

Chapitre 07 : Construction d'un temps virtuel

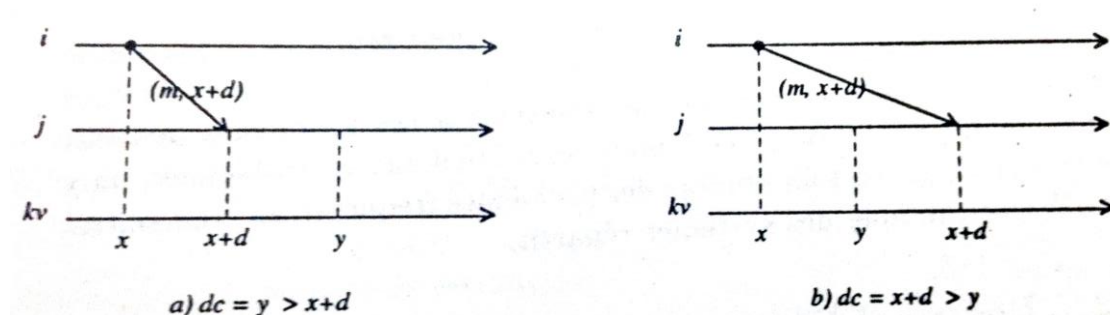
I. Concept de temps virtuel

Dans un contexte de temps virtuel, il existe une horloge virtuelle globale à l'ensemble des processus, qui permet d'organiser le calcul réparti et de contrôler l'évolution du système.

Le concept du temps virtuel permet d'offrir un système réparti logiquement asynchrone

Principe :

- l'émetteur d'un message précise la date virtuelle à laquelle celui-ci doit arriver (m, t) .
- Le récepteur connaît la date virtuelle à laquelle est arrivé le message.



Problème à résoudre :

- Gestion de l'évolution de l'horloge virtuelle
- La cohérence temporelle (sûreté)
 - Tous les processus ont la même perception de l'horloge virtuelle
 - Un message estampillé à la date z doit être consommé à la date z

La vivacité : Si du point de vu d'un processus, le temps virtuel peut progresser, il progressera.

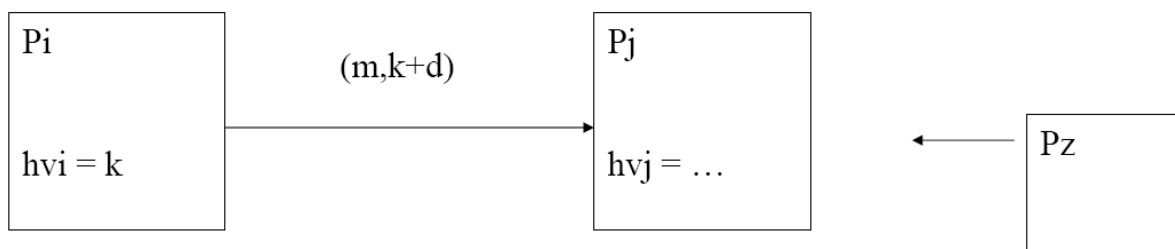
Pour un processus donné, les seules opérations qui peuvent consommer le temps virtuel sont :

attendre msg(m , dc) : l'attente d'un message m qui devra être consommé à la date dc

attendre($hv = x$) : l'attente d'une date virtuelle x

Solution décentralisée :

- Chaque processus possède une image de l'horloge virtuelle
- Quand un message est consommé, le processus récepteur met son horloge à jour.

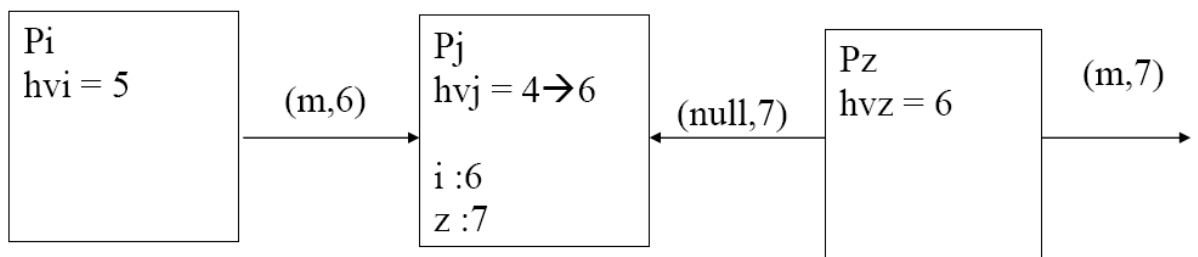


Problème d'interblocage :

