Programmation parallèle et distribuée

Arnaud Labourel
Courriel : arnaud.labourel@lif.univ-mrs.fr

Université de Provence

26 janvier 2012

But du cours

- Introduction aux systèmes
  - parallèles,
  - concurrents,
  - répartis,
  - distribués.
- Théorie et Applications.
- Réalisation d'une application en Java.

Introduction

Ce cours s'appuie sur :

- cours Système
- cours Réseaux
- Java

Sources :

- Cours de E. Goubault
- Y. Robert et A. Legrand *Algorithmique Parallèle*.
- A. Tanenbaum : *Distributed Systems*.
- Brian Goetz et al : *Java Concurrency in Practice*.

Horaires

- **Cours :** jeudi de 8h30 à 10h30
  (10 séances, débute le 26 janvier)
- **TD :** jeudi de 10h30 à 12h30
  (10 séances, débute le 2 février)
- **TP :**
  - groupe 1 : mardi de 14h à 16h
    (10 séances, débute le 31 janvier)
  - groupe 2 : mardi de 16h à 18h
    (10 séances, débute le 31 janvier)

Fonctionnement

Page web :
`http://pageperso.lif.univ-mrs.fr/~arnaud.labourel/PPD/index.html`

- Partiel mi-mars
- Projet en deuxième moitié du semestre

Evaluation : $\frac{3\max(Ex, \frac{2Ex+Pa}{3})+TP}{4}$

Plan du cours

Parallélisme

Définition (Petit Robert)

Inform. Technique d'accroissement des performances des ordinateurs utilisant plusieurs processeurs fonctionnant simultanément.

Systèmes Parallèles

Découper un gros problème en de nombreux petits problèmes :

- Super calculateurs pour le calcul scientifique
- Stations multiprocesseurs
- Grilles
  - SETI@Home
  - folding@home
  - défi cryptographiques (`distributed.net`)
  - ....

Parallélisme à différents niveaux

- A l'intérieur du processeur
  - instructions en parallèle (pipeline)
  - multi-cœurs
- Multi-processeurs

Le parallélisme au sein du processeur

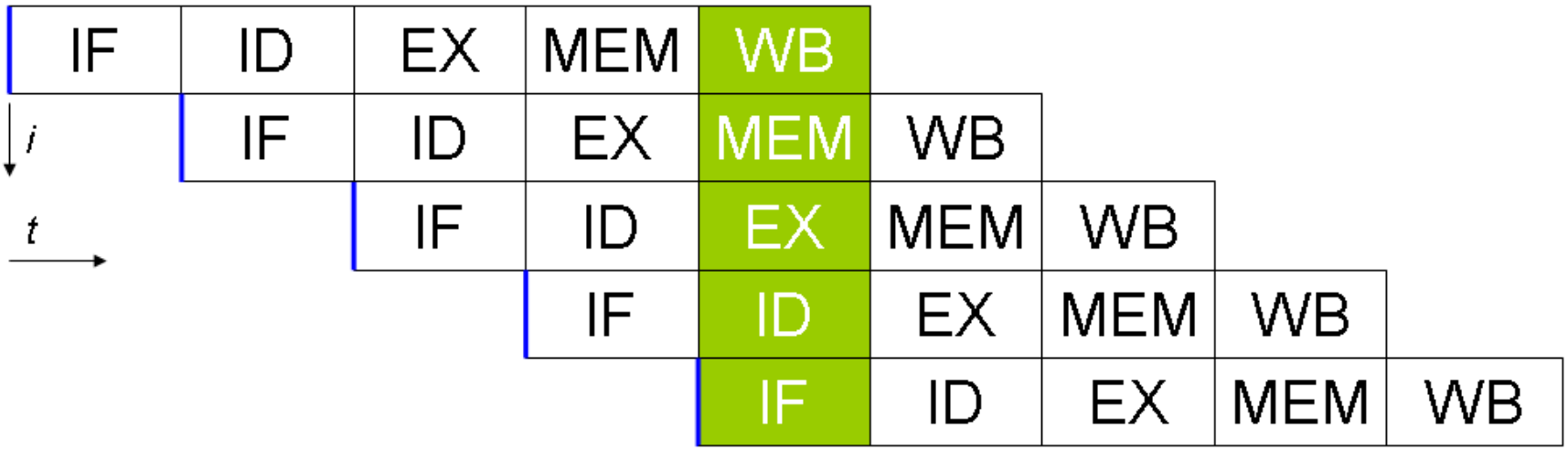
L'exécution d'une instruction (multiplication par ex.) nécessite plusieurs cycles d'horloge.

$\Rightarrow$ On exécute en parallèle les instructions indépendantes pour gagner du temps.

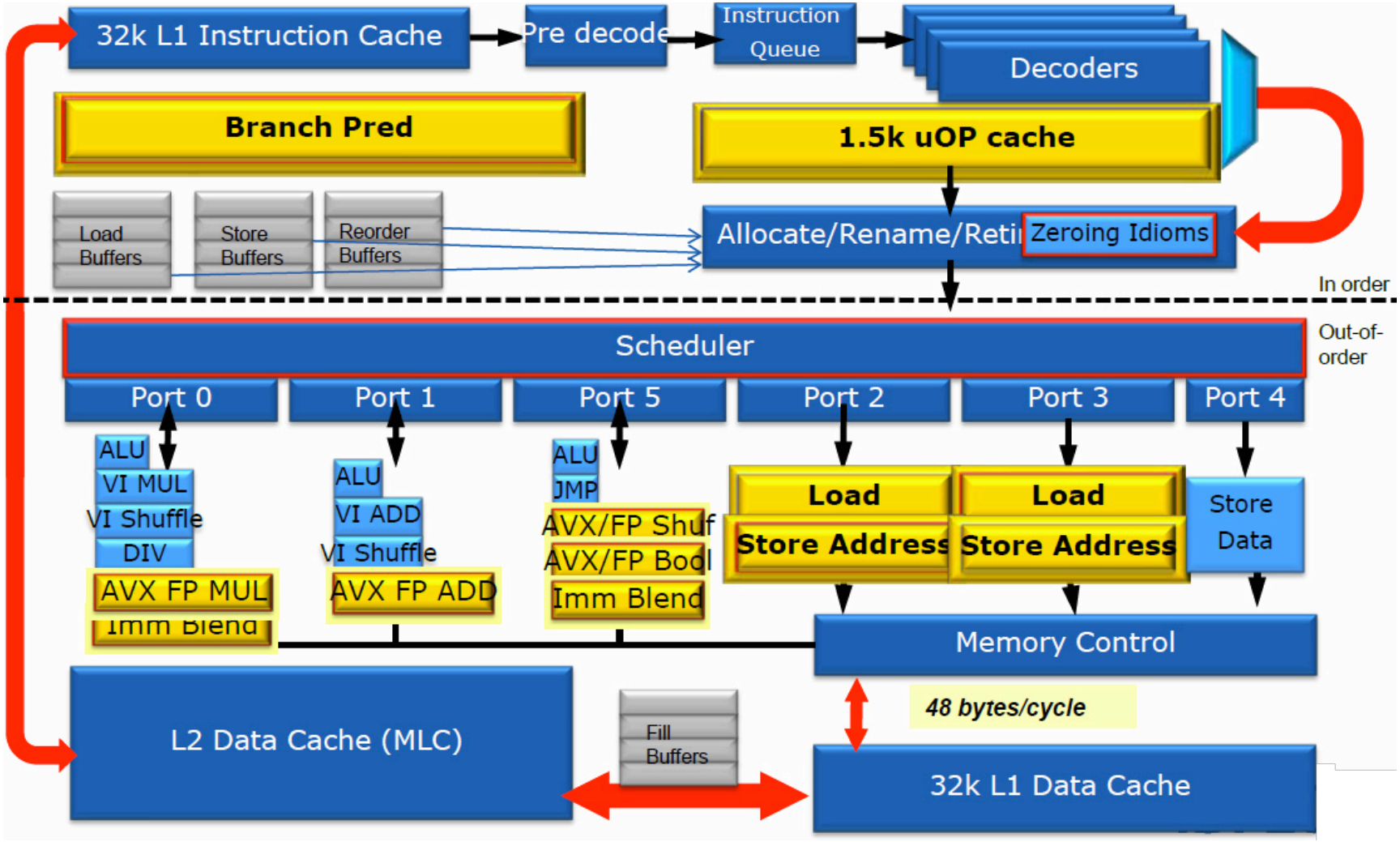
Un processeur possède plusieurs Unités Arithmétique et Logiques

