

CONSTRUCTION D'UN TEMPS VIRTUEL GLOBAL

PR. DJAMEL EDDINE SAIDOUNI

EQUIPE CFSC, LABORATOIRE MISC

DÉPARTEMENT IFA, FACULTÉ NTIC

DJAMEL.SAIDOUNI@UNIV-CONSTANTINE2.DZ

0559082425

CONCEPT DE TEMPS VIRTUEL

Le temps virtuel:

- Dans les systèmes qualifiés de temps-réel, le temps physique constitue le point de référence commun à l'ensemble des processus leur permettant de se synchroniser entre eux et avec l'environnement de façon à réaliser le contrôle voulu. Cette utilisation du temps étendue au temps logique a donné naissance à la notion de ***temps virtuel***.
- Dans un contexte de temps virtuel il existe donc une horloge virtuelle, globale à l'ensemble des processus, qui permet d'organiser le calcul réparti et de contrôler l'évolution du système.

Caractéristiques:

- A tout instant de l'exécution du programme réparti l'horloge virtuelle hv offre la même date à tous les processus.
- Les processus peuvent faire intervenir des dates virtuelles dans leurs calculs et assujettir leur progression à hv.

CONCEPT DE TEMPS VIRTUEL

CARACTÉRISTIQUES (SUITE)

- Les processus ont à leur disposition la primitive:

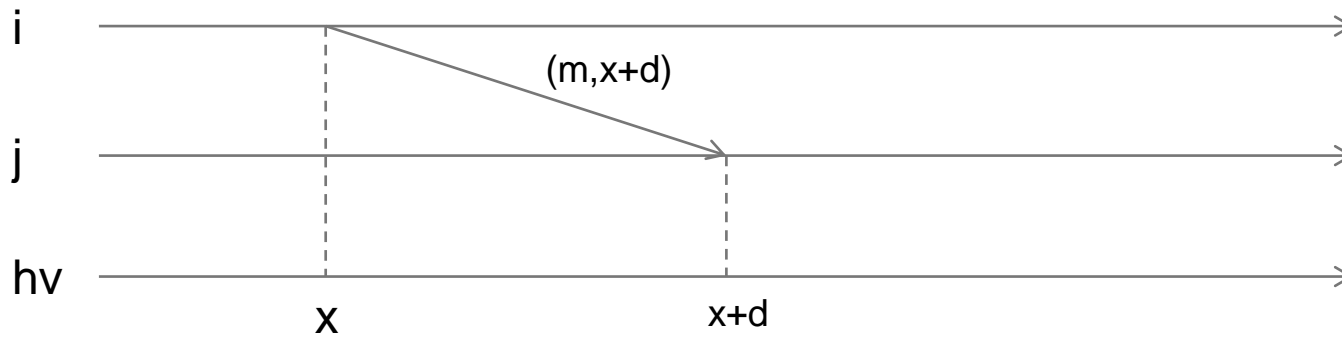
$\text{Attendre}(h_v = x)$

qui bloque leur progression jusqu'à ce que l'horloge h_v ait atteint la valeur x

- Les communications entre processus se font dans le temps virtuel. L'émetteur i d'un message précise la durée, en unités de temps virtuel, de transfert de ce message. Ainsi si à la date virtuel x le site i envoie un message m à j en précisant la durée de transfert d (d positif), ce message sera délivré au site j exactement à la date $x+d$. De plus i envoie un message avec date d'arrivée y ($y=x+d$). Les messages futurs qu'il enverra devront avoir une date d'arrivée supérieure ou égale à y . Les messages sont estampillés par leurs dates d'arrivée. Ce type de communication est qualifié de ***synchrone***.

CONCEPT DE TEMPS VIRTUEL

CARACTÉRISTIQUES (SUITE)



- Un message $(m, x+d)$ est donc disponible chez son destinataire j à la date $x+d$; celui-ci peut le consommer dès sa réception ou plus tard à une date supérieure à $x+d$.
- Contrairement aux communications, les processus sont asynchrones.
- Toutes interactions entre processus se font au moyen de messages estampillés. Le temps virtuel constitue ainsi le seul mécanisme permettant aux processus de se synchroniser.

