

Comment améliorer le temps de chargement de ses pages

[Retour au sommaire de la lettre](#)

Domaine :	Recherche	Référencement
Niveau :	Pour tous	Avancé

Google vient de l'annoncer : le temps de chargement des pages et le temps de réaction de vos serveurs vont devenir de plus en plus importants pour votre référencement puisque ce sont maintenant des critères officiels de pertinence de l'algorithme du moteur de recherches. Mais ces paramètres sont également stratégiques pour vos utilisateurs, qui "switchent" rapidement vers la concurrence si vos pages mettent trop de temps à s'afficher. Voici, dans cet article, un panorama très complet des différentes solutions à votre disposition pour optimiser ces temps de réaction et faire de vos pages des Formule 1 du Web. Les possibilités sont nombreuses, aussi n'hésitez pas à les mettre en place. Google vous en remerciera en vous attribuant de meilleurs positionnements mais il y a fort à parier que vos visiteurs seront, eux, encore plus ravis de vos initiatives !...

Depuis que Google a inventé le navigateur annoncé comme étant le plus rapide au monde (Chrome), les webmasters sont maintenant priés de concevoir des pages web dont le temps de chargement est optimisé. "La vitesse va être un facteur important dans l'évolution du référencement en 2010" (Matt Cutts). Cette prédiction datant d'un an vient juste d'être confirmée officiellement par Google :

<http://googlewebmastercentral.blogspot.com/2010/04/using-site-speed-in-web-search-ranking.html>.

Important pour votre référencement, mais pas seulement ! Les performances globales d'un site sont également bénéfiques pour votre business. À l'inverse, le temps de latence de vos pages affectent directement votre taux de conversion (<http://bit.ly/9f4M8B>). Alors, retrouvez vos manches et faites de votre site, un véritable bolide de course !

Le SEO à 180 km/heure

Les internautes sont de plus en plus impatients et, près de la moitié d'entre eux, n'attendent pas plus de 2 secondes pour qu'une page se charge

(http://www.akamai.at/dl/whitepapers/ecommerce_website_perf_wp.pdf).

Voici d'autres chiffres éloquentes (<http://performance.survol.fr/2008/06/a-quoi-ca-sert/>) :

- Perdre 500ms, c'est aussi perdre 20% de trafic pour Google ;
- Réduire de 25% le poids d'une page "Google", c'est gagner 25% d'utilisateurs à moyen terme ;
- Augmenter la latence de 100ms signifie, pour Amazon, 1% de ventes en moins ;
- Perdre 400ms, c'est avoir 5 à 9% de taux d'abandon supplémentaire pour Yahoo!.
- Enfin, on estime généralement qu'au-delà de 4 secondes d'attente, le taux de rebond devient réellement important...

Un chercheur, spécialisé dans ces questions, a réussi à mettre en corrélation le temps de chargement d'une page et son taux de rebond. Le script Google Analytics qui permet cette observation est accessible à cette adresse :

<http://www.stevesouders.com/blog/2009/07/27/wikia-fast-pages-retain-users/#comment-730>.

Une autre société spécialisée (<http://blog.webcopyplus.com/2009/06/22/poll-user-place-more-weight-on-design/>) a réalisé un sondage auprès d'un panel d'internautes. Le but était d'analyser les raisons qui les motivaient à quitter prématurément un site... Voici les résultats :

- Design : 24.2% ;
- Contenu : 24.8% ;
- Lenteur : 50.9%.

Les problèmes de lenteur entraînent donc cette série de conséquences :

- Une moins bonne indexation de la part des moteurs ;
- Une expérience "déceptive" chez les internautes ;

- Moins de profondeur de visites ;
- Moins de recherches de la part des utilisateurs ;
- Moins d'affinage des recherches ;
- Moins de revenus "On line" car moins de conversions.

En bref, les mauvaises performances d'un site web vous font perdre de l'argent...

À l'inverse, la vitesse fait partie des éléments fidélisant : "*Users really respond to speed*" (Marissa Mayer).

Nous allons donc voir comment améliorer le temps de téléchargement de la page (son poids) mais aussi les temps de rendu de la page web (Temps de téléchargement vs Temps de chargement).

Quels problèmes pour quelles solutions ?

Bien entendu, le choix de l'hébergeur est l'élément à considérer en priorité. Nous pouvons, à la fois, nous interroger sur le temps de réponse d'un serveur mais aussi sur l'adéquation entre une offre d'hébergement et le trafic généré par un site web.

Voici une autre suggestion de bon sens : limitez le nombre de résolutions DNS en hébergeant l'ensemble de vos fichiers sur un même serveur.

Mais, pour la suite de cet article, nous allons voir qu'il existe d'autres solutions possibles : optimisation du code, optimisation des images, optimisation du cache serveur et, indirectement, du cache des navigateurs.

Les outils de test

La première démarche consiste à identifier les causes et, donc, à se doter des meilleurs outils de "Benchmark" : "On n'améliore réellement que ce qu'on mesure". Nous verrons ensuite quels sont les remèdes à apporter...

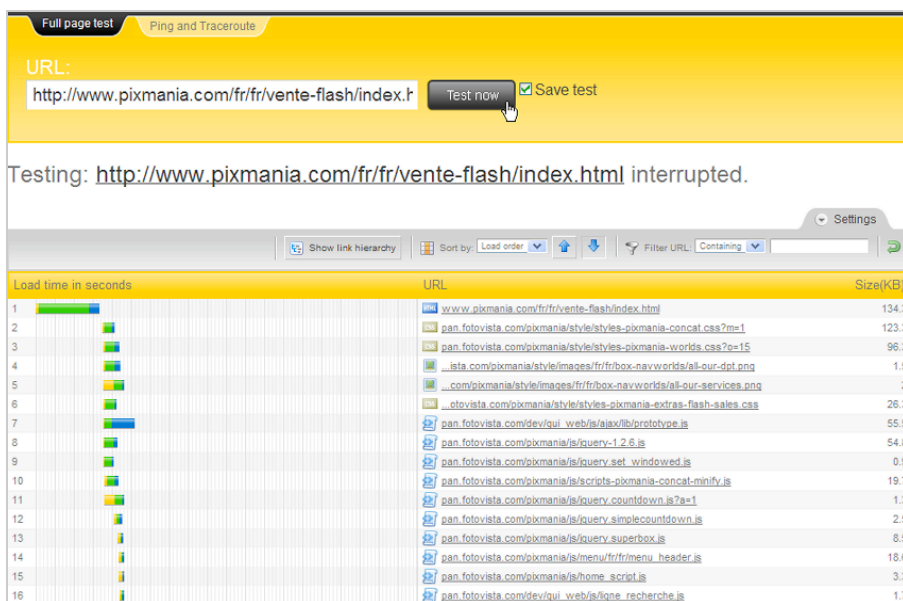
Pingdom Tools

Une bonne manière de se familiariser avec ce service est d'accéder directement à cette page : <http://tools.pingdom.com/>.

Saisissez l'adresse URL d'une page web puis cliquez sur *Test now*.

Il est possible de sauvegarder le test en récupérant l'adresse URL de la page.

Le principe est simple : cet outil va simuler l'affichage de la page dans un navigateur et mesurer le temps de chargement de chacun des éléments qui la composent.