Les instructions peuvent être réarrangées afin d'améliorer le rendement (out of order execution).

Exemple de séquençage des instructions



Exemple d'architecture avec pipeline



Multi-cœur

Cœur

Unité exécutant les instructions dans un processeur

Multi-cœur : plusieurs unités exécutant des instructions en parallèle.

Multi-cœur (tous sur un même circuit) $\neq$ multi-processeur (circuits différents)

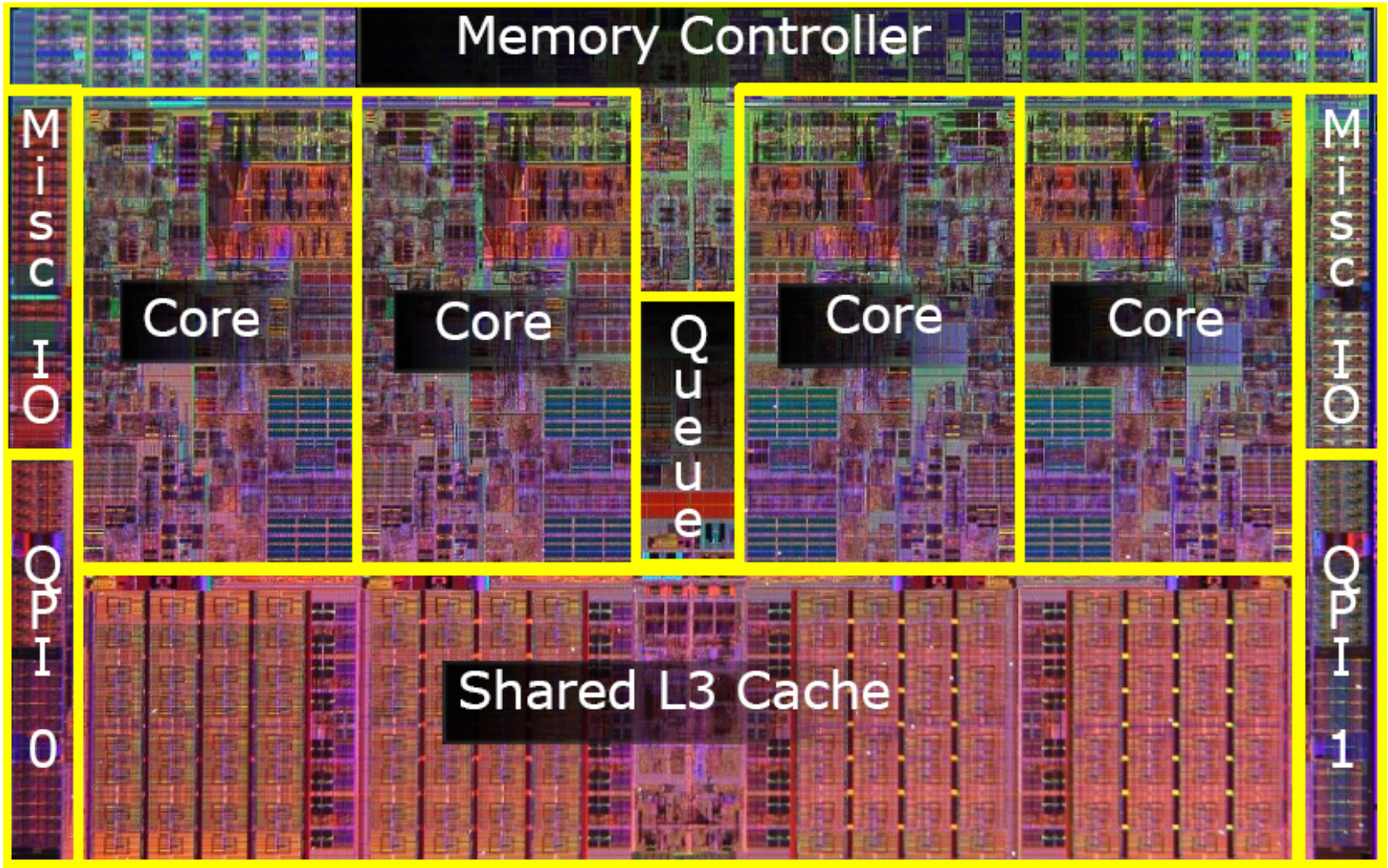
Processeur multi-cœur

Un très grande partie des processeurs actuels sont multi-cœurs (généralement de 2 à 8 cœurs).

Encore plus de cœurs dans l'avenir

- prototype tera-scale (80 cœurs) d'Intel
- processeurs multi-cœurs pour les appareils mobiles (smartphones, tablettes)

Exemple d'architecture multi-cœur



Le parallélisme multiprocesseurs

Un concept ancien : premier ordinateur multi-processeur en 1962, premier ordinateur multi-processeur parallèle en 1969.