CONCEPT DE TEMPS VIRTUEL

CARACTÉRISTIQUES (SUITE)

- Outre la primitive d'attente, les processus ont à leur disposition les deux primitives:

Attendre msg(m,dc)

Et attendre msg(m,dc) ou (hv=y)

La première permet d'attendre un message. La variable dc indique la date à laquelle l'attente a pris fin, c'est-à-dire la date de la consommation du message par son récepteur j.

Si $x+d$ est sa date d'arrivée sur j on a $dc \geq x+d$.

Ainsi si dans la figure (a) le code de j est :

Attendre(hv = x)

Attendre msg(m,dc)

On aura $dc = y$; dans la figure (b) on aura $dc = x+d$

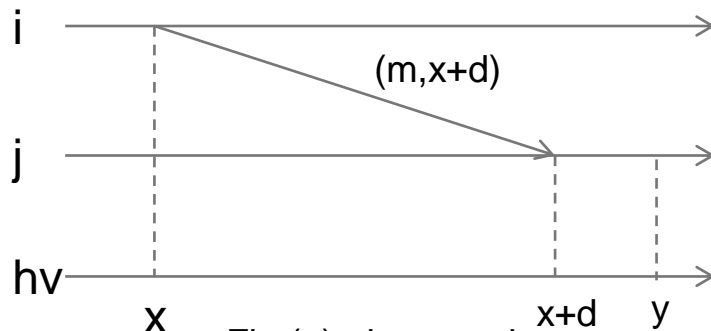


Fig.(a): $dc=y > x+d$

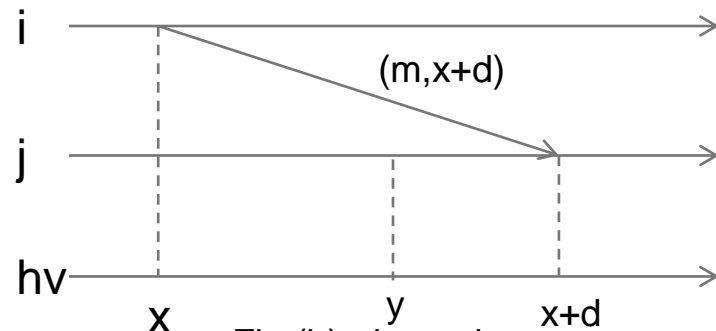


Fig.(b): $dc=x+d > y$

CONCEPT DE TEMPS VIRTUEL

INTÉRÊTS

Remarques:

- Les seules opérations qui, pour un processus donné, peuvent consommer du temps sont donc l'attente d'un message ou l'attente d'une date virtuelle; ce sont donc ces opérations qui font progresser le temps virtuel.
- Le concept de temps virtuel permet d'offrir un système réparti **logiquement synchrone** sur lequel on peut facilement concevoir et implémenter des applications utilisant ce type de synchronisme.
- **Le temps simulé** constitue l'exemple type du temps virtuel: il s'agit d'un temps logique global au processus décrivant le modèle de simulation et qui règle leurs évolutions et leurs synchronisations.
- **Dans les systèmes transactionnels** utilisant un contrôle de la concurrence fondé sur l'estampillage, le temps logique construit par ce mécanisme d'estampillage constitue un temps virtuel doté des caractéristiques indiquées précédemment.
- Indépendamment de son utilité immédiate pour résoudre un problème particulier, le temps virtuel est également intéressant par les problèmes que soulève sa mise en œuvre. Les solutions à ces problèmes peuvent exportées pour résoudre d'autres problèmes fréquemment rencontrés dans l'algorithmique distribué.

