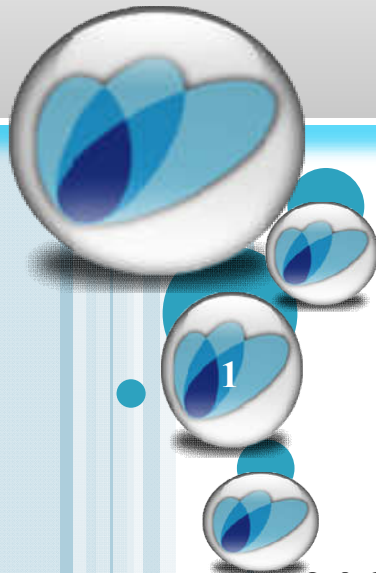
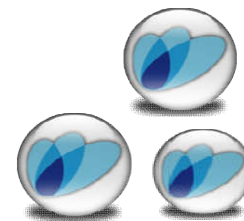


Cours Construction d'Applications Réparties (CAR) 1^{ère} Année Master Génie Logiciel



2020-2021



Mme Z.LAAREDJ

Objectifs de cour

- Présenter les principes de construction et le fonctionnement des applications réparties
- Apprendre la conception d'applications réparties
 - motivations et concepts
 - architecture et exemples
 - problèmes et solutions
- Connaitre les technologies majeures pour le développement des applications réparties.
- Comprendre les solutions
 - Sockets Java et RPC
 - Web Services
 - Objets réparties en Java (RMI, CORBA)

Introduction aux applications réparties

Plan

- Introduction aux systèmes distribués/répartis
 - Présentation générale
 - Exemples de systèmes distribués
 - Intérêts des systèmes distribués
- Propriétés des systèmes distribués
- Applications réparties

Introduction aux systèmes distribués

➤ Présentation générale

■ **Système distribué en opposition à un système centralisé**

◆ **Système centralisé**

- Tout est localisé sur la même machine et accessible par le programme
- Système logiciel s'exécutant sur une seule machine
- Les applications accèdent localement aux ressources nécessaires (données, code, périphériques, mémoire ...)

◆ **Système distribué**

- Ensemble d'ordinateurs indépendants connectés en réseau et communiquent via ce réseau
- Cet ensemble apparaît du point de vue de l'utilisateur comme une seule entité

Introduction aux systèmes distribués

- **Vision matérielle d'un système distribué** : architecture matérielle
 - Machine multi-processeurs avec mémoire partagée
 - Cluster d'ordinateurs dédiés au calcul/traitement massif parallèle
 - Ordinateurs standards connectés en réseau
- **Vision logicielle d'un système distribué**
 - Système logiciel composé de plusieurs entités s'exécutant indépendamment et en parallèle sur un ensemble d'ordinateurs connectés en réseau

Introduction aux systèmes distribués

➤ Dans ce cours

- Conception logicielle des systèmes distribués: développement d'applications réparties

- Par défaut sur une architecture matérielle de type ordinateurs connectés en réseau

➤ Exemples de systèmes distribués

- Serveur FTP

- Serveur Web

Introduction aux systèmes distribués

➤ Autres exemples

■ Calculs scientifiques

- ◆ Plusieurs architectures matérielles généralement utilisées
 - Ensemble de machines identiques reliées entre elles par un réseau dédié et très rapide (cluster)
 - Ensemble de machines hétérogènes connectées dans un réseau local ou bien encore par internet

■ Principe général

- ◆ Un (ou des) serveur distribue des calculs aux machines clients
- ◆ Un client exécute son calcul puis renvoie le résultat au serveur

■ **Avantage** : utilisation d'un maximum de ressources de calcul

■ **Inconvénient** : Si réseau ou serveur plante, alors arrêt du système

Introduction aux systèmes distribués

➤ Intérêts des systèmes distribués

■ Utiliser et partager des ressources distantes

- ◆ Un même service peut être utilisé par plusieurs acteurs, situés à des endroits différents
- ◆ 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 ...

