

Systèmes et algorithmes répartis

Causalité et datation

Philippe Quéinnec, Gérard Padiou

ENSEEIH
Département Sciences du Numérique

6 juillet 2020



plan

- 1 Problème de datation
 - Temps logique
 - Horloge de Lamport
 - Horloge vectorielle de Fidge-Mattern
- 2 Les protocoles de communication
 - Délivrance ordonnée
 - Protocoles ordonnés
 - Protocole causalement ordonné
 - Diffusion causalement ordonnée



Plan

- 1 Problème de datation
 - Temps logique
 - Horloge de Lamport
 - Horloge vectorielle de Fidge-Mattern
- 2 Les protocoles de communication
 - Délivrance ordonnée
 - Protocoles ordonnés
 - Protocole causalement ordonné
 - Diffusion causalement ordonnée



Datation des événements



Objectif

Associer une date à chaque événement pour :

- **ordonner** les événements → compatible avec la causalité
- **identifier** les événements → dates uniques

Moyens

- Horloges matérielles
- Horloges logiques

Difficultés

- Pas d'horloge globale
- Tous les événements ne sont pas causalement liés
- Datation cohérente avec la relation causale :

$$\forall e, e' : e \prec e' \Rightarrow d_e < d_{e'}$$

Horloges matérielles



Idée

- Utiliser les horloges matérielles de chaque site
- **Risque** : Datation incohérente de l'événement de réception d'un message : la date de réception précède la date d'émission
- **Possible** si l'horloge du site de réception est en retard (suffisamment) sur celle du site de l'émetteur

Difficultés

- Cohérence avec la causalité
- Datation définissant un ordre total
- Pas d'unicité des dates

[Précis 3.3.1 pp.43–44]



Horloges matérielles



Solutions

- Synchronisation des horloges locales :
invariant $\max_{i=1..N}(h_i) - \min_{i=1..N}(h_i) < \epsilon$
- Causalité respectée si ϵ est inférieur au temps de transmission d'un message
- Unicité en utilisant couple (*date*, *id du site*)

Protocole de synchronisation d'horloges possible mais dans des contextes réseaux assurant une certaine qualité de service ou par l'usage d'un signal externe (horloge atomique, GPS)



Temps logique



Principes de base

- Associer à chaque site une horloge logique locale
- L'horloge compte les événements au lieu du temps réel
- Surcharger les messages avec leur date d'émission
- **Recaler si nécessaire** l'horloge locale d'un site lors de chaque réception de message
- **Avantage** : vision plus abstraite d'un calcul réparti

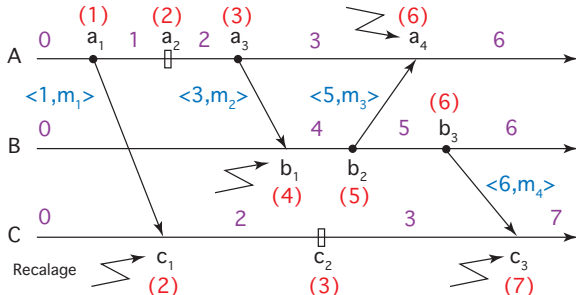
[Précis 3.3.2 pp.44–47]



Une tentative...



- Un compteur « horloge » sur chaque site
- Surcharge des messages et recalage de compteur



😊 $e \prec e' \Rightarrow d(e) < d(e')$

😊😞 Mêmes dates \Rightarrow pas causalement liés

😞 mais $d(e) < d(e') \not\Rightarrow e \prec e'$ (ex : $d(c_2) < d(b_3) \not\Rightarrow c_2 \prec b_3$)

Handwritten signature

Horloge de Lamport



Propriétés

- Introduit un ordre total entre événements
→ Dates distinctes pour tout couple d'événements
- Date = (compteur local, numéro de site)
ordre total sur les sites \Rightarrow ordre lexicographique total

```
struct Date { int cpt; // compteur d'événements
              int s;   // numéro de site
            }
bool prec (Date d1, Date d2) {
    return (d1.cpt < d2.cpt)
        || ((d1.cpt == d2.cpt) && (d1.s < d2.s));
}
```

1. *Time, Clocks and the Ordering of Events in a Distributed System*, Leslie Lamport. Communications of the ACM, July 1978.

