

UE NSY103

L'Informatique Répartie



Sommaire

I- Introduction

II- Les terminaux Virtuels

III- Les Serveurs Web

IV- Les Applications Réparties



I- Introduction

Le concept d'Informatique Répartie ou Informatique Distribuée s'oppose au concept d'Informatique Centralisée.

Le Mode Terminal

- Chronologiquement, l'informatique s'est d'abord organisée de façon centralisée. Les premiers ordinateurs étaient volumineux et très coûteux, les seules fonctions réparties étaient l'affichage et la saisie sur un poste de travail non-intelligent, c'est-à-dire ne comportant aucun traitement local - même pas de présentation graphique - , c'est le mode terminal.

I- Introduction

Le Mode Client-Serveur

Avec la baisse continue du coût des machines puis l'affirmation du micro-ordinateur comme "poste de travail intelligent" dans les années 80, les architectures de réseau informatique se sont de plus en plus orientées vers une répartition distribuée ou en réseau des ressources informatique.

- L'informatique répartie correspond donc à une poste de travail dit "Client" qui est un micro-ordinateur disposant de logiciels locaux (en général les outils de bureautique) et pouvant accéder à des logiciels répartis, hébergés par des ordinateurs dits "Serveurs" : C'est le mode "Client-Serveur".
- Le mode de fonctionnement est du type "Requête-réponse" : Le client envoie une requête au serveur, le serveur traite la requête et envoie la réponse au client.
- Dans le modèle Client-Serveur l'IHM est gérée par le poste client, il reste à organiser la répartition des traitements et des données entre le poste client et le poste serveur. Selon l'importance du traitement local on parlera de "Client léger" ou de "Client lourd".
- Le mot "serveur" a été lancé par France Télécom pour ses éditeurs de contenu "Minitel"

I- Introduction

Le Mode Client-Serveur (suite)

Les évolutions vers le modèle client-serveur se traduisent par un renforcement du découpage des applications en plusieurs niveaux ou tiers

(faux ami : Tiers signifie ,niveau en anglais), entre les fonctions de base suivantes :

- 1- L'interface homme-machine "IHM"
- 2- Le moniteur transactionnel
- 3- Le logiciel applicatif
- 4- Le "Système de Gestion de Base de Données"
- ... autres



I- Introduction

Répartition des traitements

- **Le terminal virtuel** : Seules les fonctions basiques de l'IHM (graphisme low-level) sont déportées sur le poste client,
- **Les clients légers** : Seule l'IHM est déportée sur le poste client,
- **Les clients lourds** : L'IHM ainsi qu'une partie des traitements et des données sont déportées sur le poste client.

Synchronisation des traitements

- **Le mode synchrone** : Après envoi d'une requête le logiciel client est bloqué en attente de la réponse.
- **Le mode asynchrone** : Possibilités de requêtes-réponses multiples avec gestion de files d'attentes (de requêtes et de réponses) de part et d'autre.

I- Introduction

Avantages du mode Client-Serveur

- **L'ergonomie du poste de travail :** Gestion locale de l'IHM en mode graphique.
- **La répartition des traitements :** Ce qui évite l'effondrement des serveurs centraux.
- **L'extensibilité (Scalability en anglais) :**
 - En augmentant le nombre de serveurs, ce qui est plus facile à mettre en œuvre que le remplacement d'un ordinateur central.
- La possibilité de continuer de travailler en mode dégradé en local en cas d'indisponibilité des serveurs centraux.



I- Introduction

Inconvénients du mode Client-Serveur

- **Le premier inconvénient du mode client-serveur c'est la logistique du déploiement :**
 - Il faut non seulement mettre à jour le serveur mais il faut également mettre à jour tous les postes clients.
 - Ce qui complique le problème c'est que le plus souvent le parc de postes clients est hétérogène.
 - Le déploiement correspond à un véritable surcoût, même si une partie de ces installations peuvent s'effectuer en réseau. Dans le cas du mode terminal le déploiement se limite à la mise à jour de l'ordinateur central.
- **Le deuxième inconvénient c'est l'aspect Sécurité :**
 - Il est plus facile de sécuriser une seule machine centrale qu'un réseau de serveurs et de postes de travail.