LES PROBLÈMES DE MISE EN OEUVRE

PROBLÈME À RÉSOUDRE

- Implémentation du système synchrone offert par le temps virtuel sur un système réparti asynchrone.
- Les divers sites ont à priori des vitesses de calcul différentes et les temps de transfert effectifs des messages sont quelconques mais finis.
- Une solution à ce problème doit vérifier les propriétés suivantes:
 - i. **Propriété de sûreté:**
 - a. **Synchronisme des sites:** Tous les sites ont la même perception logique de l'horloge virtuelle.
 - b. **Synchronisme des communications:** Un message estampillé z est délivré à la date virtuelle z à son destinataire (ce dernier peut ne le consommer que plus tard, cela dépend de son calcul).
 - ii. **Propriété de vivacité:** Si, du point de vue de l'application qui l'utilise, le temps virtuel peut progresser, alors il progressera.

LES PROBLÈMES DE MISE EN OEUVRE

CONTRAINTE DE COHÉRENCE

- **Solution centralisée:** Les propriétés précédentes peuvent facilement être obtenues aisément avec un contrôle centralisé, un site jouant le rôle de coordinateur central; ce site fait progresser h_v et en avertit les autres sites qui l'informent de leurs attentes. Cette solution n'est pas performante.
- **Solution distribuée:** Offrir le maximum de parallélisme possible en autorisant le plus d'asynchronisme entre les sites tout en garantissant les propriétés de sûreté et de vivacité. Pour cela:
 - Associer à chaque site i une horloge locale h_{vi} qui va représenter localement h_v .
 - Lorsqu'un site i exécute attendre ($h_v=x$) il suffit de recalculer $h_{vi} := x$ et de laisser continuer son calcul.
 - Le seul problème qui reste à résoudre est d'assurer le synchronisme des communications. Les messages sont physiquement délivrés sur un site dans un ordre qui ne correspond pas forcément à celui de leurs estampilles (qui sont des dates virtuelles d'arrivée)

LES PROBLÈMES DE MISE EN OEUVRE

CONTRAINTE DE COHÉRENCE (SUITE)

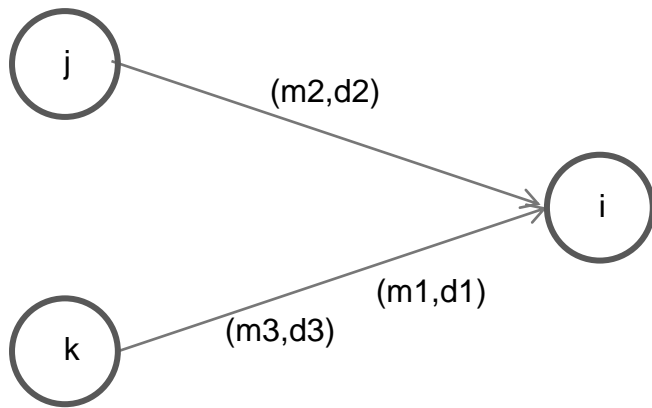
- Il faut donc assurer la contrainte de cohérence suivante:

CC = « *Les messages délivrés à un site le sont dans l'ordre non-décroissant de leurs estampilles* »

- Cette contrainte exprime la cohérence dans le temps virtuel, c'est à dire que le futur ne peut y influencer le passé.
- **Exemple:** Considérons le site i destinataire des deux messages:
 $m(j)$ émis par j , date virtuelle d'arrivée $d(j)$
 $m(k)$ émis par k , date virtuelle d'arrivée $d(k)$
avec $d(j) < d(k)$. Appelons $tr(j)$ (resp. $tr(k)$) le calcul effectué par i lorsqu'il traite le message $m(j)$ (resp. $m(k)$). Si CC est respectée le site i exécute :
 A la date virtuelle $d(j)$: le traitement $tr(j)$
 A la date virtuelle $d(k)$: le traitement $tr(k)$
 $Tr(k)$ peut accéder des variables d'état de i modifiées par $tr(j)$.

L'INTERBLOCCAGE ET SES CLASSES DE SOLUTIONS

Le respect de la contrainte de cohérence sur chaque site permet de garantir le synchronisme des communications dans le temps virtuel, c-à-d la propriété de sûreté, mais ne permet pas de garantir la vivacité: Un interblocage peut survenir alors que le temps virtuel pourrait progresser.



