

Séquence 2 - Détection d'interblocage Jean.Saquet@info.unicaen.fr

1 Définition du problème

Un système réparti est constitué de processus communicants au moyen de lignes bidirectionnelles fiables. On s'intéresse au problème de l'interblocage de tout ou partie des processus du système, et à sa détection.

On notera $G=(X,E)$ le graphe orienté dont l'ensemble des sommets X est l'ensemble des processus du système et E l'ensemble des arcs défini par :

Si i et j sont deux éléments de X , il existe un arc de i vers j si et seulement si le processus i est en attente d'un événement ne pouvant être produit que par le processus voisin j (on dira alors que j est un successeur de i dans ce graphe G).

Attention, ce graphe n'est pas le graphe représentant le réseau. Les sommets sont les mêmes, mais pas les arcs. On remarquera que :

- Si j est successeur de i , il existe un lien de communication entre i et j .
- Un processus peut avoir plusieurs successeurs, dans le cas où il est en attente d'un des événements pouvant être produit par un des processus voisins, de manière non déterministe.
- Si un processus est dans un état dont il peut sortir par un événement local (par exemple débordement d'un timer), il n'a pas de successeur dans le graphe G .

On supposera que tout processus i sait calculer à chaque instant l'ensemble $S(i)$ de ses successeurs.

Enfin, on dira qu'un ensemble B (inclus dans X) de processus est un ensemble de blocage si et seulement si :

- $B \supseteq S(B)$ (où $S(B)$ est la réunion des $S(i)$ pour tous les i appartenant à B)
- $\forall i \in B, S(i) \neq \emptyset$

Intuitivement, cela signifie que tous les éléments de B sont en attente d'un événement ne pouvant être produit que par un autre élément de B .

2 Une autre formulation des ensembles de blocage

La définition ci-dessus a le mérite d'être assez intuitive, mais n'est par contre pas facilement utilisable dans un environnement réparti. En effet, bien que chaque site puisse évaluer l'ensemble de ses successeurs, la vérification de $B \supseteq S(B)$ impose de globaliser les informations de tous les éléments de B , et on ne connaît pas B a priori (sinon le problème est résolu).

On va donc essayer de caractériser le fait qu'un site donné i puisse appartenir à un ensemble de blocage, avec des propriétés facilement évaluables par un algorithme réparti. On note S^* la fermeture réflexive et transitive de S . En particulier, si i est un élément de X , $S^*(i)$ est l'ensemble constitué de i lui-même et de ses descendants (successeurs immédiats et descendants des successeurs).

\Rightarrow 1. Montrer que i appartient à un ensemble de blocage si et seulement si l'ensemble $S^*(i)$ ne contient pas de sommet sans successeur.

3 Algorithme réparti de détection d'interblocage

Le but est, pour un processus donné (noté *init*) du système, de savoir s'il fait partie d'un ensemble de blocage ou non.

Principe de l'algorithme : le processus *init* va lancer un parcours du graphe de ses descendants, et va collecter une information lui permettant de répondre à cette question. L'information collectée sera ici un simple booléen lui indiquant s'il fait partie d'un ensemble de blocage (FAUX = il n'y a pas de blocage).

Ce parcours de collecte est assez "standard" :

- Un processus qui reçoit le message de parcours pour la première fois, et qui n'a pas de successeur, va renvoyer à son père un acquittement avec le paramètre FAUX.
- Tout processus collecte les réponses de ses successeurs, lorsqu'il en a, et envoie une réponse à son père qui est la conjonction (ET) des réponses reçues.
- Un processus qui reçoit un message de parcours qu'il avait déjà reçu renvoie un acquittement avec VRAI, élément neutre de la conjonction.

⇒ **2.** *Écrire les détails de cet algorithme réparti.*