Algorithmes distribués

Master informatique
2011-2012

Stella MARC-ZWECKER
stella@unistra.fr

Plan prévisionnel du cours

- 1) Aspects algorithmiques des SD
- 2) Le temps logique (rappel ?)
- 3) Algorithmes d'exclusion mutuelle
- 4) Algorithmes d'élection
- 5) Etat global
- 6) Détection de l'interblocage et partage des ressources
- 7) Algorithmes de terminaison

$\Rightarrow$ Passage à l'évaluation continuée : modalités (exercices rendus, interro de synthèse, interro surprise, présence)

Chapitre 1

Aspects algorithmiques des systèmes répartis

Chap 1 Aspects algoritmiques des systèmes répartis

- Caractéristiques d'un support distribué :
  - Absence de mémoire commune
  - Degré de fiabilité des voies de communication
  - Echanges de messages

1) Définitions et concepts [Raynal]

2) Illustrations et exemples [Verjus]

1. Définitions et concepts

- 1.1. Caractéristiques des systèmes répartis

1.1.1. Caractéristiques générales

- Absence d'un état global
- Pas de « temps absolu » :
  - Evenements perçus par un processus :
    - » Ceux dont il est le siège
    - » Ceux issus des relations de causalité entre l'émission et la réception d'un même message

  ⇒ Impossible d'ordonner deux événements quelconques

1.1. Caractéristiques des systèmes répartis

1.1.2. Distribution des données

- Duplication des données :
  - La donnée x est recopiée en n exemplaires x_1,…, x_n sur n sites
  - Assurer cohérence mutuelle des copies : x_1 =…= x_n = x
- Partitionnement des données :
  - Chaque partition se trouve sur un site donné

1.1.3. Distribution du contrôle

- Pas de relation hiérarchique entre les processus
- Privilège tournant : « maître » temporaire
- Meilleure résistance aux pannes

1.2. Eléments de l'algorithmique distribuée

1.2.1. Les processus

- Processus : programme qui s'exécute
- Doté d'une structure de contrôle non déterministe :
  - Attente de plusieurs événements possibles
  - Modélisation par automates à états finis : réseaux de Nutt

1.2.2. Les voies de communication

- Propriétés structurelles des liaisons
  - Anneau : structure en boucle
  - Étoile : structure centralisée
  - Arbre : structure hiérarchisée
  - Maillage complet : structure fortement connexe

=> Structures logicielles

1.2.2. Les voies de communication

- Propriétés comportementales des liaisons

Hypothèses sur le comportement des voies de communication :

- H1 : transmission sans duplication de messages
- H2 : transmission sans altération de messages
- H3 : pas de déséquencement des messages
- H4 : délai d'acheminement des messages fini : pas de perte
- H5 : délai d'acheminement des messages borné : on peut savoir si un message est perdu

=> Couches logicielles pour mettre en œuvre ces hypothèses

1.3. Qualités d'un algorithme distribué

1.3.1. Degré de répartition

- Lié à la symétrie des rôles joués par les processus :
  - Non symétrie : chaque processus exécute un texte différent
  - Symétrie de texte : les processus exécutent le même texte, qui fait référence au nom du processus qui l'exécute
    => comportement différent en fonction des noms
  - Symétrie forte : les processus exécutent le même texte, dans lequel leur nom n'apparaît pas
    => comportement différent en fonction des messages reçus

=> Ces degrés de symétrie permettent de définir le degré de répartition du contrôle

1.3. Qualités d'un algorithme distribué

1.3.2. Résistance aux pannes

- Ce critère est lié à la symétrie :
  - Un algorithme est d'autant plus résistant que le degré de répartition est élevé

1.3.3. Hypothèses sur le réseau

- L'algorithme est d'autant plus intéressant qu'il fait le moins d'hypothèses sur les voies de communication (H1-H5)

1.3.4. Trafic engendré

- La performance se mesure par :
  - le nombre de messages échangés
  - La charge induite sur les voies
  - Le temps d'attente dans les processus