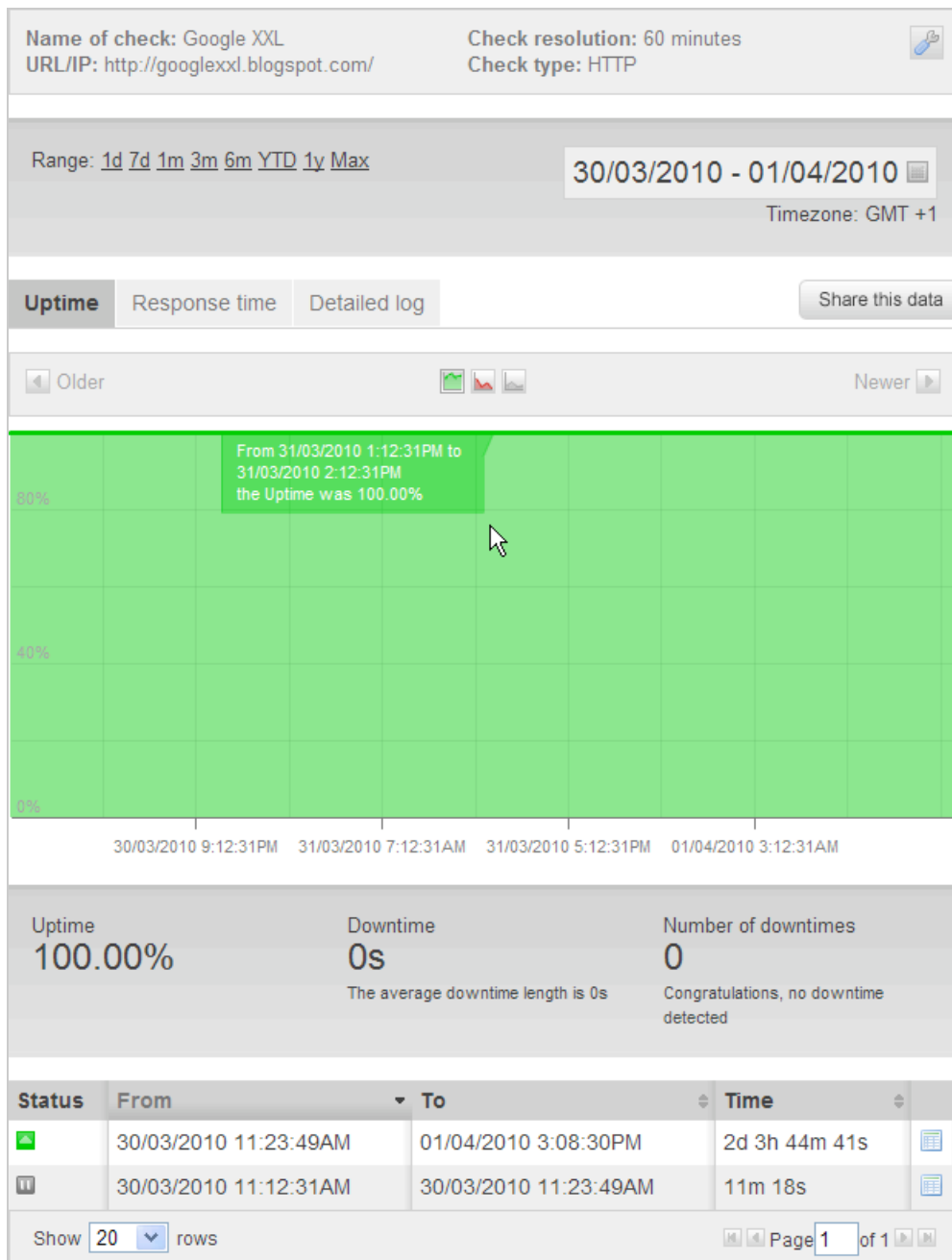


Les différentes barres de progression indiquent :

- Le "Timestamp" de départ (en jaune) ;
- L'horodatage de début de connexion (en vert) ;
- Le téléchargement du premier octet (en bleu) ;
- Le téléchargement du dernier octet (fin de la barre en bleu).

En bref, vous visualisez, du premier coup d'œil, les éléments qui sont les plus lourds en termes de téléchargement.

Le menu déroulant *Sort by* permet de classer les différents éléments qui composent votre page par critères : taille, temps de chargement, ordre de chargement, type de fichiers, etc. Juste à gauche de la fenêtre, vous pouvez vous inscrire au service gratuit proposé par Royal Pingdom. Cet utilitaire permet de vérifier la disponibilité d'un serveur à un intervalle de temps donné (toutes les 5 minutes, par exemple).



1. Une fois que vous avez validé votre inscription, cliquez sur le bouton *Add new check*.

2. Indiquez un nom pour votre test puis définissez l'intervalle de temps voulu.
 3. Cochez le bouton radio *http* puis, dans la rubrique *Check settings*, saisissez l'adresse URL de votre site web.
 4. Cliquez sur le bouton *Test check* afin de vérifier si l'adresse URL donnée est valide puis cliquez sur le bouton *Add check*.
- Vous aurez accès, en permanence, à l'ensemble des statistiques de votre site et serez prévenu dès que le serveur n'est plus joignable ou "en carafe".

L'"uptime" correspond au pourcentage du temps pendant lequel votre site est accessible. La grande majorité des hébergeurs garantissent un uptime très performant mais sans donner la mesure exacte de la disponibilité réelle du serveur. À titre d'exemple, un uptime de 99% correspond à un peu plus de 7 heures d'indisponibilité par mois. Ce qui représente une sacrée perte de clients potentiels !

Le lien *Settings* permet de paramétrer une alerte par SMS. Vous pouvez également utiliser l'API Pingdom et, bien sûr, profiter des avantages offerts par les versions payantes. C'est, à notre avis, le service le plus efficace dans sa catégorie...

Google Webmaster Tools

Les Outils Google pour les webmasters proposent une fonctionnalité permettant de se faire une première idée :

Développez la rubrique *labos* puis cliquez sur le lien *Performances du site*.

- Le temps de chargement moyen de vos pages va être indiqué ainsi qu'une comparaison avec la moyenne des autres sites ;
- Le graphique indique l'évolution du temps de chargement moyen des pages de votre site au cours des 5 derniers mois.



Des adresses URL "témoin" sont listées avec, à chaque fois, des suggestions : compression avec Gzip, groupement des fichiers JavaScript, limitation du nombre de résolution DNS, etc.

