

# MTサイト高速化のための王道

コンテンツがもっと広がる、つながる、CMSプラットフォーム

2014.09.29

**主に CMS 設計ご担当者向け王道**

## 再構築処理

- ウェブサイト・ブログ単位に実行
- 登録されたテンプレートの種類単位に実行
- テンプレートエンジンの処理
  - テンプレートを DB から読み込み
  - テンプレートタグを処理 (DB からデータを読み込み)
  - 処理結果をメモリ上に保存
  - 出力ファイルの内容と比較 (出力先は DB で管理)
  - ファイルを更新

## 🔄 王道 1) 適切にウェブサイト・ブログを分割

- 再構築時間は記事数に応じて変化
- 実装例
  - ナビゲーションメニュー単位で分割  
ウェブサイト: グローバルナビ  
ブログ: サイドナビ
  - 年・カテゴリ単位で分割

## 🔄 王道 2) 不要なアーカイブは作成しない

- アーカイブ: ある切り口から記事を整理したページ
- カテゴリアーカイブは処理時間がかかる
- 実装例
  - ブログの適切な分割 → アーカイブ作成が不要
  - オウンドメディア以外 → 月別アーカイブ不要

## 王道 3) 不要なプラグインは削除

- mt.cgi の初期化では以下を必ずチェック
  - plugins ディレクトリ内のプラグイン
  - 設定ファイル mt-config.cgi の設定内容
- 実装例
  - 不要なものは plugins ディレクトリから削除

## 王道 4) 変数を活用

- 1ページ内で同じテンプレートタグを利用する場合  
変数に登録すると処理がはやい

```
<mt:EntryPermalink setvar="pgUrl">
```

## 🔄 王道 5) テンプレートモジュールを活用

- モジュール: 共通の処理を登録
- 処理結果をキャッシュ化
- 処理結果をファイルに出力 SSI で利用
  - SSI: Server Side Include
  - ウェブページで「共通ライブラリ」化することも可能

## テンプレートモジュール利用例

```
<mt:Ignore><mt:SetVar name="debug" e="0"></mt:Ignore>
<mt:Unless name="debug">
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();
a=s.createElement(o), m=s.getElementsByTagName(o)[0];
a.async=1;a.src=g;m.parentNode.insertBefore(a,m)}
(window,document,'script','//www.google-analytics.com/analytics.js','ga');
ga('create','UA-XXXXX','auto');
ga('send','pageview');
</script>
</mt:Unless>
```

```
<mt:Include module="GoogleAnalytics" blog_id="1" debug="0">
```

```
<!--#include virtual="/common/include/ga.html" -->
```

## 王道 6) 余計な計算処理はしない

- わかりきった URL は直接記述
- DB へのアクセスを極力減らす

## テンプレート例

```
<mt:Unless name="compress"
  regex_replace='/https?:\//\//www\.example\.com/g', ''
  regex_replace='/^\s+/gm', ''>
<body>

</body></mt:Unless>
```

<body>…</body> において

- http(s)://www.example.com/privacy.html を /privacy.html に変換
- 先頭の空白・空白行を削除

TIPS:

- Google の URL スタイル (//www.example.com/privacy.html) に変更可能

## 🔄 王道 7) アクセス頻度にあわせ時間差で再構築

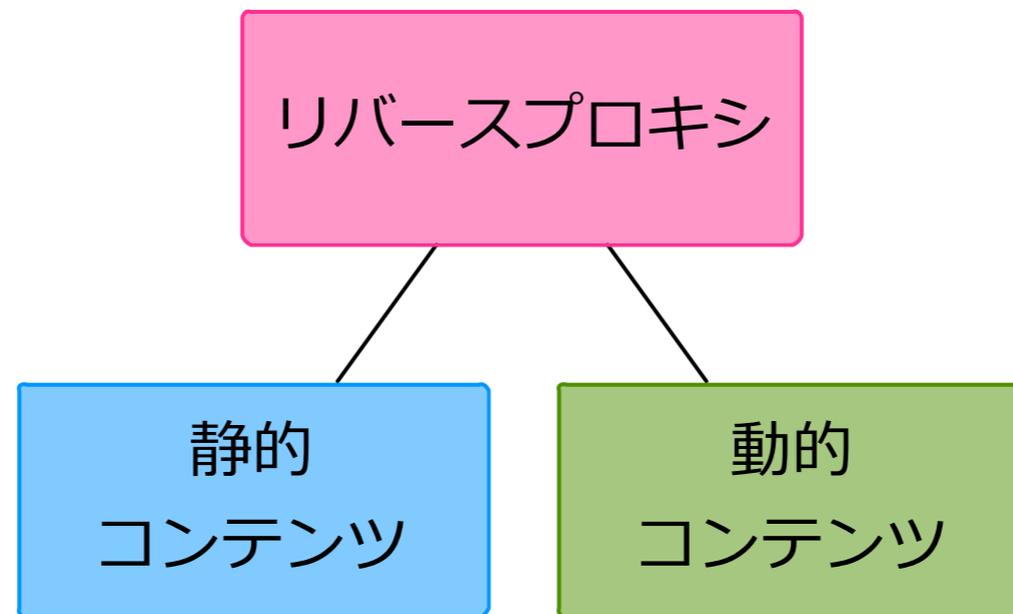
- 時間差で再構築することでサーバーへの負荷を軽減
  - アクセス頻度の低いアーカイブ
  - コメントの投稿が頻繁にある記事

