

Cours 1

Chapitre 1

Introduction aux systèmes distribués

- 1.1 Qu'est-ce qu'un système distribué (réparti)
- 1.2 Objectifs & Propriétés souhaitées
- 1.3 Les difficultés
- 1.4 Un aspect pratique : les middlewares
- 1.5 Types des systèmes distribués
- 1.6 Les fondamentaux théoriques

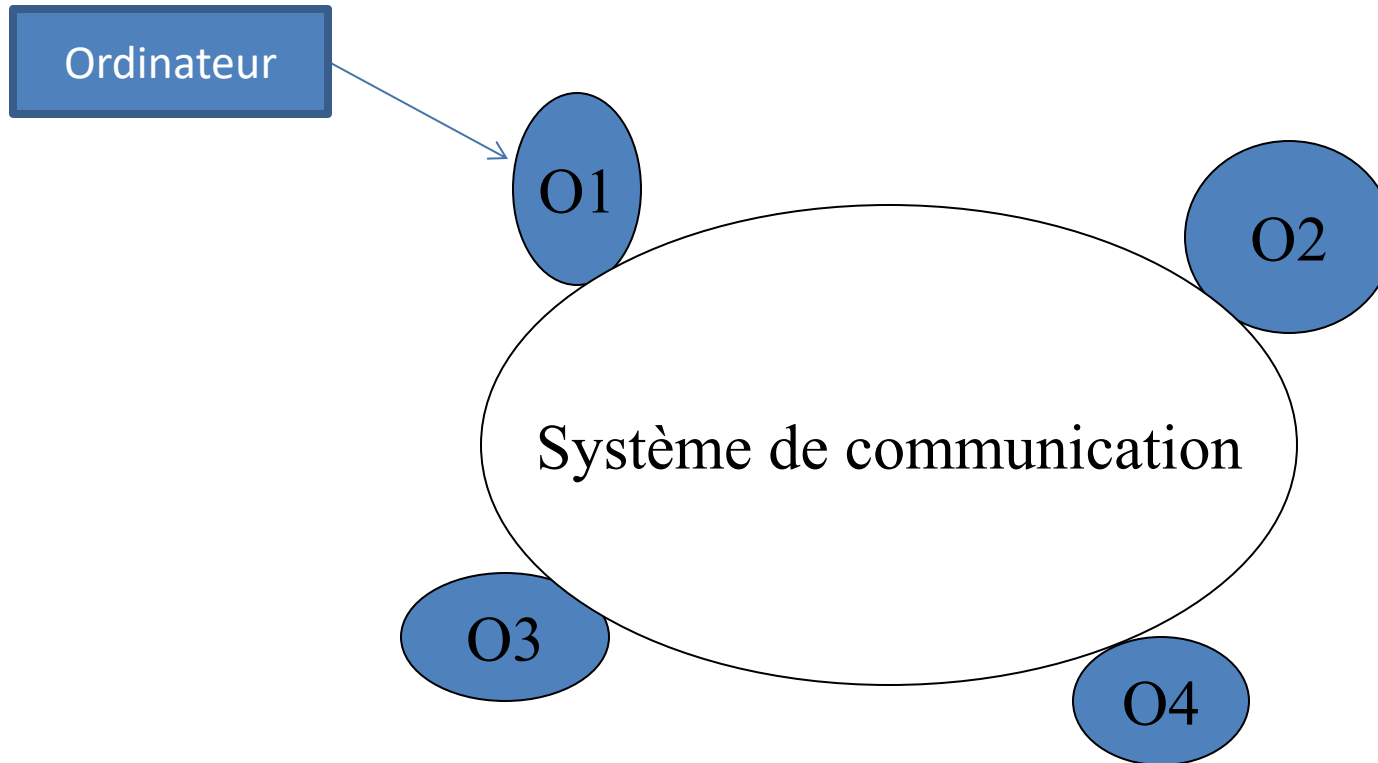
Qu'est qu'un système distribué ?

- Systeme distribué vs système centralisé
- Système centralisé : tout est localisé sur la même machine et accessible par le programme
 - Système logiciel s'exécutant sur une seule machine
 - Accédant localement aux ressources nécessaires (données, code, périphériques, mémoire ...)

Qu'est qu'un système distribué ?

- Système distribué : des définitions parmi d'autres
 - Ensemble de **ressources physiques et logiques (codes-données)** géographiquement **dispersées et reliées par un réseau de communication**.
 - A distributed system is a collection of autonomous computing elements that appears to its users as a single coherent system. [*Textbook*]
 - Distributed systems consist of autonomous computers that work together to give the appearance of a single coherent system. [*Textbook*]

Qu'est qu'un système distribué ?



