



Le temps dans les système distribués

Master 2 Informatique - UFR S.A.T

Pr. Ousmane THIARE

othiare@ugb.edu.sn
[www.ousmanethiare.com]

April 16, 2020

Le temps dans les système distribués

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- 1 Le temps dans les système distribués
- 2 Temps logique
- 3 Horloges logiques (Algorithme de Lamport, 1978)
- 4 Les horloges vectorielles (Fidge et Mattern, 1988)
- 5 Utilisation des horloges logiques
- 6 Utilisation d'horloge vectorielle



Le temps dans
les système
distribués

Temps logique

Horloges
logiques
(Algorithme de
Lamport, 1978)

Les horloges
vectorielles
(Fidge et
Mattern, 1988)

Utilisation des
horloges
logiques

Utilisation
d'horloge
vectorielle

Le temps dans les système distribués

Le temps dans les système distribués

Introduction

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- Un S.D. est organisé comme un ensemble de processus qui s'exécutent sur des sites reliés par un système de communication et qui communiquent en échangeant des messages. Sur chaque site, il est possible de définir un état local, qui est modifié par l'exécution des processus du site.



Le temps dans les système distribués

Introduction

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- Un S.D. est organisé comme un ensemble de processus qui s'exécutent sur des sites reliés par un système de communication et qui communiquent en échangeant des messages. Sur chaque site, il est possible de définir un état local, qui est modifié par l'exécution des processus du site.
- On appelle événement un changement local d'état sur un site, ou bien l'envoi ou la réception d'un message par un processus.



Le temps dans les système distribués

Introduction

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- Un S.D. est organisé comme un ensemble de processus qui s'exécutent sur des sites reliés par un système de communication et qui communiquent en échangeant des messages. Sur chaque site, il est possible de définir un état local, qui est modifié par l'exécution des processus du site.
- On appelle événement un changement local d'état sur un site, ou bien l'envoi ou la réception d'un message par un processus.
- Une horloge locale unique permet de dater chacun des événements. Tous les événements ont des dates différentes. Mais, dans les s.d., bien que chaque site dispose d'une horloge locale, il n'existe pas d'horloge réelle globale, alors que l'ordre dans lequel surviennent les événements est parfois primordial.



Exemple

- C'est à la machine qui attend depuis le plus longtemps de rentrer en section critique.

Il est donc nécessaire de définir des méthodes qui permettent d'assurer une relation causale entre les différents événements. On va pour cela créer des horloges logiques ou physiques.



Exemple

- C'est à la machine qui attend depuis le plus longtemps de rentrer en section critique.
- Un message ne peut pas avoir une date de réception antérieure à sa date d'émission.

Il est donc nécessaire de définir des méthodes qui permettent d'assurer une relation causale entre les différents événements. On va pour cela créer des horloges logiques ou physiques.



Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Exemple

- C'est à la machine qui attend depuis le plus longtemps de rentrer en section critique.
- Un message ne peut pas avoir une date de réception antérieure à sa date d'émission.
- Deux sites modifient une donnée partagée. Quelle est la modification la plus récente?

Il est donc nécessaire de définir des méthodes qui permettent d'assurer une relation causale entre les différents événements. On va pour cela créer des horloges logiques ou physiques.



Le temps dans les système distribués

Relation de causalité

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Cette relation n'est mise en oeuvre que sur les événements. Il y a deux contraintes séquentielles fondamentales:

- **C1:** Les événements apparaissant sur un site donné sont totalement ordonnés, c'est à dire qu'on sait qui précède qui pour tout événement, et qu'il ne peut pas y avoir d'égalité.

Soit A et B deux événements. On dit que A précède B et on note $A \rightarrow B$ si et seulement si l'une des trois conditions suivantes est vraie:



Le temps dans les système distribués

Relation de causalité

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Cette relation n'est mise en oeuvre que sur les événements. Il y a deux contraintes séquentielles fondamentales:

- **C1:** Les événements apparaissant sur un site donné sont totalement ordonnés, c'est à dire qu'on sait qui précède qui pour tout événement, et qu'il ne peut pas y avoir d'égalité.
- **C2:** Etant donné un message quelconque M, l'émission de M précède sa réception.

Soit A et B deux événements. On dit que A précède B et on note $A \rightarrow B$ si et seulement si l'une des trois conditions suivantes est vraie:



Le temps dans les système distribués

Relation de causalité

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Cette relation n'est mise en oeuvre que sur les événements. Il y a deux contraintes séquentielles fondamentales:

- **C1:** Les événements apparaissant sur un site donné sont totalement ordonnés, c'est à dire qu'on sait qui précède qui pour tout événement, et qu'il ne peut pas y avoir d'égalité.
- **C2:** Etant donné un message quelconque M, l'émission de M précède sa réception.
- Une relation d'ordre entre les événements vérifiant ces contraintes a été définie par Lamport.

