Université de Strasbourg
UFR Mathématique et Informatique

M1 Informatique
2020/2021

Algorithmes Distribués

TD n° 1 : Temps Logique (Horloges & Diffusion)

I Horloge de Lamport

L'horloge de Lamport a été inventée par Leslie Lamport en 1978. Il s'agit du premier type d'horloge logique introduit en informatique. L'horloge de Lamport respecte la dépendance causale ($e \xrightarrow{\bullet} e' \Rightarrow H(e) < H(e')$) mais non sa réciproque ($H(e) < H(e') \not\Rightarrow e \xrightarrow{\bullet} e'$).

Sur chaque site S_i on trouve une variable entière H_i dite horloge locale, initialisée à 0. La date locale d'un événement E est notée $d(E)$. Avec cette notation, l'algorithme de Lamport fonctionne comme suit :

- Pour chaque événement E qui correspond à un calcul local, le site S_i incrémente son horloge locale H_i et date l'événement E par $d(E) = H_i$.
- Lors de l'émission d'un message M par S_i, S_i incrémente son horloge locale H_i et estampille le message M par (H_i, i). Cet événement (l'envoi du message M) est daté par $d(E) = H_i$.
- Lors de la réception d'un message M estampillé (H_j, j) par S_i, S_i recale son horloge de la manière suivante :

  $$H_i = \text{Max}(H_i, H_j); \quad H_i = H_i + 1;$$

  Cet événement (la réception du message M) est daté par $d(E) = H_i$

La date globale $D(E)$ d'un événement E est alors notée $(d(E), i)$ où i est le numéro du site où à eu lieu l'événement et $d(E)$ sa date locale sur ce site.

Q 1. Appliquez cet algorithme à l'exemple illustré sur la figure 1. Donnez la date globale de chaque événement.

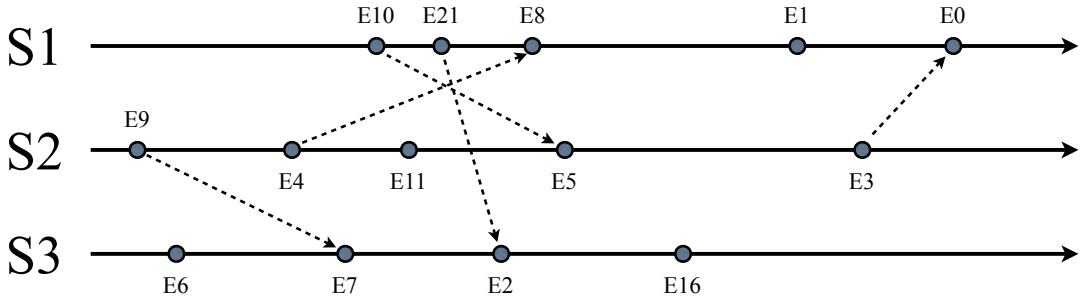

FIGURE 1 – Evénements répartis entre 3 sites avec dépendances causales

Q 2. Donnez les événements qui précédent causalement l'événement E_4, E_2, puis E_0.

Q 3. Donnez deux événements qui vérifient la relation $H(e) < H(e')$ mais qui n'ont pas de dépendance causale.

Q 4. En vous basant sur les estampilles de Lamport, donnez un ordre total des événements illustrés sur la figure 1. Qu'en déduisez-vous ?

II Horloge de Mattern

L'horloge de Mattern a été inventé par Mattern et Fidge dans les années 1989 - 1991. Ce type d'horloge assure la réciproque de la dépendance causale. Elle permet de savoir si deux événements sont concurrents (i.e. parallèles ou non dépendants causalement). Par contre elle ne définit pas un ordre total. Pour finir, alors qu'elle fournit toutes

les informations nécessaires pour la définition de la précédence causale (la dépendance causale pour la gestion de la réception des messages), elle ne suffit pas pour garantir l'ordre causal dans le cas général (pour la livraison des messages).

Sur chaque site S_i avec $i = 1, ..., n$ on définit une horloge vectorielle comme un vecteur $V_i[1...n]$ initialisé à 0. Á chaque événement E sur le site i, $V_i[i]$ est modifié de la manière suivante :

- $V_i[i] = V_i[i] + 1$ (quel que soit l'événement : calcul local, émission ou réception)
- si l'événement E correspond à l'envoi d'un message M par S_i, M est estampillé par V_m = valeur de l'horloge V_i au moment de l'envoi.
- si l'événement E correspond à la réception par S_i d'un message M estampillé par le vecteur V_m, S_i modifie son vecteur local comme suit :

$$V_i[k] = max(V_i[k], V_m[k]) \text{ pour } k = 1, ..., n$$

Dans tous les cas, l'événement E est daté par l'horloge V_i.