Objectif d'un système distribué : Échanger de l'information en vue de coordonner la réalisation **d'une tâche commune**

Un peu d'histoire

- Le concept de **système distribué** n'a pas été introduit par une seule personne à un instant précis, mais il a **émergé progressivement** dans les années 1970 avec plusieurs chercheurs clés.

Origines du concept de systèmes distribués

- Années 1960
 - L'idée dominante était celle du **temps partagé (time-sharing)** sur des ordinateurs centraux.
 - **ARPANET (1969)** : le premier réseau à commutation de paquets — base des communications entre ordinateurs indépendants.
 - **Project Genie (Berkeley)** et **Multics (MIT)** : ont introduit des concepts de partage de ressources à distance.

Origines du concept de systèmes distribués

- Années 1970 *naissance du concept formel*, parmi les **chercheurs fondateurs** :
 - Leslie Lamport (1978) : *Time, Clocks, and the Ordering of Events in a Distributed System* », *Communications of the ACM*, 1978.
 - Andrew D. Birrell et Bruce Nelson (1984) : **Remote Procedure Call (RPC)** au Xerox PARC.
 - Andrew S. Tanenbaum (années 1980) : A popularisé et formalisé la discipline avec ses livres et ses systèmes expérimentaux (*Amoeba*, *MINIX*).

Origines du concept de systèmes distribués - conclusion -

- Le *père fondateur théorique* du concept de système distribué est **Leslie Lamport**, mais la discipline est le fruit des travaux collectifs d'une génération de chercheurs entre **1974 et 1985**.

1.2 Objectifs & Propriétés souhaitées d'un système distribué

- **Utiliser et partager des ressources distantes**
 - Système de fichiers : utiliser ses fichiers à partir de n'importe quelle machine
 - Imprimante : partagée entre toutes les machines
- **Optimiser l'utilisation des ressources disponibles**
 - Calculs scientifiques distribués sur un ensemble de machines
- **Système plus robuste**
 - Duplication pour **fiabilité** : deux serveurs de fichiers dupliqués, avec sauvegarde
 - Plusieurs éléments identiques pour résister à la montée en charge ...

1.3 Les difficultés (Caractéristiques) & Bases de la conception des systèmes distribués

- Caractéristiques clés doivent être prises en compte lors de la conception d'un système distribué :
- Hétérogénéité ■ Distribution et coordination
- Partage de ressources ■ Transparence
- Expansible ■ Sécurité
- Performance ■ Concurrence
- Tolérance aux fautes

Particularités des systèmes distribués

- **Distribution et coordination**
 - Connaissance des éléments formant le système : besoin d'identification et de localisation.
- **Communication à distance**
 - Techniques et protocoles de communication
 - Contraintes du réseau : fiabilité (perte de données) et temps de réponse (dépendant du type de réseau et de sa charge)
- **Naturellement concurrent et parallèle**
 - Chaque élément sur chaque machine est autonome
 - Besoin de synchronisation, coordination entre éléments distants et pour l'accès aux ressources (exclusion mutuelle ...)

Particularités des systèmes distribués

- **Hétérogénéité:**
 - Des machines utilisées (puissance, architecture matérielle...)
 - Des systèmes d'exploitation différents tournant sur ces machines
 - Des réseaux utilisés : impact sur performances, débit, disponibilité ...
 - Réseau local rapide
 - Internet
 - Réseaux sans fil

Particularités des systèmes distribués

- **Partage de ressources** : matérielles (disque, imprimante) et logiques (fichier, base de données)
 - **Gestionnaire de ressources** : Protocole de gestion des concurrences.

Particularités des systèmes distribués

- **Expansible** : extensibilité du système (permet à des composants d'être rajoutés ou remplacés).
 - **Matériel** : périphériques, mémoire, interfaces, ...
 - **Logiciel** : protocole, pilote, ...

Particularités des systèmes distribués

- **Performance (Passage à l'échelle):** aptitude du système à bien fonctionner quand le nombre d'utilisateurs augmente ou celui des ressources
 - Est-ce que le système peut continuer à fonctionner efficacement avec l'augmentation de la charge (utilisateurs, ordinateurs,.etc)?

