



速さが重要！

ウェブサイト高速化セミナー



本資料の無断での複写・複製・転載・改変・配布等を禁止します



株式会社ディテイルクラウドクリエイティブ

<https://detail-cc.com/>

東京都北区赤羽1-30-1 高栄ビル4F

TEL: 03-6454-4082/ FAX: 03-6454-4083

Mail: info@detail-cc.com



表示速度はユーザ満足のための最重要要素

表示速度はユーザー行動にどう影響するか？

amazon.com[®]

「サイト表示が0.1秒遅くなると、売り上げが1%減少し、1秒高速化すると10%の売上が向上する」 [Web experiments generate insights and promote innovation.](#)

Google

「表示速度が0.5秒遅くなると、検索数が20%減少する」 [Marissa Mayer at Web 2.0](#)

W. マイナビ

「3秒を過ぎると57%のユーザーがしびれを切らし、訪問を諦める」

[3秒が許容範囲 - Webサイトのパフォーマンスが重要な理由](#)

表示速度はユーザー行動にどう影響するか？ Google社 [モバイルページの速度に関する業界の新しいベンチマークへの取り組み](#)



As page load time goes from:

1s to 3s the probability of bounce **increases 32%**
1秒から3秒になると**直帰率は32%増加**

1s to 5s the probability of bounce **increases 90%**
1秒から5秒になると**直帰率は90%増加**

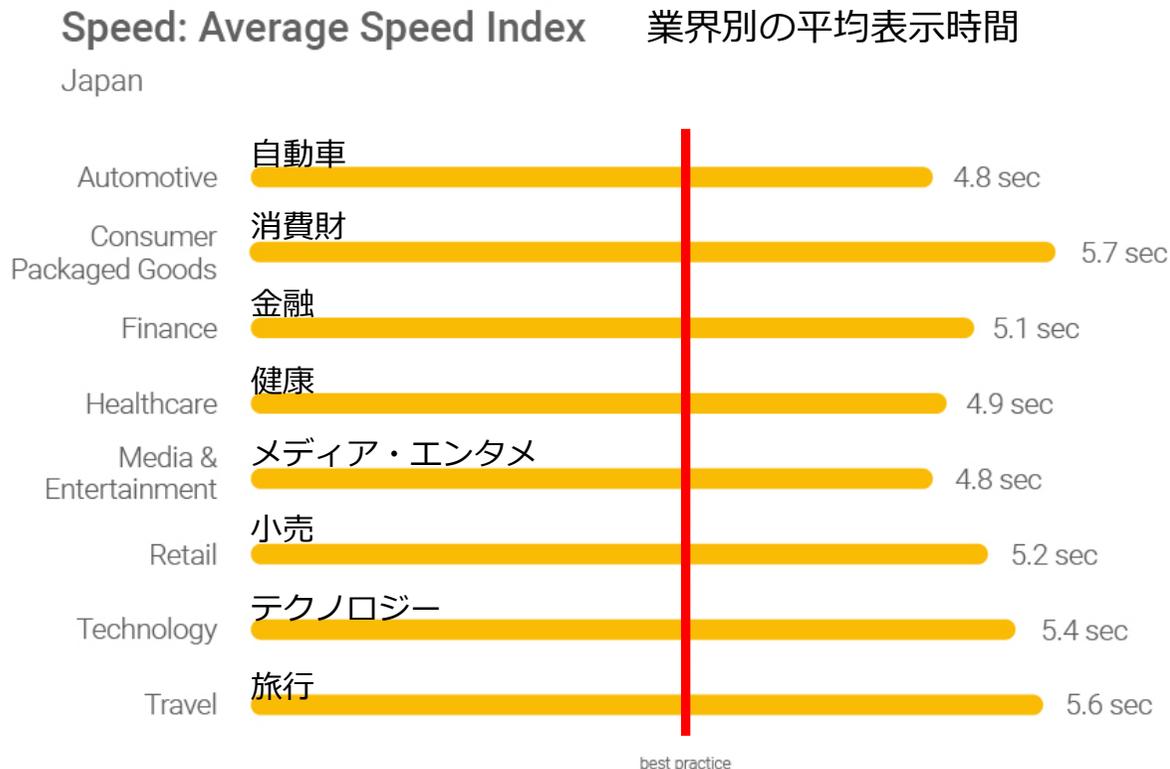
1s to 6s the probability of bounce **increases 106%**
1秒から6秒になると**直帰率は106%増加**

1s to 10s the probability of bounce **increases 123%**
1秒から10秒になると**直帰率は123%増加**

※直帰率：1ページだけ見て他のページを見ずにWebサイトから離脱するユーザーの割合

合格点は3秒！

Google社 [モバイルページの速度に関する業界の新しいベンチマークへの取り組み](#)



多くのサイトは**落第点**！

優良の基準：3秒

Googleは表示速度の速いWebサイトを優先する

「いくらコンテンツが良くても」「いくらデザインが良くても」
表示が遅ければ「見てもらえない」「しっかりに見る気になれない」

だから

表示速度はすべてに優先する前提要素なのです！

表示速度は非常に重要であるため Googleは検索結果の順位決定の要因としてウェブサイトの表示速度を含めていると公式にアナウンスしています。

現在では表示速度の高速化はSEOの主戦場のひとつとなっています。

まずは現状を把握

Webサイト速度のチェック方法

3つのおすすめ方法

1. PageSpeed Insights
2. Gtmetrix
3. Google Search Console

PageSpeed Insights <https://developers.google.com/speed/pagespeed/insights/>

Google社が提供する表示速度のチェックツールです。
以前は速度評価表示のみでしたが、改善アドバイスの項目も徐々に増えており、利用価値の高いツールとなりました。

特徴

- 点数表示がわかりやすい
- 分析結果表示が速い

手軽なのでざっくり分析することに向いています。



GTmetrix <https://gtmetrix.com/>

表示速度の低下の原因になっている個所について分析時間はかかるものの、非常に細かく分析してくれる有用なツールです。

特徴

- 精密な分析が可能
- 過去の分析履歴が保存されている
- 分析履歴同士で比較ができる

The screenshot shows the GTmetrix interface with a navigation bar containing buttons for '再テスト' (Retest), '比較する' (Compare), 'ページ設定' (Page Settings), 'モニター' (Monitor), and 'アラートを設定する' (Set Alerts). The main content area displays a '最新のパフォーマンスレポート' (Latest Performance Report) for the URL <https://www.cao.go.jp/>. The report was generated on Sunday, July 2, 2019, at 19:33 PM -0700. The test was performed from a server in Canada (Banque Parajon) using Chrome (Desktop) 62.0.3202.94, PageSpeed 1.15-gt1.2, and YSlow 3.1.8. A hint suggests using a CDN. Below the report, there are two sections: 'パフォーマンススコア' (Performance Score) and 'ページ詳細' (Page Details). The Performance Score section shows PageSpeed Score at E (55%) and YSlow Score at E (58%). The Page Details section shows a full load time of 3.1 seconds, a total page size of 1.39MB, and 52 requests.