On suppose que $d1 < d2 < d3$. Afin de respecter la contrainte CC, le site i dans le temps virtuel **reçoit et consomme m1** à la date virtuelle **d1**, puis **reçoit et consomme m2** à la date virtuelle **d2**. Cependant i **ne peut pas recevoir et consommer m3** ! En effet le site j peut envoyer un message à i avec une date d'arrivée $d2'$ tel que $d3 > d2' > d2$

Le respect de la contrainte CC peut donc introduire un interblocage qui n'existe pas au niveau du programme réparti utilisant le temps virtuel.

L'INTERBLOCAGE ET SES CLASSES DE SOLUTIONS (SUITE)

Le respect de la contrainte CC peut donc introduire un interblocage qui n'existe pas au niveau du programme réparti utilisant le temps virtuel.

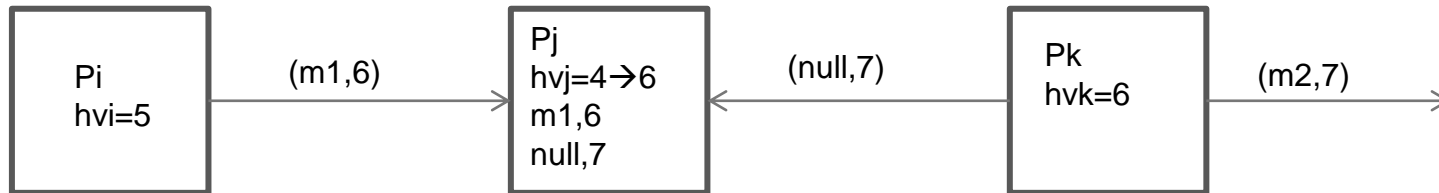
Afin d'éliminer ces situations et de garantir la propriété de vivacité deux classes de solutions sont envisageables.

- **La méthode pessimiste:** Cette méthode évite l'interblocage en rajoutant des messages de contrôle qui vont transporter des dates virtuelles pour faire progresser le système.
- **La méthode optimiste:** Dans cette méthode les messages sont reçus et délivrés à l'application dans leur ordre d'arrivée physique. Ceci peut conduire à violer la contrainte CC; si tel est le cas, le site récepteur « défait » son calcul pour revenir le reprendre de façon à respecter la contrainte CC.

LA MÉTHODE PESSIMISTE DE CHANDY ET MISRA

INTRODUCTION DE MESSAGES DE CONTRÔLE

- L'idée de base sur laquelle repose cette stratégie de prévention de l'interblocage consiste à rajouter des messages de contrôle afin de faire progresser le temps virtuel sur un site; ces messages sont appelés messages **null**. De plus les canaux de communication sont supposés **fifo**.
- Soit une partie du graphe des communications d'un programme réparti utilisant le temps virtuel. Le site i peut envoyer des messages aux sites j_1 et j_2 .



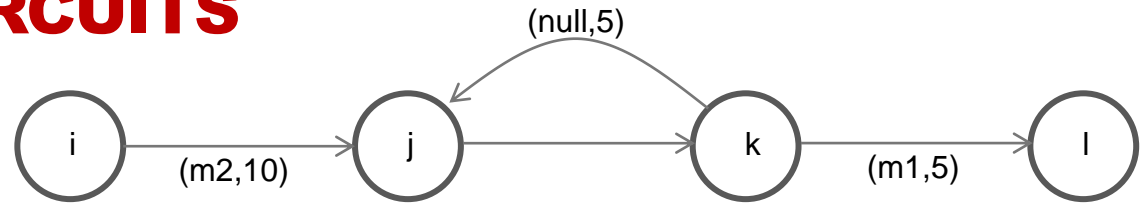
Lorsque p_k envoie un message (m,y) à un site, p_l par exemple, il envoie un message de contrôle $(null,y)$ sur chacun des autres canaux sortants, ici vers p_j . Lorsque ce message sera dans le tampon d'entrée de p_j relatif au canal (j,k) , le site p_j saura que le site p_k ne lui enverra désormais que des messages avec des estampilles supérieures ou égales à y .

Ainsi, si dans le tampon de p_j il y a des messages d'estampilles inférieures à y , ceux-ci peuvent être délivrés au site p_j .