Similaire au multi-cœur à l'exception de la mémoire cache qui n'est pas partagée

Quelques exemples de multiprocesseurs

- **Serveurs :**
  plusieurs sockets pour accueillir des processeurs.
- **Supercalculateurs :**
  - Le plus puissant : Computer K
    - 88,128 processeurs 8-cœurs (2GHz)
  - Le deuxième : Tianhe-1A
    - 14 336 processeurs
    - 7 168 processeurs graphiques

Les problèmes posés par la multiplication des processeurs

Problème d'optimisation

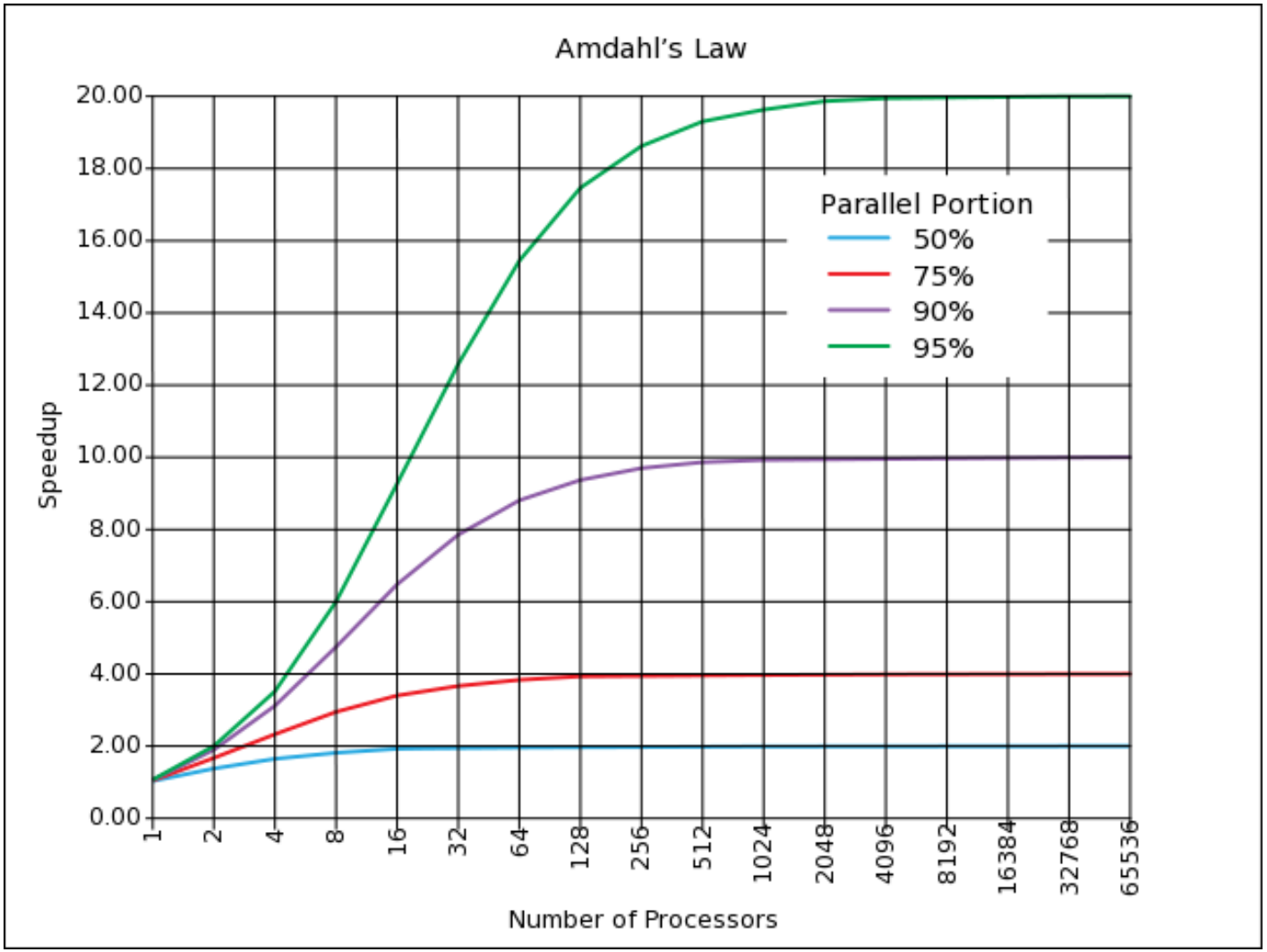
Une très grande partie des programmes actuels n'utilise pas la puissance des multi-cœurs.

Limite théorique : loi d'Amdahl

$$Rendement = \frac{1}{(1-s) + \frac{s}{N}}$$

s : proportion d'activité parallélisable
N : nombre de processeurs

Loi d'Amdahl


Amdahl's Law
Speedup
Number of Processors
Parallel Portion
50%
75%
90%
95%
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
20.00
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536

Systèmes Concurrents

Définition

Eléments situés dans un même espace de ressources.

- Système multitâche
  - tous les systèmes actuels
  - sauf si les ressources sont limitées
- Problème de *partage des ressources*
  - mémoire partagée
  - exclusion mutuelle
  - processus léger (*thread*)