パフォーマンススコア		ページ詳細		
PageSpeedスコア	Yスロースコア	全負荷時間	総ページサイズ	リクエスト
E (55%)	E (58%)	3.1秒 [^]	1.39MB [^]	52 [^]

Google Search Console

速度要因についての分析はありませんが、以下の2つの特徴があるため非常に利用価値の高いツールです。

• サイト内の全ページの速度を把握することができる

Search Consoleでは全ページについて把握できます。

Search Consoleで遅いページを特定し、その後PageSpeed InsightsやGTmetrixで改善箇所を特定するという流れで改善をすると労力を節約できます。

• 速度の推移を把握できる

過去の速度を把握できるので、改善の前後の効果がひと目でわかります。



高速化の実際の方法

1. 高速なWebサーバーに移転：高速化の代表的な手法

No	高速化手法	手間	効果	おすすめ度
1	高速なWebサーバーに移転	中	★★★★★	★★★★★
2	ソーシャルボタンをはずす	小	★★	★★★★★
3	画像圧縮	小～中	★★★★★	★★★★★
4	Keep aliveの有効化	小	★★★★	★★★★★
5	ブラウザのキャッシュの有効化	小	★★★★	★★★★★
6	gzip圧縮	小	★★★★★	★★★★★
7	遅延読み込み	小	★★★★	★★★★★
8	Javascriptの非同期化	小	★★★★	★★★★★
9	AMP対応	最大	★★★★★	★★★★★
10	CSSスプライトの導入	中	★★	★★★★★
11	DNSのルックアップを減らす	小～中	★★★★	★★★★
12	CDN導入	中	★～★★★★★	★★★★

※上記手法は独自開発システムの高速化については基本的には想定していません。

Wordpressや静的HTMLで構築されたWebサイトを想定しています。

1. 高速なWebサーバーに移転：サーバ移転効果

サーバーの導入時期が何年も昔からで、ずっと使い続けている場合は、新しいサーバーに移転することで**表示速度が向上**するケースがあります。

さらにこのようなケースの場合の副次効果として、管理画面のレスポンスも良好で作業効率が向上、サーバー費用も抑えられるといった効果もあります。

圧倒的に表示速度が向上というケースもありますので、現在使っているサーバーをまずご確認ください。

1. 高速なWebサーバーに移転：Webサーバのソフトウェアの速度比較

Webサーバーとは「パソコンやスマホなどのブラウザを使ってWebサイトにアクセスした場合に、ブラウザに必要な情報を送り返すソフトウェア」のことです。

Nginxはこれまで主流で使われていたApacheに比べて高速ですが、LiteSpeedは更に高速です。

Wordpressで使用した場合3倍以上の応答速度が期待できます。

LSWS with LSCache Performance Gain

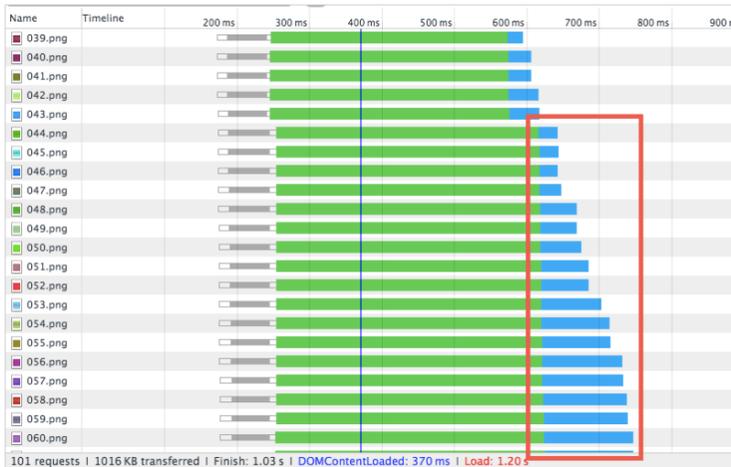
Concurrent Users	Keep-Alive	Apache		Nginx	LiteSpeed	
		Apache / WPRocket	Apache / W3TC	Nginx / FCGI	LSWS / W3TC	LSWS / WPRocket
10	No	5.6x	3.2x	4.9x	1.2x	1.2x
	Yes	7.4x	4.0x	6.4x	1.1x	1.1x
100	No	5.7x	3.2x	5.0x	1.2x	1.2x
	Yes	6.3x	3.5x	5.0x	1.1x	1.1x

出典：<https://www.litespeedtech.com/benchmarks/wordpress>

1. 高速なWebサーバーに移転：HTTP/2の使えるサーバ

HTTP/2は多数のファイルを同時に処理できます。

【HTTP/2のタイムライン】



多数の画像ファイルを同時にダウンロードするケース。

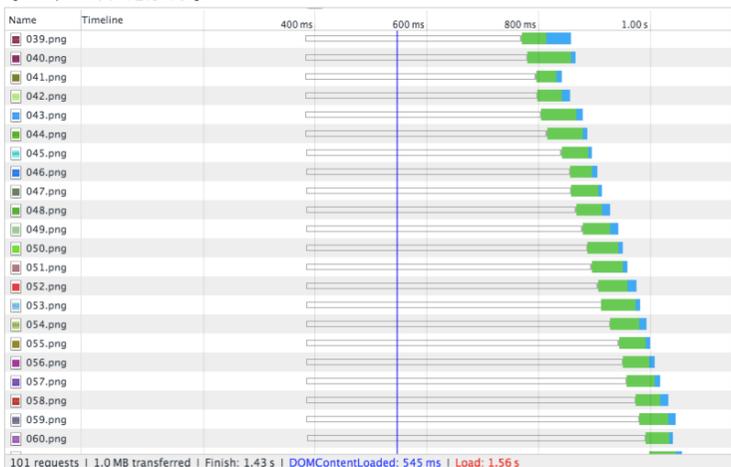
赤枠内が同時に画像のダウンロードが行われている箇所で、下のHTTP/1.1では処理の待ちが発生していることがわかります。

