

FORMATION APACHE
LE SERVEUR WEB LEADER DU MARCHÉ

« Maîtriser les points essentiels du serveur web Apache »

Table des matières

Théorie.....	3
Définitions.....	3
HTML.....	3
HTTP.....	3
Apache.....	3
Page web.....	4
Principes.....	4
Le modèle client-serveur.....	4
Le cas d'HTTP.....	4
Le cas des pages dynamiques.....	5
Le cas plus complexe des architectures 3-tiers.....	5
Grands principes d'Apache.....	6
Les droits (htaccess ..).....	6
Les virtual hosts.....	6
Le reverse proxy.....	6
Pratique.....	7
Installation.....	7
Page HTML.....	7
Configuration de base.....	7
Arrêter / redémarrer Apache.....	7
Directives de configuration.....	7

I. Théorie

1 Définitions

HTML

L'Hypertext Markup Language, généralement abrégé HTML, est le langage informatique créé et utilisé pour écrire les pages Web. HTML permet en particulier d'insérer des hyperliens dans du texte, donc de créer de l'hypertexte, d'où le nom du langage.

Définition wikipédia (<http://fr.wikipedia.org/wiki/HTML>)

HTTP

Le Hypertext Transfer Protocol, abrégé HTTP, littéralement « protocole de transfert hypertexte », est un protocole de communication informatique client-serveur développé pour le World Wide Web. Il est utilisé pour transférer les documents (document HTML, image, feuille de style, etc.) entre le serveur HTTP et le navigateur Web lorsqu'un visiteur consulte un site Web.

HTTPS (Secured) est la variante du HTTP sécurisé avec les protocoles SSL ou TLS. Il permet au visiteur de vérifier l'identité du site auquel il accède grâce à un certificat d'authentification. Il permet également de chiffrer la communication. Il est généralement utilisé pour les transactions financières en ligne : commerce électronique, banque en ligne, courtage en ligne, etc.

Définition wikipédia (<http://fr.wikipedia.org/wiki/HTTP>)

Apache

Le serveur HTTP le plus répandu (environ 70% des sites web selon netcraft) sur Internet

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

Apache has been the most popular web server on the Internet since April 1996. The November 2005 Netcraft Web Server Survey found that more than 70% of the web sites on the Internet are using Apache, thus making it more widely used than all other web servers combined.

The Apache HTTP Server is a project of the Apache Software Foundation.

Page web

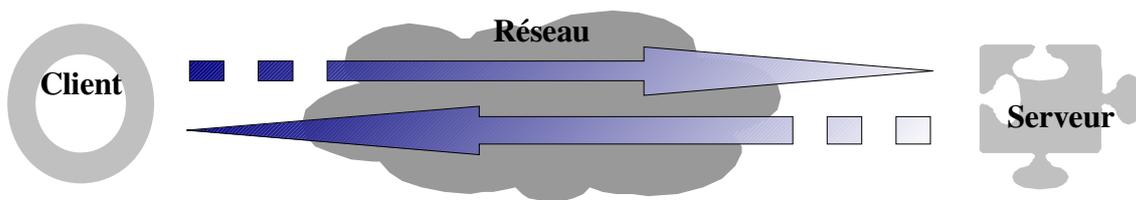
Une page web est un **fichier** qui s'utilise principalement avec un navigateur web. Il permet de définir du contenu (texte, images) ainsi que la mise en page de ce contenu. On parle souvent de page web statique ou à l'inverse dynamique, cela représente le processus de construction de la page :

- une page web statique est un fichier dont le contenu ne change pas sans une intervention humaine
- une page web dynamique est un fichier générée par un serveur web et dont le contenu peut changer selon différents paramètres, par exemple, l'état d'un compte bancaire sur le site d'une banque

2 Principes

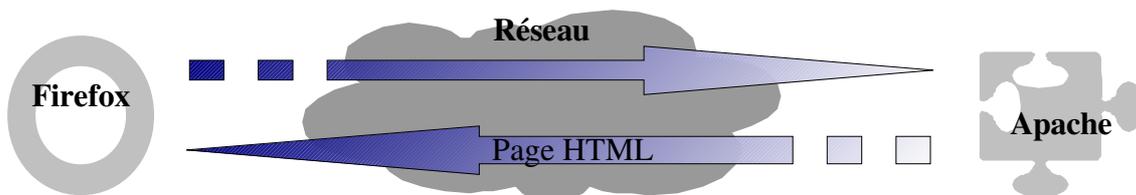
Le modèle client-serveur

Un client accède à une ressource via un ¹serveur :



Le cas d'HTTP

Un navigateur web (client) demande une page web à un serveur web :



Une fois la page téléchargée le navigateur peut l'afficher sur l'écran de l'ordinateur du client.

Le cas des pages dynamiques

Nous avons vu dans les définitions qu'il existe deux grandes catégories de pages web, les pages statiques et les pages dynamiques. Les dynamiques sont certainement celles que l'on retrouve le plus souvent sur la toile car elles permettent d'afficher des contenus qui changent au cours du temps.

