Systèmes répartis (suite)

Rappel - définition

- Définition générale

C'est un ensemble de processus communiquants, 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, Susuki/Kasami)

Election (LeLann, Chang/Roberts)

Autres algorithmes (1)

- **Exclusion mutuelle**
  - **Algorithme de Kerry Raymond**

  But : avoir en permanence un arbre dont la racine possède le jeton.

- 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 requêtes 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 hp 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)

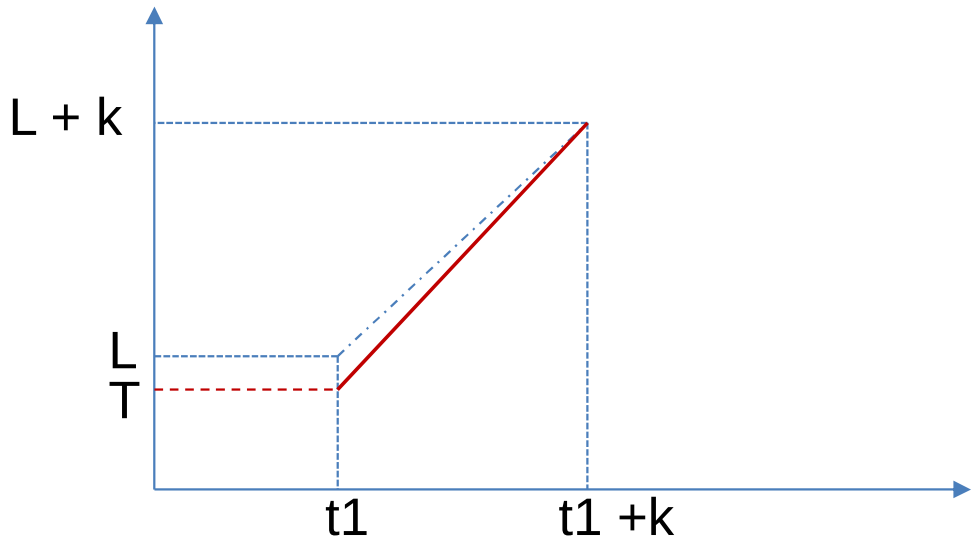
- But : créer un référentiel de temps commun à tout le système
  - chaque événement est associé à une date d'occurrence
    (action i, temps → hp(i))
  - Cette date d'occurrence doit avoir une signification dans le système

- <u>Création d'horloge Ci qui respectent les propriétés suivantes :</u>

  1. Propriété de croissance
     $\forall i, \forall t, \forall d > 0,\ Ci(t+d) \geq Ci(t)$
  2. Propriété d'accord
     $\forall i, \forall j, \forall t,\ | Ci(t) - Cj(t) | < \varepsilon$
  3. Propriété de justesse
     $\forall i, \forall t,\ \beta + (1 - \rho)\, t \leq Ci(t) \leq \beta + (1 + \rho)\, t$

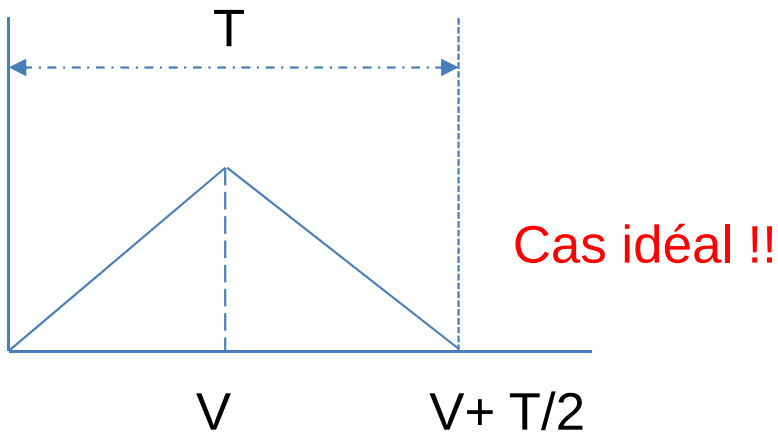
La mesure du temps (5)

- Algorithme de synchronisation
  - Fonction d'ajustement : Ci(t) = hpi(t) + A (t)
    - Calcul de A(t) ??
    - Création d'une fonction linéaire par morceau



Algorithme de Cristian

- Synchronisation basique
  - But : synchroniser une horloge à partir d'une horloge distante
  - Problème : temps de transfert inconnu
    - Solution : mesurer le temps d'aller-retour
      - D demande l'heure au Serveur, et reçoit la valeur V.
        D a mesuré le temps d'aller-retour T entre la demande et la réponse
        → temps_final = V + (T/2)



NTP (Network Time Protocol)

- Temps sur l'internet :
  - NTP4 -> RFC 5905
  - NTP3 -> RFC 1305
- Fournit l'heure UTC
  - Ne gère pas les fuseaux horaires
  - Ne gère pas les heures d'été et d'hiver
- Fonctionne sur le port 123 d'UDP
- Utilisation de l'algorithme de Marzullo
- Gestion du temps par des serveurs en strate
- Le temps est stocké sur 64 bits, début en janvier 1900
  - 32 bits pour les secondes, soit 136 ans
  - 32 bits pour la précision, 233 picosecondes