

# Exploitation de données sur Internet/Intranet

Jacques THOORENS



# Chapitre 1

## Clients et serveurs

D'un point de vue architectural, l'histoire de l'informatique peut se voir de la manière suivante :

- les gros *mainframes* sont hébergés dans des locaux spécifiques (1945-60)
- on attache aux ordinateurs des terminaux plus ou moins intelligents, qui permettent à des utilisateurs d'employer l'ordinateur à distance en se partageant les ressources (*time sharing*) (1960-75)
- les micro-ordinateurs permettent à la puissance informatique de descendre sur le bureau (1975-90)
- de grands réseaux mondiaux se constituent pour échanger des données (ancêtres d'Internet), notamment entre les universités (1970-90)
- on relie les ordinateurs personnels entre eux ou à des serveurs (1990-95)
- explosion d'Internet et généralisation des réseaux locaux (1995-...)

### 1.1 Concepts

**le serveur** : mot valise qui désigne tout et n'importe quoi. Un serveur est une ressource informatique qui fournit des services. Selon le cas, cela peut-être une machine quelconque qui possède et partage un périphérique particulier (imprimante, scanner...), un ordinateur renforcé (au niveau matériel ou logiciel) qui joue un rôle majeur dans un réseau (contrôleur de domaine Windows, serveur intranet, serveur de bases de données). Microsoft a tendance à identifier la notion de serveur à un ordinateur possédant une version spéciale de Windows (plus chère).

**le client** : également fort ambigu. Cela peut être un ordinateur qui consomme ou utilise les ressources fournies par le serveur.

**client-serveur** : couple fréquent en architecture. Le serveur effectue l'essentiel de la tâche, le client se contente de gérer l'encodage et la mise en page de présentation. Exemple emblématique : un SGBD qui va gérer les tables d'une base de données et un client qui enverra des requêtes SQL avant de se charger d'afficher les résultats. Oracle et SQL Developer sont basés sur ce principe.

**modèle à trois couches** : inspiré du modèle MVC en programmation. Le serveur garde son rôle de gestionnaire des informations métier. Le client devient léger (par exemple un navigateur) et se borne à afficher les données qu'on lui envoie. Entre les deux, un serveur d'applications qui gère les données du serveur métier et envoie des écrans à afficher. Exemple : Oracle - ISQL (géré par un programme en Java qui fait office de serveur Web) - navigateur.

**Internet** : un réseau mondial reliant d'abord des réseaux militaires et académiques. La hantise de la guerre nucléaire a fait craindre la destruction des nœuds d'intersection. Le routage IP permet à un « paquet » de données de parvenir à destination sans que l'émetteur doit spécifier le chemin. Ce qu'on appelle aujourd'hui Internet est un réseau de réseaux fournissant principalement l'accès à des sites respectant le protocole HTTP et servant des pages HTML. Ce langage a permis de généraliser l'utilisation des liens entre les sites et prône la convivialité par l'emploi de ressources multimédia (principalement des images). Les moteurs de recherches ont également joué un rôle moteur en permettant de trouver l'adresse d'un site.

**Intranet** : la mise en application des technologies d'Internet au niveau d'un réseau local d'entreprise pour une diffusion interne de l'information. Du point de vue du concepteur et du programmeur, la seule différence réside dans la moindre importance de la sécurité (encore que...).

### 1.1.1 Critiques

Le client du modèle client-serveur se charge d'une grande partie des traitements. Il contient toute une logique de programmation. Comme il a été distribué, il est impossible de le modifier autrement que par distribution de nouvelles versions (lourd et dangereux si des versions anciennes subsistent). En outre, chaque plate-forme nécessite la création d'un client spécifique. Par contre, il y a peu de trafic sur le réseau.

Le modèle à trois couches alourdit le trafic du réseau (chaque page et chaque modification doit passer du serveur d'application au client). Il permet par contre une grande souplesse dans l'adaptation des applications et, en principe, la plate-forme du client final n'entre pas en ligne de compte.

### 1.1.2 Nouvelles tendances

#### les clients lourds

On utilise le navigateur (Firefox et son environnement XUL) ou le système d'exploitation (Windows Vista et XAML) pour faire tourner une application envoyée par le serveur. Le trafic réseau tend à diminuer. Inconvénient : l'application est intimement liée à son hôte (spécialement pour XAML). Java utilisé dans le navigateur s'apparente à cette technologie, bien qu'elle soit plus ancienne et plutôt en déclin.

