



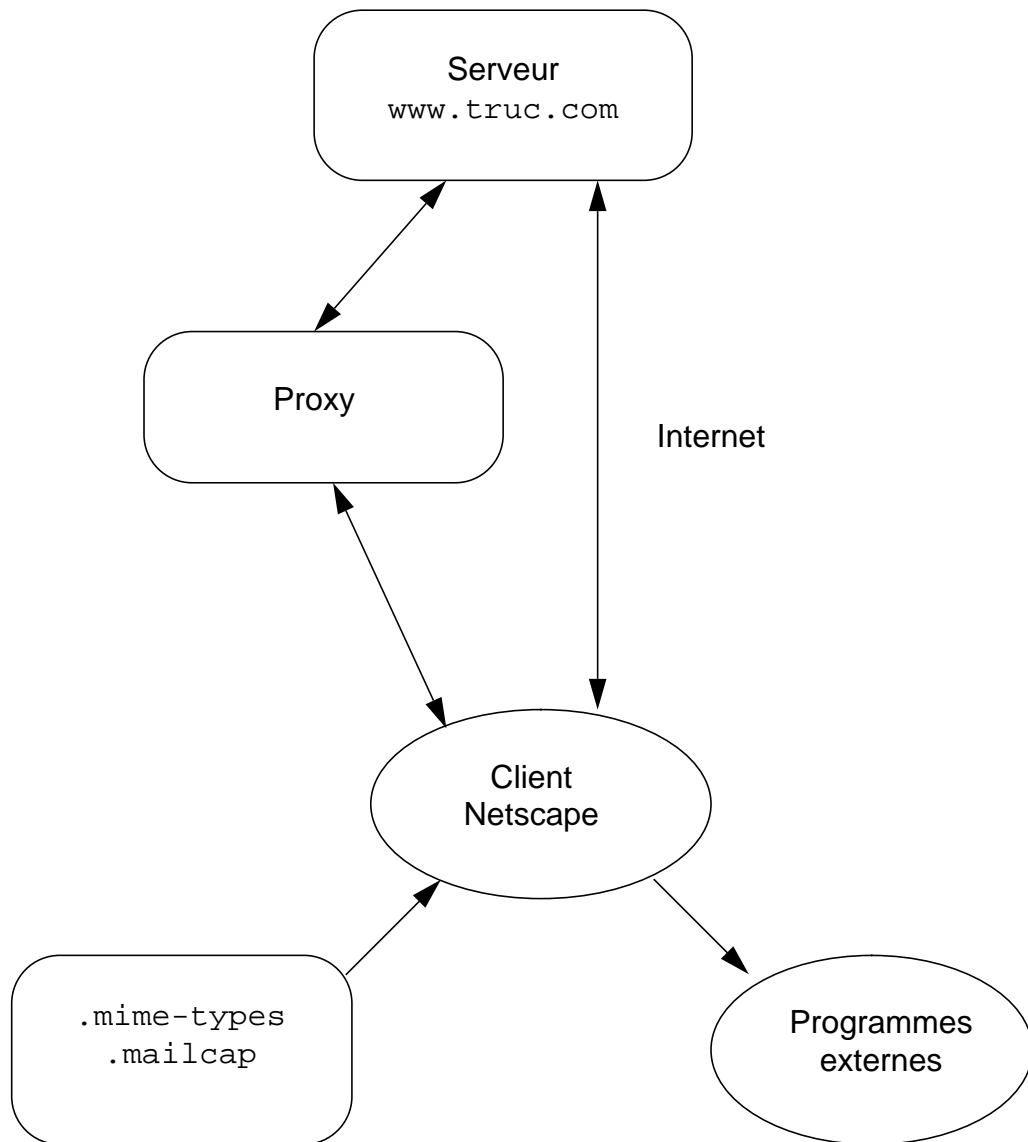
Internet

Protocole et serveurs HTTP

Michel Buffa, Stéphane Laveau

Université de Nice Sophia Antipolis
UFR Sciences
Parc Valrose, Nice
Téléphone : 92 96 5115 — Télécopie : 92 96 51 55
Email : buffa@essi.fr

Le Web: Comment ca marche?



HTTP: Le protocole

- TCP
 - Port 80 par défaut.
- «Connectionless»
- Client :
 - Requête
 - Entêtes
- Serveur :
 - Code Réponse
 - Entêtes
 - Réponse
- Conçu pour être le plus simple possible.