グラフ引用：

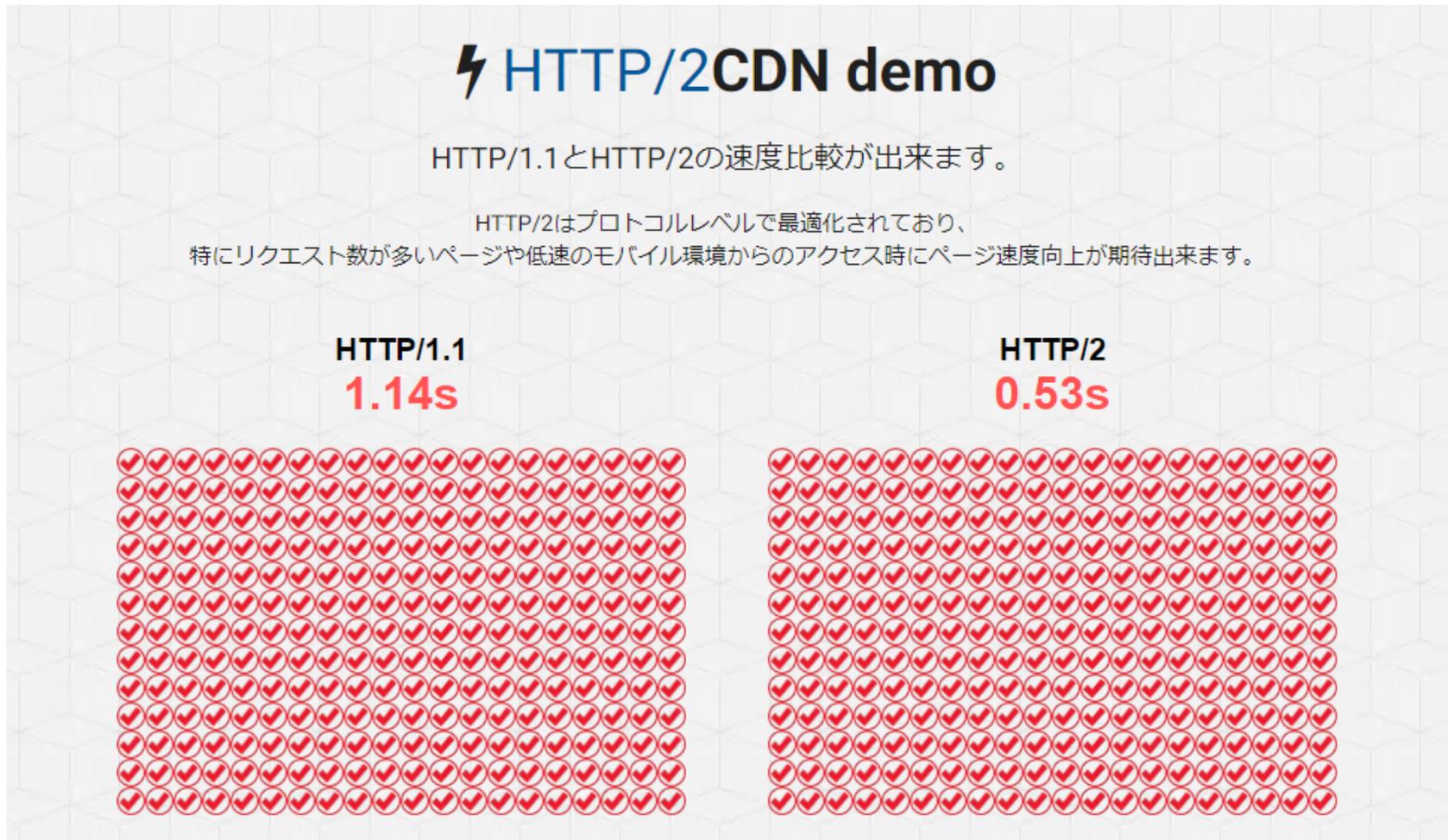
HTTP/2 と HTTP/1.1 のページ読み込みスピードを比較

<https://blog.apar.jp/web/2544/>

【HTTP/1.1のタイムライン】



1. 高速なWebサーバーに移転：HTTP/2とHTTP/1.1の比較



引用：HTTP/2CDN demo <https://http2.redbox.ne.jp/>

2. ソーシャルボタンを設置するページを精査する



ソーシャルボタンはページの表示速度を下げる原因になることもあります。

ツイート数、いいね数といった数値を外部から読み込むため大きくレスポンスが悪化します。特にFacebookのソーシャルの処理は遅いです。

「サイトトップ」「商品詳細」ページといったようなコンテンツではないページについてはソーシャルボタンを外すことを推奨します。

作業工数が非常に小さい割には効果が高いのでまず行うべき対策だといえます。

コラムコンテンツといったソーシャルボタンが不可欠なページの場合、7.にて紹介する「遅延読み込み」で紹介している方法を活用するなどして影響を軽減しましょう。

3. 画像ファイルのサイズを小さくする

Webページの読み込みデータ量の大部分は画像データだといっても差し支えありません。今まで画像について圧縮せずそのままアップロードしていたような場合、表示速度が数分の1になることもまれではありません。最重要な高速化手法といえます。以下の4つが重要な画像の圧縮方法です。

- 画像サイズを画面に表示されるサイズよりも大きい画像でアップロードされていたら、サイズを小さくしてサイズを減らします。
- 画像の品質を問題ない程度に下げてデータ量を減らします。
- 画像フォーマットを圧縮率の高い形式にします
- Exif情報を削除します

3. 画像ファイルのサイズを小さくする：大きさを縮小



記事中の画像をレイアウトに合わせて縮小し、サムネイル画像としてそのまま使われているケースは非常に多くあります。縦横の長さがサムネイルについて半分であれば最初からサムネイル用の画像を用意することで、単純計算でファイルサイズは4分の1まで小さく(75%削減)することができます。

サムネイル用画像の生成及び置き換えはWordpressの場合はプラグインを使うと簡便です。

EWWW Image Optimizer <https://ja.wordpress.org/plugins/ewww-image-optimizer/>

3. 画像ファイルのサイズを小さくする：画像の品質を問題ない程度に下げる①

JPEGやWebPの画像は圧縮率を大きくすることで、ファイルサイズを小さくすることができます。圧縮率を高くするとファイルサイズは小さくなりますが、画質は悪化します。圧縮率によってどれだけ画質が悪化するのか検証してみましょう。

画像処理アルゴリズム向けの試験用標準画像 lena_std.tif をJPEG画像に変換した場合の画質を比較します。 <http://www.lenna.org/>

画質80と画質50では変わらないように見えますが、ズームして比べたらどうでしょうか？

画質100



画質80



画質50



→ 圧縮率大（画質低）

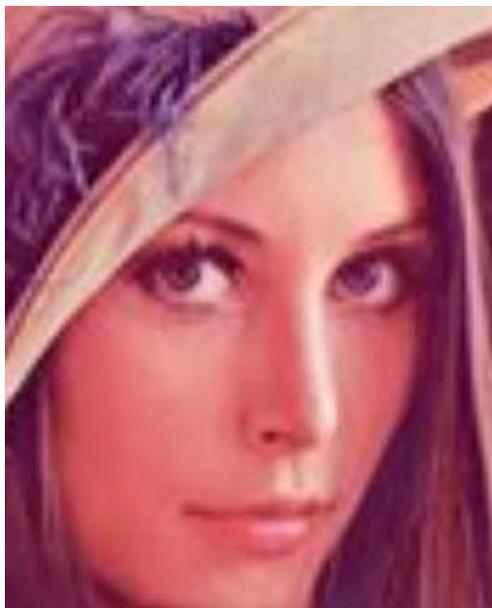
3. 画像ファイルのサイズを小さくする：画像の品質を問題ない程度に下げる②

ズームすると、細かい部分でディテールが失われたりノイズが出ていることがわかります。髪の毛の質感、頬部分のノイズ、目の部分など、画質の低下が分かります。しかし、ズームしない状態なら気づかないかもしれません。

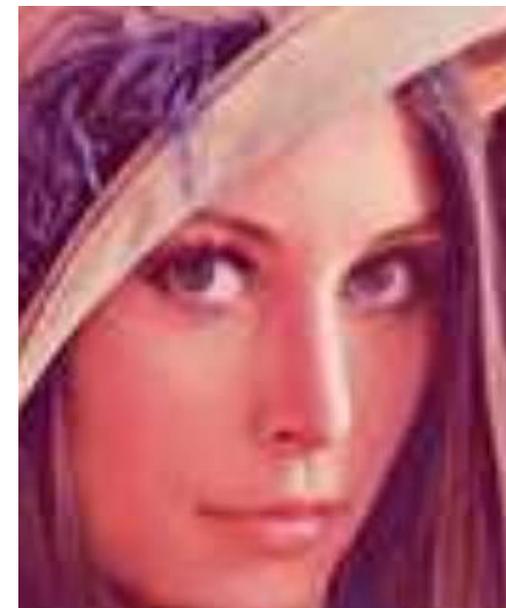
画質100



画質80



画質50

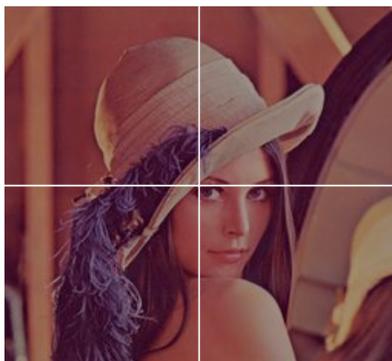


→ 圧縮率大（画質低）

3. 画像ファイルのサイズを小さくする：画像の品質を問題ない程度に下げる③

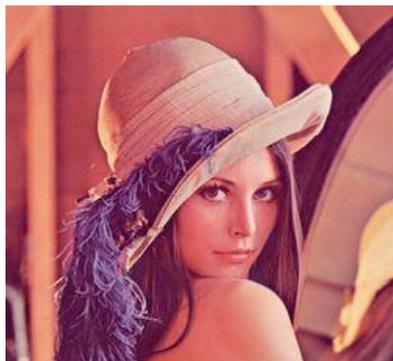
画質やピクセル数によるファイルサイズの変化は比例しません。

画質100 (395 KB)



