

## Modèles et Approches Formels pour les Systèmes Distribués

**M. Mosbah**  
**mosbah@labri.fr**  
**www.labri.fr/visidia/**

### Objectifs du cours

- Connaître les caractéristiques d'un système distribué (SD)
- Comprendre les concepts et les paradigmes fondamentaux d'un SD (les dessous)
- Au delà des technologies, maîtriser « le distribué » dans le cadre d'un modèle formel qui offre une abstraction et une simplicité
- Bien comprendre l'algorithmique distribuée( élection, arbre recouvrant, exclusion mutuelle, etc)
- Pouvoir raisonner dans un environnement distribué

- Prouver la correction et la validité d'un algorithme distribué
- Concevoir des algorithmes distribués
- Manipuler et implémenter ces algorithmes.
- et surtout avoir beaucoup de recul par rapport aux technologies.

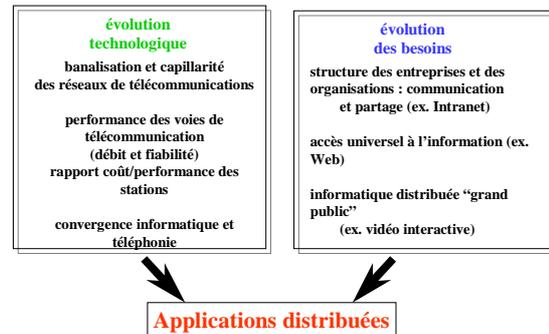
### I/ Introduction aux systèmes distribués

- Les réseaux d'ordinateurs sont partout !!!
  - Internet, réseaux de téléphones mobiles, réseaux locaux, etc.
- Les progrès technologiques des ordinateurs et le haut débit permettent aux différentes applications sur des ordinateurs distincts (et distants) de coopérer pour effectuer des tâches coordonnées.

## Les progrès technologiques

- Avant les années 80, les ordinateurs étaient encombrants et chers (les systèmes centralisés)
- A partir de la mi-80, deux nouveautés:
  - Microprocesseurs (moins chers et très rapide)
  - LANs and WANs
- Les ordinateurs en réseaux non seulement faisables, mais simples.

## Technologie + besoins



## II/ Définitions d'un SD

**Définition [Tanenbaum]:** *Un ensemble d'ordinateurs indépendants qui apparaît à un utilisateur comme un système unique et cohérent*

- Les machines sont autonomes
- Les utilisateurs ont l'impression d'utiliser un seul système.

- Définition [Lamport]
  - *A distributed system is one on which I can't do my work some computer has failed that I never heard of.*

Un système réparti est un système qui vous empêche de travailler quand une machine dont vous n'avez jamais entendu parler tombe en panne.

## Définition (pour ce cours)

- Un système distribué est un ensemble d'entités autonomes de calcul (ordinateurs, PDA, processeurs, processus, processus léger etc.) interconnectées et qui peuvent communiquer.
- Exemples:
  - réseau physique de machines
  - Un logiciel avec plusieurs processus sur une même machine.

## Pourquoi des systèmes répartis ?

- Aspects économiques (rapport prix/performance)
- Adaptation de la structure d'un système à celle des applications (géographique ou fonctionnelle)
- Besoin d'intégration (applications existantes)
- Besoin de communication et de partage d'information
- Réalisation de systèmes à haute disponibilité
- Partage de ressources (programmes, données, services)
- Réalisation de systèmes à grande capacité d'évolution

## Exemples:

- WWW
- Contrôle du trafic aérien
- Système de courtage
- Banques
- Super calcul distribué
- Système de fichier distribué
- DNS
- Systèmes Pair-à-pair (P2P)

## Quelques domaines d'application des systèmes répartis

- CFAO, Ingénierie simultanée
  - Coopération d'équipes pour la conception d'un produit
  - Production coopérative de documents
  - Partage cohérent d'information
- Gestion intégrée des informations d'une entreprise
  - Intégration de l'existant
- Contrôle et organisation d'activités en temps réel
- Centres de documentation, bibliothèques
  - Recherche, navigation, visualisation multimédia
- Systèmes d'aide à la formation

