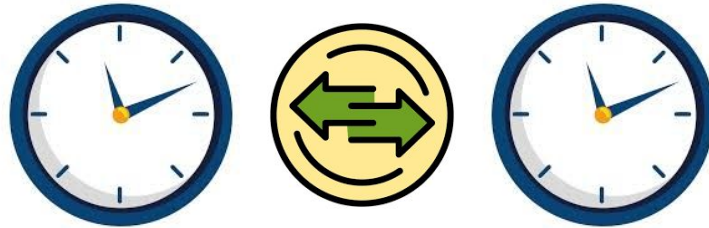


Les horloges logiques

Logical clocks



Sommaire :

- 1) Contexte et motivations
- 2) Événement et causalité d'événements
- 3) Horloge logique linéaire
- 4) Horloge logique vectorielle
- 5) Horloge logique matricielle

1) Contexte et motivations

Un système distribué implique l'exécution de plusieurs processus sur différentes machines.

Il n'existe pas de notion globale du temps physique dans les systèmes distribués car il est dépendant et peut varier entre chaque machine.

Selon Leslie Lamport en 1978:

“the logical nature of time is of primary importance when designing or analyzing distributed systems.”

Il est important de pouvoir déterminer l'ordre dans lequel les événements sont exécuté pour pouvoir assurer le bon fonctionnement du système :

Cohérence des données, Synchronisation, Debug, Suivi...

2) Événement et causalité d'événements

Groupe de n processus avec des canaux de communication bi-directionnels (FIFO et Lossless) entre chaque paire de processus.

Un processus peut effectuer 3 types d'événements :

- calcul interne
- envoi de message
- réception de message

Relation de dépendance causale entre 2 événements e_1 et e_2 :

$e_1 \rightarrow e_2$ signifie que “ e_1 se produit avant e_2 ” (ordre partiel)

Cône de causalité de e : ensemble des événements nécessaires avant e

3) Horloge logique linéaire (Lamport, 1978)

Principe :

Le temps logique est représenté par un entier

Chaque processus P_i possède sa propre horloge logique h_i qui représente sa vue locale du temps global

Règles :