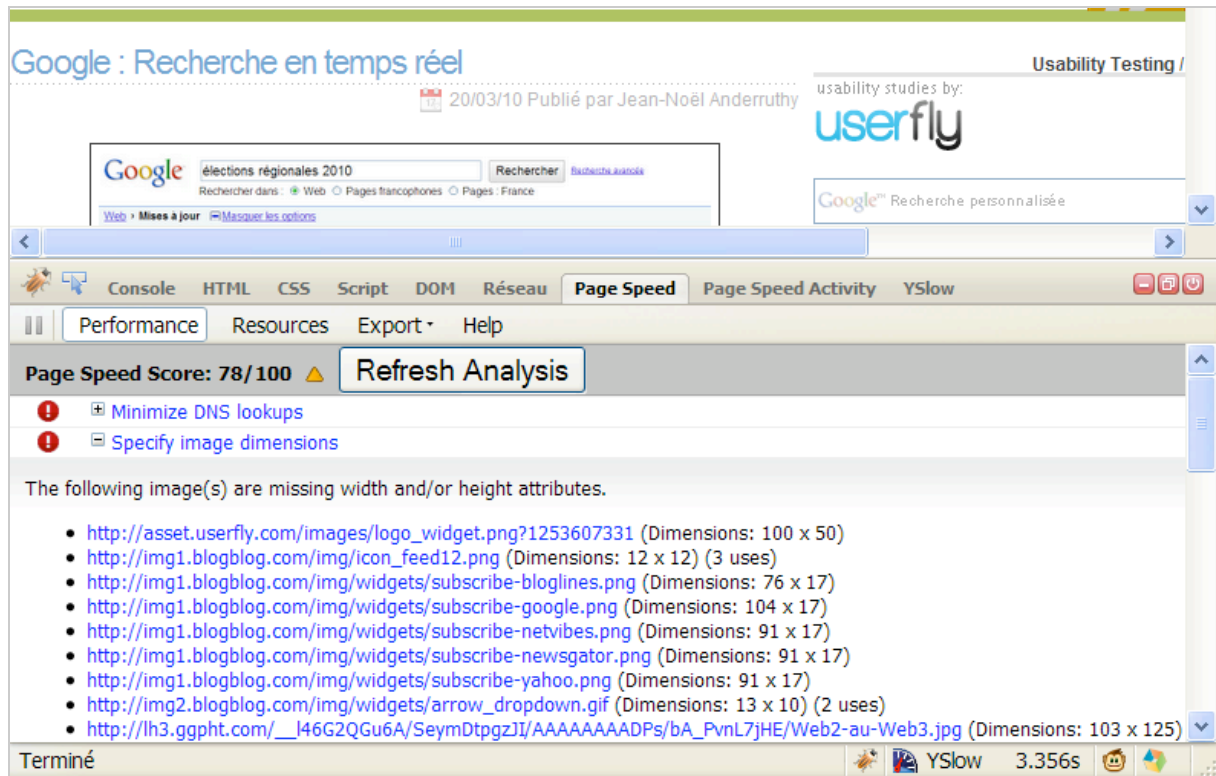
URL
 /2009/09/google-chrome-frame.html + Détails : économisez jusqu'à 9,17 Ko , 5 requêtes, 5 résolutions DNS
 /2009/05/le-plan-cadastral-avec-google-earth.html + Détails : économisez jusqu'à 951 octets , 4 requêtes, 5 résolutions DNS
 /2008/11/videos-youtube-en-tres-haute-qualite.html + Détails : économisez jusqu'à 1,05 Ko , 4 requêtes, 4 résolutions DNS
 /2009/03/dessiner-avec-google-docs.html + Détails : économisez jusqu'à 9,17 Ko , 5 requêtes, 5 résolutions DNS
 /2009/08/backtweets-en-flux-rss-avec-yahoo-pipes.html + Détails : économisez jusqu'à 9,02 Ko , 7 requêtes, 4 résolutions DNS
 /2007/10/creer-votre-profil-dans-google-maps.html + Détails : économisez jusqu'à 951 octets , 4 requêtes, 5 résolutions DNS
 /2008/07/api-arcgis-pour-google-maps.html + Détails : économisez jusqu'à 1,51 Ko , 4 requêtes, 5 résolutions DNS
 /2008/06/l-api-panoramio-avec-google-maps.html + Détails : économisez jusqu'à 19,9 Ko , 4 requêtes, 5 résolutions DNS
 /2008/06/google-map-marker.html + Détails : économisez jusqu'à 1,05 Ko , 4 requêtes, 5 résolutions DNS

La page renvoie directement vers un module pour Firefox appelé Page Speed...

Page Speed

Accédez à cette adresse : <http://code.google.com/speed/page-speed/download.html> puis cliquez sur le bouton permettant d'installer la dernière version de Page Speed. Attention, cette extension en requiert une autre appelée **Firebug**, que vous pouvez télécharger à partir de cette page : <https://addons.mozilla.org/fr/firefox/addon/1843>. Ouvrez la page web sur laquelle va porter votre test...

1. Cliquez sur *Outils – Firebug – Ouvrir Firebug*.
 2. Cliquez sur l'onglet *Page Speed* puis sur le bouton *Analyse Performance*.
- Nous sommes déjà dans le vif du sujet puisque Page Speed :
- Liste les problèmes ou sources d'amélioration possibles ;
 - Suggère les modifications à apporter ;
 - Génère des versions optimisées de certains de vos fichiers.



Cliquez, par exemple, sur le lien *Optimized version* afin d'enregistrer la version compressée d'une image ou d'un fichier JavaScript.

Sous Windows, la nouvelle version du fichier sera stockée dans ce type d'arborescence :
`\Documents and Settings\"Nom Utilisateur\"Local Settings\Temp\page-speed-[css | javascript | images].`

Différentes icônes sont associées au niveau de priorité de chacune des alertes :

- Haute (avec un impact important sur les performances de votre site) ;
- Moyenne (avec des conséquences relativement majeures) ;
- Fonctionnement correct avec, tout de même, des possibilités d'amélioration ;
- Informations diverses.

Rappelez-vous que, mis à part les leviers d'importance haute, une optimisation poussée d'une page web passe par une succession d'amélioration qui, au final, diminueront de manière conséquente le temps de chargement (et de rendu) de vos pages.

Il est possible d'analyser une page en particulier en cliquant sur l'onglet *Page Speed Activity*. Cliquez ensuite sur le bouton *Record Activity*.

Cliquez sur l'onglet *Ressources* afin d'afficher le détail de chacun des éléments analysés, l'exporter au format JSON ou le publier directement sur ce site : <http://www.showslow.com/>. Nous allons examiner rapidement un autre de ces outils...

YSlow

Yslow, utilitaire proposé par Yahoo!, est une extension Firefox que vous pouvez installer à partir de cette adresse : <https://addons.mozilla.org/en-US/firefox/addon/5369>. Là encore, ce module supplémentaire nécessite que Firebug soit déjà présent.

L'ensemble des bonnes pratiques édictées par Yahoo! sont visibles sur cette page :

http://developer.yahoo.com/performance/rules.html#min_dom.

1. De nouveau, ouvrez Firebug puis cliquez sur l'onglet *YSlow*.
2. Accédez à une page web puis cliquez sur le bouton *Run test*.

Différents onglets seront visibles :

- *Grade* : indique la performance globale de la page avec, pour chacune des catégories, une note allant de A à F ;

- *Components* : liste les différents éléments qui composent la page web ;
- *Statistics* : affiche le poids des requêtes HTTP (sans qu'aucun élément n'ait été mis en cache) ainsi que l'importance respective de chacune des catégories d'éléments qui composent la page ;
- *Tools* : offre un certain nombre d'outils permettant de mettre directement les mains dans le cambouis...

Grade D Overall performance score 67 Ruleset applied: YSlow(V2) URL: http://googlexl.blogspot.com/

ALL (22) FILTER BY: **CONTENT (6)** | **COOKIE (2)** | **CSS (6)** | **IMAGES (2)** | **JAVASCRIPT (4)** | **SERVER (5)**

F Make fewer HTTP requests
F Use a Content Delivery Network (CDN)
F Add Expires headers
A Compress components with gzip
A Put CSS at top
A Put JavaScript at bottom
A Avoid CSS expressions
n/a Make JavaScript and CSS external
F Reduce DNS lookups
F Minify JavaScript and CSS
A Avoid URL redirects
A Remove duplicate JavaScript and CSS

Grade F on Use a Content Delivery Network (CDN)

There are 50 static components that are not on CDN.

You can specify CDN hostnames in your preferences. See [YSlow FAQ](#) for details.