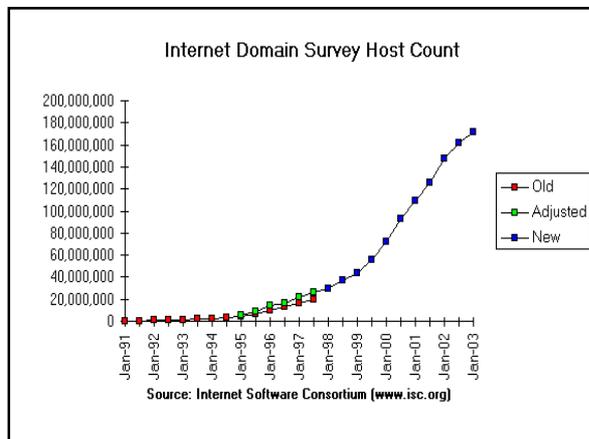
### III/ Objectifs d'un système distribué

- **Transparence (Masquer la répartition)**
  - Uniformité des accès locaux et distants  
La séparation physique entre machines et les différences matériels/logiciels pour les accès sont invisibles par l'utilisateur.
  - Localisation des ressources non perceptible  
(nom logique ex: URL <http://www.labri.fr/>)
  - Migration des ressources possible sans interférence avec la localisation physique  
(ex. transférer un objet uniquement par son nom logique sans modification de ce nom et sans modification de l'environnement d'un utilisateur)

- Réplication de ressources non visible
- Concurrence d'accès aux ressources non perceptible  
(ex. accès à un même fichier ou une table dans une base de données: mécanisme de verrou ou de transaction)
- Invisibilité du parallélisme offert par l'environnement d'exécution
- Tolérance aux pannes permettant à un utilisateur de ne pas s'interrompre (ou même se rendre compte) à cause d'une panne d'une ressource

- **Ouverture**
  - Services offerts selon des règles standards qui décrivent la syntaxe et la sémantique de ces services (Interfaces publiées, ex. IDL)
  - Interopérabilité des matériels (de fournisseurs différents)
  - Portabilité
  - Flexibilité (facilité d'utilisation et de configuration)
  - Extensibilité (ajout/MAJ de composants sans en affecter les autres)

- **Mise à l'échelle (scalability)**
  - fonctionne efficacement dans différentes échelles:
    - Deux postes de travail et un serveur de fichiers
    - Réseau local avec plusieurs centaines de postes de travail et serveurs de fichiers
    - Plusieurs réseaux locaux reliés pour former un Internet  
( En janvier 03, le nombre de machines reliés à Internet est de : 171 638 297)



- Tolérance aux pannes
  - Pannes franches
  - Pannes byzantines
  - Détection de pannes (difficulté et même impossibilité de détection pour certains systèmes, suspicion de machines) e.g. connexion par un navigateur à un serveur distant qui répond pas !!
  - Correction d'erreurs (de fichiers/messages corrompus)
  - Reprise sur pannes (techniques de journalisation dans les BD)  
(éventuellement système dégradé)

- Sécurité
  - Confidentialité (authentification)
  - Intégrité (protection contre les falsification et les corruptions)
  - Disponibilité (accès aux ressources)  
e.g. commerce électronique, banque en ligne.

## IV/ Systèmes distribués vs parallèles

**Systèmes Parallèles.** Une machine multiprocesseurs avec un environnement du type SIMD (tous les processeurs exécutent le même programme et ont une vision uniforme de l'état global du système).  
Extensible à un réseau de machines asynchrones fortement couplées

**Systèmes distribués.** processus indépendants sur des machines distinctes et communiquant par échange de messages asynchrones (en général, des réseaux faiblement couplés).

*Pas de consensus sur ces définitions...*

## Caractéristiques du parallélisme/distribué

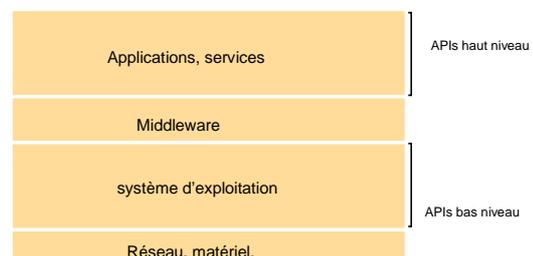
- Objectifs: optimiser les solutions d'un problème (e.g. calcul scientifique, calcul matriciel, tri)
- Calcul de complexité : temps et accès mémoire (pas le temps de communication ou nombre de messages)
- La topologie est généralement fixe (grille, hypercube, grappes)

## V/ Interlogiciel (Middleware)

- Le *middleware* (interlogiciel) est la couche logicielle située entre les couches basses (systèmes d'exploitation, protocoles de communication) et les applications dans un système informatique réparti (CORBA, EJB, COM, etc.).