<http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTTP2.html>

Requêtes

- **GET :**
 - Document
 - Script
 - Mise à jour

- **POST :**
 - Script

- **HEAD :**
 - Entête

- **PUT,DELETE, etc...**

<http://www.w3.org/hypertext/WWW/Protocols/HTTP/Request.html>

Classes de réponses

- **2xx: Succès**
 - 200: OK
- **3xx: Redirection**
 - 304: Inchangé
- **4xx: Erreur Client**
 - 401 : Non Autorisé
 - 404 : Document inexistant
- **5xx: Erreur Serveur**
 - 500: Erreur script (entre autres)
- **600 : Erreur Gateway/Cache**
 - non standard

<http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTRESP.html>

Entête de requête

- **From: adresse email**
 - Non envoyé par la majorité des clients pour raisons de confidentialité.
- **Accept: Une liste de types MIME**
 - image/gif, text/*, */*
 - application/postscript, video/mpeg
- **Accept-Encoding: Une liste de méthode de codage MIME**
 - ex: x-compress, x-gzip x-zip
- **Accept-Language: Une liste de langues désirées.**
 - Passe à la trappe dans 99% des cas, mais existe.

Entête de requête (2)

- **User-Agent: Un identificateur du client**
 - utile pour répondre différemment suivant les clients, ou pour déboguer.
 - ex: Mozilla 2.0b3 (X11; SunOS 2.4)
- **Referer: la page d'où l'on vient.**
- **Authorization: login passwd**
 - Assez faible question sécurité
- **If-Modified-Since: Date**
 - permet de transférer la page que si elle a été modifiée (*très* utile pour les caches)
- **Pragma: no-cache**
 - C'est le seul défini, en théorie c'est un peu fourre-tout.

<http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTRQ-Headers.html>

Entête de réponse

- Server : Type de serveur
- Date :
- Last-Modified:
 - Utile pour les caches.
- Content-Type: Le type MIME
 - text/html
 - image/gif
 - application/postscript
 - text/plain
 - audio/basic
 - video/mpeg
- Content-Length: la longueur des données en octets.
 - Utile pour savoir si tout s'est bien passé.
 - Non obligatoire...

Entête de réponse(2)

- Content-Encoding
 - x-compress, x-gzip, x-zip
- Content-Language

http://www.w3.org/hypertext/WWW/Protocols/HTTP/Object_Headers.html

http://www.w3.org/hypertext/WWW/Protocols/rfc1341/0_Abstract.html

http://home.netscape.com/assist/helper_apps/mime.html

http://www.yahoo.com/Computers_and_Internet/Multimedia/MIME/

<http://andrew2.andrew.cmu.edu/cyrus/email/standards-MIME.html>

Réponse



- Un seul objet par réponse
 - Pas de type MIME multipart/* à part dans certaines animations (voir plus tard).
 - Pour l'instant....

Un serveur

- Différents types:
 - Netscape (Unix/Windows NT)
 - Apache
 - NCSA 1.5
 - CERN (dépassé)
 - WebStar (Windows)
 - MacHTTP (Mac)
 - Spinner
 - etc...

<http://www.proper.com/www/servers-survey.html>

http://www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/HTTP/Servers/

- Apache:
 - Gratuit (licence de type GNU)
 - Modulaire
 - Simple (!)
 - Efficace (pas de forking furieux)
 - Développement actif

<http://www.apache.org/>

Structure d'un serveur

- Transformation URL -> fichier ou script.
- Vérification d'identité.
 - Le client est il qui il prétend être ?
- Vérification d'accès.
 - Le client est il autorisé à effectuer cette requête ?
- Détermination du type MIME des données.
- Envoi de la réponse au client.
- Mise à jour les logs.

Configuration

- 3 fichiers en général
 - httpd.conf (ou option -f)
 - srm.conf (ou ResourceConfig)
 - access.conf (ou AccessConfig)
 - mime.types (ou TypesConfig)

- 3 fichiers de logs (par défaut)
 - pid
 - error_log
 - access_log

Que peut faire un serveur?