#### Ajax

Cette technologie nouvelle allie les avantages du modèle à trois couches et des clients lourds, en remplaçant la station d'accueil par un framework écrit en Javascript. Ce framework permet de ne plus considérer la page comme un tout à réenvoyer depuis le serveur à chaque modification. Les scripts permettent une gestion intelligente des encodages de l'utilisateur. Si une action de ce dernier nécessite une réaction du serveur, cette réaction se cantonne à une partie de la page qui est réexpédiée partiellement (généralement sous la forme d'une portion de XML).

Un inconvénient : il exige la maîtrise du langage d'application utilisé sur le serveur, de Javascript et du framework développé ou choisi. Il pose également le problème du référencement : la page chargée par le robot du moteur de recherche ne contient pratiquement rien de ce que va lire l'internaute. On parle aussi de problèmes de sécurité, qui ne semblent pas encore très maîtrisés.

## 1.2 Techniques et outils

### 1.2.1 Langages

#### HTML

Lié aux débuts d'Internet, HTML est un langage de description des pages Internet. Sa principale caractéristique est la possibilité de gérer des extensions multimédia et de permettre la définition de liens vers d'autres pages. Cette technologie ne permet que des pages statiques (le contenu est défini une fois pour toutes). Les différents navigateurs ont proposé des extensions du langage qui ont rendu la réalisation des pages un véritable casse-tête. En outre, les concepteurs de sites avaient l'habitude de mélanger contenu et aspect, rendant ainsi la maintenance pratiquement impossible. Cette situation a provoqué la nécessité d'un nouveau standard et la naissance de XHTML, plus conforme à la norme XML. Parallèlement, l'industrie a forcé la préparation d'une version de HTML intégrant de nouvelles fonctionnalités (surtout liées au multimédia). Cette version, nommée HTML 5, est toujours en phase de définition, mais la plupart des navigateurs en implémentent certaines normes.

#### XML

Né après HTML, XML se veut un moyen de représenter du contenu de manière lisible à la fois par les ordinateurs et les êtres humains. Basé comme HTML sur le principe des balises, il introduit deux notions fondamentales :

- un document XML doit être *bien formé*, c'est-à-dire se conformer à un ensemble de règles communes à tous les documents XML (balises en minuscules, conventions d'écriture, nécessité de fermer toutes les balises, interdiction de chevauchement...)
- la *validité* se définit par rapport à une norme externe (soit un DTD ou *Document Type Definition*, soit un XML-schéma) qui spécifie les balises autorisées, les attributs et leur types, ainsi que la structure hiérarchique du document.

XML peut servir à structurer n'importe quel type d'information : une description bibliographique, les paramètres d'un programmes, un texte complet avec sa mise en page, des informations échangées par deux programmes différents...

#### XHTML

Pour résoudre les différents problèmes rencontrés par HTML au cours de son évolution, le consortium W3C a proposé plusieurs normes pour un nouveau langage. Parmi ceux-ci, XHTML se base sur XML dont il respecte la syntaxe et les contraintes (publication de DTD disponibles pour la vérification). On encourage les utilisateurs à ne plus utiliser de caractéristiques de présentation au sein d'XHTML et de les placer dans des feuilles de styles.

#### CSS

Les feuilles de styles permettent de spécifier l'aspect de documents XML (et parmi eux XHTML). Un style associe des caractéristiques d'apparence à une balise (mise dans un contexte et éventuellement assortie d'un identifiant unique ou d'une classe). En jouant sur les valeurs par défaut et sur l'héritage des styles du parent, on parvient à des mises en page complexes. Ce système permet d'alléger les documents (X)HTML et surtout de donner des styles communs à plusieurs pages. Malheureusement, tous les navigateurs ne respectent pas les standards et cela oblige les concepteurs de sites à prendre en compte un nombre croissant de programmes clients.

