もっと速くなる!
WordPress高速化 Tips

modshrink @mayoibi

最適化の結果

施行前

Latest Performance Report for:
http://www.modshrink.com/

Download PDF

Report generated: Thu, Sep 12, 2013, 10:58 PM -0700
Test Server Region: Vancouver, Canada
Using: Firefox (Desktop) 14.0.1, Page Speed 1.12.16, YSlow 3.1.6

Looks like you're running WordPress
Have a look at our WP optimization tips »

Summary

Page Speed Grade:	YSlow Grade:	
(61%)↓ D	(69%)↓ D	**Page load time:** 6.20s **Total page size:** 1.27MB **Total number of requests:** 75

施行後

Page Speed Grade:	YSlow Grade:	
(93%)↑ A	(79%)↕ C	**Page load time:** 5.66s **Total page size:** 391KB **Total number of requests:** 44

検証ツール

Google Page Speed

http://developers.google.com/speed/pagespeed/insights/

ページを解析して、高速化に必要なポイントを指摘してくれる

GTmetrix

http://gtmetrix.com/

Yahoo!の評価ツールYSlowとPageSpeedを同時に検証しグラフ化

これのPageSpeedでのA評価を目標とする

Firebug, Choromeのデベロッパーツール

改善箇所

- サーバをVPSに移行
- APCを有効化
- WPの不要なプラグインを停止
- Sass導入
- PageSpeedの警告を解消
    - レンダリングブロックJS/CSSの排除
    - ブラウザキャッシュの利用
    - 画像の最適化
    - ページの圧縮

そもそもの経緯

- キタジマさんのWPの管理画面を見る
- すごく**速い**
- 俺も速くしよう

VPS移行

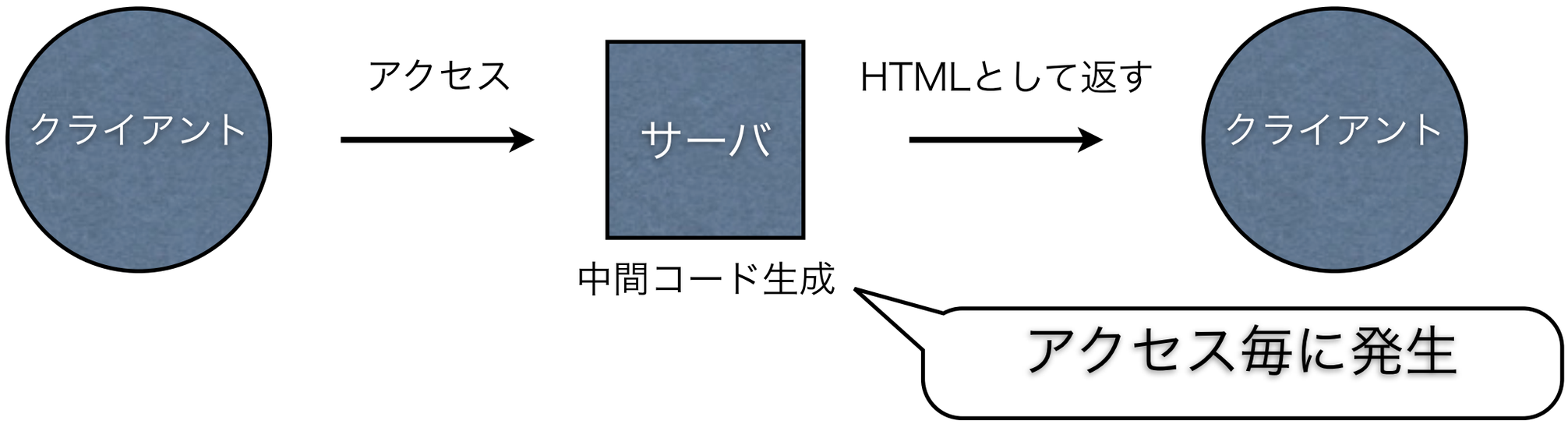
- キタジマ「さくら**VPS**使ってます」
- 即、さくら**VPS**を契約
- 即、サイト移行 (テスト無し)
- 俺「あれ、変わんなくね?」