Mémoire partagée

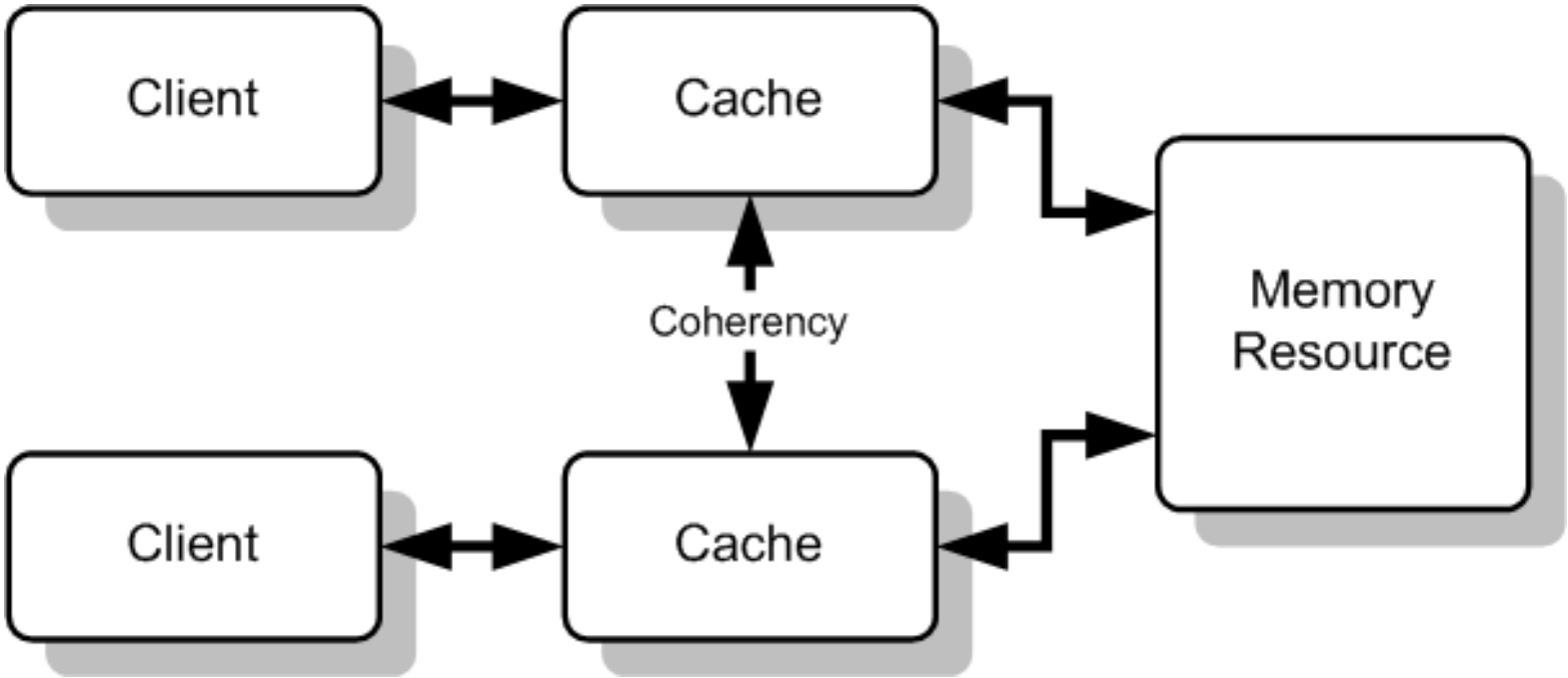
Définition

Plusieurs processus partagent un espace mémoire.

Problèmes :

- cohérence de cache
- accès concurrents

Cohérence du cache



Les caches des différents processeurs peuvent contenir les mêmes données qui seraient modifiées de manière indépendante.

Exclusion mutuelle

Définition

Primitive de synchronisation utilisée en programmation informatique pour éviter que des ressources partagées d'un système ne soient utilisées en même temps.

Méthodes d'exclusion mutuelle vues au chapitre 4 du cours.

Pourquoi utiliser l'exclusion mutuelle ?

Exemple de programmes exécutés en parallèle
global = 0 ;

Programme 1 :
global = global + 1 ;

Programme 2 :
global = global + 2 ;

Pourquoi utiliser l'exclusion mutuelle ?

Exemple d'exécution :

Programme 1

$R1 \leftarrow 0$

$R1 \leftarrow R1 + 1$ (1)

$global \leftarrow R1$ (1)

Programme 2

$R2 \leftarrow 0$

$R2 \leftarrow R2 + 2$ (2)

$global \leftarrow R2$ (2)

Au final, *global* est égal à 2 au lieu de 3 qui est le résultat attendu.

Processus léger

Définition

Ensemble d'instructions à exécuter se partageant le même espoir mémoire, chacun ayant une pile d'appel distincte.

Utilité :

- Séparer les différentes fonctionnalités d'un processus : gestions des périphériques, calculs lourds.
- Gains par rapport à l'utilisation de processus distincts : commutation de contexte moins couteux.

Systèmes distribués : terme générique

Définition (Tanenbaum)

Un *système distribué* est une collection d'ordinateurs indépendants qui apparait comme *un seul système* pour ses utilisateurs.

Exemple de systèmes distribués

- **Grille informatique :** infrastructure virtuelle constituée d'un ensemble de ressources informatiques :
  - hétérogènes
  - délocalisées
- **Grappe de serveurs :** plusieurs ordinateurs en réseau qui vont apparaître comme un seul ordinateur ayant plus de capacité :
  - homogènes
  - localisés

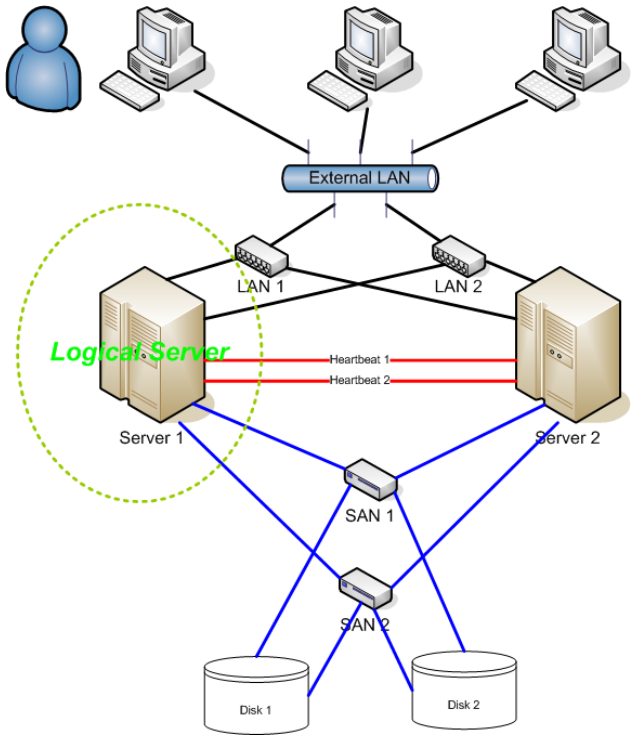
Exemple de grilles informatiques

Projets de calcul distribué :

- **SETI@home :** recherche de signe de vie extra-terrestre
  - 5,2 millions de participants
  - puissance comparable au meilleur super-calculateur actuel
- **Folding@home :** étude du repliement des protéines
  - 4.51 millions de participants
  - 400 000 maximum en même temps
  - plus puissant que le meilleur super-calculateur actuel
- **Distributed.net :** challenge de déchiffrage de code

Exemple de grappe de serveurs

- **Super-calculateurs virtuels :**
  System X (grappe de 1100 serveurs)
- **Duplication de serveurs :**



Systèmes distribués

Terme générique mais aussi :

- Réseau
  - paradigme client/serveur
  - systèmes *n*-tiers
- (auto)Gestion de réseaux
  - routage
  - arbre couvrant
  - élection
  - gestion des défaillances
- Systèmes concurrents
  - principe du découpage d'un problème en nombreuses tâches spécialisées

Architecture client/serveur

Définition

Mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs clients du serveur. Chaque logiciel client peut envoyer des requêtes à un serveur.

