

Informatique Répartie - Introduction -

Ichrak MEHREZ
(*m_ichrak@hotmail.fr*)

Définitions

- **Définition [Tanenbaum]:** Système réparti

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 (a transparence)

- **Définition 2 [Lamport]:**

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

Définitions

- **Définition 3 [Coulouris et *al*]:**

Un système réparti est un ensemble de machines autonomes connectées par un réseau, et équipées d'un logiciel dédié à la coordination des activités du système ainsi qu'au partage de ses ressources. Coulouris et al. [COU 94].

Définitions

- Un système Réparti (Distribué) est un ensemble d'entités autonomes de calcul (ordinateurs, processeurs, processus, processus léger, etc.) interconnectées et qui communiquent par envoi de messages.
- Exemples:
 - ❑ Réseau physique de machines
 - ❑ Un logiciel avec plusieurs processus sur un même machine
- *Réparti vs Distribué*
 - ❑ Distribué ➔ fabricants d'ordinateurs
 - ❑ Réparti ➔ génie logiciel

Les systèmes répartis: pourquoi?

■ *Objectifs généraux:*

- ❑ 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 communications et de partage d'information
- ❑ Réalisation du système à haute disponibilité
- ❑ Partage des ressources (programmes, données, services)
- ❑ Réalisation de systèmes à grande capacité d'évolution)

Les systèmes répartis: pourquoi?

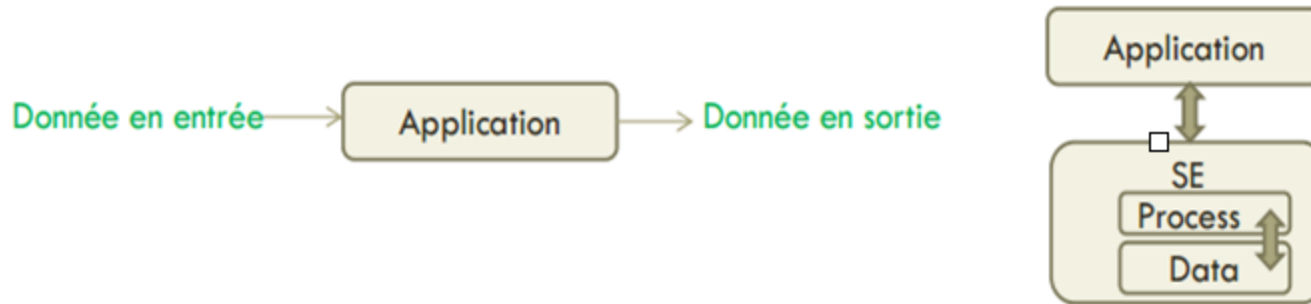
- *Objectifs spécifiques de l'informatique répartie*

- Proposer des méthodes et outils pour simplifier le développement d'application réseau client/serveur, en essayant de s'abstraire de l'aspect « distant »: proposer une programmation « naturelle ».
- Pour les application « lourdes »
 - Décomposer les applications en ensembles de services
 - Rationnaliser la répartition des services pour limiter les échanges d'informations

Application répartie

- *Application informatique*: un programme exécutable sur une machine ou plusieurs machines qui représente la logique de traitement des données manipulées par l'application.
 - ❑ Elle s'exécute en mémoire vive au dessus d'un système d'exploitation.
 - ❑ Avant exécution, elle est stockée sur un support persistant (DD)
 - ❑ L'application émet un résultat sous forme de données soit affichées, soit enregistrées sur un disque

Application répartie



- Dans ce schéma, les traitements, les données d'entrées, les données de sortie sont sur une seule machine.
- Un grand nombre d'applications ne s'exécutent pas intégralement sur un seul nœud de calcul: il s'agit d'application réparties (distribuées)
- Application **répartie**= traitement **coopératifs** sur des données **réparties**.