Particularités des systèmes distribués

- **Transparence :**
 - cacher aux utilisateurs la distribution des ressources (le système réparti est vu comme un tout)
 - But est de cacher l'architecture, le fonctionnement de l'application ou du système distribué pour apparaître à l'utilisateur comme une application unique cohérente.
 - L'ISO définit RM-ODP (Reference Model for Open Distributed Processing) qui comprend **8** formes de transparence :
 - Accès, localisation, concurrence, réplication, mobilité, panne, performance, passage à échelle (scalabilité)

Particularités des systèmes distribués

8 formes de transparence :

1. Transparence d'accès

- Accès aux ressources distantes et locales en utilisant des opérations identiques

2. Transparence de localisation

- Accès aux éléments/ressources sans le besoin de connaître leur localisation

3. Transparence de concurrence

- Exécution possible de plusieurs processus en parallèle sans produire des interférences entre eux.

4. Transparence de réplication

- Possibilité de dupliquer certains éléments/ressources pour augmenter la fiabilité du systèmes sans que l'utilisateur sache l'existence de ces répliquas.

5. Transparence de mobilité

- Possibilité de déplacer des éléments/ressources sans affecter les opérations normales des utilisateurs.

6. Transparence de panne

- Cacher l'existence de pannes et permettre aux utilisateurs et aux applications de compléter leurs tâches.

Particularités des systèmes distribués

7. Transparence de performance

- Possibilité de reconfigurer le système pour en augmenter les performances avec l'augmentation de la charge.

8. Transparence de passage à l'échelle

- Doit supporter l'augmentation de la taille du système (nombre d'éléments, de ressources ...) sans le besoin de reconfigurer la structure du système ou les algorithmes de l'application

Particularités des systèmes distribués

Tolérance aux fautes: Capacité d'un système à gérer et résister à un ensemble de problèmes par fournir un service et remplir la fonction du système en dépit des fautes.

1. Tolérance matérielle.

1. Serveurs redondants (*Microsoft Failover Cluster, Kubernetes node failover*)
2. Disques en RAID (*Redundant Array of Inexpensive Disks*)

2. Tolérance logicielle.

1. Réplication de bases de données (*Google Spanner, MongoDB Replica Sets.*)
2. Rollback et Checkpointing (*Hadoop YARN, MPI*)
3. Protocoles de consensus tolérants aux fautes (*Paxos, Raft, ou PBFT*)

1.3 Objectifs & Propriétés souhaitées

résumé:

Mimic the electrical power grid



Question ?

- Comment peut on arrivé à prendre toutes ces contraintes de **conception** des systèmes distribués !?
- Solutions partielles :
 - Réplication de données
 - Mais le problème d'incohérence des données s'engendre.
 - Calcule distribué.
 - Allocation de processeurs
 - Ordonnancement des processus
 - Communication.
 - Minimiser le Temps de réponse
 - Tolérance aux pannes (lien, paquets)

1.4 Un aspect pratique : les middlewares

- Un middleware (ou intergiciel) est une couche logicielle qui se place entre l'application et le système d'exploitation, pour offrir des services utiles à la construction de systèmes distribués.