Introduction aux systèmes distribués

➤ Inconvénients

- **Si problème au niveau du réseau** : le système marche mal ou plus du tout

- **Bien souvent, un élément est central au fonctionnement du système : serveur**

 - ◆ Si serveur plante : plus rien ne fonctionne

 - ◆ Goulot potentiel d'étranglement si débit d'information très important

- **Sans élément central**

 - ◆ Gestion du système totalement décentralisée et distribuée

 - ◆ Nécessite la mise en place d'algorithmes +/- complexes

Les systèmes distribués

- Un système réparti (ou distribué de «distributed system») est un système comprenant un ensemble de processus et un système de communication
- Ensemble composé d'éléments reliés par un système de communication
 - Les éléments ont des fonctions de traitement (processeurs), de stockage (mémoire), de relation avec le monde extérieur (capteurs, actionneurs)
 - Les différents éléments du système ne fonctionnent pas indépendamment mais collaborent à une ou plusieurs tâches communes.
 - Conséquence : une partie au moins de l'état global du système est partagée entre plusieurs éléments (sinon, on aurait un fonctionnement indépendant)

Les systèmes distribués

➤ **Système distribué = éclaté**

- Connaissance des éléments formant le système : besoin de d'identification et de localisation
- Gestion du déploiement et de la présence d'éléments essentiels

➤ **Communication à distance est centrale**

- Techniques et protocoles de communication
- Contraintes de réseau : fiabilité (perte de données) et temps de propagation (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...)

Les propriétés des systèmes distribués

➤ Hétérogénéité

- Des machines utilisées (puissance, architecture matérielle...)
- Des systèmes d'exploitation tournant sur ces machines
- Des langages de programmation des éléments logiciels formant le système
- Des réseaux utilisés : impact sur performances, débit, disponibilité ...
- Réseau local rapide
- Internet
- Réseaux sans fil

Les propriétés des systèmes distribués

➤ Fiabilité des systèmes distribués

Nombreux points de pannes ou de problèmes potentiels:

■ Réseau

- ◆ Une partie du réseau peut-être inaccessible
- ◆ Les temps de communication peuvent varier considérablement selon la charge du réseau
- ◆ Le réseau peut perdre des données transmises

■ Machine

- ◆ Une ou plusieurs machines peut planter, engendrant une paralysie partielle ou totale du système

■ Peut augmenter la fiabilité par redondance, duplication de certains éléments

- ◆ Mais rend plus complexe la gestion du système

Les propriétés des systèmes distribués

➤ Fiabilité des systèmes distribués

■ Tolérance aux fautes

- ◆ Une partie du réseau peut-être inaccessible
- ◆ Capacité d'un système à gérer et résister à un ensemble de problèmes
- ◆ Le système doit pouvoir fonctionner (au moins de façon dégradée) même en cas de défaillance de certains de ses éléments
- ◆ Le système doit pouvoir résister à des perturbations du système de communication (perte de message, déconnexion temporaire, performances dégradées)
- ◆ Le système doit pouvoir facilement s'adapter pour réagir à des changements d'environnement ou de conditions d'utilisation

Les propriétés des systèmes distribués

➤ Sécurité

- Nature d'un système distribué fait qu'il est beaucoup plus sujet à des attaques

- ◆ Communications à travers le réseau peuvent être interceptées
- ◆ On ne connaît pas toujours bien un élément distant avec qui on communique

- Le système doit pouvoir résister à des attaques contre sa sécurité (violation de la confidentialité, de l'intégrité, usage indu de ressources, déni de service)

■ Solutions

- ◆ Connexion sécurisée par authentification avec les éléments distants
- ◆ Cryptage des messages circulant sur le réseau

Les propriétés des systèmes distribués

➤ **Transparence**

- Fait pour une fonctionnalité, un élément d'être invisible ou caché à l'utilisateur ou un autre élément formant le système distribué

 - ◆ Devrait plutôt parler d'opacité dans certains cas ...

- But: 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 plusieurs transparences (norme RM-ODP): accès, localisation, concurrence, réplication, mobilité, panne, performance, échelle

Les propriétés des systèmes distribués

- **Transparence d'accès** : accès à des ressources distantes aussi facilement que localement et accès aux données indépendamment de leur format de représentation
- **Transparence de localisation** : accès aux éléments/ressources indépendamment de leur localisation
- **Transparence de concurrence** : exécution possible de plusieurs processus en parallèle avec utilisation de ressources partagées
- **Transparence de réplication** : possibilité de dupliquer certains éléments/ressources pour augmenter la fiabilité

Les propriétés des systèmes distribués

- **Transparence de mobilité** : possibilité de déplacer des éléments/ressources
- **Transparence de panne** : doit supporter qu'un ou plusieurs éléments tombe en panne
- **Transparence de performance** : possibilité de reconfigurer le système pour en augmenter les performances
- **Transparence d'échelle** : doit supporter l'augmentation de la taille du système (nombre d'éléments, de ressources ...), ainsi le système doit préserver ses performances lorsque sa taille croît (nombre d'éléments, nombre d'utilisateurs, étendue géographique)