Application répartie

- Une application répartie est constituée d'un ensemble de processus (d'objets, d'agents, d'acteurs) s'exécutant sur plusieurs sites et reliés par un réseau de communication. Ces processus :
 - ❑ Communiquent entre eux via un réseau de communication
 - ❑ Evoluent de manière cohérente
 - ❑ Remplissent une fonction identifiable
 - ❑ Ne sont pas forcément interdépendants

Système vs 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, fournissent des services communs de plus haut niveau d'abstraction
- **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

Services et Interfaces

- Un système est un ensemble de composants (au sens non technique du terme) qui interagissent.
- Un service est « un comportement défini par **contrat**, qui peut être implémenté et fourni par un composant pour être utilisé par un autre composant, sur la base exclusive du contrat » (*)
- Un service est accessible via une ou plusieurs interfaces
- Une interface décrit l'interaction entre client et fournisseur du service
 - Point de vue opérationnel : définition des opérations et structures de données qui concourent à la réalisation du service
 - Point de vue contractuel : définition du contrat entre client et fournisseur

(*) Bieber and Carpenter, Introduction to Service-Oriented Programming, <http://www.openwings.org>

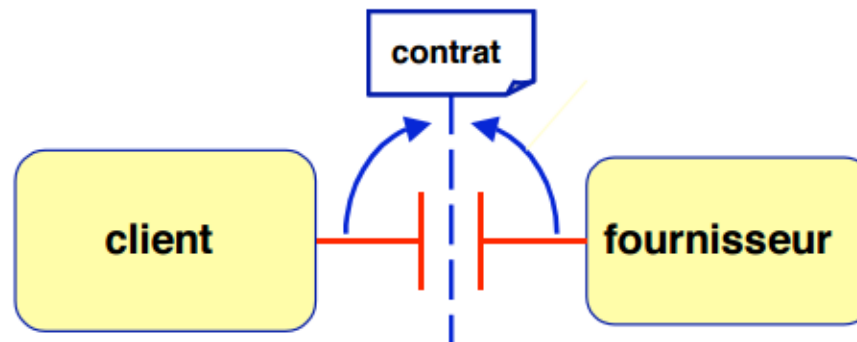
Définitions d'interfaces

La fourniture d'un service met en jeu deux interfaces

- Interface requise (côté client)
- Interface fournie (côté fournisseur)

Le contrat spécifie la compatibilité (conformité) entre ces interfaces

- Au delà de l'interface, chaque partie est une « boîte noire » pour l'autre (principe d'encapsulation)
- Conséquence : client ou fournisseur peuvent être remplacés du moment que le composant remplaçant respecte le contrat (est conforme)



Définitions d'interfaces

■ Partie « opérationnelle »

- ❑ Interface Definition Language (IDL)
 - Pas de standard, mais s'appuie sur un langage existant
 - ❑ IDL CORBA sur C++
 - ❑ Java et C# définissent leur propre IDL

■ Partie « contractuelle »: plusieurs niveaux de contrats

- ❑ Sur la forme : spécification de types -> conformité syntaxique
- ❑ Sur le comportement
- ❑ Sur les interactions entre méthodes
- ❑ Sur les aspects non fonctionnels (performances, etc.)

Exemples d'applications réparties

- Coordination d'activités :
 - Systèmes à flots de données (workflow).
 - Systèmes à agents.
- Communication et partage d'informations :
 - Bibliothèques virtuelles.
 - Collecticiels : pour le travail coopératif, les bibliothèques, musées et magasins virtuels sur Internet, la presse et le commerce électronique, etc. Quelques exemples de collecticiels :
 - Édition coopérative.
 - Téléconférence.
 - Ingénierie concourante.