Les middlewares

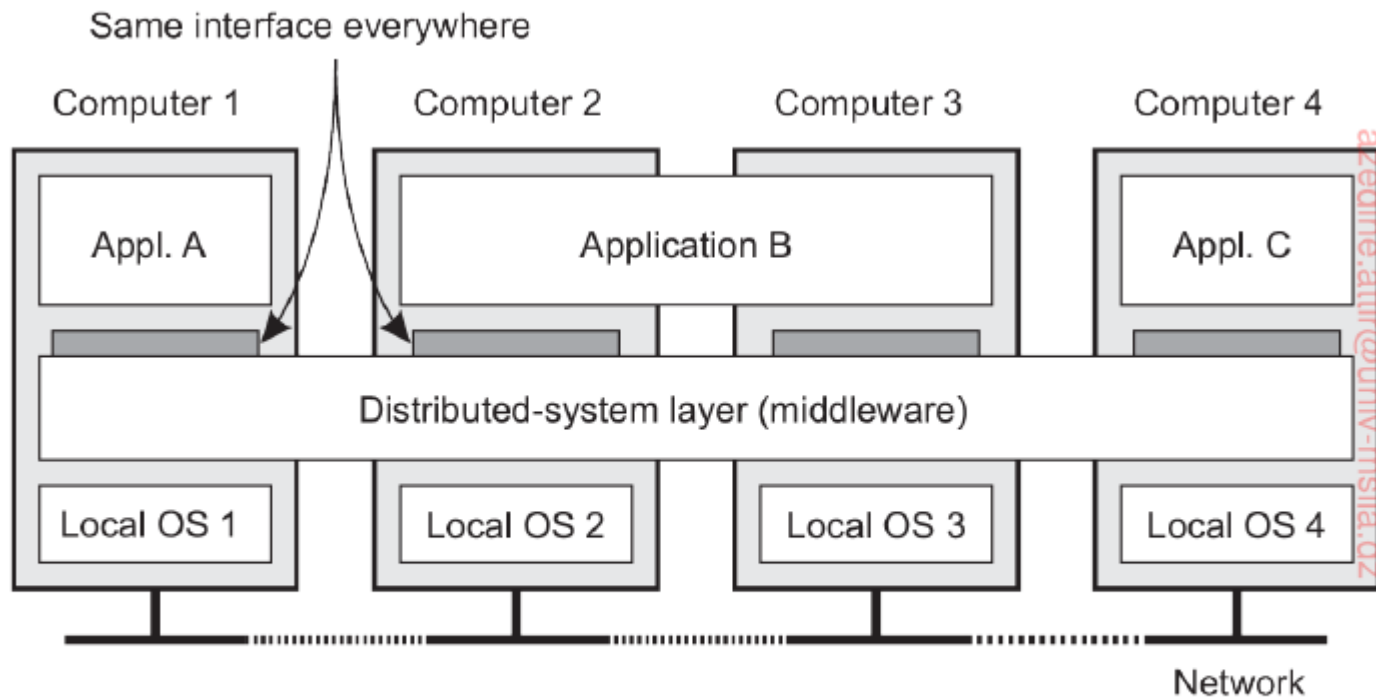


Figure 1.1: A distributed system organized in a middleware layer, which extends over multiple machines, offering each application the same interface.

1.4 Un aspect pratique : les middlewares

- Exemples :
 - **Communication** : gRPC, RMI, CORBA.
 - **Messagerie** : Kafka, RabbitMQ.
 - **Bases de données distribuées** : Google Spanner
MongoDB.
 - **Cloud et microservices** : Kubernetes, Docker
Swarm Istio

1.5 Types of distributed systems

Type	Exemple concret
Calcul distribué (clusters, grid, cloud)	AWS, Azure, Supercalculateurs
Stockage distribué (FS, DB)	HDFS, Google Spanner, Cassandra
Communication distribuée	gRPC, Kafka, ZooKeeper
Pair-à-pair (P2P)	BitTorrent, Blockchain, IPFS
Tolérance aux fautes	Blockchain, Paxos, Raft
Temps réel	Systèmes avioniques, IoT critique

1.6 Les fondamentaux théoriques

- Temps : synchronisation des horloges
- Exécution répartie
- Etat global
- Communication : Direct & Indirect Communication -TP-
- Processus.
- Nommage.
- Synchronisation : EM, deadlock, election, termination,...
- Consistency and replication : modèles, protocoles et implémentations.
- ...

Réseau vs système distribué

Aspect	Réseau	Système Distribué
Définition	Infrastructure de communication entre machines	Ensemble de machines qui coopèrent pour donner l'illusion d'un seul système
Niveau	Matériel + protocole de communication (couche transport)	Logiciel/middleware au-dessus du réseau
But	Connecter	Coordonner et collaborer
Exemple	Internet, LAN, Wi-Fi	Google Drive, Netflix, Blockchain, Cloud

- **Réseau** = les routes et autoroutes entre les villes.
- **Système distribué** = un service de livraison (Amazon, DHL) qui utilise ces routes, avec organisation, suivi, gestion des colis perdus et optimisation.

Concepts Liés

Notion	Caractéristiques	Exemple
Système parallèle	Processeurs fortement couplés, mémoire partagée, haute performance	Supercalculateurs, GPU
Système distribué	Machines autonomes reliées par un réseau, service unifié	Google Docs, Blockchain
Réseau	Infrastructure de communication, pas de service global par défaut	Internet, LAN
Multiprocesseur	Plusieurs CPU dans une seule machine	Serveurs multi-cœurs
Cluster	Groupe homogène de serveurs proches, souvent dédié au calcul	Hadoop cluster

Pour plus de connaissances sur les systèmes distribués lire :

1. TextBook : Distributed Systems 3rd edition (2017).
2. Abdelouahid Derhab, Systèmes distribués, maitre de recherche A (CERIST), notes de cours.