Les générateurs de sites statiques

Xavier Van de Woestyne

Septembre 2015

J'administre la page atelier-prog depuis quelques temps pour offrir des exercices variés dans le monde de la programmation. J'ai récemment eu l'occasion de proposer un article sur la création d'un générateur de site statiques car je trouve que c'est un exercice relativement polyvalent (et que ça permet, éventuellement, à stimuler la motivation des gens à écrire). On m'a très justement fait remarquer que j'étais tristement peu exhaustif sur "ce qui est à faire" et j'ai décidé de répondre à ce manque d'exhaustivité en écrivant un article sur le sujet.
Cet article ne présentera pas l'intégralité des générateurs, ni même un didacticiel sur comment se servir d'un générateur en particulier. Je tâcherai de donner du haut de ma subjectivité une opinion sur le générateur de pages statiques, dans l'absolu, et éventuellement formuler quelques pistes d'implémentation. Le sujet m'intéresse assez, d'ailleur, cette page est elle même issue d'un générateur maison qui maintient l'organisation de ce blog !

Entre les pages statiques et les pages dynamiques

Dans le monde de l'internet… il existe deux types de sites web. Les pages dynamiques et les pages statiques. Une page dynamique est une page qui est générée, par le serveur, au moment où l'utilisateur l'a demande. L'avantage est qu'il devient plus facile de créer de l'interaction avec l'utilisateurs. Par exemple, en PHP:

```
<h1>Mon site</h1>
<span>Il est : <?php echo date('H:i'); ?></span>
```

Cette page renverra l'heure à laquelle la page a été demandée par l'utilisateur. Dans un site statique, le serveur se contente de renvoyer invariablement la même page.
Grâce aux sites web dynamiques, il a été possible de concevoir de l'interaction,

et les applications ont pu commencer à avoir un cycle de vie qui n'était plus spécialement contrôlé par le webmaster.
Essayons d'imaginer un forum de discussion sans interaction possible. Même si l'on peut difficilement se passer, dans un web moderne, de dynamisme, je suis convaincu qu'il est parfois nécéssaire de s'interroger sur l'intérêt d'utiliser l'artillerie lourde pour certaines pages. Par exemple un blog.
En effet, il arrive parfois que la majeure partie des modifications d'une page soient effectuées par le webmaster (c'est le cas de mon blog). Dans ce genre de contexte, il est intéressant de réfléchir à l'usage d'un serveur servant un langage pour faire des pages dynamiques (comme PHP), donc plus complexe à heberger qu'un simple serveur HTTP pour afficher du HTML simple.

> De mon point de vue, l'utilisation du mot "dynamique" est devenu un peu flou, étant donnée la popularité du JavaScript, qui introduit une notion de dynamisme dans les pages, mais c'est un autre débat.

Certains se diront que même pour une page personnelle ne nécéssitant pas d'interaction utilisateur, une technologie servant des pages dynamiques est un énorme plus car on automatise les rendus des pages. L'enjeu de cet article est donc de présenter comment automatiser son flôt de publication au moyen d'un générateur de pages statiques.

Comparatif entre les pages dynamiques et statiques

Ce tableau est évidemment à prendre dans un contexte pertinent. Je répète qu'à mon sens, les deux types de sites ont chacun leurs avantages et inconvénient et bien évidemment leurs domaines d'expertises.

Dynamiques
Plus compliqué à heberger
Interaction possible avec l'utilisateur
Compliqué (et coûteux) de faire des générations overkill (conversion de Markdown, génération de diagram
Complexe à sécuriser correctement
Dépendance d'un cadriciel web

Mon blog est un site statique

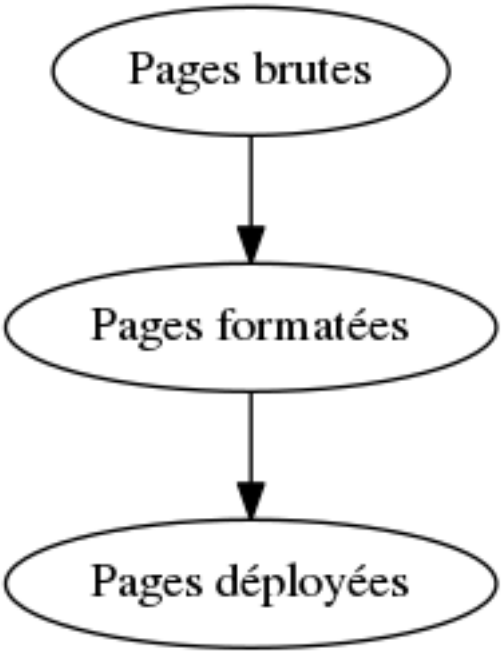
Pour mon cas, j'avais impérativement besoin d'un site facile à heberger (sur les github-pages, pour ne pas m'encombrer d'un serveur HTTP, FTP etc. et les github-pages ne servent *que* des sites statiques.