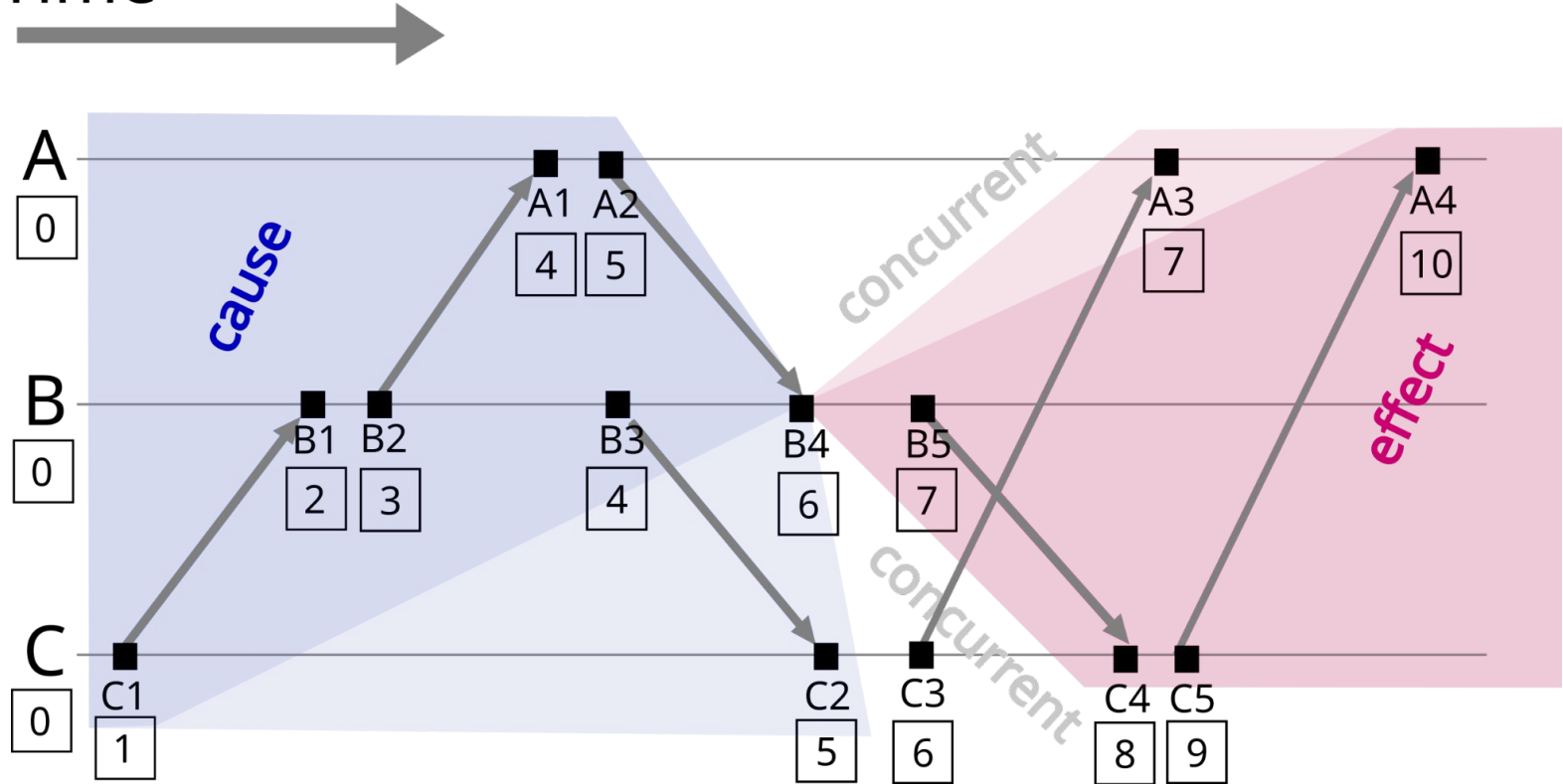
- R1 : Avant chaque événement, le processus incrémente son horloge locale de d
 $h_i := h_i + d$
- R2 : Lors de la réception d'un message (m, h) , le processus mets à jour son horloge locale avec la valeur maximum entre h_i et h , puis applique la règle R1, avant de délivrer le message m
 $h_i := \max(h_i, h) + d$

Propriétés :

- Monotonie : $e_1 \rightarrow e_2 \Rightarrow h(e_1) < h(e_2)$
- Ordre total en ajoutant l'envoyeur aux messages : e_1 avant $e_2 \Leftrightarrow (h_1 < h_2 \text{ OU } (h_1 = h_2 \text{ ET } P_1 < P_2))$

3) Horloge logique linéaire (Lamport, 1978)

Time



4) Horloge logique vectorielle (Fidge & Mattern, 1980s)

Principe :

Le temps est représenté par un vecteur de taille n (nombre de processus dans le groupe)

Chaque processus P_i possède sa propre horloge logique v_i où les valeurs $v_i[k]$ représentent la vue de P_i du temps local de P_k