## Java

Initialement créé pour être embarqué dans des appareils divers, Java s'est rapidement retrouvé dans une machine virtuelle qui s'exécute au sein d'un navigateur. Ce langage met l'accent sur la portabilité des programmes et la sécurisation (accès limité aux ressources locales, interdiction de charger des informations en provenance de sites étrangers au site d'origine..). Il a souvent exigé des machines puissantes. L'idée était séduisante, mais on voit de moins en moins de sites en Java proposant des appliquestes<sup>1</sup>. Java s'est concentré sur d'autres marchés : les programmes indépendants, les serveurs et la gestion de nombreux appareils électroniques.

## Javascript

Sans rapport avec le précédent, c'est également un langage utilisé sur le client. Il permet d'accéder à toute la structure du document en cours d'affichage à travers un modèle hiérarchique (DOM). Cet accès se fait en lecture et en écriture. Il autorise également l'affichage de boîtes de dialogues indiquant des informations et des erreurs. Très puissant, il souffre également de quelques limitations dues aux incompatibilités entre les différents navigateurs. Ici encore, un lourd travail de variantes doit s'envisager.

## Flash

Langage propriétaire d'Adobe, il permet la réalisation d'animations multimédia. Pour certains domaines, notamment artistiques, il sert à réaliser des sites superbes, mais totalement fermés. Il est gourmand en ressources et en temps de chargement. Totalement inutilisable par les personnes ayant des problèmes de vision.

## PHP

Langage dérivé de Perl, lui-même descendant de C, il tourne sur le serveur. Souvent inséré sous forme de module dans le serveur HTTP, il provoque l'interprétation des pages qui possèdent une extension reconnue (.php, .phpx où x représente la version). En mélangeant instructions HTML et instructions PHP, on permet la génération d'un document original, généralement en réponse à des valeurs renvoyées précédemment par l'internaute. Ce langage a connu plusieurs versions, dont la dernière PHP5 propose un modèle objet plus sophistiqué et l'intégration de XML. Il anime un fort pourcentage de sites dynamiques. En sa faveur, on peut mentionner aussi une communauté de développeurs extrêmement active. La sortie de PHP6, prévue initialement pour juin 2008, devrait apporter par la suite le support d'Unicode et la définition des espaces de noms, dont l'absence actuelle contrarie la créativité des concepteurs. Suite à des choix malencontreux, cette version a été abandonnée et les innovations les plus remarquables ont été intégrées à PHP 5.3 en attendant la suite.

## ASP

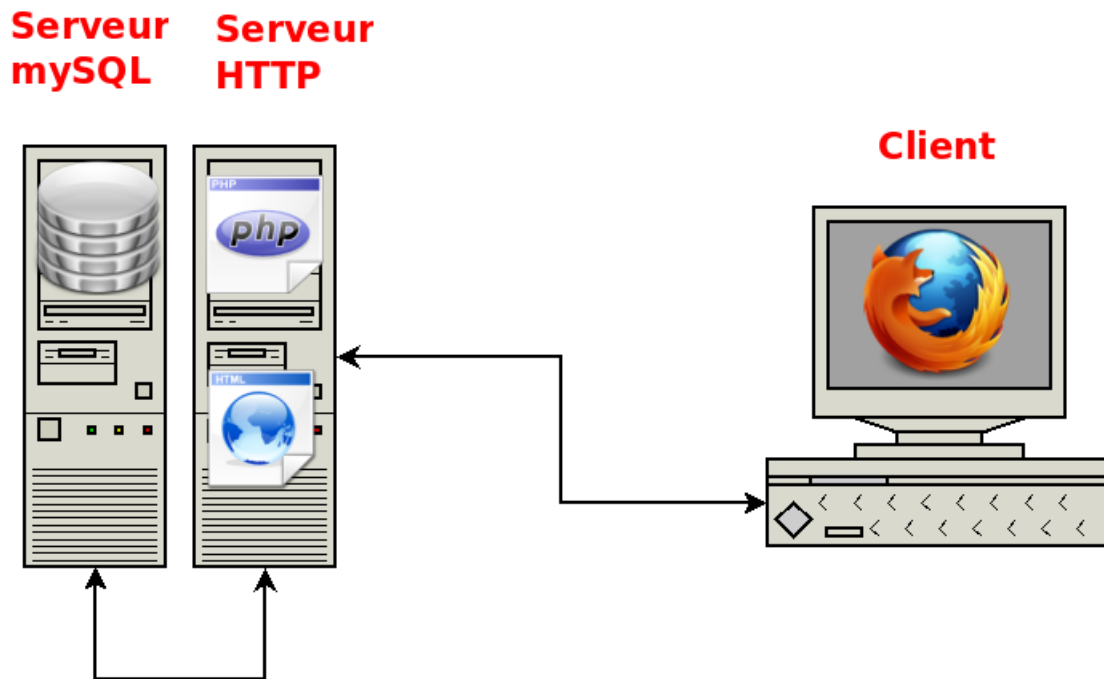
À la différence de PHP, ASP n'est pas un langage, mais une structure composée d'objets accessibles par deux autres langages (VBScript ou JScript). Il permet la création dynamique de pages HTML, mais ne peut tourner que sur un serveur IIS de Microsoft. Il est peu à peu abandonné au profit de ASP.net qui intègre le même concept dans la plate-forme de développement .net. Son utilisation requiert un environnement extrêmement lourd et complexe, fortement intégré à toute une gamme de produits commerciaux.

