

Programmation répartie

Principes et algorithmes

Introduction

Hinde Bouziane

bouziane@lirmm.fr

Principale question (1/2)

Comment réaliser une **tâche commune** dans un **système réparti** ?

Système réparti ?

- Un système (informatique) dans lequel les ressources (calcul, stockage, utilisateurs) et le contrôle ne sont pas centralisés.
- Ensemble d'entités sans mémoire commune coopérant via un système de communication (asynchrone).

⇒ A quoi ces définitions vous font-elles penser ?

Principale question (2/2)

Comment réaliser une **tâche commune** dans un **système réparti** ?

Tâche commune : exemples

- coloration d'un graphe
 - diffusion dans un réseau
 - exclusion mutuelle / partage d'une ressource
 - ...
- ⇒ Etude d'un exemple au tableau.

Description / conception

En donnant un algorithme dit réparti / distribué.

Algorithme réparti / distribué (1/2)

Définition

Description d'un calcul à effectuer par des entités autonomes ayant pour but de solutionner un même problème. Chaque entité possède une partie de l'information globale du système réparti et peut communiquer (uniquement) avec ses voisins.

Algorithme réparti / distribué (2/2)

Il n'y a pas de contrôle centralisé dans un algorithme distribué

- Contrôle centralisé :
 - un site particulier joue le rôle central d'arbitre : il prend toutes les décisions et a généralement besoin de connaissances dispersées dans le système réparti.
 - à ne pas faire pour définir un algorithme distribué.
- Contrôle réparti / distribué :
 - aucune entité arbitre : tous les sites sont égaux, en droit et en devoir.
 - le contrôle est donc réparti sur les sites et chaque site a une connaissance locale / partielle du système global (limitée à son voisinage)

Modélisation d'un système réparti

Modélisation par un graphe

- Les noeuds du graphe représentent les entités de calcul
 - nommées : noeuds, sites, processus, processeurs, machines, etc.
 - les noeuds peuvent avoir des identifiants : P_1 , P_2 , ... A , B , ...
 - se rappeler que ce sont des processus qui calculent et échangent des données (envoi/réception de messages) et non du matériel.
- Les arêtes du graphe représentent les liens de communication entre les entités de calcul
 - nommés : canaux de communications, liens, arêtes, etc.

Remarques

- Le comportement d'un algorithme réparti dépend du graphe modélisant le système et d'hypothèses sur les communications (liens FIFO, etc.).
- Faire la différence entre les liens physiques reliant des processeurs et les liens logiciels.
- Dans cette UE (TP) : ce graphe sera à construire à l'aide de connexions de sockets utilisant TCP.
- Il faut comprendre qu'avant "d'exécuter" un algorithme, il sera nécessaire de mettre en place les processus et les canaux de communications.

En pratique dans cette UE

- Le lancement des sites se fait au préalable : mettre en relation chaque site avec son voisinage avant qu'il ne commence son calcul.
- La mise en place d'un graphe (avec éventuelles initialisations des sites) sera centralisée, i.e. contrôlée par un processus supplémentaire. Ce processus se termine une fois le graphe d'interconnexions construit. C'est l'unique étape qui se fait de manière centralisée.
- Mise en oeuvre lors des premiers TP (avec hypothèse : un graphe peut être quelconque et peut être de grande taille).

Conception d'un algorithme réparti

Conventions de base

- Un algorithme décrit la tâche d'un site. Nous présenterons l'algorithme d'un site $P_i, 1 \leq i \leq N$
- Tous les sites ont le même algorithme. Conséquence : ces sites ont des variables locales portant le même nom.
- Un algorithme réagit aux réceptions de messages (orienté évènement)
- Un algorithme envoie des messages

Syntaxe et structure

- Pseudo code classique avec des structures de contrôle (tests, boucles, etc.)
- Structure par blocs :
 - un bloc pour l'initialisation des variables locales
 - un/des blocs pour l'envoi de message(s) (selon contexte)
 - un bloc pour la réception d'un type de messages (nb blocs = nb types)

