Temps logique

0.1 Ordre partiel et causalité

Relation *est arrivé avant* (du point de vue d'un observateur interne au système, i.e. un processus) : relation d'ordre partiel sur l'ensemble des événements se produisant dans le système (ce n'est généralement pas une relation d'ordre total).

Leslie Lamport : *Time, Clocks and the Ordering of Events in a Distributed System* (Communications of the ACM, juillet 1978, vol.21, n° 7, p. 558-565), puis Fidge et Mattern en 1989.

0.1.1 Rappel

Une relation d'ordre (partiel) $\mathcal{R}$ sur un ensemble E est une relation binaire sur E qui est à la fois réflexive, antisymétrique et transitive. Deux éléments x et y de E sont dit comparables selon $\mathcal{R}$ si $(x \mathcal{R} y)$ *ou* $(y \mathcal{R} x)$. Dans le cas contraire, ils sont dits incomparables, ce qu'on note $x \mid y$. Une relation d'ordre (partiel) pour laquelle tous les éléments de E sont comparables 2 à 2 est appelée relation d'ordre total.

0.1.2 Exécution d'un programme parallèle et événements

Exécution d'un programme parallèle $\equiv$ ensemble (partiellement) ordonné d'événements : la relation d'ordre décrit les rapports de «causalité» entre ces événements.

Événements considérés : communications (émissions et réceptions de message) et éventuellement des événements locaux (i.e. événements n'induisant pas de communication). Le fait de prendre en compte tel ou tel événement dépend de ce que l'on souhaite observer en cours d'exécution.

0.1.3 Relation $\rightarrow$ (*est arrivé avant*)

La relation *est arrivé avant*, notée $\rightarrow$, sur l'ensemble E des événements pris en compte dans l'exécution d'un programme parallèle est la relation d'ordre partiel définie comme suit. Soit a et b deux éléments de E :

$\boxed{\Pi_1 \text{ : causalité intra-processus}}$ Si a et b sont des événements d'un même processus et si $a = b$ ou si a se produit strictement avant b au cours de l'exécution de ce processus, alors $a \rightarrow b$.

$\boxed{\Pi_2 \text{ : causalité inter-processus}}$ Si a est l'envoi d'un message par un processus et b la réception de ce message par un autre processus, alors $a \rightarrow b$.

En termes de causalité, un événement a peut influencer un événement b si et seulement si $a \rightarrow b$. Deux événements a et b sont dits «concurrents» s'ils ne sont pas comparables selon la relation $\rightarrow$ (on note souvent cela : $a \parallel b$, par analogie avec la notation $a \mid b$ et pour insister sur l'indépendance causale des événements). Aucun des deux ne peut influencer l'autre, autrement dit : *non* $(a \rightarrow b)$ *et non* $(b \rightarrow a)$.

Le *temps logique* défini par la relation $\rightarrow$ constitue une «approximation» du temps physique global, conservant deux de ses propriétés essentielles, et deux seulement, qui peuvent s'exprimer ainsi : vitesse strictement positive d'exécution des processus (cf. Π_1) et vitesse strictement positive de transfert des messages (cf. Π_2). Autrement dit, le temps logique s'écoule et il s'écoule dans le même sens que le temps physique global, mais on ne peut en dire davantage.

Le temps logique constitue la «meilleure approximation» du temps physique global pouvant être perçue par un observateur interne à un système distribué, i.e. un processus (sauf hypothèses particulières).

0.2 Diagramme espace-temps

Quand on considère un diagramme espace-temps, $a \rightarrow b$ si et seulement si $a = b$ ou si on peut aller de a à b en se déplaçant «vers l'avant», c'est-à-dire le long d'axes correspondant à la progression de processus et le long de flèches représentant des transferts de messages.

Sur un diagramme espace-temps, on ne représente pas ce qui peut être retrouvé par transitivité.

0.3 Horloges logiques de Lamport

La définition de la notion d'horloge logique fait abstraction de toute notion de processus et de programme parallèle pour ne considérer que des ensembles d'événements. Reste ensuite à la mettre en œuvre.