**MT の原理を理解すればこわくない**

**主にインフラご担当者向け王道**

## 🔄 王道 8) 静的コンテンツと動的コンテンツを分化

- ウェブサーバーに対する負荷が異なる
- リバースプロキシを利用
  - 静的コンテンツと動的コンテンツの処理をわけける



Developer Tools - http://www.sixapart.jp/

Elements Network Sources Timeline Profiles Resources Audits Console

Preserve log Disable cache

Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency	Timeline
www.sixapart.jp	GET	200 OK	text/html	Other	9.8 KB 32.5 KB	238 ms 107 ms	
reset.css /share/css	GET	200 OK	text/css	www.sixapart.jp/:24 Parser	1.1 KB 1.7 KB	50 ms 49 ms	
common.css /share/css	GET	200 OK	text/css	www.sixapart.jp/:25 Parser	7.2 KB 29.7 KB	89 ms 57 ms	
dropdown.js /share/js	GET	200 OK	application/x...	www.sixapart.jp/:34 Parser	8.2 KB 7.9 KB	253 ms 215 ms	
plusone.js apis.google.com/js	GET	304 Not Modified	application/j...	www.sixapart.jp/:35 Parser	438 B 35.6 KB	333 ms 329 ms	
sa_common.css /share/css	GET	200 OK	text/css	www.sixapart.jp/:29 Parser	3.7 KB 14.5 KB	178 ms 158 ms	
top.css /share/css	GET	200 OK	text/css	www.sixapart.jp/:30 Parser	3.1 KB 12.0 KB	173 ms 159 ms	
slider.js /share/js	GET	200 OK	application/x...	www.sixapart.jp/:42 Parser	2.1 KB 1.8 KB	246 ms 242 ms	
feed.js /share/js	GET	200 OK	application/x...	www.sixapart.jp/:51 Parser	5.6 KB 5.3 KB	243 ms 221 ms	
jquery.js /share/js	GET	200 OK	application/x...	www.sixapart.jp/:32 Parser	92.0 KB 91.7 KB	263 ms 121 ms	
print.css /share/css	GET	200 OK	text/css	www.sixapart.jp/:27 Parser	565 B 289 B	160 ms 157 ms	
dropdown.css /share/css	GET	200 OK	text/css	www.sixapart.jp/:26 Parser	1.2 KB 2.4 KB	162 ms 156 ms	
common1308.js /share/js	GET	200 OK	application/x...	www.sixapart.jp/:33 Parser	4.7 KB 4.4 KB	178 ms 164 ms	
button-only.gif b.st-hatena.com/images/entry-button	GET	200 OK	image/gif	www.sixapart.jp/:631 Parser	(from cache)	12 ms 12 ms	
noteit.js static.evernote.com	GET	200 OK	application/j...	www.sixapart.jp/:640 Parser	(from cache)	51 ms 51 ms	
widgets.js platform.twitter.com	GET	304 Not Modified	application/j...	www.sixapart.jp/:625 Parser	458 B 98.6 KB	137 ms 117 ms	
bookmark_button.js b.st-hatena.com/js	GET	304 Not Modified	application/x...	www.sixapart.jp/:632 Parser	244 B 30.3 KB	132 ms 127 ms	
article-clipper.png static.evernote.com	GET	200 OK	image/png	www.sixapart.jp/:641 Parser	1.3 KB 888 B	142 ms 134 ms	
typenavi_logo_lekumobb.png /share/images	GET	200 OK	image/png	www.sixapart.jp/:102 Parser	4.3 KB 4.0 KB	82 ms 68 ms	
typenavi_logo_mt.png /share/images	GET	200 OK	image/png	www.sixapart.jp/:110 Parser	3.5 KB 3.2 KB	89 ms 86 ms	

106 requests | 1.2 MB transferred | 5.51 s (load: 4.92 s, DOMContentLoaded: 1.41 s)



## 王道 9)

# HTML 以外の静的コンテンツの処理を最適化する

- リクエストの多くは HTML 以外
- 対策
  - リクエストの回数を減らす
  - 画像最適化や minify ツールでファイルサイズを減らす
  - JavaScript の読み込みタイミングを考える

## 🔄 静的コンテンツへのリクエストを減らす

- ブラウザ側のキャッシュを有効活用
  - 拡張子別に有効期限を設定
  - 期限が有効な間はリクエストされない
- インラインイメージの利用（IE8 以降）
- より高速なウェブサーバーを利用
  - Apache よりかは nginx
- CDN（例: CloudFront）の利用