Exemple : un algorithme de diffusion d'informations (1/3)

Contexte et suppositions

- Illustration d'une description d'algorithme (notations, etc) sans analyse.
- Diffusion dans un réseau non structuré (graphe quelconque) de N sites.
- Chaque site S_i , $1 \leq i \leq N$ connaît ses voisins et sait les distinguer. Soit V l'ensemble de ces voisins.
- Les échanges de messages sont en mode point à point et sont asynchrones.
- Initialement, un sous ensemble de sites dispose d'une information à diffuser.

Algorithme

Un site disposant d'une information initiale à diffuser envoie cette dernière à tous ses voisins. Un site qui reçoit un message pour la première fois l'envoie à son tour à tous ses voisins différents de celui qui vient de le lui envoyer. Un site qui reçoit un message déjà reçu précédemment l'ignore.

Exemple : un algorithme de diffusion d'informations (2/3)

Algorithme 1 : Algorithme de diffusion pour un site S_i

Initialisation :

source : initialisée à vrai si S_i est source d'une information d_i à diffuser;
setMsg : ensemble des messages reçus, initialisé à ϕ ;

Procédure diffuser_donnée(d) :

pour $S_k \in V$ **faire**

└ envoyer un message $\langle d, i \rangle$ à S_k ;

Lors de la réception d'un message $\langle M, \text{emetteur} \rangle$ **depuis** S_j :

si $\langle M, \text{emetteur} \rangle \notin \text{setMsg}$ **alors**

└ ajouter $\langle M, \text{emetteur} \rangle$ à *setMsg*;

└ **pour** $S_k \in V - \{S_j\}$ **faire**

└└ envoyer $\langle M, \text{emetteur} \rangle$ à S_k ;

Exemple : un algorithme de diffusion d'informations (3/3)

Utilisation

L'algorithme précédent répond au problème de la diffusion. Il est complété (sans en faire partie) par un ou des exemples d'utilisation par les processus P_i . Il s'agit de définir un algorithme que nous appellerons *bloc d'utilisation*.

Algorithme 2 : Bloc d'utilisation par S_i

Variable : d_i donnée à diffuser.

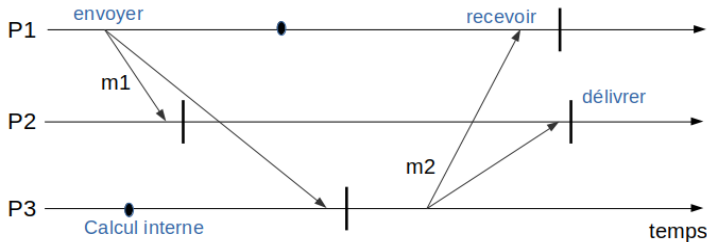
si source alors

```
    saisir une donnée  $d_i$ ;  
    diffuser_donnée(<  $d_i$ ,  $i$  >);
```

Comportement d'un algorithme réparti

Graphe de précedence immédiate

- Capture d'un comportement global du système
 - ordre possible d'évènements (locaux) dans le temps ;
 - les messages en transit entre sites voisins ;
 - etc.
- Évènement local : envoi / réception d'un message, délivrance d'un message ou calcul interne.



Étude d'un algorithme réparti : ce qui nous intéresse

Théorie

- Son principe.
- La correction de son comportement global.
- Sa complexité : en nombre de messages et/ou taille des messages et/ou temps des échanges de messages, etc.
- Sa tolérance aux pannes.
- Ses avantages et ses inconvénients.

Pratique

Eventuelle implémentation en TP.

Exercice : reprendre l'exemple de la diffusion et énumérer les notions nécessaires pour son implémentation.

Dans la suite de cette UE

- Programmation réseau avec les sockets (révision + multiplexage des E/S)
- Programmation multi-thread (révision)
- Notion d'état global d'un système réparti (horloges)
- Etude de quelques algorithmes répartis