- http://www.blogger.com/static/v1/widgets/1892504587-widget_css_bundle.css
- <http://www.blogger.com/dyn-css/authorization.css?...>
- <http://feeds.feedburner.com/~s/GoogleXxl?...>
- <http://feeds.feedburner.com/~s/GoogleXxl?...>
- http://pagead2.googlesyndication.com/pagead/show_ads.js
- http://pagead2.googlesyndication.com/pagead/expansion_embed.js
- http://googleads.g.doubleclick.net/pagead/test_domain.js
- <http://asset.userfly.com/users/40795/userfly.js>
- <http://www.google.com/coop/cse/brand?...>
- <http://www.google.com/friendconnect/script/friendconnect.js>

Les deux graphiques (onglet *Statistics*) effectuent une comparaison entre le nombre d'éléments téléchargés quand le cache de votre navigateur est vide et ce même nombre dès que le navigateur a mis les premiers éléments en cache.

Statistics The page has a total of **58** HTTP requests and a total weight of **1053.1K** bytes with empty cache

WEIGHT GRAPHS

Empty Cache	HTTP Requests - 58	Total Weight - 1053.1K	Primed Cache	HTTP Requests - 43	Total Weight - 958.1K
	2 HTML/Text	18.9K		2 HTML/Text	18.9K
	13 JavaScript File	103.0K		9 JavaScript File	30.7K
	2 Stylesheet File	2.7K		1 Stylesheet File	0.02K
	3 IFrame	12.2K		3 IFrame	12.2K
	1 Flash Object	1.0K		1 Flash Object	1.0K
	11 CSS Image	8.0K		9 CSS Image	0.3K
	22 Image	899.8K		14 Image	887.5K
	1 Favicon	3.6K		1 Favicon	3.6K
	1 XMLHttpRequest	2.9K		1 XMLHttpRequest	2.9K
	1 Redirect	0.6K		1 Redirect	0.6K

Copyright © 2009 Yahoo! Inc. All rights reserved.

Trois outils vous permettent d'être directement opérationnel :

- *YUI CSS Compressor* : génère une version optimisée de votre fichier de feuille de style ;
- *All Smush.it™* : compresse automatiquement l'ensemble des images présentes sur votre site ;

- *Printable View* : affiche un récapitulatif "User Friendly" des différents problèmes qui ont été détectés.

D'autres outils en ligne

Une liste complète d'outils Google — ou provenant d'autres développeurs — est disponible à cette adresse : <http://code.google.com/intl/fr-FR/speed/tools.html>.

Il existe de nombreuses applications permettant d'optimiser le code HTML de vos pages dont **HTML Tidy** (<http://www.w3.org/People/Raggett/tidy/>). Mais, le moins que l'on puisse dire, est qu'il n'est pas d'un maniement facile...

Diminuer le nombre de Ko téléchargés

Le but ici est de limiter la bande passante afin de faciliter le chargement des pages et, éventuellement, de diminuer ses coûts d'hébergement. Une arme imparable : la compression...

Selon le même principe des fichiers présents sur votre disque (WinZip, WinRar, etc.), nous allons activer la compression sur un serveur Apache en utilisant le module *Deflate* ou *Gzip*.

Il existe différentes versions :

- Apache 2.0 : module `mod_deflate` utilisant `zlib` ;
- Apache 1.3 : `mod_gzip` ou `mod_deflate`.

1. Après avoir vérifié que le module *deflate* est bien activé, éditez le fichier `.htaccess`.
2. Ajoutez les lignes de commande voulues en fonction des extensions de fichiers pour lesquelles vous souhaitez activer la compression :

`SetOutputFilter DEFLATE`

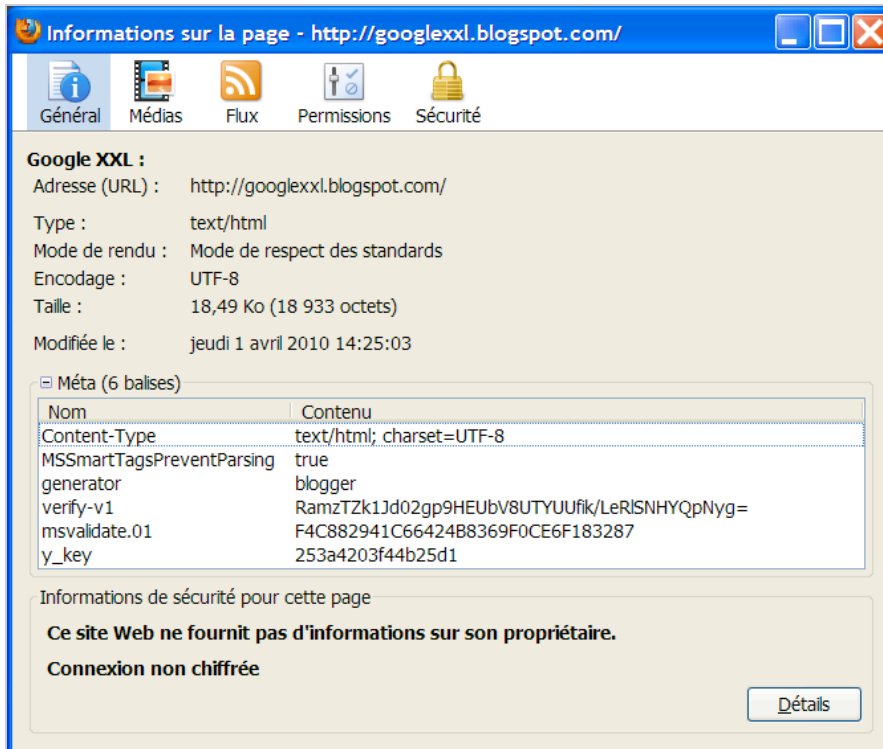
`AddOutputFilterByType DEFLATE text/plain text/xml text/html text/css image/svg+xml application/xhtml+xml application/xml application/rss+xml application/atom_xml application/x-javascript application/x-httpd-php application/x-httpd-fastphp application/x-httpd-eruby`

La compression fonctionne pour les éléments texte (HTML, XML JSON, CSS et fichiers de script).

Il est inutile de compresser les images, les fichiers PDF ou les vidéos.

Notez aussi que les fichiers de moins de 2 Ko ne nécessitent aucune compression.

Afin de vérifier les progrès réalisés, affichez dans Firefox les informations de la page. Vous constaterez une différence entre le poids de la page et celle indiquée...



- Si vous possédez la version 1.3 d'Apache, lisez le tutoriel complémentaire qui est visible à cette adresse : <http://www.alsacreations.com/article/lire/914-compression-pages-html-css-gzip-deflate.html> ;

- Pour un serveur IIS, les explications nécessaires sont données à cette adresse : <http://www.technos-sources.com/tutorial-activer-compression-qzip-dans-iis6-1.aspx>.

Activer le cache du navigateur

Avant toute chose, il faut signaler l'existence de cette étude de Yahoo! (<http://yuiblog.com/blog/2007/01/04/performance-research-part-2/>) qui indiquait qu'entre 40 et 60% des internautes viennent sur votre site avec un cache qui est vierge de toutes informations.

Les serveurs Apache gèrent déjà la mise en cache : à chaque fois qu'un élément est appelé, une requête "Get conditionnelle" est envoyée. Quand le navigateur charge une page et qu'il voit que l'élément est déjà présent dans le cache, il va faire parvenir une requête au serveur en lui demandant si le fichier a été modifié ou non. Si ce n'est pas le cas, le serveur renvoie une requête http 304 en lui indiquant que la version en cache est toujours d' "actualité".

En apparence, le système semble au point mais, dans la réalité, ce fonctionnement génère un grand nombre de requêtes inutiles. Prenons le cas d'un internaute qui vient régulièrement sur votre site. Lors de chacune de ses visites, le navigateur va interroger le serveur pour vérifier si l'élément contrôlé a été modifié depuis son dernier passage.