Architecture client/serveur

Caractéristiques d'un serveur :

- initialement passif (en attente d'une requête) ;
- à l'écoute, prêt à répondre aux requêtes envoyées par des clients
- dès qu'une requête lui parvient, il la traite et envoie une réponse

Caractéristiques d'un client :

- initialement actif
- envoie des requêtes au serveur ;
- il attend et reçoit les réponses du serveur.

Architecture 3-tiers

Définition

Modèle d'architecture séparant en trois couches :

- présentation des données (affichage)
- traitement des données
  (partie fonctionnelle, logique)
- accès aux données persistantes
  (serveurs base de données)

Architecture 3-tiers

Presentation tier

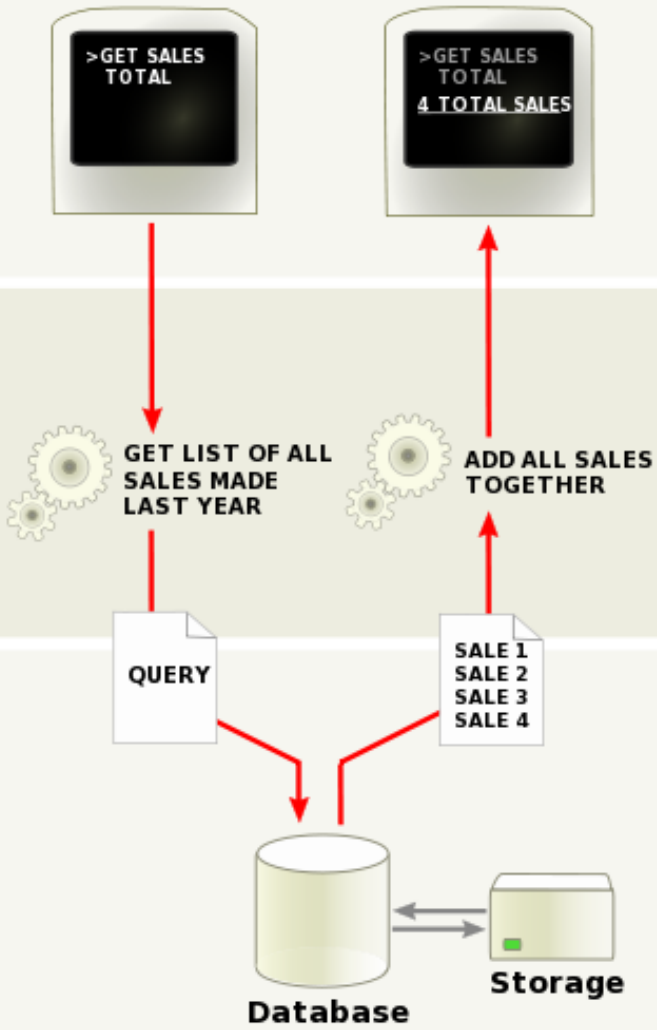
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



Routage

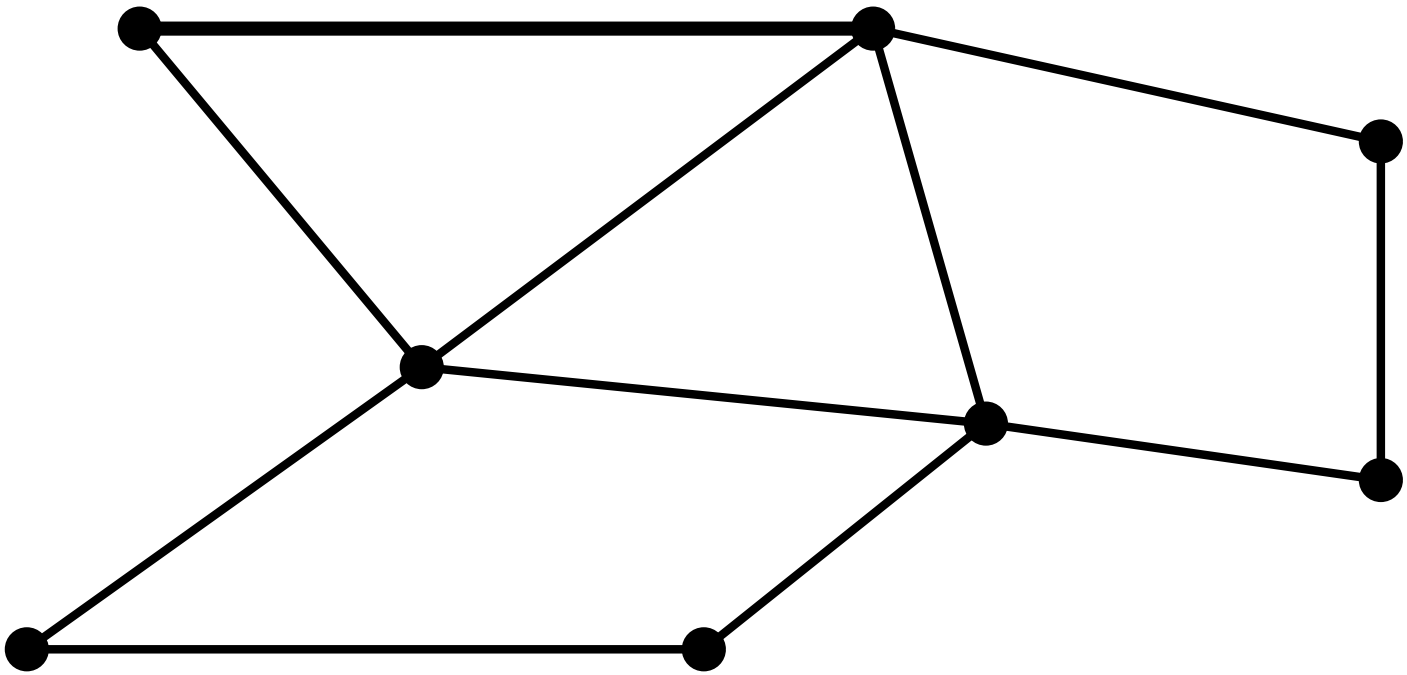
Définition

Mécanisme de choix des chemins d'acheminements des données d'un expéditeur jusqu'à un ou plusieurs destinataires dans un réseau.

