Algorithmique distribuée

Master 1 RID

Plan

- **Algorithmique distribuée**
  - Éléments de base d'un système distribué
  - Modèles conceptuels de systèmes distribués
    - Modèle d'interaction
    - Modèle de fautes
      - Franches
      - Par omission
      - Arbitraires ou byzantines
    - Détecteur de pannes
      - Classification
      - Réalisation

Algorithmique distribuée

- Si l'on veut développer des logiciels fiables et robustes dans un contexte distribué, il faut:
  - Tenir compte des spécificités des systèmes distribués ,Notamment les temps de communication et la multiplication des possibilités de pannes
  - Développer des algorithmes dédiés aux systèmes distribués : but de l'algorithmique distribuée

Algorithmique distribuée

- Développement d'algorithmes dédiés aux systèmes distribués et prenant en compte les spécificités de ces systèmes
- On y retrouve notamment des adaptations de problèmes classiques en parallélisme
  - Exclusion mutuelle, élection (d'un maître) ...
- et aussi des problèmes typiques des systèmes distribués
  - Horloge globale, état global, diffusion causale, consensus ...

Algorithmique distribuée

- S'intéresse principalement à deux grandes familles de problems
  - Synchronisation et coordination entre processus distants
  - Entente sur valeurs communes et cohérence globale dans un contexte non fiable (crash de processus, perte de messages ...)

Algorithmique distribuée

- Les algorithmes distribués s'appuie sur des caractéristiques du système
  - Caractéristiques des éléments formant le système
    - Fiable, pouvant se planter, pouvant envoyer des messages erronés ...
  - Caractéristiques de la communication
    - Fiable ou non fiable, temps de propagation borné ou pas...

Algorithmique distribuée

- Un algorithme distribué se base donc sur un modèle de système distribué qui caractérise
  - Les processus
  - La communication
  - Les pannes

Processus

- Élément logiciel effectuant une tâche, un calcul
  - Exécution d'un ensemble d'instructions
- Une instruction correspond à un événement local au processus
  - Dont les événements d'émission et de réception de messages
- Les instructions sont généralement considérées comme atomiques
- Il possède une mémoire locale
- Il possède un état local
  - Ensemble de ses données et des valeurs de ses variables locales
- Il possède un identifiant qu'il connaît
- Pas ou peu de connaissance des autres processus du système et de leur état
- Les processus d'un système s'exécutent en parallèle

Canal de communication

- Canal de communication point à point entre 2 processus.
- **Caractéristiques :**
  - **Uni ou bi-directionnel**.
  - **Fiable ou non** : perd/modifie ou pas des messages.
  - **Ordre de réception par rapport à l'émission.**
    - FIFO = les messages sont reçus dans l'ordre où ils sont émis.
  - **Synchrone ou asynchrone.**
    - **Synchrone :** l'émetteur et le récepteur se synchronisent pour réaliser l'émission et/ou la réception.
    - **Asynchrone :** pas de synchronisation entre émetteur et récepteur.
  - **Taille des tampons de message cotés émetteur et récepteur.**
    - Limitée ou illimitée