Les propriétés des systèmes distribués

➤ Quelques difficultés

Propriété	Difficulté engendrée
Asynchronisme du système de communication (pas de borne supérieure stricte pour le temps de transmission d'un message)	Difficulté de détecter les défaillances
Dynamisme (la composition du système change en permanence)	Difficulté de définir un état global du système et la difficulté d'administrer le système
Grande taille (nombre de composants, d'utilisateurs, dispersion géographique)	La capacité de croissance (scalability) est une propriété importante, mais difficile à réaliser

Applications réparties

➤ Distinction entre “système” et “application”

■ **Système** : gestion des ressources communes et de l'infrastructure, lié de manière étroite au matériel sous-jacent

- ◆ Système d'exploitation : gestion de chaque élément
- ◆ Système de communication : échange d'information entre les éléments
- ◆ Caractéristiques communes : cachent la complexité du matériel et des communications, et fournissent des services communs de plus haut niveau d'abstraction

Applications réparties

➤ Distinction entre “système” et “application”

■ **Application** : réponse à un problème spécifique, fourniture de services à ses utilisateurs (qui peuvent être d'autres applications)

- ◆ Utilise les services généraux fournis par le système
- ◆ La distinction n'est pas toujours évidente, car certaines applications peuvent directement travailler à bas niveau (au contact du matériel). Exemple : systèmes embarqués, réseaux de capteurs

Applications réparties

➤ Distinction entre “système” et “application”

Système Réparti	Application répartie
est un ensemble de systèmes calculatoires autonomes (exp: ordinateur, serveur, terminal, etc.) sans mémoire physique commune qui communiquent à travers un réseau quelconque	est un ensemble de processus qui tournent sur un système réparti afin de fournir ou utiliser un service déterminé

Applications réparties

Application répartie = traitements coopérants sur des données réparties

- **Traitement** : description : programme, exécution : flot d'exécution (processus)
- **Coopération** = communication + synchronisation en définissant
 - Modèle d'exécution
 - Interface de programmation (et/ou langage)
 - Modèle de programmation
 - Outils de développement
 - Environnement d'exécution : services systèmes (pour différents types d'infrastructures)

Applications réparties

➤ Exemples d'applications réparties

- Navigation web, transfert de fichiers;
- Guichets de banque (GAB (Guichet Automatique de Banque), DAB (Distributeur Automatique de Banque));
- Commerce électronique;
- Jeux en réseaux;
- Télévision interactive

Applications réparties

➤ Programmation classique versus programmation répartie

- La plupart des applications réparties sont de type client/serveur: le client demande des services à un serveur

- **En programmation classique**, lorsque un programme a besoin d'un service, il appelle localement une fonction / procédure / méthode d'une librairie, d'un objet, etc.

- **En programmation répartie**, l'appel de fonction / procédure / méthode peut se faire à distance

- Proposer des méthodes / concepts / outils permettant de simplifier le développement d'application réseau client/serveur, en essayant de s'abstraire de l'aspect « distant »

Applications réparties

➤ Programmation classique versus programmation répartie

Programmation classique	Programmation répartie
<p>L'utilisateur du service et le fournisseur de service se trouvent sur la même machine:</p> <ul style="list-style-type: none">• Même OS• Même espace mémoire• Même capacité de calcul CPU• Pas de problème de transport• Disponibilité du service assuré (tant que l'on a accès à la librairie)	<p>L'utilisateur et le fournisseur de service ne se trouvent pas sur la même machine: deux machines différentes (sans compter celles traversées)</p> <ul style="list-style-type: none">• OS différents• Espace mémoire non unitaire : "passer un pointeur comme argument"?• Problème de transport : pare-feu (firewall), réseau, etc.• Retrouver le service ? où se trouve-t-il ? qui le propose?

Applications réparties

➤ Programmation classique versus programmation répartie

Programmation classique	Programmation répartie
<p>Un même langage de programmation (sinon Utilisation de binding)</p> <ul style="list-style-type: none">• Même paradigme de programmation• Même représentation des types de base• Même représentation de L'information composite	<p>Deux langages différents</p> <ul style="list-style-type: none">• Représentation de l'information composite différente• Association des paramètres effectifs aux paramètres formels?• Comment gérer les différents types de passage de paramètre ?• Paradigmes de programmation différents : qu'est ce qu'un objet pour un langage procédural ? comment gérer les erreurs ?

Modèles d'interaction dans un système distribué

➤ Les éléments distribués interagissent, communiquent entre eux selon plusieurs modèles possibles

- Client/Serveur
- Diffusion de messages
- Mémoire partagée
- Pair à pair
- ...