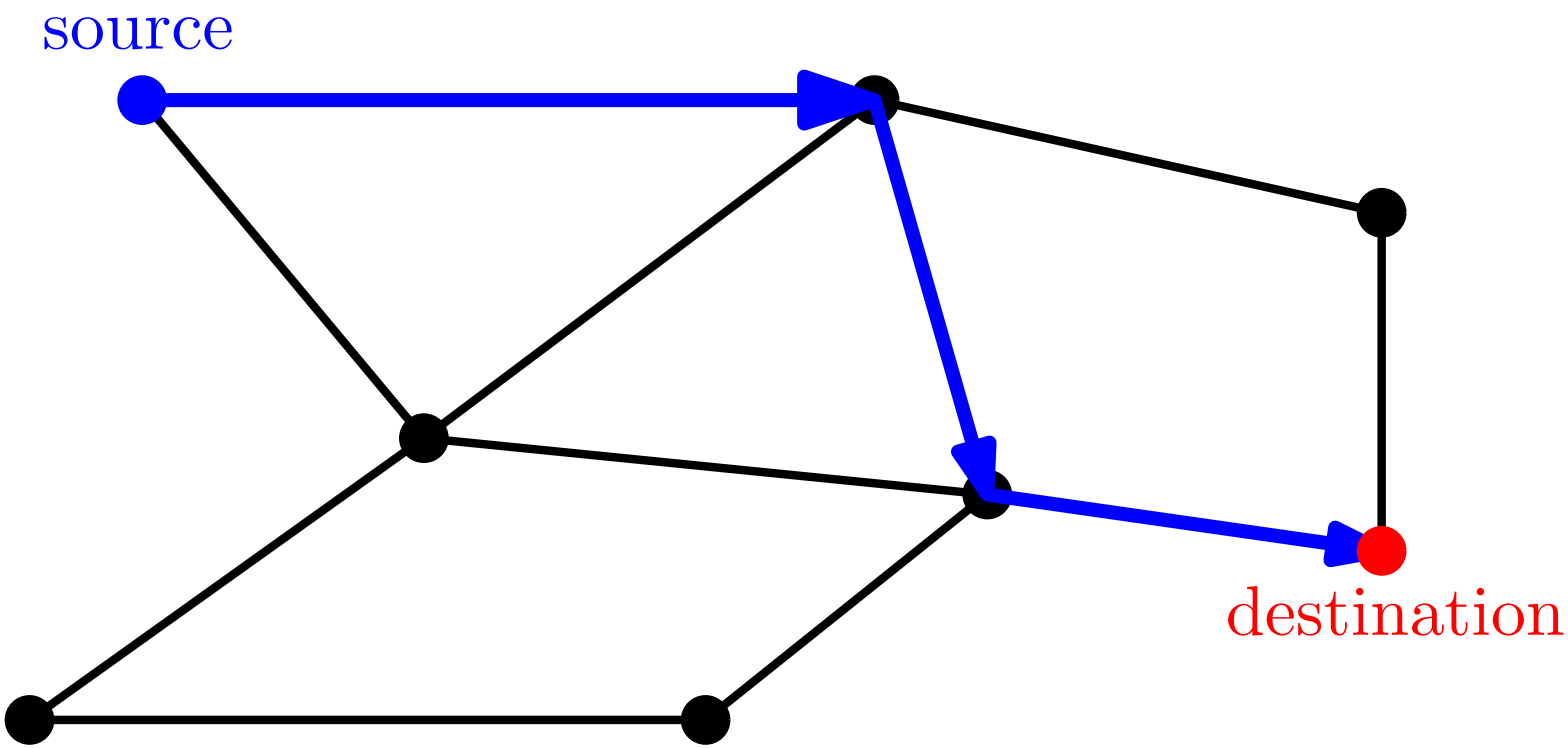
Différent type de routage :

- unicast : une seule destination
- broadcast : toutes les machines,
- multicast : un ensemble de machines
- anycast : n'importe quelle machine