512ピクセル

画質80 (42.0 KB) 元画像の1/9



512ピクセル

画質50 (22.8 KB) 元画像の1/16



512ピクセル

画質80 (13.4KB) 元画像の1/30

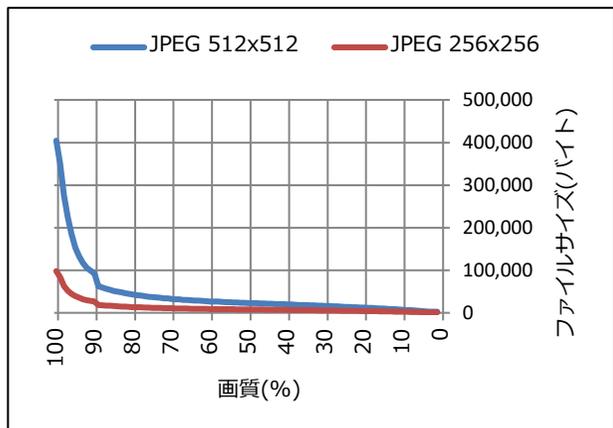


256 ピクセル

画質50 (7.94KB) 元画像の1/50



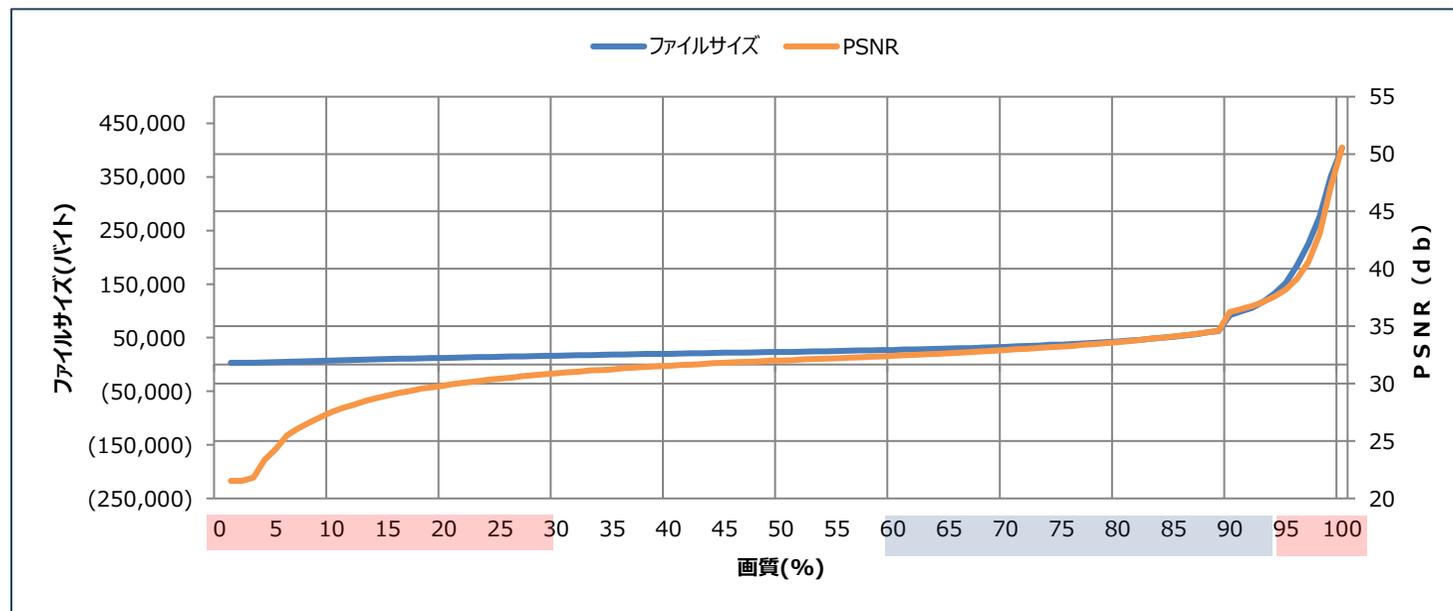
256 ピクセル



3. 画像ファイルのサイズを小さくする：画像の品質を問題ない程度に下げる④

画質の良否はPSNR(Peak Signal to Noise Ratio / ピーク信号対雑音比)という指標値を用いて、オリジナルの状態からの劣化を判断します。この数値が大きいほど画質が良く、また通常の画像は30~50dbの範囲ならば、良好な画質であるとされています。

画質を横軸に、ファイルサイズとPSNRを縦軸にプロットすると以下のグラフとなります。



高画質向け = 80~95 ファイルサイズ軽量化 = 60~80

特に、70~95の範囲はファイルサイズとPSNRのグラフ形状が近似しています。画像サイズの減少と画像の劣化の具合がほぼ比例するので、この範囲がバランスがいいといえます。

この逆に画質30以下はほぼファイルサイズが減少しないにもかかわらず、画質が大きく劣化するためこの程度の画質を使う意味はないと考えられます。

3. 画像ファイルのサイズを小さくする：新しい画像フォーマットを使う①

新しい画像フォーマットは圧縮率が高いためデータのサイズを小さくできます。

- SVG イラストやアイコンなどに向いています。ベクター画像なので拡大縮小しても画像が劣化しない特徴があります。
- WebP 主にJPEGを置き換えることを目的とした画像フォーマットです。ビットマップであり写真に向いています。Googleの示した事例ではファイルサイズは非可逆圧縮モードで（同一画像、同等画質の）JPEGと比較して25-34%小さくなり、可逆圧縮モードでPNGと比較して28%小さくなるとされています。Wordpressの場合は同様にEWWW Image Optimizerを使うと一括で変換をかけられます。