1.4. Concepts et techniques de l'algorithmique distribuée

1.4.1. Calcul diffusant [Dijkstra et Scholten]

- Topologie de communication = arbre :
  - L'initiateur du calcul est la racine de l'arbre
  - Tout processus de l'arbre ne peut émettre des messages que s'il en a lui-même reçu
- Principe du calcul diffusant :

(i) La racine émet un message vers chacun de ses fils

(ii) Un processus nœud, re-expédie tout message reçu vers ses fils, et attend que chaque fils lui ait envoyé une réponse, après quoi il envoie sa réponse à son père dans l'arbre

(iii) Si le processus est une feuille de l'arbre, il envoie immédiatement sa réponse

(iv) Lorsque la racine a reçu toutes les réponses, le calcul diffusant est terminé

1.4. Concepts et techniques de l'algorithmique distribuée

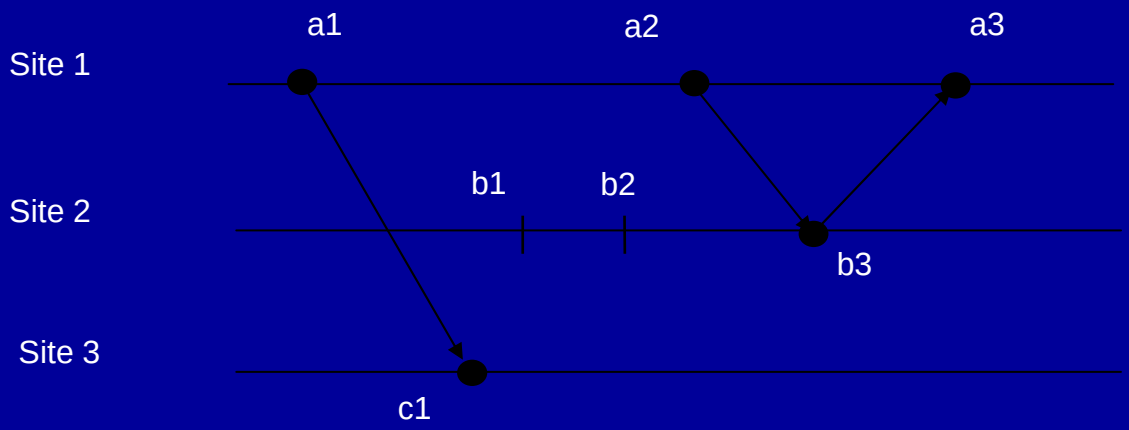
1.4.2. Jeton circulant

- Topologie de communication = anneau :
  - Circulation d'un privilège dans un ensemble de processus connectés en anneau
  - Simplicité conceptuelle : utilisé pour des algorithmes d'exclusion mutuelle et de terminaison

1.4.3. Estampillage ou horloges logiques

- Pas d'horloge globale
- Ordre relatif des événements :
  - Précédence entre deux événements au sein d'un même processus
  - Précédence de l'envoi d'un message par un processus sur la réception de ce message par un autre processus

1.4.3. Estampillage ou horloges logiques



- Dans cet exemple :
  - La séquence < a1 b1 a2 b2 b3 c1 a3 > est-elle compatible avec les relations de précédence ?
  - Ordre temporel « absolu » : < a1 c1 b1 b2 a2 b3 a3 >

- Principe d'estampillage [Lamport] utilisé dans algorithmes d'exclusion mutuelle ou de détection d'interblocage

=> Besoin de définir un ordre total entre tous les événements pour permettre un fonctionnement équitable

2. Illustrations et exemples

- 2.1. Synchronisation de processus et allocation de ressources

Exemple du parking :

- Processus = automobiles
- Entrent en compétition pour l'utilisation des N places d'un parking
- Chaque processus exécute la suite d'instructions :
  - e : entrer dans le parking
  - se garer
  - s : sortir du parking

Exemple du parking (synchronisation de processus) :

1. Analyse des séquences d'événements e et s observés par le contrôleur :

   1.1. Définir une séquence légale d'événements (en fonction de N)

   1.2. Donner pour N=3 des exemples de séquence légale et illégale

   1.3. Pour N=3, quelle est la condition pour que le contrôleur bloque l'entrée du parking ?

Exemple du parking (synchronisation de processus) :