- Buts:
  - Fournir une interface ou API de haut niveau aux applications
  - Masquer l'hétérogénéité des systèmes matériels et logiciels sous-jacents
  - Rendre la répartition aussi invisible (transparente) que possible
  - Faciliter la programmation répartie (développement, évolution, réutilisation, portabilité des applications)
- <http://www.objectweb.org/>

## Couches logicielles et matérielles dans un SD (le middleware)



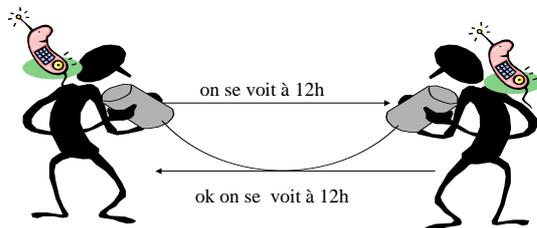
## VI/ Où sont les problèmes ?

- Quel est le nombre d'emails qui circulent sur Internet ?
- Y'a-t-il un algorithme pour détecter la terminaison distribuée d'une application ?
- Comment modéliser et exprimer les algorithmes distribués ? Comment les prouver ?
- Comment les analyser, calculer leur complexité, les classifier etc... ?
- Pour écrire une appli. Distribuée, un système distribue, il faut d'abord un algorithme...

- Nécessite de simplifier et maîtriser la complexité des systèmes et des algorithmes distribués
- Difficulté de l'algorithmique distribuée / centralisé:
  - Pas de connaissance de l'état global
  - Absence de temps universel (ou horloge globale)
  - Non déterminisme (lié souvent au problème du synchronisme)
  - Et surtout pas de modèle « universel » et standard pour l'algorithmique distribuée

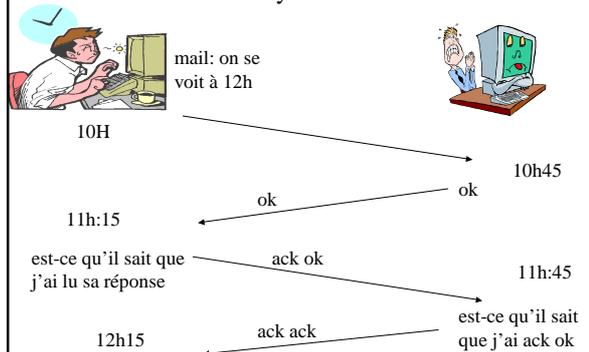
## Modèles de communication synchrone /asynchrone

### Synchrone

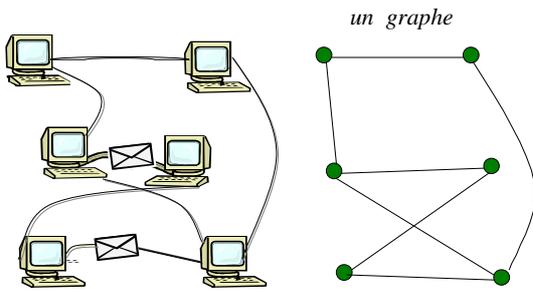


Même notion de temps, transmission instantanée, généralement bornée

### • Communication asynchrone:



## Représentation : abstraction



## Modélisation d'un système distribué

- Système distribuée : graphe (non orienté, connexe, simple)
- sommet : processus
- arête : canal de communication
- algorithme distribué local : algorithme qui s'exécute sur chaque sommet (en utilisant uniquement le contexte local)

## Les réseaux anonymes /avec identités

- anonymes: pas d'identités (numéros distincts. e.g. IP)
- avec identités (chaque sommet possède un identité (un numéro) unique)

En général, il est plus facile de construire un algo sur des graphes avec identités.

- L'état d'un processus est codé par une étiquette:
- Le changement d'état : changement d'étiquette
- Les algorithmes : arbre recouvrant, élection, terminaison, exclusion mutuelle.