Communications dans les systèmes repartis et les middlewares

Table des matières

Objectifs

À la fin de cette leçon, l'apprenant sera capable de :

- décrire les communications dans les systèmes repartis ;
- identifier un middleware;

Les communications dans les systèmes repartis

I

1. Modèle de communication par messages

- Une application produit des messages (*producteur*) et une application les consomme (consommateur)
- Le producteur (l'émetteur) et le consommateur (*récepteur*) ne communiquent pas directement entre eux mais utilisent un objet de communication intermédiaire (boîte aux lettres ou file d'attente)
- Communication *asynchrone* :
    - Les deux composants n'ont pas besoin d'être connectés en même temps grâce au système de file d'attente
    - Émission non bloquante: l'entité émettrice émet son message, et continue son traitement sans attendre que le récepteur confirme l'arrivée du message.
    - Le récepteur récupère les messages quand il le souhaite.
- Adapté à un système réparti dont les éléments en interaction sont faiblement couplés :
    - éloignement géographique des entités communicantes
    - possibilité de déconnexion temporaire d'un élément
- Deux modèles de communication :
    - *Point à point* : Les messages ne sont lus que par un seul consommateur. Une fois qu'un message es lu, il est retiré de la file d'attente.
    - *Multi-points* ou Publish/Subscribe (par abonnement) :
    - les applications consommatrices des messages s'abonnent à un topic (sujet)
    - Les messages envoyés à ce topic restent dans la file d'attente jusqu'à ce que toutes les applications abonnées aient lu le message

2. Modèle de communication par événements

- Concepts de base :
    - Événement = changement d'état survenant de manière asynchrone (par rapport à ses “clients”)
    - Réaction = exécution d'une opération prédéfinie liée à l'événement • Mode de communication asynchrone et anonyme
    - indépendance entre l'émetteur (producteur) et les destinataires (consommateurs) d'un événement
- Deux modes de consommation des messages:
    - « Mode Pull » - réception explicite : les clients viennent prendre régulièrement leurs messages

- « Mode Push » - délivrance implicite : une méthode prédéfinie est attachée à chaque type de message et elle est appelée automatiquement à chaque occurrence de l'événement. la réception d'un événement entraîne l'exécution de la réaction associée.

3. Modèle à base de composants

- Composant : module logiciel autonome et réutilisable.
- Caractéristiques d'un composant :
  - des entrées/sorties déclarées pour permettre les connexions entre plusieurs composants
  - des propriétés déclarées permettant de configurer le composant
- Simplifier la conception et le développement des applications avec les composants:
  - Réutilisabilité
  - Indépendance
  - autonomie

4. Modèle à base d'agents mobiles

- Les agents mobiles : entités logicielles permettant de construire des applications distribuées
- Un agent mobile peut se déplacer d'une machine à une autre sur réseau.
- Un agent qui s'exécute sur une machine peut, à tout moment, arrêter son exécution, se déplacer sur une autre machine et reprendre son exécution (à partir de son dernier état)
- *Exemple :* Les applets = programme exécutable inclut dans une page HTML et qui s'exécute sur le site (machine) qui télécharge la page.
- *Avantage de la mobilité :*
  - efficacité, privilégie les interactions locales ;
  - moins de communications distantes effectuées pour les échanges de messages;
  - amener le code aux données plutôt que le contraire.

5. Modèles à mémoire partagée

- *Objectif :* Replacer le programmeur dans les conditions d'un système centralisé, utiliser un espace mémoire commun pour les communications
- *Avantage:*
  - transparence de la distribution pour le développeur,
  - efficacité de développement (utilisation des paradigmes usuels de la programmation concurrente).
- *Inconvénient*: Complexité de mise en œuvre efficace d'une mémoire partagée distribuée

6. Modèle orienté services

- Un modèle d'interaction basé sur la notion de service
- Un service est un composant logiciel exécuté par un producteur à l'attention d'un consommateur

- Une nouvelle vision dans la conception des applications réparties

Exercice : Activité d'auto-évaluation

II

Exercice

Les communications dans les systèmes repartis sont (4 réponses) :

- ☐ Modèle de communication par événements
- ☐ Modèle orienté système
- ☐ Modèle à base d'application
- ☐ modèle à base de mémoire réinscriptible
- ☐ Modèle de communication par messages
- ☐ Modèle de communication par MMS
- ☐ Modèle orienté services
- ☐ Modèle à base d'argents mobiles

Exercice

Dans le modèle de communication par message (3 réponses) :

- ☐ Le producteur et le consommateur communiquent directement par le biais d'une boîte aux lettres
- ☐ nous avons communication asynchrone
- ☐ Un récepteur produit des messages et un émetteur les consomme
- ☐ Une application produit des messages et une application les consomme
- ☐ nous avons communication asynchrone
- ☐

L'émetteur et le récepteur ne communiquent pas directement entre eux mais utilisent une file d'attente