De plus, je voulais pouvoir écrire mes articles en Markdown et idéalement écrire mon *moteur de blog* en OCaml (parce que c'est un langage que j'adore). J'ai donc écrit un logiciel générant les différentes pages de mon site web et dont la construction/mise à jour des pages est régie par plusieurs petits logiciels OCaml et déployé au moyen d'un Makefile et de Git.

Le principal point faible d'un générateur de site statique porte sur les commentaires. Heureusement, il existe beaucoup de systèmes délocalisés (comme Disqus que j'utilise) que nous évoquerons plus tard dans cet article.

Fonctionnement d'un générateur de sites statiques

Comme il l'a été évoqué plus haut, un générateur n'est rien de plus qu'un logiciel (ou une collection de logiciels) qui offre les outils nécéssaire à l'automatisation de la construction du site localement et éventuellement à son déployement.



Donc plutôt que de laisser au serveur la tâche de générer à la demande toute la hiérarchie logique du site, on crée une version statique d'une collection de fichiers locaux que l'on déploie.
Pour ma part, dans ce système de blog (qui devra être revu un jour), j'utilise des données brutes qui configurent (en XML) la description formelle d'une page. Voici par exemple le fichier de description de cet article :

```
<title>Les générateurs de sites statiques</title>
<subtitle>Présentation et pistes d'implémentation</subtitle>
<desc>Présentation sommaire de ce que sont les générateurs de sites statiques,
ainsi qu'une considération sur une manière d'en implémenter un.</desc>
<keywords>programmation, static, statique, site, blog, générateurs, hakyll,
jekyll, nanoc</keywords>
<date>03/09/2015</date>
<file>raw/articles/blog_static.md</file>
<draft/>
```

Au moment où j'écris ces mots, l'article est encore en brouillon, ce qui explique la présence de la balise `<draft />`. Cette description référence l'article qui est écrit en Markdown et au moment de la génération, le système compresse le HTML et le CSS, génère les fichiers HTML compressé des articles, génère un PDF pour l'article, construit l'index et le flux RSS et il ne me reste plus qu'a déployer.

Grâce à cette méthode, pour peu que l'ordinateur possède Git (et… les dépendance que j'ai avec OCaml, Pandoc et LaTeX, que tout développeur DOIT posséder :)), je peux modifier mon blog de partout, en clonant localement le projet et en ajoutant/modifiant des fichiers.

Quelques solutions éprouvées

Les blogs statiques étant de plus en plus populaires, voici une liste non exhaustive des solutions que j'ai pu survoler :

- Stog : un système écrit en OCaml avec une multitude d'outil;
- Middleman : le chouchou des programmeurs ruby que je connais;
- Hakyll : le générateur de générateur de blog statique écrit en Haskell;
- Jekyll : une solution ruby développé par un des créateurs de Github;
- nanoc : encore une solution ruby;
- pelican : un générateur en python.

Il existe évidemment pléthore d'autres solutions, et je n'ai pas vraiment l'envie de faire un comparatif de ces solutions (il en existe sûrement déjà énormément sur le net !). Donc si vous ne voulez pas coder votre propre outil (parce que contrairement à moi, vous n'aimez pas réinventer la roue… snif), je vous invite à vous faire votre propre opinion sur la question :)
Ce site liste les générateurs les plus populaires.

Pistes pour l'implémentation d'un générateur

Dans cette rubrique, nous allons voir quelques pistes pour implémenter son propre générateur de pages statiques. Sans tomber dans *l'over-engineering* et sans

être hypocrite (en ne proposant rien de généré).
Sans proposer une implémentation, je proposerai ce qui selon moi est la structure minimale d'un générateur, et comment éviter certaines problématiques ennuyantes.
Bien que la construction d'un générateur puisse sembler assez simple, un lot de problématiques peut rapidement faire intrusion si le système n'a pas été mûrement réfléchit à sa source, de plus, il ne faut pas hésiter à utiliser les nombreux outils mis à disposition. (J'aime réinventer la roue, mais il ne faut pas exagérer !)

Les ingrédients

Même si les avis diffèrent sur ce que doit proposer un générateur de sites statiques, voici de mon point de vue ce qu'il est nécessaire d'avoir pour offrir un générateur agréable à utiliser :

- Un format de rédaction (personnellement, j'utilise Markdown) ;
- un convertisseur du format vers du HTML ;
- un outil de description minimale ;
- un outil pour intégrer les pages générées dans un gabarit ;
- un générateur de liste de publications (dans le cas d'un blog).