Soit A et B deux événements. On dit que A précède B et on note $A \rightarrow B$ si et seulement si l'une des trois conditions suivantes est vraie:



Le temps dans les système distribués

Relation de causalité

Le temps dans les système distribués

Temps logique

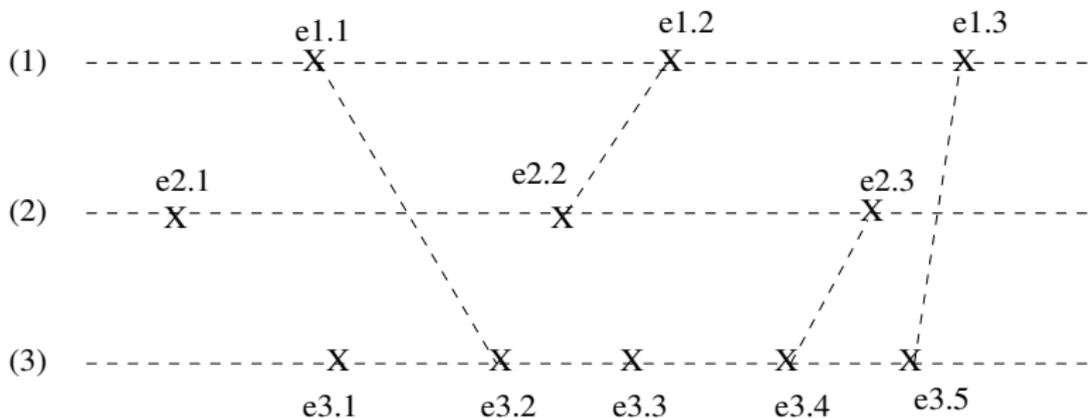
Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- A et B ont lieu sur le même site et A s'est produit avant B



Le temps dans les système distribués

Relation de causalité

Le temps dans les système distribués

Temps logique

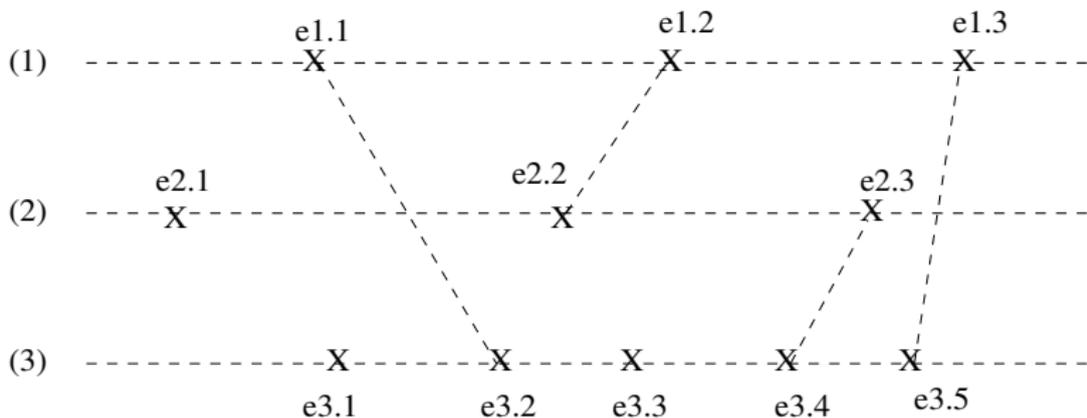
Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- A et B ont lieu sur le même site et A s'est produit avant B
- A est l'émission d'un message et B sa réception



Le temps dans les système distribués

Relation de causalité

Le temps dans les système distribués

Temps logique

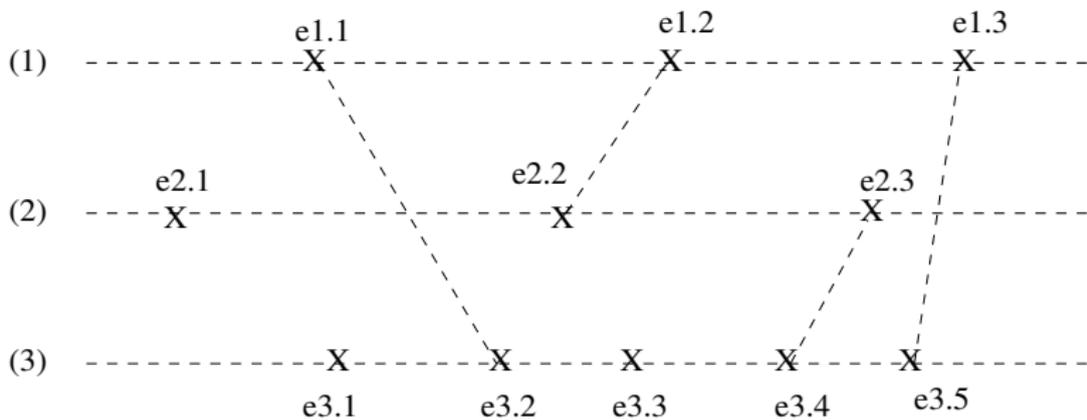
Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- A et B ont lieu sur le même site et A s'est produit avant B
- A est l'émission d'un message et B sa réception
- Il existe un événement C tel que $A \rightarrow C$ et $C \rightarrow B$



Le temps dans les système distribués

Relation de causalité

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- Deux événements liés par la relation de causalité appartiennent à un chemin causal. Par exemple, $e_{1.1}$ et $e_{3.4}$ appartiennent au chemin causal $e_{1.1}$, $e_{3.2}$, $e_{3.3}$, $e_{3.4}$.