LA MÉTHODE PESSIMISTE DE CHANDY ET MISRA

PROBLÈME DES CIRCUITS

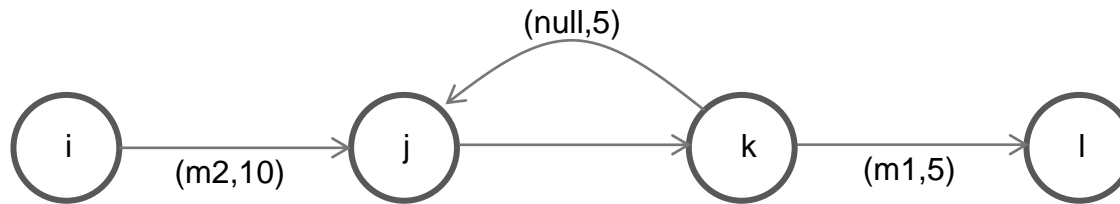
Soit le graphe des communications:



- Au temps virtuel 5 le site k envoie un message $(m1,5)$ vers l et un message de contrôle $(null,5)$ vers le site j.
- Au temps virtuel 10, le site i émet un message $(m2,10)$ vers le site j.
- Le site j qui est en attente de message et qui a deux canaux entrants peut se faire délivrer le message daté de la plus petite estampille, ici $(null,5)$, de façon à garantir le respect de la contrainte CC.
- Le site j peut alors faire progresser hvj à 5, la date d'arrivée du message délivré mais n'exécute aucun calcul puisqu'il ne s'agit pas d'un message de l'application mais d'un message de contrôle.
- Après cette consommation de $(null,5)$ on se retrouve dans la situation décrite par la figure, conduisant à un interblocage si le site k ne renvoie plus de messages.
- Pour résoudre ce problème il faut disposer d'une information supplémentaire: Dans tout circuit du graphe des communications il faut qu'il y ait au moins un processus qui puisse faire une prévision sur son comportement futur.

LA MÉTHODE PESSIMISTE DE CHANDY ET MISRA

PROBLÈME DES CIRCUITS



Supposons que p_j peut prédire son comportement futur.

- 1) k envoie $(m1,5)$ à l et $(null,5)$ à j .
- 2) j sait que k ne lui enverra pas de message avant 5 ($h_{vj} = 5$).
- 3) j reçoit $(m2,10)$.
- 4) j sait qu'il n'enverra rien avant δ .
- 5) j envoie $(null, 5 + \delta)$ à k .
- 6) k sait qu'il n'enverra rien avant $5 + \delta'$.
- 7) k envoie $(null, 5 + \delta')$ à j .
- 8) $5 + \delta'' \dots > 10 \rightarrow j$ accepte $(m2,10) \rightarrow$ Plus d'interblocage

