

WORDPRESS



ウェブサイトの スピードUP

Brian Jackson

KINSTA

Managed WordPress Hosting

WordPressウェブサイトの スピードUP

(スーパーガイド)



コードをスキャンして当ガイドの最新版をご確認ください。

<https://kinsta.com/jp/learn/speed-up-wordpress/>

発行者: Kinsta

デザイン: マヤ・サカダット

翻訳: クリスティナ・ハイナル

校正: 大平、篠原



Brian Jackson

BrianはKinstaの最高広報責任者(CMO)です。新しいオンライン成長戦略の開発、コンテンツ作成、テクニカルSEO、コミュニティとのコミュニケーションなど幅広い分野を担当しています。長い記事を書くことが大好きで、ブログ記事を2,500件以上公開しています。

Brianの最大の情熱の対象の一つは10年以上使用してきたWordPress。コンテンツ配信ネットワーク業界の経験者として現在の高性能ホスティング分野で活躍する彼は、ウェブパフォーマンスの最適化とウェブサイトのスピードアップ戦略を専門としています。

WordPressウェブサイトのスピードUP

目次

10

WordPressウェブサイトの
種類:静的または動的

13

パフォーマンスの良い
WordPressホスティング会社を
選択する

25

訪問者の近くにある
サーバーを選択する

34

プレミアムDNSは無料のDNS
より優れている

38

WordPressテーマも
影響を与える

47

WordPressプラグイン
の数を減らす

52

WordPressの最適な
設定

58

キャッシュが重要な理由
について

68

画像の最適化はマスト

77

データベースの微
調整

95

コンテンツ配信ネットワーク
(CDN)を使用する

105

メディアと電子メールを必
要に応じてオフロードする

110

ボトルネックと遅いプラグイン
の特定

129

バックエンド最適化に関する推
奨事項

143

フロントエンド最適化と
外部サービスに関する推
奨事項

170

モバイルファーストを念頭に
置いて最適化する

WordPressの最適化またはスピードアップについてのチュートリアルを長年にわたって発信してきました。しかし、必要な情報をまとめたかたちで見つけるのは、特に難しいことも。そこで、今回、このように、“WordPressターボチャージ”について知り得たすべての情報と、15年以上に渡る経験と実績を一つのガイドにまとめました。WordPressを使い始めたばかりの方でも、経験の長い開発者の方でも、このガイドがお役に立てるはずです。

Webの三分の一以上がWordPressを使用しています。これは素晴らしい事実である一方で、何千ものテーマ、プラグイン、そして技術が共存しなければならないことを意味します。WordPressの一般ユーザーにとっては、ウェブサイトが壊れたときに、何をどう処理すればいいかわからない為、厄介な話に他なりません。

今回のガイドでは、**WordPress**サイトを改善するために実施できる具体的な手順をご紹介します。便利なリソース付きですので活用ください。

第1章:

WordPress

ウェブサイトの種類:
静的または動的



最適化に取り組む前に、まずWordPressウェブサイトがそれぞれ異なる、という事実を理解しておく必要があります。課題の処理方法もそれぞれである為、ユーザーの多くがその処理方法に悩んでいます。当社では、まず該当のWordPressウェブサイトが静的であるか動的であるかを判断します。それでは、これら2種類のサイトの違いに迫りましょう。

基本的に静的なサイト

静的サイトは例えば、ブログ、中小企業のサイト、小規模のニュースサイト、個人用サイト、写真サイトなどです。静的とは、データが**あまり頻繁に変更されない**ことを意味します。(例えば一日に数回) Kinstaのウェブサイトさえ静的なウェブサイトです。

リクエストの多くはサーバーのキャッシュから恐ろしく早く配信できる為、これは非常に重要なことです。キャッシングについては、のちほど詳しくご説明します。簡単に言えば、データベースの呼び出しが少なくなり、Googleのパフォーマンスを高く維持するために必要なリソースが少なくなります。

非常に動的なサイト

その一方で、非常に動的なサイトがあります。eコマースサイト(WooCommerce、Easy Digital Downloads)、コミュニティ、メンバーシップ、フォーラム(bbPress、BuddyPress)、または学習管理システム(LMS)などがこれに該当します。動的とは、データが**頻繁に変更される**ことを意味します。(数分ごとに、あるいは数秒ごとにサーバートランザクションが行われる。)つまり、サーバーへのリクエストの中にはキャッシュから配信できないものもあり、追加のサーバーリソースとデータベースクエリが必要になります。

これらのサイトでは、**同時期に訪問するユーザーとそのセッションが多くなります**。静的なサイトである情報サイトまたは企業のWordPressサイトでは、訪問者が必要な内容を見つけるまでに5～10分くらいしか滞在しません。(通常は直帰率はさらに高く、5～10分はかなり高い数値だと言えます)ただし、動的なウェブサイトでは逆です。訪問者はインタラクションのためにウェブサイトにアクセスし、例えばオンラインコースなら何時間も滞在することも珍しいことではありません。

分かりやすい問題ですね。WordPressホスティング会社への同時訪問者の数が増えます。「キャッシュできないコンテンツ」の他、さらに悪いことに、同時訪問者の数も多すぎます！

パフォーマンスの観点でWordPressウェブサイトはそれぞれ異なります。静的なサイトと非常に動的なサイトを同じように扱わないようにしましょう。

第2章:

パフォーマンスの良い WordPressホスティング会社を 選択する



WordPressホスティング会社は、ウェブサイトのデータを保存する会社です。プラン契約をすると、お客様のすべての画像、コンテンツ、ビデオなどはホスティング会社のデータセンターにあるサーバーに保存されます。WordPressホスティング会社を使用すると、データがアクセスしやすく、管理しやすく、そして、訪問者へとルーティングしやすくなります。シンプルに聞こえるかもしれませんが…実は、もう少し複雑です。

Web上にはWordPressホスティング会社が大きく分けて3種類。3種類ともメリットとデメリットがありますので、紹介します。最初から適切なものを選択しないと、課題が次々と発生してしまいます。

1. 共有WordPressホスティング

WordPressホスティングで最も人気な手法は「共有ホスティング」と呼ばれるものです。例えば、最大手のホスティング会社ではBluehost やHostGatorといったEIGグループの会社やSiteground、GoDaddy、InMotion Hostingが共有ホスティング会社に当たります。これらの会社は通常cPanelを採用し、月額料金は3～5ドルです。

共有ホスティングプランのユーザーは遅さを感じるようになります。それは何故でしょうか。**共有ホスティング会社はサーバーを実能力以上に使ってしまう傾向があり**、ユーザーのウェブサイトもその影響を受ける場合があるからです。これらの会社は赤字にならないようにリソースを制限する為、サイトの一時停止や頻繁な500エラーの発生も珍しくありません。さらに、ウェブサイトのダウ

ンタイムも多くなります。あなたのウェブサイトが、知らぬ間に200人以上の人と同じサーバー上にあるということも起こり得るのです。他のウェブサイトで発生した性能上の問題があなたのウェブサイトに影響を与える可能性もあります。

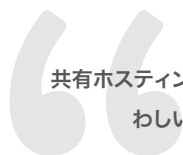


共有WordPressホスティング

全ての経費を差し引くと、どう考えても月額3ドルではホスティング会社の収益を生み出しません。特にサポートも考慮すれば、サポートチケットが一件発生しただけでも赤字になります。従って、こういった会社が収益を生み出す為にはアップセリングや隠れた料金が使われます。これらのアップセルには、マイグレーション、ドメイン登録、SSL証明書などが含まれます。もう一つよくある手法に契約割引がありますが、更新日がくると従来の金額が請求されます。

ほとんどのホスティング会社は「無限のリソース」プランというものを提供しています。あなたもおそらく目にしたことがあるでしょう。ただし、無限のリソースというものはこの世に存在しません。ホスティング会社が行うのは、多くのリソースを使ってしまうユーザーを制限することです。それに腹を立てたユーザーが解約し、リソースをあまり使用しないユーザーが利用できるスペースが増えるのです。最終的には、ホスティング会社が安いプランを勧め、お客様が多くのリソースを使わずにアップセルを購入するよう誘導するという悪質なサイクルができます。

共有ホスティングはユーザーが多くてもサポート担当者が足りない為、顧客サービスの質が低い場合があります。共有ホスティング会社は利益を得るためにコストダウンに取り組まなければならない、お客様の満足度が低いことも珍しくありません。



共有ホスティングなら、支払った金額にふさわしい品質を得られます。

2. DIY VPS WordPressホスティング

WordPressホスティングの2種類目はDIY VPS(「Do it yourself on a virtual private server」和訳:「仮想プライベートサーバーのDIY」)です。DIYを選択するのは主にブートストラップのスタートアップや、開発、サーバー管理またはWordPressの経験者です。彼らをDIY群と呼びます。彼らも通常、コスト削減には意欲的ですが、性能への関心を持ち、ビジネスの成功における性能の重要性を認識しています。一般的なセットアップには、Digital Ocean、Linode、Vultr VPSの使用と、より簡単に管理できる為のServerPilotのようなツールが含まれています。

DigitalOceanの小規模なVPSは月額5ドルから、ServerPilotの人気のプランは月額10ドルから利用できます。設定によっては、5〜15ドルまたはそれ以上の月額が発生する可能性があります。DIYのアプローチは間違いなくコスト削減に繋がる可能性がありますが、何かが壊れた場合には責任を負わない上、サーバーの最適化もご自分で行わなければなりません。

DIYのアプローチは素晴らしいかもしれませんが、気を付けなければ恨みを買うこともあります。興味本位でこのアプローチを選んではいけません。貴重な時間をご自身のビジネスを成長させるために費やすべきです。

3. マネージドWordPressホスティング

ホスティングの3種類目はKinstaも提供しているマネージドWordPressホスティングというものです。マネージドホスティング会社は、バックエンドサーバー関係の仕事をすべて実施し、必要なときにサポートを提供しています。WordPressと連携するように微調整されており、ワンクリックステージング環境および自動バックアップなどの機能を提供する場合があります。サポートチームはプラットフォームに焦点を当てており、CMSに詳しく、知識が広いです。

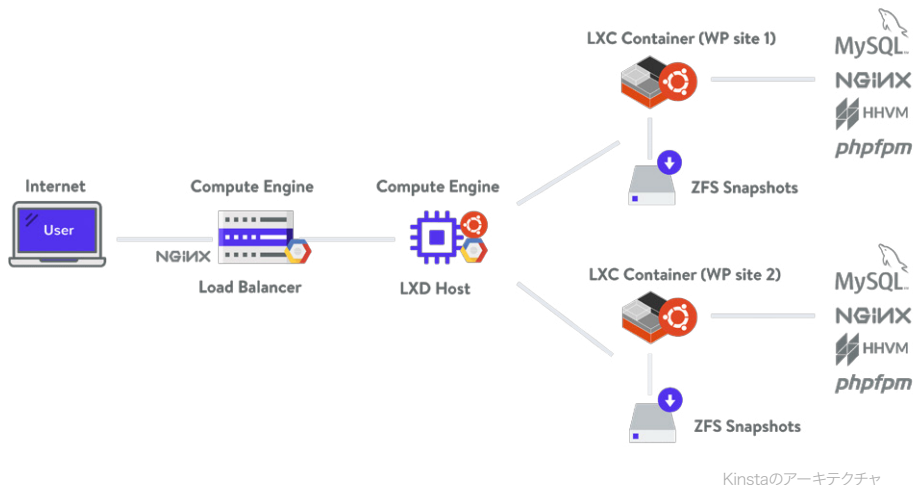
時間を節約したい方は、マネージドWordPressホスティングを選択すればよいでしょう。

マネージドWordPressホスティングのプランは、ウェブサイトの規模にもよるもので、月額額は25ドル～150ドル程度です。例えば、jQuery、Plesk、Dyn、NGINXなどの大企業やホワイトハウスでさえWordPressを使ってウェブサイトホストしています。例えばWP Engine、Flywheel、Pressable、Media Temple、Pressidium、PagelyなどのWordPressホスティング会社はあなたもご存知、もしくは既にご利用になっているかもしれません。

Kinstaはアプローチが違う

KinstaはマネージドWordPressホスティングをさらに改善してきました。当社ホスティングプラットフォームは従来のホスティングカテゴリのいずれにも該当しません。われわれのインフラストラクチャ全体はGoogle Cloud Platform上に構築され、従来の共有、VPS、専用インフラストラクチャとは一線を画します。

当社のプラットフォームのすべてのWordPressウェブサイトがその稼働に必要なソフトウェア（Linux、Nginx、PHP、MySQL）のある、隔離されている専用コンテナに保管されます。つまり、各ウェブサイトの稼働に必要なソフトウェアは100%プライベートで、お客様自身のウェブサイト間も共有されません。



各サイトコンテナは、Google Cloudの数多くのデータセンターのどちらかの仮想マシン上で動作しています。各マシンは最大96個のCPUと数百GBのRAMがあり、ハードウェアリソース（RAM / CPU）が必要になったときに、仮想マシンにより各サイトコンテナに自動的に割り当てられます。（自動スケーリングという便利な機能です。）

Review Signalは毎年、WordPressのホスティングパフォーマンスベンチマークを発表しています。Kinstaは5年連続ですべてのティアにわたり最高の企業に選ばれたのそれも、一部のプランだけではなく、StarterからEnterpriseのすべてのプランが選ばれたのです。

KinstaのLoadStorm試験もBlitz試験も基本的には完璧で、その他の試験でも問題は見つかりませんでした。Kinstaのパフォーマンスをうまく表現できるような言葉はなかなか見つかりません。



Kevin Ohashi
ReviewSignalの創業者兼務WPコンサルタント

当社の顧客サポート担当者はレベル1、レベル2といったレベル毎に分かれていません。サポートチームはWordPress開発者とLinuxホスティングエンジニアのみで構成されており、中には自サーバー管理、テーマやプラグイン作成、コア貢献の経験者が大勢います。つまり、WordPressを積極的に使用・開発する専門家に不明点を相談できるのです。

Kinstaのお客様はどなたでも、大企業やフォーチュン500をサポートしているチームメンバーとチャットすることができます。当社に応募してくる候補者から実際に採用するは1%未満です。当社の顧客サポートは世界一であると自負しております。

マネージドWordPressホスティング会社としてKinstaを選択するメリットについては、当社のウェブサイトであるkinsta.comをご確認ください。ホスティングプロバイダを選択する際には、できるだけ早いウェブサイトのスピードを実現するための下記機能があるかどうかを是非ご確認ください。

WP Engineでも細かい問題であれば簡単に解決できました。ただし、もっと複雑な問題の場合は長いやり取りが必要となり、解決に至るまでに時間がかかります。最高級のWordPressウェブサイトを運営していて緊急対応を要する案件がある場合には、これは大きな問題です。1つだけ選ぶのであれば、僕はKinstaをお勧めします。彼らは期待以上のものを提供してくれます。ウェブサイトの速度減、ダウンタイム、サポートの質などについて心配する必要はありません。



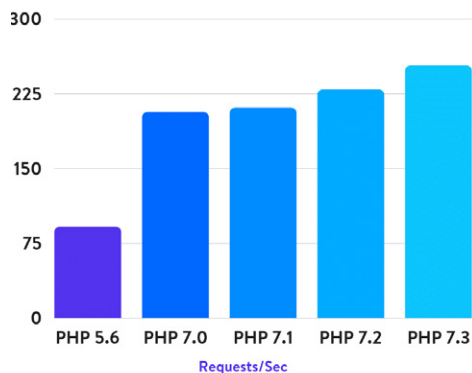
Harsh Agrawal
ShoutMeLoudの受賞歴のあるブロガー

最高のパフォーマンスのため、PHP 7以上を使用する

PHPは、主にWeb開発に使用されるオープンソースでサーバー側のスクリプトおよびプログラミング言語です。WordPressのコアソフトウェアの大部分も、プラグインとテーマもPHPで書かれており、PHPはWordPressコミュニティにとって非常に重要な言語です。ご利用のWordPressホスティング会社では少なくともPHP 7以上が利用可能であることを必ずご確認ください。

ホスティング会社毎に利用可能なPHPのバージョンがそれぞれ異なります。パフォーマンスが最も改善されたバージョンはもちろん最新のPHP 7.3です。

WordPress 5.0



当社の最新のPHPベンチマークによれば、PHP 7.3とPHP 5.6を比較すると、1秒毎に処理できるリクエスト(トランザクション)の数は3倍になっています!また、PHP 7.3はPHP 7.2よりも平均9%高速です。これはWordPressの管理ダッシュボードの反応時間にも影響を与えます。

スピードアップとセキュリティ向上の為、Kinstaでは常に最新版のPHPが利用可能です。PHPバージョン間の切り替えはワンクリックで行えます。

The screenshot displays the Kinsta dashboard interface. On the left is a dark sidebar with navigation links: Dashboard, Sites, Migrations, Kinsta DNS, Analytics, Billing, Users, Activity Log, and User Guide. The main area is titled 'editwp' and shows various site management tools. The 'PHP Engine' section is active, displaying a list of available PHP versions: PHP 7.3, PHP 7.2, PHP 7.1, PHP 7.0, and PHP 5.6. A blue arrow points to the 'Modify' button next to PHP 7.3, indicating the process of switching the site's PHP version.

最後に、PHPの代替品としてHHVMを提供しているWordPressホスティング会社にはお気を付けてください。HHVMはPHPの公式サポートを終了したため、WordPressホスティングに適したソリューションではなくなっています。

NGINXを採用しているホスティング会社を選択する

すべてのWordPressホスティング会社はその背景で、WordPressウェブサイトを稼働するためのウェブサーバーを採用します。最も一般的なのはNGINXとApacheです。

NGINXは規模を考慮したパフォーマンス最適化ができる為、NGINXを採用しているホスティング会社を強くお勧めします。NGINXは、ベンチマーク試験で、特に静的コンテンツや同時リクエストが多い状況では、他の一般的なウェブサーバーよりもパフォーマンスが優れていることが多い為、KinstaではNGINXを採用しています。

NGINXを採用している有名な会社には例えば、Autodesk、Atlassian、Intuit、T-Mobile、GitLab、DuckDuckGo、Microsoft、IBM、Google、Adobe、Salesforce、VMWare、Xerox、LinkedIn、Cisco、Facebook、Target、Citrix Systems、Twitter、Apple、Intel等があります。(出典: Sifter.com)

W3Techsによると、Apacheはすべてのウェブサイトの44.0%で使用されているため、間違いなく最も人気のあるウェブサーバーと言えます。ただし、トラフィックの最も多いウェブサイト(トップ10,000)だけを見ると、NGINXがそれらの41.9%を占めているのに対して、Apacheの占める割合はたったの18.1%です。Netflix、NASA、さらにはWordPress.comなどの最もリソースを必要とするウェブサイトがNGINXを使用しています。

ホスティング会社のネットワークが重要

WordPressホスティング会社を選択するとき、その会社が使用しているネットワークを調べる方は少ないかもしれませんが、調べてみる価値はあるでしょう。ネットワークはウェブサイトのパフォーマンスにも、WordPressダッシュボードの速さにも大きな影響を与える場合があります。コストダウンの為安いネットワークを使用し、上記の事実を教えてくれないホスティング会社が多いのも事実です。

例えば次のような点を確認してみるべきでしょう。

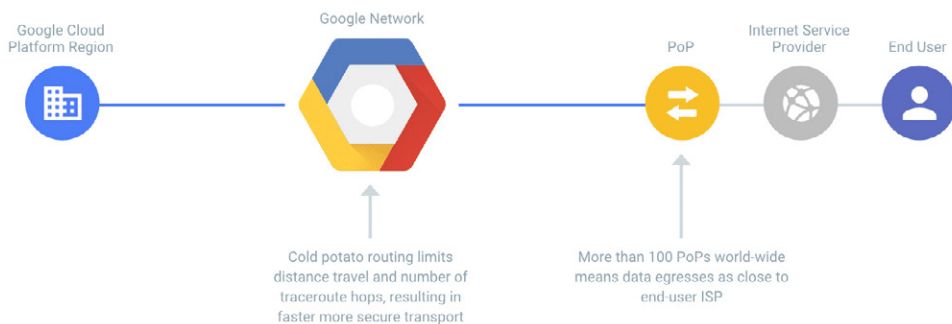
- **どのネットワークを介してデータを送信しているか。**その大部分が、公衆ISPネットワークなのか、それともGoogleやMicrosoftなどのプライベートインフラストラクチャなのか。上記の大手会社は、低レイテンシとスピードに最適化されたネットワーク、さらには会社専用の海底インターネットケーブルさえ持っています。

- **使用しているネットワークは冗長か。**ケーブルが誤って切断された場合はどうなるか。
頻繁に発生することなので確認しましょう。

2017年にGoogleは遅くて安い標準階層ネットワークの導入を発表しました。Kinstaでは、すべてのホスティングプランを対象にプレミアム階層ネットワークを採用しています。Kinstaとしては追加の費用になりますが、お客様に超高速なスピードを確保することができます。

お客様に最高のサービスを提供するために、KinstaはGoogle Cloud Platformのプレミアム階層ネットワークを使用しています。

Googleによると、プレミアム階層ネットワークは、公衆インターネット上での移動時間を短縮することにより、ネットワークパフォーマンスを向上させるとのこと。パケットは、できる限りユーザの近くにある場所でGoogleのネットワークに入り(又は出て)、Googleの基幹回線で送信され、仮想マシンに入ります。スタンダード階層なら、Googleのネットワークではなく、公衆送信 (ISP) ネットワークを介してGCPからインターネットへのトラフィックを配信します。



GCPのプレミアム階層ネットワーク

分かりやすく言い換えると：

- **プレミアム階層のパケットなら、Googleのネットワーク上に過ごす時間が長く、ネットワークをあまり出入りしない為、パフォーマンスは向上します。**(一方、費用は高くなります。)

- スタンダード階層のパケットなら、Googleのネットワーク上に過ごす時間が短く、公衆ネットワークで送信されるため、パフォーマンスが低くなります。(一方、費用は安くなります。)

上記の影響度といえば、大陸間のデータ送信の場合にプレミアム階層ネットワークはスタンダード階層ネットワークより平均して約**41%速**くなります。近くのリージョン(大陸内)のデータ送信なら、プレミアム階層は約**8%速**くなります。ネットワーキングは合計読み込み時間の一部しか占めていませんが、1ミリ秒1ミリ秒が重要な意味を持っています。

**大陸間のデータ送信の場合にGoogle Cloudのプレミアム階層ネットワーク
はスタンダード階層ネットワークより平均して約41%速く！**

冗長性も重要である為、Googleは、Googleのネットワーク上のどの2カ所間でも少なくとも3つの独立したパス(N+2冗長)を使用し、混乱が発生した場合でもトラフィックが止まらないことを担保します。

このようにネットワーキングはその背景のネットワークが重要です。ご利用のWordPressホスティング会社が評判の良い会社であり、コストダウンのために安いネットワークを採用するような会社でないことをご確認ください。

HTTP/2はマスト

HTTP/2は2015年にリリースされたウェブプロトコルで、ウェブサイトの配信をスピードアップするために設計されました。HTTP/2では、ブラウザサポートのためHTTPS(SSL)が必要です。ご利用のWordPressホスティング会社がHTTP/2をサポートしていない場合は、新しいプロバイダを探した方がよいでしょう。Web全体がHTTPSに移行する今日では、HTTP/2はあったほうが良いという性質のものではなく、マストです。

HTTP/2によるパフォーマンスの向上は、多重化、並列性、ハフマン符号化によるHPACK圧縮、ALPN拡張子、サーバープッシュなどサポートによるものです。以前はHTTPS上で動作するよう

になったときにTLSのオーバーヘッドが頻繁にありましたが、現在かなり減ってきました。近々発売されるTLS 1.3はHTTPS接続のスピードがさらにUPします。**KinstaのすべてのサーバーとCDNはHTTP/2とTLS 1.3をサポートしています。**

HTTP/2のもう一つの大きなメリットは、連結（ファイルの結合）またはドメインシャーディングの心配がないことです。これらはもはや時代遅れな最適化の手法になりました。それに加え、HTTP/3も近々登場します。

第3章:

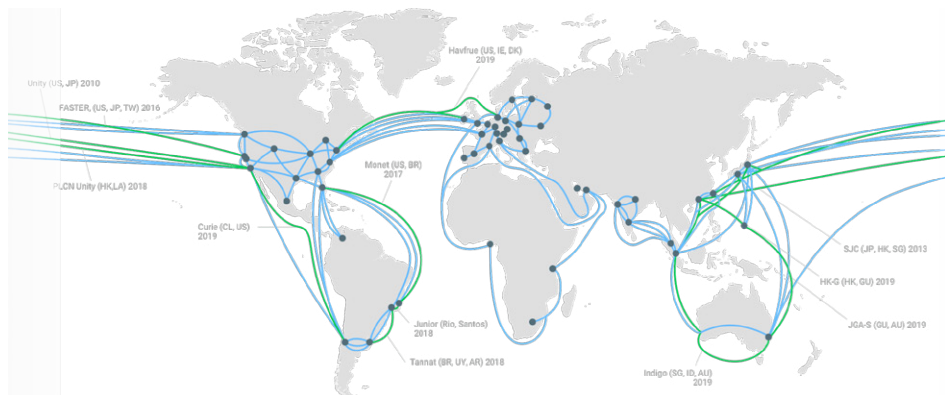
訪問者の近くにある サーバーを選択する



WordPressウェブサイトをホスティングするときに最初にすべきことの一つは、訪問者またはお客様のいる場所を特定することです。これが何故重要かというと、ウェブサイトをホスティングしている場所がネットワーク待ち時間とTTFBへの影響を与えるからです。または、SFTP速度もWordPress管理ダッシュボードも影響を受けます。

ネットワークレイテンシ(待ち時間)：ネットワーク上でのデータの送信にかかる時間およびその遅延のことを指します。言い換えると、データの packets がある地点から別の地点に移動する時間です。今日ではミリ秒単位で測定されますが、ネットワークによっては数秒かかる場合もあります。ゼロに近いほど良い値となります。

言い換えれば、距離が遠いほどレイテンシは長くなり、遅延が大きくなります。データを光ファイバケーブル上送信する上で光速であることほど重要なことはありません。たとえば、次の図はGoogle Cloud Platformネットワークの地図です。GCPが地球全体に広がる何千マイルの海底ケーブルを持っていることが、このネットワークが世界最速のネットワークの一つである理由の一つです。ただし、このような最先端のネットワークであっても、レイテンシが発生します。



ネットワークレイテンシには、次のような要因があります。

- **伝送遅延:**ワイヤレス接続や光ファイバー接続などのさまざまなメディアには送信速度の限界があるため、必ず多少の遅延が発生します。たとえば、光ファイバー接続の伝送遅延はT1回線よりも低くなります。パケットのサイズも考慮する必要があります。
- **伝播遅延:**データの packets が移動するのにかかる時間を意味します。これには、距離、ネットワーク遅延などが含まれます。
- **キューイング遅延:**帯域幅を超えると、データがホストまたはルーターで待機するキューイング遅延が発生する場合があります。ネットワークの混雑の影響を受ける可能性があります。

ネットワークレイテンシはそれほど重要ではないと主張する人もいるかもしれませんが、WordPressウェブサイトをホストする場所を選択する際に非常に重要になります。メディアとアセット(画像、JavaScript、CSS、ビデオなど)なら、より近いサーバーからコピーを配信するCDNは余分なレイテンシーの大半を補正できます。ただし、ほとんどの構成では、初期のドキュメント読み込みをホスティングサーバーにクエリする必要があります。これがレイテンシが重要な理由です。

特定の地理的な場所にいる訪問者にサービスを提供するビジネスなら、レイテンシは特に重要です。たとえば、顧客の90%がオーストラリアから来ているシドニーのeコマースストアがあるとしましょう。このビジネスの場合、ウェブサイトをヨーロッパやアメリカのサーバーではなく、オーストラリアのサーバーに配置した方が良いでしょう。

TTFB:「最初の1バイトが到着するまでの時間」(Time To First Byte)を意味します。簡単に言えば、TTFBはブラウザがサーバーからのデータの最初のバイトを受け取るまでにかかる時間のことです。そのデータを受け取るまでに時間がかかるほど、ページを表示するのにかかる時間が長くなります。こちらもゼロに近いほど良い値となります。

よくある間違いに、DNSルックアップ時間を使ってTTFBを計算するというものがあります。しかし、ネットワークにおけるTTFBの本当の計算にはネットワークレイテンシが含まれています。これには3ステップのプロセスが含まれ、その間のどこかで遅延またはレイテンシが発生する可能性があります、合計TTFBが長くなります。



1. サーバーへのリクエスト

訪問者がウェブサイトアクセスする際には、まずクライアント（ブラウザ）からサーバーにHTTPリクエストが送信されます。このステップでは、遅延を起こす要因がいくつかあります。DNSルックアップ時間が遅いと、リクエストの時間が長くなる可能性があります。サーバーが地理的に遠く離れていると、データの移動距離が長くレイテンシが発生する可能性があります。また、ファイアウォールのルールが複雑だと、ルーティング時間が長くなる可能性があります。クライアントのインターネット速度も要因の一つです。

2. サーバー処理

リクエストが送信された後、サーバーはこれを処理して応答を生成する必要があります。このステップでは、遅延を起こす要因がいくつかあります。例えば、データベースコールが遅いこと、第三者スクリプトが多すぎる、最初の応答がキャッシングされていないこと、コードまたはWordPressテーマの最適化が正しくないこと、ディスクI/Oまたはメモリなどのサーバーリソースが不十分であること等です。

3. クライアントへの応答

サーバーがリクエストを処理した後、リクエスト(最初のバイト)をクライアントに送り返す必要があります。このステップは、サーバーのネットワーク速度とクライアントのネットワーク速度の両方に大きな影響を受けます。クライアントがWi-Fiホットスポットの遅いインターネットを使用している場合、それはTTFBに反映されます。

原則として、**100ミリ秒未満のTTFBが理想的です**。Google PageSpeed Insightでは、サーバーの応答時間が200ミリ秒未満であることが推奨されています。300〜500ミリ秒の範囲は通常の値です。600ミリ秒以上の場合は、サーバー上の設定が間違っているか、より良いウェブスタックにアップグレードする必要がある可能性があります。また、TTFBを削減する方法については、下記の推奨事項に従ってください。なお、SSL/TLSネゴシエーションも重要な要素であることにご注意ください。

今回は技術的な詳細については説明しませんが、簡単に言えば、**ネットワークの待ち時間とTTFBをできるだけ減らすのが好ましいということです**。そして、これらを減らせる最も簡単な方法の一つは訪問者のいる場所に近いサーバーを選択することです。以下の手段により、最も適切なロケーションを抽出できます。

 AUDIENCE	Overview	Primary Dimension: Country City Continent Sub Continent	
	Active Users	Secondary dimension	
	Lifetime Value BETA	Country ?	Acquisition
	Cohort Analysis BETA		Users ? ↓
	Audiences NEW	32,411 % of Total: 100.00% (32,411)	
	User Explorer	1.  United States 29,470 (90.49%)	
	▶ Demographics	2.  Canada 1,329 (4.08%)	
	▶ Interests	3.  United Kingdom 521 (1.60%)	
	▼ Geo	4.  France 161 (0.49%)	
	Language	5.  Australia 157 (0.48%)	
	Location	6.  India 56 (0.17%)	
	▶ Behavior		

一つ目は、Google Analyticsでウェブサイト利用者の利用場所を確認することです。[ユーザー]→[地域]→[地域]にアクセスします。

次の例では、トラフィックの90%以上がアメリカ合衆国からのトラフィックです。したがって、WordPressウェブサイトもアメリカ合衆国にあるサーバーに配置するとよいでしょう。地元の場合、市区町村レベルまで絞り込むこともできます。しかし、通常は米国のアイオワ州などの中心的な場所をお勧めします。

手段2: eコマースデータを確認する

eコマースサイトを運営している方は、顧客の住んでいる地域も是非確認しましょう。お客様は収入を生み出す訪問者である為、最も重要な訪問者です。通常は上記のトラフィックと一致するはずですが、必ずしもそうではありません。Google Analyticsにeコマースの目的設定がある方は、その情報を上記の地域の情報と比較し、詳細な情報を得た上で判断できます。または、ご利用のeコマースプラットフォームのデータベースの情報を確認しても良いでしょう。

手段3: 簡単な待ち時間テストを実施する

各クラウドプロバイダーが現在地からの待ち時間を確認できる便利な無料のツールを提供しています。ウェブサイト用の最も適切な地域を素早く特定できます。

- **gcping.com:** (Kinstaサーバーを含むGoogle Cloud Platformのリージョンへの待ち時間の確認)
- **cloudping.info:** (Amazon Web Servicesのリージョンへの待ち時間の確認)
- **azurespeed.com:** (Azureのリージョンへの待ち時間の確認)

以下の例では、現在地からだとロサンゼルス(us-west2)が最も速いことが分かります。一方、米国全土に顧客がいる方は、西海岸と東海岸の両方からの訪問者の待ち時間を短くするために、米国のアイオワ州(us-central1)を選択するとよいでしょう。

特定のGCPリージョンまでの待ち時間を測定する

REGION	MEDIAN LATENCY
Los Angeles, USA us-west2	28ms
Global HTTP Load Balancer global	31ms
Oregon, USA us-west1	53ms
Northern Virginia, USA us-east4	82ms
Iowa, USA us-central1	83ms
South Carolina, USA us-east1	88ms

Kinstaでは、世界中の20カ所のデータセンタが利用可能です。待ち時間もTTFBも低い場所は選択しやすくネットワークホップの数も減ります。



訪問者の近くにあるサーバーを選択する

待ち時間とTTFBを減らすその他の手段

サーバーの場所以外にも待ち時間を減らす手段があります。

- WordPressウェブサイトでキャッシングを採用する。当社の試験では、キャッシングによりTTFBがなんと90%改善されました！
- コンテンツ配信ネットワーク(CDN)を利用し、世界中のPOPからキャッシュされたアセットを配信する。これにより、ホストサーバーの近くにいない訪問者のネットワーク待ち時間が減ります。
- HTTP/2プロトコルを利用して、並列化によりラウンドトリップの回数を減らす。HTTP/2はすべてのKinstaサーバーで有効になっています。
- 外部HTTPリクエストはサーバーの場所による独自の待ち時間がある為、数を減らす。
- DNSはTTFBに影響を与えるため、検索時間が短いプレミアムDNSプロバイダを使用する。
- ページが読み込まれる間、背景でタスクを実行するためにプリフェッチとプリレンダリングを利用する。

上記の内容については後程ご説明しますのでご安心ください。

SFTPがWordPress管理ダッシュボードをスピードアップする

ウェブサイトの訪問者と顧客を常に最優先すべきなのは言うまでもありません。ただし、あまり話題には上りませんが、**パフォーマンス改善が日々の仕事にも影響を与えます**。選択したデータセンターの場所は、SFTPのダウンロードとアップロードの速度(FTPクライアントでファイルを転送する速度)とWordPress管理ダッシュボードの応答性に影響を与えます。