Exercice

Le modèle de communication par message est :

adapté à un système réparti dont les éléments en interaction sont fortement

- O couplés

adapté à un système centralisé dont les éléments en interaction sont faiblement

- O couplés
- O

adapté à un système reparti d'un réseau en étoile dont les éléments en interaction sont faiblement couplés

adapté à un système distribué dont les éléments en interaction sont faiblement

- O couplés

Exercice

Dans une communication asynchrone (3 réponses) :

- ☐

L'entité émettrice émet son message, et continue son traitement sans attendre que le récepteur confirme l'arrivée du message

- ☐ Les deux applications dialoguant doivent être connectées simultanément
- ☐ L'émetteur récupère les messages quand il le souhaite
- ☐ Les deux applications dialoguant n'ont pas besoin d'être connectés en même temps
- ☐

L'entité émettrice émet son message, et attend que le récepteur confirme l'arrivée du message avant d'émettre le prochain message

- ☐ Le récepteur récupère les messages quand il le souhaite

Exercice

Dans un modèle de communication par événement, nous avons deux modes de communications par message que sont : (2 réponses)

- ☐ Mode asynchrone
- ☐ Mode Push
- ☐ Mode Synchrone
- ☐ Mode Pull
- ☐ Mode ajournée
- ☐ Mode différée
- ☐ Mode POP

- ☐ Mode IMAP

Les middlewares

III

- Un *middleware* ou « *intergiciel* » ou « élément du milieu » est l'ensemble des couches réseau et services logiciel qui permettent le dialogue entre les différents composants d'une application répartie.
- Gartner Group définit le middleware comme une interface de communication universelle entre processu

1. Motivations

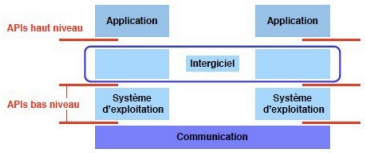
Dans un système réparti, l'interface fournie par les systèmes d'exploitation et de communication est encore trop complexe pour être utilisée directement par les applications :

- Hétérogénéité (matérielle et logicielle)
- Complexité des mécanismes (bas niveau)
- Nécessité de gérer la répartition

Solution

Introduire une couche logicielle intermédiaire (répartie) entre les niveaux bas (systèmes et communication) et le niveau haut : c'est l'*intergiciel* (*Middleware* en anglais) .

- L'intergiciel joue un rôle analogue à celui d'un “super-système d'exploitation” pour un système réparti
- Représente l'élément le plus important de tout système réparti.
- Positionnement du middleware (couche du milieu)



Fonctions

- Masquer l'hétérogénéité (des machines, systèmes, protocoles de communication)
- Fournir une API (Application Programming Interface) de haut niveau
    - Permet de masquer la complexité des échanges
    - Facilite le développement d'une application répartie
- Rendre la répartition aussi invisible (transparente) que possible

- Fournir des services répartis d'usage courant

2. Services du middleware

- Conversion
    - permet la communication entre machines mettant en œuvre des formats différents de données
    - prise en charge par la FAP (Format And Protocol)
- Nommage
    - permet d'identifier la machine serveur sur laquelle est localisé le service demandé afin d'en déduire le chemin d'accès.
    - fait, souvent, appel aux services d'un annuaire.
- Sécurité : permet de garantir la confidentialité et la sécurité des données à l'aide de mécanismes d'authentification et de cryptage des informations
- Communication : permet la transmission des données entre les deux

3. Les Types de middlewares

- Middleware *RPC* (Remote Procedure Call) : RPC de SUN
- Middlewares *orientés objets distribués* : Java RMI, Corba
- Middlewares *orientés composants distribués* : EJB, Corba, DCOM
- Middlewares *orientés messages* : JMS de Sun, MSMQ de Microsoft, MQSeries de IBM
- Middlewares *orientés sevices* : Web Services (XML-RPC, SOAP)
- Middlewares *orientés SGBD* : ODBC (Open DataBase Connectivity), JDBC de Sun
- Middlewares *transactionnels* : JTS de Sun, MTS de Microsoft

Exercice : Activité d'auto-évaluation

IV

Exercice

Les services d'un middleware sont (4 réponses) :

- ☐ Sécurité
- ☐ Scheduling
- ☐ Conversion
- ☐ Adressage
- ☐ Communication
- ☐ Codage
- ☐ Nommage
- ☐ Homothétie

Exercice

Quatre types de middlewares sont :

- ☐ middlewares orientés systèmes
- ☐ middlewares orientés SGBD
- ☐ middlewares à base de logiciel
- ☐ middlewares orientés objets distribués
- ☐ middlewares orientés applications
- ☐ middlewares orientés areas
- ☐ middlewares transactionnel
- ☐ middlewares RPC

Exercice

Les fonctions d'un middleware (3 réponses) :

- ☐ Rendre la répartition aussi transparente que possible
- ☐ Interconnecter plusieurs processeurs
- ☐ Masquer les données applicatives
- ☐ Masquer l'hétérogénéité
- ☐ Rendre dynamique une application
- ☐ Fournir une API de haut niveau
- ☐ Rendre la répartition des application non transparente
- ☐ Fournir une API de bas niveau