Le temps dans les système distribués

Relation de causalité

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- Deux événements liés par la relation de causalité appartiennent à un chemin causal. Par exemple, $e_{1.1}$ et $e_{3.4}$ appartiennent au chemin causal $e_{1.1}$, $e_{3.2}$, $e_{3.3}$, $e_{3.4}$.
- De manière plus générale, si on considère pour un événement donné l'ensemble de tous ses prédécesseurs, on obtient l'ensemble de tous les événements qui en sont la cause [potentielle].



Le temps dans les système distribués

Relation de causalité

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- Deux événements liés par la relation de causalité appartiennent à un chemin causal. Par exemple, $e1.1$ et $e3.4$ appartiennent au chemin causal $e1.1, e3.2, e3.3, e3.4$.
- De manière plus générale, si on considère pour un événement donné l'ensemble de tous ses prédécesseurs, on obtient l'ensemble de tous les événements qui en sont la cause [potentielle].
- La relation de causalité entre événements crée donc un ordre partiel sur ces événements. Deux événements A et B sont dits concurrents (ou causalement indépendants) et notés $A||B$ si aucun des deux ne précède causalement l'autre, et ne peut donc influencer sur lui. Par exemple, $e2.1||e1.1$.



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

On construit un ordre total qui soit cohérent avec la relation de causalité. On associe à chaque événement une estampille telle que si on compare les estampilles de 2 événements, l'estampille de la cause est inférieure à celle de l'effet.

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 $d(E)$.

- Pour chaque événement E ne correspondant pas à l'envoi ou à la réception d'un message, le site S_i incrémente l'horloge locale H_i de 1 et date cet événement par $d(E)=H_i$.



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

On construit un ordre total qui soit cohérent avec la relation de causalité. On associe à chaque événement une estampille telle que si on compare les estampilles de 2 événements, l'estampille de la cause est inférieure à celle de l'effet.

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 $d(E)$.

- Pour chaque événement E ne correspondant pas à l'envoi ou à la réception d'un message, le site S_i incrémente l'horloge locale H_i de 1 et date cet événement par $d(E)=H_i$.
- Lors de l'émission d'un message M par S_i , on incrémente l'horloge locale H_i de 1, on estampille M par (H_i, i) et on date l'émission par $d(E)=H_i$.



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans
les système
distribués

Temps logique

Horloges
logiques
(Algorithme de
Lamport, 1978)

Les horloges
vectorielles
(Fidge et
Mattern, 1988)

Utilisation des
horloges
logiques

Utilisation
d'horloge
vectorielle

- Lors de la réception d'un message estampillé (H_j, j) par S_i , S_i recale son horloge locale de la manière suivante:

La date globale $d(E)$ d'un événement E est alors $(d(E), i)$ où i est le numéro du site où a lieu l'événement et $d(E)$ sa date locale.



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

- Lors de la réception d'un message estampillé (H_j, j) par S_i , S_i recalcule son horloge locale de la manière suivante:
 - Si $H_i < H_j$, alors $H_i = H_j + 1$

La date globale $d(E)$ d'un événement E est alors $(d(E), i)$ où i est le numéro du site où a lieu l'événement et $d(E)$ sa date locale.



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans
les système
distribués

Temps logique

Horloges
logiques
(Algorithme de
Lamport, 1978)

Les horloges
vectorielles
(Fidge et
Mattern, 1988)

Utilisation des
horloges
logiques

Utilisation
d'horloge
vectorielle

- Lors de la réception d'un message estampillé (H_j, j) par S_i , S_i recale son horloge locale de la manière suivante:
 - Si $H_i < H_j$, alors $H_i = H_j + 1$
 - Sinon $H_i = H_i + 1$

La date globale $d(E)$ d'un événement E est alors $(d(E), i)$ où i est le numéro du site où a lieu l'événement et $d(E)$ sa date locale.



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans
les système
distribués

Temps logique

Horloges
logiques
(Algorithme de
Lamport, 1978)

Les horloges
vectorielles
(Fidge et
Mattern, 1988)

Utilisation des
horloges
logiques

Utilisation
d'horloge
vectorielle

- Lors de la réception d'un message estampillé (H_j, j) par S_i , S_i recale son horloge locale de la manière suivante:
 - Si $H_i < H_j$, alors $H_i = H_j + 1$
 - Sinon $H_i = H_i + 1$
 - On a ensuite $d(E) = H_i$.

La date globale $d(E)$ d'un événement E est alors $(d(E), i)$ où i est le numéro du site où a lieu l'événement et $d(E)$ sa date locale.