つまり、訪問者にとって最適な場所を選択する一方で、ウェブサイト管理への影響も考慮する必要があります。WordPressのメディアライブラリへのファイルのアップロードなどの仕事は、ウェブサイトがご自分の近くにあるデータセンターでホスティングされるほど速くなります。

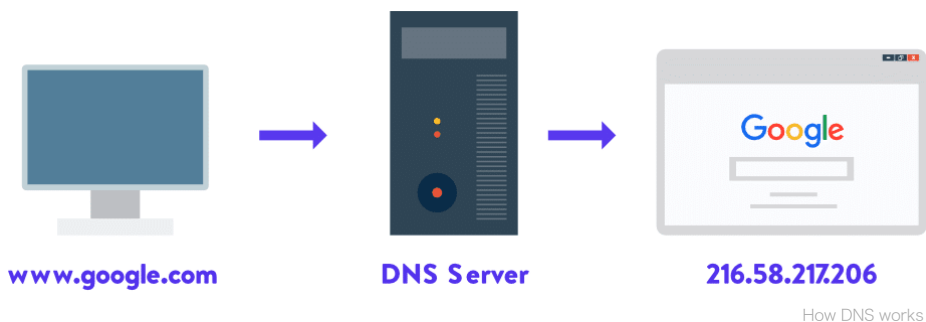
Kinstaに移行してから管理ダッシュボードの速度が驚くほど速くなったというお客様のご意見をよくいただきます。速度に影響を与える要因は多数ありますが、20ヵ所のデータセンターを持つことは間違いなくインパクトが大きいです。訪問者にもご自分にも適切なロケーションをお選びください。結局のところ、ウェブサイトの作成に多くの時間も費やすのはあなた自身であるということをお忘れなく。

第4章:

プレミアムDNSは 無料のDNSより 優れている



DNSとはDomain Name System(ドメイン名システム)の略で、ドメイン名を実際のWebサーバーと合わせることでインターネット上のトラフィックを強化するものです。例えば、kinsta.comのようなドメイン名など人間にとって分かりやすいリクエストを216.3.128.12などのコンピュータにとって分かりやすいサーバーのIPアドレスに変換します。



DNSには無料のDNSとプレミアムDNSがあります。Kinstaのすべてのお客様には、Amazon Route 53を介してのプレミアムDNSをご利用いただけます。プレミアムDNSは現在の世界においては不可欠だと考えております。

プレミアムDNSを選択する重要な理由はまず、**スピードと信頼性**です。DNSレコードを検索し、トラフィックをリダイレクトするのに、ほんの数ミリ秒であっても、時間がかかります。

通常ドメイン名登録機関から取得する無料のDNSは比較的遅いの 비해、プレミアムDNSはパフォーマンスが優れています。たとえば、当社の試験では、無料の**NameCheapのDNSがAmazon Route 53のプレミアムDNSより33%遅い**ことがわかりました。さらに、プレミアムDNSは、特にDDoS攻撃を受けた時に、セキュリティと利用可能性が高いのが特徴です。

SolveDNSのスピードテストなどのツールを使用することによりDNS検索時間を確認できます。DNSPerfもトップDNSプロバイダーのパフォーマンスデータを提供してくれます。

ドメイン登録機関の無料のDNSとプレミアムDNSの間の中間的な位置付けとして、プレミアムDNSのメリットを多く提供する無料サービスであるCloudflare DNSがあります。世界中の平均応答時間は20ミリ秒未満で、驚くほど速いサービスです。(下記を参照)

Name Server	Result	Average (milliseconds)	Standard Deviation	95% Confidence Interval
algin.ns.cloudflare.com	✔ Los Angeles: 5.65 ms	5.48 ms	2.69	[3.49, 7.47]
	✔ Dallas: 6.04 ms			
	✔ New York: 3.47 ms			
	✔ Singapore: 3.29 ms			
	✔ London: 3.81 ms			
	✔ Amsterdam: 4.04 ms			
	✔ San Francisco: 12.4 ms			
	✔ Sydney: 6.57 ms			
	✔ Tokyo: 4.02 ms			
heather.ns.cloudflare.com	✔ Los Angeles: 1.55 ms	3.44 ms	3.12	[1.13, 5.75]
	✔ Dallas: 3.98 ms			
	✔ New York: 2.06 ms			
	✔ Singapore: 2.11 ms			
	✔ London: 2.95 ms			
	✔ Amsterdam: 2.17 ms			
	✔ San Francisco: 12.04 ms			
	✔ Sydney: 1.51 ms			
	✔ Tokyo: 2.62 ms			

ただし、Cloudflareには他社のプロバイダと比較するとダウンタイムが多いというデメリットもあります。主に米国からの訪問者が多いという方は、もう一社のプレミアムDNSプロバイダーであるDNS Made Easyも検討する価値があります。この会社は過去10年間で最高のDNS稼働時間を誇る会社として有名です。

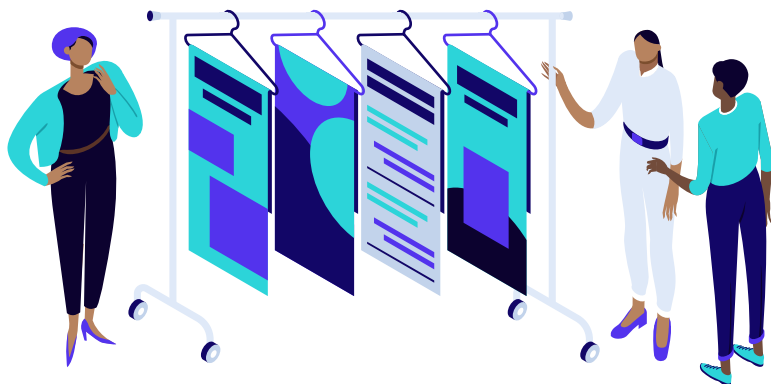
下記、各プロバイダーの過去30日間のDNSPerfによる稼働時間のデータを示しています：

- DNS Made Easy: 99.99%、これは月当たり4分23秒のダウンタイムに相当します。
- Amazon Route 53: 99.88%、これは月当たり52分35.7秒のダウンタイムに相当します。
- Cloudflare: 99.85%、これは月当たり1時間5分44.6秒のダウンタイムに相当します。

DNSプロバイダのダウンタイムはそれほど重要なのでしょうか？実は、重要な場合も、あまり重要でない場合もあります。DNSは通常、DNSレコードのTime to Live (TTL、生存時間) の値を使用してISPにキャッシュされます。したがって、DNSプロバイダが10分間停止しても、誰も気付かないでしょう。しかし、プロバイダに頻繁に長期間のダウンタイムがある場合やISPとDNSレコードの両方が非常に低いTTL値を使用している場合には、ダウンタイムは重要です。

第5章:

WordPressテーマも 影響を与える



最新のWordPressテーマは誰も利用したいものですが、採用する前に注意が必要です。無料のテーマと有料のテーマは大きく異なります。パフォーマンスにおいては、テーマに表示されるすべての要素がウェブサイトのスピードに影響を与えます。そして残念なことに、何千種類も存在するテーマの中にいいものもあれば悪いものもあります。

WordPressテーマはパフォーマンスを決定する重要な要素です。最初から正しいものを選択しましょう。

どのようなテーマを選択すればいいのでしょうか？次の2つの選択肢のいずれかをお勧めします：

- 必要な機能だけで構築されている軽量なWordPressテーマ
- 機能は多いが、未使用の機能を無効にすることが可能なWordPressテーマ

例えば、Googleフォント、Font Awesomeのアイコン、スライダー、ギャラリー、ビデオ、パララックススクリプトなどが未使用の場合に無効にすることができる機能です。事後に手動で微調整しようとするのは非常に困難です。逆に、最初から軽量なテーマや、必要機能を選択できるテーマを選択しましょう。

下記は、お勧めできるWordPressテーマです。是非、お試しください！

下記のテーマはすべてWooCommerce、Easy Digital Downloads、WPML、BuddyPress、およびbbPressとの互換性があります。以下の設定で、各テーマのスピードテストも実行しました：

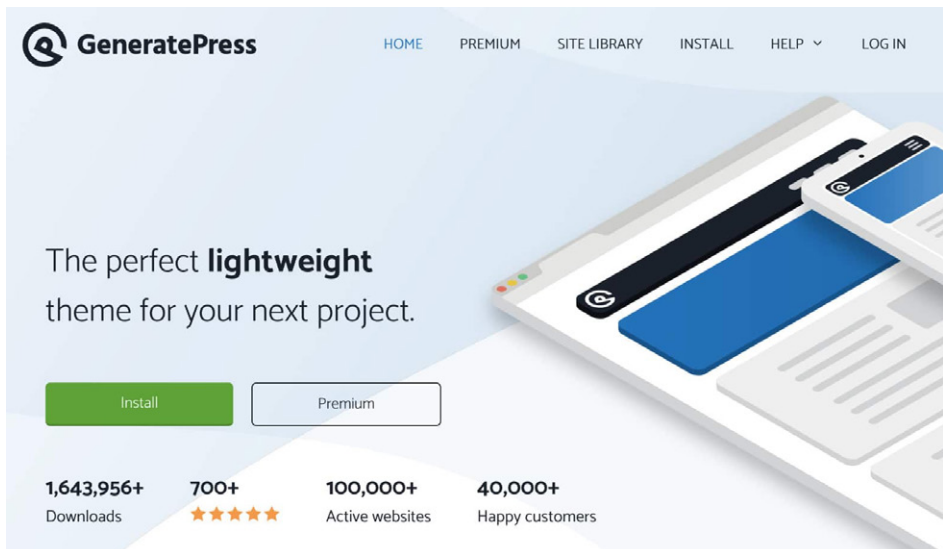
- Kinstaでホスティングし、WordPress 5.0を使用
- PHP 7.3とSSL(HTTPS)
- Kinsta CDN
- Imagifyを使用し、画像の自動圧縮を実施

GeneratePress

GeneratePressは、スピード、SEO、そして使いやすさを念頭に置いて構築された、高速で軽量（1MB未満のzipファイル）のモバイル最適化されたWordPressテーマです。カナダの開発者であるTom Usborneにより作成され、頻繁に更新され、サポートも優れています。Kinstaチームの中にも、独自のプロジェクトにGeneratePressを使用しているメンバーがいます。

無料版とプレミアム版があります。WordPress.orgを確認すると、無料版には現在20万以上のアクティブインストールと200万以上のダウンロードがあり、評価は5つ星のうち5つです。（850名以上が5つ星を付けています）。

GeneratePressのメリットの一つは、すべてのオプションがWordPress自体のカスタマイザを使用していることです。つまり、公開ボタンを押す前に、変更が表示され確認できます。また、新しいテーマコントロールパネルの使い方を学ばずに済みます。



The header of the GeneratePress website features the logo on the left and navigation links (HOME, PREMIUM, SITE LIBRARY, INSTALL, HELP, LOG IN) on the right. The hero section has a large heading, two buttons (Install, Premium), and four statistics.

GeneratePress

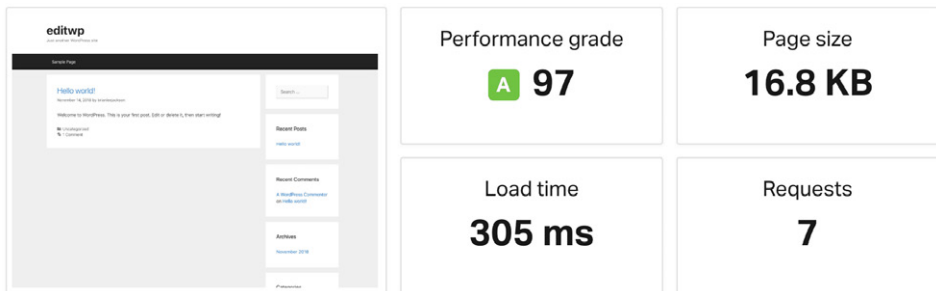
HOME PREMIUM SITE LIBRARY INSTALL HELP LOG IN

The perfect **lightweight** theme for your next project.

[Install](#) [Premium](#)

1,643,956+ Downloads
700+ ★★★★★
100,000+ Active websites
40,000+ Happy customers

一体どれくらい速いのでしょうか?GeneratePressを新しくインストールし、Pingdomスピードテストを5件実行して平均を取りました。**合計ページサイズはわずか16.8 KBで、合計読み込み時間は305ミリ秒**でした。該当のテーマの生のパフォーマンスを確認するためのベースラインテストを行うのは非常に有効です。



This section displays a screenshot of the WordPress admin interface on the left and four performance metrics on the right.

editwp
WordPress 5.9.0

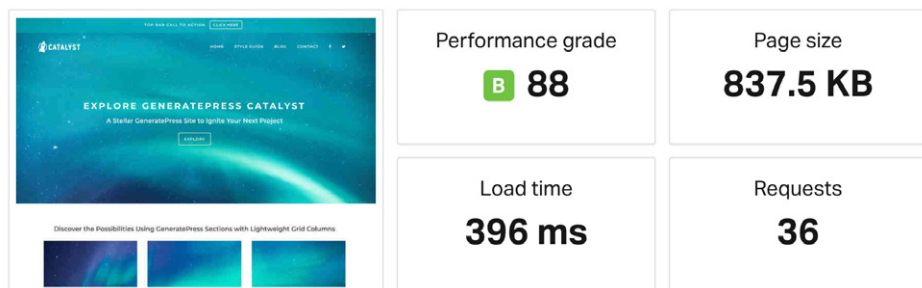
Performance grade **A 97**

Page size **16.8 KB**

Load time **305 ms**

Requests **7**

その後、GeneratePressのサイトライブラリの作成済みのテーマを一つ使用してテストをもう一連行いました。作成済みのテーマには画像、背景、新しいセクションなどが含まれています。このテストの結果も400ミリ秒未満でした。



もちろん、実際の環境では、Google Analytics、Facebookのリマーケティングピクセル、Hotjarなども実行中でしょう。ただし、それでも1秒未満という目標を容易に達成できるはずです。

下記でWordPressを最適化しスピードアップする方法をさらにご紹介します。

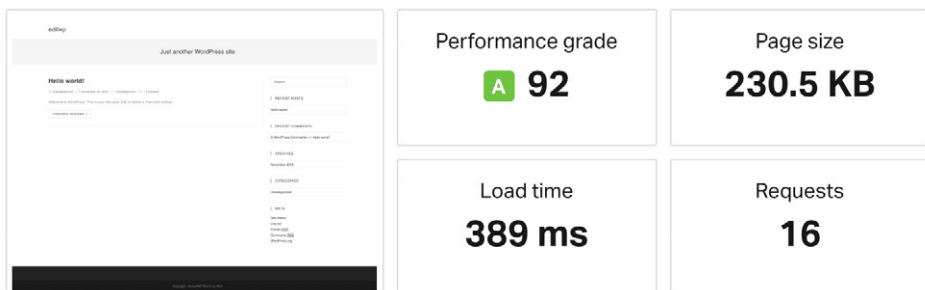
OceanWP

The banner for the OceanWP website features a dark background with a large, faint image of a sea turtle. At the top left is the 'OceanWP.' logo, and to its right are navigation links: 'Demos', 'Extensions', 'Blog', 'Support', and 'Download'. Further right is a 'Sign In / Sign Up' link. The main headline reads 'Create Beautiful Websites', with 'Beautiful' underlined in blue. Below this, a text block states: 'With 899,712 downloads, OceanWP is the fastest growing theme for WordPress. It's the favorite choice of thousands of developers and hobby-users.' A prominent blue button with the WordPress logo and the text 'FREE DOWNLOAD' is centered below the text. At the bottom, there are two stylized illustrations: on the left, a desk with a computer monitor and the word 'DEVELOPER' in a blue box; on the right, a clock icon and the word 'HOBBYIST' in a blue box.

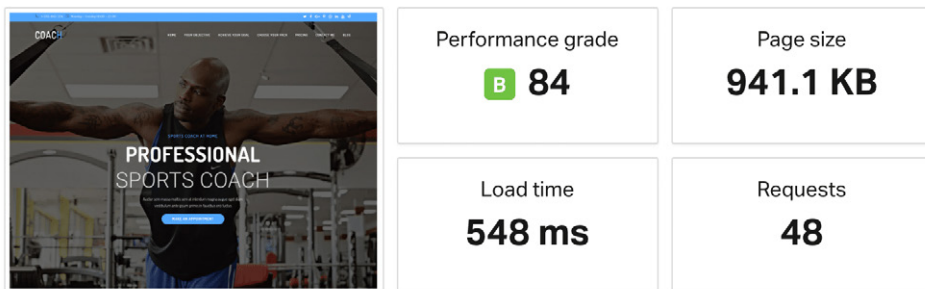
OceanWPテーマは、軽量で非常に拡張性が高く、ブログ、ポートフォリオ、ビジネスウェブサイト、WooCommerceストアなど、どのような種類のウェブサイトも美しくプロフェッショナルなデザインで作成できるテーマです。Nicolas Lecocqにより作成され、頻繁に更新され、サポートも優れています。

GeneratePressと同様に、無料版とプレミアム版の両方があります。WordPress.orgを確認すると、無料版には現在40万以上のアクティブインストールがあり、評価は5つ星のうち5つです。(2,600名以上が5つ星を付けています)。

こちらはどれくらい速いのでしょうか？OceanWPを新しくインストールし、Pingdomスピードテストを5件実行して平均を取りました。**合計ページサイズはわずか230.8 KBで、合計読み込み時間は389ミリ秒**でした。スクリプトはOceanWPの方が少し大きくなっていますが、心配すべきほどのものではありません。



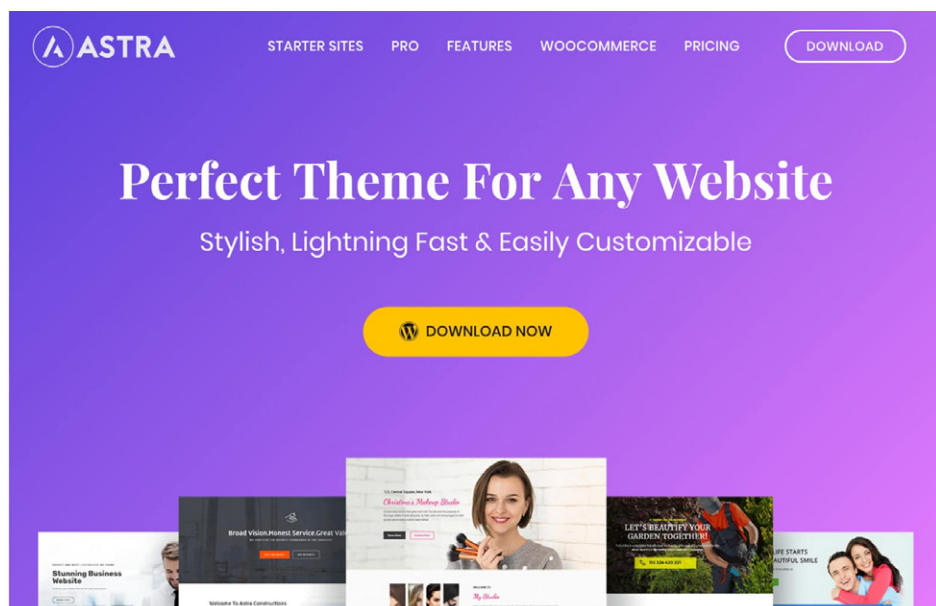
その後、OceanWPのサイトライブラリのデモテーマの一つを使用してテストをもう一連行いました。作成済みのテーマには画像、背景、新しいセクションと必須のElementorページビルダープラグインが含まれています。このテストの結果も600ミリ秒未満でした。



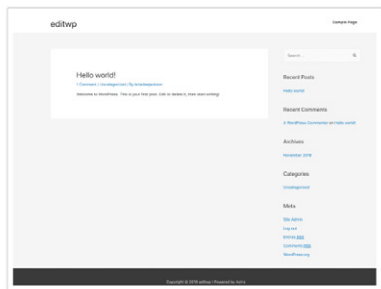
Astra

Astraは、高速で完全にカスタマイズ可能で美しく、ブログ、個人のポートフォリオ、ビジネスウェブサイト、およびWooCommerceストアなどに適切なテーマです。非常に軽量（フロントエンドで50 KB未満）であり、比類のないスピードを誇ります。Brainstorm Forceのチームにより作成され、頻繁に更新され、サポートも優れています。このチームは人気のAll In One Schema Rich Snippetsプラグインの作成者でもあるため、あなたもおそらくご存じでしょう。

GeneratePressやOceanWPと同様に、無料版とプレミアム版の両方があります。WordPress.orgを確認すると、無料版には現在40万以上のアクティブインストールと160万以上のダウンロードがあり、評価は5つ星のうち5つです（2,500名以上が5つ星を付けています）。



こちらは、どれくらい速いのでしょうか？Astraを新しくインストールし、Pingdomスピードテストを5件実行して平均を取りました。**合計ページサイズはわずか26.6 KBで、合計読み込み時間は243ミリ秒**でした。



Performance grade

A 98

Page size

26.6 KB

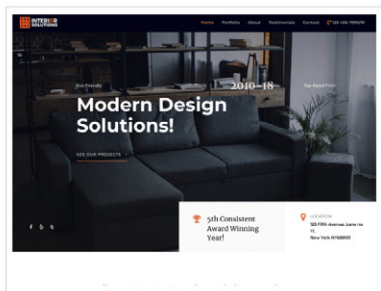
Load time

243 ms

Requests

5

その後、Astraスターキットのサイトライブラリのデモテーマの一つを使用してテストをもう一回行いました。作成済みのテーマには画像、背景、新しいセクションと必須のElementorページビルダープラグインが含まれています。このテストの結果も700ミリ秒未満でした。注：このデモの画像は完全に圧縮されていますが、最初から非常に高解像度の画像を使用しました。



Performance grade

B 86

Page size

1.6 MB

Load time

652 ms

Requests

54

なお、完全に正確な並列での比較は不可能ですので、その点はご承知おきください。上記の試験結果で示そうとしている重要なことはあくまでもこれらのWordPressテーマのいずれも、標準設定のままでも、フルデモのままでも、驚くほど速いということです！

ページビルダーの注意事項

お気づきのとおり、OceanWPとAstraはどちらもサイトライブラリのテーマを使用するにはページビルダーが必要です。ページビルダープラグインを使用する際の注意事項は下記のとおりです。

- ウェブサイトの読み込み時間が長くなるページビルダーもあります。それはコードなしでも問題なく動作するために追加のCSSとJSを読み込む必要があるためです。ページビルダーをインストールする前後に必ずWordPressウェブサイトのスピードテストを行いましょう。

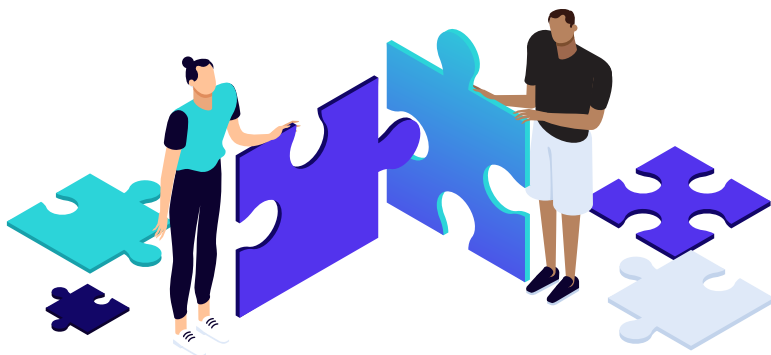
- ページビルダーをインストールするとおそらく長期間使用することになるでしょう。そのため、定期的に更新され、かつ必要なものがすべて揃っているものを選択しましょう。

そうは言っても、Elementor及びBeaver Builderなどのページビルダーは便利だと言えます。両方はパフォーマンスを念頭に置いて開発されているものです。これらのプラグインを使用すると、希望する内容を何でも構築できるため、検討する価値はあります。また、これらを使用しない場合には同じ機能の為に5つ以上のプラグインも使用しなければならないこともあるという観点ではこれらを使った方が速いとも言えるでしょう。

ただし、ページビルダープラグインが不要な方は、インストールしないことをお勧めします。また、今後の新しいGutenbergエディタがサイトデザインにどのような役割を果たすのか、にも期待したいところです。

第6章:

WordPressプラグインの 数を減らす



WordPressプラグインについても言及したいと思います。WordPressウェブサイトのスピードが落ちる恐れがある為、プラグインをたくさんインストールしすぎない方がいいと言われたことあるかもしれません。これは事実ですが、数は最も重要な要素ではありません。プラグインの数は、プラグインの品質ほど重要ではありません。

Jテーマと同様に、そのプラグインがパフォーマンスを念頭に置いて開発されたかどうか重要です。Kinstaのお客様では、30〜40のプラグインがあっても、ウェブサイトが1秒未満で読み込まれるケースは多く見られます。

ついウェブサイトにコードを追加したくなりがちですが、次の理由で必ずしも実用的ではありません。

1. 自分でコードをメンテナンスし、標準が変わるときにはその更新に従う必要があります。忙しい時間を割いて自分で行うよりも、標準を最もよく知っている開発者にその仕事を一任すべきでしょう。
2. 適切にコーディングされたプラグインには無駄な部分はありません。
3. WordPressコミュニティには、開発者のように技術に詳しい人間はそう多くありません。プラグインは問題解決に役立つソリューションです。

もちろん、そうは言っても、利用しない方がいいプラグインもあります。Kinstaでも最悪なプラグインをいくつか経験してきました。全てではないにせよ、Kinstaで禁止されているプラグインの多くは直接パフォーマンスの異常を引き起こします。

ただし、WordPressの最も人気な特徴の一つは、サードパーティ製プラグインの幅広いライブラリでしょう。しかし、WordPress.orgだけでも56,000以上の無料のプラグインがあり、他のウェブサイトにも数多く存在するため、必要とするプラグインが極めて見つけにくい場合もあります。

この記事では当社のスタッフが日常的に使用するもののみをご紹介します。もちろん、Kinstaのウェブサイトも皆様と同様にWordPressプラグインを使用しています。また、Kinstaのチームメンバーの多くは、プラグイン開発または販売に取り組んでいます。

WordPressプラグインの課題

WordPressプラグインの大きな課題は、アンインストールです。WordPressのプラグインやテーマをインストールすると、データはデータベースに保存されます。そこでの課題は、標準の手順でプラグインを削除するときにデータベースにテーブル及び行などが残ってしまうこと。時間が経つにつれ、これは大量のデータの蓄積となり、ウェブサイトのスピードが遅くなる場合さえあります。下記の例では、Wordfenceセキュリティプラグインをアンインストールしましたが、データベースにはテーブルが24個残っていました（下記参照）。wp_optionsテーブルのデータの背後にあるならば、さらに問題です。

☐	wp_wfBadLeechers	★							0	InnoDB	latin1_swedish_ci	16	KiB
☐	wp_wfBlockedIPLog	★							35	InnoDB	utf8_general_ci	16	KiB
☐	wp_wfBlocks	★							32	InnoDB	utf8_general_ci	32	KiB
☐	wp_wfBlocksAdv	★							0	InnoDB	utf8_general_ci	16	KiB
☐	wp_wfConfig	★							156	InnoDB	utf8_general_ci	16	KiB
☐	wp_wfCrawlers	★							10	InnoDB	latin1_swedish_ci	16	KiB
☐	wp_wfFileMods	★							0	InnoDB	utf8_general_ci	16	KiB
☐	wp_wfHits	★							1,907	InnoDB	latin1_swedish_ci	1.3	MiB
☐	wp_wfHoover	★							0	InnoDB	utf8_general_ci	32	KiB
☐	wp_wfIssues	★							0	InnoDB	utf8_general_ci	16	KiB
☐	wp_wfKnownFileList	★							0	InnoDB	utf8_general_ci	16	KiB
☐	wp_wfLeechers	★							156,922	InnoDB	latin1_swedish_ci	13.5	MiB
☐	wp_wfLockedOut	★							0	InnoDB	utf8_general_ci	16	KiB
☐	wp_wfLocs	★							4	InnoDB	utf8_general_ci	16	KiB

☐ wp_wfLogins	★							5 InnoDB utf8_general_ci	48 K1B
☐ wp_wfNet404s	★							1 InnoDB utf8_general_ci	32 K1B
☐ wp_wfNotifications	★							2 InnoDB utf8_general_ci	16 K1B
☐ wp_wfPendingIssues	★							0 InnoDB utf8_general_ci	16 K1B
☐ wp_wfReverseCache	★							2 InnoDB latin1_swedish_ci	16 K1B
☐ wp_wfScanners	★							618 InnoDB latin1_swedish_ci	64 K1B
☐ wp_wfSNIPCache	★							0 InnoDB utf8_general_ci	64 K1B
☐ wp_wfStatus	★							40 InnoDB utf8_general_ci	48 K1B
☐ wp_wfThrottleLog	★							0 InnoDB utf8_general_ci	32 K1B
☐ wp_wfVulnScanners	★							0 InnoDB latin1_swedish_ci	16 K1B

WordFenceのテーブル

多くのプラグインは、データベース以外にもフォルダやファイルを残します。経験から言うと、ロギング用に追加のディレクトリを作成するセキュリティプラグインまたはキャッシングプラグインでよくあることです。たとえば、Wordfenceプラグインが削除された後は、wp-contentディレクトリに「wflogs」フォルダが残ります。もちろん、Wordfenceだけの話ではありません。市場に流通するプラグインとテーマの多くで同じようなことが起きます。

Remote site: /www/ /public/wp-content/wflogs			
		- upgrade - uploads - wflogs	
Filename	File...	Filetype	Last modified
..			
.htaccess	133	HTACCESS File	7/24/2017 10:54:49 AM
attack-data.php	40,0...	PHP File	7/24/2017 10:54:49 AM
config.php	594,...	PHP File	7/24/2017 10:57:30 AM
ips.php	51	PHP File	7/24/2017 10:54:50 AM
rules.php	101,...	PHP File	7/24/2017 10:54:59 AM
wafRules.rules	44,2...	RULES File	7/24/2017 10:54:59 AM

WordFenceのログ

なぜ開発者はこのようなことをするのでしょうか？

それでは、開発者はなぜプラグインをアンインストールまたは削除するときの自己掃除オプションを付けないのでしょうか？実は、付けてはいるのですが、あまり明確にしていないのです。その理由は以下の通りです。

- 1. ユーザーの設定を保持したいため。**WordPressプラグインを削除しても、後でもう一度インストールするときに、すべての設定とデータはそのまま残ります。非常に便利ですが、あまり効率的ではありません。
- 2. パフォーマンスを気にしないため。**テーブルを残してもパフォーマンスに影響しない、という開発者もいます。しかし、10年にわたり数千の行やテーブルを生成し、何百ものプラグインを使用したウェブサイトを想像してみてください。データベースのクエリはWordPressウェブサイトのパフォーマンスに大きな影響を与えます。そして、開発者が十分注意していないとプラグインはリクエストを数多く生成することがあります。上手に書かれたプラグインなら結び付けられているテーブルや行のみをクエリしますが、そうとは限りません。Kinstaでは、wp_optionsテーブルに不要な自動読み込みデータが残っていたために長いデータベースクエリが発生しウェブサイトが破壊されるという事態を経験したことがあります。
- 3. 開発者のミス。**WordPressプラグインハンドブックにさえ、「経験の浅い開発者は、誤ってこの目的のために非アクティブ化を使用することがある」とあります。

もちろん、プラグインを正しく削除する方法もあります。詳細なチュートリアルは、当社のウェブサイト <https://kinsta.com/speed-up-wordpress> でご覧いただけます。

- WordPressプラグインをアンインストールする方法について（正しい方法）
- 残ってしまったテーブルを手動で削除する方法について

第7章:

WordPressの 最適な設定



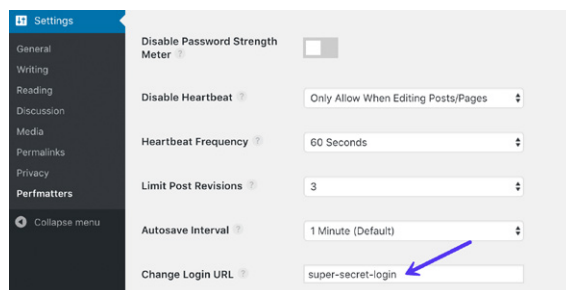
次に最適なWordPressの設定について見ていきましょう。WordPressウェブサイトのスピードアップのための改善方法をご紹介します。下記の改善事項の多くは非常に微妙な変更ですが、その積み重ねが改善へと繋がります。

WordPressのログインURLを変更する

デフォルトでは、WordPressウェブサイトのログインURLはdomain.com/wp-adminです。しかし、ボット、ハッカー、スクリプトなどもこのことをよく知っています。URLを変更することで、ブルートフォース攻撃からウェブサイトを守ることができます。

または、WordPressのログインURLを変更することで、「429 リクエストが多すぎる」などのよくあるエラーも予防できます。もちろん完全な解決策にはなりませんが、有効な手法です。

WordPressのログインURLを変更するには、次のプラグインのどちらかを使用することをお勧めします。



PerfmattersでWordPressログインURLを変更する

- **WPS Hide Login**
(WordPress.orgより無料で入手可能)
- **Perfmatters** (プレミアムだが、その他のパフォーマンス最適化機能もあり。Kinstaのチームメンバーにより開発。perfmatters.ioより入手可能)

プラグインとテーマの自動更新機能の無効化または微調整

WordPress管理ダッシュボードが遅いことの原因として、ネットワーク、データセンターの場所、さらにPHPのバージョンなどが考えられます。そして、あまり知られていないもう一つの要因は、バックグラウンドで実行されるWordPressの自動更新機能です。WordPressプラグイン及びテーマを数多く持っていることが悪影響を及ぼす一例です。WeFosterのチームはこの課題に「Third Party Plugin Update Check Syndrome」(和訳: サードパーティ製プラグインの自動更新シンドローム)またはTPPUCSという名前を付けています。

問題なのは、WordPressの組み込みの自動更新機能が外部のGETリクエストをすることです。これは定期的に起きる場合と頻繁に起きる場合があります。頻繁に発生すると、管理ダッシュボードに遅延が見られるようになります。

これはWordPressの自動更新機能の構造による問題です。WordPress管理ダッシュボードの読み込み時間が遅いことにお悩みの方は、次の対策を試してみると良いでしょう。それは自動更新機能を無効にするという方法です。注: アップデートを手動で確認する予定がある場合にのみ行ってください。アップデートの多くにはセキュリティ改善及びバグ修正が含まれています。

更新機能を無効にするには、次のプラグインのどちらかをお勧めします。(どちらもWordPress.orgで入手可能)

- **Disable All WordPress Updates:** 完全に無料で、設定機能はありません。その名の通り、WordPressのすべてのアップデートを無効にします。

- **Easy Updates Manager:**アップデートを実行するかどうかを選択できる機能が
あります。コアバージョンは無料です。

カレンダーのリマインダーを設定し、週に一度プラグインを無効にし、アップデートがあるかを確認し、更新後に再びプラグインを有効にすると良いでしょう。

ピンバックを無効にする

ピンバックとは、他のブログであなたのウェブサイトへのリンクが貼られたときに作成される自動化されたコメントのことです。ブログ内に自分の記事をリンクしたときに作成される自己ピンバックもあります。

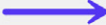
無駄なクエリが生成されスパムの数が増えるだけなので、無効にすることをお勧めします。特にトラフィックの多いウェブサイトはそうですが、WordPressサイトのコール回数は少ないほどいいものです。それに、自己ピンバックは迷惑でしかありません。ピンバックを無効にするには、以下の手順に従ってください。

ステップ1:他のブログからのピンバックを無効にする

WordPressダッシュボードで、「設定」→「ディスカッション」にアクセスします。「ディスカッション設定」で、「他のブログからの通知 (ピンバック・トラックバック) を受け付ける」のチェックボックスをオフにします。

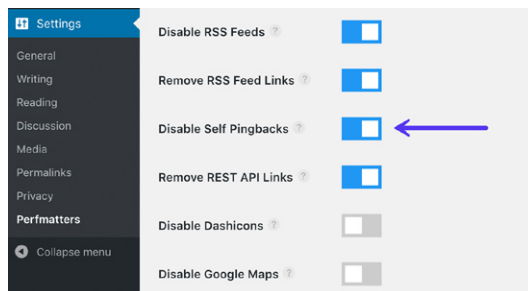
Discussion Settings

Default article settings

- ☐ Attempt to notify any blogs linked to from the article
-  ☐ Allow link notifications from other blogs (pingbacks and trackbacks) on new articles
- ☒ Allow people to post comments on new articles

(These settings may be overridden for individual articles.)

