

NSI première - IHM sur le web

Modèle Client-Server

Modèle Client-Server

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» (C'est plus compliqué. Pour les puristes nous dirons donc que la communication va être établie avec le serveur [www](http://www.google.fr) du domaine [google.fr](http://www.google.fr). 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 (ou bonjour [www](http://www.google.fr) se trouvant dans le domaine [google.fr](http://www.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...). Une seule machine peut servir de nombreuses applications.

Un serveur

Afin 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, Facebook ou Netflix 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").

salle 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".



Figure 1: serveur



Figure 2: salle-serveur

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 ».

Le site que vous consultez en ce moment est *statique*.

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. Les plus utilisés aujourd'hui pour générer des pages web côté serveur sont PHP, Java et Python. On trouve aussi Javascript avec node.js

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>
```

Comme vous pouvez le constater, le PHP permet de générer des pages HTML dynamiquement. Inutile de préciser que cet exemple est volontairement très simple, le PHP est capable de générer des pages HTML bien plus complexes. En particulier l'exemple ci-dessus est parfaitement réalisable en Javascript *côté client*.

Ce que le Javascript côté client ne permet pas de faire est *d'écrire des informations sur le serveur*. Ainsi, pour créer des comptes utilisateurs il est indispensable d'avoir des instructions côté serveur.

Nous allons apprendre à générer dynamiquement des pages web côté serveur en utilisant Python très prochainement.