II- Les terminaux Virtuels

Le Terminal virtuel est un mécanisme émulant le terminal local d'un ordinateur sur un poste de travail distant.

- Il s'agit d'un client ultra-léger, seule les primitives graphiques de base sont émulées sur le poste client, les traitements et l'IHM sont traités par le serveur.
- Il s'agit d'un client universel, tout logiciel local (y compris le système d'exploitation lui-même) peut être accessible à distance et cela sans nécessiter d'adaptation de ce logiciel.
- Le terminal virtuel possède les avantages du mode terminal en matière de déploiement et de sécurité.
- Le terminal virtuel peut-être utilisé sur des postes de travail sans disques durs.

Exemples du marché :

- Telnet (mode texte).
- Terminal X Linux (mode graphique).
- VNC (Virtual Network Computing, mode graphique open source).

III- Les Serveurs Web

La solution serveur Web est fondée sur un serveur HTTP (Hyper Text Transport Protocol) et un logiciel client particulier appelé "Navigateur".

- Le navigateur est un client léger, seule l'IHM est traitée par le poste client.
- Le navigateur est un logiciel client universel pour toutes les applications développées selon le standard "serveur HTTP".
- Le Serveur Web possède les avantages du mode terminal en matière de déploiement.
- Il est également possible de répartir la charge de travail, des langages de programmation tels que le "Javascript" et des concepts tels que Ajax (Asynchronous JavaScript and XML) permettent aux serveurs de déporter automatiquement une partie de leurs traitements sur les postes clients.

IV- Les Applications Réparties

Appel de Procédures Distantes

Il s'agit de fournir aux applications une boîte à outils pour utiliser des procédures distantes de façon analogue à l'utilisation des procédures locales.

La solution la plus répandue, baptisée RPC (Remote Procedure Call) possède les caractéristiques suivantes :

- Mode non connecté : Un échange requête-réponse ne nécessite pas l'établissement d'une connexion préalable.
- Mode synchrone : Requête bloquante tant que le sous-programme distant n'a pas retourné sa réponse. Pour ne pas rester bloqué, il faut créer un processus (ou un thread) par requête.
- Transparence par rapport aux couches réseau. Le programmeur n'a pas à gérer de reprises au niveau des échanges réseau et l'application est portable dans différents contextes réseaux.



IV- Les Applications Réparties

Appel de Procédures Distantes (suite)

Les appels de procédures distantes sont généralement utilisés dans le cas d'applications avec traitements centralisés et IHM répartie.

- Cette technique tend aujourd'hui à être remplacée par les "invocations de méthodes distantes" étant donné le passage à la programmation orientée objet en matière d'informatique répartie. (cf. chapitre suivant)

exemples :

