










WordPress を高速に表示させるプロのテクニック

*Prime Strategy*

# 本日のメニュー

-  自己紹介
-  高速化の必要性
-  高速化を分けて考える
-  初期レスポンスの改善手法
-  ページ取得完了までの所要時間改善手法
-  参考: 所要時間比較
-  質疑応答

*Prime Strategy*

# 自己紹介

## WordPress プライム・ストラテジー株式会社

- WordPress Consultants 認定企業(アジアでは2番目)
- WordPressをCMSに利用したWebサイト構築を行っています。



財団法人 日本自動車研究所  
<http://www.jari.or.jp/>

組織での運用を行うため、ワークフロー、公開済記事の更新プレビューをプラグインで実現



新華通信ネットジャパン  
<http://www.xinhua.jp/>

12万超の記事をWordPressで管理。広告・バナーの管理、アクセスランキングは専用のプラグインを開発。




- その他、公式プラグイン、テーマを公開中

## WordPress 大曲 仁 (jim912)

- WordPress公式プラグイン「PS Auto Sitemap」「PS Disable Auto Formatting」を開発。
- 案件独自のプラグイン(ワークフロー、取引指標・金融イベント)も担当。
- フォーラムにも出没中です。

# Prime Strategy

Loadingのアイコンがまわりっぱなしになっていませんか？

-  遅いサイトは、ユーザーがストレスを感じて逃げてしまいます。
-  遅いサイトは、サーバに負荷がかかっていてトラブルが発生する可能性も高くなります。
-  遅いサイトは、SEO的にもマイナス。

**だから、即実践！**

*Prime Strategy*

# 高速化を分けて考える

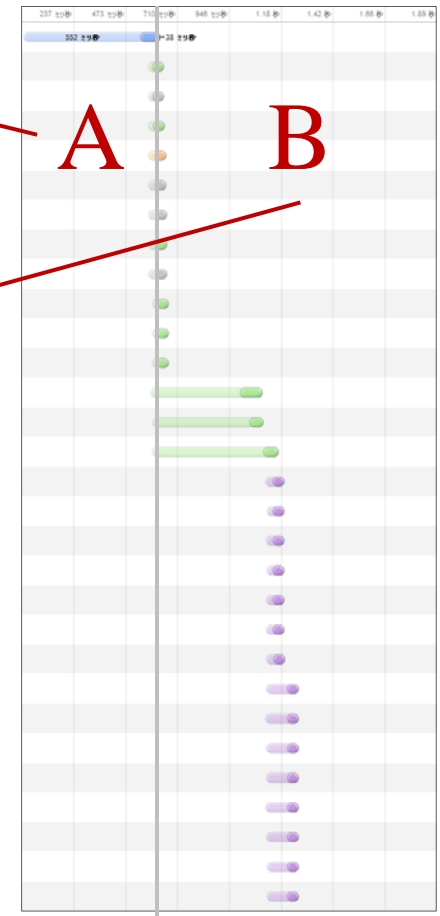
## 初期レスポンスの改善 (HTMLソースの取得)

- これが完了しないことには表示が始まらない。
- 遅いと、ユーザーは情報を得られない状態で待たされるため、ストレスを受けやすい。

## ページ取得完了までの所要時間改善 (リソースファイルの取得)

- 画像、CSS、Javascriptの読み込みを如何に早くするか。
- 最近では、タイムリミットが3秒と言われている。

SafariのWeb Inspector表示



# Prime Strategy

# 初期レスポンスの改善手法

## WordPress サーバ選定

- WordPressのようなアクセス時にコンテンツを生成する動的サイトは、サーバ性能(特にCPU、メモリ)によるパフォーマンスの差が顕著。
- 後述するバイトコードキャッシュ、クエリキャッシュを十二分に活用するために、また、利用できる環境を構築するためにも、共有サーバではなく専用環境がおすすめ。(VPSは、比較的安価に導入可能)