- Le processus j doit-il consommer le message (m,k+d), ou existe-t-il un message arrivant de z à une date inférieure. J est bloqué et se met en attente.
- z, peut-être dans le même cas, à cause d'autres processus : interblocage.

II. Méthode pessimiste de Chandy et Misra

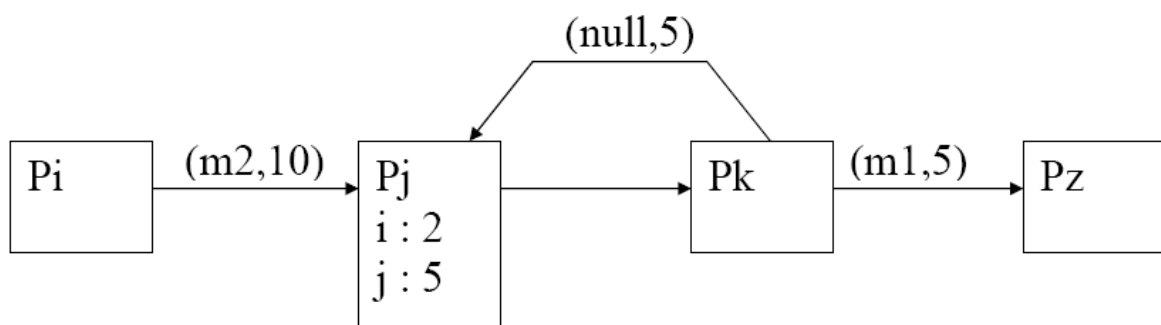
- Quand un processus n envoie un message (m,y) à un processus p, il envoie également un message (null,y) aux autres processus.
- Lorsqu'un processus j reçoit (null,y) d'un processus z, il sait que ce dernier ne lui enverra pas de message avant y.
- Chaque processus connaît donc la date minimale à partir de laquelle chacun des processus entrant peut lui transmettre un message. Il est donc capable de décider du prochain message à accepter et de la valeur de son hv.



- Ici : Si Pj ne possède que deux liaisons entrantes (en provenance de Pi et de Pz), il peut consommer (m,6) et positionner hvj à 6.

Interblocage et méthode pessimiste :

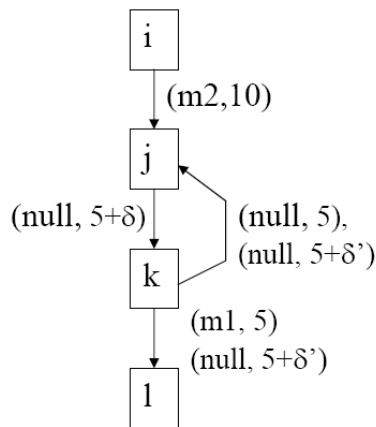
- Pk peut recevoir des messages de Pj qui peut en recevoir de Pk



- 1) k envoie (m1,5) à z, (null,5) à j.
- 2) j reçoit (m2,10), il consomme (null,5) et met hvj = 5.
- 3) j met (m2,10) en attente car il attend le prochain message de k (pour faire un choix).
- 4) k attend un message de j (qui est en attente de k), conditionnant le prochain message qu'il enverra → Interblocage.

Méthode pessimiste et anticipation.

