Optimiser les images de son site web avec Littleutils

Gazette Linux n°119 — Octobre 2005

Brian Lindholm

Article paru dans le n°119 de la Gazette Linux d'octobre 2005.
Traduction française par Deny `<deny CHEZ monaco POINT net>`.
Relecture de la traduction française par Joëlle Cornavin `<jcornavi CHEZ club TIRET internet POINT fr>`.

Il y a de nombreuses années, je prenais beaucoup de photos au travail avec un appareil photo numérique dernier cri. Il fournissait des images au format JPEG. Je projetais de modifier les images pour corriger des problèmes de contraste quand je découvris que l'appareil avait beaucoup de mal à compresser les fichiers.

Grâce au logiciel standard fourni par le groupe JPEG indépendant (qui contient l'étonnant utilitaire jpegtran), je pus souvent réduire de 50 % la taille des fichiers. Sans perte de qualité. Comment cela fonctionne-t-il ? Une image JPEG standard emploie des paramètres d'encodage pour les sous-sections de l'image. Une image JPEG optimisée utilise ces paramètres pour l'image entière. Bien que l'algorithme de JPEG altère déjà le document, l'utilitaire jpegtran préserve l'« intégrité » de l'image JPEG originale de façon exacte. Le nouveau document est donc intact en ce qui concerne le fichier original.

Voici la commande que j'ai employée ici :

Malheureusement, j'avais pris plusieurs centaines de photos, et les traiter une par une à la main aurait pris un temps infini. Donc... j'ai écrit un de mes premiers scripts *shell* majeurs. Une fois que je l'eus rendu opérationnel, j'ai pu optimiser toutes les images en moins de dix minutes ! Le script opt-jpg était né.

Quelques semaines plus tard, j'ai découvert que les images JPEG progressives étaient parfois plus petites, mais pas toujours. Ainsi, j'ai modifié mon script pour l'essayer des deux manières, afin de conserver les images les plus réduites. Cela a fait l'objet d'un script plus astucieux, mais les résultats en valaient la peine. Le script opt-jpg a été amélioré.

Et même quelque temps plus tard, un regrettable incident avec des images BMP incorrectement nommées m'a conduit à ajouter un mécanisme de contrôle d'erreur, de façon à ce que le script ne modifie pas des fichiers non JPEG. Le programme opt-jpg devint plus puissant.

Un peu plus de temps m'a permis de créer un script d'optimisation pour GIF (axé sur gifsicle) et un autre pour PNG (axé sur pngcrush), appelés respectivement opt-gif et opt-png. Ces routines fonctionnent par réduction de table de couleur, changements de filtre et autres astuces de ce genre. Vous seriez stupéfait de la quantité d'images disponibles utilisant une table considérable de 256 couleurs pour n'employer que 3 ou 4 de ces couleurs. J'ai récemment conditionné tous ces scripts sous forme de paquetage et les ai publiés sous l'appellation littleutils (http://sourceforge.net/projects/miscutils/).

Bien que ma motivation originelle à écrire ces scripts concerne les appareils photo numériques poussifs, ils conviennent également pour l'optimisation de l'ensemble des graphiques d'un site *web*. Pourquoi améliorer vos graphiques ? Pour économiser de l'espace sur votre disque dur. Pour faire tenir davantage d'images sur votre site. Pour réduire la quantité de temps que mettent les visiteurs de votre site à charger vos pages. Les raisons sont évidentes.

Mais comment cela fonctionne-t-il ? Nous le démontrerons avec les pages *web* de la Gazette Linux© elle-même. Copions tout d'abord les fichiers du site *web* sur un disque dur local. La séquence de commandes suivante (sous bash) s'en charge :

Avant de commencer, nous devons déterminer l'espace que nécessitent nos images actuelles: :

Téléchargez ensuite et installez littleutils (http://sourceforge.net/projects/miscutils/). C'est une routine **./configure**, **make** et **make install** tout à fait classique. Une fois cette opération effectuée, nous pouvons optimiser les images :

Au bout d'un délai un peu long (l'optimisation PNG est particulièrement lente), les images seront entièrement optimisées. Si nous répétons les commandes tar, nous obtenons les résultats suivants (une amélioration de plus de 6 méga-octets !) :

De plus, si vous parcourez les résultats, vous constaterez que plusieurs fichiers sont incorrectement nommés. En particulier, il y avait beaucoup d'images GIF se faisant passer pour des images PNG. (Il semble que quelques personnes ici pensent que **mv image.gif image.png** est un moyen pratique pour convertir les fichiers images. Pas vraiment...) Il y avait même quelques images BMP Windows© paraissant être des images PNG. Une liste complète de ces fichiers se trouve ici : badfile.txt (outils/lg119-C/badfile.txt). Si ces fichiers sont correctement renommés et optimisés (ou mieux encore, correctement convertis et optimisés), vous pouvez économiser davantage d'espace disque.