ステップ2: 自己ピンバックを無効にする



Perfmattersで自己ピンバックを無効にする

自己ピンバックを無効にするには、無料のNo Self Pingsプラグイン (WordPress.orgで入手可能) 及び、Perfmatters(perfmatters.ioで入手可能)などのプレミアムプラグインを使用することができます。

あるいは、ご利用のWordPressテーマのfunctions.phpファイルに次のコードを追加することにより、自己ピンバックを無効にすることができます。なお、WordPressテーマのソースを誤って編集するとウェブサイトが壊れる恐れがありますので、ご注意ください。なお、無料のCode Snippetsプラグイン (WordPress.orgで入手可能) を使用すると、上記のようなPHPスニペットが追加しやすくなります。これで、テーマを触らずに済みます。

```
function wpsites_disable_self_pingbacks( &$links ) {  
    foreach ( $links as $l => $link )  
        if ( 0 === strpos( $link, get_option( 'home' ) ) )  
            unset($links[$l]);  
}  
  
add_action( 'pre_ping', 'wpsites_disable_self_pingbacks' );
```

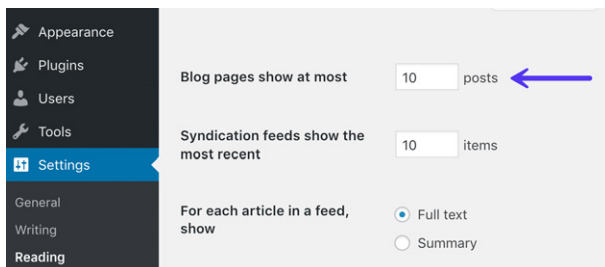
ブログフィードの投稿数を制限する

ホームページであるか、個別のページであるかにかかわらず、ブログフィードでサムネイルが50個も同時に読み込まれるのは好ましくありません。トラフィックの多いブログにとっては、ホームページはウェブサイトの最も重要なページであり、速く読み込まれる必要があります。リクエストとメディアの数が少ないほど、パフォーマンスは向上します。

これこそが、ページネーションが発明された理由です(下記参照)。ページネーションはブログフィードの最後に表示されるもので、クリックすると次のページにアクセスできます。通常は数字ですが、「次へ/前へ」と表示される場合もあります。ご利用中のWordPressテーマにすでにカスタムのページネーションが組み込まれているはずです。



デフォルトではWordPressはWordPressの新規インストールの制限を10件に設定していますが、このデフォルト設定は数え切れないほど何度も変更されてきました。したがって、ご利用の値をご確認ください。当社の推奨は8〜12件の投稿です。Kinstaのブログのホームページでは12件を表示しています。



設定はWordPress管理ダッシュボードの「設定」→「表示設定」にあります。アクセスした後に、「1ページに表示する最大投稿数」の値を変更します。

第8章:

キャッシュが重要な 理由について



キャッシングは間違いなくWordPressをスピードアップする最も重要で実施しやすい方法の一つです!しかし、キャッシングの使い方を説明する前に、まずキャッシングの概要とその種類を理解しておきましょう。

キャッシングとは?

要するに、WordPressウェブサイトの各ウェブページを訪問するには、サーバーへのリクエスト、そのリクエストのサーバーによる処理(データベースクエリなどを含む)と、処理後の結果がサーバーによりユーザーのブラウザへ送信されることが必要です。結果としてはウェブサイトそのものが表示されます。

たとえば、ヘッダー、画像、メニュー、ブログがあるとします。サーバーはこれらすべての要求を処理する必要があるため、ウェブページ全体が読み込まれるまでには時間がかかります。

そこでWordPressのキャッシングプラグインが登場します。キャッシングは、ファイルを構成に応じてディスクまたはRAMに保存するようにサーバーに指示します。したがって、過去に配信したのと同じコンテンツを記憶し複製することができます。基本的には、ページビューを生成するのに必要

な作業量が減ります。その結果、**ウェブページはキャッシュから直接に読み込まれると、はるかに速くなるのです。**

その他のメリットは例えば以下の通りです。

- **サーバーのリソース使用量が減る**：リソース使用量が少ないとウェブサイトが速くなるため、スピードにつながるメリットです。それに、サーバーへの負担も少なくなります。メンバーシップサイトなどの非常に動的なサイトの場合や、キャッシュから配信できるものとできないものを特定する場合には、非常に重要です。
- **TTFBが低くなる**：キャッシングはTTFBを下げる最も簡単な方法の一つです。当社の試験ではキャッシングは通常TTFBを最大90%削減しています。

キャッシングの種類

キャッシングについて、一般的に使用されるアプローチは2種類です。

1. サーバーレベルでのキャッシング
2. プラグインでのキャッシング

1. サーバーレベルでのキャッシング

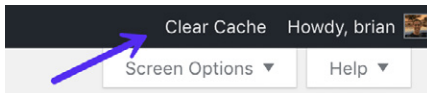
サーバーレベルでのキャッシングは、エンドユーザーにとっては、最も簡単なアプローチの一つです。この場合、キャッシングをWordPressホスティング会社に任せることになります。Kinstaでは、ソフトウェアレベルまたはサーバーレベルで自動的に適用される次の**4種類のキャッシュ**を採用しております。

- バイトコードキャッシュ
- オブジェクトキャッシュ
- ページキャッシュ
- CDNキャッシュ

つまり、複雑で使いにくいキャッシングプラグインを使わずに済むということです。「最高のキャッシングプラグイン」を検索する手間を省き、より重要な仕事に集中することができます。

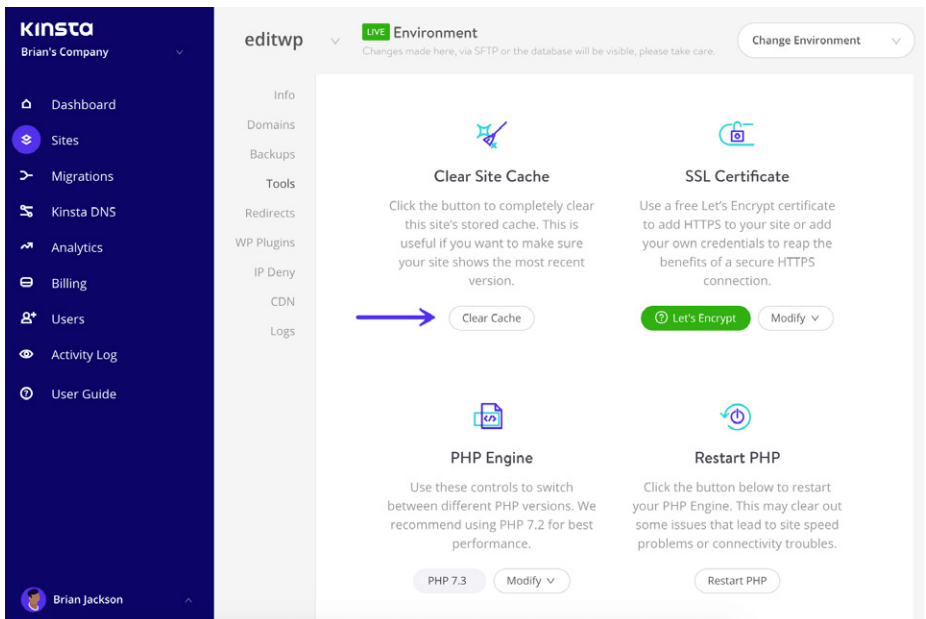
ページキャッシュは、標準のWordPressで何も設定しなくてもすぐに機能するように設定されています。一切触らなくて結構です！WordPressウェブサイトを起動するだけで、ページキャッシュが始まります。

WooCommerce及びEasy Digital Downloadsなどのeコマースサイトを対象にしたキャッシュルールもご用意しております。デフォルトでは、カート、マイアカウント、チェックアウトなど、キャッシュされるべきではない特定のページはキャッシュから除外されます。woocommerce_items_in_cartクッキーまたはedd_items_in_cartクッキーが検出されると、ユーザーはキャッシュを自動的にバイパスすることにより、スムーズなチェックアウトプロセスが保証されます。



管理ツールバーを使用することによりWordPressウェブサイトのキャッシュをいつでもクリアできます。

MyKinstaダッシュボードにも統合されている為、「ツール」をクリックし、「キャッシュをクリア」をクリックするだけです。



キャッシュが重要な理由について

2. プラグインでのキャッシング

ホスティング会社がキャッシュを提供していない場合は、サードパーティ製のWordPressキャッシングプラグインを使用することができます。経験からいうと、次のいずれかをお勧めします。

- **WP Rocket** (プレミアム、wp-rocket.meで入手可能)
- **Cache Enabler** (無料、WordPress.orgで入手可能)
- **W3 Total Cache** (無料、WordPress.orgで入手可能)

KinstaはWP Rocketも完全にサポートしています！当社の内蔵キャッシュと競合するため、通常はキャッシングプラグインの利用を許可していません。ただし、WP Rocket 3.0以降をKinstaのサーバーで実行している場合には、そのページキャッシュ機能は自動的に無効にされます。

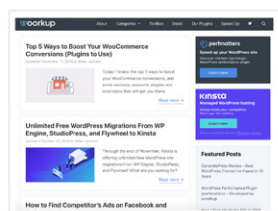
これにより、Kinstaのお客様には当社の高速なサーバーレベルキャッシングをご利用いただける一方、WP Rocketが提供する素晴らしい最適化機能もご活用いただけます。

キャッシングを利用すべきか否か

キャッシュはどの程度の影響を及ぼすのでしょうか。

速度とTTFBの両方への影響を確認する為にKinstaのサーバーレベルのキャッシュで複数の試験を行いました。

キャッシング無し

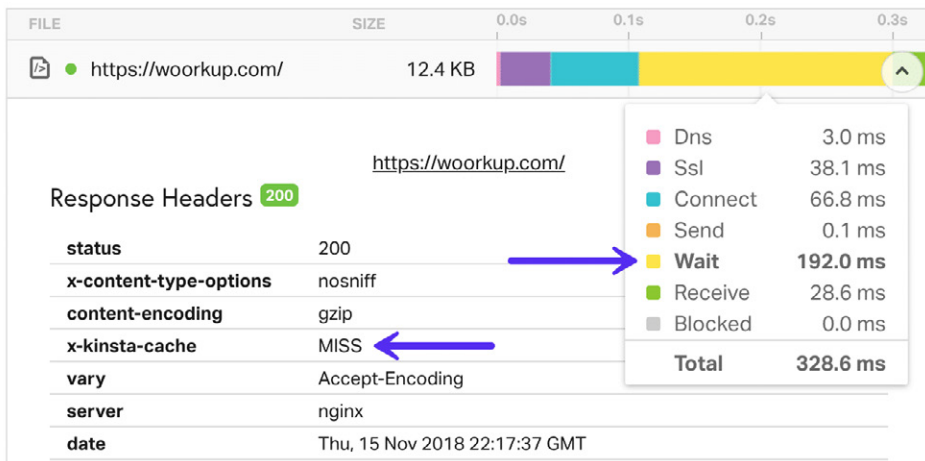


Performance grade B 89	Page size 230.6 KB
Load time 507 ms	Requests 40

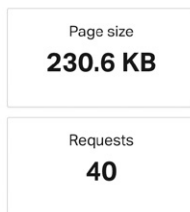
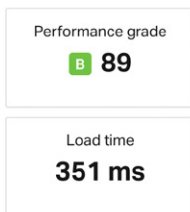
まずキャッシュを無効にし、Pingdomで5つの試験を実行し、その平均をとりました。

キャッシング無しでのTTFB

キャッシュの有無によるTTFBの違いに注意することも重要です。Pingdomの帯グラフの黄色い「待機」という帯はTTFBのことです。ご覧のとおり、キャッシング無しでのTTFBは192ミリ秒です。x-kinsta-cacheヘッダーにMISSが表示されているため、キャッシュから配信されていないことがわかります。



キャッシングあり

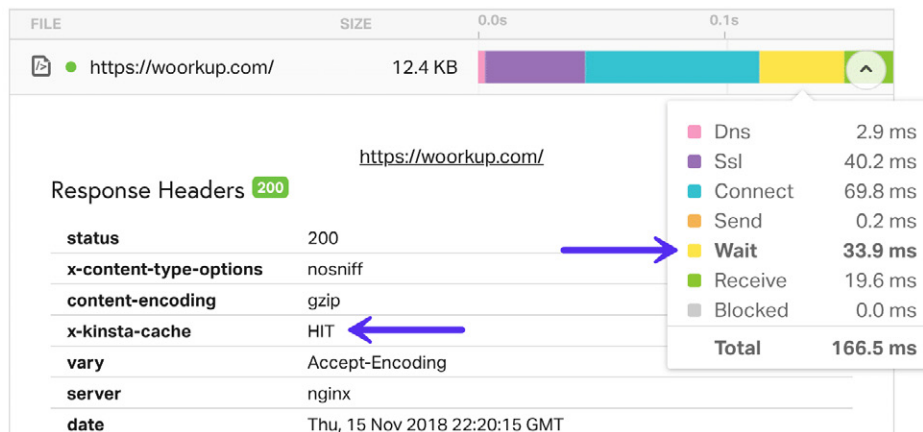


次に、サーバーレベルでのキャッシュを有効にし、Pingdomで5つの試験を実行し、その平均をとりました。

ご覧のとおり、サーバーレベルのキャッシュにより、ページの読み込み時間が33.77%改善されました！特別な作業は全く必要ありませんでした。試験の対象になったウェブサイトはかなり最適化されていたものですが、最適化されていない大規模なサイトであればさらに大きな改善が見られるでしょう。

キャッシングでのTTFB

キャッシングを有効にすると、TTFBは35ミリ秒未満であることがわかります。x-kinsta-cacheヘッダーにHITが表示されているため、キャッシュから配信されていることがわかります。



WordPressホスティング会社が提供するキャッシングと同様にCDNキャッシングも非常に重要です。CDNについては、後ほど詳しくご説明します。

WordPressキャッシュにより、ページの読み込み
時間が33%以上改善される場合があります！

メンバーシップサイトでのキャッシングに関する課題

メンバーシップサイトには、絶えず変化するキャッシュ不可能なコンテンツまたはページが多数あります。メンバーのログインページ(サイトの規模によってはよく読み込まれる可能性があります)、デジタル商品やコースのチェックアウトページ、フォーラムなどは、一般的にキャッシュできないという問題点があります。

しかし、それだけではありません。標準のWordPressウェブサイトでは、**ログインしたユーザー**に対してもWordPressダッシュボードはキャッシュされません。作成者や管理者が数人しかいない場合はこれで問題ありませんが、ダッシュボードを使用しているメンバーが何千人もいる場合には、一人もサーバー上のキャッシュから配信できないため、すぐにパフォーマンスの問題が発生します。つまり、処理には力強いアーキテクチャが必要なのです。共有ホスティング会社は、通常その期待に応えることが出来ません。

非常に動的なサイトのオブジェクトキャッシング

WordPressのメンバーシップサイトの場合、完全に活用できない為一般的なキャッシング設定では不十分です。そこで、**オブジェクトキャッシングが登場します。**

オブジェクトキャッシュはデータベースクエリの結果を保存し、次回その特定のデータが要求されるとデータベースにクエリを実行せずにキャッシュから配信します。これによりPHPの実行時間が短縮され、データベースの負荷が減ります。メンバーシップサイトなら非常に重要なことです！WordPressでオブジェクトキャッシュを実装する方法は次のとおりです。

1. W3 Total Cacheなどのサードパーティ製キャッシングソリューション
2. Redis(推奨)
3. Memcached

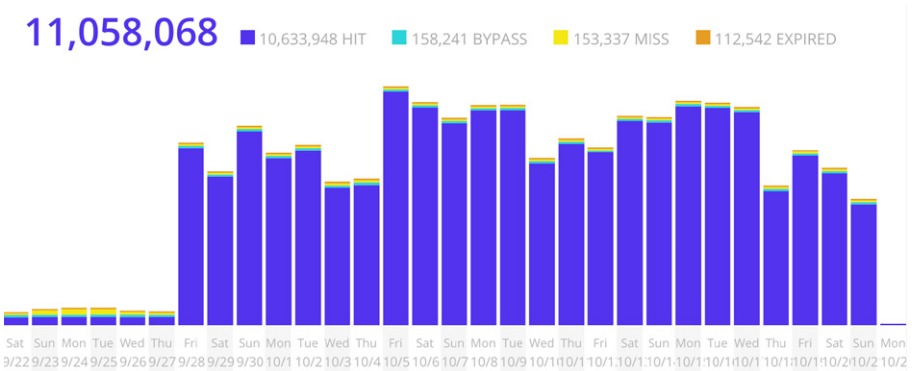
KinstaはアドオンとしてRedisを提供しており、メンバーシップサイトを運営中のお客様に永続的なオブジェクトキャッシングを最大限に活用いただけます。

キャッシングを分析する

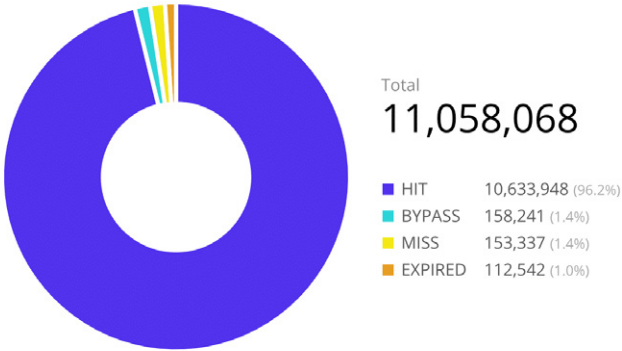
上記のx-kinsta-cacheヘッダーを覚えていますか？ホスティング会社またはキャッシングソリューションによっては、ヘッダーの名前が多少異なる場合があります。WordPressウェブサイトからリクエストが出されるたびに、そのヘッダーにはHIT、BYPASS、MISS、EXPIREDなどの値があります。これにより、キャッシュのパフォーマンスを確認できます。

ウェブサイトのできるだけ多くをキャッシュから配信する必要がある為、WordPressウェブサイトのキャッシュヒット率を上げることが重要です。Kinstaでは、MyKinsta分析ツールのデータとKinstaキャッシュログを分析し、キャッシュ可能なGETリクエストをBYPASSしているキャッシュがあるか、または排除可能なPOSTリクエストがあるかを特定できます。

キャッシュコンポーネントスタック(下記参照)を使用すると、HIT、BYPASS、MISS、またはEXPIREDなどのリクエストの状態を確認できます。過去24時間、7日、または30日でデータをフィルタリングできます。



キャッシュコンポーネントのグラフではキャッシュ率が確認できます。キャッシュから配信する割合が高いほど望ましいです。以下の例で示すように、例のWordPressウェブサイトのキャッシュヒット率は96.2%と非常に良好な結果です！

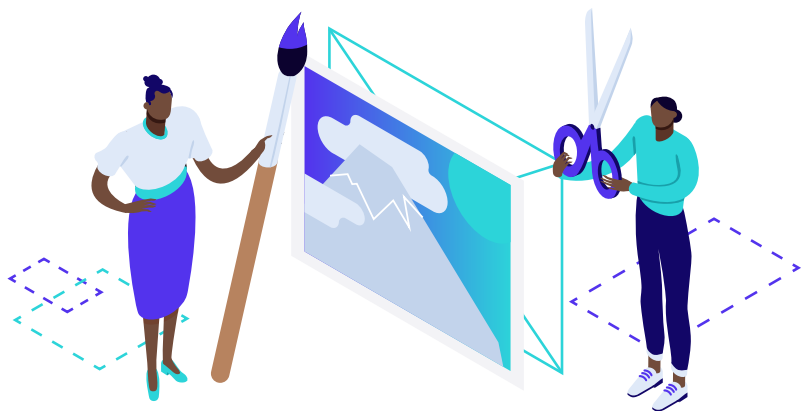


キャッシュバイパスのトップリストでは、キャッシュから配信されていないリクエストが確認できます。通常、CRONジョブ、admin-ajaxリクエスト、eコマースチェックアウトページ、クエリ文字列、UTMパラメータなどがあります。

PATH	REQUESTS
/wp-cron.php?server_triggered_cronjob	28,977
/wp-admin/admin-ajax.php?action=edd_jilt_get_cart_data	22,279
/?ref=1	796
/	667
/sitemap_index.xml	490
/?ajax=1&ht-kb-search=1&s=	435
/account/	372
/checkout/	329
/?ref=19&campaign=bw	292

第9章:

画像の最適化はマスト

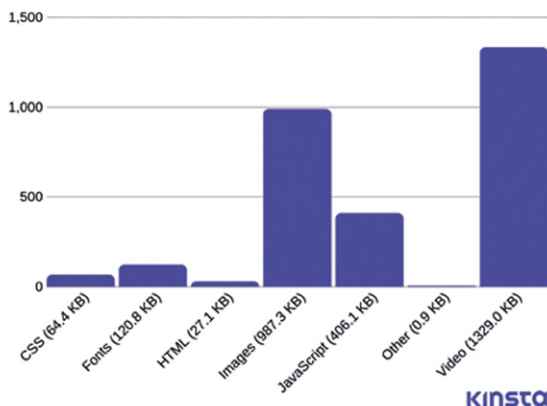


画像の最適化は取り組みやすい手法ですが、合計読み込み時間に大きな影響を与えます。これはプラスアルファの工夫ではなく、マストと言えるでしょう！

大きな画像はウェブページのスピードを落とし、ユーザーエクスペリエンスが悪化します。画像の最適化とは、プラグインまたはスクリプトを使用しファイルサイズを圧縮することによりページの読み込み時間を改善することです。圧縮は非可逆圧縮と可逆圧縮の2種類があります。

HTTP Archiveによると、2019年8月の時点で、画像は**ウェブページの全体的なページの重さの平均34%を占めている、**とのこと。したがって、はるかに最適化しづらいビデオの次に最適化を始めるべきはやはり画像です！JavaScript、CSS、Fontsよりも重要です。皮肉なことに、画像最適化は最も簡単な工夫であるにもかかわらず、ウェブサイト所有者の多くはこれを見逃しています。

1ページあたりの 平均バイト数(KB)



2017年12月に、画像は全体的なページの重さの54%も占めていました。Web全体で、画像の最適化が進んでいるようです！しかし、34%は無視できない数字です。ウェブサイト上にビデオコンテンツのない方にとっては一番の悩みの種は間違いなく画像でしょう。

画像は全体的なページの重さの34%も占めています。最適化しましょう！

バランスを考える(ファイルサイズと画質)

画像を書式設定する主な目的は、**最小ファイルサイズと納得できる画質のバランスを見つけることです**。これらの最適化を実行する方法は複数あります。最も一般的な方法の一つは、WordPressにアップロードする前に画像を圧縮することです。圧縮はAdobe PhotoshopやAffinity Photoなどのツール、またはGoogleの新規オンラインアプリSquooshを使用することによりできます。ただし、プラグインを使用して自動的に実行することもできる為、後程詳しくご説明します。

考慮すべき2つの主な項目は、**ファイルフォーマット**と使用する**圧縮種類**です。ファイルフォーマットと圧縮種類の正しい組み合わせを選択することにより、画像のサイズを最大5分の1にまで減らすことができます。最も効果的なものを探し出すには、各組み合わせを試してみる必要があります。

画像の修正を始める前に、最適なファイルタイプを選択しましょう。使用可能なファイルタイプはいくつかあります。

- **PNG:** 高画質の画像を生成しますが、ファイルサイズも大きくなります。可逆圧縮の画像フォーマットとして作成されますが、非可逆な場合もあります。
- **JPEG:** 非可逆圧縮も可逆圧縮も使用しています。画質とファイルサイズのバランスがとれるように画質レベルを調整できます。

カラフルな画像にはJPEG(またはJPG)を使用し、シンプルな画像にはPNGを使用することが理想的です。

GIFはどうでしょうか？アニメーションGIFは見えて楽しいですが、ウェブパフォーマンスを低下させます。GIFの多くは1 MB以上もする為、ソーシャルメディア及びSlackのみでの使用をお勧めします。どうしてもブログの記事でGIFを使用したい方は、アニメーションGIFの圧縮方法をご確認ください。

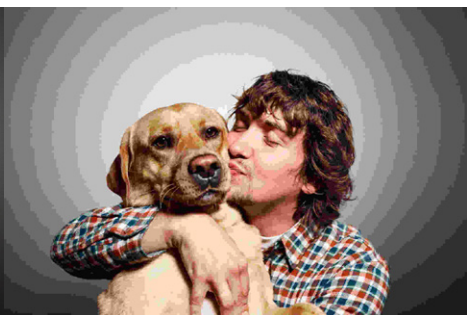
圧縮の画質とサイズ

以下は画像を圧縮しすぎた場合の例です。1つ目の画像は非常に低い圧縮率を使用しているため、最高の解像度が得られますが、ファイルサイズは大きくなります。2つ目は、非常に高い圧縮率を使用しているため、非常に低画質の画像になりますがファイルサイズは小さくなります。注：元の画像は2.06 MBです。

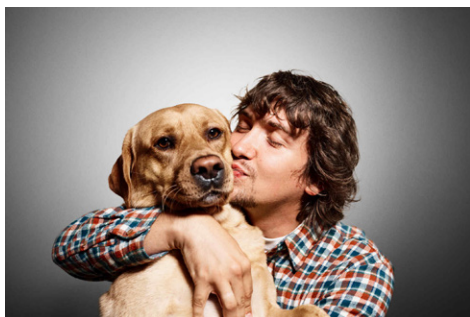
ご覧のとおり、上記の最初の画像は590 KBです。一枚の写真としては大きすぎます！一般的には、ウェブページの全体的なページの重さは1~2 MB未満であることが理想的です。590 KBはその4分の1に当たります。2番目の画像は美しくありませんが、サイズはたったの68 KBです。圧縮率(画質)とファイルサイズの適正なバランスを見つける必要があります。



低圧縮(高画質)のJPG:590 KB590 KB



高圧縮(低画質)のJPG:68 KB68 KB



中圧縮(高画質)のJPG:151 KB151 KB

そこで、画像を中程度の圧縮率で再び圧縮しました。ご覧の通り、画質はきれいで、ファイルサイズは高解像度の写真なら納得できる151 KBです。元の写真の低圧縮版より約4倍軽くなりました。Kinstaでは最高のパフォーマンスのために、画像をできるだけ100 KB未満にしています。

非可逆圧縮と可逆圧縮

圧縮は、非可逆圧縮と可逆圧縮の2種類があります。

非可逆圧縮では、**画像のデータの一部が削除されます**。したがって、画質が低下する場合があります(画質の低下、またはピクセル化)、画像をどこまで圧縮するか注意しなければなりません。画質が悪化するだけでなく、画像を一旦圧縮すると元に戻すことはできないためです。もちろん、非可逆圧縮の大きな利点の一つは、**ファイルサイズを大幅に削減できることです**。



非可逆圧縮とは異なり、**可逆圧縮は画像の画質を悪化させません**。何故そのようなことが可能かというと、圧縮の際に主に不要なメタデータ(画像を撮った装置により自動的に生成されたデータ)が削除されるからです。この方法の最大のデメリットは、**ファイルサイズが大幅に縮小されない**ことです。言い換えれば、時間が経つにつれディスクの使用量が増えます。

両方の方法を実施してみて、自分にとってどちらが最も効果的かを見つける必要があるでしょう。ただし、多くの場合、画質があまり悪化せずサイズを70%も(場合によっては90%も)削減できる**非可逆圧縮をお勧め**します。例えば、ページ1枚に画像が15枚もあると、ウェブサイトの読み込み時間が大幅に改善されます！

画像圧縮プラグイン

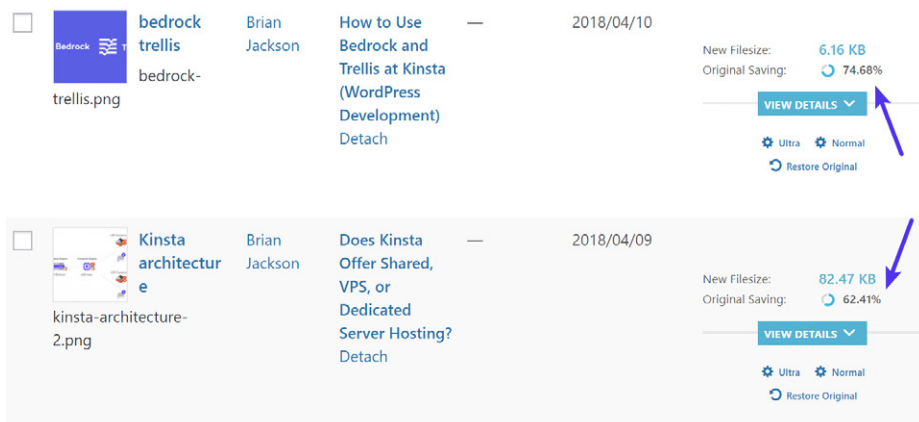
上記のプロセスを完全に自動化できる素晴らしいWordPress画像圧縮プラグインが複数あります！当社のお勧めのプラグインは下記のとおりです。(リストのすべてのプラグインはWordPress.orgで入手可能)

- **Imagify** (非可逆圧縮と可逆圧縮：外部画像最適化)
- **WP Smush** (非可逆圧縮と可逆圧縮：外部画像最適化)
- **Optimole** (非可逆圧縮と可逆圧縮：外部画像最適化)
- **EWWW Cloud** (非可逆圧縮と可逆圧縮：外部画像最適化)
- **ShortPixel** (非可逆圧縮と可逆圧縮：外部画像最適化)

画像最適化プラグインを選択する最も重要なポイントは、**圧縮と最適化を外部(プラグインのサーバー上)で行う**ことです。これにより、ウェブサイトの負荷が軽減されます。上記のものはすべて外部サーバーを使用しています。

KinstaのウェブサイトではImagifyプラグインを使用しています。画像がWordPressのメディアライブラリにアップロードされる際に自動的に圧縮される為、心配する必要はありません。時間の経過とともに、画像圧縮の適切なレベルを理解できるようになります。圧縮種類はノーマル、アグレッシブ、そしてウルトラの3種類があります。

Kinstaではアグレッシブモードを使用しており、画像にもよりますが約60～70%の削減となります。注：当社の画像はほとんど写真ではなくアイコンまたはイラストである為、JPEGよりもPNGを使用しています。



画像圧縮を使用することによりウェブサイトのスピードがどこまで改善するかは、画像の元のサイズと圧縮後の状態により異なります。ただし、当社が行ったスピードテストでは、高品質の画像圧縮ソリューションのおかげで**ページの読み込み時間が80%以上短縮される**ケースがあることがわかりました。

遅延読み込み

画像が多い場合は、遅延読み込みを検討する価値があります。遅延読み込みは、表示されるコンテンツのみを読み込み、残りのコンテンツのダウンロードとレンダリングを遅らせる最適化手法です。

遅延読み込みの概要は次のとおりです。

- ブラウザは、画像をダウンロードしたりビデオをプリロードしたりすることなく、ウェブページのDOM(文書オブジェクトモデル)を構築します。
- JavaScriptを使用して、ページの読み込みの際に最初に表示されるコンテンツをベースにダウンロードするイメージとプリロードするビデオが特定されます。特定された画像とビデオは正常にダウンロード、プリロードされます。
- 追加のビデオのダウンロードとレンダリングは、訪問者がページをスクロールダウンして追加のコンテンツが表示されるまで遅延されます。

結論としては、画像のダウンロードとビデオの読み込みは実際に必要になったときに行われます。そうすると、高解像度の画像や埋め込みビデオの多いウェブサイトならパフォーマンスが大幅に向上する可能性があります。

ありがたいことに、遅延読み込みを簡単に有効にできる素晴らしい無料のWordPressプラグインはたくさん存在します。(WordPress.orgで入手可能)

- **a3 Lazy Load**
- **Lazy Load by WP Rocket**
- **Lazy Load for Videos**

画像最適化のその他の工夫

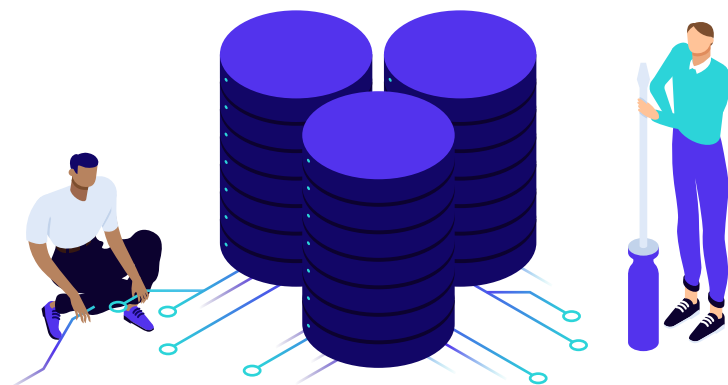
以下に画像最適化のその他の便利な工夫をご紹介します。

- 列またはDIVの幅だけに合わせたサイズの画像をアップロードする時代は終わりました。レスポンシブな画像はWordPress(バージョン4.4以降)では基本設定のままで利用可能で、モバイルユーザーには自動的にサイズを減らした画像が表示されます。
- SVGも画像を保存する便利な方法です。Kinstaのウェブサイトの手描きのイラストはすべてSVG(ベクトル)です。SVGは通常ファイルサイズが小さくなりますが、必ずしもそうとは限りません。