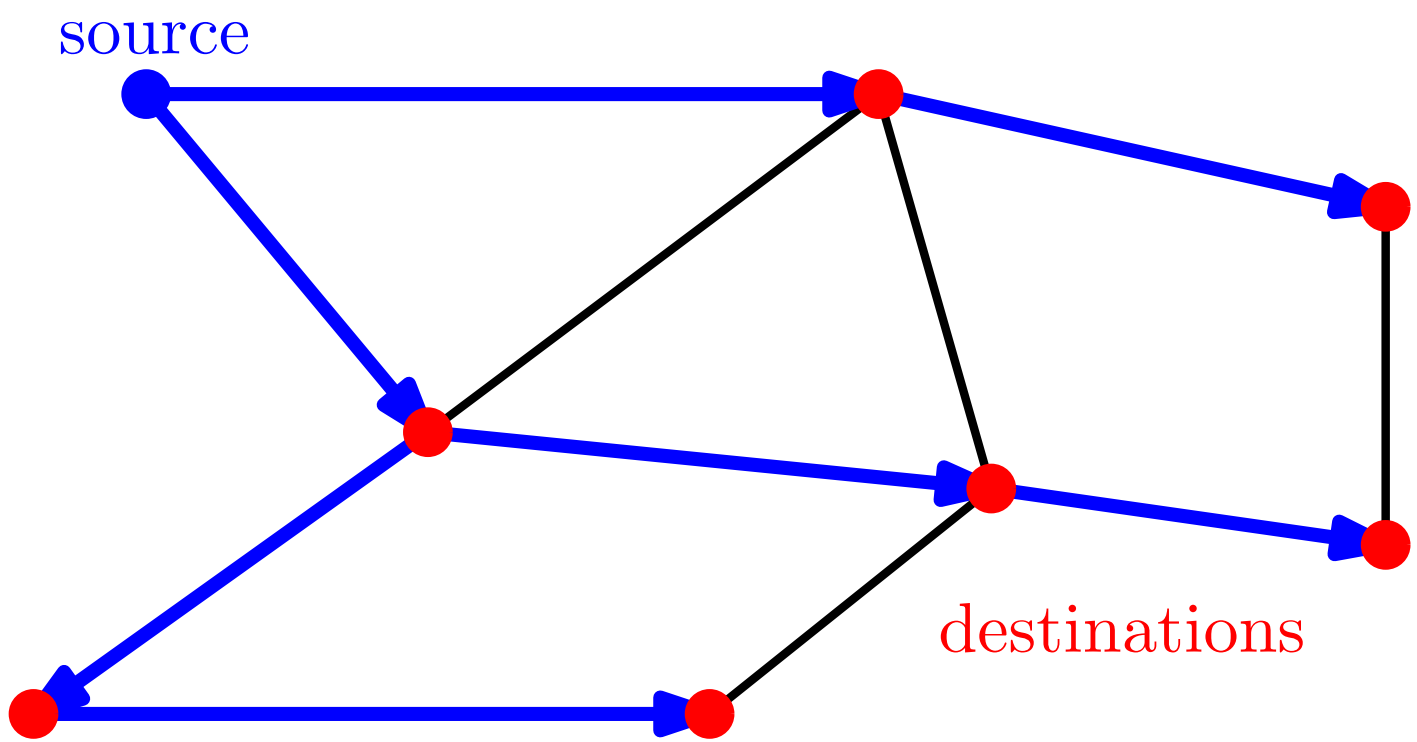
Exemple de réseau



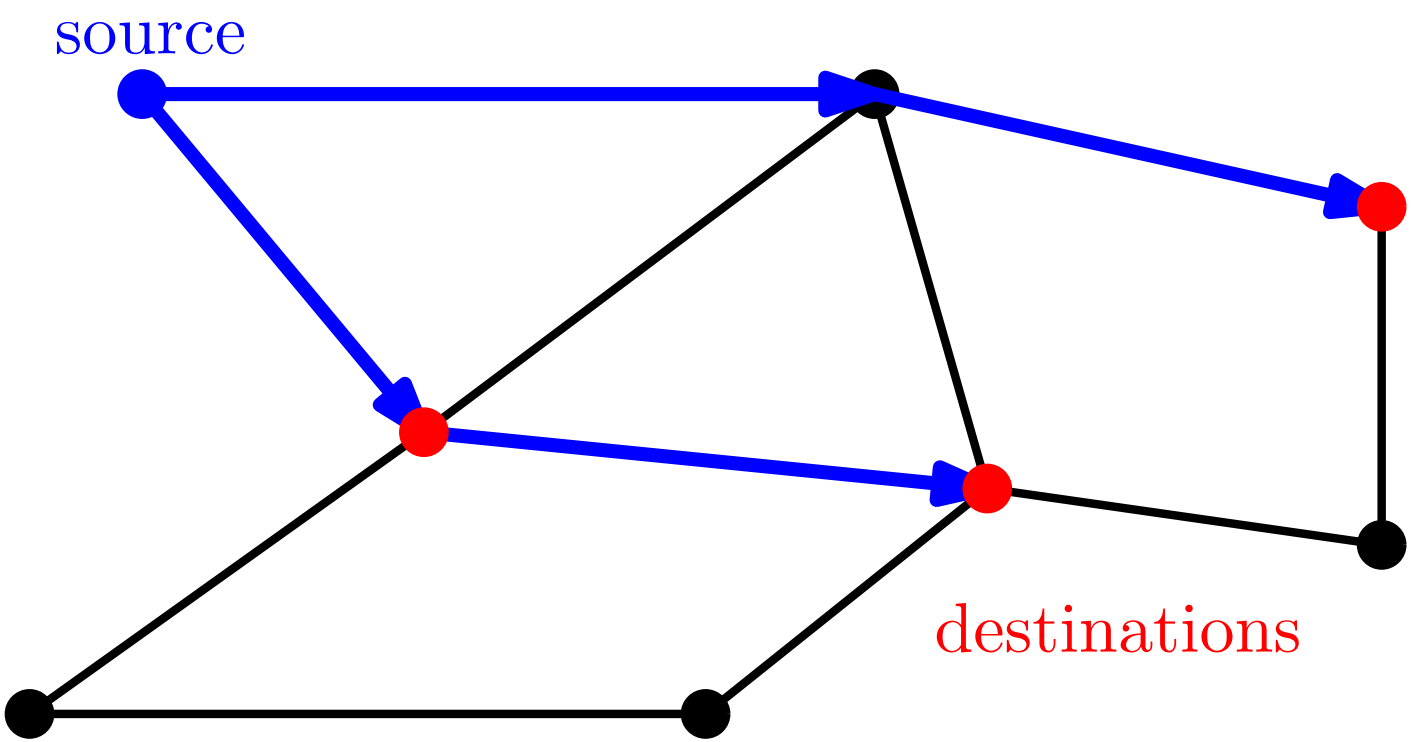
Exemple de routage unicast



Exemple de routage broadcast



Exemple de routage multicast



Élection

Définition

Procédé (algorithme) permettant d'élire un unique processus comme chef de file (leader) d'une tache.

But : choisir un chef qui prendra toutes les décisions.

Utile dans de nombreux algorithmes comme celui de l'arbre couvrant.

Arbre couvrant

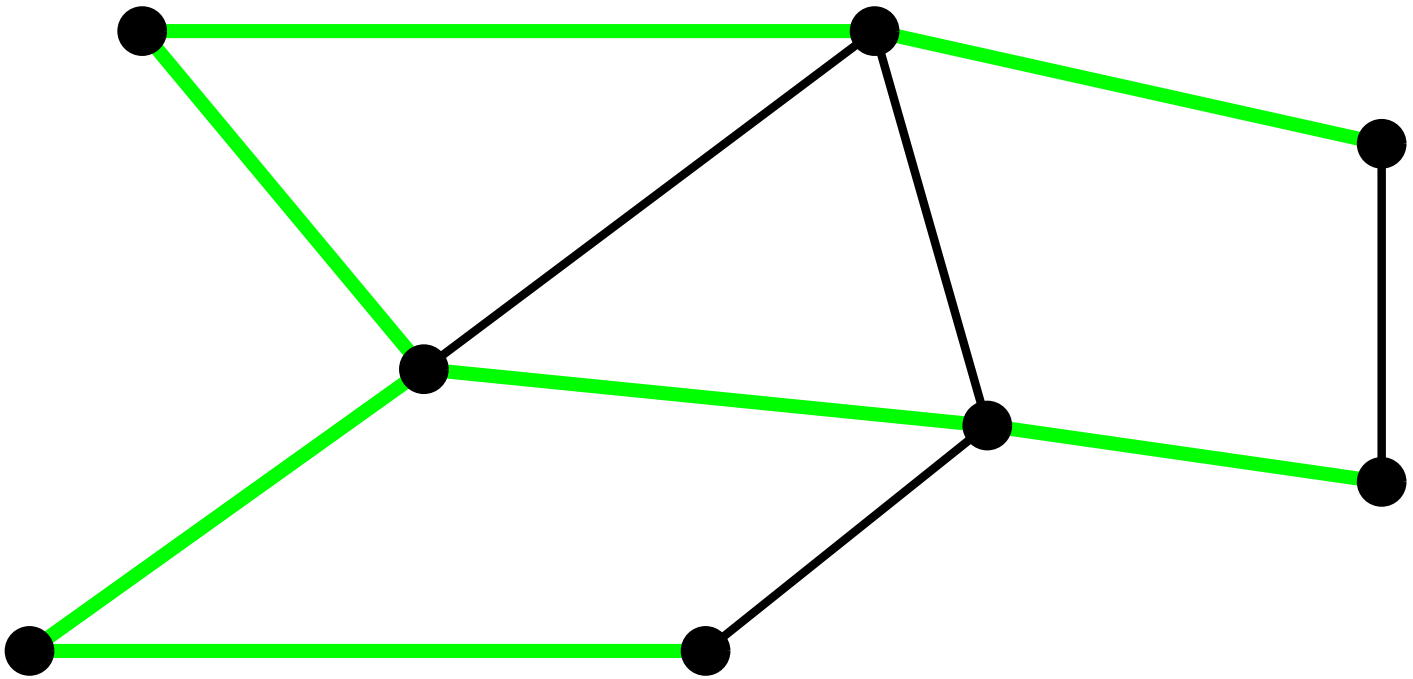
Définition (Théorie des graphes)