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans les système distribués

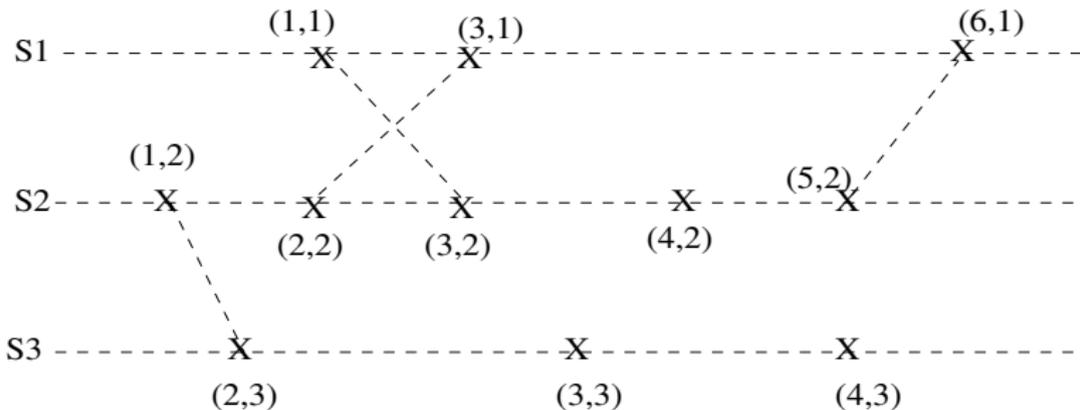
Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans
les système
distribués

Temps logique

Horloges
logiques
(Algorithme de
Lamport, 1978)

Les horloges
vectorielles
(Fidge et
Mattern, 1988)

Utilisation des
horloges
logiques

Utilisation
d'horloge
vectorielle

On a une relation d'ordre total notée \Rightarrow .
 $(d(E1), i1) \Rightarrow (d(E2), i2)$ si et seulement si l'une des deux
conditions suivantes est vraie:

- $d(E1) < d(E2)$



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

On a une relation d'ordre total notée \Rightarrow .
 $(d(E1), i1) \Rightarrow (d(E2), i2)$ si et seulement si l'une des deux conditions suivantes est vraie:

- $d(E1) < d(E2)$
- $d(E1) = d(E2)$ et $i1 < i2$



Le temps dans les système distribués

Horloges logiques (Algorithme de Lamport, 1978)

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Ce système d'estampillage est très intéressant lorsque le problème à résoudre nécessite un ordre total sur les événements respectant les relations de causalité qui les lient. C'est le cas par exemple de l'exclusion mutuelle.

Propriété

- Si H est la valeur d'horloge associée à un événement A , alors il y a exactement $H - 1$ événements qui précèdent causalement A sur le plus long chemin causal se terminant par A . En d'autres termes, il s'agit là du nombre maximum d'événements qui peuvent avoir lieu avant que A se produise.

La relation \implies ordonne bien les événements mais elle perd la relation de causalité, c'est à dire l'information selon laquelle deux événements sont indépendants \implies algo horloges vectorielles.



Le temps dans les système distribués

Les horloges vectorielles (Fidge et Mattern, 1988)

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

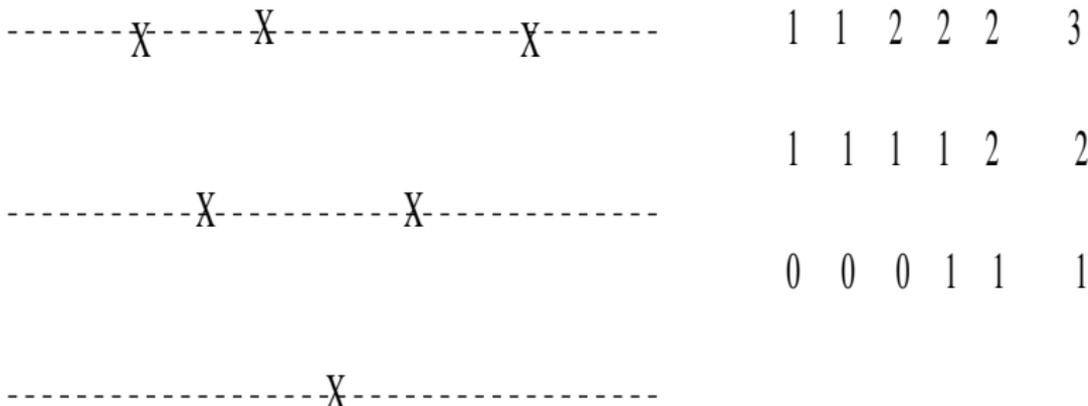
Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Principe

Considérons un observateur extérieur au système qui en percevrait tous les événements et qui compterait pour chaque site le nombre d'événements qui ont eu lieu depuis le début.