Dans l'absolu, je ne pense pas qu'il soit nécéssaire d'avoir plus. L'idée générale à cette énumération est de permettre à l'utilisateur du générateur (en l'occurence… nous), d'écrire un article dans un format de texte qui lui plaît (personnellement, j'ai en horreur le BBcode), de lancer le générateur qui va construire une liste en utilisant la description des articles.

Outils

La liste des outils présente, en outre, une ou plusieurs problématiques liées à la construction d'un générateur de sites statiques.

Pandoc: conversion de textes Pandoc est un convertisseur de textes complètement overkill qui peut convertir énormément de formats en énorméments de formats.
Je pense qu'un des points compliqué dans l'écriture d'un générateur de pages statiques est la construction d'un convertisseur de texte (dans mon cas, pour aller de Markdown vers HTML et PDF). Je me sers de Pandoc pour rendre cette tâche aussi simple qu'un simple appel de ligne de commande. L'outil est très riche et bien documenté (et écrit en Haskell). Je le recommande !

Github: déploiement facile Même s'il est toujours possible de déployer manuellement son site web, c'est assez ennuyeux quand on est actif sur sa page. Une solution (utilisée par Grim) est l'implémentation d'un client FTP pour automatiser le déployement, cependant, même si l'exercice est intéressant, c'est tout de même beaucoup de travail.
Depuis un petit temps, Github offre un service d'hebergement de pages statiques (sur un projet précis ou sur une branche orpheline). Déployer son application devient donc extrèmement simple, il suffit de faire un `git push`.

Disqus: les commentaires La teneur statique du site ne permet pas de faire du dynamisme. Pour un blog, le gros point ennuyant est qu'il devient "nativement" impossible d'ajouter un système de commentaires. Il existe beaucoup de services qui permettent d'installer des commentaires sur une page statique au moyen d'une petite ligne JavaScript très courte. Personnellement, j'ai pris l'habitude de me servir de Disqus qui agrège une communauté d'une certaine taille dans le monde des blogs de programmation et qui s'intègre très facilement dans une page (de plus, on ne doit pas se soucier de paramétrer la page, le service s'en charge tout seul).
Il suffit donc de prévoir dans son gabarit destiné aux pages commentables, l'intégration de la portion de code nécéssaire à l'embarquement des commentaires.

Tup: calcul des dépendances Un site statique est amené à être régénéré à chaque modification. Le calcul de ce qui est nécessaire à régénérer est bien plus complexe qu'il n'y paraît. En effet, dans la majeure partie des cas, consulter la date de modification du fichier suffit, mais il existe des cas plus fourbes, comme par exemple lorsque le générateur possède un système de tags apparents et qu'un tag est modifié, ce qui implique la modification de toutes les pages référencées par ce tag.
Bien que je ne m'en sois jamais servi, il parait que Tup est un bon outil pour calculer efficacement ces dépendances.

Synthèses des outils Pour terminer la liste des outils, voici une micro-synthèse (un peu enrichie) de solutions. Tup ne sera pas reprise car c'est un outil qui répond à un besoin un peu trop spécifique et ne concerne que les systèmes de blogs avancés.

Hébergement

- Github page, ma solution favorite ;
- Dropbox (lol) ;
- Site44, une solution d'hebergement extra-simple ;
- KISSr, une abstraction sur Dropbox.

Convertisseurs de textes

Je ne connais que Pandoc que je re-recommande !

Commentaires embarqués

- Disqus, le système que j'utilise ;
- Isso, un outil libre (similaire à Disqus).

Analyses

- Google Analytics, pusique Google sait déjà tout de nous ;
- Piwik, une alternative libre !

Le JavaScript à la rescousse

Même si je ne suis pas féru de l'usage absolument immodéré du JavaScript, ce dernier offre tout de même des raccourcis indéniables ! Je pense que générer une partie acceptable de données JSon peut être une solution "simple" pour faciliter l'indexation (et le problème des tags). De plus, le Javascript permet facilement de mettre en place de petites choses qui agrémentent la navigation, comme par exemple, la barre de recherche présente sur la liste des publications de mon blog.

Conclusion

J'espère que ce bref article aura motivé certaines personnes intéressées par l'atelier créer son blog statique et que éventuellement, nous verrons **plein** de générateurs naître. Et donc en plus, idéalement, plein de blogs naître. Si par malheur vous ne savez pas sur quoi écrire, vous pourrez toujours rédiger un premier article qui explique comment vous avez réussi à faire votre propre générateur de blogs statiques ! Pour ma part, je pense me relancer dans le développement de piplet lorsque je ne serai plus trop pris par mon travail. Merci de votre lecture !