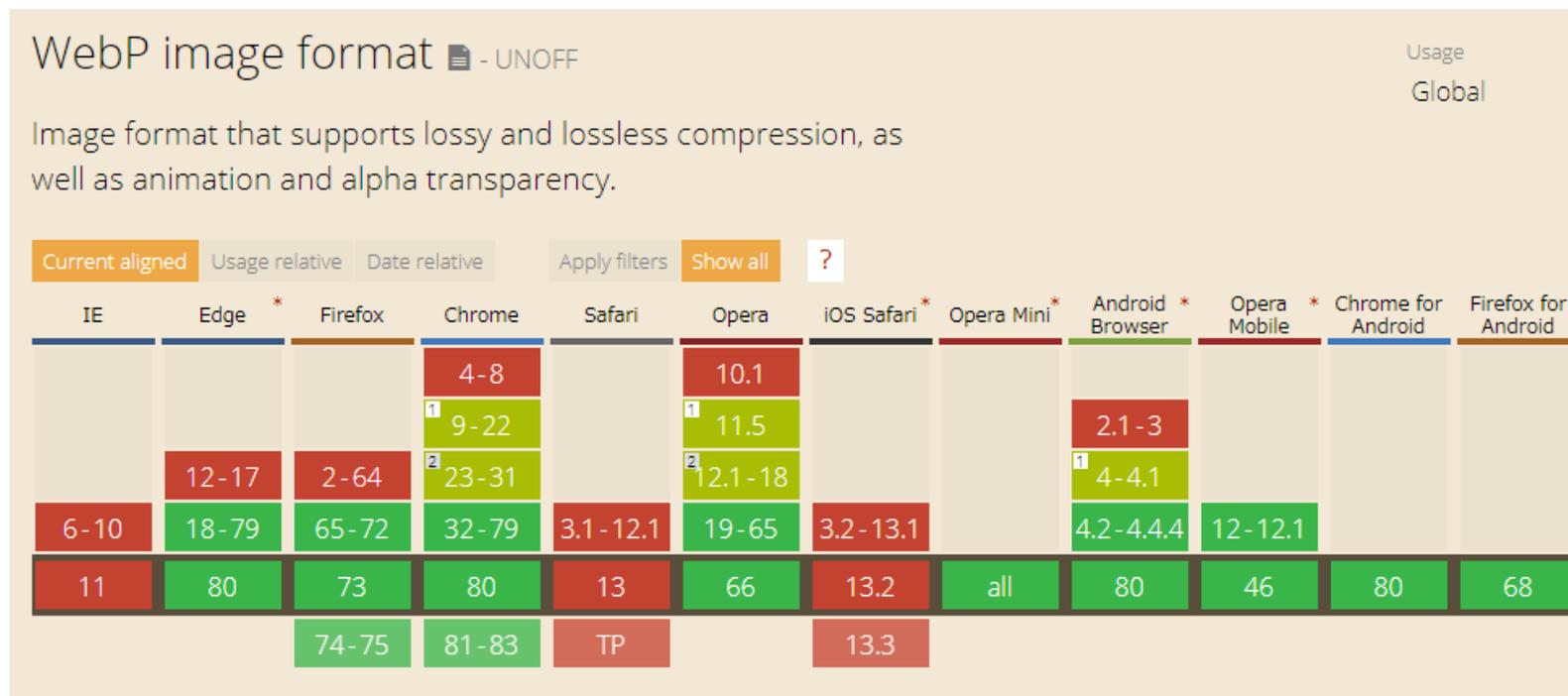


SVGやWebP画像を作る方法にはWordpressのプラグインだけではなく下記のような方法もあります。

- PhotoshopやIllustratorで書き出す
- Webにある無料の変換ツールを使う 例) <https://convertio.co/ja/png-svg/>

3. 画像ファイルのサイズを小さくする：新しい画像フォーマットを使う②

WebPフォーマットに対応していないブラウザへの対応 引用 <https://caniuse.com/#feat=webp> 2020/2月現在



ブラウザ別のWebPへの対応状況 ※数字はブラウザのバージョンを表します

SafariはWebPに対応していないことがわかります。

WebPに対応していないブラウザでアクセスした場合には、もともとのPNGやJPEGで表示するようにしておく安心です。

3. 画像ファイルのサイズを小さくする：新しい画像フォーマットを使う③

WebPフォーマットに対応していないブラウザへの対応

WordPressの場合はEWWW Image Optimizerを使うと、.htaccessに挿入する下記のコードを生成してくれます。

「リライトルールを挿入する」のボタンを押すと .htaccessに下記のコードが自動的に挿入されます。しかし、.htaccessの末尾に挿入されるので（末尾だと正常に動作しない可能性があるため）、エディタを使って.htaccessの先頭に挿入して、FTPを使ってアップロードしたほうがよいでしょう。

```
# BEGIN EWWWIO
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{HTTP_ACCEPT} image/webp
RewriteCond %{REQUEST_FILENAME} (.*)\.(jpe?g|png)$
RewriteCond %{REQUEST_FILENAME}.webp -f
RewriteCond %{QUERY_STRING} !type=original
RewriteRule (.+)\.(jpe?g|png)$ %{REQUEST_URI}.webp [T=image/webp,E=accept:1,L]
</IfModule>
```

3. 画像ファイルのサイズを小さくする：Exif情報の削除①

写真ファイルには下記のようなExif情報が付与されていることがあります。削除することでファイルサイズを小さくすることができます。

撮影日時 -日付と秒までの時刻が付加される。

位置情報（ジオタグ）

撮影方向 -電子コンパス付きのカメラやスマートフォンで撮影した場合に写真を撮影した際のカメラの方角の情報が付加される。

撮影機器のメーカー名（製造・販売元）

撮影機器のモデル名（カメラ付き携帯電話・スマートフォンの機種名など）

画像全体の解像度

水平・垂直方向の単位あたり解像度

シャッター速度

絞り（F値）

ISO感度

測光モード

フラッシュ使用の有無

露光補正ステップ値

焦点距離

色空間（カラースペース）

サムネイル（160×120画素）

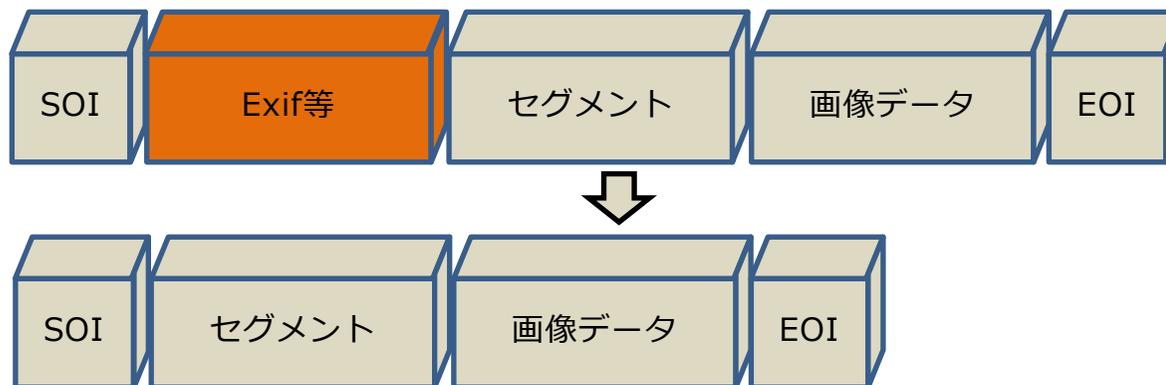
※このサイトでExif情報を確認できます。<http://exif-check.org/>

画像のファイルサイズを小さくするのみならず、位置情報については個人情報（家の中で撮影した写真など）の漏洩につながるので削除するべきです。

3. 画像ファイルのサイズを小さくする：Exif情報の削除②

画像ファイルには、撮影時の情報（カメラの機種やセッティング、撮影した場所）やサムネイル画像が含まれる場合があります。あるいは一部の画像処理ソフトは、編集時の情報を“Exif”として記録する場合があります。**これらの情報は通常は画像の品質に大きな影響を与えません。**