On considère les échanges entre 4 sites illustrés sur la figure 2.

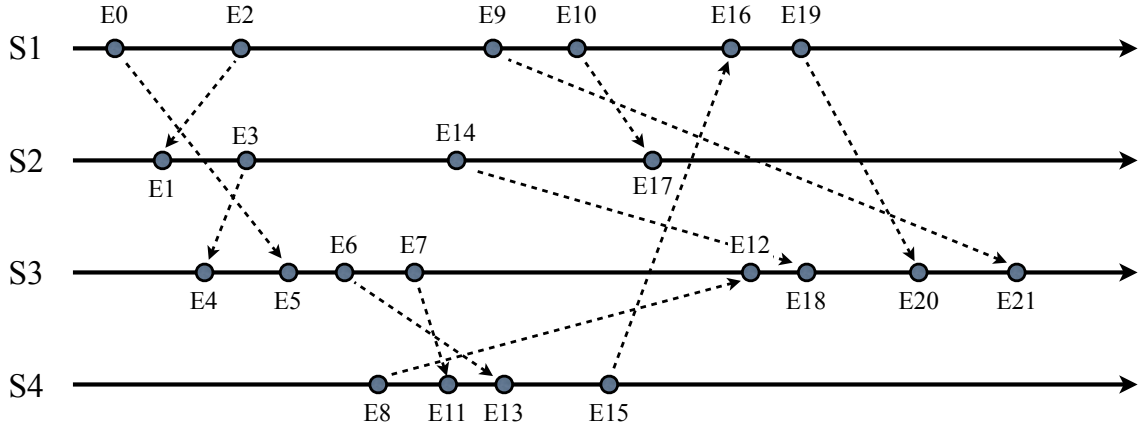

FIGURE 2 – Evénements répartis entre 4 sites avec dépendances causales

Q 1. Datez les événements E0 - E21, puis montrez que les événements E10 et E15 sont indépendants. Qu'en est-il de E2 et E15 ?

Q 2. Donnez l'équation permettant de déterminer si deux événements e_i et e_j, estampillés respectivement par les vecteurs V_i et V_j sur n sites, vérifient $e_i \xrightarrow{\bullet} e_j$? ou bien que e_i et e_j sont concurrents ? Montrez, dans le cas général, que précédence directe et indirecte ne sont pas distinguables avec seulement deux estampilles. Commentez.

Q 3. En vous basant sur les estampilles de Mattern, donnez le graphe des dépendances causales en rapport avec la figure 2. Qu'en déduisez-vous ?

Q 4. Démontrez la propriété fondamentale des estampilles vectorielles (Charron-Bost) : $e \xrightarrow{\bullet} e' \Leftrightarrow EV_e \subseteq EV_{e'}$

III Protocole d'ordre causal

Un protocole d'ordre causal est un protocole qui assure que les messages reçus sur un même site sont **livrés** en respectant les dépendances causales entre les événements d'émission de ces messages.

Pour un message m, on notera $e(m)$ sa date d'émission, $r(m)$ sa date de réception et $l(m)$ sa date de livraison, c'est-à-dire le moment où le message sera réellement livré au processus récepteur (la livraison peut être décalée dans le futur par rapport à la réception).

Q 1. Mettez en évidence le non-respect des dépendances causales en émission pour le chronogramme exposé sur la figure 3. Placez les événements de livraison des messages sur le chronogramme en respectant ces dépendances causales.

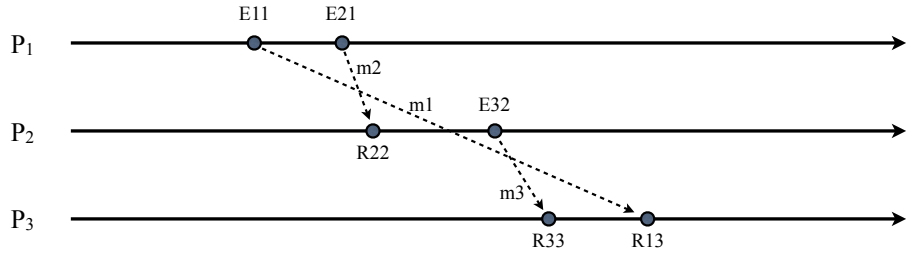

FIGURE 3 – Protocole d'ordre causal

Q 2. Déterminez la relation générale entre les événements associés à deux messages pour que l'ordre causal de leur émission soit respecté lors de leur livraison.