0.3.1 Définition

Une horloge logique de Lamport λ sur un ensemble partiellement ordonné $(E, \rightarrow)$ d'événements est une fonction totale quelconque de E vers l'ensemble $\mathbb{N}$ des entiers naturels muni de sa relation d'ordre habituelle $\leq$, qui vérifie :

$$\forall e \in E,\ e' \in E \textit{ distincts } : e \rightarrow e' \implies \lambda(e) < \lambda(e') \tag{1}$$

L'entier $\lambda(e)$ est appelé date logique de Lamport de l'événement e.

0.3.2 Remarque capitale

Attention, Lamport n'impose pas que :

$$\forall e \in E,\ e' \in E \textit{ distincts } : \lambda(e) < \lambda(e') \implies e \rightarrow e' \tag{2}$$

En ne satisfaisant pas cette dernière relation, la notion d'horloge logique de Lamport permet simplement d'affirmer, sur la base d'une relation $\textit{non}\,(\lambda(e) < \lambda(e'))$, qu'un événement e n'est pas *arrivé avant* un événement e', autrement dit qu'on a : $\textit{non}\,(e \rightarrow e')$.

Cette limitation des horloges de Lamport tient au fait que la relation $\rightarrow$ n'est pas en général une relation d'ordre total, alors que la relation d'ordre habituelle $\leq$ sur les entiers naturels en est une : l'ensemble ordonné $(E, \rightarrow)$ est n-dimensionnel tandis que l'ensemble ordonné $(\mathbb{N}, \leq)$ est mono-dimensionnel. Une horloge logique de Lamport introduit donc une perte d'information en matière de temps logique : la relation $\leq$ ne peut en général pas refléter exactement la relation $\rightarrow$.

Pour remédier à cette limitation, Fidge et Mattern ont défini, en 1989, une autre notion d'horloge logique à valeurs dans $\mathbb{N}^n$ (n étant le nombre de processus du programme parallèle) permettant aux processus de situer, exactement cette fois, en termes d'ordre partiel, les événements auxquels ils participent : pour ce type d'horloge, des relations analogues aux relations (1) et (2) ci-dessus sont vérifiées (elles sont réunies en la relation (3) ci-après). Si bien que deux événements sont concurrents si et seulement si leurs dates logiques de Fidge et Mattern ne sont pas comparables selon la relation d'ordre qu'ils considèrent sur l'ensemble $\mathbb{N}^n$ (voir ci-après). L'horloge logique de Fidge et Mattern n'introduit aucune perte d'information en matière de temps logique.

0.3.3 Conditions à respecter pour une mise en œuvre

Rattachons maintenant la notion d'horloge logique de Lamport à celles de processus et de programme parallèle. Considérons l'ensemble E des événements que constitue l'exécution d'un programme parallèle comprenant n processus $\mathtt{P_1}, \ldots, \mathtt{P_n}$. Notons E_i l'ensemble des événements d'un processus $\mathtt{P_i}$ au cours de cette exécution. On a : $E = \bigcup_{i=1}^{n} E_i$. Associons à tout processus $\mathtt{P_i}$ une fonction totale C_i de E_i vers $\mathbb{N}$ (fonction qu'on pourra implanter à l'aide d'une variable **locale** de type entier). Sur cette base, on peut définir la fonction totale suivante C de E vers $\mathbb{N}$:

$$\forall i \in \{1, \ldots, n\},\ \forall e \in E_i\ :\ C(e) = C_i(e)$$

Mais peut-on énoncer une condition suffisante simple pour que la fonction C soit une horloge logique de Lamport sur l'ensemble E ?

La réponse positive réside dans l'énoncé suivant : la fonction C est une horloge logique si elle possède les deux propriétés suivantes (démonstration immédiate) :

$\boxed{\Lambda_1 : \text{prise en compte de } \Pi_1}$ Pour tout couple (a, b) d'événements appartenant à un même ensemble E_i, autrement dit concernant un même processus $\mathtt{P_i}$, et tels que a se produise strictement avant b au cours de l'exécution de $\mathtt{P_i}$, on a : $C_i(a) < C_i(b)$.

$\boxed{\Lambda_2 : \text{prise en compte de } \Pi_2}$ Pour tout couple (a, b) d'événements tels que a soit l'émission d'un message par un processus $\mathtt{P_i}$ et b sa réception par un processus $\mathtt{P_j}$, on a : $C_i(a) < C_j(b)$.