- 画像内にテキストを配置する代わりにアイコンフォントを使用しましょう。アイコンフォントはスケーリングしても見た目がきれいで、スペースの使用量も少なくなります。フォントジェネレータを使用することにより、さらに最適化することができます。フォントジェネレータを使用し、アイコンフォントファイルのサイズを**97.59%減らした実績**をご参照ください。
- また、登場直後から注目を集めてきたGoogleの新しい画像フォーマットであるWebPも検討に値します。

第10章:

データベースの微調整



次はWordPressデータベースを微調整する工夫についてです。自動車が時間が経つにつれ劣化すると同様にデータベースもメンテナンスする必要があります。

特に、**クエリが多く生成される**ためMySQLデータベースから情報を取得する際の待ち時間が長くなるメンバーシップサイトは、注意が必要です。これらのサイトには動的なコンテンツとデータが多いこと、またはナビゲーション用に数多くの検索クエリまたはWP_Queryが使用されることが要因として挙げられます。

言うまでもなく、データベースに対して継続的、同時期にクエリを実行するユーザーも多数います。

InnoDB MySQLストレージエンジンを使用する

古いウェブサイトの多くではデータベースにまだにMyISAMストレージエンジンが使用されています。近年、InnoDBの方がパフォーマンスも良く、信頼性も高くなっています。

たとえてみると、InnoDBは合成潤滑油で、MyISAMは通常の潤滑油です。

MyISAMと比較したInnoDBのメリットは以下の通りです。

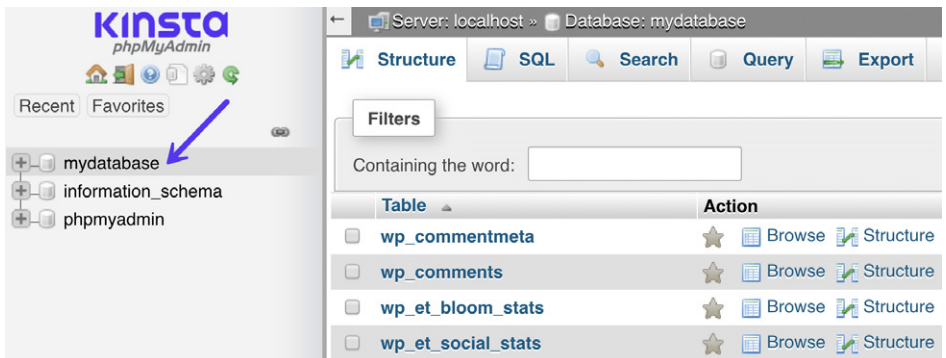
- InnoDBには**行レベルのロック**があり、MyISAMにはテーブルレベルのロックしかありません。これにより、クエリの処理速度が向上します。
- InnoDBには**外部キー** (RDBMS) とリレーションシップの制約のサポートを含む、参照の整合性というものがありますが、MyISAMにはありません (DMBS)。
- InnoDBは**トランザクション**をサポートしています。つまり、コミットしてロールバックすることができます。MyISAMはトランザクションをサポートしていません。
- InnoDBは自動回復のためにトランザクションログを使用している為、高い信頼性を誇ります。MyISAMはトランザクションをサポートしていません。

ご利用のストレージエンジンがInnoDBであるかMyISAMであるか分からないという方については、新しいWordPressウェブサイトをお持ちの場合、InnoDB MySQLストレージエンジンの可能性が高いでしょう。、古いWordPressサイトをお持ちの場合には、確認した方が良いかもしれません。MyISAMとInnoDBのテーブルを混在したウェブサイトもあります。

その場合は統一するだけで改善が見られます。確認するには以下のステップに従ってください。

ステップ1

phpMyAdminにログインしてMySQLデータベースをクリックします。



ステップ2

検索するか、「種類」の列で並べ替えると、テーブルが使用しているストレージエンジンの種類を確認できます。以下の例では、2つのテーブルがまだMyISAMを使用していることがわかります。

						0	InnoDB	utf8_unicode_ci	64 KiB
						0	InnoDB	utf8_unicode_ci	48 KiB
						0	InnoDB	utf8_unicode_ci	64 KiB
						0	InnoDB	utf8_unicode_ci	80 KiB
						0	InnoDB	utf8_unicode_ci	48 KiB
						0	InnoDB	utf8mb4_unicode_ci	32 KiB
						113	MyISAM	utf8mb4_unicode_ci	80 KiB
						471	MyISAM	utf8mb4_unicode_ci	208 KiB
						0	InnoDB	utf8mb4_unicode_ci	48 KiB
						1	InnoDB	utf8mb4_unicode_ci	16 KiB

Operations Triggers

Table options

Rename table to: wp_comments

☐ Adjust privileges

Table comments:

Storage Engine: MyISAM

Collation:

PACK_KEYS:

CHECKSUM: ☐

DELAY_KEY_WRITE: ☐

AUTO_INCREMENT: 2890

ROW_FORMAT: DYNAMIC

Go

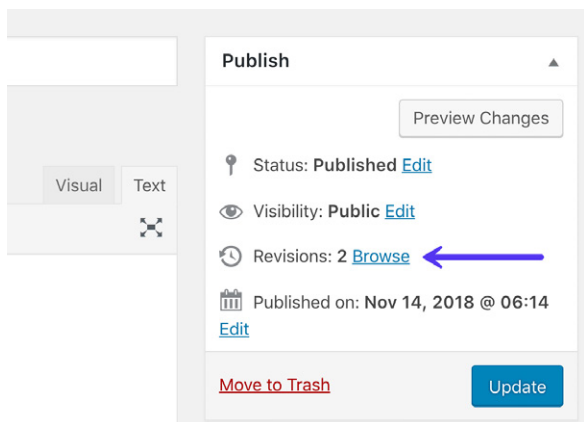
それらをInnoDBに移行します。この作業はできればホスティング会社に依頼した方が良いでしょう。Kinstaでは、すべてのお客様のデータベーステーブルがマイグレーションチームにより自動的にInnoDBに変換されます。

ただし、phpMyAdminを使用して手動で変換することもできます。myISAMテーブルをクリックし、「操作」タブをクリックして、ストレージエンジンを切り替えるだけで完了です。

開発に明るい方であれば、WP-CLIを使用して変換することもできます。

固定ページおよび投稿のリビジョンを削除または制限する

WordPressで固定ページおよび投稿を保存するたびに、リビジョンというものが作成されます。下書きのものでも、公開済みのものでも同じです。リビジョンは過去のコンテンツ内容に戻したいという時に便利です。

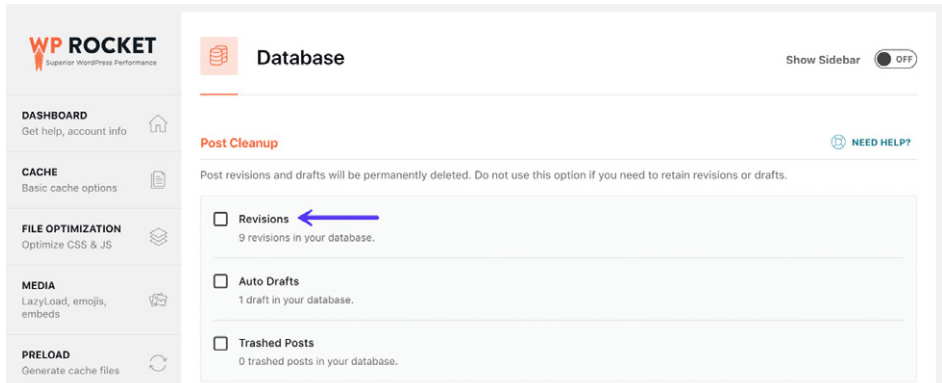


ただし、リビジョンはWordPressサイトのパフォーマンスを低下させる場合もあります。大規模なサイトでは、リビジョンはデータベース内で何千行にもなり、中には不要なものもあるでしょう。行数が多いほど、データベースのサイズが大きくなり、ストレージスペースが消費されます。インデックスが発明されたのはこのためですが、それでもこの課題に直面しているWordPressウェブサイトもあります。対処方法はいくつかあります。

1. 古いリビジョンを削除する

固定ページと投稿の多い、古いWordPressウェブサイトをお持ちの場合は、そろそろクリーンアップして古いリビジョンを削除した方が良いでしょう。MySQLでも実施できますが、ウェブ上には悪いコードスニペットが多い為、まずウェブサイトのバックアップを取って、WP-Sweepなどの無料のプラグインを使用することを勧めます。

当社スタッフお気に入りのWP Rocketプラグインにもリビジョンを削除するデータベース最適化機能があります。



WP-CLIに詳しい方は、以下の便利なコマンドも使用できます。

データベースにあるリビジョンの数を把握するには、SSH経由でご利用のサーバーにログインし、次のコマンドを実行します。

wp revisions list

[illegible]

エラーが発生した場合は、まず以下のコマンドを使用し、wp-revisions-cliパッケージをインストールする必要があります。

```
wp package install trepmal/wp-revisions-cli
```

その後、次のコマンドを実行してリビジョンを削除できます。

```
wp revisions clean
```

2. リビジョンの数を制限する

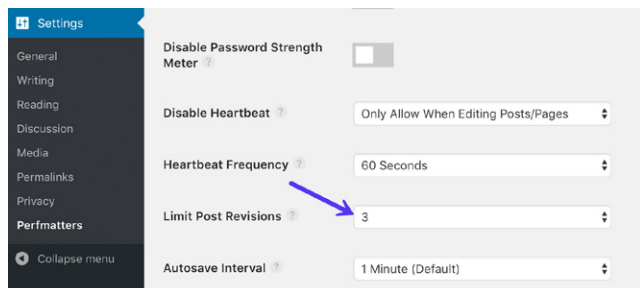
Kinstaでも実施しているもう一つのアプローチは、投稿または固定ページ毎に保存できるリビジョンの数を制限することです。特に、記事を頻繁に更新している場合には、その値を10などに設定することによりリビジョンが手に負えないような数になることはありません。

リビジョンの数を制限するには、wp-config.phpファイルに次のコードを追加します。以下のコードは 'ABSPATH' の上に挿入する必要があります。そうでないと機能しません。データベースに保存しておきたいリビジョン数に合わせて数値を変更しても構いません。

```
define( 'WP_POST_REVISIONS', 10 );
```

```
59
60 /**
61  * WordPress Database Table prefix.
62  *
63  * You can have multiple installations in one database if you give each
64  * a unique prefix. Only numbers, letters, and underscores please!
65  */
66 $table_prefix = 'wp_';
67
68 define( 'WP_POST_REVISIONS', 10 );
69
70 /* That's all, stop editing! Happy blogging. */
71
72 /** Absolute path to the WordPress directory. */
73 if ( ! defined( 'ABSPATH' ) )
74     define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76 /** Sets up WordPress vars and included files. */
77 require_once ABSPATH . 'wp-settings.php';
78
```

Line 28, Column 1



Perfmattersでリビジョン数を制限する

または、Perfmatters (perfmatters.ioで入手可能)などのプラグインを利用し、リビジョンの数を制限することもできます。

3. リビジョン機能を無効にする

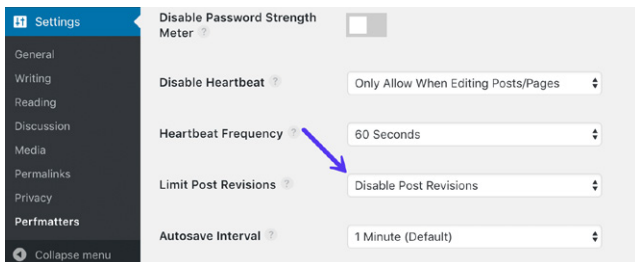
最後になりますが、ウェブサイトのリビジョン機能を完全に無効にすることもできます。この対策を実施する方は、まず前述の手順に従いリビジョンを削除した上でリビジョン機能を無効にしてください。そうするとデータベースには古いリビジョンが残らず、新しいものは追加されません。

リビジョン機能を無効にするには、wp-config.phpファイルに次のコードを追加します。以下のコードは 'ABSPATH' の上に挿入する必要があります。そうでないと機能しません。

```
define('WP_POST_REVISIONS', false);
```

```
60  /**
61   * WordPress Database Table prefix.
62   *
63   * You can have multiple installations in one database if you give each
64   * a unique prefix. Only numbers, letters, and underscores please!
65   */
66  $table_prefix = 'wp_';
67
68  define('WP_POST_REVISIONS', false); ←
69
70  /* That's all, stop editing! Happy blogging. */
71
72  /** Absolute path to the WordPress directory. */
73  if ( ! defined( 'ABSPATH' ) )
74      define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76  /** Sets up WordPress vars and included files. */
77  require_once ABSPATH . 'wp-settings.php';
78
```

Line 33, Column 34



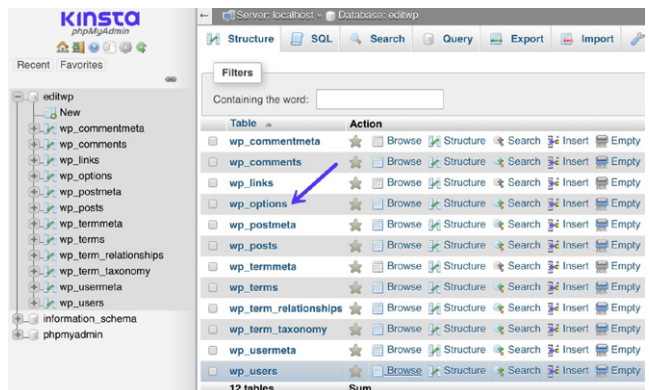
Perfmattersでリビジョン機能を無効にする

または、Perfmattersなどのプラグインを利用し、リビジョン機能を無効にすることもできます。

wp_optionsテーブルと自動読み込みデータを整理する

WordPressとデータベースの全体的パフォーマンスにおいて、wp_optionsテーブルは見落とされがちです。特に大規模のウェブサイトでは、サードパーティ製プラグイン及びテーマにより残された自動読み込みデータのせいでクエリ時間が遅くなることがあります。当社チームも毎日のようにこの問題を経験しています！

wp_optionsテーブルには、WordPressウェブサイトのあらゆるデータがあります。例えば、以下の通りです。

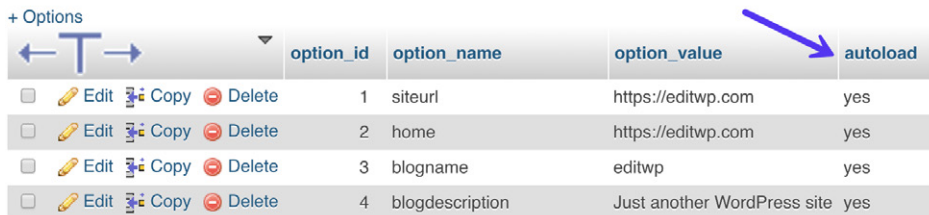


- サイトURL、ホームURL、管理者メールアドレス、デフォルトのカテゴリ、ページごとの投稿数、時間のフォーマットなど
- プラグイン、テーマ、ウィジェットの設定
- 一時的にキャッシュされたデータ

このテーブルには、以下の項目(列)があります。

- option_id
- option_name
- option_value
- autoload (パフォーマンスの観点ではここがポイントです。)

+ Options



	option_id	option_name	option_value	autoload
☐ Edit Copy Delete	1	siteurl	https://editwp.com	yes
☐ Edit Copy Delete	2	home	https://editwp.com	yes
☐ Edit Copy Delete	3	blogname	editwp	yes
☐ Edit Copy Delete	4	blogdescription	Just another WordPress site	yes

wp_optionsテーブルについて理解しておくべき重要なポイントの一つは、**autoload**(自動読み込み)列です。これには、yesまたはnoの値(フラグ)があり、wp_load_alloptions()関数にそのデータが読み込まれるかどうかを制御します。自動読み込みデータとは、**WordPressウェブサイトの各ページに読み込まれているデータ**のことです。考え方は、特定のスクリプトがウェブサイト全体に読み込まれないようにする手順と同様です。autoload属性は、開発者向けに「yes」の基本設定になっていますが、各プラグインが各ページにそのデータ全体を読み込む必要はありません。

wp_optionsテーブルに大量の自動読み込みデータがあるときに、WordPressサイトに異常が発生する場合があります。一般的な理由は以下の通りです。

- あるプラグインが「no」に設定すべきなのに、データが自動読み込みされている。お問い合わせプラグインがその分かりやすい例でしょう。本当に各ページでデータを読み込むべきでしょうか。それともお問い合わせのページのみで十分でしょうか。
- あるプラグインまたはテーマが既にWordPressサイトから削除されているが、関連のオプションはまだwp_optionsテーブルに残っており、リクエスト毎に不要な自動読み込みデータに対するクエリが実行されている。
- プラグインまたはテーマの開発者が、独自のテーブルを利用するのではなく、wp_optionsテーブルデータが読み込まれている。開発者の中には追加のテーブルを作成しないプラグインがいいとする人もいる為、議論を双方の立場から考えることもできます。一方、wp_optionsテーブルは数千行を処理するために設計されたものではありません。

自動読み込みデータはどの程度が多すぎるのでしょうか。もちろん場合にもよりますが、300 KB～1 MB程度が理想的です。3～5 MBの範囲に近づく場合、最適化したり自動読み込みのデータから除外したりできるものがあるはずで。そして10 MB以上では、すぐに対処する必要があります。必ずしも問題が発生するとは限りませんが、手始めに実施してみましょう。

自動読み込みデータの削除は、Kinstaチームがお客様によくご提供しているサポートです。

wp_optionsテーブルを最近クリアしましたか？おそらく、実施していないのではないのでしょうか。是非お試しください。

一時データを削除する

オブジェクトキャッシュを採用していない限り、WordPressは一時レコードwp_optionsテーブルに保存します。これには有効期限が設定され、時間が経つにつれ消えるはずですが、必ずしもそうとは限りません。何千もの古い一時レコードがあるデータベースをいくつか見たことがあります。あるウェブサイトでは、wp_optionsテーブルに**695,000行以上を生成した**一時的なレコードがありました。考えるだけで恐ろしい！

The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible. The main panel shows the 'wp_options' table with a message: 'Showing rows 0 - 25 (695046 total, query took 0.0145 seconds.)'. The table has columns: option_id, option_name, option_value, and autoload. The first few rows are visible, showing image processing queue batches.

option_id	option_name	option_value	autoload
524735	wp_image_processing_queue_batch_ca41bb3d7a31eb31c7...	a:9:{0:a:4:{s:7:"post_id";s:5:"15675";s:5:"width...}	no
524736	wp_image_processing_queue_batch_f0c9c1a34e29df33f9...	a:10:{0:a:4:{s:7:"post_id";s:5:"15675";s:5:"width...}	no
524737	wp_image_processing_queue_batch_a204c4fd748a844d...	a:1:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...}	no
524738	wp_image_processing_queue_batch_76b5a2bb0f51d738f0...	a:2:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...}	no
524739	wp_image_processing_queue_batch_7352d2a1b5ce4b9769...	a:3:{0:a:4:{s:7:"post_id";s:5:"22004";s:5:"width...}	no

一時データはデフォルトで自動読み込みされない為、ご注意ください。以下のようなクエリを実行し、自動読み込みの一時データがあるかどうかを確認します。

```
SELECT *  
FROM `wp_options`  
WHERE `autoload` = 'yes'  
AND `option_name` LIKE '%transient%'
```

上記より安全な方法は、wp_optionsテーブルから期限切れの一時データのみを削除する Transient CleanerやDelete Expired Transientsなどの無料のプラグインを利用することです。一方、WordPress4.9で追加された新機能もあり、期限切れの一時データを処理してくれます。あなたのウェブサイトでも自動的に行われているはずです。

WP Rocketにも、データベース最適化オプションの中に一時データを削除する機能があります。

The screenshot displays the WP Rocket settings page. On the left is a sidebar menu with categories: ADVANCED RULES, DATABASE, CDN, HEARTBEAT, ADD-ONS, IMAGE OPTIMIZATION, and TOOLS. The 'Transients Cleanup' section is highlighted with a blue arrow. The main content area is divided into two parts: 'Comments Cleanup' and 'Transients Cleanup'. The 'Comments Cleanup' section shows options for 'Spam Comments' (0 in database) and 'Trashed Comments' (0 in database). The 'Transients Cleanup' section shows options for 'Expired transients' (2 in database) and 'All transients' (24 in database). A note states: 'Transients are temporary options; they are safe to remove. They will be automatically regenerated as your plugins require them.'

WordPressセッションを削除する

cronジョブがシンクロしなくなり、きちんと機能しない為、セッションが適切に削除されない、なんてこともよく発生する問題の一つです。データベースに大量のwp_session_行があることもありま

す。以下の例のウェブサイトではwp_optionsテーブルに300万行以上が表示されており、テーブルのサイズは600 MB以上でした。

wp_links	0	InnoDB	utf8mb4_unicode_ci	32	KiB
wp_options	~3,419,339	InnoDB	utf8mb4_unicode_ci	620.9	MiB
wp_postmeta	12,986	InnoDB	utf8mb4_unicode_ci	9.1	MiB
wp_posts	2,367	InnoDB	utf8mb4_unicode_ci	4	MiB

以下のようなクエリを実行し、この問題が発生しているかどうかを確認します。

```
SELECT *
FROM `wp_options`
WHERE `option_name` LIKE '_wp_session_%'
```

<< < 6491 > >> | Number of rows: 500 | Filter rows: Search this table | Sort by key: None

option_id	option_name	option_value	autoload
3301835	_wp_session_254a6d16b4400d5aabf3c48b41b72d2b	a:1:{s:10:"edd_errors";s:0:""};	no
3301836	_wp_session_expires_254a6d16b4400d5aabf3c48b41b72d...	1520801306	no
3301837	_wp_session_747c86aefae52149d1068a60f240249f	a:1:{s:10:"edd_errors";s:0:""};	no
3301838	_wp_session_expires_747c86aefae52149d1068a60f24024...	1520801306	no
3301839	_wp_session_17aa53659273873116d83993bae7cfd8	a:1:{s:10:"edd_errors";s:0:""};	no
3301840	_wp_session_expires_17aa53659273873116d83993bae7cf...	1520801306	no
3301841	_wp_session_f3f29f822eec97951504e4d66afcd081	a:1:{s:10:"edd_errors";s:0:""};	no
3301842	_wp_session_expires_f3f29f822eec97951504e4d66afcd0...	1520801306	no
3301843	_wp_session_2d4be84abb1e7956e229506e1c26e60e	a:1:{s:10:"edd_errors";s:0:""};	no

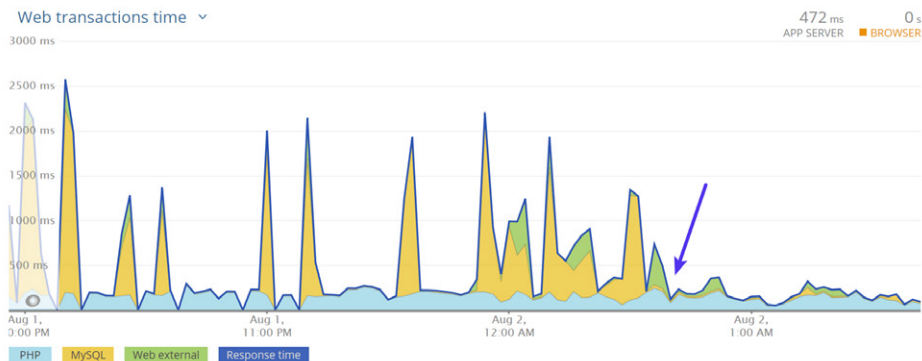
これらは、ほとんどの場合、次のコマンドを実行し安全に削除できます。(本来はcronジョブがこれを削除するべきところでした)

```
DELETE FROM `wp_options`
WHERE `option_name` LIKE '_wp_session_%'
```

不要なwp_session_行を削除した後、テーブルは1,000行未満となり、サイズは11 MBになりました。

wp_links	0	InnoDB	utf8mb4_unicode_ci	32	KiB
wp_options	672	InnoDB	utf8mb4_unicode_ci	11.5	MiB
wp_postmeta	12,986	InnoDB	utf8mb4_unicode_ci	9.1	MiB
wp_posts	2,367	InnoDB	utf8mb4_unicode_ci	4	MiB

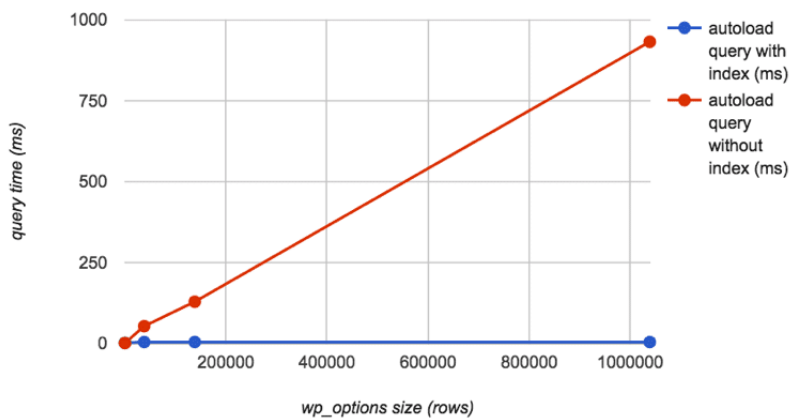
該当ウェブサイトのMySQLのばらつきも改善されました。



自動読み込みにインデックスを追加する

wp_optionsテーブルをクリーンアップするだけでは不十分な場合には、自動読み込み欄に「インデックス」を追加しましょう。インデックスを追加することにより自動読み込み欄がより効率的に検索されるようになります。10upのチームは、典型的な数の自動読み込みレコードを使用し、wp_optionsクエリに自動読み込みインデックスを追加するとパフォーマンスが改善することを示すために、wp_optionsテーブルに関する複数の試験を行いました。

wp_optionsのサイズとクエリ時間の比較



画像の出典: 10up

WordPressの永続オブジェクトキャッシュにRedisを使用する

Redisはオープンソースのインメモリデータストアです。WordPressなら、Redisを使用しWordPressの独自のオブジェクトキャッシュにより生成された値を永続的に保存し、キャッシュされたオブジェクトをページ読み込み間に再利用できます。

Redisなどの永続オブジェクトキャッシュを使用すると、MySQLデータベースに同じオブジェクトに対して2回クエリを実行せずに、**キャッシュされたオブジェクトを再利用できます**。その結果、RedisはウェブサイトのMySQLデータベースの負荷を軽減し、ウェブサイトの応答時間を短縮し、トラフィックの変更に合わせてスケーリングできるようになります。



ページキャッシングを使用できない非常に動的なウェブサイト(WooCommerce、メンバーシップサイト、フォーラム、非常に多くのコメントのあるブログなど)なら、Redisなどの永続的なオブジェクトキャッシングを検討すると良いでしょう。

Kinstaのお客様は、Redisアドオンをご利用いただけます。

Elasticsearchを使用し WordPress検索をスピードアップする

Elasticsearchはオープンソースの全文検索エンジンです。データの索引付けと検索を驚くほど速いスピードで行うために設計されたものです。

WordPressなら、Elasticsearchを使用し**WordPressデータベースへのクエリをスピードアップ**することができます。ウェブサイトデータベースのコンテンツインデックスを構築し、MySQLクエリではできないほど速くElasticsearchでそのインデックスを検索します。



elasticsearch

時間とスキルのある知識の豊富なWordPress、Elasticsearchの開発者であれば、ElasticsearchをWordPressウェブサイトと統合することができます。ウェブサイトのWP_Queryが比較的標準的である場合には、WordPress.orgから入手可能な10up提供の無料のWordPressプラグインElasticPressをインストールすることによりElasticsearchを統合することも可能です。このプラグインはWP_Queryのオブジェクトと自動的に統合して、MySQLではなくElasticsearchを使用してクエリ結果を生成します。

WP_Queryを多用するウェブサイトではElasticsearchが役に立ちます。例えば、以下のようなウェブサイトです。

- ナビゲーションの主要な手段が検索であるウェブサイト
- サイト管理者が頻繁に注文のリストを検索する、注文の多いWooCommerceウェブサイト
- 記事が多いためMySQLクエリの結果が非常に遅いウェブサイト

Kinstaのお客様は、Redisと同様にElasticsearchアドオンもご利用いただけます。

データベースを多用する不要な機能を無効にする

当たり前のことかもしれませんが、プラグインやデータの重要度は低いわりにデータベースを多用する機能を無効にすると、違いが出ます。

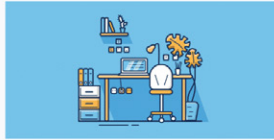
- 人気のある投稿や関連記事のウィジェットやプラグインは最悪です。ウェブサイト全体の重いクエリを使用するのが一般的です。
- あなたのサーバーを使用し画像を圧縮するタイプの画像最適化プラグインも要注意。外部で画像を最適化してくれるプラグインにしましょう。

Kinstaのブログにアクセスし各記事の最後までスクロールダウンすると、「厳選された関連記事」があります。これは当社スタッフが手動で選択し、記事に貼り付けています。これにより、クエリがほとんどなくなり、ウェブサイト全体のパフォーマンスが低下することはありません。確かに手間がかかりますが、表示される記事が管理できる為自動的に表示されるより効率が良いとも言えます。

Hand-picked related articles



Working Remotely –
Everything You Need to Know
Beforehand



The 53 Best Tools for
Freelancers to Scale a (Real)
Business



Where and How to Hire a
WordPress Developer

上記を実現するために、私たちはAdvanced Custom Fields (WordPress.orgで入手可能) プラグインを使用し、手動で各フィールドをブログの投稿タイプに割り当てました。これにより、関連するコンテンツを検索しブログの投稿に指定することができます(下記参照)。

Post Settings

Related Posts

Choose hand-picked related posts here. If none are chosen (or fewer than needed for the display) we will fill the rest up with posts from within the same category/tags.

Search...