Le temps dans les système distribués

Les horloges vectorielles (Fidge et Mattern, 1988)

Le temps dans
les système
distribués

Temps logique

Horloges
logiques
(Algorithme de
Lamport, 1978)

Les horloges
vectorielles
(Fidge et
Mattern, 1988)

Utilisation des
horloges
logiques

Utilisation
d'horloge
vectorielle

Algorithme

Sur chaque site S_i , on définit une horloge vectorielle comme un vecteur $V_i[1..n]$ initialisé à 0.

- Pour chaque événement E ne correspondant pas à l'envoi ou la réception d'un message, on incrémente $V_i[i]$ de 1.



Le temps dans les système distribués

Les horloges vectorielles (Fidge et Mattern, 1988)

Le temps dans
les système
distribués

Temps logique

Horloges
logiques
(Algorithme de
Lampert, 1978)

Les horloges
vectorielles
(Fidge et
Mattern, 1988)

Utilisation des
horloges
logiques

Utilisation
d'horloge
vectorielle

Algorithme

Sur chaque site S_i , on définit une horloge vectorielle comme un vecteur $V_i[1..n]$ initialisé à 0.

- Pour chaque événement E ne correspondant pas à l'envoi ou la réception d'un message, on incrémente $V_i[i]$ de 1.
- Si l'événement E correspond à l'envoi d'un message M par un site S_i , on incrémente $V_i[i]$ de 1 et on estampille M par $V_m=V_i$.



Le temps dans les système distribués

Les horloges vectorielles (Fidge et Mattern, 1988)

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Algorithme

Sur chaque site S_i , on définit une horloge vectorielle comme un vecteur $V_i[1..n]$ initialisé à 0.

- Pour chaque événement E ne correspondant pas à l'envoi ou la réception d'un message, on incrémente $V_i[j]$ de 1.
- Si l'événement E correspond à l'envoi d'un message M par un site S_i , on incrémente $V_i[j]$ de 1 et on estampille M par $V_m=V_i$.
- Si l'événement E correspond à la réception d'un message (M, V_m) par S_i , $V_i[j]$ est incrémenté de 1 et pour k variant de 1 à n , $V_i[k]=\max(V_i[k], V_m[k])$



Le temps dans les système distribués

Les horloges vectorielles (Fidge et Mattern, 1988)

Le temps dans les système distribués

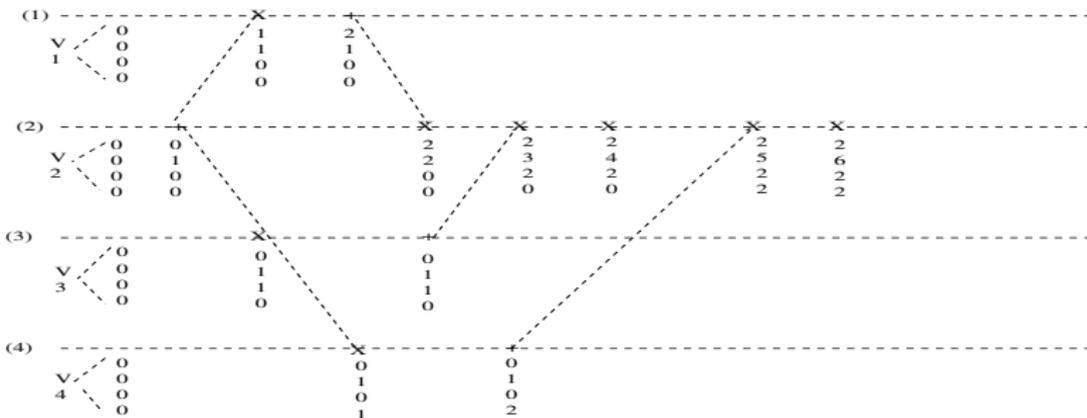
Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle



Pour deux horloges vectorielles $V1$ et $V2$, on a:

- $V1 \geq V2$ si et seulement si $V1[k] \geq V2[k]$ pour tout k

On garde donc la précédence causale, car $E1 \leftarrow E2$ sssi $V1 \geq V2$.



Le temps dans les système distribués

Les horloges vectorielles (Fidge et Mattern, 1988)

Le temps dans les système distribués

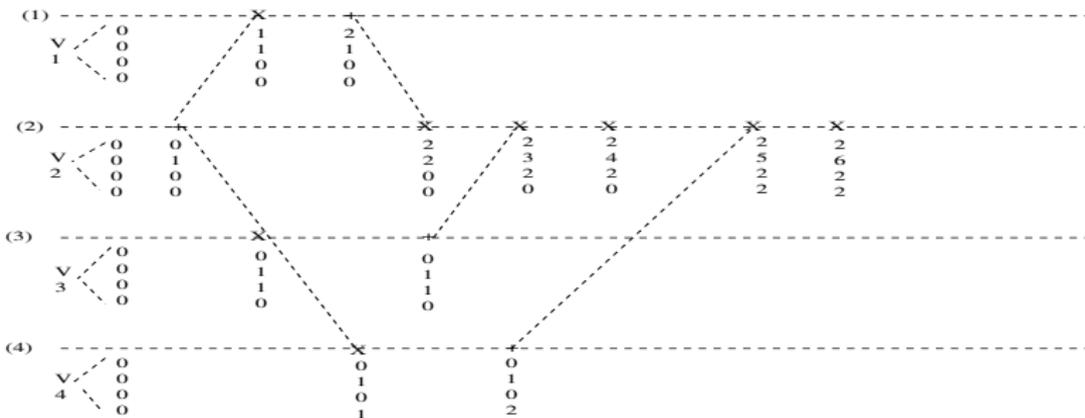
Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle



Pour deux horloges vectorielles V1 et V2, on a:

- $V1 \geq V2$ si et seulement si $V1[k] \geq V2[k]$ pour tout k
- $V1 \parallel V2$ si et seulement si on n'a ni $V1 \geq V2$, ni $V2 \geq V1$.

On garde donc la précédence causale, car $E1 \leftarrow E2$ sssi $V1 \geq V2$.



Le temps dans les système distribués

Utilisation des horloges logiques

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Le but est ici de fournir un mutex réparti. On va pour cela utiliser utiliser l'algorithme de Ricart et Agravala (1981).
On pose les hypothèses suivantes:

- Le nombre n de sites est connu



Le temps dans les système distribués

Utilisation des horloges logiques

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Le but est ici de fournir un mutex réparti. On va pour cela utiliser utiliser l'algorithme de Ricart et Agravala (1981).

On pose les hypothèses suivantes:

- Le nombre n de sites est connu
- Les canaux sont des FIFO fiables



Le temps dans les système distribués

Utilisation des horloges logiques

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Le but est ici de fournir un mutex réparti. On va pour cela utiliser utiliser l'algorithme de Ricart et Agravala (1981).
On pose les hypothèses suivantes:

- Le nombre n de sites est connu
- Les canaux sont des FIFO fiables
- L'ordre total est réalisé par l'algorithme de Lamport.



Le temps dans les système distribués

Utilisation des horloges logiques

La réalisation se fait en trois étapes:

- 1ère étape: Demande d'entrée d'un site S_i en section critique. Diffusion de $(req, H(req), S_i)$ à tous les sites, S_i compris

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle



Le temps dans les système distribués

Utilisation des horloges logiques

La réalisation se fait en trois étapes:

- 1ère étape: Demande d'entrée d'un site S_i en section critique. Diffusion de $(req, H(req), S_i)$ à tous les sites, S_j compris
- 2ème étape: Réception par S_i d'une demande d'entrée en section critique faite par S_j .

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle



Le temps dans les système distribués

Utilisation des horloges logiques

La réalisation se fait en trois étapes:

- 1ère étape: Demande d'entrée d'un site S_i en section critique. Diffusion de $(req, H(req), S_i)$ à tous les sites, S_i compris
- 2ème étape: Réception par S_i d'une demande d'entrée en section critique faite par S_j .
 - S_i n'est ni en section critique ni candidat pour y entrer: il renvoie $(ack, H(ack), S_i)$.

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle



Le temps dans les système distribués

Utilisation des horloges logiques

La réalisation se fait en trois étapes:

- 1ère étape: Demande d'entrée d'un site S_i en section critique. Diffusion de $(req, H(req), S_i)$ à tous les sites, S_i compris
- 2ème étape: Réception par S_i d'une demande d'entrée en section critique faite par S_j .
 - S_i n'est ni en section critique ni candidat pour y entrer: il renvoie $(ack, H(ack), S_i)$.
 - S_i est lui-même candidat pour rentrer en section critique:

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle



Le temps dans les système distribués

Utilisation des horloges logiques

La réalisation se fait en trois étapes:

- 1ère étape: Demande d'entrée d'un site S_i en section critique. Diffusion de $(req, H(req), S_i)$ à tous les sites, S_i compris
- 2ème étape: Réception par S_i d'une demande d'entrée en section critique faite par S_j .
 - S_i n'est ni en section critique ni candidat pour y entrer: il renvoie $(ack, H(ack), S_i)$.
 - S_i est lui-même candidat pour rentrer en section critique:
 - La demande de S_j est antérieure à sa demande: S_i renvoie un $(ack, H(ack), S_i)$

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle



Le temps dans les système distribués

Utilisation des horloges logiques

La réalisation se fait en trois étapes:

- 1ère étape: Demande d'entrée d'un site S_i en section critique. Diffusion de $(req, H(req), S_i)$ à tous les sites, S_i compris
- 2ème étape: Réception par S_i d'une demande d'entrée en section critique faite par S_j .
 - S_i n'est ni en section critique ni candidat pour y entrer: il renvoie $(ack, H(ack), S_i)$.
 - S_i est lui-même candidat pour rentrer en section critique:
 - La demande de S_j est antérieure à sa demande: S_i renvoie un $(ack, H(ack), S_i)$
 - Autrement : Mise en attente par S_i de la demande de S_j .