Cette proposition peut être reformulée ainsi : la fonction C est une horloge logique de Lamport si, lorsqu'on parcourt vers l'avant un diagramme espace-temps, les dates logiques des événements rencontrés sont strictement croissantes. Formulée encore différemment, et très informellement, cette proposition devient : C est une horloge logique de Lamport si elle est ‹‹ compatible ›› avec les relations de causalité : la date logique de Lamport d'une cause est strictement antérieure à celle de son effet.

0.3.4 Mise en œuvre

La question qui se pose maintenant est celle-ci : peut-on implémenter une horloge logique de Lamport ? Ici encore, la réponse est positive et réside dans le ‹‹ protocole ›› suivant.

Chaque processus $\mathtt{P_i}$ dispose d'une variable entière $\mathtt{h_i}$ de valeur intiale quelconque (en pratique, on choisit généralement `0`) et :

$\boxed{\text{Mise en œuvre de } \Lambda_1 : \text{causalité intra-processus}}$ $\mathtt{P_i}$ incrémente $\mathtt{h_i}$ (en pratique, on choisit généralement une incrémentation de `1`) entre deux événements successifs quelconques, l'incrémentation ne générant aucun événement (en pratique, on peut par exemple choisir de programmer une incrémentation juste après chaque événement).

$\boxed{\text{Mise en œuvre de } \Lambda_2 : \text{causalité inter-processus}}$ Lors de l'émission d'un message, $\mathtt{P_i}$ ajoute au contenu de ce message une *estampille* (‹‹ *timestamp* ››) portant la valeur courante de sa variable $\mathtt{h_i}$.

Lors de la réception d'un message d'estampille val, $\mathtt{P_i}$ teste si $val' > val$, val' étant la valeur courante de sa variable $\mathtt{h_i}$, et si ce n'est pas le cas, $\mathtt{P_i}$ affecte à $\mathtt{h_i}$ une valeur strictement supérieure à val (en pratique, on choisit généralement la valeur $val + 1$), l'exécution de l'instruction conditionnelle corespondante ne générant aucun événement. L'événement correspondant à la réception du message est supposé survenir juste à l'issue de l'exécution de cette instruction conditionnelle. Remarque : si on a décidé que toutes les incrémentations effectuées seront des incrémentations de `1` et qu'une incrémentation sera faite juste après chaque événement, on peut ‹‹ fusionner ›› l'instruction conditionnelle et cette incrémentation ; ainsi, lors de la réception du message, $\mathtt{P_i}$ affectera à sa variable $\mathtt{h_i}$ la valeur $max\,(val,\ val') + 1$ et l'événement correspondant à cette réception sera supposé survenir juste à l'issue de l'exécution de cette affectation.

La date logique $C(a) = C_i(a)$ d'un événement a concernant un processus $\mathtt{P_i}$ est alors donnée par la valeur de la variable $\mathtt{h_i}$ de $\mathtt{P_i}$ au moment où survient cet événement.

0.4 Horloge logique de Fidge et Mattern

0.4.1 Notations

On considère un programme parallèle comportant n processus notés $\mathtt{P_1}, \ldots, \mathtt{P_n}$.

Soit I l'ensemble des numéros des processus. Soit E l'ensemble des événements jugés significatifs (toutes les émissions et toutes les réceptions, et éventuellement certains événements locaux) survenant au cours d'une exécution de ce programme. Soit E_i l'ensemble des événements survenant sur le processus $\mathtt{P_i}$. Soit $\nu(e)$ (ν pour ‹‹ numéro de processus ››) le numéro i du processus sur lequel s'est produit l'événement e. Soit $J(e)$ (J pour ‹‹ jusqu'à ››) l'ensemble fini d'événements

$\{e' \in E \mid e' \to e\}$ qui constitue le «passé logique» (en y incluant l'événement lui-même) d'un événement e. En effet, outre e, ce «passé logique» est constitué de l'ensemble des événements qui ont pu influencer e. Soit $J_i(e)$ l'ensemble $J(e) \cap E_i$, pour tout numéro i.