10 Best WordPress Directory Plugins to Organize Your Listings (draft)
10 Best WordPress Security Plugins to Lock out the Bad Guys
10 High-converting WordPress Landing Page Plugins With Beauty and Optimization Combined (draft)
10 Stunning Places to Find Free Images For WordPress
10 Things Not To Do In PHP 7

How to Score 100/100 in Google PageSpeed Insights with WordPress
10 Unique WordPress Ecommerce Strategies to Skyrocket Sales
Conversion Rate Optimization Tips – 12 Easy Ways to Boost Sales

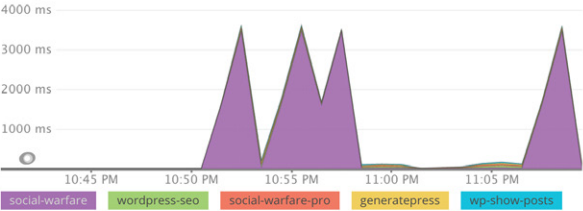
絶対に必要だという場合を除き、ビュー数または投稿数のカウンターをウェブサイトに追加するプラグインは使用しない方がいいでしょう。たとえば、フォーラム投稿内のユーザーのアバターの横にある「792件の投稿」または、フォーラム投稿の一覧の「5,243ビュー」などは避けましょう。議論が長くなるに連れて、このようなカウンターはデータベースに大きな負荷をかけます。原則として、カウンターの使用はできる限り避けましょう。

ソーシャルメディアのカウンターも同様です。たとえば、下記の例のウェブサイトでは、人気のある Social Warfareプラグインの応答時間が、その下の次のプラグインの30倍もします。キャッシングは有効になっていますが、このプラグインは明らかにパフォーマンスに悪影響です。プラグインを無効にした後に読み込み時間とWordPress管理ダッシュボードの応答性が大幅に改善されました。

Sort by Most time consuming

social-warfare	18.9 sec
wordpress-seo	624 ms
social-warfare-pro	449 ms
generatepress	425 ms
wp-show-posts	389 ms
aawp	356 ms
gp-premium	229 ms
nogluten	160 ms
imagify	115 ms
wp-review	107 ms

Top 5 plugins and themes
by response time



第11章:

コンテンツ配信ネットワーク (CDN)を使用する



CDNは、Content Delivery Network (和訳: コンテンツ配信ネットワーク) の略です。コンテンツ配信ネットワークは、画像、CSS、JavaScript、ビデオストリームなど、WordPressウェブサイトの静的な(時には動的な)コンテンツをホストするための世界中のサーバー (POPともいう) ネットワークです。

まず、全く違う目的のもので、CDNをWordPressホスティング会社と混同しないようにしましょう。CDNはホスティング会社の代替サービスではなく、ウェブサイトのスピードを向上させる追加のサービスです。もともと高速なKinstaであっても、CDNのご利用によりスピードがさらに改善できます。

CDNの概要

CDNの考え方をご説明します。たとえば、Kinstaでウェブサイトをホスティングする場合は、米国中部、ヨーロッパ、南米、アジアなどのデータセンターの場所を選択します。

たとえば、米国中央を選択するとします。その場合、ウェブサイトが物理的にアイオワ州カウンシルブラフスの「ホストサーバー」に配置されていることになります。するとヨーロッパからのお客様がウェブサイトを訪れると、読み込み時間は例えばテキサス州のダラスのお客様の読み込み時間より長くなります。

何故でしょうか。それは、データの移動距離が長くなるからです。これをレイテンシ(待ち時間)と言います。レイテンシとは、ネットワーク上でのデータの送信にかかる時間およびその遅延のことです。距離が遠いほど、レイテンシは長くなります。

CDNの種類

コンテンツ配信ネットワークには2種類あります。

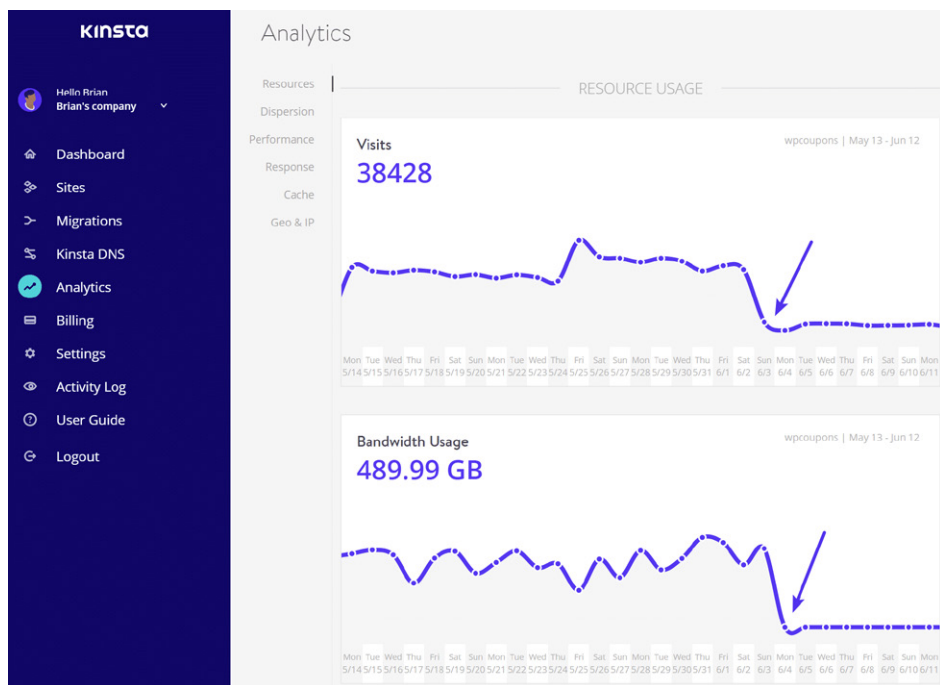
1. 通常のプル型CDN
2. リバースプロキシCDN

通常のプル型CDNは、コンテンツとメディアを完全にコピーしキャッシュしますが、クライアントからのリクエストが直接、ホスティング会社に向けて行われます。例えばKeyCDNとCDN77は通常のCDNです。

リバースプロキシCDNは少し異なります。CDNとして機能しますが、入ってくるリクエストを傍受し、クライアントとホスティング会社の間で仲介サーバーの機能を果たします。例えばCloudflareとSucuriはリバースプロキシCDNです。このような理由で、DNSをホスティング会社ではなく直接これらのプロバイダを指すように設定することになります。

リバースプロキシCDNの利点は、仲介サーバーとして機能するため、強力なウェブアプリケーションファイアウォールを提供し、悪質なトラフィックがWordPressウェブサイトとホスティング会社にアクセスすることを予防できることです。欠点は、パフォーマンスの観点で通常のプル型CDNと比較するとわずかにオーバーヘッドがあることですが、パフォーマンス機能とセキュリティ機能が多いため、この欠点は問題にならないという意見もあります。

以下の例は、当社のお客様のウェブサイトでSucuriを有効にした前後の状態です。悪意のあるトラフィックが急激に減ったことが分かります。結論としては、リバースプロキシサービスはホスティング費用を節約するのに有効だと言えます。



CDNのスピードテスト

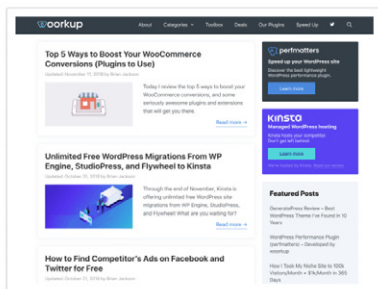
WordPressキャッシングの利点については前述しましたが、CDNキャッシングも非常に強力です。それはCDNにはホスティング会社よりもはるかに多くのサーバーロケーションがあるからです。つまり、アセット(画像、JS、CSS)を訪問者のいる場所の近くにキャッシュし、超高速で配信することができるのです。

それでは、テストによりCDNによるウェブサイトのスピードへの影響度を確認しましょう。

CDN無し

試験したウェブサイトはKinstaでホストされており、サーバーは物理的には米国のアイオワ州のデータセンターにあります。最初にPingdomのスピードテストを5件実施し、平均をとりました。

(CDNは有効にしていな)注:CDNの実力をお見せできるよう、Pingdomのヨーロッパ・イギリス・ロンドンのロケーションを使用しました。合計読み込み時間は**1.03秒**でした。



Performance grade

B 89

Page size

231.0 KB

Load time

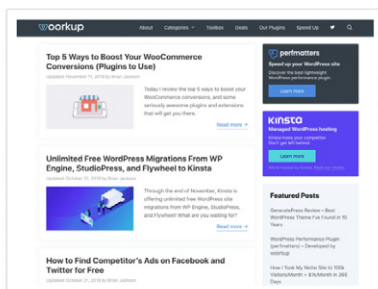
1.03 s

Requests

40

CDNあり

次に、CDNを有効にしPingdomで再びテストを5件実施しました。ヨーロッパ・イギリス・ロンドンのロケーションからの合計読み込み時間は**585ミリ秒**でした。つまり、CDNを使用することで、**ページの読み込み時間を43.2%短縮することができたのです！**すごいですね。



Performance grade

B 89

Page size

230.6 KB

Load time

585 ms

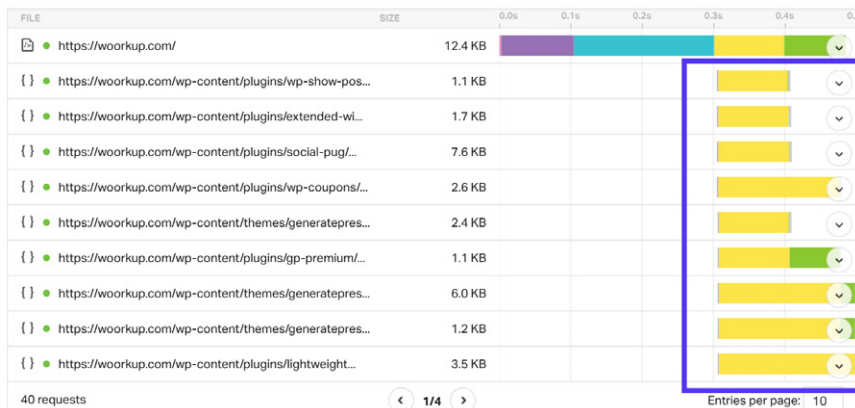
Requests

40

上記の劇的な改善の理由は、CDNにはロンドンにデータセンターがあり、すべてのアセットがそのロケーションにキャッシュされ、最小限の待ち時間で配信できたためです。

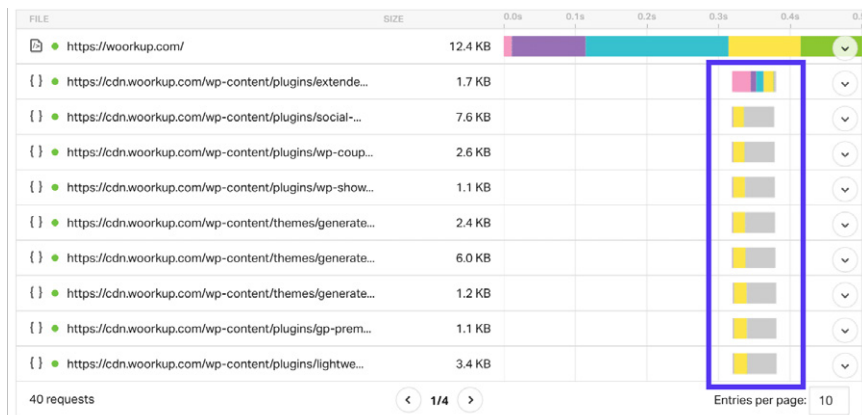
CDN無しでのTTFB

Pingdomの帯グラフの黄色い帯は待機時間で、「最初の1バイトが到着するまでの時間」(TTFB)のことです。CDNを使用しないときの平均TTFBは約98ミリ秒でした。



CDNでのTTFB

CDNを有効にすると、アセットの平均TTFBは約**15ミリ秒**に短縮。つまり、CDNを使用することで、**平均TTFBは84.69%改善したのです**。主にアセットがCDNのキャッシュから配信されたためです。



CDNを使用することで、ページの読み込み時間を43.2%短縮することができました!CDNの使用を是非ご検討ください。

CDNを有効にする手順について

WordPressウェブサイトでCDNを有効にするのはとても簡単です!次の手順に従ってください。

ステップ1

CDNプロバイダを選択し、契約します。月ごとの料金プランを提供する会社も、データ使用量をベースにした料金の会社もあります。ほとんどのプロバイダには料金計算ツールがあります。

- 各CDNプロバイダにも、似たようなマニュアルがあるはずですよ。
- kinsta.comでCoudflareのインストール手順とSucuriのインストール手順についても詳細なチュートリアルをご用意しております。

ステップ2

通常のプル型CDNをご利用の方は、CDN Enabler (WordPress.orgで入手可能)、WP Rocket (wp-rocket.meで入手可能) またはPerfmatters (perfmatters.ioで入手可能)などの無料プラグインを使用することによりCDNをWordPressウェブサイトと統合できます。これらのプラグインは自動的にアセットをCDNにリンクします。つまり、コンテンツをCDNに入れる作業は一切なく、手間のかからない方法なのです!リバースプロキシCDNの方は、通常プラグインは不要ですが、追加機能を有効にする必要がある場合があります。

Dashboard

Kinsta Cache

Posts

Media

Forms

Pages

Downloads

Jilt

Knowledge Base

Appearance

Plugins

Perfmatters Settings

OptionsCDNGoogle AnalyticsExtrasLicenseSupport

CDN

CDN options that allow you to rewrite your site URLs with your CDN URLs.

Enable CDN Rewrite ?

☐

CDN URL ?

https://cdn.perfmatters.io

Included Directories ?

wp-content,wp-includes

CDN Exclusions ?

.php,.xml

PerfmattersでWordPressのCDNを有効にする

Kinsta CDNを有効にする手順について

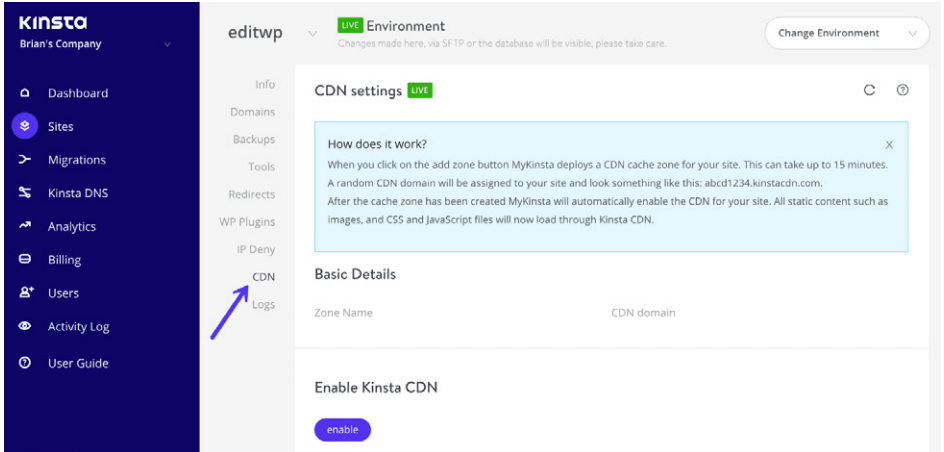
上記のCDNスピードテストの結果はいかがでしたか。このテストではKeyCDNを採用しました！そして、Kinsta CDNも同じく、34カ所以上のロケーションのあるHTTP/2とIPv6が有効になったコンテンツ配信ネットワークであるKeyCDNを採用しています。現在の配信リージョンはアメリカ、南アメリカ、ヨーロッパ、アフリカ、アジア、そしてオーストラリアです。



Kinstaのお客様には、各ホスティングプランを対象に無料のCDN帯域幅をご用意しております。2つの簡単なステップでKinsta CDNを有効にしましょう。

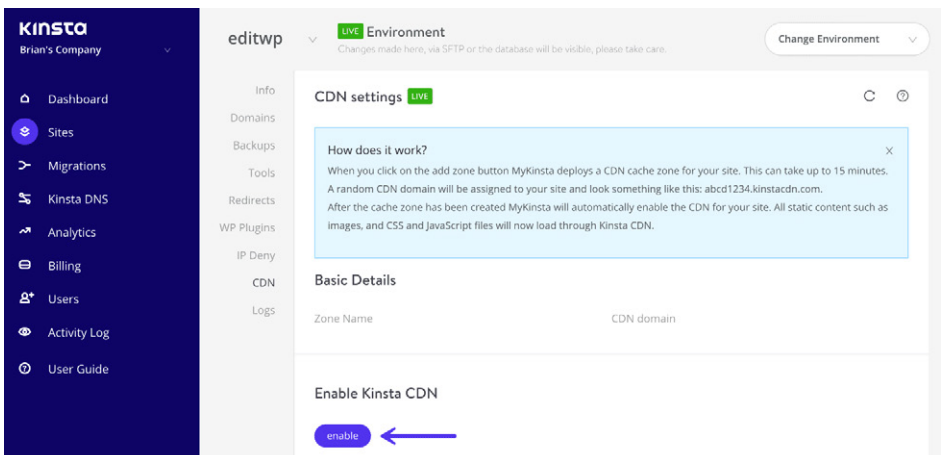
ステップ1

まず、MyKinstaダッシュボードにログインします。該当のウェブサイトをクリックし、Kinsta CDNタブをクリックします。



ステップ2

次に「Kinsta CDNを有効にする」をクリックします。数分後、CDNが自動的に機能し始め、アセットは世界中のキャッシュから配信されます。以上です。



CDNのさらなる最適化

その他の検討に値するCDNの最適化手法として、例えば下記があります。

- コメントが多ければ、gravatarにより生成されるリクエスト数が増えてsecure.gravatar.comからのトラフィックが多くなることがあります。CDNから読み込まれるように設定した方が良いでしょう。Kinstaのウェブサイトではそのようにしています。Kinstaのお客様は、是非当社のサポートチームにお問い合わせいただき、設定をご依頼ください。
- カスタムウェブフォント、Googleフォント、そしてfaviconもCDN上にホスティングすることができます。手順のチュートリアルへのリンクは、当社のウェブサイト<https://kinsta.com/speed-up-wordpress>でご覧いただけます。

第12章:

メディアと電子メールを 必要に応じてオフロードする



リクエストを生成するものはすべて、ウェブサイトのパフォーマンスに影響を与えます。何十万ものファイルを持つ大規模メディアのウェブサイトなら、これらを完全にオフロードした方が良いでしょう。オフロードは、CDNとは異なります。CDNを使用すると、元のデータはホスティング会社から消えずに残り、CDNにはその複数のコピーがあるだけです。

CDNアセットのキャッシュが期限切れになると、ホスティング会社のファイルの最新版を再クエリします。CDNは長期間ファイルをキャッシュすることを目的としていますが、POPが非常に多くてキャッシュの有効期限がリージョンごとに異なるため、再クエリが頻繁に起こる場合があります。

メディアやファイルをオフロードすると、ホスティング会社から元のファイルを別の場所に移動します。したがって、ファイルはウェブサイト上にあるように見えますが、実は完全に別の場所にあります。そうすると、ホスティング会社のクエリ数を減らすことに加えて、ディスクスペースも節約できます。

Amazon S3へメディアをオフロードする

最も人気のあるオフロードソリューションの一つはAmazon S3です。Amazon S3はストレージソリューションであり、Amazon Web Servicesの数ある商品の一つです。追加のバックアップを必要とする、または大きなファイル（ダウンロード、ソフトウェア、ビデオ、ゲーム、オーディオファイル、PDFファイルなど）を提供する大規模なウェブサイトを使用されるのが一般的です。Amazonは非常に信頼性が高いという実績があり、大規模なインフラストラクチャが用意されている為ストレージコストも極めて低いのが特徴です。S3の顧客には例えばNetflix、Airbnb、SmugMug、Nasdaqなどがあります。

パルクストレージのサービスである為、料金はWordPressホスティング会社よりもお手頃です。AWSにメディアをオフロードすることは、経費を節約する素晴らしい方法であり、最初の1年間は無料です(最大5 GBのストレージ)。また、メディアへのリクエストはAmazonから配信されるため、WordPressウェブサイトの負荷が軽減され、読み込み時間が短縮されます。



WP Offload Media Lite for Amazon S3プラグイン(WordPress.orgで入手可能)がお勧めです。このプラグインは、ファイルがメディアライブラリにアップロードされると、そのファイルをWordPressウェブサイトからAmazon S3に自動的にコピーします。また、オフロードされたメディアでCDNを使用して両方のテクノロジーのメリットを活かすことも可能です！

Google Cloud Storageにメディアをオフロードする



Google Cloud Storageも人気のあるオフロードソリューションです。KinstaはGoogle Cloud Platformを基盤としている為、Googleの技術とインフラストラクチャーには特に敬意を払っています。Googleはインフラストラクチャが大規模で、ストレージはパルクストレージという性質上、料金は非常にお手頃です。Googleの顧客は例えばSpotify、Vimeo、Coca-Cola、Philips、Evernote、Motorolaなどです。

無料のWP-Statelessプラグイン(WordPress.orgで入手可能)がお勧めです。このプラグインは、ファイルがメディアライブラリにアップロードされると、そのファイルをWordPressウェブサイトからGoogle Cloud Storageに自動的にコピーします。プラグインには、メディアのコピーをバックアップすることや、WordPressからメディアを削除してGoogle Cloud Storageから配信することなど、さまざまなモードがあります。

トランザクションメールとマーケティングメールをオフロードする

意識していない方が多いですが、電子メールもサーバーと、そのリソースに影響を与えます。電子メールの悪用に対して、アカウントを一時停止するホスティング会社（特に共有ホスティング会社）さえあります。大量のメールを送信しようとしている方にはこれは特に大きな問題です。このような理由で、サードパーティのトランザクション電子メールプロバイダーが生まれ、多くのホスティング会社が標準ポートでの電子メール配信を完全にブロックしています。**ご利用のホスティング会社を電子メール配信に使用することは、当社も決してお勧めしません。**

ニュースレターや大量の電子メールを送信する方には、以下の手段のどちらかをお勧めします。

- WordPressの一部ではないサードパーティのプロ向け電子メールマーケティングソフトウェアを使用すること
- WordPressと組み合わせてトランザクション電子メールサービスプロバイダー（HTTP、APIまたはSMTP）を使用すること

サードパーティのサービスを使用することによるその他の利点は次のとおりです。

- 電子メールの配信性が改善されます。メールプロバイダーが得意な作業をそのプロバイダーに任せましょう。
- バックリストされる可能性が減ります。
- DMARCレコードが設定できないホスティング会社もあります。

電子メールマーケティングツール

マーケティング用の電子メールは例えば、ニュースレター、新規商品および品機能のお知らせ、キャンペーン、イベントへの招待、オンボーディングのリマインダーなどです。

- **MailChimp** – KinstaはMailChimpを使用しています。
- **MailerLite**
- **Drip**

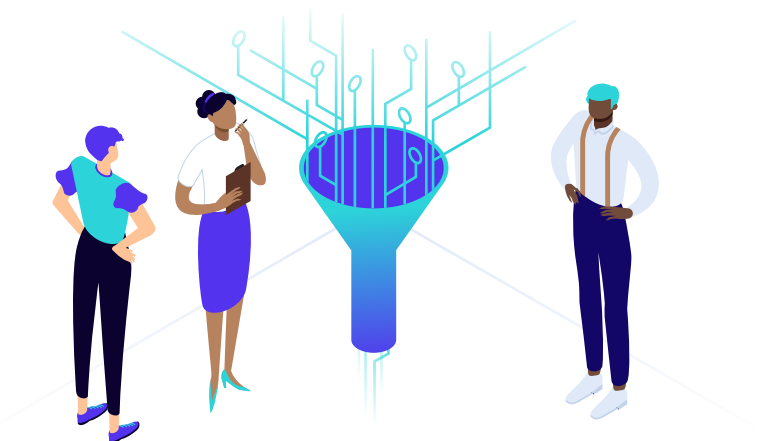
トランザクション電子メールサービス

トランザクション電子メールは例えば、WooCommerceまたはEDDからの購入の領収書、アカウント作成のお知らせ、出荷のお知らせ、アプリのエラーメッセージ、パスワードのリセットなどです。Kinstaは高い配信性を保証するためにサードパーティのSMTPプロバイダを使用しています。ただし、ボリュームにもよりますが、この作業をオフサイトに移動することをお勧めします。お勧めできるトランザクションメールサービスは以下のとおりです。

- **SendGrid** – KinstaはSendGridを使用しています。
- **Mailgun**
- **SparkPost**

第13章:

ボトルネックと 遅いプラグインの特定

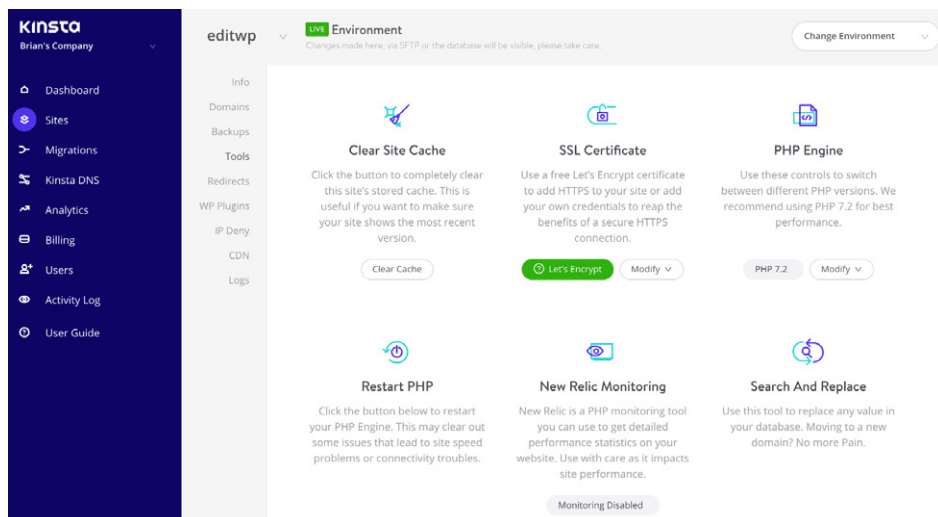


次に、WordPressウェブサイトのボトルネックの特定及び対処方法についてご説明します。

遅いプラグインとデータベースクエリを特定するに New Relicを使う

遅いデータベースクエリとプラグインを正確に特定するのに役立つ、優れたツールはいくつもあります。その中でも、KinstaはNew Relicを採用しています。New Relicは、ウェブサイトの詳細なパフォーマンス分析データが取得できるPHP監視ツールです。

Kinstaのお客様はMyKinstaダッシュボードを使用し、ご自分のNew Relicライセンスキーを追加することもできます。



ただし、ウェブサイトのパフォーマンスに影響を与えるため、十分に注意をしてNew Relicを使用してください。ウェブサイトにJavaScriptが追加されます。パフォーマンスのトラブルシューティングが必要な場合に有効にし、完了後は再び無効にすることをお勧めします。

遅いプラグインを特定する

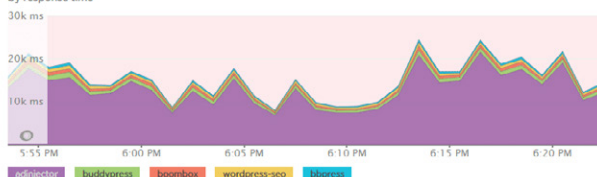
WordPressのあるプラグインのせいでウェブサイト全体のスピードが落ちるときの異常現象はプラグインの用途により異なりますが、遅いプラグインは、多くの場合、WordPressサイトのすべてのページに影響を与えます。例えば、下記の画像に表示されているウェブサイトでは、ウェブサイトのすべてのフロントエンドページのスピードが落ちました。New Relicのプラグインパフォーマンスについてのレポートは次のとおりです。

TIME PICKER
Last 30 minutes ending now

Sort by Most time consuming

adinjector	390 sec
buddypress	24.4 sec
boombox	17.6 sec
wordpress-seo	15.6 sec
bbpress	12.6 sec
easy social share buttons3	4.34 sec
zombity	4.22 sec

Top 5 plugins and themes
by response time



データを見ると、adinjectorプラグインが次に遅いプラグインの15倍以上の時間を消費していることが一目でわかります。

このようなデータが表示されると、プラグインのコーディングが不適切であるか効率が悪いと判断し、プラグインをすぐに無効にしたいくなるでしょう。場合によっては、そうした方がいいのですが、必ずしもそうとは限りません。プラグインの設定ミス、遅いデータベース、応答の遅い外部リソースなどにより、プラグインが消費する時間が長くなる場合があります。

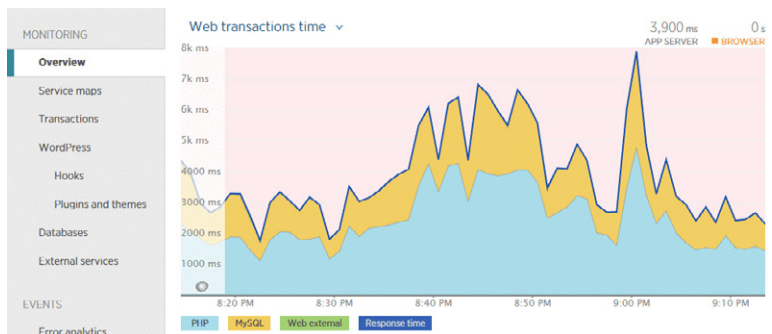
したがって、応答の遅いプラグインが表示されるときは、New Relicのその他の画面も確認し、追加の情報を取得した方が良いでしょう。プラグインを無効にする前に、トランザクション、データベース、および外部リソースをすべて確認しましょう。

データベースの負担が大きい為、ウェブサイト全体のスピードが落ちる

最適化されていないデータベースはWordPressサイト全体のスピードに悪影響を与えます。対処方法については、上述のとおりです。このデータベース関連の速度の低下は、New Relicでは次の2つの場所に表示されます。

- まず、概要画面では、MySQLアクティビティの量が多すぎる事が示されます
- 次に、「データベース」タブでは、1つ以上のデータベーステーブルに時間がかかりすぎていることが示されます。

概要画面からご説明すると、データベースに問題のあるウェブサイトは次のようになります。



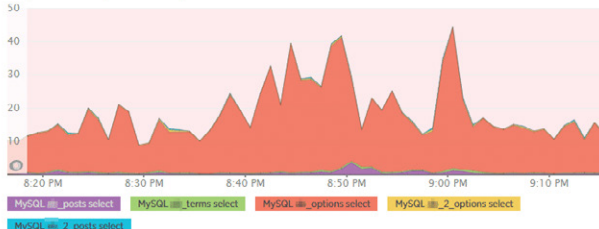
異常が発生しているデータベーステーブルまたはクエリを特定するには、「データベース」タブにアクセスします。

SORT BY
Most time consuming

MySQL options select	17.5 s
MySQL posts select	0.461 s
MySQL 2_options select	0.289 s
MySQL terms select	0.257 s
MySQL 2_posts select	0.11 s
MySQL postmeta select	0.078 s
MySQL options update	0.0467 s
MySQL termmeta select	0.0348 s

MySQL overview

Top database operations by time consumed



「データベース」タブには、最も時間のかかるテーブルとクエリの種類が表示されます。リストの項目の一つを選択すると、サンプルクエリなどの詳細情報が確認できます。

SORT BY
Most time consuming

MySQL options select	17.5 s
MySQL posts select	0.461 s
MySQL 2_options select	0.289 s
MySQL terms select	0.257 s
MySQL 2_posts select	0.11 s
MySQL postmeta select	0.078 s
MySQL options update	0.0467 s
MySQL termmeta select	0.0348 s
MySQL 2_options update	0.0309 s
MySQL other	0.0281 s
MySQL sitemeta select	0.0115 s

MySQL options select

Slow query trace 2,510 ms /archive-product

Query

SELECT option_name, option_value FROM wp_options WHERE autoload = ?

Database instance

Hz0-xxxxx-xxxxx-xxxxx-xxxxx

Database name

xxxxx-xxxxx-xxxxx-xxxxx-xxxxx

Stack trace

```
mysql_query called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-includes/wp-db.php (1687)
db::do_query called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-includes/wp-db.php (1775)
in wpdb::query called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-includes/wp-db.php (2395)
iget_results called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-includes/option.php (189)
loptions called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-includes/functions.php (1364)
blog_installed called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-includes/load.php (533)
in wp_not_installed called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-settings.php (140)
in require_once called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-config.php (65)
in require_once called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-load.php (37)
in require_once called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/wp-blog-header.php (13)
in require called at /www/xxxxx-xxxxx-xxxxx-xxxxx-xxxxx/public/index.php (17)
```

上記の例では、wp_optionsテーブルの自動読み込みデータで異常が発生していることが分かります。なお、これについては既にご説明したとおりです。wp_optionsテーブルを簡単に分析すると、約250 MBのデータもこのテーブルから自動読み込みされていることが確認できました。このウェブサイトは間違いなくデータベースのメンテナンスと最適化が必要でしょう。

kinsta.comではまた、WordPressウェブサイトのパフォーマンス異常の処理にNew Relicを使用する方法についてのチュートリアルをご用意しております。

無料のQuery Monitorプラグインを使用する

無料のWordPressプラグインQuery Monitorも利用できます。(WordPress.orgで入手可能)遅いデータベースクエリ、AJAXの呼び出し、REST APIリクエストなどの特定とデバッグの機能があります。さらに、このプラグインは、スクリプトの依存関係、ページ生成中に発生したWordPressフック、ホスティング環境の詳細、現在のページが満たす条件付きクエリタグなどのウェブサイトの詳細情報も報告してくれます。

Query Monitor

Page generation time	Peak memory usage	Database query time	Database queries	Object cache
0.0321 0.0% of 8s limit	2,048 kB 0.0% of 16,777,216 kB limit	0.0022	SELECT: 3	99.9% hit rate External object cache: true

\$wpdb Queries

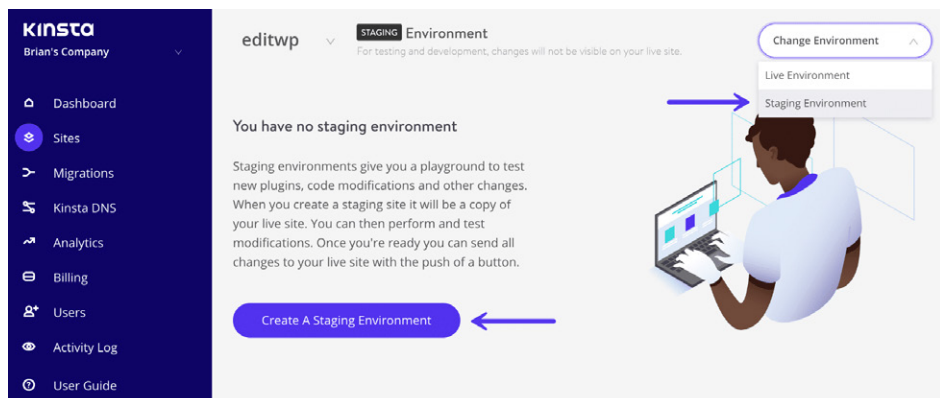
	Query	Caller	Component	Rows	Time
	All ▾	All ▾	All ▾		
1	SELECT user_id, meta_key, meta_value FROM wp_usermeta WHERE user_id IN (1) ORDER BY umeta_id ASC	update_meta_cache() wp-includes/meta.php:826	+ Plugin: coming-soon-builder	39	0.0004
2	SELECT wp_posts.* FROM wp_posts WHERE 1=1 AND (wp_posts.ID = '2') AND wp_posts.post_type = 'page' ORDER BY wp_posts.post_date DESC	WP_Query->get_posts() wp-includes/query.php:3655	+ Core	1	0.0017
3	SELECT * FROM wp_posts WHERE (post_type = 'page' AND post_status = 'publish') AND post_parent = 2 ORDER BY wp_posts.post_title ASC LIMIT 0,1	get_pages() wp-includes/post.php:4598	+ Core	0	0.0001
Total Queries: 3					0.0022

このプラグインはJohn Blackbourn氏により開発されました。彼は現在Human Madeの開発者であり、以前はWordPress.com VIPで働いていました。つまり、WordPressに非常に詳しい人物です。Query Monitorは、2013年にWordPressのプラグインディレクトリに追加され、現在10,000以上のアクティブインストールを誇ります。開発プラグインとしては素晴らしい数字です。プラグインが5つ星のうち5と評価され、開発者の間では群を抜いて人気があります。

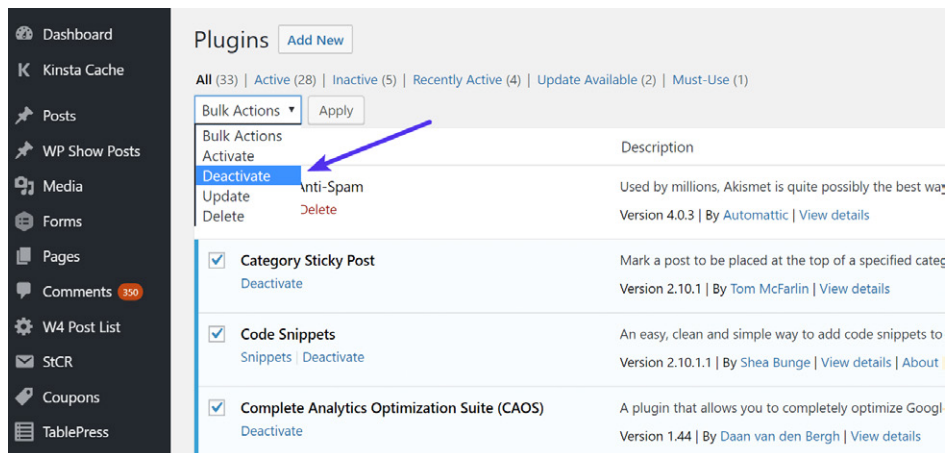
本番サイトに触れずステージングサイトを採用する

「ステージング環境がなければ、どうにもならない」と言えるほどパフォーマンス面での課題を処理する際に不可欠なものです。幸いなことに、Kinstaではワンクリックステージング環境が利用可

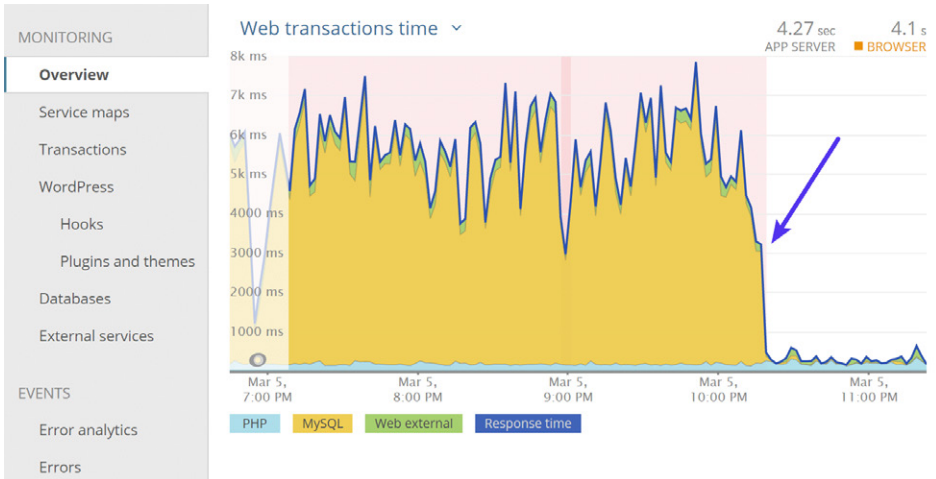
能です。ご利用のWordPressホスティング会社がステージング環境を提供していない場合には、(WordPress.orgで入手可能)WP Stagingなどのプラグインを使用することもできますが、あまり使用しやすいとは言えません。



