Modèle client-serveur

Architecture informatique distribuée dans laquelle les tâches et les charges de travail sont partagées entre :

- les <u>fournisseurs</u> d'une ressource ou d'un service, appelés <u>serveurs</u>,
- les demandeurs de services, appelés clients.

Les tâches sont exécutées par des <u>programmes qui sont répartis</u> entre les clients et les serveurs.

Souvent, les clients et les serveurs tournent sur des machines séparées et communiquent via un réseau informatique. Mais le client et le serveur peuvent résider sur le même système.

Une <u>machine hôte</u> (anglais : host) peut exécuter un ou plusieurs programmes serveurs qui partagent leurs ressources avec des clients.

Définitions

<u>Serveur</u> (anglais : server) : Programme (processus) qui peut se connecter à un programme client pour lui donner accès à un service.

Remarque : "serveur" est un rôle, pas un ordinateur en soi.

Un serveur et un client peuvent être hébergés par la même machine hôte.

<u>Client</u>: Programme qui peut demander un service à un serveur qui fournit ce service.

<u>Service</u>: Fourniture d'une fonctionnalité spécifique offert par un serveur et qu'un client peut utiliser.

Requête (anglais : request) : Demande de la part d'un client, d'un service dont il a besoin à un serveur.

<u>Réponse</u> (anglais : response) : Réponse d'un serveur à une demande d'un client.

Exemples

Serveur d'applications

- fournit des applications déjà installées.
- un client peut démarrer le programme souhaité qui n'est pas sur le disque dur local mais sur celui du serveur.

Avantages

- mise à jour unique des programmes.
- mises à jour chargées sur le serveur, qui les redistribuent aux clients (souvent, une petite installation locale est nécessaire).

Remarque : Lorsque les programmes sont exécutés directement sur le serveur, celui-ci est appelé un <u>serveur de terminaux</u>.

Serveur de messagerie électronique

envoie des mails à des clients de messagerie.

Serveur de fichiers

• permet de stocker et consulter des fichiers sur le réseau.

Serveur de données

exécute un système de gestion de base de données (SGBD).

Avantages

- de grandes quantités de données, stockées en central, sont mises à disposition.
- plusieurs clients peuvent travailler simultanément avec ces données.

Serveur web

renvoie des pages web demandées par des navigateurs web.

Fonctionnement client-serveur

Processus serveur

- attend une connexion entrante sur un ou plusieurs ports réseaux ;
- à la connexion d'un client sur le port en écoute, il ouvre un socket local au système d'exploitation;
- suite à la connexion, il communique avec le client suivant le protocole prévu par la couche application du modèle OSI;
- Un serveur est généralement capable de servir plusieurs clients simultanément.

Remarque:

On utilise également le terme serveur pour désigner un <u>ordinateur</u> <u>spécialisé</u> ou une <u>machine virtuelle</u> ayant pour unique tâche l'exécution d'un ou plusieurs processus serveur.

Processus client

- établit la connexion au serveur à destination d'un ou plusieurs ports réseaux;
- lorsque la connexion est acceptée par le serveur, il communique comme le prévoit la couche applicative du modèle OSI.

Le client et le serveur doivent utiliser le même protocole de communication au niveau de la couche transport du modèle OSI.

Types de clients

Client léger

- Le poste client accède à une application située sur un ordinateur dit « serveur » via une interface et un navigateur Web.
- L'application fonctionne <u>entièrement sur le serveur</u>, le poste client reçoit la <u>réponse « toute faite »</u> à la demande (requête) qu'il a formulée.

Client lourd

- Le poste client doit comporter un <u>système d'exploitation</u> qui exécute en local une partie des traitements.
- Le traitement de la réponse à la requête du client utilisateur va mettre en œuvre un <u>travail combiné</u> entre l'ordinateur serveur et le poste client.

Client riche

- Une interface graphique plus évoluée permet de mettre en œuvre des fonctionnalités comparables à celles d'un client "lourd".
- Les traitements sont effectués majoritairement sur le serveur.
- la réponse "semi-finie" étant envoyée au poste client, où le client "riche" est capable de la finaliser et de la présenter.

Exemple : Création d'un compte sur un site sécurisé

- champ pseudo : il faut vérifier que ce pseudo n'est pas utilisé
- champ mot de passe : il doit comporter au moins une minuscule, une majuscule, un chiffre et une ponctuation
- champ de confirmation du mot de passe : valeur identique au précédent

M3104 © Denis FINCK 2014-2016

Avantages de l'architecture client-serveur

Données

 sont centralisées sur un serveur, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.

Complexité du traitement et puissance de calcul

- sont à la charge du ou des serveurs.
- client léger sur un ordinateur terminal qui peut être simplifié au maximum.

Recherche d'information

 les serveurs étant centralisés, cette architecture est particulièrement adaptée et rapide pour retrouver et comparer de vastes quantités d'information (moteur de recherche sur le Web).

Inconvénients de l'architecture client-serveur

- Si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge.
- Si le serveur n'est plus disponible, plus aucun des clients ne fonctionne.
- Les coûts de mise en place et de maintenance peuvent être élevés.
- Les clients ne peuvent communiquer entre eux.

Dans la suite nous n'étudierons que les serveurs Web.