## 🔄 王道 10) 動的コンテンツは FastCGI/PSGI に対応

- データベースへのアクセスを Excel を例に
  - 通常処理: アクセスのたびに毎回 Excel を起動
  - FastCGI/PSGI: 最初に Excel を起動
- 2回目以降の実行で初期化処理が不要
- mt.cgi の高速化には不可欠

## 🔄 王道 11) アクセス頻度の低いページを動的処理

- 目的に応じて選択
  - ダイナミックパブリッシング  
個別の URL でリクエストの度にページ生成
  - Data API  
固定の URL でリクエストに応じて内容を変更
  - JavaScript フレームワーク (例: AngularJS)  
JavaScript で内容を変更

## 王道 12) MySQL の設定をみなおす

- キャッシュ機能で読み込みスピードを改善  
1/5 程度まで改善することはむずかしくない

**ウェブの原理を理解すればこわくない**

**主にクライアント企業むけ王道**

## サイトの高速化は「非機能要件」のひとつ

- 「非機能要件」に含まれるもの
  - 性能
  - 信頼性
  - 拡張性
  - 運用性
  - セキュリティ

## 🔄 「セキュリティ」は重要なテーマ

- 脆弱性はアプリケーションだけでなく多岐にわたる
- 脆弱性問題に早急に対応する体制が必要

たかがウェブではない

制作会社内にインフラエンジニアは少ない

保守契約を締結しているクライアント企業も少ない

## 🔄 王道 13) サイトの特性にあわせてクラウドを選択

- 月間100万PV 以下の企業サイト
  - Movable Type クラウド版
- トラフィックが劇的に変動するサイト
  - Movable Type for AWS をベースに AWS で構築
- 企業内情報システムとの連携が必要なサイト
  - Microsoft Azure で構築

# Movable Type クラウド版の特長

## メンテナンスコストを削減

シックス・アパートがサーバーのプロセス監視を行っており、問題発生時には素早く対応します。さらに、冗長構成で障害時も自動復旧するので安心。管理、メンテナンスのコストを大幅に削減できます。

## 管理画面の動作が速く快適

環境を Movable Type に最適化し、仮想サーバーの能力を最大限に発揮できるように徹底的にチューニングされています。そのため、管理画面などの動作が高速で、快適に利用できます。

## 常に最新版を使用可能で安心・安全

バージョンアップ作業や、サーバーOS、ソフトウェアのアップデートはシックス・アパートが実施。セキュリティフィックスも即座に行われ、常に安全な環境で利用できます。

## テクニカルサポート付き

利用料金には、メールでのテクニカルサポートも含まれています。技術的なトラブルや疑問が発生した場合にも安心です。

## クラウド版独自の機能で運用時の細かな要求にも対応

クラウド版は、マネージド・サービスのため、サーバー設定等の自由度が制限されますが、Basic認証や、HTTPリダイレクトの設定など、必要な設定を管理画面から行えるようになっていました。さらに、コンテンツを外部のサーバーへ配信して公開できるサーバー配信機能なども搭載しています。

# GMOクラウドプラン

GMOクラウド社のパブリッククラウド「ALTUS（アルタス）」を使用。費用をより安く抑えられます。

XSg	S2g	S4g	S450g	S4100g
CPU：0.5CPU (仮想CPU)	CPU：1CPU (仮想CPU)	CPU：1CPU (仮想CPU)	1CPU (仮想CPU)	1CPU (仮想CPU)
メモリ：0.5GB	メモリ：2GB	メモリ：4GB	メモリ：4GB	メモリ：4GB
ディスク：10GB	ディスク：10GB	ディスク：10GB	ディスク：50GB	ディスク：100GB
5,000円（税抜）／月	8,000円（税抜）／月	14,000円（税抜）／月	22,000円（税抜）／月	28,000円（税抜）／月

# IDCフロンティアプラン

IDCフロンティア社のクラウドサービスを使用。高い信頼性と、高品質なネットワークが同社により保証されています。

XSi プラン	S2i プラン	S4i プラン	S450i プラン	S4100i プラン
CPU：1CPU (0.8GHz相当)	CPU：1CPU (1.6GHz相当)	CPU：1CPU (1.6GHz相当)	CPU：1CPU (1.6GHz相当)	CPU：1CPU (1.6GHz相当)
メモリ：0.5GB	メモリ：2.0GB	メモリ：4.0GB	メモリ：4.0GB	メモリ：4.0GB
ディスク：10GB	ディスク：10GB	ディスク：10GB	ディスク：50GB	ディスク：100GB
9,500円（税抜）／月	14,200円（税抜）／月	19,000円（税抜）／月	28,000円（税抜）／月	35,000円（税抜）／月

**サイトの特性にあわせクラウドを  
選択すればこわくない**

## まとめ

- MT の原理を理解すればこわくない
- ウェブの原理を理解すればこわくない
- サイトの特性にあわせクラウドを選択すればこわくない