Une solution consiste alors à indiquer les éléments qui restent, pour ainsi dire, intangibles et notamment, les feuilles de style, les fichiers de script et les images. Là encore, il suffit d'ajouter ces lignes dans le fichier .htaccess :

```
# 1 mois
<FilesMatch "\.(ico|jpg|jpeg|png|gif|swf|js|css|png)$">
Header set Cache-Control "max-age=2592000, public"
</FilesMatch>
# 1 jour
<FilesMatch "\.(html|htm)$">
Header set Cache-Control "max-age=43200, public"
```

</FilesMatch>

La valeur indiquée pour *max-age* est simplement le nombre de secondes que contient un jour ou un mois.

Pour ceux qui souhaitent approfondir cette question, vous pouvez consulter la documentation officielle d'Apache : <http://httpd.apache.org/docs/2.1/fr/caching.html>.

Attention ! Cette méthode vous oblige, quand vous changerez une image, par exemple, à renommer cet élément afin que le navigateur soit dans l'obligation d'en télécharger la dernière version.

Vous pouvez aussi vous servir des en-têtes d'expiration HTML (*Last-Modified* et *ETag*) afin d'orienter le navigateur à bon escient. Une explication complète est visible à cette adresse : <http://code.google.com/intl/fr/speed/articles/caching.html>.

Afin d'en avoir une démonstration en temps réel, commencez, tout d'abord, à télécharger **HttpWatch** à partir de cette adresse : <http://www.httpwatch.com/download/>. Une fois cet outil installé, accédez deux fois de suite à chacune de ces deux pages : <http://stevesouders.com/hpws/expirsoff.php> et <http://stevesouders.com/hpws/expirson.php>. Le test porte sur la vitesse d'affichage des 6 étoiles rouges. Cliquez sur le lien *Click Here* afin d'afficher le temps de réponse enregistré pour chacune des actualisations.

The screenshot shows a browser window with the title 'Far Future Expires'. The main content area has a teal header with the text 'Rule 3 - Add an Expires Header' and a navigation menu with links 'Rules 1 2 3 4 5 6 7 8 9 10 11 12 13 14', where '3' is highlighted. Below the header is a section titled 'Example 2 - Far Future Expires' with a sub-header 'Examples 1 2', where '2' is highlighted. The text in this section reads: 'Roundtrip response time: 581 ms [try it again](#)'. It continues: 'This example contains six images, three scripts and one stylesheet that do have a far future Expires header. Load the page several times using the "try it again" link and compare the response time to Example 1 - No Expires. This example is faster because it uses the cache thus avoiding ten additional HTTP requests.' Below this text are six orange stars. At the bottom, there is a footer with a dog icon and the text: 'This is the companion website for [High Performance Web Sites](#) by [stevesouders.com](#) | [contact Steve](#) Steve Souders.'

Cet autre site est une référence sur le sujet : <http://www.askapache.com/htaccess/speed-up-sites-with-htaccess-caching.html>.

Activer le pré-chargement des pages

Cette technique utilise le temps que passe l'internaute sur une page précédente pour charger les éléments de la page suivante. Imaginons que vous accédez à la page d'accueil de Google. Il va y avoir un laps de temps avant que vous ayez terminé de saisir votre requête et lancer votre recherche. Ce délai va donc être utilisé par le moteur de recherche pour mettre en cache, par exemple, les éléments graphiques qui composent la page suite (la page des résultats). Tout le processus va se dérouler en tâche de fond et de manière complètement transparente pour les utilisateurs. C'est donc une manière de combler les trous et d'utiliser au mieux les "temps morts".

Un bon exemple d'application est celui proposé par le Framework Mootools (utilisé pour le développement d'applications "Rich Internet"). Une démonstration est visible à cette adresse : <http://demos111.mootools.net/Asset.images>.

Utiliser des serveurs tierces partie

Google permet aux webmasters d'utiliser une série de bibliothèques hébergées sur ses propres serveurs : <http://code.google.com/apis/ajaxlibs/>. Il existe plusieurs avantages à utiliser cette méthode :

- Les serveurs de Google ont de bonnes chances d'être beaucoup plus rapides que les vôtres ;
- Les internautes se verront servir le script par le serveur qui est le plus proche de leur localisation géographique ;
- Les visiteurs qui ont déjà accédé à d'autres sites web ont les scripts correspondants sur leur disque dur ;
- Vous économisez une part non négligeable de bande passante.

Installer le code Google Analytics asynchrone

Le code asynchrone de Google Analytics génère dynamiquement une balise `<script>` qui s'occupera des tâches de tracking. La conséquence directe est que le chargement de la page sera indépendant de celui du code de tracking. Voici l'adresse de documentation :

<http://code.google.com/apis/analytics/docs/tracking/asyncTracking.html> ainsi que des exemples de migration : <http://code.google.com/apis/analytics/docs/tracking/asyncMigrationExamples.html>.

Les feuilles de style CSS

Une des techniques les plus courantes est de compacter les fichiers CSS (mais aussi les fichiers JavaScript). Dans le cadre d'un fichier de script, l'emploi combiné d'un programme de compactage et d'une compression (Delate ou GZip) peut représenter un gain de 80% par rapport à la version originale.

Placer les styles dans le <head>

Il faut tout d'abord comprendre comment se déroule le chargement d'une page HTML. Les éléments HTML sont les premiers à être téléchargés. À chaque appel vers une feuille de style, la page sera automatiquement rafraîchie. Si nous possédions de bons yeux, nous pourrions nous apercevoir du phénomène de scintillement qui en est la conséquence directe.

Utiliser une feuille de style externe

L'idéal est d'externaliser vos feuilles de style... Mais nous pouvons aussi vouloir combiner les différentes feuilles de style en un seul fichier, et ce afin de diminuer le nombre de requêtes HTTP :

Avant :

```
<link rel="stylesheet" type="text/css" href="contenu.css" />
<link rel="stylesheet" type="text/css" href="societe.css" />
<link rel="stylesheet" type="text/css" href="contact.css" />
```

Après :

```
<link rel="stylesheet" type="text/css" href="tout.css" />
```

Par ailleurs, vous devez placer votre appel le plus au début de votre page HTML :

```
<html>
  <head>
    <title>CSS</title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
```

```
</head>  
</html>
```

Utiliser Link au lieu de @import

Il existe deux solutions permettant d'appeler une feuille de style CSS : la balise Link ou la balise @import. La seconde souffre de deux principaux inconvénients par rapport à Link :

- Moindre rapidité ;
- Ordre de chargement aléatoire.

Utiliser les propriétés sténographiques

L'astuce consiste à grouper les mêmes propriétés CSS. Voici un exemple de bonne pratique :

```
/* MARGE */  
h1 {margin:1.5em 0 0.2em 0.4em;}  
  h1 {margin-top:1.5em;  
      margin-right:0;  
      margin-bottom:0.2em;  
      margin-left:0.4em;  
  }  
/* BORDURE */  
h1 {border:1px solid #000;}  
h1 {border-width:1px;  
    border-style:solid;  
    border-color:#000;  
}
```

Cette page offre de nombreux exemples de mise en pratique : <http://www.eric-couchelou.net/web-design/compacter-ses-css-grace-aux-proprietes-stenographiques-shorthands/>.

Grouper les styles similaires

C'est souvent ce qu'il y a de plus compliqué à mettre en place mais, dans la plupart des cas, vous vous apercevrez que vous utilisez les mêmes styles pour des éléments différents.

