



# 1) Notion de client-serveur

Deux ordinateurs en réseau peuvent s'échanger des données. Dans la plupart des cas ces échanges ne sont pas "symétriques" : en effet un ordinateur A va souvent se contenter de demander des ressources (fichiers contenant du texte, photos, vidéos, sons...) à un ordinateur B. L'ordinateur B va lui se contenter de fournir des ressources à tous les ordinateurs qui lui en feront la demande. On dira alors que l'ordinateur A (celui qui demande des ressources) est un client alors que l'ordinateur B (celui qui fournit les ressources) sera qualifié de serveur.

En tapant «<http://www.google.fr>», votre machine va chercher à entrer en communication avec le serveur portant le nom «[www.google.fr](http://www.google.fr)» (en fait c'est plus compliqué, pour les puristes nous diront donc que la communication va être établie avec le serveur www du domaine [google.fr](http://google.fr), mais bon, pour la suite nous pourrions nous contenter de l'explication « simplifiée »).

Une fois la liaison établie, le client et le serveur vont échanger des informations en dialoguant :

client : bonjour [www.google.fr](http://www.google.fr) (ou bonjour www se trouvant dans le domaine [google.fr](http://google.fr)), pourrais-tu m'envoyer le fichier index.html

serveur : OK client, voici le fichier index.html

client : je constate que des images, du code css sont utilisés, peux-tu me les envoyer

serveur : OK, les voici

Évidemment ce dialogue est très imagé, mais il porte tout de même une part de « vérité ».

Sur internet, ce modèle client/serveur domine assez largement, même s'il existe des cas où un ordinateur pourra jouer tour à tour le rôle de client et le rôle de serveur, très souvent, des ordinateurs (les clients) passeront leur temps à demander des ressources à d'autres ordinateurs (les serveurs) . Par exemple, comme expliqué dans l'exemple ci-dessus on retrouve cet échange client/serveur à chaque fois que l'on visite une page web. Il y a de fortes chances pour que votre ordinateur personnel joue quasi exclusivement le rôle de client (sauf si vous êtes un adepte du "peer to peer").

N'importe quel type d'ordinateur peut jouer le rôle de serveur, mais dans le monde professionnel les serveurs sont des machines spécialisées conçues pour fonctionner 24h sur

24h. Ils peuvent aussi avoir une grosse capacité de stockage afin de stocker un grand nombre de ressources (vidéos, sons,...).



Afin d'assurer une continuité de service, dans les sociétés, plusieurs serveurs assurent exactement le même rôle (on parle de redondance). Vous vous doutez bien que Google ne possède pas qu'un seul serveur, en effet, en moyenne, chaque seconde, c'est environ 65000 clients qui se connectent aux serveurs du moteur de recherche de Google. Aucun serveur, même extrêmement performant, ne serait capable de répondre à toutes ces requêtes. Google, Amazon ou encore Facebook possèdent un très grand nombre de serveurs afin de pouvoir satisfaire les demandes des utilisateurs en permanence. Ces entreprises possèdent d'immenses salles contenant chacune des centaines ou des milliers de serveurs (ces serveurs sont rangés dans des armoires appelées "baie serveur").



Souvent les serveurs sont spécialisés dans certaines tâches, par exemple, les serveurs qui envoient aux clients des pages au format HTML sont appelés "serveur web".

## 2) du Web statique au Web dynamique

Il y a quelques années, le web était dit « statique » : le concepteur de site web écrivait son code HTML et ce code était simplement envoyé par le serveur web au client. Les personnes

qui consultaient le site avaient toutes le droit à la même page, le web était purement « consultatif ».

Les choses ont ensuite évolué : les serveurs sont aujourd'hui capables de générer eux-mêmes du code HTML. Les résultats qui s'afficheront à l'écran dépendront donc des demandes effectuées par l'utilisateur du site : le web est devenu dynamique.

Différents langages de programmation peuvent être utilisés « côté serveur » afin de permettre au serveur de générer lui-même le code HTML à envoyer. Le plus utilisé encore aujourd'hui se nomme PHP. D'autres langages sont utilisables côté serveur (pour permettre la génération dynamique de code HTML) : Java, Python...

Voici un exemple très simple de code en PHP :

```
<?php
$heure = date("H:i");
echo '<h1>Bienvenue sur mon site</h1>
      <p>Il est '.$heure.'</p>';
?>
```

Sans entrer dans les détails, si un client se connecte à un serveur web qui exécute ce code à 18h23, le serveur enverra au client le code HTML ci-dessous :

```
<h1>Bienvenue sur mon site</h1>
<p>Il est 18h23</p>
```

En revanche si un client se connecte à ce même serveur à 9h12, le serveur enverra au client le code HTML ci-dessous :

```
<h1>Bienvenue sur mon site</h1>
<p>Il est 9h12</p>
```



## activité 25.1

Expliquez en quelques lignes la notion de client et la notion de serveur.



## Ce qu'il faut savoir

- Dans la plupart des échanges au niveau d'un réseau (Internet par exemple), les ordinateurs qui communiquent ne jouent pas le même rôle. Un des ordinateurs est un client tandis que l'autre est un serveur. Un client va demander des ressources à un serveur alors qu'un serveur va passer son temps à fournir des ressources à une multitude de clients.
- N'importe quel type d'ordinateur peut jouer le rôle de serveur, mais dans le monde professionnel les serveurs sont des machines spécialisées conçues pour fonctionner 24h sur 24h (il faut être capable de fournir des ressources à un client 24h/24h)
- Afin de prévenir un "embouteillage" (trop de clients qui se connecteraient en même temps), un même service est souvent assuré par plusieurs serveurs qui vont jouer exactement le même rôle, on parle alors de redondance des serveurs.
- La redondance permet aussi d'assurer la continuité du service en cas de panne d'un des serveurs.