L'EXCLUSION MUTUELLE DANS UN ENVIRONNEMENT DISTRIBUÉ

ALGORITHMES À PERMISSIONS D'ARBITRES

SAIDOUNI Djamel Eddine

Université Constantine 2 - Abdelhamid Mehri
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'Informatique Fondamentale et ses Applications

Laboratoire de Modélisation et d'Implémentation des Systèmes Complexes

Djamel.saidouni@univ-constantine2.dz
saidounid@hotmail.com
Tel: 0559082425

RAPPEL

$\forall \boldsymbol{i \neq j, R_i \cap R_j \neq \emptyset}$

Cas centralisé: $\forall \boldsymbol{i \neq j, R_i \cap R_j}$**={k}**

Cas distribué: Algorithme réparti symétrique dans lequel chaque site joue un rôle équivalent aux autres.

- Pour un site i, soit $\boldsymbol{CR_i = \{j\ telque\ i \in R_j\}}$: Ensemble des sites qui demandent la permission à i.

Les critères de distribution sont définis par:

- **(c1):** $\forall \boldsymbol{i, |R_i| = K}$ constante ⇔ Tous les sites doivent demander et obtenir le même nombre de permissions pour pouvoir entrer en section critique (règle du même effort).
- **(c2):** $\forall \boldsymbol{i, |CR_i| = D}$ constante ⇔ Chaque site joue un rôle d'arbitre pour le même nombre de sites (règle de la même responsabilité)

L'ALGORITHME DE MAEKAWA (ALGORITHME MIXTE)

- Tout site i gère une file d'attente $\boldsymbol{file_i}$ dans laquelle il place les estampilles des requêtes qu'il reçoit (issues des sites de CR_i) et celle de sa propre requête s'il a fait une demande.
- $\boldsymbol{file_i}$ est triée dans l'ordre croissant des estampilles.
- Le site i peut entrer dans sa SC s'il reçu toutes les permissions attendues des sites de R_i (il est alors prioritaire par rapport à eux) et que l'estampille de sa requête est en tête de $file_i$ (il est alors prioritaire par rapport aux sites de CR_i).
- Lorsqu'un site sort de sa SC il renvoie des permissions pour chaque requête qui est dans $file_i$ (sites de CR_i qui lui en ont demandé), et retourne celle qu'il avait obtenues aux sites de R_i à l'aide de messages $retourperm$.

L'ALGORITHME

Lors de la réception de $requête(K, j)$

$H_i = \max(H_i, K)$;
$Priorité_i = (état_i = dedans)$
ou
$\big((état_i = demandeur)\ et\ (Last_i, i) < (K, j) \big)$;
Si $not\ Priorité_i$ **Alors *envoyer*** $permission(H_i, i)$ à j **Finsi** ;
insérer (K, j) dans $file_i$;

Lors de la réception de $permission(K, j)$

$H_i = \max(H_i, K)$;
$attendus_i = attendus_i - \{j\}$

Lors de la réception de retour$perm(K, j)$

$H_i = \max(H_i, K)$;
$supprimer((*, j))\ de\ file_i$

PROPRIÉTÉS DE L'ALGORITHME

Nombre de message de contrôle $= 3 * |R_i| \simeq 3 * K$

Temps durant lequel la SC est libre alors qu'il y a des demandeurs $= 3 * T$

Bornétude des variables : Horloges non bornées

Algorithme non adaptatif

Syntaxiquement l'algorithme ressemble à ceux à permissions d'arbitre, cependant, sémantiquement il est à permissions individuelles.