

WalkMeパフォーマンスのベストプラクティス

概要

WalkMeは、すべてのリソースを**非同期**に読み込むつまり、サイトを読み込むためにサイトを保持していないということです。WalkMeは、このようにパフォーマンスを最適化するため、エンドユーザーエクスペリエンスやサイトのパフォーマンスに目に見える影響は与えません。

WalkMe Playerのパフォーマンスは、2つの部分に分かれています：

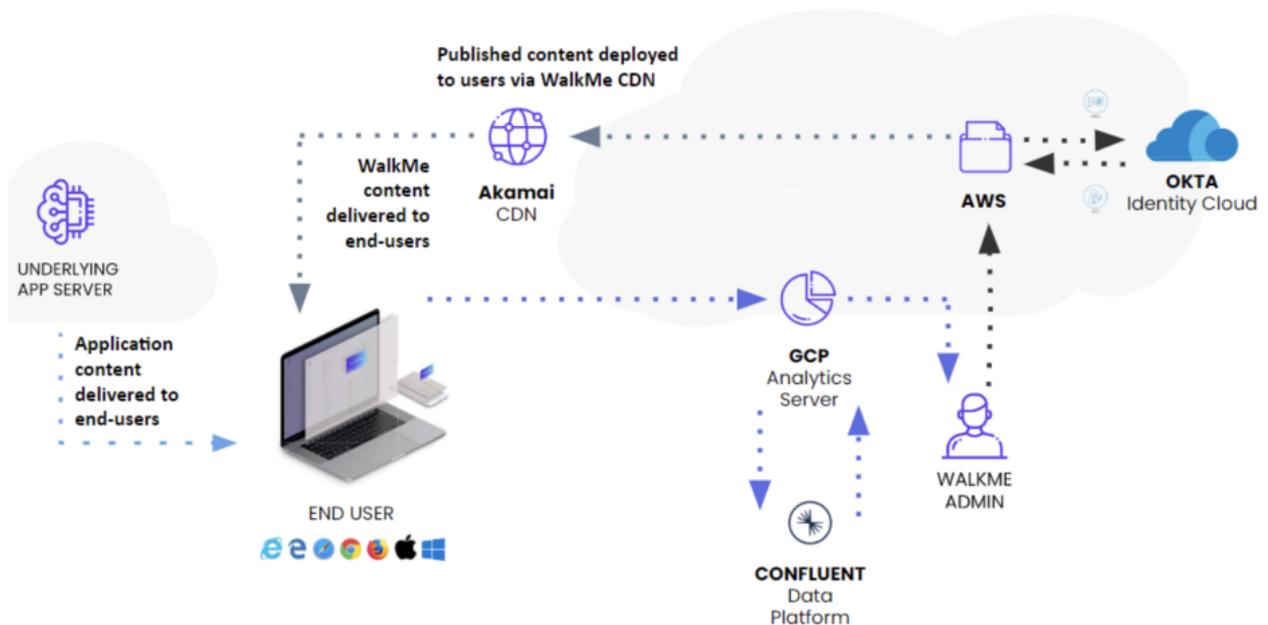
1. 読み込み性能
2. ランタイムのパフォーマンス

読み込み性能

WalkMeはブラウザ上で実行され、ブラウザリソースを読み込むために使用されます。WalkMeが読み込む速度は、ユーザーのネットワーク接続によって異なります。ネットワーク接続が高速になるとWalkMeの読み込みが全体的に速くなります。

WalkMeは、ファイルを複数の小さなファイルに分割して、大きなファイルを一度に読み込む必要がないようにします。

WalkMeエンジンは、ブラウザとWalkMeコンテンツ配信ネットワーク(CDN)にキャッシュされる静的リソースに基づいています。CDNは、エンドユーザーの物理的な場所に最も近い静的リソースを配信します。



ランタイムのパフォーマンス

WalkMe自体または公開されたコンテンツは、自然に非常に効率的に読み込まれます。WalkMeが可能な限り最高の方法でパフォーマンスと読み込みを継続するには、パフォーマンスを最適化してWalkMeコンテンツを構築することが重要です。

ページ読み込み評価

デフォルトではWalkMeは、ページの読み込み時にセグメンテーションとエンゲージメント/自動再生ルールをチェックします。サイトがシングルページのアプリケーションである場合WalkMeは、エディタのSPA(シングルページアプリケーション)タブで設定した設定に基づいて、これらのルールをチェックします。このチェックは、ページ読み込み評価と呼ばれます。

ページの読み込み評価の数と設定方法は、ページの読み込み時にWalkMeが読み込む速度に影響を与える可能性があるため、これは重要です。

WalkMeがページの読み込み時にチェックするもの：

- **セグメンテーションタグ**：セグメンテーションによりWalkMeコンテンツを表示する時期と場所を決定できます。 [続きを読む](#)
- シャウトアウト、スマートウォークスルー、およびサーベイの自動再生ルール [続きを読む](#)
- 表示ルール：動的または静的な属性に基づくルールエンジン内のロジックステートメント。 [続きを読む](#)
- **ウィジェットのセグメンテーション**：ルールエンジンで利用可能なオプションに基づい

てWalkMeウィジェットが表示される場所とタイミングをセグメントします。 [続きを読む](#)

- **UUID**設定されている場合、一意のユーザー識別子を検出します。 [続きを読む](#)

継続的な評価

WalkMeは、新しいページに移動していない場合でも、目標、エレメントの動作、またはセグメント評価などの追加のアイテムをチェックします。これは、継続的評価と呼ばれます。

