RÉÉCRITURE D'URL AVEC MOD_REWRITE

PRETTY URL

CHEMINS DE RÉPERTOIRES STRUCTURÉS SELON LE NOM

Les sites avec des milliers d'usagers hébergés utilisent des structures de répertoires comme par exemple un sous-répertoire qui est la première lettre du nom d'usager, ou un sous-répertoire qui consiste en les deux premières lettres du nom de domaine.

/~foo/anypath est **/home/f/foo/.www/anypath**

/~bar/anypath est **/home/b/bar/.www/anypath**

On utilise la règle suivante pour transformer le symbole vague en l'expansion démontrée ci-haut.

```
RewriteEngine on
RewriteRule   ^/~(([a-z])[a-z0-9]+)(.*)  /home/$2/$1/.www$3
```

PROBLÈME DU SLASH À LA FIN

Si on tente d'accéder /~quux/foo à la place de /~quux/foo/ . Alors le server cherche un fichier qui s'appelle foo et si c'est un répertoire, alors il ne comprend pas. Actuellement, ce problème est fixé par Apache si il survient au moment de la requête mais si c'est déjà après certaines transformations et réécritures d'url, alors il faut explicitement ajouter la slash aux url, par mod_rewrite pour s'assurer que les images et les objets inclus sont aussi corrrects.

Une requête pour : image.gif dans/rep/foo/index.html deviendrait /rep/image.gif

Code spécifique à placer dans le bon répertoire

```
RewriteEngine  on
RewriteBase    /rep/
RewriteRule    ^foo$  foo/  [R]
```

Code générique dans le répertoire racine

```
RewriteEngine  on
RewriteBase    /rep/
RewriteCond    %{REQUEST_FILENAME}  -d
RewriteRule    ^(.+[^/])$          $1/  [R]
```

CANONIQUE

URL CANONIQUES

Comment utiliser plusieurs noms d'url pour accéder une seule ressource, et forcer le visiteur à voir l'url préférée dans son navigateur.

On effectue une redirection explicite de toutes les url non-canoniques pour que le navigateur (browser) affiche l'url canonique dans sa ligne d'adresse et que toutes les requêtes ultérieures soient à cette adresse. Par exemple, l'url www.gooogle.com est une url valide pour google mais ceux-ci préfèrent corriger tout de suite en google.com.

On remplace /~user par /u/user et on ajoute la slash manquante à la fin de /u/user

```
RewriteRule   ^/~([^/]+)/?(.*)    /u/$1/$2  [R]
RewriteRule   ^/([uge])/([^/]+)$  /$1/$2/   [R]
```

HÔTES CANONIQUES

Dans la même idée, forcer l'utilisation d'un hôte particulier, par exemple www.example.com en lieu de example.com

```
RewriteCond %{HTTP_HOST}   !^fully\.qualified\.domain\.name [NC]
RewriteCond %{HTTP_HOST}   !^$
RewriteRule ^/(.*)         http://fully.qualified.domain.name/$1 [L,R]
```

DOCUMENTROOT DÉPLACÉ

Il est possible que les pages racines de notre site ne soient pas dans le répertoire racine de notre arborescence de répertoires web. Ceci peut être dû à des raisons de priorités. Comment s'assurer qu'un tel déplacement des fichiers permet tout de même de faire suivre les liens absolus vers les documents associés: images etc.

On redirige URL / à /e/www/:

```
RewriteEngine on
RewriteRule   ^/$  /e/www/  [R]
```

PAGES PERSONELLES DÉPLACÉES SUR UN SERVEUR DIFFÉRENT

Si on veut redireiger tous les /~usager/anypath à http://newserver/~usager/anypath

```
RewriteEngine on
RewriteRule   ^/~(.+)  http://newserver/~$1  [R,L]
```

ROBUSTESSE ET CANONIQUE

BALANCEMENT DE SURCHARGE (LOAD BALANCING)

On veut balancer le traffic de 6 mirroirs www.foo.com sur www[0-5].foo.com

On utilise mod_rewrite pour construire un proxy. Premièrement on dédit un des serveurs à être le proxy en indiquant dans le DNS que le serveur numéro 0 est le www.

```
www    IN  CNAME   www0.foo.com.
```

Ensuite, on convertit ce serveur www0.foo.com à servir seulement de proxy pour les 5 autres serveurs (www1-www5) qui feront les traitements. www0 semblera encore surchargé mais ce ne seront que des redirections, pas des traitements CGI, perl ou PHP.