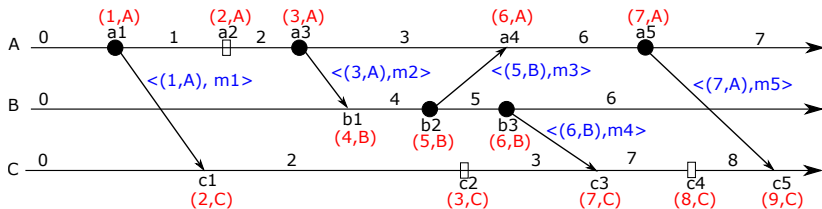


Actions associées aux événements



L'événement est daté avec le résultat de l'action.

Type d'événement sur un site s	Action sur le site s
Événement interne sur s	$H_s \leftarrow H_s + 1;$
Émission sur s de m	$H_s \leftarrow H_s + 1;$ envoi de $\langle \langle H_s, s \rangle, m \rangle$;
Réception sur s de $\langle \langle dm, s' \rangle, m \rangle$	$H_s \leftarrow \max(H_s, dm) + 1;$



Horloge vectorielle de Fidge-Mattern (1988)



Propriétés

- Datation isomorphe à l'ordre causal
- Utilisation de vecteurs de dimension égale au nombre de sites
- Pour un événement e , $HV(e)[i] =$ nombre d'événements du passé de e sur p_i (y compris e)
- Coût plus élevé : surcharge des messages par un vecteur

Expression des relations entre dates

$$\begin{aligned} D \leq D' &\triangleq \forall i : D[i] \leq D'[i] \\ D < D' &\triangleq D \leq D' \wedge \exists k : D[k] < D'[k] \\ D \parallel D' &\triangleq \neg(D < D') \wedge \neg(D' < D) \end{aligned}$$

1. *Timestamps in Message-Passing Systems That Preserve the Partial Ordering*, Colin J. Fidge. 11th Australian Computer Science Conference, 1988.
2. *Virtual Time and Global State in Distributed Systems*, Friedemann Mattern. Int'l Workshop on Parallel and Distributed Algorithms, 1989.

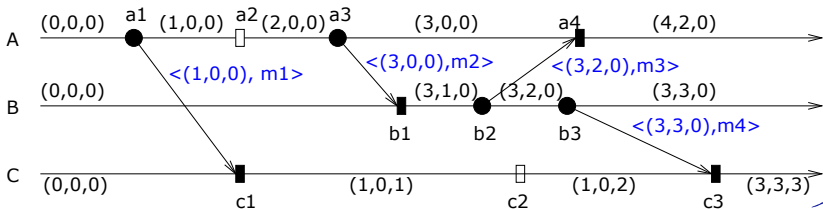


Actions associées aux événements



L'événement est daté avec le résultat de l'action.

Type d'événement sur un site s	Action sur le site s
Événement interne sur s	$H_s[s] \leftarrow H_s[s] + 1$
Émission sur s de m	$H_s[s] \leftarrow H_s[s] + 1$ envoi de $\langle H_s, m \rangle$
Réception sur s de $\langle dm, m \rangle$	$H_s[s] \leftarrow H_s[s] + 1$ $H_s[s'] \leftarrow \max(H_s[s'], dm[s']), \forall s' \neq s$



Horloge vectorielle de Fidge-Mattern



Datation isomorphe à l'ordre causal

$e \prec e' \Leftrightarrow D(e) < D(e')$ et $e \parallel e' \Leftrightarrow D(e) \parallel D(e')$

Coupure

- Date d'une coupure $C = (c_1, \dots, c_n)$:
 $HV(C) \triangleq \sup(HV(c_1), \dots, HV(c_n))$
- $C_2 \subset C_1 \Leftrightarrow HV(C_2) < HV(C_1)$
- C cohérente $\Leftrightarrow HV(C) = \langle HV(c_1)[1], \dots, HV(c_n)[n] \rangle$
- Soit C cohérente. Alors $\forall i, j : HV(c_i)[i] \geq HV(c_j)[i]$.
En effet, l'incrémement de $HV(c_i)[i]$ ne peut venir que d'un événement local ou résulter d'un message provenant du passé.
- Soit C non cohérente.
 \Rightarrow il existe un événement e_i hors coupe qui est dans le passé causal de C
 \Rightarrow Sur le site $i : HV(c_i)[i] < HV(e_i)[i]$ et $HV(e_i)[i] \leq HV(C)[i]$
 $\Rightarrow HV(C)[i] > HV(c_i)[i]$, donc $HV(C) \neq (HV(c_1)[1], \dots, HV(c_n)[n])$