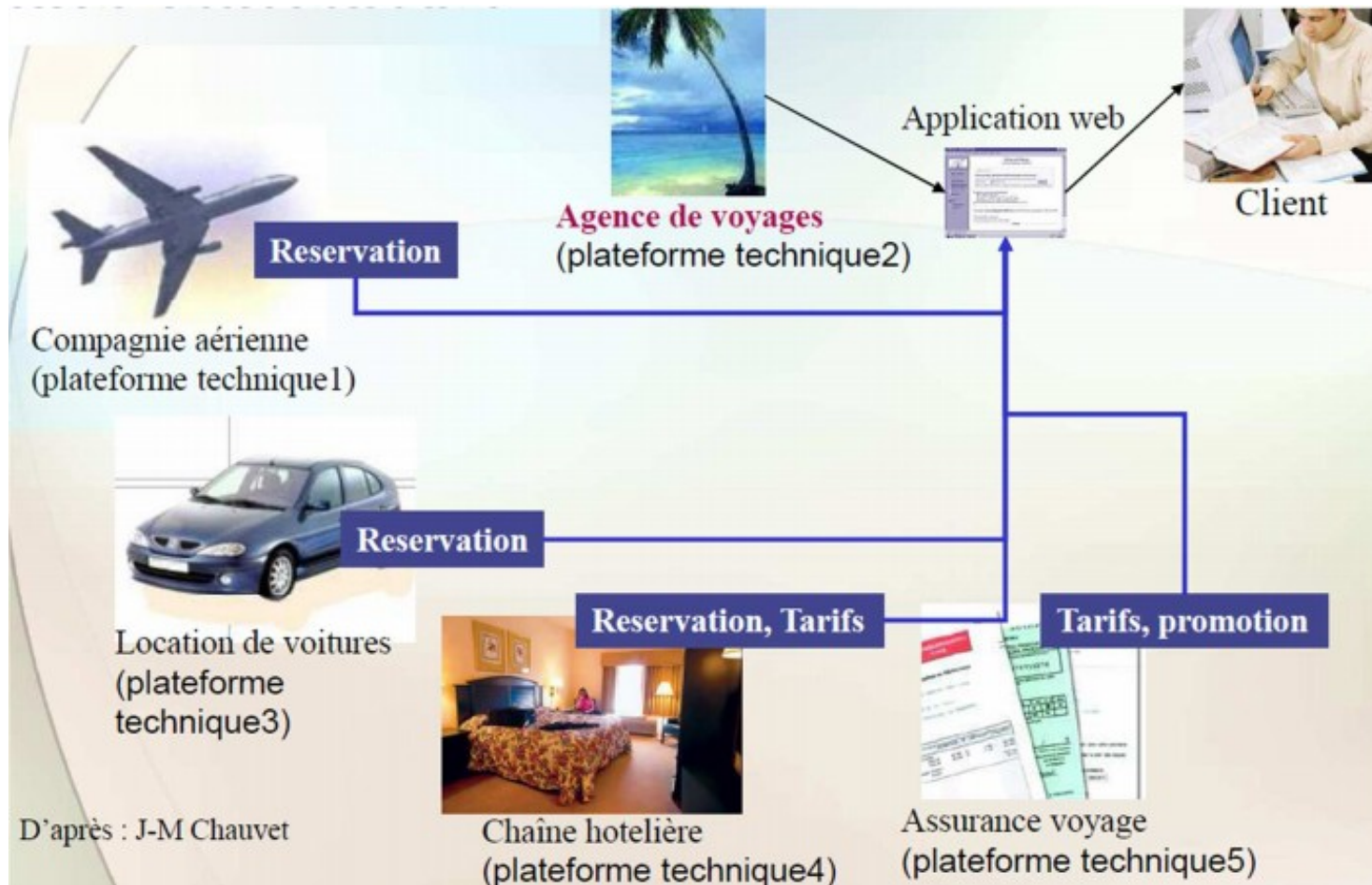
Exemples d'applications réparties

- Applications temps réel :
 - Contrôle de procédés industriels
 - Avionique
 - Localisation de mobiles
- Toutes les applications qui nécessitent des utilisateurs ou des données réparties.