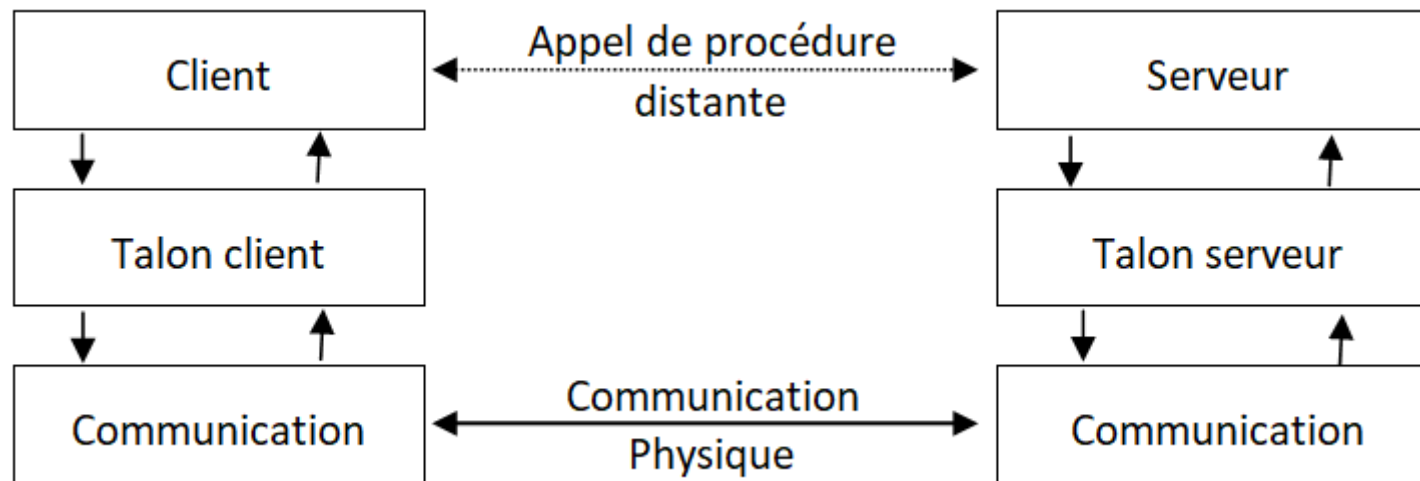
- RPC de Sun : Elaborée dans le cadre du protocole NFS (Network File System)
- RPC des spécifications DCE (Distributed Computing Environment)
- IBM de son côté implémente sa solution spécifique Open CPI-C (Common Programming Interface for Communication), qui fonctionne au-dessus de APPC (Advanced Program to Program Communication).

IV- Les Applications Réparties

Architecture RPC

Il s'agit d'un modèle à 3 couches, L'application dialogue avec une couche dite "Talon" laquelle est le représentant de la partie distante.

- Les Talons utilisent la couche Communication pour dialoguer entre eux.
- Le Talon Client est aussi appelé Stub et le Talon Serveur Skeleton



IV- Les Applications Réparties

Invocation de Méthode Distante

C'est l'informatique répartie qui a provoqué l'explosion de la programmation orientée objet.

Le modèle objet est aujourd'hui considéré comme étant le mieux adapté à l'architecture des systèmes répartis :

- Les données sensibles d'un objet ne peuvent être accédées que par ses propres méthodes.
 - Les sémaphores (concurrence d'accès aux méthodes) sont facilement implémentables grâce au cloisonnement fort entre les objets ("synchronized" en Java).
 - Les classes sont facilement maintenables en réseau, il est bien plus aisé de mettre à jour une seule classe qu'une application entière.
- Il s'agit de créer des objets instanciables à distance d'une façon analogue à l'instanciation locale.



IV- Les Applications Réparties

Invocation de Méthode Distante (suite)

Les solutions actuelles RMI (Remote Method Invocation) et Corba (Common Object Request Broker Architecture) possèdent des caractéristiques analogues à celles des RPC :

- Mode non connecté : Un échange requête-réponse ne nécessite pas l'établissement d'une connexion préalable.
- Mode synchrone : Requête bloquante tant que la méthode distante n'a pas retourné sa réponse.
- Transparence par rapport aux couches réseau. Le programmeur n'a pas à gérer de reprises au niveau des échanges réseau.

➤ Les invocations de méthodes distantes sont généralement utilisées dans le cas d'applications orientées objet avec traitements centralisés et IHM répartie.

exemples :

- RMI : Solution Sun liée à Java, suit le standard Corba aujourd'hui.
- Corba : Solution universelle préconisée par l'OMG (Object Management Group)
- Les Web Services (Cf. chapitre correspondant) sont aussi à classer dans cette catégorie.