Dans les 2 cas ce sera toujours du HTML qui sera envoyé au client, par contre la manière de fabriquer ce HTML sera complètement différente. Dans le cas de la page dynamique, nous l'avons

¹ ici le serveur est un **logiciel** exécuté sur une machine locale ou distante

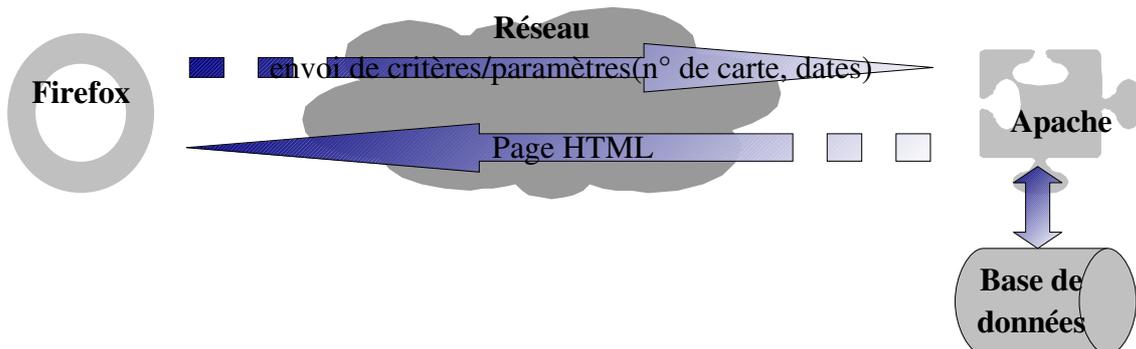
FORMATION APACHE

déjà vu, c'est un humain qui a écrit la page, et s'il veut la changer il devra la modifier « à la main ». Par contre dans le cas d'une page dynamique la page est générée via un langage de programmation tel que PHP, Perl, Python ou Javascript (pour les plus utilisés). Ce langage de programmation va permettre d'avoir un contenu qui va pouvoir varier selon les paramètres qu'aura prévu la personne qui a créé la page dynamique. Un exemple très simple est tout simplement l'affichage de la date dans la page. On peut représenter ce processus ainsi :



Le cas plus complexe des architectures 3-tiers

Afin d'afficher un contenu moins triviale que la date du jour, on couple souvent processus vu ci-dessus, une base de donnée afin d'y stocker .. des données. Ainsi on pourra par exemple y stocker les noms des utilisateurs afin de les afficher lorsque ces derniers se connectent au site.



1 On distingue 2 grandes catégories : les langages coté serveur, et les langages coté client

3 Grands principes d'Apache

Les systèmes de droits

<http://httpd.apache.org/docs/2.2/howto/auth.html>

La relation URL <-> système de fichiers

<http://httpd.apache.org/docs/2.2/urlmapping.html>

S'il ne fallait lire qu'une seule chose

http://httpd.apache.org/docs/2.2/misc/security_tips.html

II. Pratique

Cet atelier pratique suivra le plus possible la documentation d' Apache.

Installation

- Installation packagée
- Tour d'horizon de l'installation par compilation

Page HTML

- Création d'une page HTML très simple mais *valide*
- Découverte du validateur
- Modification de la page HTML pour qu'elle affiche la date

Configuration de base

- La directive *Listen*
- La directive *ErrorLog*
- Faire en sorte qu' Apache soit exécuté au démarrage de la machine

Arrêter / redémarrer Apache

- Les commandes `kill` et `tail -f`
- La commande `apachectl`

Directives de configuration

- La liste complète des directives : <http://httpd.apache.org/docs/2.2/mod/directives.html>
- Les directives les plus courantes : <http://httpd.apache.org/docs/2.2/server-wide.html>
- La directive *order* : http://httpd.apache.org/docs/2.2/mod/mod_authz_host.html#order
- Les *.htaccess* : <http://httpd.apache.org/docs/2.2/howto/htaccess.html>
 - In general, you should never use *.htaccess* files unless you don't have access to the main server configuration file. There is, for example, a prevailing misconception that user authentication should always be done in *.htaccess* files. This is simply not the case. You can put user authentication configurations in the main server configuration, and this is, in fact, the preferred way to do things
- Les *virtual hosts* : <http://httpd.apache.org/docs/2.2/vhosts/>
- Listen / virtual hosts : <http://httpd.apache.org/docs/2.2/bind.html>

FORMATION APACHE

- Listen n'implémente aucun Serveur Virtuel. Cette directive sert simplement à informer le serveur principal sur quels adresses et ports écouter. Dans le cas où aucune section <VirtualHost> n'est utilisée, le serveur répondra de la même manière pour toutes les requêtes qu'il acceptera. Cependant des sections <VirtualHost> peuvent être utilisées pour qu'Apache réagisse de façon différente à une requête selon l'adresse ou le port. Avant d'implémenter un Serveur Virtuel au moyen de la directive <VirtualHost>, la directive Listen doit être utilisée pour que le serveur écoute sur l'adresse ou le port spécifié. Une section <VirtualHost> peut alors être utilisée pour définir la réaction du Serveur Virtuel pour une adresse et un port spécifique. À noter que si un Serveur Virtuel est positionné au moyen de la directive <VirtualHost> sur une adresse et un port sur lesquels le serveur n'est pas à l'écoute, le Serveur Virtuel ne sera pas accessible.
- HTTPS : <http://httpd.apache.org/docs/2.2/ssl/>
- La directive *per-user* : http://httpd.apache.org/docs/2.2/howto/public_html.html