APC有効化

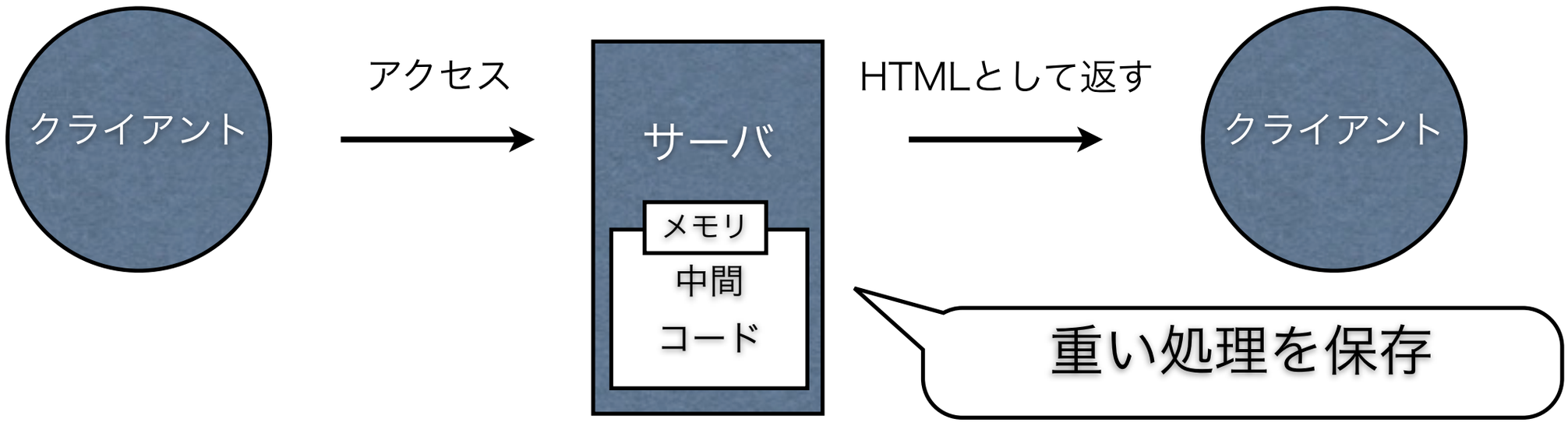
- キタジマ「**APC**有効にしてます?」
- 俺「あ**APC**ね。昨日食ったわ」

APCとは

通常のPHP処理



APCでの処理



VPS, APCを導入の結果

- なんとなく速くなった気がする
- サーバの知識が増える

不要なプラグインの停止

- 見られてない、クリックされてないプラグインや、使われない機能は停止・削除
- 運営形態によって必要な機能は変わる

停止した機能

- よく見られている記事
- 無限スクロール
- Twitterウィジェット
- ソーシャルボタン
- SEOプラグイン
- Jetpackプラグイン

プラグイン厳選の結果

- プラグインがボトルネックだった
- 沢山のJS/CSSが読み込まれるので重い
- Jetpackは便利だけど重い

Sass導入

- CSSの記述を効率化するための仕組み
- 複数のCSSをまとめる

SassでCSSをまとめる

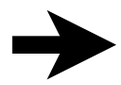
```
//*******************************************
// import reset                    リセット
//*******************************************
@import "reset";

//*******************************************
//Override Font Awesome path
//*******************************************
$FontAwesomePath: "http://www.modshrink.com/wor

//*******************************************
// import font-awesome              アイコンフォント
//*******************************************
@import "font-awesome/variables";
@import "font-awesome/mixins";
@import "font-awesome/path";
@import "font-awesome/core";
@import "font-awesome/extras";
@import "font-awesome/icons";

//*******************************************
// import google web fo             Web フォント
//*******************************************
@font-face {
  font-family: 'Berkshire Swash';
  font-style: normal;
  font-weight: 400;
  src: local('Berkshire Swash'), local('Berkshi
}

//*******************************************
// mixin                                mixin
//*******************************************
@mixin fontsize($size: 10, $base: 10) {
    font-size: $size + px;
    font-size: ($size / $base) * 1rem;
```



```
/*!
Theme Name: modshrink_s
Theme URI: http://underscores.me/
Author: mayoibi
Author URI: http://www.modshrink.com/
Description: Underscores forked.
Version: 1.0
License: GNU General Public License
*/html,body,div,span,applet,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquo
```