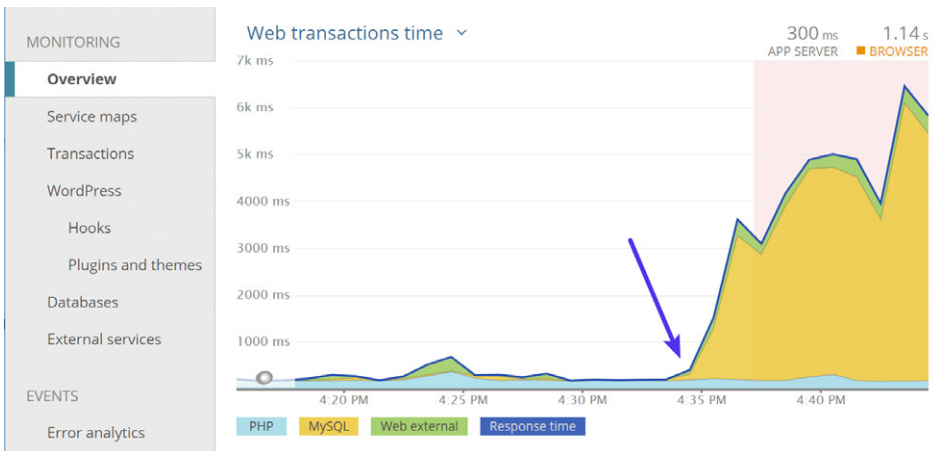
ステージング環境が立ち上がった後、すべてのプラグインを無効にします。ウェブサイトのコピーである為、本番サイトを破壊する恐れはありません。異常を絞り込む最も簡単な方法の一つです。プラグインにアクセスし、すべてを選択し、一括オプションの「無効にする」オプションを選択するだけです。



これを実行した後、New RelicまたはQuery Monitorで応答時間を監視し、結果を確認できます。次の例では、応答時間がすぐに通常に戻ったため、異常の原因はプラグインの一つであることがわかりました。次に、原因が見つかるまでプラグインを一つずつ再度有効にしていきます。



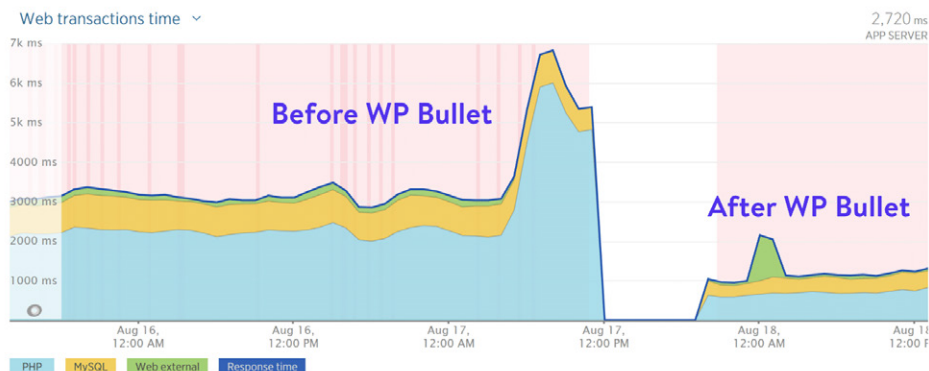
異常の原因となっていたプラグインを有効にしたときの例を示します。読み込み時間（ウェブトランザクション時間）がすぐに上がりました。



スピード低下の原因となっているプラグインが特定できた後は、どうすればいいでしょうか？次の方法をお勧めします。

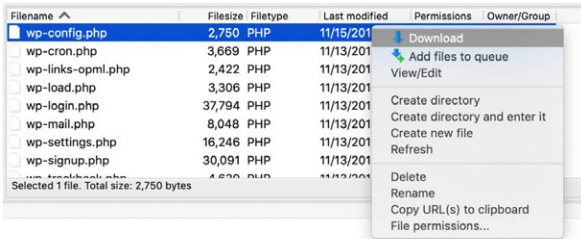
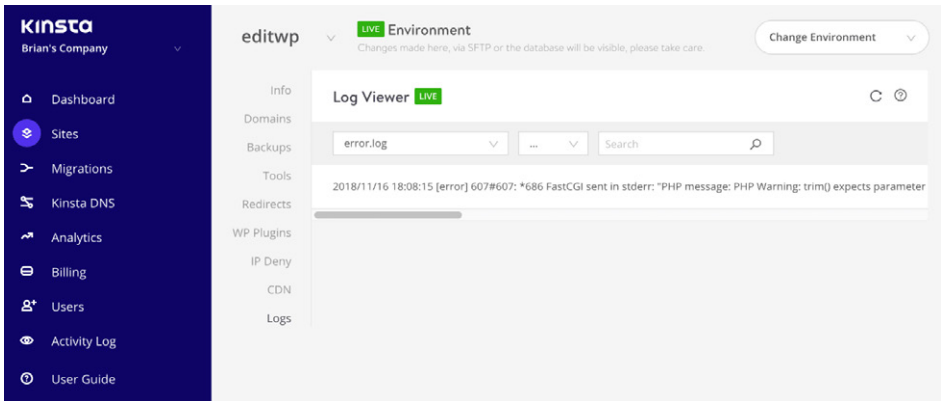
1. ご利用のプラグインとテーマが最新バージョンでない場合には、更新してください。
2. 該当するプラグイン及びテーマの開発者にお問い合わせください。
3. 同じ用途の代替となるプラグインを探してみてください。
4. PHPのご利用のバージョンが異常の原因である場合もあります。PHPエンジンをより低いバージョンに切り替え、該当のプラグイン及びテーマが正常に機能しているか確認してください。

または、WordPress開発者を雇いし異常の処理を依頼することもできます。パフォーマンス関連の課題の場合には、wp-bullet.comのMike Andreason氏をお勧めします。彼は、パフォーマンスの最適化を専門とする、フルタイムのCodeableの開発者で、Kinstaの多くのお客様のウェブサイトを次の段階に進めるサポートを担当してきました。



エラーログを確認する

エラーログを確認する作業は確かに退屈ですが、WordPressプラグイン関連のパフォーマンスの課題が明らかになります。Kinstaのお客様は、MyKinstaダッシュボードを使用することによりエラーログ、キャッシュログ、アクセスログなどを簡単に確認できます。



wp-config.phpファイルにコードを追加することにより、エラーログを有効にすることもできます。まず、SFTP経由でウェブサイトにアクセスします。次にwp-config.phpファイルを編集できるようにダウンロードします。

注:必ず最初にこのファイルのバックアップを作成してください。

`/* That's all, stop editing! Happy blogging. */` を探し、その直前に次を追加します(下記を参照)。

```
define( 'WP_DEBUG', true );
```

```
37
38
39
40 define( 'WP_DEBUG', true );
41
42 /* That's all, stop editing! Happy blogging. */
43
44 /** Absolute path to the WordPress directory. */
45 if ( !defined('ABSPATH') )
46     define('ABSPATH', dirname(__FILE__) . '/');
47
48 /** Sets up WordPress vars and included files. */
49 require_once(ABSPATH . 'wp-settings.php');
```

wp-config.phpファイルに既に上記のコードが存在しているにも関わらず「false」に設定されている場合には、単に「true」に変更してください。これでデバッグモードが有効になります。注：存在している場合には、WordPress管理画面にも警告またはエラーが表示されます。

次に、WP_DEBUG行の直後に次のコードを追加することにより、デバッグログがすべてのエラーのあるファイルについて送信されるようにします（下記を参照）。

```
define( 'WP_DEBUG_LOG', true );
```

```
38
39
40 define( 'WP_DEBUG', true );
41 define( 'WP_DEBUG_LOG', true );
42
43 /* That's all, stop editing! Happy blogging. */
44
45 /** Absolute path to the WordPress directory. */
46 if ( !defined('ABSPATH') )
47     define('ABSPATH', dirname(__FILE__) . '/');
48
49 /** Sets up WordPress vars and included files. */
50 require_once(ABSPATH . 'wp-settings.php');
```

変更を保存し、ファイルをサーバーに再びアップロードします。エラーは/wp-content/フォルダにあるdebug.logファイルに記録されます。何らかの理由でこのファイルが表示されない場合には、自分で作成できます。

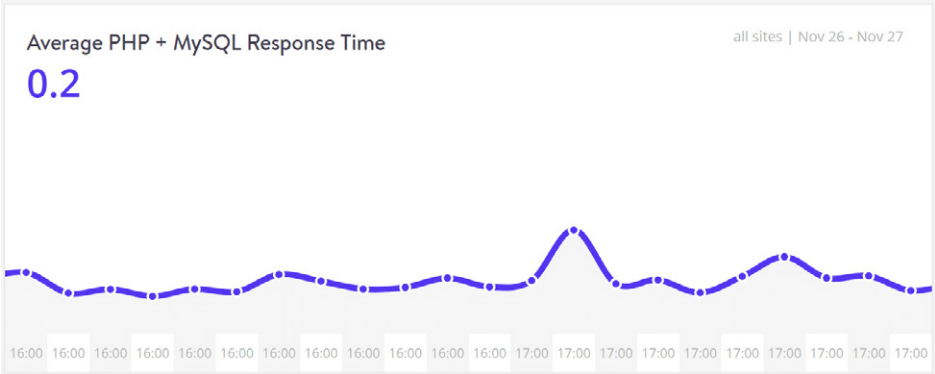
MyKinstaの分析機能を使用する

Kinstaのお客様は、MyKinstaの分析ツール内蔵のパフォーマンス関連情報もご利用いただけます。

パフォーマンス監視セクションでは、PHP+MySQLの平均応答時間、PHPのスループット、AJAXの使用状況、平均アップストリーム時間のトップリストと最大アップストリーム時間のトップリストが確認できます。

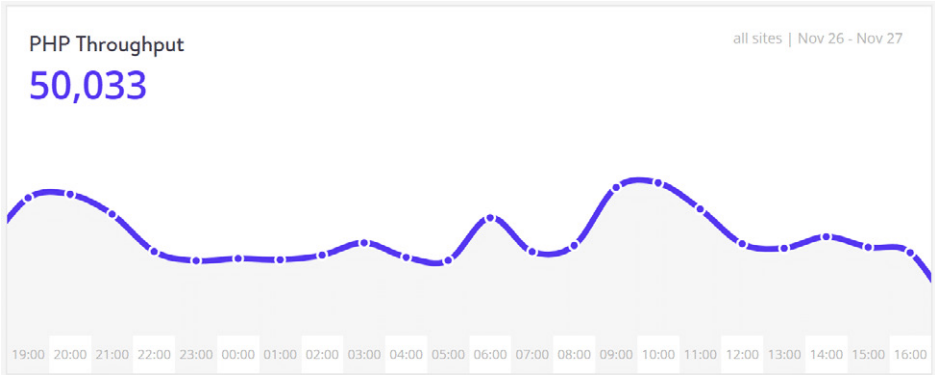
PHP+MySQLの平均応答時間

WordPressサイトにアクセスする度に、ページに表示されるデータをコンパイル、クエリするために、そこでは、PHPとMySQLが使用されることになります。このグラフは、キャッシュされていない動的リクエストに対するPHPエンジンとMySQLエンジンの平均応答時間を示しています。この応答時間を理解することにより速度低下に対処できる場合があります。



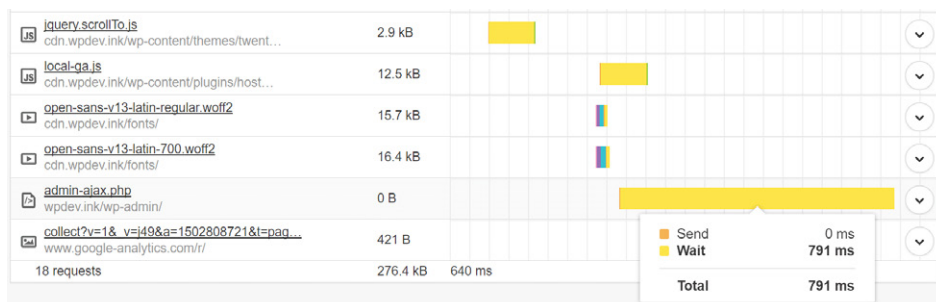
PHPのスループット

スループットとは、アプリケーションが処理できる1秒あたりのトランザクション数を意味します。このレポートでは、WordPressサイトのPHPに関するスループットが表示されます。つまり、PHPアセットがリクエストされた回数が記載されるのです。

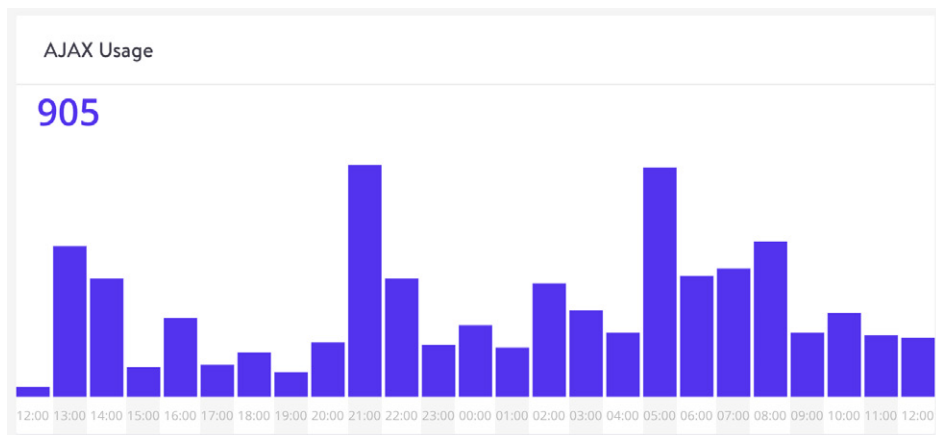


AJAXの使用状況

AJAXは、ポストバック及びページ全体の更新を必要とせずに、サーバー及びデータベースとの通信を実施するクライアントサイドのスクリプトです。WordPressなら、スピードテストでよく見かける内容でしょう。AJAX関連でよくある異常は、プラグインのせいでAJAXの使用が急上昇することとバックエンドのCPUの異常です。



MyKinstaの分析機能にあるAJAX使用状況レポートは、特定の期間中に特定のAJAXの使用の急増が発生しているかどうかを確認できるため、特に便利です。このグラフはadmin-ajaxリクエストの数を示しています。次に、前述の投稿に記載されている工夫を実施し、発生原因を絞り込むことができます。



PHP+MySQLの平均応答時間のトップリスト

こちらは、PHPとMySQLの平均応答時間のトップリストです。ここにある数値は1回限りのピークである可能性がある為、このリストを「最大アップストリーム時間のトップリスト」と比較することをお勧めします。

PATH	REQUESTS	TIME
/wp-admin/update.php?action=upload-plugin	1	2.19 s
/wp-cron.php?server_triggered_cronjob	93	1.97 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&plugin_status=all&paged=1&s&_wpnonce=70f7d2faca	1	1.47 s
/wp-admin/plugins.php?activate=true&plugin_status=all&paged=1&s=	2	1.37 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=729c628974	1	1.07 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=70f7d2faca	1	0.91 s
/wp-cron.php?doing_wp_cron=1542348999.8110280036926269531250	1	0.67 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=113bcb7711	1	0.53 s
/wp-admin/plugins.php	3	0.49 s
/wp-admin/plugin-install.php	1	0.48 s

最大アップストリーム時間のトップリスト

アップストリーム時間とは、NGINX (およびアップストリームサーバー) がリクエストを処理して応答を送信するまでにかかる時間のことです。この時間はミリ秒単位の分解能で秒単位で測定されます。

PATH	REQUESTS	TIME
/wp-cron.php?server_triggered_cronjob	93	180.00 s
/wp-admin/plugins.php?activate=true&plugin_status=all&paged=1&s=	2	2.60 s
/wp-admin/update.php?action=upload-plugin	1	2.19 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&plugin_status=all&paged=1&s&_wpnonce=70f7d2faca	1	1.47 s
/hello-world/	5	1.46 s
/wp-admin/plugins.php?action=deactivate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=729c628974	1	1.07 s
/	83	0.93 s
/wp-admin/plugins.php?action=activate&plugin=wp-rocket%2Fwp-rocket.php&_wpnonce=70f7d2faca	1	0.91 s
/wp-admin/plugins.php	3	0.85 s
/wp-cron.php?doing_wp_cron=1542348999.8110280036926269531250	1	0.67 s

ウェブサイトがハッキングされている恐れがある

パフォーマンスの異常が特定できない場合には、ウェブサイトがハッキングされているか、マルウェアに感染している、またはDDoS攻撃を受けている可能性が大いに考えられます。これはウェブサイトのスピードとWordPress管理ダッシュボードの反応性にも影響を与える場合があります。このよ

うなケースでは、次をお勧めします。

1. Cloudflare及びSucuriなどのプロキシサーバーとWAFを採用します。
2. 上記のサービスを使用して不正なIPアドレスをブロックします。Kinstaのお客様はMyKinstaダッシュボードからもIPアドレスをブロックすることができます。
3. ジオブロッキングもご利用いただけます。極めて悪質なトラフィックを生成する国もあります。攻撃を受けている場合には、一時的または恒久的にその国全体をブロックする必要がありますかもしれません。

エラーコード(HTTPステータスコード)を使用したトラブルシューティング

HTTPステータスコードとは、ウェブページの上部に貼り付けられるウェブサーバーからの短いメモのようなものです。ウェブページの一部ではなく、ページを表示するリクエストがサーバーにより受信されたときの状況を知らせる、サーバーからのメッセージであり、課題を処理する際に不可欠なものです！

40以上のステータスコードがありますが、WordPressユーザーがよく悩まされるものについてご紹介します。kinsta.comでは、次の各エラーコードの処理方法についてのチュートリアルをご用意しております。

429: 「要求が多すぎる」: ユーザーが短時間に大量の要求を送信してきたときにサーバーにより生成されます(レート制限)。ボット及びスクリプトがウェブサイトにアクセスしようとしている時に起きることがあります。この場合には、WordPressのログインURLを変更することをお勧めします。

429 Too Many Requests

nginx

500: 「サーバーにエラーが発生したため、リクエストを完了できませんでした」: 一般的なコードで、「サーバー内部エラー」を意味します。サーバー内部にエラーが発生したせいで、リクエストしたリソースが配信されません。このコードは、サードパーティ製のプラグイン、PHP不良、データベースへの接続が切断されたことにより生成されます。

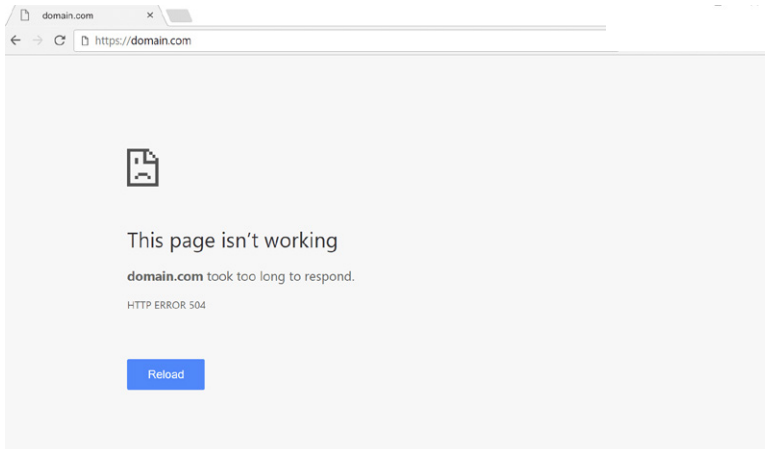
Error establishing a database connection

502:「不正なゲートウェイ」:このエラーコードは、あるサーバーが別のサーバーから無効な応答を受け取ったときに表示されます。クエリ及びリクエストに時間がかかり過ぎるため、サーバーによりキャンセルされ、データベースへの接続が切れることがあります。



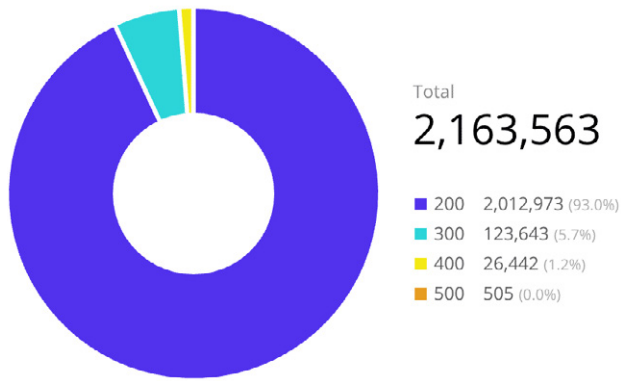
503:「サーバーは現在このリクエストを処理することができません」:このコードは、サーバーが過負荷で、追加のリクエストが処理できない時に返されます。

504:「ゲートウェイとして機能しているサーバーが、他のサーバーからの応答を待ってタイムアウトしました」:サーバーが2つあり、その一つが相手側のサーバーからの応答を待っている間にタイムアウトしているときにこのコードが返されます。



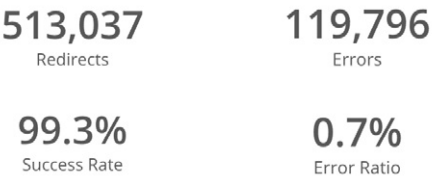
MyKinstaの分析ツールでもHTTPコードについて詳しく調べることができます。レスポンスコードの内訳レポートには、リクエストされたリソースに配信されたHTTPステータスコードの一覧が表示されます。

応答コードの内訳



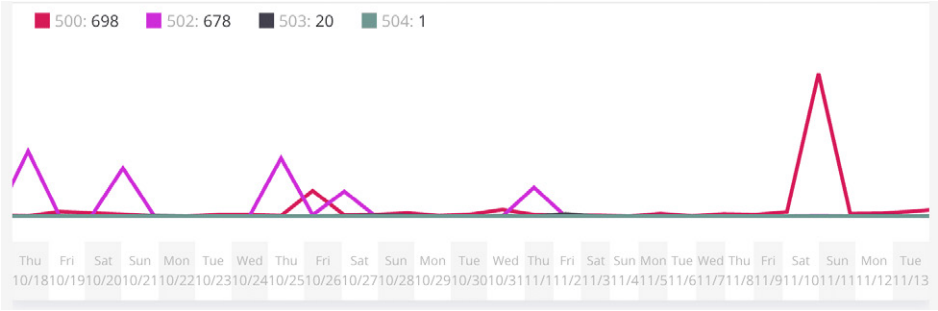
応答の分析データレポートには、発生したリダイレクトの総数、エラー、成功率、およびエラー率が表示されます。どのWordPressウェブサイトでも、多かれ少なかれエラーが発生します。これはいたって正常なことです。

応答の分析データ



500エラー、400エラー、リダイレクトなどのすべてのエラーコードには内訳のレポートがあります。

500エラー内訳



第14章:

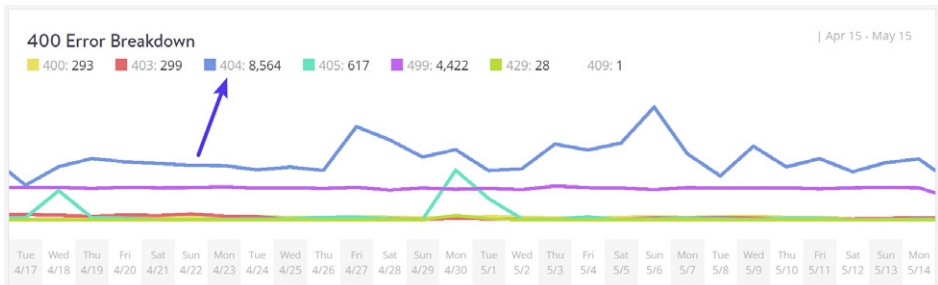
バックエンド最適化に 関する推奨事項



それでは、バックエンドを最適化することによりWordPressをスピードアップする方法についてご説明します。バックエンドとは、PHP、HTTPキャッシュヘッダー、GZIP圧縮など、完全にサーバーに処理されるものを指します。

404ページを軽くする

非常に動的なウェブサイトでは、404エラーが頻繁に発生することがわかりました。あなたのウェブサイトでも想像以上に発生しているかもしれません！当社のMyKinsta分析ツールを使用すると、正確な数を確認することができます（下記参照）。



本エラーが特に問題となる理由の一つは、404ページの多くが非常にリソースを消費するものであるためです。非常に動的なWordPressウェブサイトでは、重い404ページを避けましょう。可能な

限りデータベースへのクエリを回避する単純な404テンプレートを作成しましょう。もちろん、しっかり時間をかけて、404エラーを処理した方がより良いでしょう。本エラーはリソースを多用するだけではなく、ユーザーエクスペリエンスにも悪影響を及ぼします。

PHPワーカーの数を増加する

PHPワーカーという用語をはじめて耳にするという方もいらっしゃるかもしれませんが、Kinstaを含め、PHPワーカーをベースにリクエストを制限するホスティング会社が多く存在します。（共有ホスティング会社は通常CPU及びRAMをベースに制限しています）

特定の時間にウェブサイトが同時に処理できるリクエストの数はPHPワーカーに依存します。簡単に言うと、ウェブサイトに対するキャッシュされていないリクエストは、すべてPHPワーカーにより処理されます。たとえば、ウェブサイトへのリクエストが同時に4つもされているに関わらず、ウェブサイトにPHPワーカーが2つしかない場合には、それらのリクエストのうち2つは処理され、残りの2つは最初の2つが完了するまで待機状態になります。

WordPressのメンバーシップサイトの最大の課題の一つが、「キャッシュされていないリクエスト」であることは先にご説明したとおりです。PHPワーカーはそれらのリクエストを処理するため、非常に重要なものです。そのようなウェブサイトでは、すべてのリクエストが遅滞なく処理され、正常に完了することを担保するために追加のPHPワーカーが必要になる場合が多々あります。

PHPワーカーを使用すぎると、待機している古いリクエストをキャンセルし始める為、500エラーが発生する可能性があります。Kinstaの各ホスティングプランには、一定数のPHPワーカーがあてがわれています。ご自身のウェブサイトの適切なPHPワーカー数が分からないという方は、是非当社の営業チームまたはサポートチームへ、チャットからご相談ください。

GZIP圧縮を採用する

GZIPはファイルの圧縮と解凍に使用されるソフトウェアアプリケーションとその圧縮データのファイルフォーマットです。GZIP圧縮はサーバー側で有効にされるもので、HTML、スタイルシート、およびJavaScriptファイルのサイズをさらに縮小します。

ウェブブラウザがウェブサイトにアクセスするときに、content-encoding:gzipというHTTPヘッダーが存在するかを確認することにより、ウェブサーバーでGZIPが有効になっているかを確認します。ヘッダーが検出された場合には、ファイルの圧縮版が配信されます。一方で、検出されない場合には、非圧縮版が配信されます。GZIPを有効にしていない場合は、Google PageSpeed InsightsやGTmetrixなどのスピードテストツールで警告及びエラーが発生することがあります。

gzip圧縮を有効にすることにより、ウェブページのサイズが小さくなり、リソースをダウンロードする時間とクライアントのデータ使用量が低下し、レンダリングの開始時間が改善されます。これは現在ほとんどのホスティング会社では標準的になっていますが、そうでない会社もあるかもしれません。

ホットリンクの概念は簡単です。インターネットで画像を見つけ、自分のウェブサイトでその画像のURLを直接使用することをホットリンクと言います。該当の画像はあなたのウェブサイトに表示されますが、元のウェブサイトから配信されます。これは非常に便利ですが、実はホットリンクされたウェブサイトのリソースを使用しているため盗難にあたります。たとえてみると、隣人の車から吸い出したガソリンを自分の車に給油して使用してしまうような行為なのです。

ホットリンクは**ターゲットサーバーのリソースを大幅に消費**することがあります。例えば、共有のWordPressホスティング会社を利用しているウェブサイトの画像をHuffington Postが突然リンクしてしまったとします。ウェブサイトの時間当たりのクエリ数が数百から数十万に増加するでしょう。これにより、ホスティングアカウントが停止されることもあります。従って、(上記のような異常を処理できる)パフォーマンスの高いホスティング会社を利用するだけでなく、ホットリンク保護も有効にし、上記のような事態を予防しましょう。

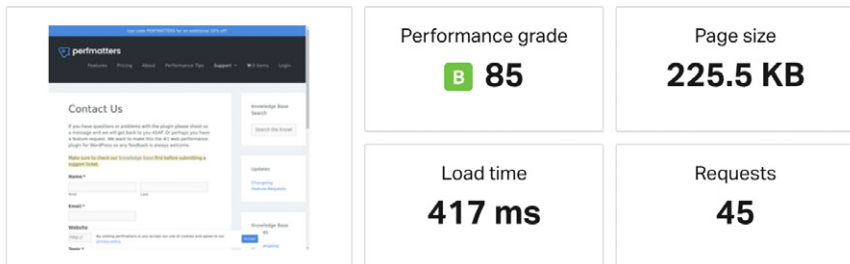
リダイレクトを最小限に抑えてサーバーレベルで追加する

リダイレクトが多すぎる状態には十分気を付けなければなりません。301リダイレクト、HTTPからHTTPSへのリダイレクト、またはwwwから非wwwへのリダイレクトは問題ありません。むしろ、必要であるときもあります。ただし、それぞれのリダイレクトがウェブサイトのパフォーマンスへ悪影響を与えます。リダイレクトを複数使用する際は、ウェブサイトのパフォーマンスへの影響を把握したうえで使用するようにしましょう。固定ページおよび投稿のリダイレクト、イメージのリダイレクトなどすべてのリダイレクトに当てはまります。

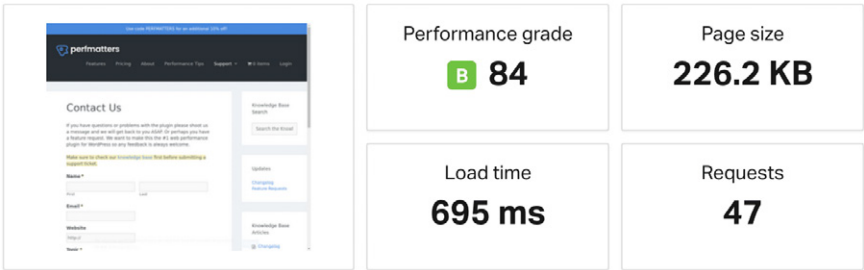
リダイレクトは、Pingdomでは応答ヘッダーのステータス301または302とともに青い点として表示されます。

FILE	SIZE	0.0s	0.1s	0.2s
 http://www.perfmatters.io/contact	238.0 B			
Response Headers http://www.perfmatters.io/contact 301				
content-length	162			
x-content-type-options	nosniff			

リダイレクトが何件あるとウェブサイトに影響を及ぼすのでしょうか。テストしてみましょう。まず、<https://perfmatters.io/contact/>の問い合わせページでスピードテストを行います。以下のとおり、読み込み時間は417ミリ秒です。



その後、URLを少し変更して(<http://www.perfmatters.io/contact>)スピードテストを再度行い、複数のリダイレクトがある場合の影響を確認します。ご覧のとおり、同じページの読み込み時間は695ミリ秒もかかりました。66%の増加です。恐ろしい！



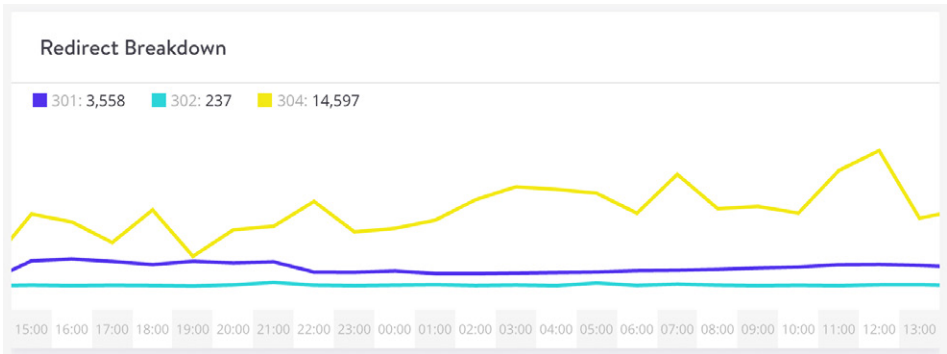
無料のWordPressプラグインを使用してリダイレクトを設定すると、そのリダイレクトはwp_redirect関数を使用するためパフォーマンス上の問題が発生することがよくあります。また、wp_optionsテーブルに自動読み込みデータを追加し、データベースの肥大化を促進するものもあります。これは、サーバーレベルで追加すべきでしょう。KinstaではMyKinstaのリダイレクトルールツールを使用することによりこれが可能となります。

The image shows a 'Add Redirect Rule' dialog box with the following fields and options:

Status	Domain	Redirect From	Redirect To
301	All Domains	^/blogpost	/newblogpost

Buttons: Cancel, Add Rule

また、MyKinstaの分析ツールでは、ウェブサイトのすべてのリダイレクトの数と内訳も確認できます。301、302、および304の総数を参照してください。



Cronジョブを暴走させないようにする

CRONジョブ (WP-Cron) はWordPressウェブサイトの繰り返しタスクをスケジュールするのに使われます。ただし、時間が経つにつれ、これらは制御不能になり、パフォーマンス異常を引き起こす場合があります。無料のWP Controlプラグイン (WordPress.orgで入手可能) を使用することにより、ウェブサイトのすべてのCronジョブが確認できます。

また、WordPress内蔵のCron処理ツール (WP-Cron) 関係でもパフォーマンスの問題が発生することがあります。ウェブサイトにPHPワーカーが足りない場合には、リクエストが入ると、WordPressがcronを生成しますが、cronはワーカーを待つ必要があるため、ずっと待機中になることもあります。その場合、WP-Cronを無効にし、代わりにシステムcronを使用した方が良いでしょう。公式のプラグインハンドブックでもこの方法が推奨されています。

WP-Cronを無効にするには、wp-config.phpファイルの「That's all, stop editing! Happy blogging.」行の直前に次のコードを追加します。注：これにより、wp-cron.phpから直接呼び出した場合ではなく、ページロードでの実行が無効になります。

```
define( 'DISABLE_WP_CRON', true );
```

```
60  /**
61   * WordPress Database Table prefix.
62   *
63   * You can have multiple installations in one database if you give each
64   * a unique prefix. Only numbers, letters, and underscores please!
65   */
66  $table_prefix = 'wp_';
67
68  define( 'DISABLE_WP_CRON', true ); ←
69
70  /* That's all, stop editing! Happy blogging. */
71
72  /** Absolute path to the WordPress directory. */
73  if ( ! defined( 'ABSPATH' ) )
74      define( 'ABSPATH', dirname( __FILE__ ) . '/' );
75
76  /** Sets up WordPress vars and included files. */
77  require_once ABSPATH . 'wp-settings.php';
78
```

Line 68, Column 33

次に、サーバーからwp-cron.phpをスケジュールする必要があります。Kinstaでは、システムcronは有効になっており、デフォルトで15分毎に実行されます。頻度を増やすには、当社のサポートチームまでお問い合わせください。SSHに詳しい方は、コマンドラインからもcronの管理ができます。

cache-controlヘッダーとexpiresヘッダーを追加する (キャッシュの長さを指定する)

WordPressウェブサイトのすべてのスクリプトに、HTTPキャッシュヘッダーが添付されているはずです。キャッシュヘッダーは、**ファイルのキャッシュの期限が切れる時点を指定します**。この問題を処理するには、WordPressホスティング会社のcache-controlヘッダーとexpiresヘッダーの設定が適切であるかどうかを確認します。適切でない場合には、スピードテストツールでexpiresヘッダーを追加する必要がある旨の警告、または「ブラウザのキャッシュを利用する」警告が表示されます。

cache-controlヘッダーがクライアント側のキャッシュをオンにしリソースの最大経過時間を設定している一方、expiresヘッダーはリソースの有効期限を指定するものです。両方のヘッダーを同時に使用することはできませんが、必ずしも両方を追加する必要はありません。cache-controlの方が新しく、推奨されているヘッダーです。

Kinstaでは、すべてのサーバーリクエストに自動的にHTTPキャッシュヘッダーを追加しています。CDNをご利用の方も、これらのヘッダーがCDNプロバイダーにより追加されるはずです。

Name	Headers	Preview	Response	Timing
☐ editwp.com	▼ Response Headers access-control-allow-origin: * cache-control: max-age=315360000 content-encoding: gzip content-type: application/javascript; charset=UTF-8 date: Wed, 07 Nov 2018 03:17:34 GMT etag: W/"5bdc7194-2efa" expires: Thu, 31 Dec 2037 23:55:55 GMT last-modified: Fri, 02 Nov 2018 15:47:32 GMT server: nginx status: 200 vary: Accept-Encoding x-content-type-options: nosniff			
☐ admin-bar.min.js?ver=5.0-beta3-43867				
☐ skip-link-focus-fix.js?ver=20151215				
☐ wp-embed.min.js?ver=5.0-beta3-43867				
☒ be9a4e9519964b933dea6723d480c95a?s=52&d=mm&r=g				
☒ be9a4e9519964b933dea6723d480c95a?s=128&d=mm&r=g				
☐ wp-emoji-release.min.js?ver=5.0-beta3-43867				
☐ data:application/fo...				
☐ url-metrics				
☐ tpc-check.html				
☐ postmessage.js				
☐ tpc-check-add-cookie.js				
12 requests 6.5 KB transferred Finish: 1.19 s DOMContentLoaded: 452 ms ...				