---

1. *Appliquette* est la traduction québécoise de l'anglais *applet*.

## 1.2.2 Programmes et extensions

Nous avons fait le choix du langage PHP et de la base de données mySQL.



### Serveurs

Deux serveurs et un module sont nécessaires pour l'utilisation de PHP :

- le serveur Apache, couramment en version 2, permettra l'envoi des pages HTML et des autres fichiers et la réception des informations placées dans les formulaires par l'internaute.
- le serveur mySQL offre l'accès à des bases de données.
- le module PHP 5, intégré dans le serveur Apache, transformera les pages php en code HTML. Notons qu'il est possible d'installer PHP sur un serveur IIS, mais que cela oblige à disposer d'un Windows en version professionnelle.

Un programme nommé PHPmyAdmin constitue un moyen convivial de gérer la base de données.

L'installation de ces trois programmes et de leurs utilitaires est assez délicate. Il est conseillé d'utiliser un installateur qui réalise les choses de manière transparente. Tant sous Linux que sous Windows, je ne peux que recommander XAMP (<http://www.apachefriends.org/fr/xampp.html>).



### Navigateurs

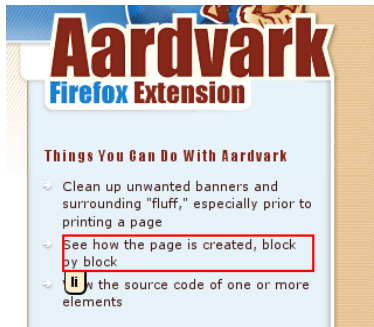
Le meilleur navigateur actuel est sans conteste **Firefox 3.6**. Il offre une grande fidélité aux standards CSS, une grande rapidité et une flopée d'extensions utiles et gratuites. Malheureusement, il faudra vérifier que les autres navigateurs rendent bien les effets attendus. Pour un projet de débutant, on se contentera de vérifier la compatibilité avec Internet Explorer 6. Pour un projet d'envergure professionnelle, il faudra vérifier la compatibilité avec la majorité des

navigateurs : Internet Explorer 6, 7 et 8, Firefox 2 et 3, Opera, Google Chrome et Safari pour Mac. Ces trois derniers ne posent en général pas trop de problèmes..

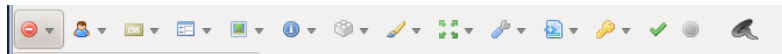
## Extensions de Firefox

Il existe des dizaines d'extensions utiles. Il faut éviter de les installer toutes, au risque de surcharger sa machine. Je conseille personnellement :

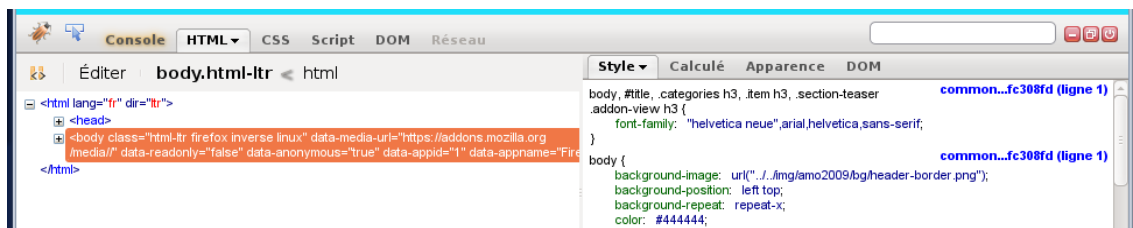
- **Aardvark**, au nom imprononçable, permet d'analyser la structure d'une page, d'en isoler des parties et de voir le code par portions (<http://karmatics.com/aardvark/> ou <https://addons.mozilla.org/fr/firefox/addon/4111/>).