2. Formalisation des contraintes de synchronisation :

On définit les compteurs suivants :

- E : compteur enregistrant le nombre d'entrées
- S : compteur enregistrant le nombre de sorties

2.1. Formaliser les contraintes de synchronisation à faire respecter par le contrôleur en fonction de E, S et N

2.2. Le contrôleur doit-il vérifier $E - S \geq 0$?

2.3. Soit Y le nombre de places libres : exprimer Y en fonction de E, S, N

2.4. Décrire l'algorithme du contrôleur dans le cas d'un système centralisé (un seul accès en E/S)

2.5. On considère un parking à deux accès, le premier en entrée, le deuxième en sortie, avec un contrôleur associé à chaque accès. Le contrôleur n°2 envoie la valeur de S au contrôleur n°1, qui a donc une valeu r retardée de S : S'

Montrer que cette valeur retardée ne viole pas les contraintes de synchronisation. Y a-t-il une perte d'efficacité ?

Exemple du parking (synchronisation de processus) :

3. Cas d'un parking à N accès : définition d'une boucle virtuelle

Principe :

- chaque accès n¶ est associé à un contrôleur qui reç oit des messages de son prédécesseur , et envoie des messages à son successeur
- Un agent contenant la valeur du compteur Y (nombre de places libres), circule sur l'anneau.

3.1. Définir les actions d'un contrôleur sur l'anneau, lorsqu'il reçoit l'agent circulant

3.2. Avec solution quel est le rythme des entrées dans le parking ? Vous semble-t-il plus efficace d'avoir un seul accès en entrée ?

3.3. Que se passe-t-il en cas de perte de l'agent circulant sur l'anneau ?

2. Illustrations et exemples

- 2.2. Terminaison

Problème : détecter la terminaison d'un ensemble de processus coopérant à une tâche

Exemple du concierge de nuit :

- des agents travaillent dans un immeuble, de nuit, à raison d'un agent au plus par bureau. Si un agent travaille, alors il occupe un bureau allumé. Il peut quitter un bureau soit pour rentrer chez lui, soit pour continuer son travail dans un autre bureau. Quand tous les agents sont partis, le travail est dit terminé.

Exemple du concierge de nuit (terminaison)

- Le concierge a pour tâche de fermer l'immeuble (et d'éteindre les lumières) lorsque tous les agents ont fini leur travail. Pour déterminer si le travail est ou non terminé, on numérote chaque bureau de 1 à N et le concierge se dote d'une casquette. Quand le concierge veut lancer la procédure de décision,il va vers le bureau n°1 avec sa casquette sur la tête.

- Les règles suivantes sont alors observées :

1. si le concierge visite un bureau occupé (et donc allumé), il y demeure jusqu'à ce que l'agent ait quitté le bureau. S'il quitte un bureau allumé, il l'éteint, se met (ou reste) tête-nue, et passe au bureau suivant (dans l'ordre 1, 2, … N). S'il quitte un bureau éteint, il passe au suivant.

2. si un agent quitte un bureau pour aller travailler dans un autre bureau, il le laisse allumé. Quand il rentre dans un bureau pour travailler, il l'allume si ce n'est déjà fait.

3. si un agent a fini son travail, il quitte son bureau et l'éteint.

Exemple du concierge de nuit (terminaison)

- Le concierge a pour tâche de fermer l'immeuble (et d'éteindre les lumières) lorsque tous les agents ont fini leur travail. Pour déterminer si le travail est ou non terminé, on numérote chaque bureau de 1 à N et le concierge se dote d'une casquette. Quand le concierge veut lancer la procédure de décision,il va vers le bureau n°1 avec sa casquette sur la tête.

- Les règles suivantes sont alors observées :

1. si le concierge visite un bureau occupé (et donc allumé), il y demeure jusqu'à ce que l'agent ait quitté le bureau. S'il quitte un bureau allumé, il l'éteint, se met (ou reste) tête-nue, et passe au bureau suivant (dans l'ordre 1, 2, … N). S'il quitte un bureau éteint, il passe au suivant.