Soit $\mathbb{N}^n = \{(x_1, \ldots, x_n) \mid \forall i \in I : \; x_i \in \mathbb{N}\}$ l'ensemble des n-uplets d'entiers naturels. Pour tout i de I, la i^e composante d'un élément x de $\mathbb{N}^n$ est notée x_i. Pour tout x et tout y de $\mathbb{N}^n$, on note $sup_n\,(x,\, y)$ l'élément z de $\mathbb{N}^n$ défini par : $\forall i \in I \;:\; z_i = max\,(x_i,\, y_i)$. Pour tout x de $\mathbb{N}^n$ et tout i de I, on note $+1_i\,(x)$ l'élément y de $\mathbb{N}^n$ défini par : $(y_i \;=\; x_i + 1)\; et \; (\forall j \in I,\; j \neq i \;:\; y_j = x_j)$.

Pour tout ensemble fini Y, le nombre d'éléments (cardinal) de Yest noté $|Y|$. Soit Y et Y' deux ensembles : $Y' \subset Y$ signifie que Y' est strictement inclus dans Y (et diffère donc de Y) et $Y' \subseteq Y$ signifie que Y' est soit strictement inclus dans Y soit égal à Y.

0.4.2 Ordre partiel sur les n-uplets d'entiers naturels

Soit $<_n$ la relation binaire sur $\mathbb{N}^n$ telle que pour tout couple $(x,\, y)$ d'éléments de $\mathbb{N}^n$:

$$(x <_n y) \;\Leftrightarrow\; (\exists i \in I \;:\; x_i < y_i)\; et \; (\forall j \in I \;:\; x_j \leq y_j)$$

Alors la relation binaire $\leq_n$ sur $\mathbb{N}^n$ telle que pour tout couple $(x,\, y)$ d'éléments de $\mathbb{N}^n$: $(x \leq_n y) \;\Leftrightarrow\; (x <_n y)\; ou \;(x = y)$ est une relation d'ordre partiel sur $\mathbb{N}^n$ (on vérifiera aisément qu'elle n'est pas totale quand $n > 1$).

0.4.3 Horloge logique de Fidge et Mattern

L'horloge logique de Fidge et Mattern est la fonction totale φ de E vers $\mathbb{N}^n$ qui à tout événement e de E associe le n-uplet d'entiers naturels $\varphi\,(e)$ tel que :

$$\forall i \in I \;:\; \varphi\,(e)_i \;=\; |J_i\,(e)|$$

L'entier $\varphi\,(e)$ est appelé date logique de Fidge et Mattern de l'événement e.

0.4.4 Propriétés de l'horloge logique de Fidge et Mattern

$$\forall e \in E,\; e' \in E \; distincts \;:\; e \;\to\; e' \;\Longleftrightarrow\; \varphi\,(e) <_n \varphi\,(e') \tag{3}$$

Ainsi, deux événements sont concurrents si et seulement si leurs dates logiques de Fidge et Mattern ne sont pas comparables selon la relation $<_n$.

L'horloge logique de Fidge et Mattern n'introduit aucune perte d'information en matière de temps logique. C'est un isomorphisme entre les ensembles ordonnés $(E, \to)$ et $(\mathbb{N}^n, \leq_n)$, tous deux n-dimensionnels. C'est précisément ce qu'exprime la relation (3).

0.4.5 Mise en œuvre

On dote chaque processus `Pi` d'un tableau d'entiers `hi[1 : n]` dont les n éléments sont initialement nuls et :

- Causalité intra-processus : cf. Π_1 Juste avant chaque événement, `Pi` affecte la valeur $+1_i(T)$ à son tableau `Hi`, où T est la valeur courante de `Hi`, l'exécution de cette instruction ne générant aucun événement.
- Causalité inter-processus : cf. Π_2 Lors de l'émission d'un message, `Pi` ajoute au contenu de ce message une estampille portant la valeur courante de son tableau `Hi`.

  Lors de la réception d'un message d'estampille E, `Pi` affecte à son tableau `Hi` la valeur $sup_n\,(E,\, T)$, où T est la valeur courante de `Hi`, l'exécution de cette instruction ne générant aucun événement. L'événement correspondant à la réception du message est supposé survenir à l'issue de la mise à jour de `Hi`. Mais il ne faut pas oublier que juste avant chaque événement, il faut affecter à `Hi` la valeur $+1_i\,(T)$. On peut donc «fusionner» les deux instructions en une affectation à `Hi` de la valeur $sup_n\,(E,\,+1_i\,(T))$, l'événement correspondant à la réception du message étant supposé survenir à son issue.

La date logique d'un événement concernant un processus `Pi` est alors donnée par la valeur du tableau `Hi` de `Pi` au moment où survient cet événement.