Ceci se réalise en établissant une règle de contact avec un script perl qui effectue le balancement au hasard.

httpd.conf

```
RewriteEngine on
RewriteMap    lb      prg:/path/to/lb.pl
RewriteRule   ^/(.+)$ ${lb:$1}          [P,L]
```

lb.pl

```
#!/path/to/perl
##
##  lb.pl -- load balancing script
##

$| = 1;

$name  = "www";     # the hostname base
$first  = 1;        # the first server (not 0 here, because 0 is myself)
$last   = 5;        # the last server in the round-robin
$domain = "foo.dom"; # the domainname

$cnt = 0;
while (<STDIN>) {
    $cnt = (($cnt+1) % ($last+1-$first));
    $server = sprintf("%s%d.%s", $name, $cnt+$first, $domain);
    print "http://$server/$_";
}
##EOF##
```

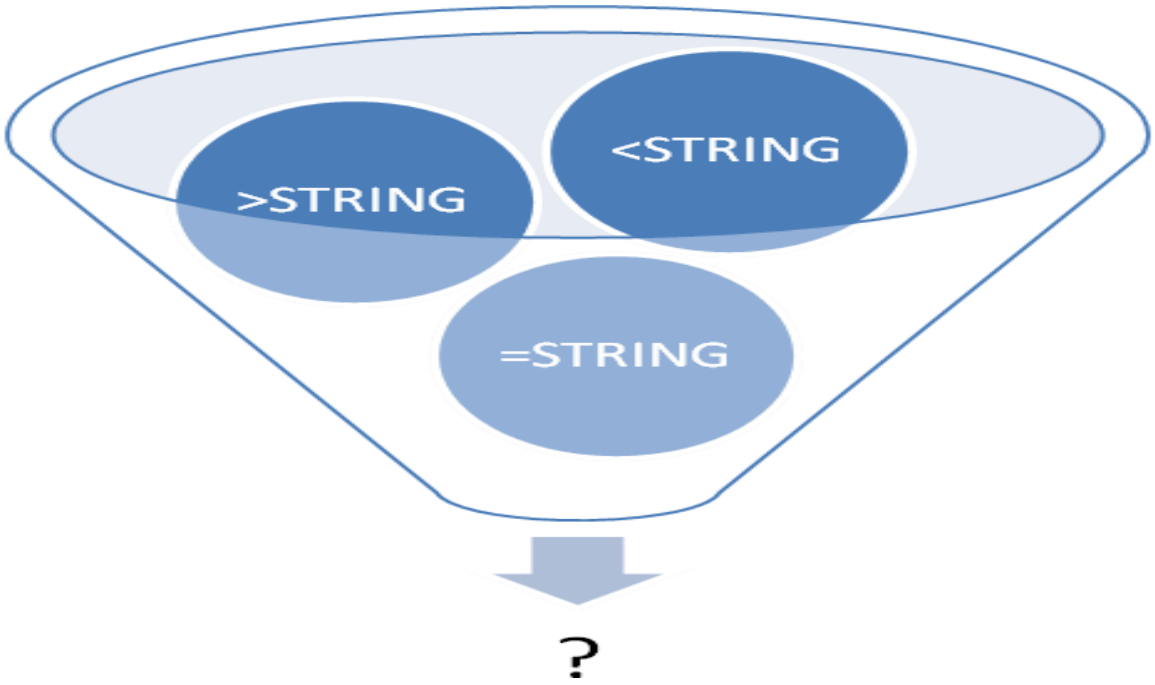
ADAPTABILITÉ

RÉÉCRITURE DÉPENDANTE DU TEMPS

Les pages qui dépendent de l'heure peuvent être traitées de diverses manières: les langages de scripts CGI ou internes aux pages peuvent réaliser des inclusions selon l'heure de la journée, avec des liens dépendant de l'heure de la journée. Ils peuvent aussi renvoyer une en-tête de redirection mais c'est très peu utilisé étant donné que ces redirection affectent le référencement par les moteurs de recherche.

Des variables de type TIME_xxx sont disponibles et peuvent être utilisées en conjonction avec les instruction de comparaison lexicographiques de chaines qui permettent de déterminer si une chaine est plus petites qu'une autre, donc si une heure vient avant l'autre.

PATRONS DE COMPARAISON LEXICOGRAPHIQUE :




```
RewriteEngine on
RewriteCond   %{TIME_HOUR}%{TIME_MIN} >0700
RewriteCond   %{TIME_HOUR}%{TIME_MIN} <1900
RewriteRule   ^test\.html$          test.jour.html
RewriteRule   ^test\.html$          test.nuit.html
```

Donc, entre 7h et 19h, la page test.jour.html est affichée alors qu'entre 19h et 7h, le programme de la nuit est affiché. Utile pour des sites d'informations, de programmes télévisés etc.

ÉCOSYSTÈME WEB

CONTENU DÉPENDANT DU BROWSER