➤ Abstraction/primitive de communication basique

- Envoi de message d'un élément vers un autre élément
- A partir d'envois de messages, peut construire les protocoles de communication correspondant à un modèle d'interaction

Modèles d'interaction dans un système distribué

➤ Rôle des messages

- Données échangées entre les éléments (demande de requête, résultat d'une requête, donnée de toute nature, ...)
- Gestion, contrôle des protocoles (acquiescement: message bien reçu, synchronisation, coordination, ...)

Modèles d'interaction dans un système distribué

➤ **Modèle Client/serveur** : deux rôles distincts

- **Client**: demande que des requêtes ou des services lui soient rendus
- **Serveur**: répond aux requêtes des clients

➤ **Interaction**

- Message du client vers le serveur pour faire une requête
- Exécution d'un traitement par le serveur pour répondre à la requête
- Message du serveur vers le client avec le résultat de la requête

➤ **Exemple**: Serveur Web

- Client: navigateur web de l'utilisateur
- Requêtes: récupérer le contenu d'une page HTML gérée ou générée par le serveur.

Modèles d'interaction dans un système distribué

➤ **Modèle Client/serveur** : modèle le plus répandu

- Fonctionnement simple
- Abstraction de l'appel d'un service: proche de l'appel d'une opération sur un élément logiciel (interaction de base en programmation)

➤ **Particularité du modèle**

- Liens forts entre le client et le serveur
- Un client peut aussi jouer le rôle de serveur (et vice-versa) dans une autre interaction
- Nécessité généralement pour le client de connaître précisément le serveur (sa localisation , exemple URL du site web)
- Interaction de type « 1 vers 1 »

Modèles d'interaction dans un système distribué

➤ Diffusion des messages : deux rôles distincts

- **Emetteur:** envoi des messages (ou évènements) à destination de tous les récepteurs (diffusion de type broadcast avec la possibilité de préciser un sous-ensemble de récepteurs (multicast))

- **Récepteurs:** reçoivent les messages envoyés

■ Interaction

- ◆ Emetteur: envoi un message

- ◆ Middleware s'occupe de transmettre ce message à chaque récepteur,

Modèles d'interaction dans un système distribué

➤ Diffusion des messages : deux modes de réception

- Le récepteur va vérifier lui-même qu'il a reçu un message (PULL) via boîtes aux lettres
- Le récepteur est prévenu que le message est disponible et il lui est transmis (PUSH)
 - ◆ Le facteur sonne à la porte pour remettre en main propre le courrier

➤ Particularités du modèle

- Dépendance plus faible entre les participants
- Pas besoin pour l'émetteur d'être directement connecté aux récepteurs ni même de savoir combien ils sont
- Interaction de type « 1 vers N »

Modèles d'interaction dans un système distribué

➤ Mémoire partagée

- Les éléments communiquent via une mémoire partagée à l'aide d'une interface d'accès à la mémoire
 - ◆ Ajout d'une donnée à la mémoire
 - ◆ Lecture d'une donnée dans la mémoire
 - ◆ Retrait d'une donnée de la mémoire
- Le middleware gère l'accès à la mémoire pour chacun des participants
- **Particularité du modèle** : aucun lien, aucune interaction directe entre les participants

Modèles d'interaction dans un système distribué

➤ Mémoire partagée

- Complexité du modèle: dans la gestion de la mémoire (on est dans un système distribué, comment gérer une mémoire dans ce contexte)

- Plusieurs solutions

- ◆ Déployer toute la mémoire sur un seul site (accès simple mais goulot potentiel d'étranglement)

- ◆ Eclater la mémoire sur plusieurs sites (avec ou sans duplication des données , cependant il faut mettre en place des algorithmes +/- complexes de gestion de mémoire distribuée)

Modèles d'interaction dans un système distribué

➤ Mémoire partagée

- Modèle pair à pair (peer to peer)
- Un seul rôle: pas de distinction entre les participants : chaque participant est connecté avec tous les participants d'un groupe et tout le monde effectue les mêmes types d'actions, et pour partager des données, il faut effectuer un calcul commun ...

■ Exemples

- ◆ **Modèles d'échanges de fichiers (bit-torrent)** : avec parfois un mode hybride client/serveur – P2P (le serveur sert à connaître la liste des fichiers et effectuer des recherches, le mode P2P est utilisé ensuite pour les transferts, chacun envoie une partie du fichier à d'autres participants)
- ◆ **Algorithmes de consensus** : chacun mesure une valeur (la même en théorie), l'envoie aux autres et localement chacun exécute le même algorithme pour élire la bonne valeur

Communication dans un système distribué

- Sockets
- Appel de procédure à distance: RPC
- Objets réparties: Java RMI, CORBA
- Web services
- EJB