Sous-graphe connexe et acyclique contenant tous les sommets d'un graphe.

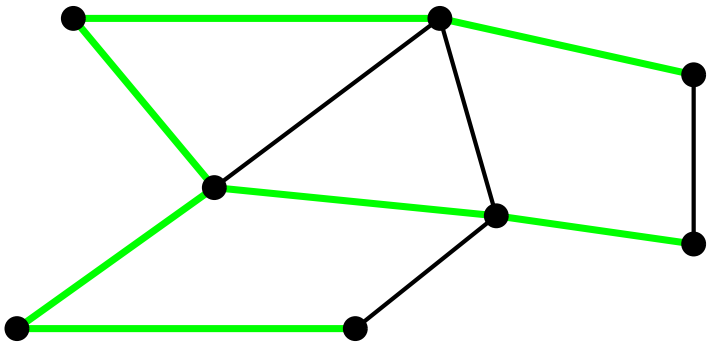
Utilité :

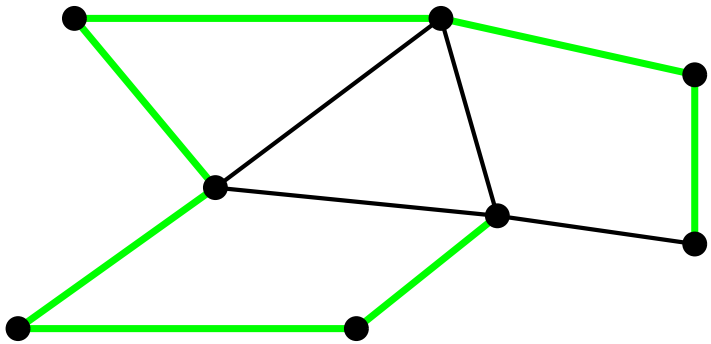
- Éviter les cycles dans un réseau commuté ethernet (Spanning Tree Protocol)
- Plus généralement, faciliter le routage en le limitant à l'arbre couvrant.

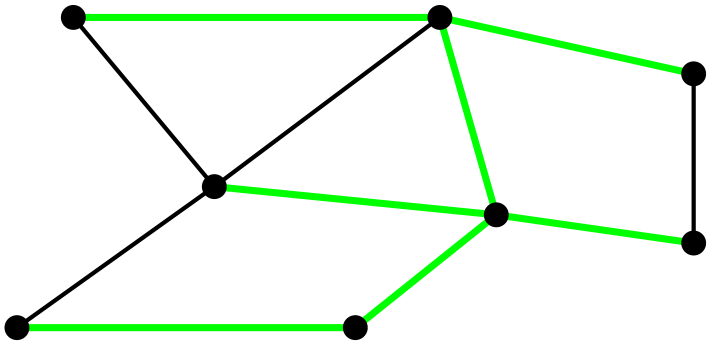
Exemple d'arbres couvrant

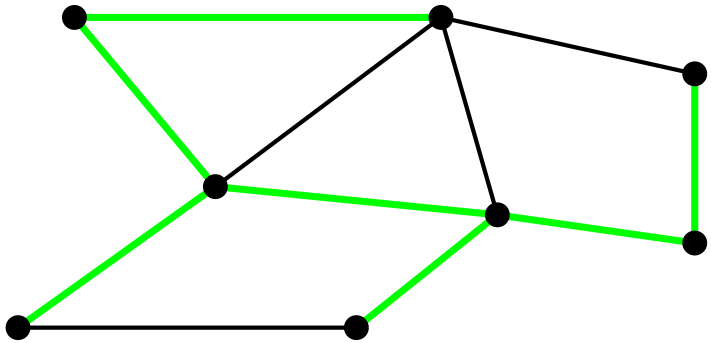


Plusieurs arbres couvrants possibles









Gestion des pannes

Le problème

Les systèmes informatiques fiables doivent être capables de gérer des composants défectueux qui donnent des informations conflictuelles aux différentes parties du système.

Les pannes peuvent survenir pour de nombreuses raisons :

- défaillance matérielle
- défaillance logiciel (bugs)
- attaque malicieuse (DoS)

Une analogie

Problème des généraux byzantins

Des généraux doivent mener une attaque malgré la présence de traitres essayant de semer la confusion parmi les généraux loyaux.
But : les généraux loyaux doivent se mettre d'accord sur un plan de bataille.

- Toujours réalisable si les traitres ne peuvent pas se faire passer pour un autre général
- Réalisable si et seulement si il y a moins d'un tiers de traitres dans les autres cas

Remarques sur la calculabilité et la complexité

- Calculabilité
  - défaillances
  - information dynamique incomplète
- Complexité
  - Coûts différents :
    - calcul interne
    - communication
  - Nombres de messages
  - Nombres de “tours” (round)

Systèmes répartis

- Système assez hétérogène
- Réseau + technologie objet
- Intergiciel (middleware)
  - DCOM
  - Corba
  - Java RMI

Interlogiciel

Définition

Logiciel tiers qui créant un réseau d'échange d'informations entre différentes applications informatiques.

Exemple :
Java RMI : interface de programmation (API) pour le langage Java qui permet d'appeler des méthodes distantes sur un serveur.

A Propos de ces définitions

- pas de véritable consensus
- beaucoup de points communs
  - simultanéité
  - problématique du partage des ressources (au sens large)
  - coordination
- parfois très entremêlée

  programmation concurrente sur une station multi-processeurs

Pour récapituler (Tentative...)

	Système Homogène	Système Hétérogène
conception descendante (top-down)	Parallèle	Concurrent
conception ascendante (bottom-up)	Distribué	Réparti