Q 3. Montrez que les horloges de Lamport (scalaires) et celles de Mattern (vectorielles) ne permettent pas de détecter le non-respect des dépendances causales en émission et de bien ordonner les livraisons des messages.

IV Diffusion causale

On peut représenter sur un chronogramme la diffusion d'un message comme un ensemble de sous-messages ayant un événement commun d'émission et un événement de réception pour chaque processus. Le chronogramme de la figure 4 représente trois processus qui communiquent par diffusion de message (quatre diffusions sont réalisées ici : messages m_1, m_2, m_3 et m_4). On considérera dans cet exercice des processus fiables en terme de communication.

Un message m diffusé est caractérisable par les événements suivants :
- $e_i(m)$ qui est la date d'émission du message (sa diffusion depuis P_i) ;
- pour chaque processus P_i, $r_i(m)$ est la date de réception du message par son système de communication ;
- pour chaque processus P_i, $l_i(m)$ est la date de livraison du message (sur P_i) par son système de communication.

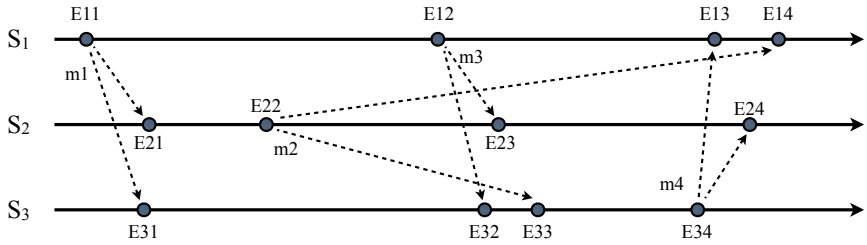

FIGURE 4 – Diffusion de message

Le chronogramme illustré sur la figure 4 représente uniquement les événements d'émission et la livraison immédiate d'un message dès sa réception, sans se préoccuper d'assurer des contraintes au niveau de l'ordre de livraison des messages ($l_i(m) = r_i(m)$).

La diffusion causale est un modèle de communication qui assure que si l'émission par diffusion d'un message m' dépend causalement de la diffusion d'un message m, alors tout processus livrera m avant m'.

Q 1. Mettez en évidence sur le chronogramme de la figure 4 un non-respect des conditions de la diffusion causale.

Q 2. Dans le cas des horloges vectorielles, déterminez la condition pour qu'un site S_j livre un message m reçu, daté par V_m et envoyé par un site S_i, tout en respectant la diffusion causale.

Q 3. Datez les événements du chronogramme selon l'algorithme **CBCAST** en remplissant le tableau suivant. Comment ce système d'horodatage permet de déterminer si la diffusion causale est respectée et, le cas échéant, de retarder la livraison d'un message ?

Evt	Site	H avant	Mess reçu	Action	Mess émis	H après
E11	S1					
E21	S2					
E31	S3					
E22	S2					
E12	S1					
E23	S2					
E32	S3					
E33	S3					
E34	S3					
E24	S2					
E13	S1					
E14	S1					

Rappel du fonctionnement de CBCAST :

L'algorithme Causal BroadCAST fonctionne comme suit :

- Chaque site S_i utilise une horloge vectorielle $H_i[N]$ (N sites).
- Pour chaque message m à diffuser par S_i, S_i traite m puis incrémente son horloge vectorielle $H_i[i]++$, et estampille m par $V_m = H_i$.
- À la réception d'un message m diffusé par S_i et estampillé par V_m, le récepteur S_j le met en attente avant livraison jusqu'à ce que la condition attendue en question 2 soit respectée.
- À la livraison du message m envoyé par S_i, S_j incrémente son horloge vectorielle comme suit $H_j[i]++$.

V Estampilles matricielles

Représentez l'évolution des horloges matricielles des processus P_1, P_2 et P_3 de l'exemple donné en figure 5. Annotez directement vos estampilles sur le schéma de la figure 5 depuis l'état initial (événements a, j, u) jusqu'aux événements pour lesquels l'estampille matricielle est déjà fournie (événements e, p, z).
Rappelez pourquoi l'ordre de livraison causale est respectée.