Avant :

```
h1 {padding:4px 0; font-family:verdana; font-weight:100;}  
#box1 .heading {padding:4px 0; font-family:verdana; font-weight:100;}  
#box2 .heading {padding:4px 0; font-family:verdana; font-weight:100;}
```

Après :

```
h1, #box1 .heading, #box2 .heading {padding:4px 0; font-family:verdana; font-weight:100;}
```

Réduire la taille des fichiers CSS

Le principe ici est de supprimer l'ensemble des éléments inutiles : retour chariot, dernier point-virgule, commentaires, codages des couleurs (3 caractères au lieu de 6), indication des pixels, etc.

Avant :

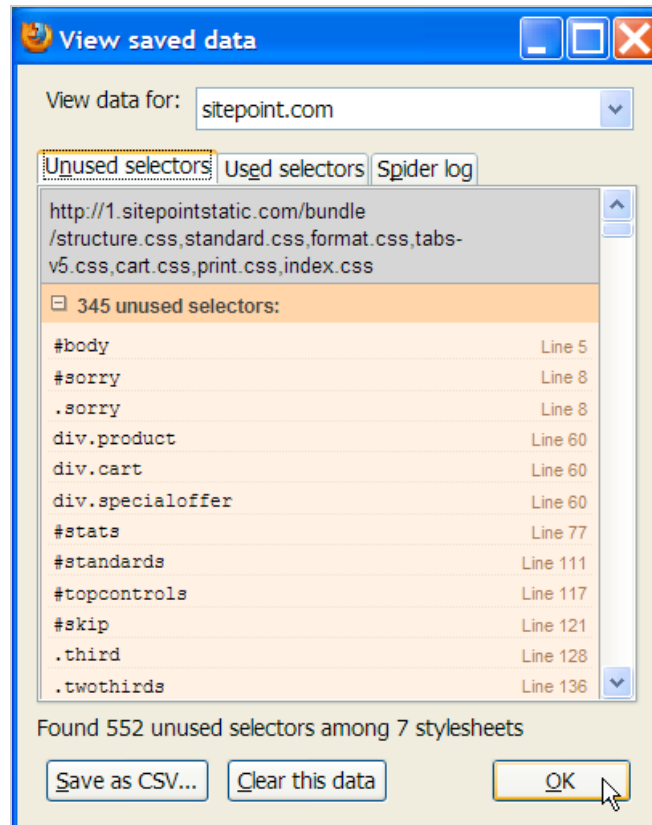
```
/* Titres H3 */  
H3 {  
  font-family:verdana;
```

```
padding:0px;  
color: #112233;  
text-decoration:underline;  
}
```

Après :

```
H3 {font-family:verdana;padding:0;color: #123;text-decoration:underline}
```

Une extension Firefox permet de détecter quelles sont les classes de style qui ne sont pas utilisées : <http://www.sitepoint.com/dustmeselectors/>.



Utiliser la compression CSS

Il existe une multitude d'outils en ligne qui permettent d'optimiser, de nettoyer et de compresser un fichier CSS :

CleanCSS : <http://www.cleancss.com/> ;

CSS Optimizer : <http://www.cssoptimiser.com/> ;

Flumpcakes : <http://flumpcakes.co.uk/css/optimiser/> ;

OrganizeCSS : <http://www.styleneat.com/> ;

CSSCompressor : <http://www.cssdrive.com/index.php/main/csscompressor/> ;

YUI Compressor : <http://www.refresh-sf.com/yui/>.

Champ CSS:

CSS-Code:

```
form { padding: 0px; margin: 0px; }

ul#onglet {
  background-color: #ffffff;
  font: bold 10px verdana, arial,
sans-serif;
  list-style-type: none;
  padding-bottom: 24px;
  border-bottom: 1px solid #aaaaaa;
  margin: 0;
}

ul#onglet li {
  float: left;
  height: 21px;
  background-color: #ffffff;
  margin: 2px 2px 0 2px;
  border: 1px solid #aaaaaa;
}

ul#onglet li.active {
  border-bottom: 1px solid #eeeeee;
  background-color: #eeeeee;
}
```

CSS en provenance d'une URL:

Mise en page du code:

Compression (Mise en page du code):
Normale (équilibre entre la lisibilité et la taille)

Sur mesure [Gabarit](#)

Options

- Trier les sélecteurs (attention)
- Trier les propriétés
- Fusionner les sélecteurs avec les mêmes propriétés
0 1 2 (?)
- Fusionner les propriétés (sténographie)
- Seulement les optimisations sûres
- Compresser les couleurs
- Compresser font-weight
- Sélecteurs en minuscules
- Cases (Minuscules & Majuscule) pour les propriétés:
 Minuscules Majuscule
- enlever les antislashes inutiles
- convertir !important-hack
- Enlever le dernier ;
- Sauvegarder les commentaires
- Supprimer les propriétés non valide CSS2.1
- Sauver en tant que fichier**

Messages

- 3 Optimised color: Changed "#ffffff" to "#fff"
- 7 Optimised color: Changed "#000055" to "#005"
- 9 Optimised font-weight: Changed "normal" to "400"
- 14 Optimised number: Changed "0px" to "0"
- 17 Optimised color: Changed "#ffffff" to "#fff"
- 21 Optimised color: Changed "#aaaaaa" to "#aaa"

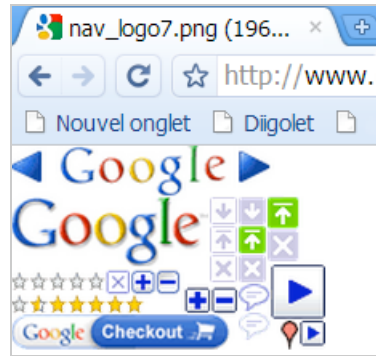
Entrée: 17.541KB, Sortie: 11.222KB, Facteur de Compression: 36% (-6319 Bytes)

```
<!-- body {
background:#fff;
font-family:Verdana,Helvetica;
```

Les "Sprites CSS"

Il arrive souvent qu'un site web repose sur un "Template" composé de plusieurs dizaines d'images : boutons, logo, menus, puces, etc. À chaque chargement de page, le navigateur doit donc procéder à autant de requête HTTP pour vérifier la validité de ces différents composants. Le principe d'un *Sprite* CSS est de regrouper cette multitude de fichiers (de petites images) dans une seule image.

Les poids lourds du Web utilisent cette technique dont, par exemple, Google : http://www.google.com/images/nav_logo7.png.



Le navigateur n'aura donc plus qu'une seule image à télécharger (dans l'exemple qui suit, [sprite.png](#)).

La première étape consiste à créer votre fichier contenant les différentes images séparées par une bordure de 1 pixel. La bordure n'est pas indispensable mais va rendre les choses beaucoup plus simples pour la suite.

Il faut ensuite créer une image GIF transparente faisant 1 pixel de côté ([transparent.gif](#)). Nous allons ensuite définir une classe appelée "sprite" en utilisant cette déclaration :

```
.sprite {background:url(..images/sprite.png);}
```

Une fois cette étape franchie – et puisque les images ont toutes la même hauteur –, nous allons créer une classe appelée "icônes".

```
sprite {background:url(..images/sprite.png);}  
.icônes {height:70px;}
```

Si la hauteur de nos images est, à chaque fois, la même, il n'en va forcément de même pour leur largeur. De fait, il faut alors créer une classe CSS pour chacun d'eux :

```
.sprite {background:url(..images/sprite.png);}  
.icônes {height:70px;}  
/* Icônes */  
.rss {width:60px;}  
.facebook {width:60px;}
```

Nous allons maintenant assigner une position à chacune de ces images. Le principe consiste à utiliser des valeurs négatives puisqu'au coin supérieur droit de l'image "parente" correspondra ces deux valeurs : x=0px (abscisse ou axe horizontal) et y=0px (ordonnée ou axe vertical). De fait, nous allons déplacer à chaque fois notre image d'arrière-plan afin d'afficher les différentes images dont nous avons besoin.

