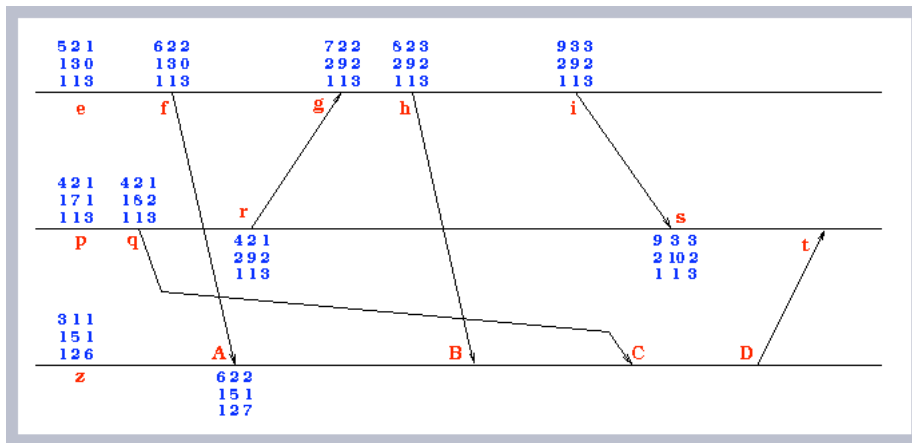


FIGURE 2 – datation avec l'horloge matricielle

Question 2 : Comment HM_i est-elle modifiée lors d'une réception d'un message afin d'assurer la délivrance causale ?

Question 3 : Supposons que l'horloge matricielle HM_3 du site 3 est

$$HM_3 = \begin{pmatrix} HM_3[1,1] & HM_3[1,2] & HM_3[1,3] \\ HM_3[2,1] & HM_3[2,2] & HM_3[2,3] \\ HM_3[3,1] & HM_3[3,2] & HM_3[3,3] \end{pmatrix} = \begin{pmatrix} 6 & 2 & 2 \\ 1 & 5 & 1 \\ 1 & 2 & 7 \end{pmatrix}$$

A quoi correspond l'élément de l'horloge matricielle $HM_3[3,1]$, pour le site 3 ?

Question 4 : Même question pour les éléments $HM_3[1,3]$, $HM_3[2,3]$ pour le site 3.

Le site 3 reçoit le message m en provenance du site 1. L'estampille du message m est

$$EM_m = \begin{pmatrix} 8 & 2 & 3 \\ 2 & 9 & 2 \\ 1 & 1 & 3 \end{pmatrix}$$

Question 5 : Que peut déduire le site 3 par rapport aux éléments $EM_m[1,3]$, $EM_m[2,3]$?

Question 6 : Le site 3 peut-il délivrer le message m (délivrance causale) ? Justifier votre réponse.

Exercice : Section critique à entrées multiples

On considère le problème de la k -exclusion mutuelle : une ressource qui doit être accédée en exclusion mutuelle est disponible en k exemplaires. Toutes les demandes des processus ne portent que sur un exemplaire de la ressource et un processus n'émet pas de nouvelle demande tant que sa demande en cours n'a pas été satisfaite. On peut donc avoir jusqu'à k processus simultanément en section critique.

Nous allons nous inspirer de l'algorithme proposé en cours (proposé par Ricart & Agrawala en 1983).

Question 1 : Combien d'autorisations faut-il pour recevoir pour garantir qu'au moins une ressource critique est libre ?

Question 2 : Proposer une extension du l'algorithme proposé en cours.