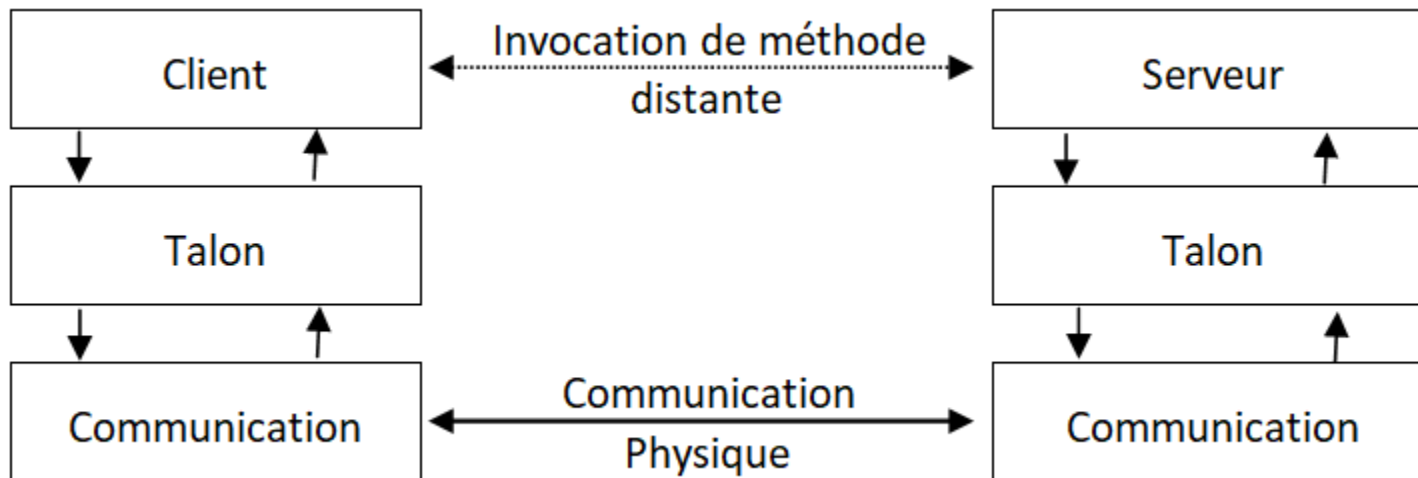


IV- Les Applications Réparties

Architecture RMI ou Corba

Comme dans le cas des appels de procédures distantes, Il s'agit d'un modèle à 3 couches :

- L'application dialogue avec une couche appelée Talon laquelle est le représentant de la partie distante.
- Les Talons utilisent la couche Communication pour dialoguer entre eux.
- Le Talon Client est aussi appelé Stub et le Talon Serveur Skeleton.



IV- Les Applications Réparties

Les Applications Orientées Messages

Il s'agit là de gérer le mode client-serveur asynchrone : Le client peut envoyer plusieurs requêtes simultanément.

- Le serveur va traiter les requêtes (des différents clients) dans l'ordre chronologique, par priorités ou selon l'importance des travaux à effectuer.

On parlera alors de gestion de gestion de files d'attente de requêtes-messages, cela implique :

- la gestion d'identifiants associant message-requête et message-réponse.
 - la gestion de files d'attentes de messages.
- Ce mode de communication a été baptisé MOM (Message Oriented Middleware), son fonctionnement dit "point à point" est différent de celui d'une communication directe d'application à application car premièrement le producteur ne spécifie pas une application cible mais une file d'attente et deuxièmement plusieurs applications peuvent lire ou écrire des messages dans une même file d'attente donnée.



IV- Les Applications Réparties

Les Applications Orientées Messages (suite)

Les "MOM" se prêtent parfaitement à l'intégration des application hétérogènes en entreprise grâce à la souplesse du concept de message intermédiaire en point à point.

- Cela explique la présence du mot Middleware (Intergiciel) dans le sigle MOM.
- Les serveurs orientés message sont généralement utilisés dans les cas d'applications nécessitant la gestion asynchrone des requêtes.
- Les serveurs orientés message sont les piliers de l'Intégration des Applications d'Entreprise (Middleware).

exemples :

- - JMS de Sun
- - MSMQ de Microsoft
- - IBM Websphère (MQ Series), basé sur les technologies Sun J2EE et JMS.
- - Le courrier électronique Internet (protocoles SMTP et POP).

EXERCICES

Vous pensez avoir bien assimilé les concepts présentés dans ce cours.
Vous devez alors passer aux exercices, ce sont eux qui
vous permettront de valider vos connaissances .