Exifデータを削除することでデータを軽くできますが、画像編集ソフトにより、これを残す、捨てる、という挙動は異なります。



- ・ SOI = データの始まり
- ・ セグメント = 画像圧縮に不可欠な情報（量子化テーブル、ハフマンテーブルなど）
- ・ EOI = データの終了

3. 画像ファイルのサイズを小さくする：Exif情報の削除③

JPEGの終端情報(EOI)以降に付与される無効なデータを消す

JPEGに関する仕様のうち、デジタルカメラ向けファイルフォーマット規格 DCF2.0 で「主画像のEOI(End of Image: 画像の終端記号) 以降の解釈できないデータは読み飛ばすことを推奨する」と規定しています。つまり、EOI以降に画像以外のデータが存在することを許容しています。

このために撮影機器によっては EOI 以降に、表示には使用しない不要なデータが付与されることがあります。

また、EOI以降のデータの取り扱いについては画像編集ソフトによっても異なります。一部のソフトウェアではEOI以降のデータを消さないケースもあります。



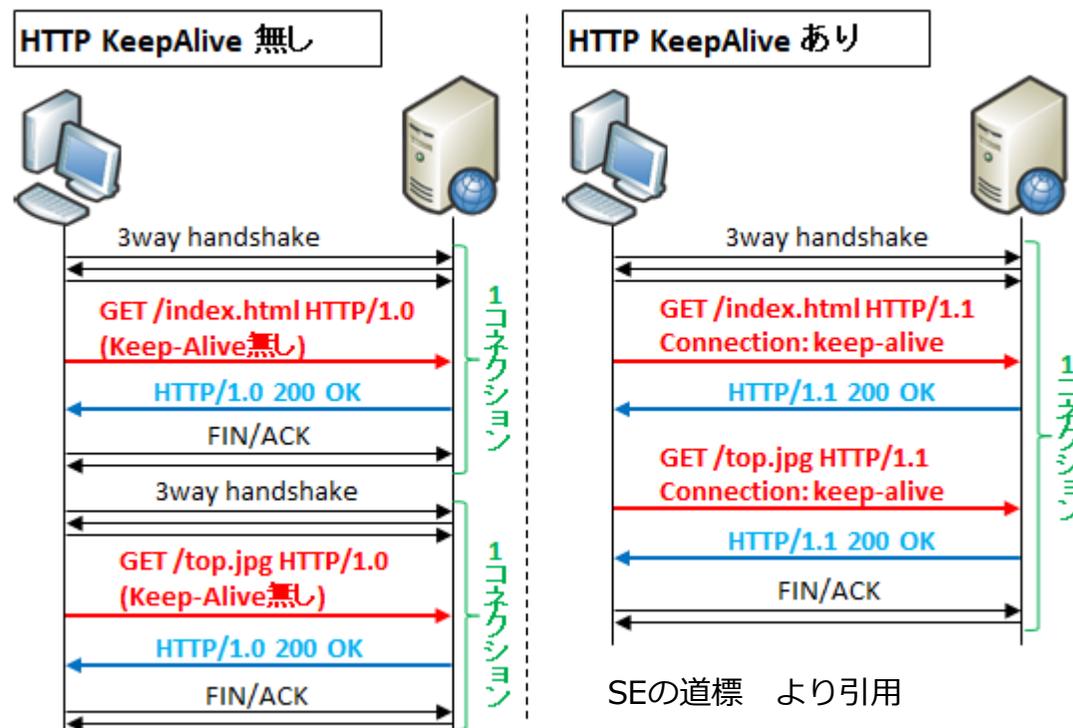
4. Keep aliveの有効化①

Webページは数多くのファイルの読み込みを経て初めて表示することができます。通常は数十程度のファイル数が必要です。Keep aliveの設定をしないと1つのファイルの読み込みが終わったらサーバとパソコンやスマホ間の通信を切断します。

次のファイルを取得するときにもまたサーバとの通信を開始するための手続きを行うこととなります。

Keep aliveの設定をし一度通信開始の手続きを行えば、ずっとサーバとの接続が維持され、都度都度手続きを行う必要がなくなるので高速化します。

※Keep aliveの設定はサーバにより異なります。サーバ内にて有効化するだけで簡単に設定できる場合もありますので、まずは使用しているサーバをご確認ください。



5. ブラウザのキャッシュの有効化

サーバから一度パソコンやスマホに読み込んだファイルについて、保存をしておき再利用するように指示します。これも設定だけすれば効果が出るのでおすすめします。

.htaccess に書き加えるだけで設定は完了です。

6. のgzip圧縮も.htaccessの編集であるため、6に合わせて方法を記載します。

6. gzip圧縮の有効化

HTML、CSS、JavaScript、XML、WEBフォントファイルなどをサーバから送信する際に、自動的に圧縮してデータ量を減らすことで高速化します。

5. のブラウザのキャッシュの有効化と合わせて .htaccess に左記の情報を書き加えます。

この情報はWeb上に多数あるので、信頼できそうなページからダウンロードして編集すればよいでしょう。

```
1 AddType image/x-icon .ico
2
3 <IfModule mod_deflate.c>
4 SetOutputFilter DEFLATE
5 BrowserMatch ^Mozilla/4 gzip-only-text/html
6 BrowserMatch ^Mozilla/4.[0-9] no-gzip
7 BrowserMatch \bMSIE[E] !no-gzip !gzip-only-text/html
8 SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png|ico)$ no-gzip dont-vary
9 SetEnvIfNoCase Request_URI _\.txt$ no-gzip
10 #DeflateCompressionLevel 4
11 AddOutputFilterByType DEFLATE text/plain
12 AddOutputFilterByType DEFLATE text/html
13 AddOutputFilterByType DEFLATE text/xml
14 AddOutputFilterByType DEFLATE text/css
15 AddOutputFilterByType DEFLATE application/xhtml+xml
16 AddOutputFilterByType DEFLATE application/xml
17 AddOutputFilterByType DEFLATE application/rss+xml
18 AddOutputFilterByType DEFLATE application/atom+xml
19 AddOutputFilterByType DEFLATE application/x-javascript
20 AddOutputFilterByType DEFLATE application/x-httpd-php
21 </IfModule>
22
23 <ifModule mod_expires.c>
24 ExpiresActive On
25 ExpiresDefault "access plus 1 seconds"
26 ExpiresByType text/html "access plus 1 seconds"
27 ExpiresByType image/gif "access plus 2592000 seconds"
28 ExpiresByType image/jpeg "access plus 2592000 seconds"
29 ExpiresByType image/png "access plus 2592000 seconds"
30 ExpiresByType image/x-icon "access plus 2592000 seconds"
31 ExpiresByType text/css "access plus 604800 seconds"
32 ExpiresByType text/javascript "access plus 216000 seconds"
33 ExpiresByType application/x-javascript "access plus 216000 seconds"
34 ExpiresByType application/x-shockwave-flash "access plus 216000 seconds"
35 </ifModule>
36
37 FileETag none
38
39 <FilesMatch "^(wp-config\.php|wp-mail\.php|install\.php|\.ht)">
40 order allow,deny
41 deny from all
42 </FilesMatch>
```

<http://mugcup8.com/htaccess-compress-pagespeed-insight/>
より引用

7. 遅延読み込み

画像ファイルはデータサイズが大きいため、表示速度に関わる大きな要因です。そこでページを開いたときにはまだ画面に表示されない画像（下にスクロールして初めて表示される画像）の読み込みを後回しにすることで、体感的な表示速度を大きく改善します。