レンダリングブロックJS/CSSの排除

- コンテンツの読み込み中に更に読み込まれるJavaScriptやCSS
- ブログパーツや広告のコードに付いてくる外部読み込みのscriptタグも注意

こんなの

```
</article><!-- #post-## -->
<script type="text/javascript" src="http://www.google.co.jp/coop/cse/brand?form=cse-search-box&amp;lang=ja"></script>
```

レンダリングブロックJS/CSSの対策

- CSSをページの最初に読み込ませる
- JSよりもCSSの読み込みを先に書く

ブラウザキャッシュの利用

- .htaccessでキャッシュ期限を設定
- ExpiresActive(最終アクセスからの期限)
- Cache-Control(キャッシュ生成時からの期限)
- 更新頻度によって慎重に指定

CSSがキャッシュされるとまずい?

- 投稿が無くてもCSSを変えることはある
- CSSとテンプレート(HTML)の不一致が起こる
- サーバのCSSは新しいのに見てる人のは古い
- キャッシュの有無で見栄えが違う・崩れる事態

CSSのキャッシュ対策

- Fingerprinting
- クエリを付けて新しいファイルとして読み込ませる
- 例: style.css?ver=20131019
- PHPやWPの関数で自動化

ページの圧縮

- サーバでコンテンツをgzipに圧縮する
- ブラウザで解凍して表示
- 最近のブラウザは解凍に対応
- .htaccessやWPプラグインで設定

gzip圧縮

.htaccessに
AddOutputFilterByType DEFLATE text/plain
とMIME毎に書いていく。

あるいは
WordPressのプラグイン
沢山あるので「gzip」で検索

HTTPヘッダの確認

1回目のアクセス

HTTP/1.1 200 OK
Date: Fri, 18 Oct 2013 23:40:28 GMT
Server: Apache
Last-Modified: Wed, 07 Nov 2012 18:49:10 GMT
Content-Encoding: gzip
~~~~~
Cache-Control: max-age=3600
Expires: Fri, 18 Oct 2013 23:43:28 GMT
Content-Type: image/png

## 期間内の再アクセス

HTTP/1.1 304 Not Modified
~~~~~

画像の最適化

以前のサムネイル仕様

ページの最初の画像を取得して、幅を指定せずにそのまま縮小表示

問題点

画像は小さいのに、ファイルサイズが凄く大きい

アスペクト比が不揃い

改善後の仕様

・ページの最初の画像を取得する
・画像にサムネイルがあれば75pxサイズのものを表示
・Flickrの画像が最初にあれば、Flickrのサムネイル(75px)を表示
・該当の画像がなければプロフィールアイコンを出す

おしまい


モバイル
87 / 100
パソコン
93 / 100
提案の概要
▸ スクロールせずに見えるコンテンツのレンダリングブロック JavaScript/CSS を排除する
このページには、ブロッキング CSS リソースが 1 あります。これが原因で、ページのレンダリングに遅延が発生しています。
▸ ブラウザのキャッシュを活用する
静的リソースの HTTP ヘッダー内で、有効期日や最大経過時間を設定すると、ブラウザがネットワークからではなくローカル ディスクから以前にダウンロードしたリソースを読み込むようになります。
▸ 画像を最適化する
画像を適切にフォーマット化して圧縮すると、データ サイズを大きく削減できます。
▾ サーバーの応答時間を短縮する
Google のテストでは、お使いのサーバーは 0.27秒で応答しました。サーバーの応答時間が遅くなる要因はたくさんあります。サーバーが多くの時間を費やしている箇所を監視し、測定する方法については、Google の推奨事項をお読みください。
▸ 圧縮を有効にする
gzip や deflate を使用してリソースを圧縮することで、ネットワークで送信されるバイト数を減らすことができます。
▸ 5 件のルールに合格
modshrink