Canal de communication

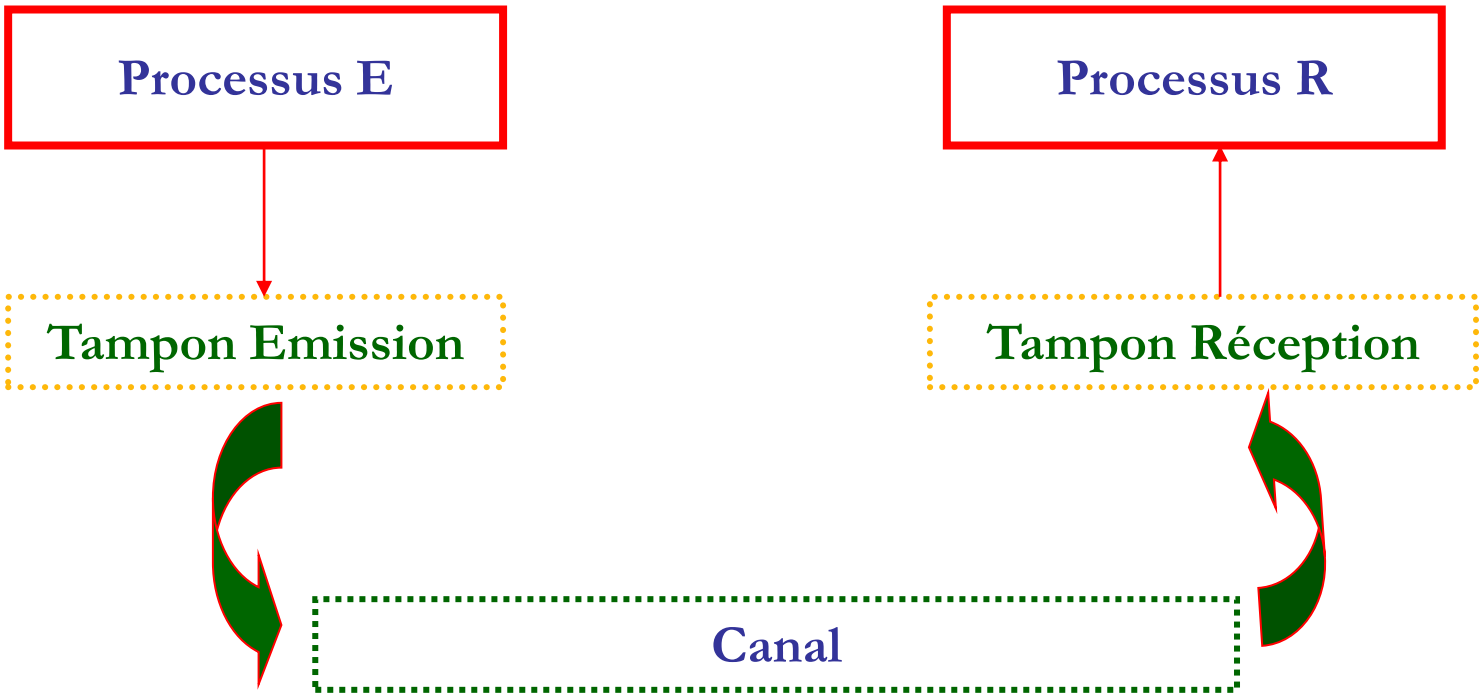
- Exemple : modèle des sockets TCP
  - **Fiable.**
  - **FIFO.**
  - **Bidirectionnel.**
  - **Asynchrone en émission.**
    - Emetteur n'est pas bloqué quand il émet quoique fasse le récepteur.
  - **Synchrone pour la réception.**
    - On reçoit quand l'émetteur émet.
    - Sauf si données non lues dans le tampon coté récepteur.

Canal de communication

- Performances d'un canal
  - **Latence** : délai entre l'émission d'un message et sa réception.
  - **Bande passante** : nombre d'informations transitant par unité de temps exprimé en bits/seconde.
  - **Gigue** : variation de temps pour délivrer des messages d'une série (variation de la latence).
    - Important dans le cadre des données multimédia nécessitant des synchronisations.
      - La voix sur IP
  - **Perte de messages :** due à des phénomènes de congestion sur le réseau.

Canal de communication

Liaison canal/processus (canal uni-directionnel)



Modèles conceptuels de systèmes distribués

- On modélise un système distribué selon plusieurs dimensions
  - Modèle d'interaction
    - Modèle de canal.
    - Modèle temporel.
      - Synchrone / Asynchrone
  - Modèle de fautes
  - Modèle de sécurité
    - Non traité dans ce cours

Modèles d'interaction

- **Contient un modèle de canal et un modèle temporel.**
- **Modèle de canal :** uni ou bi-directionnel, fiable ou non, ordre de réception par rapport à l'émission, synchrone ou asynchrone, taille des tampons de message cotés émetteur et récepteur.
- **Modèle temporel :** deux modèles principaux :
  - **Système distribué synchrone :** possède les 3 caractéristiques suivantes :
    - Le temps pour exécuter une étape du processus est compris entre une borne min et une borne max connues.
    - Un message émis est reçu dans un délai max connu.
    - L'horloge locale d'un processus dérive au plus d'une certaine durée connue et bornée du temps réel.
  - **Système distribué asynchrone :** il n'y aucune borne.
    - Sur la durée d'exécution d'une étape du processus.
    - Sur le délai de réception d'un message.
    - Sur la dérive de l'horloge locale d'un processus par rapport au temps réel.