これらのヘッダーがご利用のサーバーにない場合でも、手動で追加できます。

Nginxでcache-controlヘッダーを追加する

Nginxでcache-controlヘッダーを追加するには、サーバーコンフィギュレーションのサーバーの場所またはブロックに次のコードを追加します。

```
location ~* \.(js|css|png|jpg|jpeg|gif|svg|ico)$ {
    expires 30d;
    add_header Cache-Control "public, no-transform";
}
```

Nginxでexpiresヘッダーを追加する

Nginxでexpiresヘッダーを追加するには、サーバーブロックに次のコードを追加します。この例では、ファイルの種類に基づいて異なる有効期限を指定する方法を確認できます。

```
location ~* \.(jpg|jpeg|gif|png|svg)$ {
    expires 365d;
}

location ~* \.(pdf|css|html|js|swf)$ {
    expires 2d;
}
```

Apacheでcache-controlヘッダーを追加する

Apacheでcache-controlヘッダーを追加するには、.htaccessファイルに次のコードを追加します。コードのスニペットは、ファイルの上部または下部(# BEGIN WordPressの前または# END WordPressの後)に追加できます。

```
<filesMatch ".(ico|pdf|flv|jpg|jpeg|png|gif|svg|js|css|swf)$">
Header set Cache-Control "max-age=84600, public"
</filesMatch>
```

Apacheでexpiresヘッダーを追加する

Apacheでexpiresヘッダーを追加するには、.htaccessファイルに次のコードを追加します。

```
## EXPIRES HEADER CACHING ##
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType image/jpg "access 1 year"
ExpiresByType image/jpeg "access 1 year"
ExpiresByType image/gif "access 1 year"
ExpiresByType image/png "access 1 year"
ExpiresByType image/svg "access 1 year"
ExpiresByType text/css "access 1 month"
ExpiresByType application/pdf "access 1 month"
ExpiresByType application/javascript "access 1 month"
ExpiresByType application/x-javascript "access 1 month"
ExpiresByType application/x-shockwave-flash "access 1 month"
ExpiresByType image/x-icon "access 1 year"
ExpiresDefault "access 2 days"
</IfModule>
## EXPIRES HEADER CACHING ##
```

HTTPキャッシュヘッダーを追加できるのはサーバー上のリソースのみですのでご注意ください。サードパーティのリクエストでブラウザのキャッシュを利用する必要があるかもしれないという旨の警告が表示された場合には、そのサードパーティのサーバーのリクエストにアクセスできないため、どうしようもありません。この現象はGoogle Analyticsのスクリプトと、FacebookやTwitterなどのマーケティング 픽セルでよく発生します。

Google Analyticsのスクリプトの場合には、Perfmatters(perfmatters.ioで入手可能)及びWP Rocket(wp-rocket.meで入手可能)などのプラグインを使用してローカル、あるいはご利用のCDNにホストするという処理方法もありますが、これは公式にサポートされていません。

Last-ModifiedヘッダーとETagヘッダーを追加する (キャッシュ検証)

次に、last-modifiedとetagの2つのヘッダーのセットも必要になります。

cache-controlとexpiresヘッダーは、あるファイルが前回のリクエスト以降に**変更されているか**をブラウザーが判断するためのもの(つまり、キャッシュを検証するもの)です。last-modifiedとetagヘッダーは**キャッシュを検証し、その長さを指定するため**、すべてのオリジンサーバーの応答に含める必要があります。これらが正しく設定されていない場合には、「キャッシュバリデータを指定する」必要ありという旨の警告が表示されることがあります。

Name	× Headers Preview Response Timing
☐ editwp.com	▼ Response Headers
☐ css?family=Libre+Franklin%3A300%2C300i%...	accept-ranges: bytes
☒ header.jpg	access-control-allow-origin: *
☐ global.js	cache-control: max-age=31556940
☐ jquery.scrollTo.js	content-length: 114854
☐ wp-embed.min.js	content-type: image/jpeg
☐ style.css	date: Mon, 19 Nov 2018 03:14:55 GMT
☐ jquery.js	etag: "5bebbd30-1c0a6" ←
☐ skip-link-focus-fix.js	expires: Tue, 19 Nov 2019 09:03:55 GMT
☐ wp-emoji-release.min.js	last-modified: Wed, 14 Nov 2018 06:14:08 GMT ←

ヘッダーがない場合は、毎回リソースに対する新しいリクエストが生成されるため、サーバーの負荷が高くなります。キャッシュヘッダを利用することにより、2回目以降のリクエストがサーバーから読み込まれなくなり、帯域幅が節約され、パフォーマンスが向上します。

Kinstaでは、すべてのサーバーリクエストに自動的に上記のキャッシュヘッダーを追加します。CDNをご利用の方も、これらのヘッダーがCDNプロバイダーにより追加されるはずです。cache-controlとexpiresと同様に、これらのHTTPヘッダーを外部リソースに手動で設定することはできません。

Last-Modifiedヘッダー

last-modifiedヘッダーは原則、サーバーから自動的に送信されます。**通常は手動で追加する必要はありません**。ブラウザのキャッシュ内のファイルが前回のリクエスト以降に変更されているかどうかを確認するために送信されるものです。Pingdomのヘッダーリクエストまたは、ChromeのDevToolsを使用することによりLast-Modifiedヘッダーの値を確認できます。

ETagヘッダー

ETagヘッダーはlast-modifiedヘッダーと非常によく似ています。これもファイルのキャッシュを検証するためのものです。Apache 2.4以降をご利用の場合には、ETagヘッダーはFileETagディレクティブを使用して自動的に追加されます。またNGINXに関しては、ETagヘッダーが2016年以来デフォルトで有効になっています。

次のコードを使用することにより、NginxでETagヘッダーを手動にて有効にすることができます。

```
etag on
```

vary: Accept-Encodingヘッダーを追加する

クライアントがコンテンツの圧縮版を処理できるかどうかをブラウザに伝える為、すべてのオリジンサーバーの応答にvary: Accept-Encodingヘッダーを含める必要があります。これが正しく設定されていない場合には、「キャッシュバリデータを指定する」必要があるという旨の警告が表示されることがあります。

たとえば、gzip圧縮のない古いブラウザとgzip圧縮のある新しいブラウザがあるとします。vary: Accept-Encodingヘッダーを利用していない場合に、Webサーバー及びCDNが非圧縮版をキャッシュし、誤って新しい方のブラウザに配信することがあり、WordPressウェブサイトのパフォーマンスが低下します。本ヘッダーを使用することにより、Webサーバー及びCDNによる適切なバージョンの配信を保証できます。

Name	× Headers Preview Response Timing
 editwp.com	▼ Response Headers content-encoding: gzip content-type: text/html; charset=UTF-8 date: Mon, 19 Nov 2018 03:14:57 GMT server: nginx status: 200 vary: Accept-Encoding ← x-content-type-options: nosniff x-kinsta-cache: HIT
 css?family=Libre+Franklin%3A300%2C300i%...	
 header.jpg	
 global.js	
 jquery.scrollTo.js	
 wp-embed.min.js	
 style.css	
 jquery.js	
 skip-link-focus-fix.js	

Kinstaは、すべてのサーバーリクエストに自動的に上記のキャッシュヘッダーを追加します。CDNをご利用の方も、このヘッダーがCDNプロバイダーにより追加されるはずで、上記で説明した他のキャッシュヘッダーと同様に、本ヘッダーも外部リソースに手動で設定することはできません。

ApacheでVary: Accept-Encodingヘッダーを追加する

Apacheでvary: Accept-Encodingヘッダーを追加するには、.htaccessファイルに次のコードを追加します。

```
<IfModule mod_headers.c>
    <FilesMatch ".(js|css|xml|gz|html)$">
        Header append Vary: Accept-Encoding
    </FilesMatch>
</IfModule>
```

NginxでVary: Accept-Encodingヘッダーを追加する

Nginxでvary: Accept-Encodingヘッダーを追加するには、設定ファイルに次のコードを追加します。Nginxのすべての設定ファイルは/etc/nginx/ディレクトリにあります。プライマリー設定ファイルは/etc/nginx/nginx.confです。

```
gzip_vary on
```

wp-config.phpでWordPressのメモリ上限を変更する

WordPress Codexで説明されているように、Version 2.5から、WP_MEMORY_LIMITオプションを使用してPHPが消費するメモリの最大値を設定できます。メッセージ「Allowed memory size of xxxxxx bytes exhausted」などが表示される場合に必要となる設定です。

デフォルトでは、WordPressはPHPのメモリを40MB(マルチサイトでは64MB)まで増加します。/
wp-includes/default-constants.phpの32~44行にこのコードがあります。

なお、ホスティング会社によるサーバー上のPHPmemory_limitもあります。この2つは異なります。Kinstaのmemory_limitのデフォルト設定は256Mです。メモリ不足エラーが発生した場合、WordPressのPHPが使用するメモリの上限を上げることができます。

wp-config.phpファイルの「That's all, stop editing! Happy blogging.」行の直前に次のコードを追加します。

```
define( 'WP_MEMORY_LIMIT', '256M' );
```

WordPressのメモリ上限の課題についてはsaotn.orgのJan Reilink氏による素晴らしい記事も是非ご確認ください。彼はコードのもう一つの使い方について紹介しています。手動で設定するのではなく、PHPのmemory_limitの値に設定することができます。

```
define( 'WP_MEMORY_LIMIT', ini_get( 'memory_limit' ) );
```

第15章:

フロントエンド最適化と外部 サービスに関する推奨事項



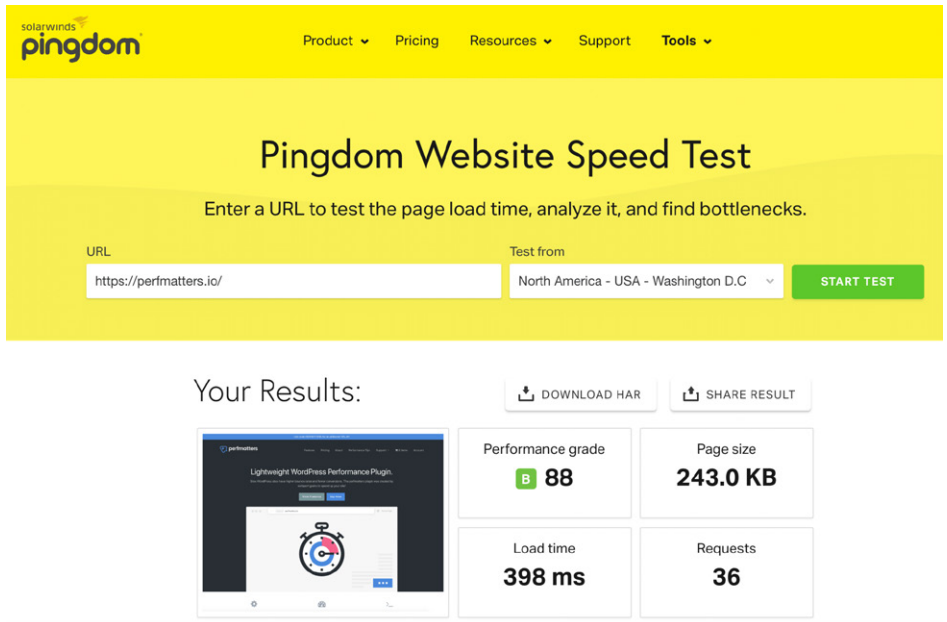
続いて、フロントエンドを最適化することによりWordPressをスピードアップする方法についてご説明します。フロントエンドとは、CSS、JavaScript、画像など、完全にクライアント側のブラウザで処理されるものを指します。また、ウェブサイトに読み込まれている外部サービスの分析とそのサービスの合計読み込み時間に与える影響もこれに含まれます。

フロントエンド最適化には、次の2つの重要な目的があります。

- **Webページ全体のサイズを縮小すること。**CSS、JavaScript、画像などのサイズが重要です。4MBのウェブサイトは、1MBのウェブサイトより読み込み速度ははるかに遅くなります。ページの重さが読み込み時間に与える影響と、それだけに集中しない方がいいというトピックについてのPaul Calvano氏による素晴らしい記事も是非ご参照ください。
- **HTTPリクエストと外部サービスの数を減らすこと。**HTTP 2では、単一のTCP接続を使用することにより複数のリクエストと応答を同時に送信できるようになりました。これはパフォーマンスにおける素晴らしい改善ですが、HTTPリクエストの数を減らすことにより、WordPressウェブサイトのさらなるスピードアップが可能です。これには、外部リクエストとサービスの数を減らすことも含まれます。それぞれ、DNSルックアップ、TLS接続、およびネットワーク遅延などさらなる遅延を引き起こすのです。

ベースラインを定義するためにWordPressウェブサイトのスピードテストを実施する

ウェブサイトのフロントエンドを最適化しようとするときは、まずベースラインを定義した方が良いでしょう。そのため、スピードテストを実施しましょう。スピードテストを実施する方法は多数あります。kinsta.comではPingdomの使用法とGTmetrixの使用法についての詳細なガイドをご用意しております。



Pingdom Website Speed Test

Enter a URL to test the page load time, analyze it, and find bottlenecks.

URL: Test from: [START TEST](#)

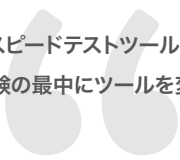
Your Results:

- Performance grade: **B 88**
- Page size: **243.0 KB**
- Load time: **398 ms**
- Requests: **36**

スピードテストの注意事項

1. ツールを一つ選択し、途中で変えないこと

わたしたちのお気に入りにはPingdom、GTmetrix、WebPageTest、PageSpeed Insights、そしてChrome DevToolsです。ただ、ずっと同じものを使用するのであれば、どのスピードテストツールを使用しても構いません。スピードの測定方法と定量化方法は各ツールで異なるため、いずれかを選び、試験や最適化の途中で絶対に変えないようにしましょう。Googleも一つに絞ることを推奨しているようです。



どのスピードテストツールを選択するかどうかよりも、
試験の最中にツールを変えないことが重要です。

2. 満点にこだわらない

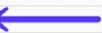
Google PageSpeed Insightsなどのツールには、スピード及びパフォーマンスの点数が出ます。ただし、点数はウェブサイトのスピードやユーザーが感じるパフォーマンスほど重要ではないことを覚えておきましょう。点数は、パフォーマンスを把握する指標に過ぎません。満点の100点や「A」にこだわるのは時間の無駄です。外部のスクリプトや広告の多い大規模なウェブサイトの場合、満点を取ることはできませんが、これはまったく問題ありません。

3. 試験を実施するロケーションも重要

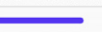
スピードテストの際に選択するロケーションは大きな影響を与えます。前述のとおり、選択するデータセンターの場所は極めて重要です。TTFB、ネットワークレイテンシなどが関連しています。そのため、データセンターに近いロケーション、遠いロケーションの両方からウェブサイトを試験しましょう。そうすることで、CDNのWordPressウェブサイトへの影響も確認できます。

4. キャッシュを考慮し、試験を複数回実施する

上記のキャッシュについてのセクションでご説明したように、WordPressホスティング会社またはCDNのキャッシュが最近クリアされた場合または期限切れになった場合には、HTTPヘッダーは「MISS」になります。つまり、ウェブサイト及びアセットはキャッシュから配信されません。

FILE	SIZE	0.0s	0.1s	0.2s
 https://perfmatters.io/	9.6 KB			
https://perfmatters.io/				
Response Headers 200				
status	200			
x-content-type-options	nosniff			
content-encoding	gzip			
set-cookie	PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1...			
x-kinsta-cache	MISS 			
expires	Thu, 19 Nov 1981 08:52:00 GMT			

ウェブサイト全体のスピードを正しく把握するには、すべてのデータがキャッシュから読み込まれ、初期ページおよびすべてのアセットが「HIT」である必要があります。このため、スピードテストを複数回実行することもあるでしょう。その場合には平均をとってください。

FILE	SIZE	0.0s	0.1s	0.2s
 https://perfmatters.io/	9.6 KB			
https://perfmatters.io/				
Response Headers 200				
status	200			
x-content-type-options	nosniff			
content-encoding	gzip			
set-cookie	PHPSESSID=d33c7f3bfa8f6c0a9ba55f3a6804e1...			
x-kinsta-cache	HIT 			
expires	Thu, 19 Nov 1981 08:52:00 GMT			

続いては、WordPressサイトでできるフロントエンドの最適化についてご説明します。

「クエリ文字列を削除せよ」

「クエリ文字列を削除せよ」警告もスピードテストツールでよく発生する推奨事項の一つです。その内容は何でしょうか？CSSファイルとJavaScriptファイルのURLの末尾には、通常<https://domain.com/file.min.css?ver=4.5.3>などのファイルバージョンがあります。ただし、クエリ文字

列をキャッシングできないサーバーとプロキシサーバーがありますので、削除することによりキャッシングが改善される場合があります。

テーマのfunctions.phpファイルに次のコードを手動で追加します。

```
function remove_query_strings() {
    if(!is_admin()) {
        add_filter('script_loader_src', 'remove_query_strings_split', 15);
        add_filter('style_loader_src', 'remove_query_strings_split', 15);
    }
}

function remove_query_strings_split($src){
    $output = preg_split("/(&ver|\?ver)/", $src);
    return $output[0];
}

add_action('init', 'remove_query_strings');
```

または、Perfmatters(perfmatters.ioで入手可能)などのプレミアムプラグインを使用して、クエリ文字列をワンクリックで削除できます。

クエリ文字列あり

Name	Status
 editwp.com	200
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600%2C600i...	200
 style.css?ver=4.9.8	200
 jquery.js?ver=1.12.4	200
 header.jpg	200
 skip-link-focus-fix.js?ver=1.0	200
 global.js?ver=1.0	200
 jquery.scrollTo.js?ver=2.1.2	200
 jizDREVIthGc8qDIbSTKq4XkRiUf2zcZiVbJ.woff2	200

クエリ文字列を読み込むスクリプトの例は下記のとおりです。

クエリ文字列なし(コード実行後)

Name	Status	
 editwp.com	200	
 css?family=Libre+Franklin%3A300%2C300%2C400%2C400...C600%2C600i...	200	
 style.css ←	200	
 jquery.js	200	
 header.jpg	200	
 skip-link-focus-fix.js	200	
 global.js	200	クエリ文字列が削除
 jquery.scrollTo.js	200	されたスクリプトの例
 jizDREVIthGc8qDIbSTKq4XkRiUf2zcZiVbJ.woff2	200	は下記のとおりです。

ただし、ウェブサイトのクエリ文字列を削除する前に、まずクエリ文字列が使用されている理由を理解することが重要です。ファイルのバージョン管理は、WordPressの開発者がキャッシュの問題を回避するための方法の一つです。

たとえば、更新の際にstyle.cssを?ver=4.6から?ver=4.7に変更すると、それはまったく新しいURLとして認識され、キャッシュされません。クエリ文字列を削除してプラグインを更新すると、キャッシュされたバージョンが引き続き配信される場合があります。場合により、キャッシュされたリソースの期限が切れるか、キャッシュが完全にフラッシュされるまで、ウェブサイトの外観が壊れる可能性があります。

また、クエリ文字列をキャッシュできるCDNもあります。Kinsta CDNは、キャッシュ可能(デフォルトでこれを行う設定)となっています。つまり、Kinstaのお客様なら、クエリ文字列は既にキャッシュされているということです。

レンダリングをブロックしているJavaScriptとCSSを排除する

ページの素早い読み込みを邪魔するファイルがある場合には、レンダリングをブロックしているJavaScriptとCSSについての警告が表示されることがあります。重要なコンテンツの表示に不要な条件付きのJSとCSSもあり、これらがレンダリングをブロックしないようにasync属性とdefer属性を使用します。

Opportunity	Estimated Savings
1 Eliminate render-blocking resources	2.24 s ^
Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. Learn more .	

レンダリングをブロックしているJavaScriptとCSSを排除するには次に従ってください。

クリティカルレンダリングパスからJSを削除する

JavaScriptをクリティカルレンダリングパスから削除するには、JavaScriptリソースを呼び出すためにスクリプトのHTML要素にdefer属性またはasync属性を追加します。

- **async属性は**、HTMLをパースせずにすぐにリソースのダウンロードを開始するようブラウザに指示します。リソースが利用可能になると、HTMLパースが一時停止され、リソースが読み込まれます。
- **defer属性は**、HTMLパースが完了するまでリソースのダウンロードを延期するようブラウザに指示します。ブラウザがHTMLの方を完了した後に、延期されたすべてのスクリプトを、文章に表示されている順序でダウンロードしてレンダリングします。

CSSリソースの配信を最適化する

CSSの配信を最適化するには、レンダリングをブロックしないための方法を考える必要があります。

- 重要なコンテンツをレンダリングするために必要なスタイルを識別し、HTMLとインラインで配信
- 必要な場合にのみ、CSSを条件付きにてデバイスで使用
- 残りのCSSを非同期で読み込む

上記をすべて実施するのは難しく、ウェブサイトを読み込まれるスクリプトにより調整が間違いなく必要です。下記はこれを実施する上で便利なWordPressプラグインです。

- Autoptimize (無料で、WordPress.orgで入手可能)
- Async JavaScript (無料で、WordPress.orgで入手可能)
- Hummingbird (無料で、WordPress.orgで入手可能)

詳細については、レンダリングをブロックしているJavaScriptやCSSを排除することについての当社の記事をご参照ください。

WordPressで外部CSSとJavaScriptを結合する

外部CSS結合についての警告が表示されるのは、cdn.domain.comなどの外部ドメインでCSSファイルをホストするためにCDNを使用しているときに発生します。これまでの処理方法は、CSSファイルを連結または結合し、単一のリクエストで読み込まれるようにすることでした。

HTTP/2をサポートするプロバイダでHTTPSを利用する場合には、この警告は以前ほど重要ではありません。HTTP/2では、単一の接続を介して複数のCSSファイルを同時に読み込むことができ、ブラウザの86%以上がHTTP/2をサポートしています。

しかし、必ずしもこの最適化対策が不要であるとも言えません。これによりWordPressウェブサイトをさらにスピードアップできるケースも見たことがあります。ファイルのサイズと数によるため、必ず自分のウェブサイトで試験してみることをお勧めします。

外部のCSSファイルとJavaScriptファイルを結合する上で便利なのが、無料のAutooptimizeプラグインの使用です。(WordPress.orgで入手可能) 結合後に、「autooptimize_xxxx.css」または「autooptimize_xxxx.js」ファイルができます。CDNからの読み込みもサポートしています。無料のWP Rocketプラグインも利用できます。(wp-rocket.meで入手可能)

Name	Status	Domain
 editwp.com	200	editwp.com
 autooptimize_23d90d8a1eaacef4363d9e02c50ae363.css ←	200	mk0editwpn6...
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...C600...	200	fonts.googlea...
 jquery.js?ver=1.12.4	200	mk0editwpn6...
 header.jpg	200	mk0editwpn6...
 autooptimize_3fde0f2c0341cdc5afd7cc0c0e403f92.js ←	200	mk0editwpn6...
 jizDREVItHgc8qDIbSTKq4XkRiUf2zcZiVbJ.woff2	200	fonts.gstatic.c...
 jizAREVItHgc8qDIbSTKq4XkRi24_Sl0q1vjitOh.woff2	200	fonts.gstatic.c...
 jizAREVItHgc8qDIbSTKq4XkRi20-Sl0q1vjitOh.woff2	200	fonts.gstatic.c...

HTML、CSS、JavaScriptの縮小を実施する

HTML、CSS、およびJavaScriptのリソースの縮小により、ブラウザがダウンロードするデータの量を減らすことができます。縮小は言うなれば、コメントや空白などの不要な文字をソースコードから削除するプロセスです。これらの文字は開発において非常に便利であるものの、ブラウザがページをレンダリングする際には完全に不要です。

縮小されていないHTML

これは縮小されていないHTMLコードの例です。

```
1 <!DOCTYPE html>
2 <html lang="en-US" class="no-js no-svg">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="profile" href="http://gmpg.org/xfn/11">
7
8 <script>(function(html){html.className = html.className.replace(/\bno-
9 js\b/, 'js')})(document.documentElement);</script>
10 <title>editwp &#8211; Just another WordPress site</title>
11 <link rel='dns-prefetch' href='//fonts.googleapis.com' />
```

縮小されているHTML

こちらは縮小されたHTMLコードの例です。

```
1 <!DOCTYPE html><html lang="en-US" class="no-js no-svg"><head><meta
2 charset="UTF-8"><meta name="viewport" content="width=device-width,
3 initial-scale=1"><link rel="profile" href="http://gmpg.org/xfn/11">
4 <script>(function(html){html.className = html.className.replace(/\bno-
5 js\b/, 'js')})(document.documentElement);</script> <link type="text/css"
6 media="all" href="https://mk0editwpn6qbfuw5o9.kinstacdn.com/wp-
7 content/cache/autoptimize/css/autoptimize_499850c34eb7b4e7cbd9d8b41b2a5
8 7d6.css" rel="stylesheet" /><style type="text/css"
9 media="print">#wpadminbar{display:none}</style><style type="text/css"
10 media="screen">html{margin-top:32px !important}* html body{margin-
```

無料のAutoptimizeプラグインまたはWP Rocketを使用することによりファイルを簡単に縮小できます。

クッキーフリーのドメインを使用する

原則として、画像、JavaScript、CSSなどのコンテンツを配信する際に、オーバーヘッドが生じるため、HTTPクッキーを付ける理由は一切ありません。サーバーがあるドメインに対しクッキーを設定すると、そのドメインに対するすべてのHTTPリクエストにそのクッキーが含まれることになります。この警告は、リクエスト数の多いウェブサイトによく見られます。

HTTP/2などの新しいプロトコルではこの警告は以前ほど重大ではないため、無視してもかまいません。新しい接続のコストは、すべてのデータを同じ接続を介しストリーミングするよりも高くなります。

この警告に対する簡単な対処法はこちら。クッキーを無視し、クライアントによるSet-Cookie応答ヘッダーの受信を阻害してしまうクッキーを削除できるCDNプロバイダを使用してみましょう。KeyCDNは、そんな機能を提供するCDNプロバイダの一つです。デフォルトでは次の2つのオプションが有効になっています。これは、ウェブサイトを移行したり、別のサブドメインから静的アセットを配信するよう設定したり…といった手間なく行える簡単な方法です。

Cache Cookies *

enabled

By default, files with cookies are not cacheable. However, enabling this option will ignore the presence of cookies and therefore force the edge servers to cache these files.

Strip Cookies *

enabled

This feature strips the cookies received from the origin server. The client will not receive the `Set-Cookie` response header. **It is highly recommended to enable this directive if you enable Cache Cookies.**

Cloudflareをご利用の場合には、Cloudflareのネットワークを介して配信されているリソースのクッキーを無効にすることはできません。CloudFlareは独自のセキュリティクッキーをヘッダーに含みます。繰り返しますが、このクッキーは非常に小さく、パフォーマンスへの影響はごくわずかです。ただし、CloudFlareを利用する際には、この警告を回避する方法はありません。

2番目の方法として、新しいドメインまたはサブドメインから静的アセットを配信するようにWordPressを再設定することもできます。

WordPressの埋め込み機能を無効にする

WordPress 4.4がリリースされた際、oEmbed機能がコアに含まれました。これにより、URLを貼り付けるだけでYouTubeの動画、ツイートなどのリソースをウェブサイトに埋め込むことが可能になりました。URLは自動的に埋め込みとして変換され、ビジュアルエディタでライブプレビューも表示されます。このアップデートにより、WordPress自体がoEmbedプロバイダーになりました。

この機能は多くの人にとって便利で、有効のままにしても構いません。ただし、この機能は、WordPressウェブサイトでwp-embed.min.jsファイルを読み込むための追加のHTTPリクエストも生成します。そしてこれはウェブサイト全体に負荷をかけます。このファイルは1.7 KBしかありませんが、このようなファイルが時間が経つにつれて重なります。リクエスト自体がコンテンツのダウンロードサイズよりも大きいことも。

Name	Status	Domain
 editwp.com	200	editwp.com
 global.js	200	mk0editwpn...
 css?family=Libre+Franklin%3A300%2C300i%2C400%2C400...	200	fonts.google...
 jquery.scrollTo.js	200	mk0editwpn...
 wp-embed.min.js ←	200	mk0editwpn...
 style.css	200	mk0editwpn...
 jquery.js	200	mk0editwpn...
 header.jpg	200	mk0editwpn...
 skip-link-focus-fix.js	200	mk0editwpn...
 wp-emoji-release.min.js	200	editwp.com

WordPressの埋め込み機能を無効にする方法はいくつかあります。次のプラグインをお勧めします。

- Disable Embeds (無料で、WordPress.orgで入手可能)
- Perfmatters(perfmatters.ioで入手可能)

コードでも埋め込み機能を無効にすることができます。非常に長いコードである為、当社のウェブサイト<https://kinsta.com/knowledgebase/disable-embeds-wordpress/>からどうぞ。

WordPressの絵文字を無効にする

埋め込みと同様に、WordPress 4.2では、古いブラウザを対象に絵文字のサポートがコアに追加されました。ただし、この機能は、WordPressサイトでwp-emoji-release.min.jsファイルを読み込むための追加のHTTPリクエストも生成します。ウェブサイト全体にとっての負荷です。

このファイルは10.5 KBしかありませんが、ウェブサイトで絵文字を使用していないなら完全に無駄です。

Name	Status	Domain
 editwp.com	200	editwp.com
 global.js	200	mk0editwpn...
 css?family=Libre+Franklin%3A300%2C300%2C400%2C400...	200	fonts.google...
 jquery.scrollTo.js	200	mk0editwpn...
 wp-embed.min.js	200	mk0editwpn...
 style.css	200	mk0editwpn...
 jquery.js	200	mk0editwpn...
 header.jpg	200	mk0editwpn...
 skip-link-focus-fix.js	200	mk0editwpn...
 wp-emoji-release.min.js ←	200	editwp.com
 jizDREvIHgc8qDlbSTKq4XkRiUf2zcZiVbJ.woff2	200	fonts.gstatic....