- **Web developer** offre une panoplie d'outils indispensables pour le développeur. Une version française se trouve sur le site de l'auteur (<http://chrispederick.com/work/webdeveloper>).



- **Firebug** sert à examiner la structure du document DOM et intercepte toutes les erreurs de style ou de javascript. Il sera surtout utile pour la rédaction des scripts (<https://addons.mozilla.org/firefox/1843/>).



- **FireFTP** est un client FTP intégré à Firefox. Il existe également en version française (<http://extensions.geckozone.org/FireFTP>).

## Clients FTP

Un client FTP sera utile pour télécharger (uploader) les fichiers vers le site final. Il en existe plusieurs. Personnellement j'utilise le client FireFTP pour Firefox et le client intégré dans le logiciel Midnight Commander. Sous Windows, il existe un excellent client intégré à Total Commander, qui offre d'autres services (<http://www.ghisler.com/>). Ce programme est payant, mais peut-être essayé gratuitement sans limite de temps<sup>2</sup>.

2. Je fais une petite page de publicité pour ce logiciel dont la licence m'a coûté 900FB il y a douze ans et que je peux continuer à utiliser pour mon plus grand confort quand je m'égare dans Windows.

## 1.3 Procédure de réalisation d'un site Web

### 1.3.1 Analyse du problème

On se référera au cours d'analyse pour mener une étude préalable, une étude technique et une étude détaillée du site à réaliser. On insistera sur les points suivants :

- quels utilisateurs seront amenés à consulter le site ?
- quelles sont leurs attentes ?
- énumérer les fonctions à prévoir ainsi que la logique de navigation entre elles

Au terme de cette étude, on doit disposer d'un schéma de la base de données et d'un plan du futur site.

### 1.3.2 Création du site sur la machine de développement

La création d'un site se décompose en trois parties :

1. la création de la base de données sur le serveur MySQL au moyen des outils de manipulations (le plus simple est d'utiliser phpMyAdmin). Un certain nombre de données préalables peuvent être encodées.
2. créer l'arborescence du site avec les fichiers statiques éventuels (HTML), les fichiers dynamiques (PHP), les feuilles de styles (CSS), les images, les bibliothèques éventuelles (en javascript ou en PHP). Un site un peu complexe demandera une arborescence soignée (conçue dans la phase d'analyse).
3. les tests de mise au point.

### 1.3.3 Installation du site sur le serveur de production

L'installation sur le site final suppose les étapes suivantes :

1. l'exportation de la base de données et sa réimportation sur le site final (encore une fois phpMyAdmin constitue l'outil universel).
2. l'*upload* des fichiers vers le site à l'aide d'un client FTP<sup>3</sup>.
3. la sécurisation des répertoires afin d'éviter les intrusions.

Quelques remarques sur les contraintes imposées par le site de production :

- la base de données, son nom, le nom de l'utilisateur et son mot de passe ne sont pas toujours gérables par le réalisateur du site. Il convient donc de ne pas dupliquer ces paramètres dans tous les scripts qui utilisent la base de données. Une bonne pratique consiste à placer tous ces paramètres dans un fichier lu en inclusion par tous les scripts qui en ont besoin.
- la plupart des fournisseurs utilisent Linux ou un système Unix comme système d'exploitation. Il en résulte que les noms des fichiers et des tables sont sensibles aux majuscules. Il convient, si on développe sous Windows, de faire très attention à ce point sous peine d'avoir des liens non fonctionnels. Pour éviter ce problème, je conseille d'adopter des conventions universelles de nommage : par exemple, tous les noms de fichiers et de tables en minuscules.
- si on dispose d'un site devant partager plusieurs applications mais qu'on ne dispose que d'une seule base de données, on s'évitera des conflits d'homonymie en donnant un préfixe propre à l'application à chacune des tables.

---

3. Personnellement j'utilise le protocole SSH et le logiciel `rsync` pour réaliser la synchronisation, mais il suppose un fournisseur qui propose cette option.