

# Rapport Synthétique : Les Types de Systèmes Distribués

MORSLI Mohamed Abdennour

20 octobre 2025

## Devoir maison 1 de System distribués

### 1. Introduction

Un **système distribué** est un ensemble d'ordinateurs autonomes interconnectés qui coopèrent pour exécuter des tâches comme s'ils formaient un système unique. Ces systèmes sont essentiels dans les domaines du cloud computing, des applications web, du big data et de l'Internet des objets (IoT).

L'objectif de ce rapport est de présenter trois types majeurs de systèmes distribués, leurs principes de fonctionnement, leurs architectures et des exemples concrets.

### 2. Les Systèmes Distribués de Type Client–Serveur

#### 2.1. Principe de fonctionnement

Le modèle **client–serveur** repose sur la communication entre des clients et un ou plusieurs serveurs centralisés. Les clients envoient des requêtes et le serveur les traite avant de renvoyer la réponse.

#### 2.2. Architecture

L'architecture est généralement **centralisée** :

- **1-Tier** : toutes les fonctions sur un seul système.
- **2-Tiers** : séparation entre client et serveur (ex. application connectée à une base de données).
- **3-Tiers ou N-Tiers** : ajout de couches intermédiaires pour une meilleure scalabilité.

### 2.3. Répartition

Les serveurs peuvent être physiquement distribués sur plusieurs machines, mais logiquement perçus comme un seul service, souvent équilibrés via un *load balancer*.

### 2.4. Exemples réels

- Serveurs web (Apache, Nginx) répondant aux navigateurs clients.
- Applications bancaires connectées à des serveurs centraux.
- Jeux en ligne multijoueurs.

## 3. Les Systèmes Pair-à-Pair (P2P)

### 3.1. Principe de fonctionnement

Contrairement au modèle client-serveur, le modèle **P2P** est **décentralisé**. Chaque nœud (ou pair) agit à la fois comme client et serveur, partageant des ressources (fichiers, bande passante, puissance de calcul).

### 3.2. Architecture

Deux variantes existent :

- **P2P pur** : tous les pairs sont égaux (ex. BitTorrent).
- **P2P hybride** : un serveur central aide à la découverte des pairs (ex. Skype).

Les données sont réparties entre pairs via des tables de hachage distribuées (*Distributed Hash Table – DHT*).

### 3.3. Répartition

Les fichiers sont fragmentés et stockés sur plusieurs pairs. Lorsqu'un utilisateur télécharge un fichier, il reçoit différents fragments de plusieurs pairs simultanément.

### 3.4. Exemples réels

- BitTorrent : partage de fichiers fragmentés.
- Blockchain / Bitcoin : validation décentralisée des transactions.
- IPFS : stockage distribué de fichiers.

## 4. Les Systèmes Distribués Basés sur le Cloud (Cloud Computing)

### 4.1. Principe de fonctionnement

Les systèmes de **cloud computing** offrent des ressources informatiques accessibles via Internet à la demande. Ils reposent sur la **virtualisation** et la **mutualisation** des

ressources, permettant à plusieurs utilisateurs d'utiliser la même infrastructure sans interférer entre eux.

## 4.2. Architecture

L'architecture comprend trois couches principales :

- **IaaS (Infrastructure as a Service)** : machines virtuelles, stockage (ex. AWS EC2).
- **PaaS (Platform as a Service)** : environnements de développement prêts à l'emploi.
- **SaaS (Software as a Service)** : applications accessibles via navigateur (ex. Gmail, Zoom).

Les services sont répartis sur plusieurs *data centers* interconnectés.

## 4.3. Répartition

La répartition s'appuie sur des **clusters** de serveurs, des **conteneurs** (Docker, Kubernetes) et des mécanismes de réplication pour garantir la disponibilité.

## 4.4. Exemples réels

- Amazon Web Services (AWS), Google Cloud Platform, Microsoft Azure.
- Netflix : serveurs mondiaux pour le streaming.
- Dropbox : réplication de fichiers pour disponibilité maximale.

## 5. Comparaison Synthétique

Type	Architecture	Répartition	Points forts	Exemples
<b>Client–Serveur</b>	Centralisée	Clients → Serveur	Simplicité, gestion centralisée	Applications web, banques
<b>Pair-à-Pair (P2P)</b>	Décentralisée	Entre pairs	Résilience, rapidité	BitTorrent, Blockchain
<b>Cloud Computing</b>	Hybride / Multi-niveau	Data centers globaux	Scalabilité, flexibilité	AWS, Google Cloud, Netflix

## 6. Conclusion

Les systèmes distribués visent à répartir les ressources et les traitements pour améliorer les **performances**, la **fiabilité** et la **disponibilité**. Le modèle **client–serveur** domine les applications web classiques, le **P2P** favorise la décentralisation, et le **cloud computing** offre la puissance et la flexibilité nécessaires aux applications à grande échelle.

L'avenir des systèmes distribués combine ces approches, notamment dans les architectures modernes **microservices** et **edge computing**, pour des solutions plus performantes et résilientes.