WordPressで絵文字を無効にする方法はいくつかあります。次のプラグインをお勧めします。

- Disable Emojis (無料で、WordPress.orgで入手可能)
- Perfmatters (perfmatters.ioで入手可能)

コードでも絵文字機能を無効にすることができます。非常に長いコードである為、当社のウェブサイト<https://kinsta.com/knowledgebase/disable-embeds-wordpress/>からご覧ください。

WordPressのコメント機能のスピードアップ及び無効化

ウェブサイトのコメントセクションが賑わうと、パフォーマンスの問題が発生することがあります。コメントを機能させるためのリソースを考慮しましょう。

- 既存のコメントを表示するためにデータベースのクエリが実行されます。
- 新しいコメントごとにデータベースエントリが作成されます。
- コメントとコメントのメタデータは、訪問者のブラウザに受信かつ処理されます。
- Gravatarなどの外部リソースがリクエスト、ダウンロード、または読み込まれます (個別のDNSルックアップが必要です)。

- コメントシステムが想定どおりに機能するには、重いJavaScriptおよびjQueryリソースをダウンロードして処理する必要がある場合も多く見られます。

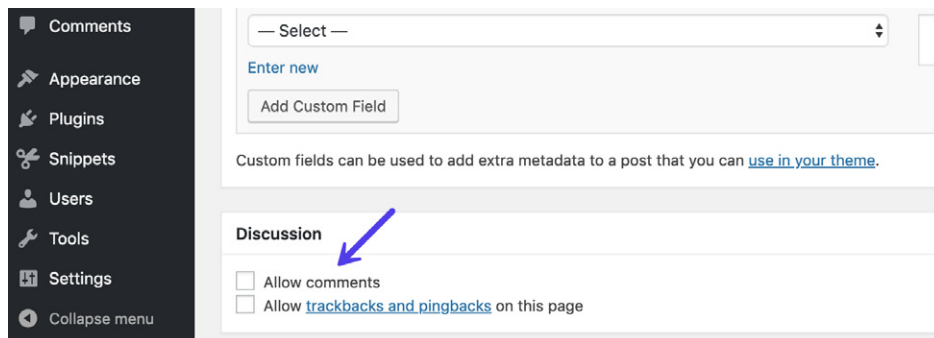
WordPressのコメントをスピードアップできるオプションは4つあります。

オプション1:コメント機能を無効にする

ウェブサイトのコメントの数が少なく、あまり価値を加えていない状態であればコメント機能を完全に無効にした方が良いでしょう。Googleはコメントもページのコンテンツとしてクロールするため、コメントはSEOに影響を与える場合がありますので、その点ご注意ください。質の高いコメントのみを承認してください。コメント機能を無効にする次の3つの方法をご参照ください。

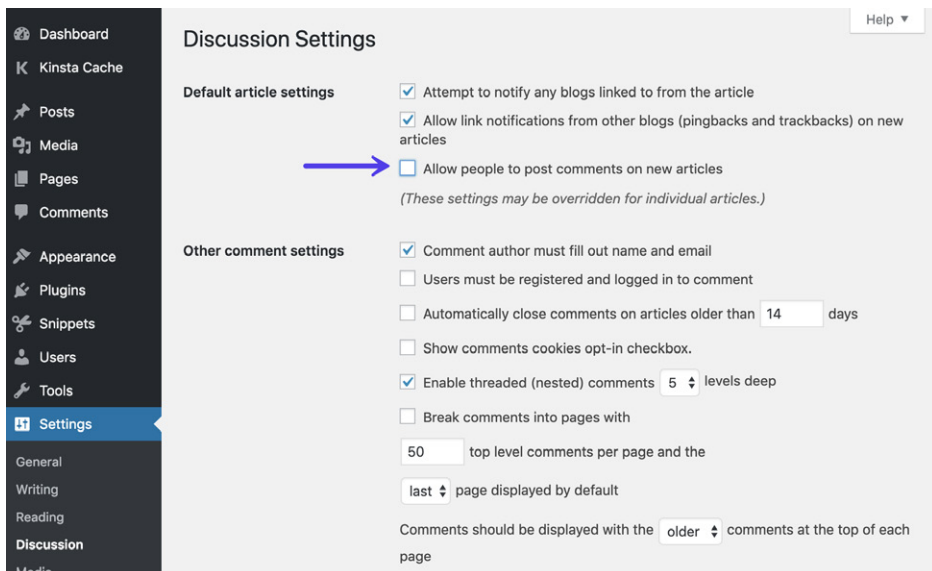
1. WordPressの「オプション」でコメント機能を無効にする

特定の固定ページや投稿のコメント機能を個別に無効にするには、編集画面にアクセスして「ディスカッション」セクションの「コメントを許可」をオフにします。「ディスカッション」セクションが表示されない場合は、スクロールアップして「表示オプション」で有効化してください。



コメント欄を完全に非表示することもできます。これを行うには、WordPressダッシュボードの「設定」メニューの「ディスカッション」をクリックし、「新しい投稿へのコメントを許可する」オプションをオフにします。ただし、この方法のデメリットは古い記事への影響がなく、新しい記事にのみ影響することです。

より細かい設定を希望する方、またすべてのコメントを非表示にしたい方には、以下のいずれかの方法をお勧めします。



2. プラグインでコメント機能を無効にする

コメントを無効化する最も簡単な方法の一つは、Disable Comments (WordPress.orgで入手可能)などの無料のプラグインを使用することです。すべてのコメントをワンクリックで無効にするか、投稿・固定ページ・メディア投稿タイプごとに無効にすることができます。

3. コードでコメント機能を無効にする

また、コードを使用してもコメントを無効にすることができます。単純なHTMLのコメントでWordPressテーマ内のコメントセクションを非表示にするだけです。

page.phpファイルの次のコードを

```
<?php comments_template( '', true ); ?>
```

下記のとおり置き換えるだけです。


```
<!-- Begin Comment
<?php comments_template( '', true ); ?>
End Comment -->
```

これでコメント欄が固定ページから削除されます。ブログの投稿からも削除するには、single.php ファイルにアクセスし、次のコードを

```
<?php comments_template(); ?>
```

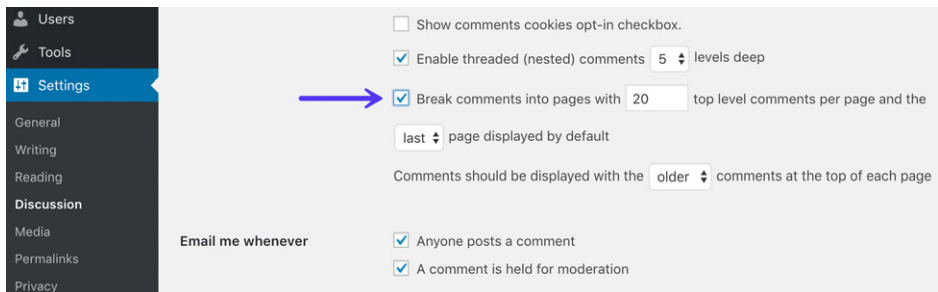
下記のとおり置き換えるだけです。

```
<!-- Begin Comment
<?php comments_template(); ?>
End Comment -->
```

オプション2:WordPress独自のコメント機能を最適化する

2番目のオプションは、WordPress独自のコメント機能を最適化することです。例えば、最初のページの読み込みで読み込まれるコメントの数を減らすことが考えられます。

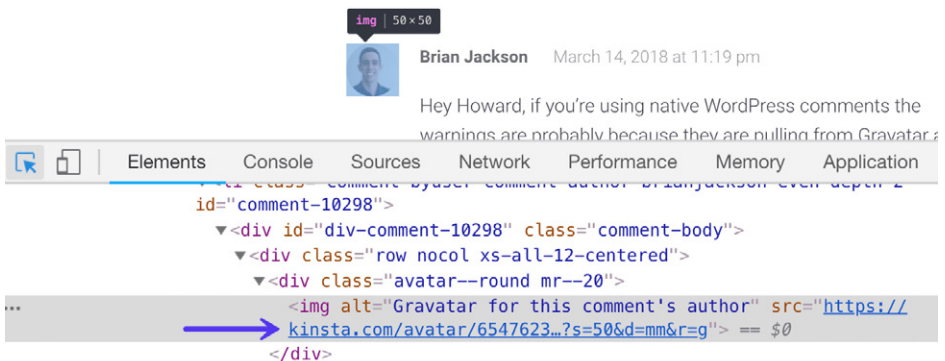
1. WordPress管理ダッシュボードで「設定」→「ディスカッション」の順にアクセスします。
2. 「他のコメント設定」を探します。
3. 「1ページあたり〇〇件のコメントを含む複数ページに分割」のチェックボックスを選択し、最初のページの読み込みで表示したいコメントの数を入力します。



CDNでGravatarをホストする方法もあります。これはKinstaのアプローチです。

デフォルトでは、WordPressのコメントが読み込まれる際に、各Gravatarが別途のHTTPリクエストを必要とします。そのため、ページに50人のコメントがある場合には、その全員のGravatarをダウンロードするには50件のHTTPリクエストが必要になります。ご想像のとおり、これはページのスピードにかなりの影響を与えます。それに、gravatar.comへの外部DNSルックアップが遅くなることがあり、場合によってはタイムアウトになることも。

KinstaのブログのGravatarを確認すると、Kinsta.comから読み込まれていることがわかります。(当社のCDNを含む)



オプション3: サードパーティのコメントシステムを使用する

3つ目のオプションは、サードパーティのコメントシステムを使用することです。ウェブサイトが安価でリソース不足の共有サーバーでホストされている場合は、サードパーティのコメントシステムを使用することにより、コメントの多いページの読み込み速度が改善されます。画像の最適化と同様に、負担をオフロードできます。一方、Kinstaなどの高品質のウェブホスティング会社をご利用の場合には、サードパーティに切り替えても改善が見られるどころか、ウェブサイトの読み込み速度が低下することがあります。

WordPressでRSSフィードを無効にするには、次の2つの方法があります。

1. プラグインを使用してRSSフィードを無効にする

WordPressのRSSフィードを無効化する最初の方法は、Disable Feeds (WordPress.orgで入手可能)などの無料のプラグインを使用することです。このプラグインは、すべてのリクエストをリダイレクトすることにより、WordPressウェブサイトのすべてのRSSやAtomフィードを無効にします。

また、Perfmatters (Kinstaのチームメンバーにより開発。perfmatters.ioで入手可能)などのプレミアムプラグインを使用して、RSSフィードもそのリンクも無効にすることができます。

2. コードを使用してRSSフィードを無効にする

プラグインを使用したくない方は、コードでもRSSフィードを無効化できます。次のコードをご利用のテーマのfunctions.phpファイルに追加してください。

```
function itsme_disable_feed() {  
    wp_die( __( 'No feed available, please visit the <a href="'. esc_url(  
home_url( '/' ) ) .'">homepage</a>!' ) );  
}  
  
add_action( 'do_feed', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_rdf', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_rss', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_rss2', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_atom', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_rss2_comments', 'itsme_disable_feed', 1 );  
add_action( 'do_feed_atom_comments', 'itsme_disable_feed', 1 );
```

WordPressはまた、ウェブページのヘッダーにRSSフィードへのリンクも生成します。さらにもう一步踏み込むと、固定ページのHTMLコードからこれらのリンクを削除できます。次のコードをご利用のテーマのfunctions.phpファイルに追加します。

```
remove_action( 'wp_head', 'feed_links_extra', 3 );  
remove_action( 'wp_head', 'feed_links', 2 );
```

プリフェッチとプリコネクトを使用する

プリフェッチ、プリコネクトなどのリソースヒント及びディレクティブは舞台裏でWordPressをスピードアップする素晴らしい手段です。

プリフェッチ

DNSプリフェッチを使用すると、ユーザーがリンクをクリックする前にドメイン名を解決（バックグラウンドでDNSルックアップを実行）することにより、パフォーマンスが改善されます。使用するには、WordPressウェブサイトのヘッダーに`rel="dns-prefetch"`タグを追加します。

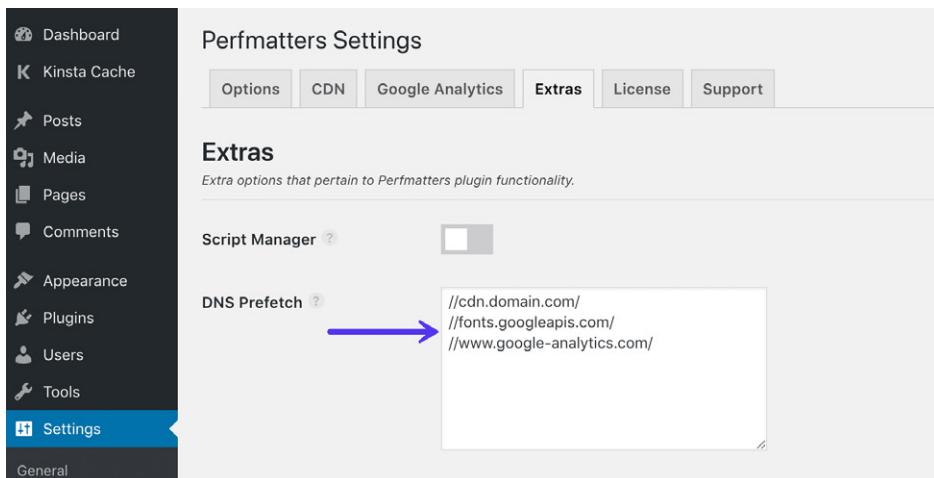
```
<link rel="dns-prefetch" href="//domain.com">
```

DNSプリフェッチを使用する一般的な目的は、CDN URL、Google フォント、Google Analytics などの併用です。

```
<link rel="dns-prefetch" href="//cdn.domain.com/">
<link rel="dns-prefetch" href="//fonts.googleapis.com/">
<link rel="dns-prefetch" href="//www.google-analytics.com">
```

プリフェッチは、現代のブラウザのほとんどでサポートされています。実施方法は、以下のようにいくつかあります。

1. Head, Footer and Post Injections (WordPress.orgで入手可能)などの無料のプラグインを使用して、ヘッダーにコードを追加できます。
2. または、テーマの`header.php`ファイルに次のコードを手動で追加することもできます。
3. または、Perfmatters (perfmatters.ioで入手可能)などのプレミアムプラグインを使用してもDNSプリフェッチを採用できます。Perfmattersプラグインの「Extras」タブをクリックし、ドメインを追加するだけです。フォーマット：`//domain.tld`（行ごとに1つ）



プリコネクト

プリコネクトを使用すると、ブラウザはHTTPリクエストの前に早い段階で接続ができるため、ラウンドトリップレイテンシがなくなり、ユーザーの時間を節約できます。

プリコネクトは最適化ツールボックスの重要なツールです。リクエストパスから費用のかかるラウンドトリップの多くを排除し、時には、リクエストの待ち時間を数百〜数千ミリ秒も減らすことができます。– Ilya Grigorik



Ilya Grigorik

使用するには、WordPressウェブサイトのヘッダーに`rel="preconnect"`タグを追加します。

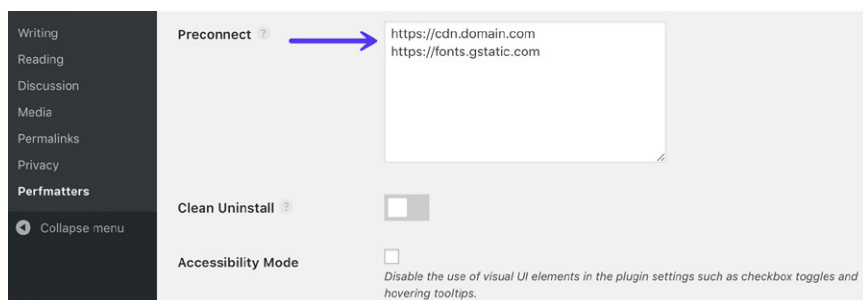
```
<link rel="preconnect" href="//domain.com">
```

想定される目的は例えば、CDN、URLまたはGoogleフォントなどです。

```
<link rel="preconnect" href="https://cdn.domain.com">
<link rel="preconnect" href="https://fonts.gstatic.com">
```

プリコネクトは、Internet Explorer、Safari、IOS SafariとOpera Miniを除き、ほとんどの現代のブラウザで利用できます。やり方は以下の通りいくつかあります。

1. Head, Footer and Post Injections (WordPress.orgで入手可能)などの無料のプラグインを使用して、ヘッダーにコードを追加できます。
2. または、テーマのheader.phpファイルに次のコードを手動で追加することもできます。
3. または、Perfmatters(perfmatters.ioで入手可能)などのプレミアムプラグインを使用してもプリコネクトを採用できます。Perfmattersプラグインの「Extras」タブをクリックし、ドメインを追加するだけです。フォーマット:scheme://domain.tld(行ごとに1つ)



固定ページ及び投稿ごとにスクリプトを無効にする

WordPressをスピードアップするためのもう一つの非常に強力な手段として、各固定ページ及び投稿に読み込まれている各リクエストを確認することができます。ウェブサイト全体で読み込まれているものの、本当はそうするべきではない…そんなスクリプトがある可能性は高いでしょう。

Perfmatters(perfmatters.ioで入手可能)などの「Script Manager」(スクリプトマネージャ)機能を内蔵したプレミアムプラグインを使用することがひとつの手です。これにより、固定ページ及び投稿ごと

に、また、ウェブサイト全体を対象にしてワンクリックでスクリプト (CSSおよびJavaScript) を無効にすることができます。繰り返しますが、このプラグインはKinstaスタッフの一人により開発されました。

使用用途の例をいくつかご紹介します。

- 人気のContact Form 7プラグインは、すべての固定ページと投稿に読み込まれます。そこで、ウェブサイト全体を対象に無効にし、お問い合わせのページのみで有効にすることができます。
- ソーシャルメディアのシェアプラグインは投稿でのみ読み込まれるべきです。そこで、ウェブサイト全体を対象に無効にし、カスタムとして特定の投稿の種類のみを対象に有効にすることができます。
- Table of contents plugin (TOC)プラグインは、すべての固定ページと投稿で読み込まれます。スクリプトマネージャを使用すると、読み込まれる場所を指定できます。
- WordPress 5.0にアップグレードしつつも、Gutenbergブロックエディタではなく、旧型エディタまたはサードパーティ製のエディタを使用している、という場合には、`/wp-includes/css/dist/block-library/style.min.css`と`/wp-includes/css/dist/block-library/theme.min.css`という2つのウェブサイト全体に追加されているフロントエンドスクリプトを無効化した方がいいでしょう。

でも、なんで、こんなコードのプラグインが存在するの？

そもそも、ページ上でプラグインが検出されたときに限りそのスクリプトが読み込まれるようにしないプラグインの開発者は…いったい何を考えているのでしょうか？。答えは複雑です。たとえば、Contact Form 7のようなプラグインは、どこにでもプラグインを配置できるショートコードを備えています。これにはウィジェットへのドロップも含まれています。WordPressでは、スクリプトをデキューする時、スクリプトからのデータのクエリがはるかに難しく (投稿、固定ページのメタデータからデータをクエリする場合に比べて) なります。

したがって、ユーザビリティが理由となる場合が多く見られます。プラグインが壊れる可能性が少ないほど、サポートチケットの数が少なくなります。一方、この課題を超えて、パフォーマンスを念頭に置いてコーディングする方法は実はあります。残念なことに、ダウンロードとユーザーが圧倒的に多いため、ユーザビリティを優先してコーディングする場合が大多数です。

Script Managerの紹介

Script Managerについて少しご説明します。ツールバーでスクリプトマネージャをクリックすると、そのURLに読み込まれているすべてのスクリプト（JavaScriptファイルとCSSファイルの両方）が表示されます。ここでの、オプションは二つあります。

1. **ステータスがオン**：（デフォルト設定）
2. **ステータスがオフ**：ウェブサイト全体を対象にして無効にする（その後、現在のURLを含め、どの投稿の種類を対象に有効にするかが指定できます）
3. **ステータスがオフ**：現在のURLでのみ無効にする（これはホームページで特に便利で
4. **ステータスがオフ**：例外（現在のURL、投稿の種類、またはアーカイブ）

The screenshot shows the 'Script Manager' interface. On the left is a sidebar with 'perfmatters' logo and three menu items: 'Script Manager' (highlighted), 'Global View', and 'Settings'. The main content area is titled 'Script Manager' with the subtitle 'Manage scripts loading on the current page.' Below this is a 'Plugins' section. The first plugin is 'Extended Widget Options', which has a green 'ON' toggle. It contains two scripts: 'jquery-widgetopts' (JS, 6.1 KB) and 'widgetopts-styles' (CSS, 13.6 KB), both with green 'ON' toggles. The second plugin is 'Social Warfare', which has a red 'OFF' toggle. It contains two scripts: 'social_warfare_script' (JS, 10 KB) and 'social_warfare' (CSS, 45.2 KB), both with red 'OFF' toggles. Below each 'OFF' toggle, there are options to 'Disabled' (Everywhere or Current URL) and 'Exceptions' (Current URL, Post Types: Posts, Pages, Lists, Coupons).

中身は**プラグイン名**と**またはテーマ名**とにまとめられています。したがって、プラグイン全体を一括で無効にするのは非常に簡単です。WordPressプラグインは通常、JavaScriptファイルもCSS

ファイルもあります。WordPressテーマにはファイルが10個以上あることも。Script Managerは、複雑なURL構造またはインストールの正規表現もサポートしています。

設定変更を選択した後は、必ず下部にある「保存」ボタンをクリックしてください。次に、ウェブサイトスピードツールで試験し、スクリプトが固定ページや投稿で読み込まれなくなったことを確認しましょう。その前にキャッシュのクリアをお忘れなく。万が一、外観での問題が発生した場合には、設定でスクリプトを再度有効にして通常の状態に戻すことができます。

woorkupによるスピードテストでは、**合計読み込み時間を20.2%短縮できました**。ホームページだけでHTTPリクエストの数が46から30に減少。ページサイズも506.3KBから451.6KBに縮小しました。

サードパーティパフォーマンスを分析する

ウェブサイトから外部への呼び出しのすべてが読み込み時間に影響します。さらに悪いことに、その中には断続的に発生し、異常の原因が特定しにくいものもあります。

サードパーティの外部サービスは、サーバーの外部からWordPressウェブサイトと通信を行います。その典型的な例を以下にご紹介します。

- Twitter、Facebook、Instagramなどのソーシャルメディアプラットフォーム（ウィジェットまたはコンバージョンピクセル）
- Google Adsense、Media.net、BuySellAds、Amazon Associatesなどのサードパーティ広告ネットワーク
- Google Analytics、Crazy Egg、Hotjar、AdRollなどのウェブサイト分析スクリプトとトラッキングスクリプト
- Optimizely、VWO、UnbounceなどのA/Bテストツール
- Disqus、Jetpack、FacebookのコメントなどのWordPressのコメントシステム
- VaultPress、Sucuri、CodeGuardなどのバックアップおよびセキュリティツール
- SumoMe、HelloBarなどのソーシャルシェアツール
- KeyCDN、Amazon CloudFront、CDN77、StackPathなどのCDNネットワーク
- 外部ホストのJavaScript

上記サードパーティのトラッカーによるパフォーマンスへの影響度はどれほどでしょうか？当社のケーススタディでは、サードパーティのスクリプトにより**ページ読み込み時間が86.08%も増加した**ことがわかりました。

GhosteryがAlexaを使用し米国の上位500ドメインを測定したところ、驚くべき結果が出ましたが、私たちは驚きませんでした。トラッカーがまったくブロックされていないと、ウェブサイトは二分の一にまで遅くなりました。つまり、サードパーティのトラッキングスクリプトは、ページの読み込み速度が遅くなる主な原因の一つなのです。

AVERAGE PAGE LOAD TIME WITH vs. WITHOUT TRACKERS

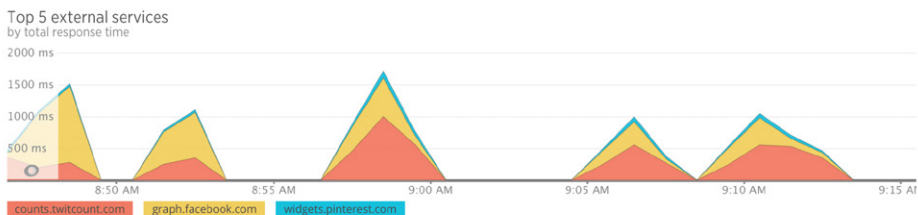


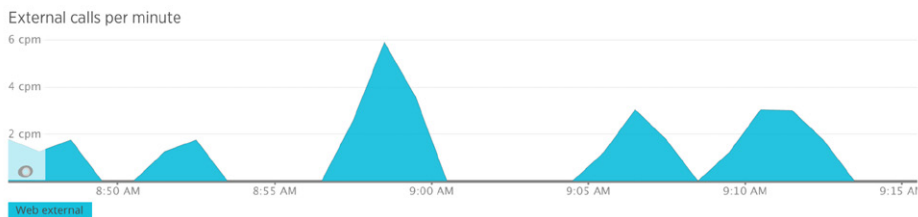
Average page load time (seconds) measured by Ghostery, of the top 500 US domains according to Alexa.

画像の出典: Ghostery

十分にご注意ください。劣悪なサードパーティAPIの呼び出しがたった1件発生するだけで、ウェブサイト全体がタイムアウトしてしまう場合があります。あり得ないとも言い切れず、実際にはよくある現象です。しかも、数え切れないほどの回数で発生します。

New Relicは、外部サービスを長期にわたって監視できる優れた手段です。以下の例では、[twitcount.com](https://counts.twitcount.com)、graph.facebook.com、およびwidgets.pinterest.comへの外部呼び出しが行われているのがわかります。

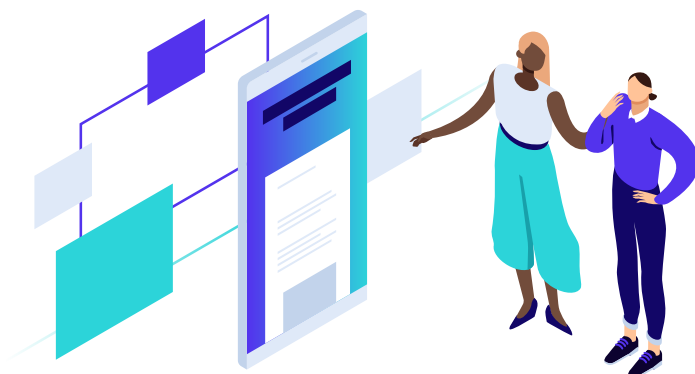




ウェブサイトに新しい機能及びプラグインを追加するときには、そこから読み込まれる外部リソースを調査することは非常に重要です。少ないほど理想的です！

第16章:

モバイルファーストを 念頭に置いて最適化する



Googleは2018年3月26日にモバイル ファーストインデックスを発表しました。以前は、Googleのクロール、インデックス作成、及びランキングシステムではウェブサイトのデスクトップ版が使用されていました。モバイル ファーストインデックスでは、Googlebotがインデックス作成と検索結果のランキングにWordPressウェブサイトのモバイル版を使用ようになります。これにより、モバイルユーザーの検索エクスペリエンスが向上する、という仕組みです。

モバイルを優先するようにウェブサイトを最適化する際に、**スピードが最も注目すべき要素の一つになります**。スピードは、ユーザビリティと直帰率をはじめ、顧客が次回もウェブサイトにアクセスしてくれるかどうかを決定するもので、重要な役割を果たしています。スピードは、モバイル向けのGoogle検索とGoogle広告のランディングページの判断要素になっています。

モバイルでのユーザーエクスペリエンスが悪いと、ユーザーの大半は二度と戻らないでしょう。Googleの最新のページスピードレポートによると、2018年にモバイルウェブサイトの読み込みにかかる平均時間は15秒でした。あなたならそこまで待ちますか？多分、待ちませんね。

ユーザーはもっと優れたサービスを期待しています。このレポートによると、**モバイルサイトの訪問者の53%が読み込みに3秒以上かかるとそのサイトの読み込みを諦めています**。

モバイルエクスペリエンスが遅いと、コンバージョン率が低下するだけでなく、見込み客のコンバージョンのチャンスさえなくなります。ページの読み込み時間がほんの数秒だけ増加するにつれ、直帰率が指数関数的に成長します。モバイル向けに最適化する際に次を考慮してください。

モバイルによるトラフィックを確認する

優先順位が少し変わる可能性があるため、モバイルによるトラフィックの量を確認することは重要です。Google Analyticsの「ユーザー」→「モバイル」→「概要」で、ウェブサイトにアクセスするモバイルデバイスの数を確認できます。ご覧の通り、例のウェブサイトのトラフィックの67%以上がモバイルからのトラフィックです。かなりの割合ですね！

▼ Interests

▼ Geo

▼ Behavior

▼ Technology

▲ Mobile

Overview

Devices

▼ Cross Device BETA

▼ Custom

▼ Benchmarking

Users Flow

Acquisition

Primary Dimension: Device Category

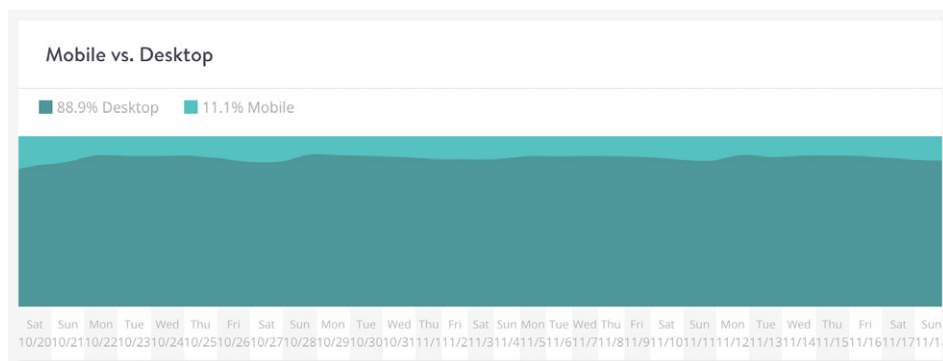
Plot Rows

Secondary dimension ▼

Sort Type: Default ▼

☐	Device Category ?	Acquisition	
		Users ? ↓	New Users ?
		26,614 % of Total: 100.00% (26,614)	24,262 % of Total: 100.00% (24,261)
☐	1. mobile	18,048 (67.93%)	16,174 (66.66%)
☐	2. desktop	6,783 (25.53%)	6,459 (26.62%)
☐	3. tablet	1,739 (6.54%)	1,629 (6.71%)

Kinstaのお客様は、MyKinstaの分析機能を使用してモバイルによるトラフィックとデスクトップによるトラフィックを比較することもできます。ご覧の通り、例のウェブサイトのトラフィックの88%以上がデスクトップからのトラフィックです。そのため、推定に頼ることなく、必ず確認することが重要です。いくらモバイルの方が重要だと言われていても、あなたのウェブサイトもそうとは限りません。データを確認してください。



ウェブサイトがレスポンシブであることを確認する

Tested on: Nov 18, 2018 at 11:13 PM

Page is mobile friendly

This page is easy to use on a mobile device

Additional resources

- Open site-wide mobile usability report
- Learn more about mobile-friendly pages
- Post comments or questions to our discussion group

SCREENSHOT HTML

Kinsta

Premium WordPress hosting for everyone, small or large

Kinsta is a managed WordPress hosting provider that helps take care of all your needs regarding your website. We run our services on cutting-edge technology and take support seriously.

VIEW PLANS

A cookie is a piece of information that a website stores on a visitor's computer. We use this for some functionality on our website to work properly, collecting analytics to understand and improve a visitor's experience, and for personalized advertising. You can accept all cookies at once or fine-tune your preferences in the cookie settings.

COOKIE SETTINGS ACCEPT COOKIES

2019年—ウェブサイトがレスポンシブであることはもはや必須! つまり、コンテンツをモバイル用に自動的に縮小するメディアクエリの採用です。まだ採用していないという方は、おそらく競争から取り残されています。本記事で前述したWordPressテーマはすべて完全にレスポンシブ対応であり、どんなデバイスでも見栄えが保証されます。

Googleのモバイルフレンドリーツールを使用して、ウェブサイトがすべての要件を満たしているか確認できます。

srcsetが正常に機能しているかチェック

以前は、画像をスケーリングでアップロードし、CSSによるサイズ変更をしないことが非常に重要でした。ただ、WordPress 4.4では(CSSで縮小されない)レスポンシブな画像がサポートされるようになった為、これは以前ほど重要ではなくなりました。WordPressは自動的にメディアライブラリにアップロードされた各画像に対していくつかのサイズの画像を生成します。利用可能なサイズがすべてsrcset属性にあり、ブラウザは最も適切なサイズのみをダウンロードし、それ以外を無視できるようになりました。下記はコードの一例です。

```

```

なお、サードパーティ製の画像プラグインとカスタム設定のせいで、これが正常に機能していないことがよくあります。そのため、画面サイズに合わせた画像サイズのsrcset属性が正しく追加されていることを再確認することが重要です。画像の最適化は極めて重要になりました。

Google AMPも役に立つかも

Google AMP (Accelerated Mobile Pages Project、和訳:高速モバイルページプロジェクト)は、2015年10月にローンチされました。このプロジェクトは、ウェブサイトが軽量のウェブページを構築できる既存のWeb技術をベースにした新しいオープンフレームワークであるAMP HTMLに基づいています。簡単に言えば、現在のウェブページの軽いバージョンを配信できる手段です。

当社のメンバーはGoogle AMPに対して愛憎入り混じる感情を抱いています。コミュニティの多くもそうです。当社の試験で良い結果は出ませんでした。全然でないわけではありません。ウェブサイトはそれぞれ異なり、Google AMPも継続的に改善されています。

次のプラグインのいずれかを使用することにより、WordPressウェブサイトでGoogle AMPをすぐに使い始められます。

- AMP for WordPress (無料で、WordPress.orgで入手可能)
- AMP for WP (無料で、WordPress.orgで入手可能)

まとめ

ご覧の通り、当社のスタッフはWordPressのスピードアップに夢中です。ウェブサイトが速いと、検索結果のランキング、検索エンジンでのクローリングの精度とコンバージョン率が改善され、訪問者がウェブサイトで過ごす時間が長くなり、直帰率が減少されます。誰もが速いウェブサイトを訪れるのが好きである--この事実は言うまでもありません！

本スピードアップガイドの内容の一部でも、参考になり、あなたのWordPressサイトに適用できるものがあれば幸いです。

<https://kinsta.com/affiliates>にアクセスして当社のWordPressホスティングアフィリエイトプログラムも是非ご確認ください。紹介ごとに最大500ドルを稼ぎ、さらに、**毎月の支払いに対して10%の手数料を獲得**できます。業界随一のホスティングアフィリエイトプログラムです。Webと一緒に加速させましょう！

Kinstaの詳細は当社のウェブサイト<https://kinsta.com>をご覧ください。

お読みいただき
ありがとうございます。



メモ

WordPressウェブサイトのスピードは意外と重要です。

あなたのウェブサイトはWooCommerceストアであろうとブログであろうと、ページの読み込み時間は以下の項目に影響を及ぼします。

- ・ キーワードランキングの改善（スピードはGoogleが使用する要素です）
- ・ コンバージョン率の向上
- ・ GoogleやBingなどの検索エンジンによるクロージングの精度向上
- ・ ウェブサイトでの平均滞在時間が長くなること
- ・ 直帰率の低下

言うまでもなく、読み込みの速いウェブサイトだと訪問者のユーザーエクスペリエンスが向上します。

インターネット上の全てのウェブサイトの3分の1がWordPressで構築されています。これは素晴らしいことである一方、何千ものテーマ、プラグイン、そして技術が共存しなければなりません。WordPressの一般のユーザーにとっては、ウェブサイトが壊れたときに、何をどう処理すればいいかわからない為、これこそ困難に他なりません。

Kinstaスタッフがこなす上記の異常処理は日常業務で、その知識は膨大です。当ガイドでは、今すぐ実施できるWordPressウェブサイトのスピードアップの手順について詳しくご説明します。初心者向けのアドバイスを始め高度な戦略など便利な内容が盛りだくさんです。15年以上の経験とそこから得られた教訓をご紹介します。

KINSTA