## WordPress PHPの速度改善

- APC、eAcceleratorなどのバイトコードキャッシュは威力大。**重要**  
構文解析にかかる時間を丸ごと削減  
WordPressの実行時間でおおよそ50%前後の時間短縮。
- PHP Server APIはapacheの方が高速に動作  
パーミッションによる実行のコントロールができなくなるデメリットはあるが、これも専用環境なら問題なし。

Prime Strategy

# 初期レスポンスの改善手法

## WordPress MySQLの最適化

- Key buffer、Query cacheを出来る範囲で大きく。 **重要**  
DBからの読み出しが早くなり、レスポンスが向上する。

## WordPressで出来ること

- WP-Super-cacheなどのキャッシュプラグインの導入。  
アクセスが多い(DBへの負荷が高い)サイト、リピーターが多いサイトには効果大。  
ローカルキャッシュがない初回訪問時の効果は大きくないので、逆効果になる場合も。
- 表示する記事をカスタマイズするためのquery\_postsはコスト増。  
スキルがあるなら、SQL文にフックしよう。(wp-includes/query.php)

## WordPress まとめ

- キャッシュをいかに上手く利用できるかがキー。  
そのためにも、メモリを豊富に利用でき、設定を変更できる環境の準備が大切。

# Prime Strategy

## WordPress Webサーバの設定

### ➤ Keep Aliveの有効化 **重要**

HTTPリクエストの転送終了後も一定時間接続を閉じないため、連続したリクエストの接続確立コストを削減できる。

## WordPressの工夫

### ➤ mod\_rewriteでリソースファイルの拡張子を除外

パーマリンク設定時は、存在しないファイルへのリクエストもWordPressが処理してしまうため、大きな遅延が発生してしまう。faviconは特に盲点。これを防ぐために、RewriteCond の指定を追加。

#### 追加記述例

```
RewriteCond % {REQUEST_URI} !¥.(gif)|(css)|(js)|(swf)|(jpeg)|(jpg)|(jpe)|(png)|(ico)|(swd)|(pdf)|(doc)|(docx)|(xls)|(xlsx)$
```

※ mod\_rewrite\_rulesフィルターフックで追加可能。

# Prime Strategy

## WordPress その他の構成要素

- GIFから8bitPNGへのリプレース  
アニメーション機能が無い以外は、同等で、概してファイルサイズが小さくなる。
- CSSスプライトの活用  
※ 複数の画像を1つのファイルにまとめ、背景画像の位置指定で個々の画像を表示する方法  
転送ファイル数を少なくできるため、転送時のロスを削減できる。
- ブログパーツ、広告、API等の利用はほどほどに **重要**  
他サーバとの通信が発生する要素は、別コネクションとなるため遅延の原因になりやすい。
- Javascript、CSSの圧縮  
空白文字列を削除し、ファイルサイズを縮小

## WordPress まとめ

- あたりまえですが、ファイル数とファイルサイズは少なく。  
但し効果はファイル数 > ファイルサイズ
- パーツのごった煮サイトは、重くて食えないサイト。

# Prime Strategy

# 参考:所要時間比較

## 異なるサーバに同じWordPressを設置し、表示に要する時間を計測

- サーバA (共用サーバ、Keep Alive: 無、PHP Server API: Apache、バイトコードキャッシュ: 無)
- サーバB (共用サーバ、Keep Alive: 有、PHP Server API: CGI、バイトコードキャッシュ: 無)
- WEXAL® CMSサーバ(VPS、Keep Alive: 有、PHP Server API: Apache、バイトコードキャッシュ: 有(APC))

### 計測方法

キャッシュを無効化した状態で、トップページにアクセス。要した時間の30回平均。

	サーバA	サーバB	WEXAL® CMSサーバ
WordPress処理時間	0.398秒	0.444秒	0.154秒
読み込み完了時間	8.540秒	1.688秒	0.991秒

Postsテーブルのレコード数 : 400  
トップページの構成ファイル数: 70

計測日時: 2009年10月15日 17:00-18:00  
WordPress処理時間: フッター部分に配したtimer\_stop関数で取得した値  
読み込み完了時間: Firefox3.5.3 + Extend StatusBarでのページ読み込み時間

# Prime Strategy



ご清聴ありがとうございました。

*Prime Strategy*