> Merci d'avoir détecté ces problèmes pour nous, Brian ; ils sont maintenant résolus. J'ai quelque plaisir à noter que toutes les erreurs, à l'exception d'une classe, dataient d'avant mon arrivée à la rédaction de la Linux Gazette©. Mes propres erreurs — mea culpa — venaient du fait que convert ne change pas réellement le type de l'image si le fichier original présente une extension mal adaptée; ce script est également corrigé maintenant, avec les types d'images correctement affectés.
>
> — Ben

Bien que cet exemple montre clairement que l'on peut utiliser littleutils pour économiser un espace disque considérable, il reste deux mises en garde majeures :

[1]L'optimisation des images dans littleutils est agressive, toutes les informations externes étant éliminées, y compris les commentaires, les balises auxiliaires, les informations sur l'espace colorimétrique, etc. Vous devez effectuer quelques tests avant d'améliorer votre collection Photoshop entière.

[2] L'optimisation des images dans littleutils ne préserve pas l'entrelacement. L'entrelacement des images GIF et PNG sera toujours supprimé, et les images JPEG peuvent être converties en images progressives ou non progressives (selon ce qui est le plus léger). Si l'entrelacement est particulièrement

important pour vous, ignorz l'optimisation ou modifiez les scripts pour conserver l'entrelacement que vous souhaitez.

Cependant, pour la plupart des objectifs des sites *web*, les scripts d'optimisation faisant partie de littleutils fonctionnent parfaitement. Bonne optimisation !

Pour une optimisation plus poussée de votre site *web*, vous pourriez aussi envisager d'employer l'utilitaire repeats, faisant également partie de littleutils. Ce script astucieux trouvera les fichiers en double dans n'importe quelle arborescence de répertoires. S'il est lancé dans le répertoire de la Linux Gazette©, les fichiers dupliqués suivants seront trouvés : repeats.txt (outils/lg119-C/repeats.txt). Pour réduire plus encore l'espace disque requis, toutes les copies à l'exception d'une seule pourraient être supprimées et les références HTML des doublons supprimés pourraient pointer sur la copie restante.

Brian Lindholm, diplômé de technologie du Virginia Tech est également un ingénieur en mécanique confirmé qui a commencé à programmer en BASIC sur un TRS-80© Model I (en 1980). À la fin des années 80, il a opté pour le Pascal et le C sur un IBM© compatible PC.

Les années passant, Brian est devenu de plus en plus contrarié par l'instabilité et le coût des différents systèmes d'exploitation de Microsoft©. Il détestait en particulier de ne pas avoir le contrôle total de son système. Heureusement pour lui, cependant, il avait un compagnon de chambre à l'université qui utilisait Linux (l'époque de Linux 0.9 et de Slackware 1.0). Cette initiation fut tout ce qu'il lui fallait.

Au fil du temps, il en apprit de plus de plus, et il arrive à présent à maintenir avec bonheur la stabilité de son système Debian (en dépit de deux mises à jour majeures : des versions 2.2 à 3.0 et des versions 3.0 à 3.1). À noter que son système Debian n'est JAMAIS tombé en panne. JAMAIS, en dehors de coupures de courant, de tentatives d'amorçage sur la mauvaise partition et d'un lecteur Zip IDE particulièrement capricieux qu'il n'a jamais réussi à démonter.] Il a un coup de cœoeur pour Vim et trouve que Perl est très utile pour son travail.

Quand il n'utilise pas Linux, Brian travaille à la conception d'équipements de production d'électricité (équipements pour de grandes centrales), enregistre occasionnellement des disques en public, lit beaucoup trop de science fiction et parcourt les sentiers des Appalaches aussi souvent que possible.