

Systemes répartis (suite)

Rappel - définition

- Définition générale

C'est un ensemble de processus communicants, répartis sur un réseau de machines le plus souvent hétérogènes, et coopérant à la résolution d'un problème commun.

- Quelques exemples :
 - Le plus connu : DNS, ...

La mesure du temps (1)

- Différentes mesures
 - Utilisation des horloges logiques
 - Horloge scalaire de Lamport
 - Horloge vectorielle
 - Horloge matricielle

But : Création d'un ordre

Utilisation de cet ordre pour résoudre des problèmes :

Exclusion mutuelle (Lamport, Ricart/Agrawala, Suzuki/Kasami)

Election (LeLann, Chang/Roberts)

Autres algorithmes (1)

- Exclusion mutuelle
 - **Algorithme de Kerry Raymond**
- Hypothèses : Les communications sont fiables et FIFO
Chaque nœud connaît ses voisins et on a un arbre recouvrant
- Chaque nœud a une FIFO, et 3 variables :
 - `jeton_present`,
 - `voisin` qui correspond au voisin se rapprochant du jeton,
 - `requete_en_cours`.
- **Un nœud peut rentrer en SC s'il a le jeton.**
- *Demande d'entrée en SC* :
 - si FIFO est vide et `jeton_present`=vrai, alors entrée en SC
 - sinon si `requete_en_cours` = faux
 - alors on envoie **REQUETE** à voisin, on insère « SELF » dans FIFO
 - et `requete_en_cours` <- VRAI

Autres algorithmes (2)

- Réception d'une **REQUETE** par *i*:
 - si **jeton_present** = vrai et hors section critique,
alors envoie **JETON** au demandeur et modification de **voisin**,
 - sinon **REQUETE** est inséré dans FIFO, et
 - si **requête_en_cours** = faux , **REQUETE** est envoyé à **voisin** et
requete_en_cours <- VRAI
- Réception de **JETON** par *i* ou sortie de la SC :
 - i* retire une requête *k* de FIFO, **requete_en_cours** <- FAUX
 - si *k*=*i*, *i* rentre en SC
 - sinon *i* envoie **JETON** à *k* et si FIFO non vide, alors *i* émet **REQUETE** à *k* et
requete_en_cours <- VRAI
 - modification de **voisin**

Nb de messages : Ricart = $2*(n-1)$, Susuki = n , Raymond = $\log(n)$

Autres algorithmes (1)

- Election
 - **Algorithme STP (Spanning Tree Protocol)**
 - Créé par Radia Perlmann
- Hypothèses : Les communications sont fiables et FIFO
Chaque nœud a un identifiant unique
- Création d'un arbre avec élection d'un chef.
- Utilisation de requête pour savoir qui sera élu

La mesure du temps (2)

- Différentes mesures
 - Utilisation des horloges synchrones
 - Une horloge globale au système
 - Utilisation d'un tick horloge
 - Chaque système peut avoir accès à la pulsation
 - 2 règles à respecter
 - Synchronisme des processus
 - Synchronisme des canaux

Comment transformer un système asynchrone en système synchrone ?

La mesure du temps (3)

- Différentes mesures

- Utilisation du temps physique

- Définition : Une horloge physique h_p est un dispositif permettant de bâtir une métrique du temps, c'est un compteur d'événements périodiques discrets caractérisé par une granularité.
 - Deux événements sont **indépendants** si dates non distinctes
(*problème de granularité*)
 - Sinon, relation de causalité, l'un précède l'autre.

- Mais, dérive obligatoire et constante de chaque horloge par rapport au temps de référence

coefficient : ρ

La mesure du temps (4)

- But : créer un référentiel de temps commun à tout le système
 - On a : chaque événement est associé à une date d'occurrence
(action i , temps $\rightarrow hp(i)$)
 - Cette date d'occurrence doit avoir une signification dans le système
- Création d'horloge C_i qui respectent les propriétés suivantes :
 1. Propriété de croissance
 $\forall i, \forall t, \forall d > 0, C_i(t+d) \geq C_i(t)$
 2. Propriété d'accord
 $\forall i, \forall j, \forall t, |C_i(t) - C_j(t)| < \epsilon$
 3. Propriété de justesse
 $\forall i, \forall t, \beta + (1 - \rho) t \leq C_i(t) \leq \beta + (1 + \rho) t$

La mesure du temps (5)

- Algorithme de synchronisation
 - Fonction d'ajustement : $C_i(t) = h_{pi}(t) + A(t)$
 - Calcul de $A(t)$??
 - Création d'une fonction linéaire par morceau