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle



Le temps dans les système distribués

Utilisation des horloges logiques

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

La réalisation se fait en trois étapes:

- 1ère étape: Demande d'entrée d'un site S_i en section critique. Diffusion de $(req, H(req), S_i)$ à tous les sites, S_i compris
- 2ème étape: Réception par S_i d'une demande d'entrée en section critique faite par S_j .
 - S_i n'est ni en section critique ni candidat pour y entrer: il renvoie $(ack, H(ack), S_i)$.
 - S_i est lui-même candidat pour rentrer en section critique:
 - La demande de S_j est antérieure à sa demande: S_i renvoie un $(ack, H(ack), S_i)$
 - Autrement : Mise en attente par S_i de la demande de S_j .
 - S_i est en section critique : Mise en attente par S_i de la demande S_j



Le temps dans les système distribués

Utilisation des horloges logiques

Le temps dans les système distribués

Temps logique

Horloges logiques
(Algorithme de Lamport, 1978)

Les horloges vectorielles
(Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

La réalisation se fait en trois étapes:

- 1ère étape: Demande d'entrée d'un site S_i en section critique. Diffusion de $(req, H(req), S_i)$ à tous les sites, S_i compris
- 2ème étape: Réception par S_i d'une demande d'entrée en section critique faite par S_j .
 - S_i n'est ni en section critique ni candidat pour y entrer: il renvoie $(ack, H(ack), S_i)$.
 - S_i est lui-même candidat pour rentrer en section critique:
 - La demande de S_j est antérieure à sa demande: S_i renvoie un $(ack, H(ack), S_i)$
 - Autrement : Mise en attente par S_i de la demande de S_j .
 - S_i est en section critique : Mise en attente par S_i de la demande S_j
- 3ème étape: Libération de la section critique : On envoie $(ack, H(ack), S_i)$ à toutes les demandes mises en attentes par S_i

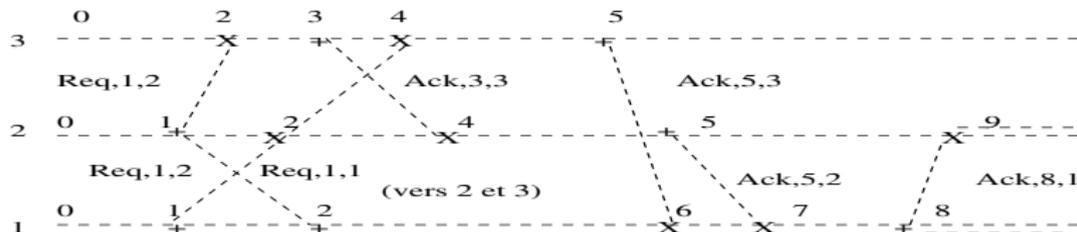


Le temps dans les système distribués

Utilisation des horloges logiques

Dans cet exemple, les machines 1 et 2 ont besoin d'entrer en section critique. Les conventions de représentation sont les suivantes:

- -: Exécution normale

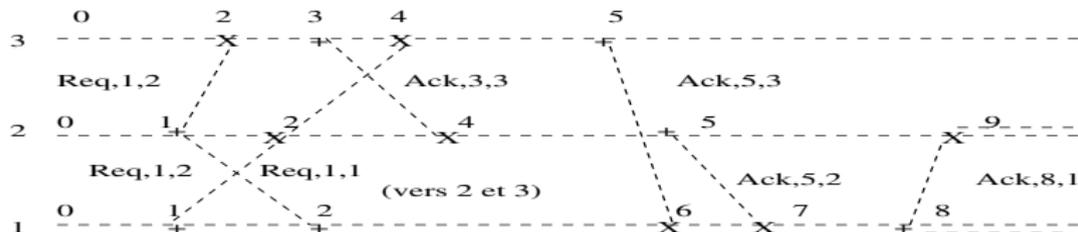


Le temps dans les système distribués

Utilisation des horloges logiques

Dans cet exemple, les machines 1 et 2 ont besoin d'entrer en section critique. Les conventions de représentation sont les suivantes:

- -: Exécution normale
- =: Exécution en section critique

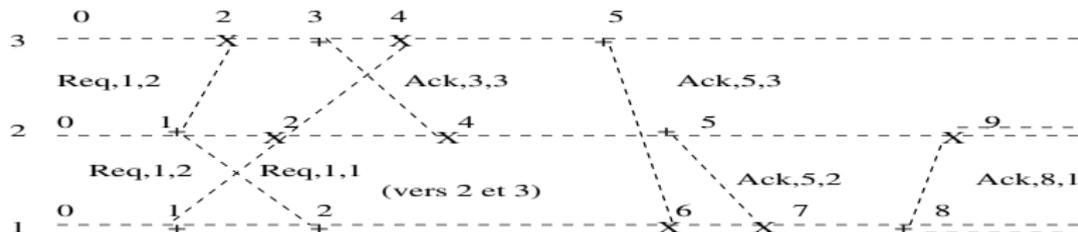


Le temps dans les système distribués

Utilisation des horloges logiques

Dans cet exemple, les machines 1 et 2 ont besoin d'entrer en section critique. Les conventions de représentation sont les suivantes:

- -: Exécution normale
- =: Exécution en section critique
- +: Emission d'un message

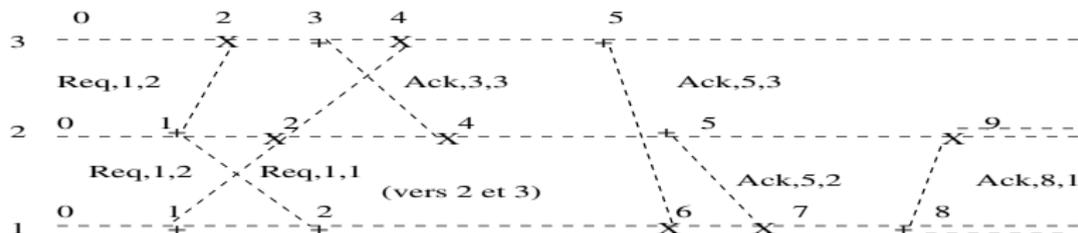


Le temps dans les système distribués

Utilisation des horloges logiques

Dans cet exemple, les machines 1 et 2 ont besoin d'entrer en section critique. Les conventions de représentation sont les suivantes:

- -: Exécution normale
- =: Exécution en section critique
- +: Emission d'un message
- x: Réception d'un message



Le temps dans les système distribués

Utilisation d'horloge vectorielle

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Le but est de permettre une diffusion fiable avec ordre causal.

ATTENTION!!!: Ici, on estampille uniquement les émissions.

On procède en 4 étapes:

- Avant diffusion du message M , le site S_i incrémente $V_i[i]$ de 1.



Le temps dans les système distribués

Utilisation d'horloge vectorielle

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Le but est de permettre une diffusion fiable avec ordre causal.

ATTENTION!!!: Ici, on estampille uniquement les émissions.

On procède en 4 étapes:

- Avant diffusion du message M , le site S_i incrémente $V_i[i]$ de 1.
- On estampille le message M par $V_m=V_i$



Le temps dans les système distribués

Utilisation d'horloge vectorielle

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Le but est de permettre une diffusion fiable avec ordre causal.

ATTENTION!!!: Ici, on estampille uniquement les émissions.

On procède en 4 étapes:

- Avant diffusion du message M , le site S_i incrémente $V_i[i]$ de 1.
- On estampille le message M par $V_m=V_i$
- A la réception sur S_j de (M, V_m) diffusé par S_i , on attend que:



Le temps dans les système distribués

Utilisation d'horloge vectorielle

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Le but est de permettre une diffusion fiable avec ordre causal.

ATTENTION!!!: Ici, on estampille uniquement les émissions.

On procède en 4 étapes:

- Avant diffusion du message M , le site S_i incrémente $V_i[i]$ de 1.
- On estampille le message M par $V_m=V_i$
- A la réception sur S_j de (M, V_m) diffusé par S_i , on attend que:
 - Toutes les diffusions précédentes de S_i soient arrivées sur S_j , c'est à dire que $V_j[i]=V_m[i]-1$.



Le temps dans les système distribués

Utilisation d'horloge vectorielle

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Le but est de permettre une diffusion fiable avec ordre causal.

ATTENTION!!!: Ici, on estampille uniquement les émissions.

On procède en 4 étapes:

- Avant diffusion du message M , le site S_i incrémente $V_i[i]$ de 1.
- On estampille le message M par $V_m=V_i$
- A la réception sur S_j de (M, V_m) diffusé par S_i , on attend que:
 - Toutes les diffusions précédentes de S_i soient arrivées sur S_j , c'est à dire que $V_j[i]=V_m[i]-1$.
 - Toutes les diffusions antérieures à m et reçues sur S_i aient été aussi reçues par S_j , c'est à dire que $V_j[k] \geq V_m[k]$ pour tout k différent de i .



Le temps dans les système distribués

Utilisation d'horloge vectorielle

Le temps dans les système distribués

Temps logique

Horloges logiques (Algorithme de Lamport, 1978)

Les horloges vectorielles (Fidge et Mattern, 1988)

Utilisation des horloges logiques

Utilisation d'horloge vectorielle

Le but est de permettre une diffusion fiable avec ordre causal.

ATTENTION!!!: Ici, on estampille uniquement les émissions.

On procède en 4 étapes:

- Avant diffusion du message M , le site S_i incrémente $V_i[i]$ de 1.
- On estampille le message M par $V_m=V_i$
- A la réception sur S_j de (M, V_m) diffusé par S_i , on attend que:
 - Toutes les diffusions précédentes de S_i soient arrivées sur S_j , c'est à dire que $V_j[i]=V_m[i]-1$.
 - Toutes les diffusions antérieures à m et reçues sur S_i aient été aussi reçues par S_j , c'est à dire que $V_j[k] \geq V_m[k]$ pour tout k différent de i .
- Mise à jour de $V_j[i]$.

