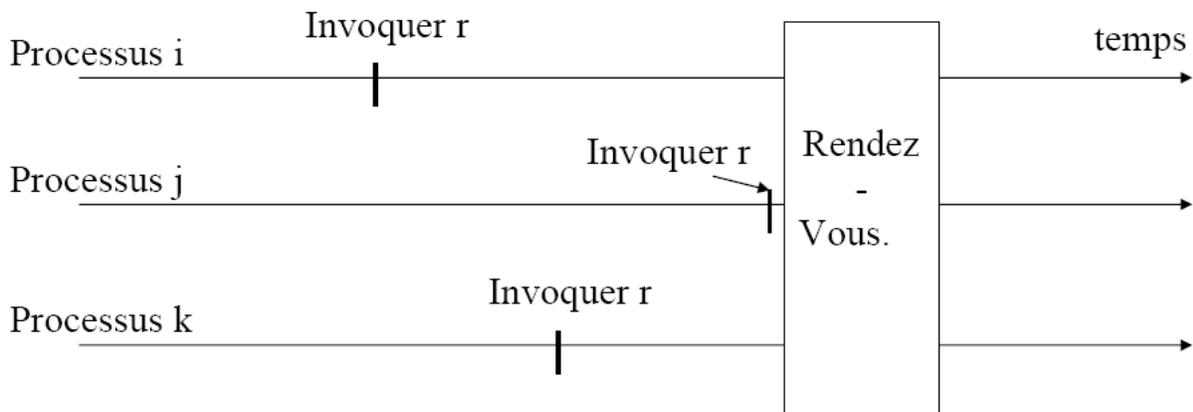


Coordination par rendez-vous

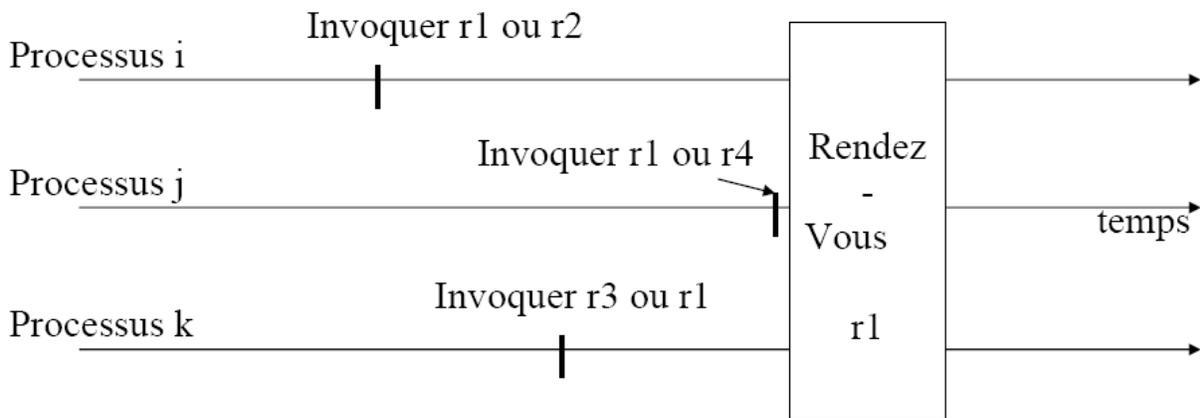
1. Concept de rendez-vous

Un rendez-vous est caractérisé par la présence simultanément d'un certain nombre de processus à un point de contrôle qui a été défini au préalable.

Un rendez-vous r est défini par l'ensemble des processus (sites) concernés par r : $r = \{i,j,k\}$



2. Choix non déterministe entre plusieurs rendez-vous



Si plusieurs rendez-vous sont possibles, le processus s'engage dans un (quelconque) et un seul d'entre eux.

Sûreté :

- Si à un instant donné un site i , qui invoque un rendez-vous r , s'y engage alors tous les sites qui interviennent dans r , l'ont aussi invoqué et s'y engagent également.
- Un site ne peut engager que dans un rendez-vous à la fois.

Vivacité : si des rendez-vous qui ont été invoqués sont possibles, alors aux moins l'un d'entre eux aura lieu.

3. Spécification de Bagrodia

Associer à chaque site i deux compteurs:

- ✓ w_i : nombre de fois où i a été demandeur
- ✓ n_i : nombre de fois où i est entré en rendez-vous

Invariants : chaque processus ne s'engage que dans un rendez-vous à la fois :

$$\forall i : n_i \leq w_i \leq n_i + 1$$

Un rendez-vous r ne peut avoir lieu que si tous les sites sont prêts (ont demandé l'invocation de r) :

$$\forall r : (\forall i \in r : w_i = n_i + 1) \Leftrightarrow r \text{ peut avoir lieu.}$$

4. Principe de la solution distribuée pour k rendez-vous

- Utiliser un certain nombre de contrôleurs CTL_1, CTL_2, \dots placés sur différents sites.
- Un contrôleur CTL_j peut gérer plusieurs rendez-vous (rdv_ctl_j).
- Lorsqu'un site i invoque un rendez-vous r , il envoie une requête (message prêt(i)) à tous les contrôleurs qui gèrent les rendez-vous du site i (ctl_par_i).
- Afin d'assurer qu'un seul contrôleur puisse décider à la fois, on doit utiliser un mécanisme d'exclusion mutuelle entre les contrôleurs. Pour ce faire, on utilise un jeton qui circule entre les contrôleurs, placés sur un anneau, et seul le contrôleur qui possède le jeton peut décider quel rendez-vous peut avoir lieu.
- Le jeton contient, pour chaque site, le nombre de fois d'engagement dans les rendez-vous invoqués ($nx(i)$).

- Si pour un site i , le nombre d'invocation ($nbpret(i)$) est supérieur au nombre d'engagements ($nbpret(i) = nx(i) + 1$) alors le site i est un demandeur et est en attente de la réponse d'un contrôleur.
- Si un rendez-vous a lieu (tous les sites appartenant à r sont prêts : $\forall i \in r : nbpret_j(i) := nx(i) + 1$), alors le contrôleur informe tous les sites concernés par l'envoi d'un message $rdv_ok(r)$ qu'ils peuvent engager dans r .

Comportement d'un site i

Lors de l'invocation de (r_1 ou r_2ou r_k)

$etat_i := demandeur ;$

$\forall CTL_j \in ctl_par_i : \text{envoyer } prêt(i) \text{ à } CTL_j ;$

$w_i := w_i + 1 ;$

Lors de la réception de $rdv_ok(r)$:

$etat_i := dedans ;$

$n_i := n_i + 1 ;$

Comportement d'un Contrôleur j

Lors de la réception de $prêt(i)$:

$nbpret_j(i) := nbpret_j(i) + 1 ;$

Lors de la réception de jeton :

$\forall r \in rdv_ctl_j$ **faire**

si ($\forall i \in r : nbpret_j(i) := nx(i) + 1$) **alors**

$\forall i \in r$ **faire envoyer** $rdv_ok(r)$ **à** i ;

$nx(i) := nx(i) + 1 ;$

fait

fsi

fait

envoyer jeton au contrôleur suivant ;

Exemple