Il est actuellement impossible pour les développeurs web de réaliser des pages sophistiquées avc duc ode portable. Chaque version de navigateur interprète différemment certains codes. Une façon de faire est d'inclure les différents codes dans la même page est d'afficher les bons selon la détection du navigateur avec le langage de script qui lit la variable d'environnement $USER_AGENT.

Une autre manière de faire est de préparer deux pages complètement séparées et de rediriger vers la bonne selon le navigateur.

Cette fonctionnalité serait particulièrement utile pour un site qui proposerait un produit qui s'ajoute justement à ce navigateur !

```
RewriteCond %{HTTP_USER_AGENT}  ^Mozilla/3.*
RewriteRule ^foo\.html$         foo.NS.html         [L]
RewriteCond %{HTTP_USER_AGENT}  ^Lynx/.*        [OR]
RewriteCond %{HTTP_USER_AGENT}  ^Mozilla/[12].*
RewriteRule ^foo\.html$         foo.20.html         [L]
RewriteRule ^foo\.html$         foo.32.html         [L]
```

BLOQUER L'UTILISATION DES IMAGES DU SERVEUR PAR LES AUTRES SITES WEB

Voici la description d'un problème souvent rencontrés par les graphistes qui produisent de populaires images. Les autres webmestres qui créent leur pages web non pas en copiant l'image d'un autre site sur leur serveur mais en liant cette images avec l'url, autrement dit, il y a vol de bande passante.

Pour solutionner ce problème, on exclut alors les envois d'images dont le referrer n'est pas notre site

```
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://www.mon_site.com/.*$ [NC]
RewriteRule .*\.gif$        -                           [F]
RewriteCond %{HTTP_REFERER}         !^$
RewriteCond %{HTTP_REFERER}         !.*/image-avec-gif\.html$
RewriteRule ^inlined-avec-foo\.gif$   -                   [F]
```

BLOQUER LES ROBOTS INDÉSIRABLES

Quand l'indication gentille de /robots.txt est insuffisante et n'est pas respectée, on peut contrôler les robots nous-même avec une détection du User-Agent

```
RewriteCond %{HTTP_USER_AGENT}   ^NameOfBadRobot.*
RewriteCond %{REMOTE_ADDR}       ^123\.45\.67\.[8-9]$
RewriteRule ^/~quux/foo/arc/.+   -   [F]
```