- Wordpressの場合

プラグインを使うと簡便です。

Native Lazyload <https://ja.wordpress.org/plugins/native-lazyload/>
Googleが開発しているプラグインであり、実績も多く安定しています。

- Wordpress以外ライブラリ echo.js がサイズが小さく動作が速いためおすすめです。
<https://syncer.jp/echojs>
上記ページに導入方法など詳しく記載がありますが難しいものではありません。

echo.jsのサンプルデモ

<https://syncer.jp/static/html/demo/699/echo.html>

効果が非常にわかりやすいです。

8. JavaScriptの非同期化

ページの中に複数のJavaScriptが含まれている場合は（たいていはありますよね）、JavaScriptは1つ1つ原則的には読み込まれた順番に実行されます。

そのため、すべてのJavaScriptの実行が終わらない限りページの描画が終わらないこととなります。そこで、JavaScriptを非同期にすることで、複数同時に実行することを許可させ、描画を早めることができます(厳密に言えば違うのですが、そう考えてほぼ差し支えありません)。

Wordpressの場合はプラグインを使うと簡便です。

Async JavaScript <https://ja.wordpress.org/plugins/async-javascript/>
導入実績が多く頻繁にアップデート行われているため安心感があります。

Wordpressではない場合でも、JavaScriptの非同期化は難しくありません。

導入されていないのであれば導入することを推奨します。ただし、JavaScriptの実行順序が一定にならなくなるので、ページの表示が崩れることもあります。設定後は必ず正しく表示されているかの確認が必要です。

参考 <https://lazy-se.net/wp-plgin-async-javascript/>

9. AMP (Accelerated Mobile Pages) 対応①

AMPとはGoogleとTwitterが共同で構築した、モバイルでのウェブサイト閲覧を高速化するための技術・HTML規格です。

“

AMP のテクノロジーを利用しているウェブページは、従来に比べ平均 4 倍の速度で表示され、データ量も約 1/10 に抑えられるため、ほとんどのページが瞬時に表示されます。

”

Google Japan Blog 『[モバイルウェブをもっと速く](#)』より

AMPに準拠してページを構築することで、従来のWebページとは**全く違うレベル**での高速化表示が可能になります。

9. AMP (Accelerated Mobile Pages) 対応②

AMP対応ページをGoogleは優遇します。



左右にスライドするカルーセル表示



実際にページを表示するのはGoogleのサーバーです。

ブラウザが表示するURL :

<https://www.google.com/amp/s/www.nikkansports.com/m/entertainment/news/amp/202003040000092.html>

本来のURL :

<https://www.nikkansports.com/m/entertainment/news/amp/202003040000092.html>

検索結果に目立つように表示させることがあります。
そして高速なGoogleのサーバーを使って表示されます。

9. AMP (Accelerated Mobile Pages) 対応③

PageSpeed Grade	A (92%)	F (2%) -90%
YSlow Grade	A (98%)	D (67%) -31%
Fully Loaded time	3.4s	9.6s +6.2s
Total page size	957KB	6.84MB +5.90MB
Total # of requests	18	55 +37

Gtmetrixでの効果比較

Fの項目の多くがAになっており読み込み時間の65%削減につながったことがわかります。

PageSpeed	YSlow	Waterfall	Filmstrip	Graphs
Avoid a character set in the meta tag	A (100)	A (100)		
Avoid bad requests	A (100)	A (100)		
Avoid CSS @import	A (100)	A (100)		
Avoid landing page redirects	A (100)	A (100)		
Avoid Plugins (deprecated)	A (100)	A (100)		
Combine images using CSS sprites	A (100)	A (100)		
Defer parsing of JavaScript	A (100)	F (8)		
Enable compression	A (100)	A (100)		
Enable Keep-Alive	A (100)	A (100)		
Inline small CSS	A (100)	B (87)		
Inline small JavaScript	A (100)	A (100)		
Leverage browser caching	B (88)	D (54)		
Minty CSS	A (100)	A (94)		
Minty HTML	A (99)	A (98)		
Minty JavaScript	A (99)	A (93)		
Minimize redirects	A (100)	A (100)		
Minimize request size	A (100)	A (100)		
Optimize images	D (64)	A (99)		
Optimize the order of styles and scripts	A (100)	A (100)		
Prefer asynchronous resources	A (100)	A (100)		
Put CSS in the document head	A (100)	A (100)		
Serve resources from a consistent URL	A (100)	B (87)		
Serve scaled images	B (83)	F (8)		
Specify a cache validator	A (97)	A (97)		
Specify a character set early	A (100)	A (100)		
Specify a Vary: Accept-Encoding header	A (100)	A (100)		
Specify image dimensions	A (100)	A (100)		

9. AMP (Accelerated Mobile Pages) 対応④



PageSpeed Insightsにてスコアを計測



https://www.nikkansports.com/m/entertainment/news/202003040000092_m.html



非AMPページ（モバイル版）



<https://www.nikkansports.com/m/entertainment/news/amp/202003040000092.html>



AMPページ（日刊スポーツサーバー）



<https://www.google.com/amp/s/www.nikkansports.com/m/entertainment/news/amp/202003040000092.html>



AMPページ（Googleサーバー）

非AMPページに比較すると、AMPページは高速であることがわかります。Googleのサーバーを使うことによる速度向上は計測上はありませんでした。しかしながら、Googleのサーバーの効果はアクセスが集中しても速度が落ちないことが特徴です。詳細は後述のCDNにて解説します。

9. AMP (Accelerated Mobile Pages) 対応⑤

Wordpressであればプラグインを使うことで比較的簡単に実装が可能です。



The screenshot shows the WordPress.org plugin page for AMP. It features a blue lightning bolt icon in a circle. The text includes the plugin name 'AMP', a description 'Official AMP plugin, supported by the AMP team. Formerl...', and the author 'AMP Project Contributors'. There is a '今すぐ更新' (Update Now) button and a '詳細情報' (More Information) link. At the bottom, it shows a 4.5-star rating from 205 reviews, over 400,000 active installations, and a note that it is compatible with the current WordPress version.

上記プラグインはWordpress公式プラグインで、AutomatticやGoogleが開発に参加しているので安心感があります。※AutomatticはWordpress.comの運営会社

ただし、レイアウトはそのまま再現されませんし、JavaScriptで実装されている機能はそのままでは動作しないなどの違いがあります。

9. AMP (Accelerated Mobile Pages) 対応⑥：Canonical AMPについて

これからのトレンドはCanonical AMP（かもしれません）

Canonical AMPとはAMPページしかないWebページのことです。デザインの自由度が低いAMPページに全て統一してしまうことにはなりますが、以下のようなメリットがあります。

- **表示速度が高速化によるコンバージョン率の向上**
Canonical AMPを採用したAliExpressは4%コンバージョン率が向上し、1,000億円の売上向上効果があったとのことです。
- **パソコンからの閲覧でも速度向上**
通常パソコンで閲覧する場合はAMPページではない通常のページを閲覧することになるため、速度向上のメリットがありません。しかしCanonical AMPであればパソコンでも速度向上の恩恵が受けられます。
- **一貫したデザイン・ナビゲーションの提供**
通常のページとAMPページのデザインが異なると、ページ遷移する際にナビゲーションが変わったりしてユーザーに対して違和感を与えることがあります。
- **デザインの変更時の作業量の削減**
デザインの変更を行う際にAMP版と通常版があると、両方それぞれのHTMLやCSSの変更が必要になります。Canonical AMPであればAMP版だけの変更で済みます。