2. si un agent quitte un bureau pour aller travailler dans un autre bureau, il le laisse allumé. Quand il rentre dans un bureau pour travailler, il l'allume si ce n'est déjà fait.

3. si un agent a fini son travail, il quitte son bureau et l'éteint.

Exemple du concierge de nuit (terminaison)

- En suivant ces règles,lorsque le concierge a visité tous les bureaux et qu'il porte sa casquette, il peut affirmer que le travail est terminé. Dans le cas contraire il remet sa casquette et recommence sa visite par le premier bureau.

- Exercice : Montrer que cette procédure se termine

  1) Montrer que lorsque tous les agents ont fini de travailler, le concierge détectera la terminaison

  2) Le concierge peut-il détecter à tort la terminaison ?

  3) Que se passe-t-il si un agent oublie d'éteindre la lumière en rentrant chez lui ? Cela affecte-t-il la détection de terminaison ?

  4) Que se passe-t-il si un agent éteint la lumière en allant travailler dans un autre bureau ? Cela affecte-t-il la détection de terminaison ?

2. Illustrations et exemples

- 2.3. Exclusion mutuelle

Problème : allocation de p ressources à q utilisateurs (processus)
lorsque p=1

Exemple du tunnel :

- Considérons un tunnel ferroviaire, avec une seule voie bidirectionnelle, qui doit être empruntée par un seul train à la fois
- En l'absence de feux de signalisation (ou en cas de panne), définir un protocole de signalisation « manuel » à réaliser par deux employés situés aux deux extrémités du tunnel

2. Illustrations et exemples

- 2.4. Interblocage

Problème : détecter un ensemble de processus interbloqués, i.e
dans l'attente de ressources qu'ils détiennent mutuellement

Exemple du carrefour :

- Considérons un carrefour croisé entre deux routes, sans feux (ou avec des feux en panne), ni panneaux de signalisation.
- Quatre véhicules arrivent en même temps en provenance des quatre voies. Pour éviter une collision elles doivent toutes s'arrêter, puis appliquer la règle de priorité à droite. Or chaque voiture en ayant une à sa droite, reste bloquée.
- Définir un protocole de signalisation « manuel » à réaliser par les conducteurs des automobiles. Combien de conducteurs doivent effectuer ce protocole ? Que se passe-t-il si tous les conducteurs l'exécutent en même temps ?

$\Rightarrow$ Que se passe-t-il dans le cas d'un rond-point ?

2. Illustrations et exemples

- 2.5. Raisonnement par induction

Problème : raisonnement et apprentissage sont des éléments importants de la « prise de décision » au sein d'un groupe de processus répartis

Exemple des époux infidèles :

- A Mamajorque, règne une reine qui dit toujours la vérité. Les femmes mariées de la ville sont, d'une part obéissantes à leur reine, et d'autre part ont un raisonnement logique sans défaut. En sus, chaque femme (chaque processus) est capable d'entendre chaque coup de fusil (chaque message) tiré dans Mamajorque. Un jour la reine réunit toutes les femmes mariées de sa cité et leur déclare :

- « Il y a, à Mamajorque, un ou plusieurs époux infidèles. Bien qu'aucune de vous ne sait si son mari est ou non fidèle, chacune de vous sait bien sûr qui sont les autres maris infidèles (puisque vous êtes soit impliquées, soit dans le secret). Je vous interdis de discuter de la fidélité de votre propre époux avec les autres femmes. Cependant si vous découvriez (par raisonnement) que votre mari est infidèle, alors vous devez le tuer d'un coup de fusil à minuit du jour où vous aurez fait cette découverte. »

Exemple des époux infidèles (raisonnement par induction)

- 39 nuits silencieuses se passent.
- Durant la 40^{e}, on entend à minuit de très nombreux coups de feu.
- Que peut-on en conclure ? Raisonner par induction
- Généraliser le problème en fonction du N^{e} jour de la « mise à feu »