Par exemple, notre bouton nommé "rss" doit être déplacé de 196 pixels sur la gauche et de 2 pixels vers le haut (1 + 1).



← 196 pixels

```
.sprite {background:url(..images/sprite.png);}  
.icônes {height:70px;}  
/* Icônes */  
.rss {width:60px; background-position:-196px -2px;}  
.facebook {width:60px;}
```


Procédez de la même façon pour les autres images...

Il ne reste plus qu'à les inclure dans le code HTML en utilisant ce type de syntaxe :

```


```

Pour un exemple complet de mise en œuvre, n'hésitez à lire cette page :

<http://www.websiteoptimization.com/speed/tweak/css-sprites/>. Ce site propose aussi des démonstrations qui sont très parlantes : <http://css-tricks.com/css-sprites/>.

Signalons qu'il existe des "Sprites Generator" (<http://fr.spritegen.website-performance.org/> ou <http://www.blogsdna.com/813/top-6-best-online-and-offline-css-sprites-generator.htm>) mais, à notre avis, les services rendus par ce type d'applications sont assez limités.

L'utilisation des "Sprites" peut rendre les tâches de maintenance d'un site particulièrement compliquées. Vous devez mettre en place des procédures précises afin que les différents intervenants apportent des modifications au site en toute connaissance de cause.

Enfin, nous pouvons nous interroger sur l'utilisation des "Mega Sprites" : dans l'optique d'un bon référencement, seules les images fonctionnelles doivent être placées en arrière-plan. Celles qui sont importantes, et qui montrent vos produits ou illustrent l'excellence de vos services, doivent faire partie des éléments "inline" de la page web.

Les fichiers JavaScript

La position des JavaScript est important et il est conseillé de les placer le plus possible en bas de page et juste avant la balise de fermeture </body>.

Imaginons une page HTML qui comporte un certain nombre d'images importantes et du code JavaScript qui est appelé avant. Le chargement des images sera conditionné à celui du code JavaScript. Pourquoi ? Pendant le chargement d'un script, le téléchargement en parallèle est bloqué du fait que le code peut modifier les éléments et l'aspect de la page correspondante. Par ailleurs, cette méthode permet de contrôler l'ordre de chargement des scripts.

Selon un principe similaire, pensez à charger vos fichiers "Inline" après les scripts hébergés sur votre serveur.

En résumé, les scripts externes bloquent le chargement des éléments qui sont placés en-dessous d'eux tandis que les fichiers "Inline" empêchent le chargement de toute la page.

Accédez à cette page :

<http://stevesouders.com/cuzillion/?ex=10100&title=Inline+Scripts+Block> afin d'en avoir une démonstration de visu.

Pour information, un script "Inline" est simplement du code qui se retrouvera au milieu d'autres instructions HTML ou CSS. Par exemple :

```
<div style="width: 400px; margin: 1em auto;">
  <div style="float: left; width: 400px; padding: 1em 2em;">
    <span id="actus" onclick="Fonction()">Actualités</span>
  </div>
</div>
```

Afin d'illustrer son propos, un blogueur s'est amusé à réécrire la page d'accueil de Google en mettant en pratique cette démarche : <http://www.456bereastreet.com/lab/google/>.

Compresser le JavaScript

Il existe des outils qui permettent la compression de fichiers JavaScript :

Shrinksafe : <http://o.dojotoolkit.org/docs/shrinksafe> ;

Paker : <http://dean.edwards.name/packer/> ;

JavaScript Minifer : <http://javascript.crockford.com/jsmin.html> ;

CloserCompiler : <http://closure-compiler.appspot.com/home>.

dean.edwards.name/packer/ my | weblog | about | contact | search

A JavaScript Compressor. version 3.0

Paste: [Help](#)

```
mid=3c65ce5f8978be90f9e07f16f8aa5573&l=fr&collid=10000&l=fr&sortBydate=1&mot="+definition;
}
}

function bookmarksite(title, url){
if (document.all)
window.external.AddFavorite(url, title);
else if (window.sidebar)
window.sidebar.addPanel(title, url, "")
}
}
```

Base62 encode
Shrink variables

Copy:

```
function show_props(obj,obj_name){var result="";for(var i in obj)result+=obj_name+"."+i+" = "+obj[i]+"\n";return result;}function
Semaine() {this[0]="Dimanche";this[1]="Lundi";this[2]="Mardi";this[3]="Mercredi";this[4]="Jeudi";this[5]="Vendredi";this[6]="Samedi"}fu
nction
Mois() {this[0]="01";this[1]="02";this[2]="03";this[3]="04";this[4]="05";this[5]="06";this[6]="07";this[7]="08";this[8]="09";this[9]="1
0";this[10]="11";this[11]="12"}function jourcourant(){var semaine=new Semaine();var mois=new Mois();var myDate=new Date();var
nav=navigator.appName.charAt(0);if(nav=="N") {document.writeln(semaine[myDate.getDay()]+
"+myDate.getDate()+"/"+mois[myDate.getMonth()]+"/"+(1900+myDate.getYear())} else {document.writeln(semaine[myDate.getDay()]+
"+myDate.getDate()+"/"+mois[myDate.getMonth()]+"/"+(myDate.getYear())} }function
verif(moteur) {definition=document.moteur.mot.value;methode=document.moteur.meth.value;web=document.moteur.ou[0].checked;synomia=docume
nt.moteur.ou[1].checked;if (!definition||definition==" ") {return alert("Merci d'entrer un mot
```

compression ratio: 1642/2006=0.819

Also available as [.NET](#), [perl](#) and [PHP](#) applications.

last modified: 2010/01/07

Par ailleurs, de nombreuses librairies Ajax sont disponibles en version compacte...

Il faut noter que vous pouvez vérifier la qualité de votre code en utilisant ce service en ligne : <http://www.jslint.com/>.

De nombreuses autres pistes sont proposées sur cette page : <http://javascript-reference.info/speed-up-your-javascript-load-time.htm> (optimisation de l'AJAX, ThinkVitamin, etc.).

Grouper les fichiers JavaScript

Cette technique est exposée sur cette page web : <http://rakaz.nl/2006/12/make-your-pages-load-faster-by-combining-and-compressing-javascript-and-css-files.html>.

Imaginons que vous ayez ces lignes de code :

```
http://www.exemple.fr/javascript/contenu.js
http://www.exemple.fr/javascript/contact.js
http://www.exemple.fr/javascript/effets.js
```

L'appel à ces trois fichiers de script se fera à l'aide d'une seule ligne :

```
http://www.exemple.fr/javascript/contenu.js, contact.js, effets.js.
```

Le script PHP est téléchargeable à cette adresse :

```
http://rakaz.nl/projects/combine/combine.phps.
```

Il faut :

Adapter les chemins et les noms des fichiers qui sont appelés ;

Créer sur votre serveur un répertoire cache ;

Éditer le fichier `.htaccess` afin de rediriger les appels vers les fichiers de script (et, éventuellement, les fichiers CSS) :

```
RewriteEngine On
RewriteBase /
RewriteRule ^css/(.*\.css) /combine.php?type=css&files=$1
```

RewriteRule ^javascript/(.*\.js) /combine.php?type=javascript&files=\$1

Les images

Un premier principe consiste à enregistrer directement les images à la taille voulue. Ainsi, vous n'aurez pas à les redimensionner en HTML. Par ailleurs, spécifiez la taille de chacune des images afin que le navigateur n'ait pas besoin d'ajuster le contenu après coup. Par exemple :

```
<IMG SRC="image.png" height="100" width="100" / >
```

Les fichiers JPEG et GIF sont moins volumineux que les fichiers BMP ; les fichiers PNG sont moins gourmands que les fichiers JPEG.