Synchrone / Asynchrone

- **Un modèle synchrone** est un modèle où les contraintes temporelles sont bornées.
  - On sait qu'un processus évoluera dans un temps borné.
  - On sait qu'un message arrivera en un certain délai.
  - On connaît la limite de dérive des horloges locales.

- **Un modèle asynchrone** n'offre aucune borne temporelle.
  - Modèle bien plus contraignant et rendant impossible ou difficile la mise en œuvre de certains algorithmes distribués.

Modèle de fautes

- 4 grands types de fautes ou pannes
  - Franches : le processus ne fait plus rien
  - Par omission : des messages sont perdus ou non délivrés
  - Arbitraires ou byzantines : le processus renvoie des valeurs fausses et/ou fait « n'importe quoi »
  - Fautes temporelles

Processus correct : processus non planté, qui ne fait pas de fautes.

Fautes Franches

- Le processus ne fait plus rien.

- **Classification des fautes :**

  - **Crash-stop :** un processus s'arrête et reste hors-service.

  - **Crash-recovery :** un processus s'arrête mais il peut se relancer après un certain temps.

Fautes par omission

- Processus ou canal ne réalise pas une action.
- **Classification des fautes :**
  - **Omission du canal :** un message positionné dans un tampon d'émission d'un processus n'arrive jamais dans le tampon de réception du destinataire.
  - **Omission en émission :** un processus envoie un message mais il n'est pas placé dans le tampon d'émission.
  - **Omission en réception :** un message est placé dans le tampon de réception d'un processus mais le processus ne le reçoit pas.

Fautes arbitraires/byzantines

- Fautes quelconques et souvent difficilement détectables.
- **Processus :**
  - Calcul erroné ou non fait.
  - État interne faux.
  - Valeur fausse renvoyée pour une requête.
- **Canal :**
  - Message modifié, dupliqué voire supprimé.
  - Message « créé ».
  - En pratique peu de fautes arbitraires sur les canaux car les couches réseaux assurent la fiabilité de la communication.

Fautes temporelles

- Uniquement pour un système distribué synchrone : dépassement des bornes temporelles
  - Une étape du processus s'exécute en plus de temps que la borne prévue
  - L'horloge locale dérive d'une valeur supérieure à la borne prévue
  - Un message émis est reçu après un délai supérieur à la borne prévue

Détecteur de Fautes ou pannes

- **Définition :** élément associé aux processus «essayant» de déterminer l'état des autres processus.
  - Généralement, on se base sur une communication fiable.
    - Messages non perdus.
- **Classification des caractéristiques des détecteurs :**
  - **Complétude :** un processus fautif est soupçonné de l'être par un processus correct.
  - **Exactitude :** un processus correct n'est jamais soupçonné par aucun processus correct.
    - **Deux variantes à chaque fois :**
      - **Forte :** par tous les processus.
      - **Faible :** par au moins un processus.

Classification détecteurs de fautes

- **Complétude :**
  - **Forte :** tout processus fautif finit par être soupçonné **par tous les processus corrects.**
  - **Faible :** tout processus fautif finit par être soupçonné **par au moins un processus correct.**
- **Exactitude:**
  - **Forte :** aucun processus correct n'est jamais soupçonné par aucun processus correct
  - **Faible :** au moins un processus correct n'est jamais soupçonné par aucun processus correct

Réalisation de détecteurs de pannes : synchrone vs asynchrone

- **Système synchrone :**
  - Grace au délai borné de réception des messages, on sait qu'on recevra un message dans un certain délai si le processus est vivant.
  - **Peut construire un détecteur de pannes fiable (parfait).**
- **Système asynchrone :**
  - Temps de propagation des messages non bornés.
    - Difficulté (voire impossibilité) de différencier un message lent d'un processus mort.
  - **Détecteur parfait est non réalisable dans un système asynchrone.**
    - Mais souvent on peut se contenter de détecteurs imparfaits.