MÉTHODE OPTIMISTE DE JEFFERSON

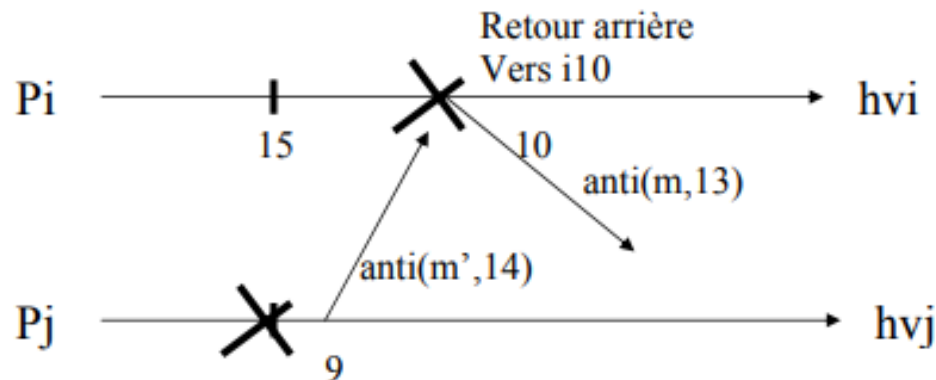
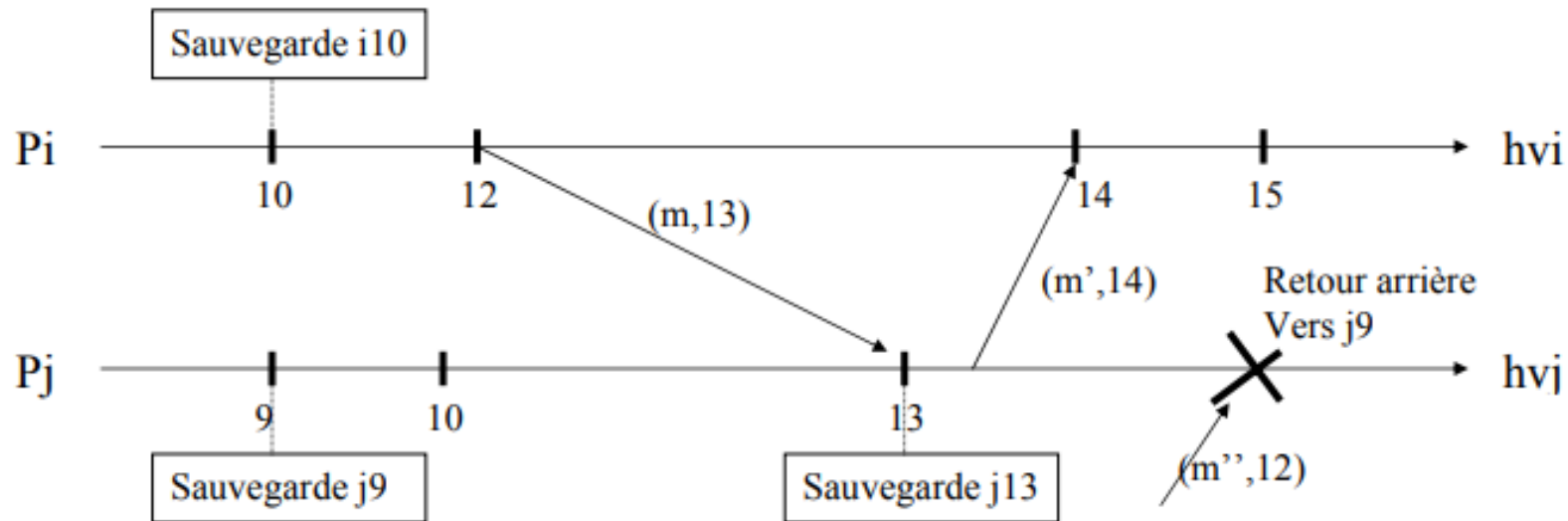
PRINCIPE :

- Elaboration d'une sauvegarde des états locaux des processus :
 - Ensemble des valeurs des variables du processus à un instant donné.
 - Une sauvegarde des messages émis et reçus. f
 - Une prise en compte immédiate d'un message (m, hv) dès qu'il arrive. f
 - Si un message (m', hv') arrive avec $hv' < hv$: Exécution d'un retour arrière de l'évolution du processus : f
 - Recalage de l'état du processus sur l'état sauvegardé précédent ou égal à hv' f
 - Annulation des messages émis depuis hv' , par l'envoi d'un anti message. f
- La réception de l'anti-message d'un message déjà consommé provoque un retour arrière du récepteur à la date du message.

MÉTHODE OPTIMISTE DE JEFFERSON

PRINCIPE :

♦ Exemple :



- Compromis # sauvegarde \leftrightarrow précision des décalage (temps)
- Quelles sauvegardes peuvent être effacées ?

LE TEMPS VIRTUEL GLOBAL

- Le TVG est la borne inférieure sur les dates de retour arrière possibles

$$\text{TVG} = \min \left(\begin{array}{l} \{hvi: 1 \leq i \leq n\} \\ \cup \\ \{t : \text{un message } (m,t) \text{ est en transit}\} \end{array} \right)$$

- Le TVG est calculé à l'aide d'un algorithme réparti d'observation (Chapitre précédent).