Rappel du fonctionnement des horloges matricielles :

- Lorsqu'un événement local se produit sur le site S_i : $HM_i[i,i]$ est incrémenté ;
- Lorsqu'un message est envoyé de S_i vers le site S_j : $HM_i[i,i]$ et $HMi[i,j]$ sont incrémentés ;
- Lorsqu'un message m en provenance de S_j est reçu par S_i, il faut s'assurer que tous les messages envoyés antérieurement vers S_i sont effectivement arrivés. Cela suppose donc que S_i ait reçu :
  — tous les messages en provenance de S_j : $EM_m[j,i] = HM_i[j,i] + 1$ (ordre FIFO sur le canal (j,i)) ;
  — tous les messages envoyés depuis d'autres sites que S_j : $\forall k \neq i$ et $k \neq j, EM_m[k,i] \leq HM_i[k,i]$.
- Si toutes ces conditions sont vérifiées, le message est livrable et l'horloge du site i est mise à jour : $HM_i[i,i]++$, $HM_i[j,i]++$ et $HM_i[k,l] = max(HM_i[k,l], EM_m[k,l])$ pour le reste de la matrice.
- Si les conditions ne sont pas toutes vérifiées, la livraison du message est retardée jusqu'à ce qu'elles le deviennent (grâce aux messages en retard) et l'horloge n'est pas mise à jour.

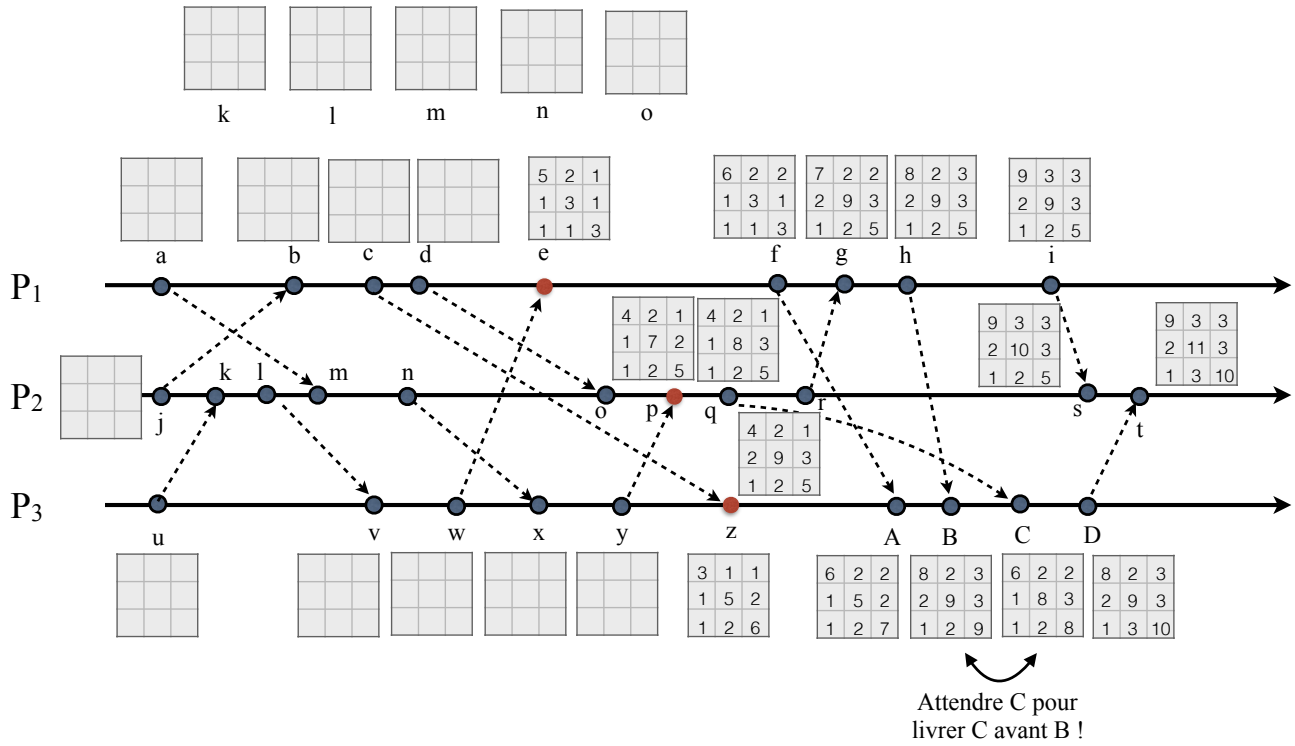

FIGURE 5 – Horloges matricielles