- A la réception de (null, y) , un processus sachant (par anticipation) qu'il n'enverra rien avant $y + \delta$, envoi sur toute ses liaisons $(\text{null}, y + \delta)$



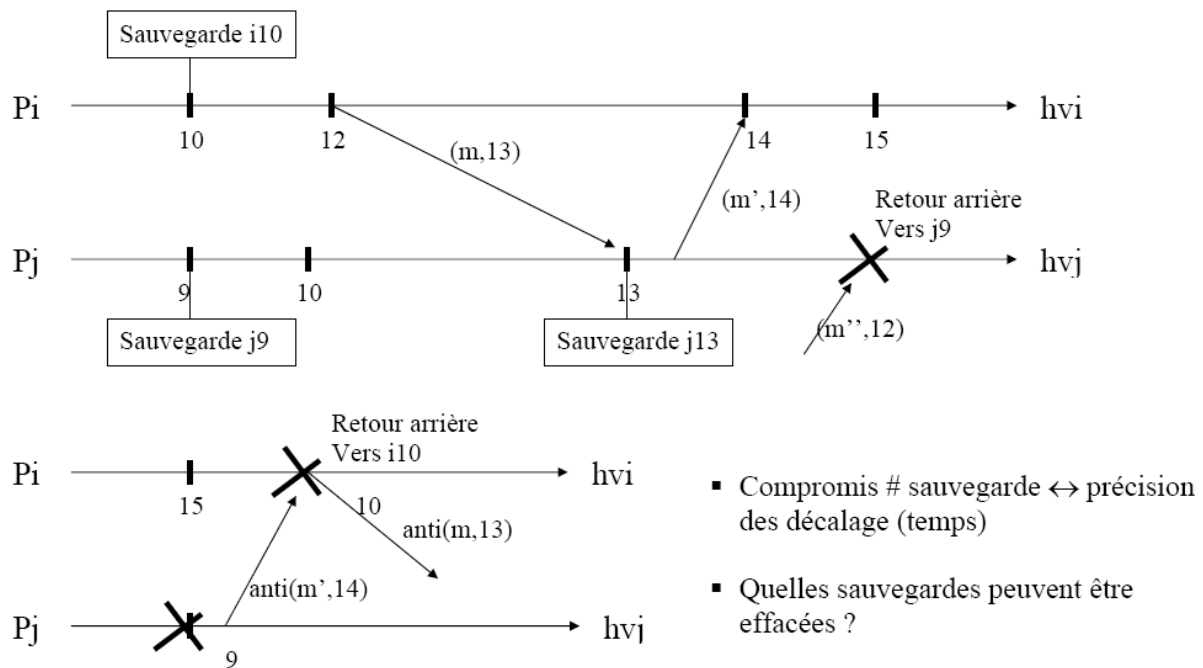
- 1) k envoi $(m1, 5)$ à l et $(\text{null}, 5)$ à j.
- 2) j sait que k ne lui enverra pas de message avant 5 ($hvj = 5$).
- 3) j recoit $(m2, 10)$.
- 4) j sait qu'il n'enverra rien avant δ .
- 5) j envoi $(\text{null}, 5 + \delta)$ à k
- 6) k sait qu'il n'enverra rien avant $5 + \delta'$
- 7) k envoi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

III. Méthode optimiste de Jefferson

Principe (faire, défaire, et refaire):

- 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.
- Une prise en compte (consommation) immédiate d'un message (m, hv) dès qu'il arrive (donc éliminer les possibilités d'interblocage).
- Si un message (m', hv') arrive avec $hv' < hv$: Exécution d'un retour arrière de l'évolution du processus :
 - Recalage de l'état du processus sur l'état sauvegardé précédent ou égal à hv'
 - Annulation des messages émis depuis hv' , par l'envoi d'un anti message.
- La réception de l'anti-message d'un message déjà consommé provoque un retour arrière du récepteur à la date du message.

Exemple



IV. Temps Virtuel Global (GVT)

Les sauvegardes peuvent très vite occuper tout l'espace mémoire disponible. Afin de pouvoir libérer une partie de la place correspondante à ces sauvegardes et de pouvoir réaliser les actions dites définitives, on introduit la notion de GVT.

Le GVT est la borne inférieure sur les dates de retour arrière possibles.

$$\text{GVT} = \min(\{h_{vi} : 1 \leq i \leq n\} \cup \{t : \text{un message } (m, t) \text{ est en transit}\})$$

Le GVT est calculé à l'aide d'un algorithme réparti d'observation (chapitre 6)