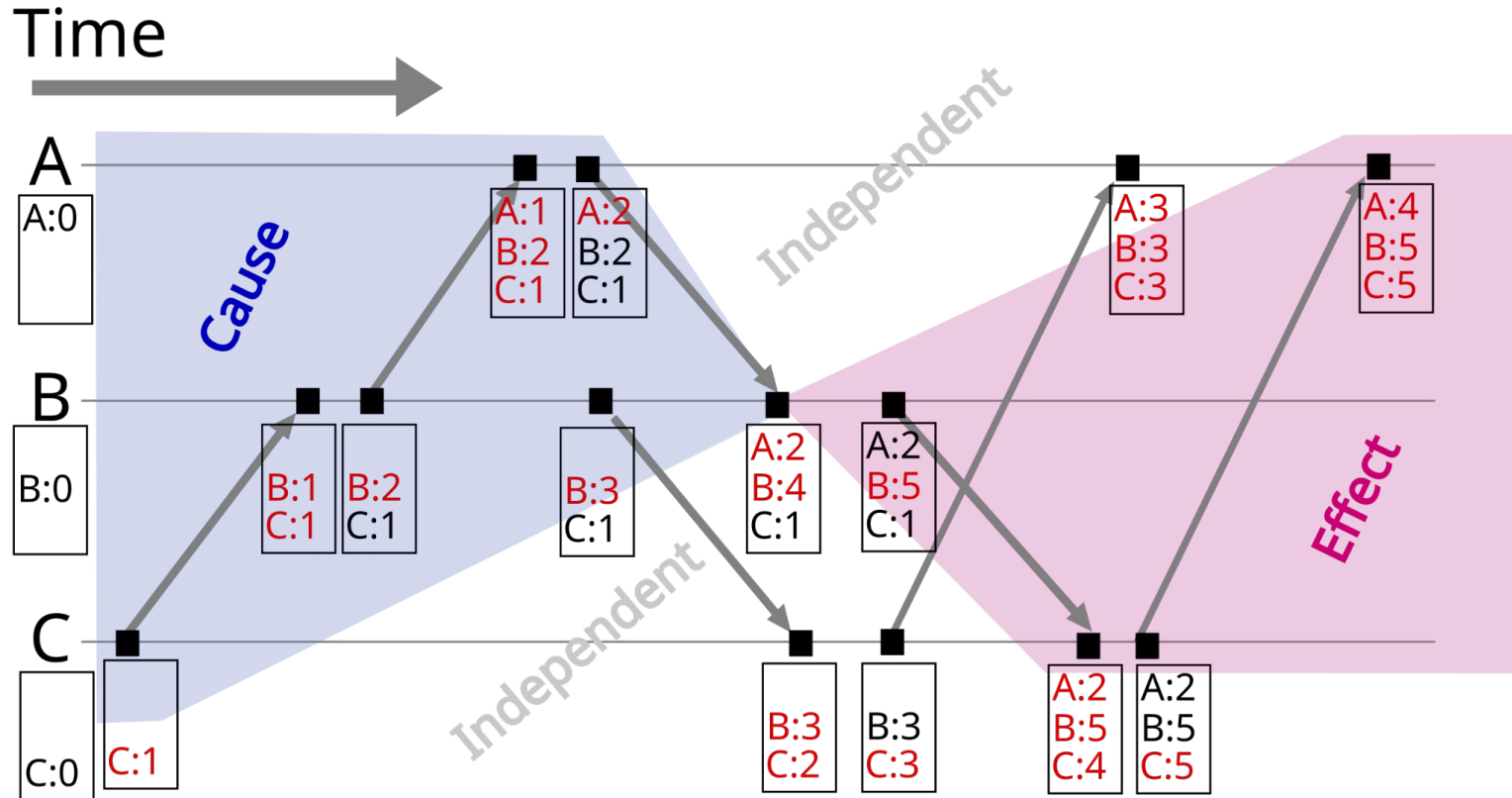
Règles :

- R1 : Avant de produire un événement, le processus incrémente son propre compteur de d
 $v_i[i] := v_i[i] + d$
- R2 : Lors de la réception d'un message (m, v) , le processus met à jour toutes les valeurs de son vecteur avec le max entre les valeurs présents et celles reçues, puis applique R1
 $\forall k \text{ de } [1, n] : v_i[k] := \max(v_i[k], v[k]) + d$

Propriétés :

- Événements liés : $e_1 \rightarrow e_2 \Leftrightarrow v_1 < v_2 \Leftrightarrow \forall k : v_1[k] \leq v_2[k] \text{ ET } \exists k : v_1[k] < v_2[k]$
- Événements indépendants/concurrents : $e_1 \parallel e_2 \Leftrightarrow v_1 < v_2 \text{ ET } v_1 > v_2$

4) Horloge logique vectorielle (Fidge & Mattern, 1980s)



5) Horloge logique matricielle (Wuu & Bernstein, 1984)

Principe :

Le temps est représenté par une matrice $n \times n$.