- Servir une page en ajoutant les entêtes appropriés.
 - Cas le plus commun
- Servir une page telle quelle.
 - Utile pour des pages à expiration rapide par exemple.

http://www.apache.org/docs/mod_asis.html

- Exécuter un script et renvoyer le résultat comme une page
 - Interprétation de formulaires, etc...

http://www.apache.org/docs/mod_cgi.html

- Exécuter un script et renvoyer le résultat tel quel.
 - Script dits nph-
 - Roulettes et code de réponse autre que 200 OK.

Que peut faire un serveur?(2)

- Vérifier la machine d'origine (ou le dernier proxy sur la chaîne).
 - Certaines autorisations.
 - Redirection suivant la localisation géographique (foireux).

http://www.apache.org/docs/mod_access.html

- Vérifier un mot de passe et un login.
 - Autorisation un petit peu élaborée.
 - Equivalent au login Unix (=faible)

http://www.apache.org/docs/mod_auth.html

- Construire une page au vol avec des ordres spécifiques.
 - Equivalent de #include
 - Permet d'insérer le résultat de scripts dans des pages (compteurs, etc...)

http://www.apache.org/docs/mod_include.html

Que peut faire un serveur?(3)

- Reconnaître les entêtes de requêtes et servir des pages différentes suivant le client ou ses préférences.
 - Avoir des pages différentes pour chaque browser.
 - Parler la langue demandée.

<http://www.apache.org/docs/content-negotiation.html>

http://www.apache.org/docs/mod_mime.html

- Lister des directories.

http://www.apache.org/docs/mod_dir.html

- Encrypter les données et sécuriser les transactions.

<http://www.algroup.co.uk/Apache-SSL/>

Que peut faire un serveur?(4)

- Renvoyer des pages d'erreurs personnalisées.
 - Rediriger les non-autorisés vers un formulaire d'inscription

<http://www.apache.org/docs/custom-error.html>

- Répondre différemment suivant le nom de la machine.
 - Différentes adresses IP ou interface virtuelle.
 - Permet d'avoir plein de serveurs peu actifs sur la même machine appartenant à des organisations différentes.

<http://www.apache.org/docs/virtual-host.html>

Caches/Proxies

- Firewall/sécurité
 - Un réseau isolé communique via une machine passerelle.
 - Proxies
- Diminuer la charge sur les réseaux.
 - Plusieurs utilisateurs vont chercher la même page. Elle traverse plusieurs fois l'Atlantique, et c'est lent.
 - Caches

<http://harvest.cs.colorado.edu/>

<http://www.w3.org/pub/WWW/Daemon/User/Proxies/Proxies.html>

http://home.netscape.com/comprod/proxy_server.html

Caches/Proxies: Problèmes

- Que cacher (ou pas) ? Heuristiques:
 - cgi-bin
 - ?
- Confidentialité.
 - Documents réservés à une organisation.
 - Logs de toutes les opérations par machine.
- Comment déterminer le temps à vivre ?
 - Expires:
 - Last-modified:
 - 2 durées de vie : l'une sans vérification, l'autre avec.

Caches/Proxies: Problèmes

- Place disque.
 - Il faut environ 500M pour une centaine de personnes.
- Les caches font apparaître plusieurs utilisateurs comme un seul.
- Les données dans le cache sont parfois erronées. Il faut que l'utilisateur puisse forcer le rechargement.

<http://www.hensa.ac.uk/wwwcache/>

<http://www.hensa.ac.uk/wwwcache/local.cache.guidelines.html>

<http://web.pasteur.fr/other/computer/cache/>

HTTP: Le Futur

- **Connections persistantes.**
 - Keep-Alive (Netscape, NSCA 1.5)
- **HTTP-NG**
 - Entête en binaire
 - Connections persistantes
- **Sécurité/Encryption**
 - HTTPS (Netscape)
 - SHTTP, SMIME (Spry)
- **Paiement**

<http://hoohoo.ncsa.uiuc.edu/docs/howto/KeepAlive.html>

<http://www.w3.org/pub/WWW/Protocols/HTTP-NG/http-ng-status.html>

<http://www.ics.uci.edu/pub/ietf/http/>

<http://home.netscape.com/newsref/ref/netscape-security.html>

<http://www.spry.com/secure.htm>

<http://www.w3.org/pub/WWW/Payments/>

<http://www.globeonline.fr/>

<http://www.fv.com/>