Exemple: Agence de voyage

- Un produit « voyage » est la combinaison de plusieurs produits:
 - ❑ Gestion de réservation de billets de transport,
 - ❑ Gestion de réservation des hôtels
 - ❑ Gestion de réservation des voitures de location
 - ❑ ..
- Le résultat d'information récupérées auprès de différents fournisseurs:
 - ❑ Compagnies aériennes
 - ❑ Chaines hôtelières
 - ❑ Agence de location de voitures

Exemple: Agence de voyage



Phases de Construction d'applications réparties

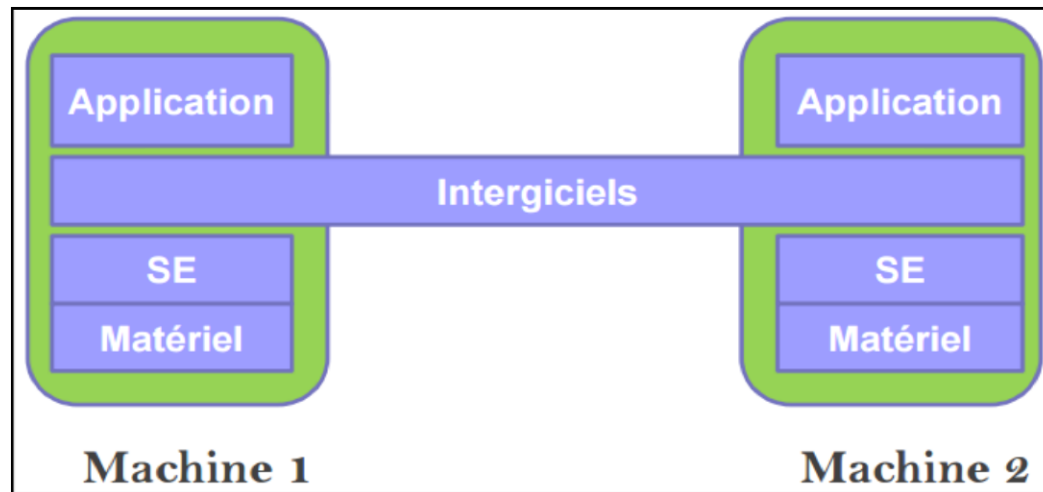
1. Conception de l'architecture de l'application.
2. Programmation des entités logicielles :
 - ❑ Utilisation d'un mécanisme de communication des composants répartis, appelés *Intergiciel* ou *middleware*, tels que Socket, RMI, RPC, CORBA, etc.
 - ❑ Programmation en fonction du modèle d'exécution adopté.

Phases de Construction d'applications réparties

3. Configuration des entités de diverses provenances :
 - ❑ leur permettre de communiquer, d'échanger des données.
 - ❑ leur permettre d'échanger des informations de contrôle.
 - ❑ leur permettre de se comprendre.
4. L'installation et le déploiement.
5. Administration :
 - ❑ Surveillance
 - ❑ Maintenance
 - ❑ Evolution des applications.

Intergiciel (Middleware)

- Couche logicielle intermédiaire située entre l'application et le système d'exploitation de la machine.



- L'intergiciel est nécessaire pour développer des applications réparties. Il assure la communication entre leurs différents composants d'une manière transparente.

Avantages du réparti

- Organisationnel
 - Décentraliser les responsabilités
 - Découpage en unité
- Fiabilité et disponibilité
 - Individualisation des défaillances
 - Duplication des constituants de l'application
- Performance
 - Partage de la charge
- Maintenance et évolution

Inconvénients du réparti

- Une mise en œuvre plus délicate
 - Gestion des erreurs
 - Suivi des exécutions
- Pas de vision globale instantanée
 - Délais des transmissions
- Administration plus lourde
 - Installation
 - Configuration
 - Surveillance
- Coût
 - Formation
 - Achat des environnements

Répartie vs Centralisée

Informatique Centralisée	Informatique Répartie
Un seul OS	Différents OS
Mémoire partagée	Mémoires individuelles
Pas de réseau	Pbs de communication: réseau, firewall, ..
Service local	Retrouver le service? qui le propose? Où se trouve t il?
Un seul langage de programmation	Différents langages: représentations différentes de l'information, passage de paramètres, différents paradigmes
Les users travaillent sur une seule machine	Les users ont l'impression de travailler sur une seule machine: la transparence