Chaque processus P_i possède une matrice M_i telle que :

$M_i[k,l]$ correspond à la vue qu'a P_i des connaissances de P_k sur l'horloge logique de P_l

La ligne $M_i[i,:]$ fonctionne de la même manière que v_i

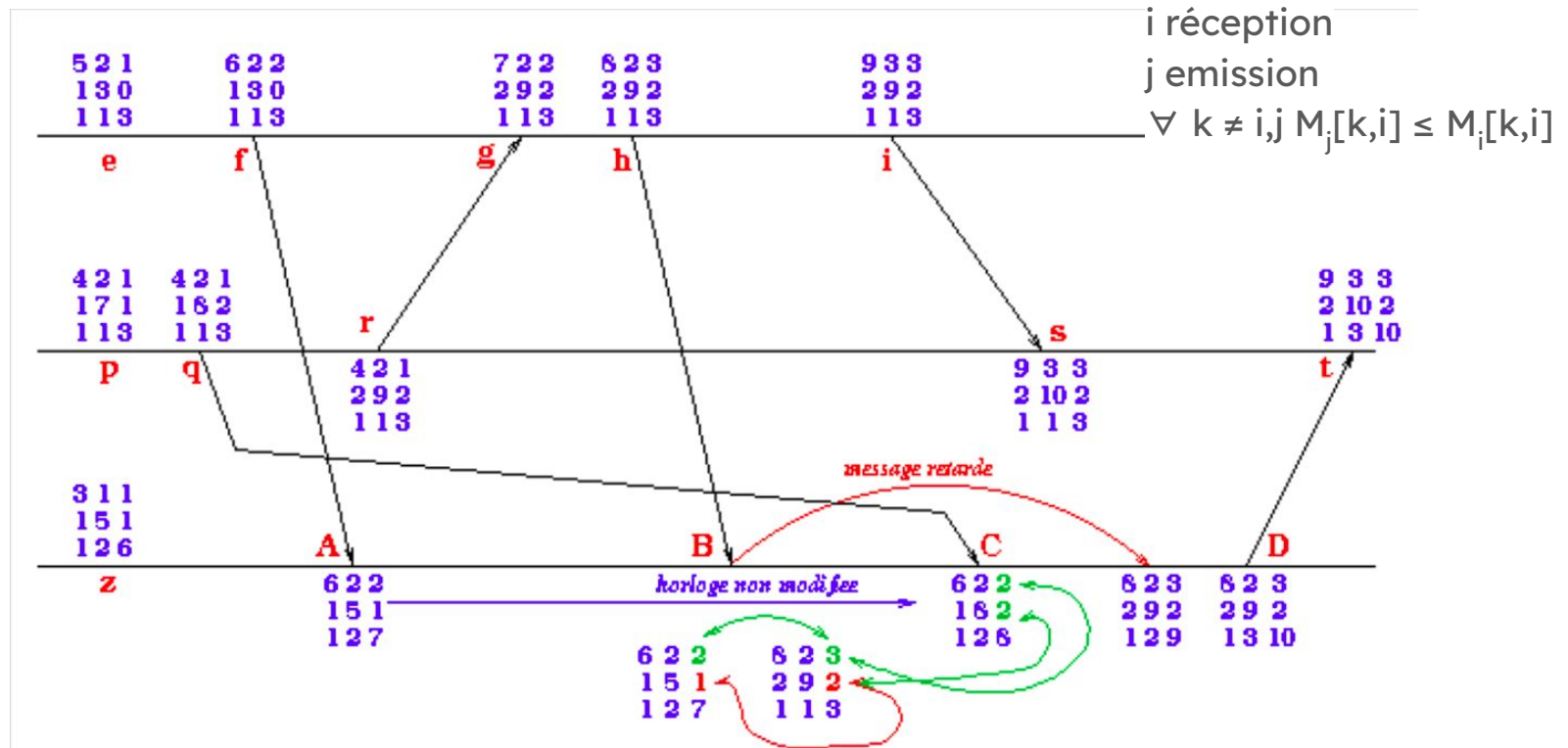
Règles :

- R1 : Avant chaque événement, le processus incrémente son horloge locale de d
 $M_i[i,i] := M_i[i,i] + d$
- R2 : A la réception d'un message accompagné de la matrice de son envoyeur M_j , le processus met à jour ses connaissances à partir de M_j
 $\forall k,l \in [1,n] : M_i[k,l] := \max(M_i[k,l], M_j[k,l])$

Propriétés :

- Celles de l'horloge vectorielle
- $\min_k (M_i[k,i]) \geq t \Rightarrow$ Le processus P_i sait que tous les autres processus connaissent ses progrès jusqu'à son temps t

5) Horloge logique matricielle (Wuu & Bernstein, 1984)



Les horloges logiques

Questions ?