Le format PNG

Les images PNG peuvent être de 5 codages différents : Grayscale, Couleurs vraies (TrueColor), Couleurs indexées (Indexed-color), Grayscale avec transparence alpha et Couleurs vraies avec transparence alpha. Il existe deux sous-catégories : bit avec transparence (chaque pixel peut être soit opaque soit transparent) et palette avec transparence (chaque pixel peut être semi-transparent).

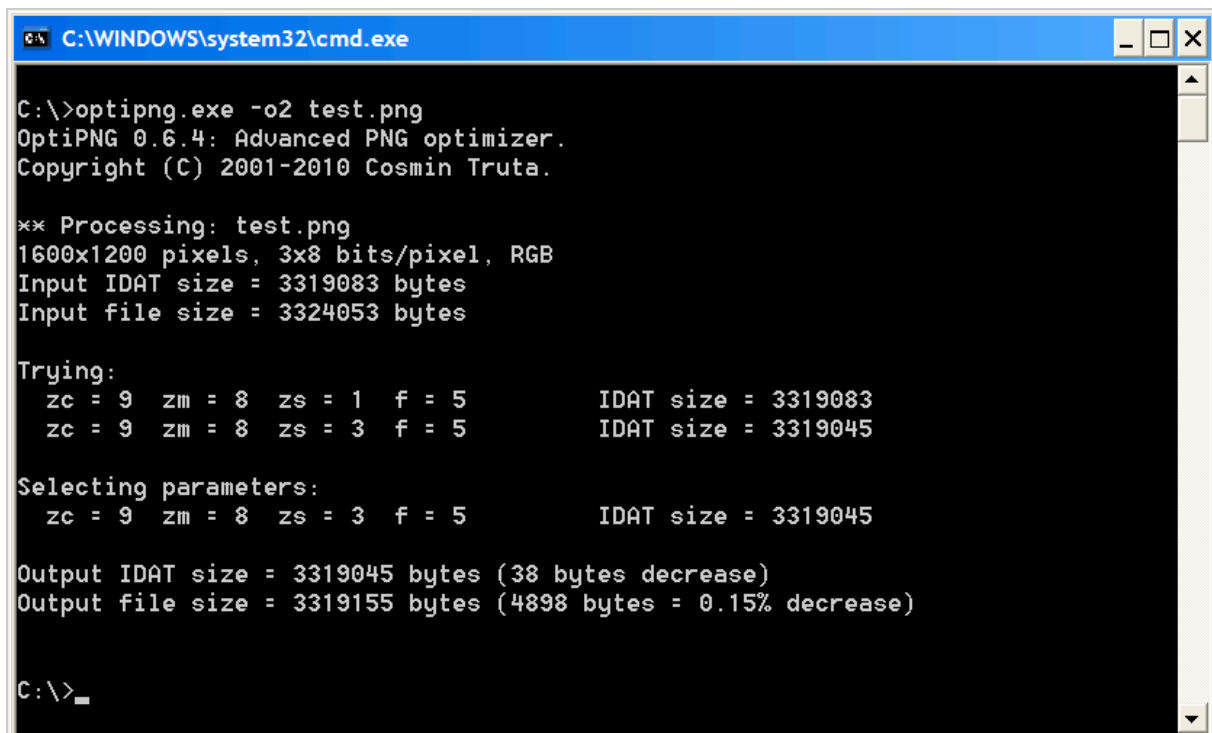
Deux outils (en Invite de commandes !) vous permettent d'effectuer une compression avancée des images au format PNG :

<http://optipng.sourceforge.net/> ;

<http://advancemame.sourceforge.net/comp-readme.html>.

Ils effectueront les opérations suivantes :

- Choix du meilleur type de codage en fonction de votre image ;
- Application du meilleur filtre de transparence ;
- Choix de la meilleure méthode de compression à adopter et de la réduction de la profondeur des couleurs.



```
C:\WINDOWS\system32\cmd.exe

C:\>optipng.exe -o2 test.png
OptiPNG 0.6.4: Advanced PNG optimizer.
Copyright (C) 2001-2010 Cosmin Truta.

** Processing: test.png
1600x1200 pixels, 3x8 bits/pixel, RGB
Input IDAT size = 3319083 bytes
Input file size = 3324053 bytes

Trying:
  zc = 9  zm = 8  zs = 1  f = 5          IDAT size = 3319083
  zc = 9  zm = 8  zs = 3  f = 5          IDAT size = 3319045

Selecting parameters:
  zc = 9  zm = 8  zs = 3  f = 5          IDAT size = 3319045

Output IDAT size = 3319045 bytes (38 bytes decrease)
Output file size = 3319155 bytes (4898 bytes = 0.15% decrease)

C:\>_
```

Des techniques plus avancées sont expliquées sur cette page :

<http://www.smashingmagazine.com/2009/07/25/png-optimization-guide-more-clever-techniques/> :

- Postérisation des images ;
 - Suppression ou modification de la transparence ;
 - Masquage de zone, etc.
- Pour les "Photoshopeurs", cette vidéo en fait un tour d'horizon assez exhaustif : <http://vimeo.com/5685903>.

Le format JPG

Jpegtran est un outil – d'Invite de commandes – qui permet de compresser les images au format JPEG sans perte de qualité : <http://sylvana.net/jpegcrop/jpegtran/>. L'application va retirer l'ensemble des métadonnées présentes dans l'image.

Des techniques beaucoup plus avancées sont expliquées à cette adresse : <http://www.smashingmagazine.com/2009/07/01/clever-jpeg-optimization-techniques/>.

Booster votre référencement en boostant vos pages web !

En fin de compte, le principal avantage de cette démarche est que les effets sont permanents : meilleur taux de conversion des visiteurs et plus forte fidélisation de ces mêmes visiteurs. C'est une chose de mettre en place des stratégies permettant d'augmenter son trafic mais elles supposent toujours deux contraintes :

- Un investissement important en temps ou/et en argent (Google AdWords, référencement naturel, etc.)
- Des effets qui ne seront jamais permanents sur le long terme.

Nous aurions tendance à dire alors que vouloir développer son trafic sans définir, en amont, des politiques efficaces en termes de performance revient, pour ainsi dire, à vouloir remplir un saut percé.

Pour terminer cet article, signalons ces deux excellentes sources sur le sujet :

- Le site de Steve Souders : une référence absolue en matière d'optimisation de site web : <http://stevesouders.com> ;
- Le blog français d'Éric Daspet qui, bien qu'actuellement en sommeil, offre de nombreux articles intéressants : <http://performance.survol.fr>.

Google a publié un manuel complet sur tous les vecteurs d'optimisation possibles : http://code.google.com/speed/page-speed/docs/rules_intro.html ainsi qu'un grand nombre de tutoriaux : <http://code.google.com/speed/articles/> (dont l'optimisation du Web mobile, des gadgets, etc.).

Bonne optimisation !

Jean-Noël Anderruthy, *webmaster spécialisé dans les technologies Google.*

Réagissez à cet article sur le blog des abonnés d'Abondance :
<http://blog-abonnes.abondance.com/2010/04/comment-ameliorer-le-temps-de.html>