10. CSSスプライトの導入

よく使用するアイコンなどの小さな画像については右の画像のように1枚の画像にまとめます。

こうすることによって、画像ファイルの読み込みの回数を減らすことができます。

一度読み込んでしまえば、この画像の中のどの部分を表示するかをCSSで指示するだけで表示できるようになります。右記はYahoo!のアイコン画像の例です。Yahoo!のサービスのアイコンは1つのCSSスプライトとしてまとめられています。



実装もそれほど難しくはありませんが、画像の変更が生じた場合にはメンテナンスが面倒になるため、めったに変わることがなく、よく使う小さいサイズの画像ファイルが多数ある場合にはおすすめしたい手法です。

参考：<https://uxmilk.jp/61374>

※HTTP/2を導入した場合は、効果がない可能性が高いです。
なぜならCSSスプライトの使用は3way handshakeの回数削減及び、サーバーへのリクエスト数を減らす効果があります。しかし、HTTP/2においては3way handshakeが都度行われません、また複数のリクエストを同時に処理できるようになります。そのため、むしろCSSで画像を切り出す処理が増える分だけ逆に遅くなる可能性が高くなります。

1 1. DNSのルックアップを減らす

様々な別のドメインからのファイルの取得が必要になると、その都度そのドメインのサーバの所在を調べないといけないため、その時間がかかります。

Google Analytics google-analytics.com

Yahoo!タグマネージャー i.yimg.jp

b91.yahoo.co.jp

このような外部のドメインへのアクセスは致し方がないのですが、Wordpressのプラグインなどで使用されていることもありますので、極力減らすことが望ましいです。

12. CDN (Content Delivery Network) の導入

<https://zenlogic.jp/option/cdn/> より引用

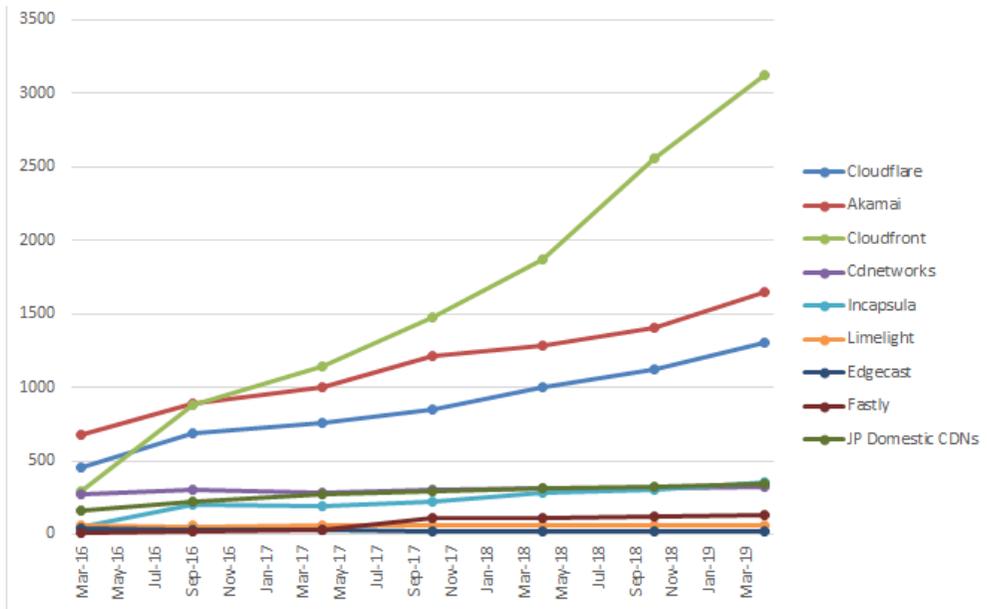


CDNとはインターネット上に多数あるキャッシュサーバに、サイトのデータをコピーしユーザがアクセスしているネットワークから近いサーバが応答するという仕組みです。CDNには以下のメリットがあります。

- 高速化：インターネットの経路の近いサーバが応答するため高速化します。ただし、もともとのサーバが高速な場合は逆に遅くなることがあります。
- 負荷の集中に強い：瞬間的に非常に多くのアクセスが発生した場合でも、多数のキャッシュサーバが分担してアクセスをさばきます。同時アクセスが数千というようなレベルであっても難なくさばくことが可能です。
- ハッキングされる危険性を大きく減らす：一般のユーザーから見えるサーバはキャッシュサーバだけになるので、おもとのコンテンツサーバが攻撃を受ける危険性が大きく減ります。

12. CDNは何を使うか

JPドメインサイトのCDNユーザ数推移



<https://tech.jstream.jp/blog/cdn/cdn-share-apl2019/> より引用

おすすめは**Cloudflare**です。費用が安く、設定が簡単です。エンジニアでなくても普通のWebサイトの製作者でも充分設定が可能です。

ユーザー登録をし、設定画面でサイト情報を入力して、DNSサーバーの向け先をCloudflareに変えるだけです。作業そのものは10分程度で終了します。

主に使われているCDNについて

- Cloudflare**
 無料でも使用できることが特徴です。独自の高速化技術Mirageを利用できるProというプランでも月額20ドルで利用できます。アクセス数が増えても料金が変わらないことも特徴です。
- Akamai**
 CDNのサービスでは世界最大手です。最も古くからあり、高速なネットワークを所有しています。最適なネットワーク経路の選択といった高速化機能に非常に優れています。月額費用は非公開で、都度見積となります。
- Cloudfront**
 Amazonが提供するサービスで、近年ではCDNの中では最もメジャーになりつつあります。アクセス数に応じて料金変動します。※目安 1ページ平均2MBで、月間100万PVある場合は約25,000円程度

最後に

高速化は非常に奥が深く、試行錯誤が必要な対策です。

しかしながら、本資料に掲載の方法はCDN、CSSスプライト以外はほぼ確実に高速化に寄与する手法です。

また技術的にも高度な手法は紹介していませんので、経験の浅いエンジニアでも実装が可能です。是非トライしてみてください。

■ その他オンラインセミナーのご案内(ぜひともご参加くださいませ)

■ 【無料】最新SEOセミナー

SEO超初心者の方からある程度知識を持った方までどなたでも安心してご参加いただける内容となっております。SEOセミナーご参加者様がのべ40,000名を突破し、参加者の99.7%の方から「満足」「大満足」のお声をいただいております。お申込みは下記より。

<https://detail-cc.com/seomaster/seminar.html>

■ 【無料】Google Analytics&Google Search Consoleセミナー

Google Analytics&Google Search Consoleの活用法を徹底解説しているセミナーです。

お申込みは下記より。

https://detail-cc.com/seomaster/seminar_analytics.html

■ 【無料】コンテンツマーケティングセミナー

コンテンツマーケティングについて基礎から分かりやすく学べるセミナーです。お申込みは下記より。

https://detail-cc.com/content_marketing/index.html

■ 【無料】Googleマイビジネス徹底活用術セミナー

Googleマイビジネスの活用方法について徹底解説しているセミナーです。お申込みは下記より。

https://detail-cc.com/business_seminar/

■ 【無料】SEO極～KIWAMI cloud～使い方実践セミナー

購入者の方はもちろん、購入をご検討中の方もご参加いただけます。お申込みは下記より。

https://detail-cc.com/seomaster/seminar_workshop.html