継続的な評価の数と設定方法は、パフォーマンスの観点から留意するために重要です。

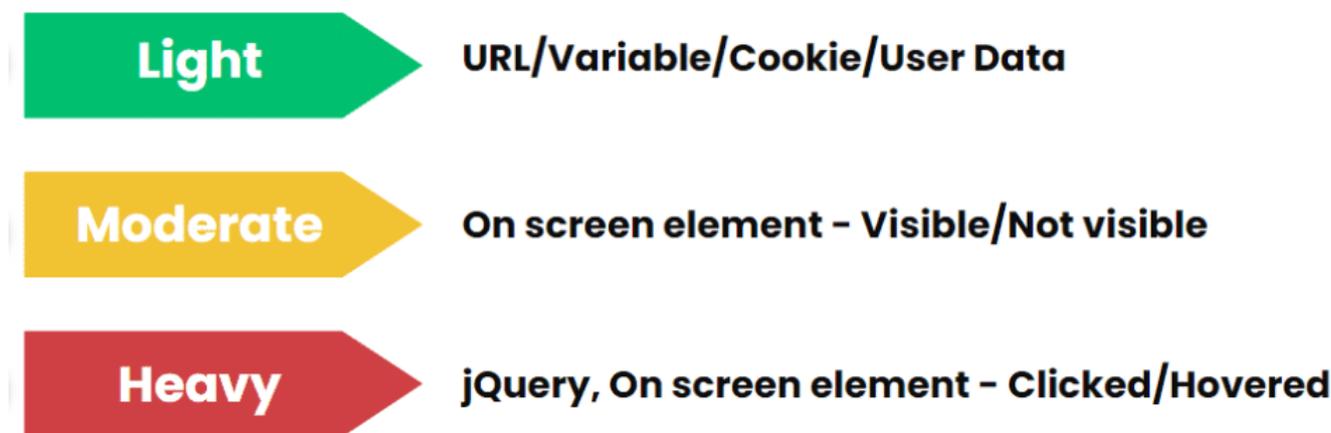
WalkMeが継続的に評価するもの：

- ラジオボタンが選択されている場合、選択したエレメントオプション：
 - ページの読み込み時に表示されたままにする
 - ユーザーアクションの結果として表示 / 非表示
- 「常にチェック」がオンになっている場合は、条件を表示します
- スマートウォークスルーおよびウォークスルー目標（フロー開始後2時間）
- Onboardingのゴール
 - スマートウォークスルーが開始されたかどうかに関係なく、常に追跡されます

ビルドしてWalkMeパフォーマンスを最適化

Rule Engine (ルールエンジン)

ルールエンジンは、自動再生ルール、目標、およびセグメンテーションなどの構築に使用できます。考慮する条件のパフォーマンスを最適化するためのルールを作成する際の、階層に従う：



ルールエンジン条件パフォーマンス階層

一般に、WalkMeがルールエンジンで評価する最速の条件は、URL変数Cookieおよびユーザーデータです。これらは、すばやく評価してWalkMeコンテンツを誰とどこに表示できるため、最初に使用

したいルールです。

画面上のエLEMENTとjQueryの含量を評価するための遅い条件。

より遅い条件を使用することは、必ずしも評価が目に見える速度になることを意味しません。しかし、統計的にいくつかのルールが迅速にチェックするのが簡単であることを知っている、それは速い条件が常に存在することを確認することがベストプラクティスです。高速条件がオプションではない場合は、より遅い条件のみを使用して持っているルールの数を制限してみてください。

構築時に実行するためのパフォーマンスルール

1. 重いルールよりも軽いルールを常に選択するようにしてください（両方を適用できます）
2. 重いルールがある場合は、常に軽いルールを添付してください。軽いルールが間違っている場合WalkMeは重いルールを評価しようとさえしません
3. 重いルールをあまり含まないようにしてください。

セグメンテーション

WalkMeは、セグメンテーションが真実である場合、画面上のエLEMENTの検索を開始し、アイテムを読み込みます。セグメントタグは、アイテムのグループに不変のルールを添付します。たとえば、ユーザーグループまたはオフィスの場所。セグメントタグは、大規模なアイテムグループについてWalkMeがルールを一度だけ評価できるため、非常に効率的です。同じセグメンテーションロジックに従う多数のアイテムがある場合は、複数のアイテムで同じ表示条件を再利用するよりも、セグメンテーションタグを使用します。

スマートチップとランチャーのための継続的な評価

表示条件で設定された条件がページ読み込み時に偽の場合、ランチャー/スマートチップは画面に描画されません。「表示と非表示」オプション（以下を参照）を使用してWalkMeにやり直すように伝えますWalkMeは、スマートチップまたはランチャーを再生するかどうかを確認するたびに、条件を最初に調べます。

ページ読み込み時に表示条件で設定された条件がTrueであるが、後に偽（ページ更新なし）に変更される可能性がある場合、「常にチェック」オプションを使用して、偽の場合にスマートチップ/ランチャーを削除するようにWalkMeに指示します。

この機能は、絶対に必要なときにのみ、賢明に使用します。これを一握りのスマートチップに追加すると、読み込み時間が短縮されます。多くのスマートチップにこの機能を追加する必要がある場合は、エンドユーザーエクスペリエンスをQAして、遅延の可能性を確認します。

Rules SmartTips および Launchers

1つのページに何個のスマートチップまたはランチャーを付けることができるかについては、サイトによって異なります。サイトが必要とするスマートチップとランチャーを、WalkMeがページで実行する

必要がある評価が長くなることを理解して、サイトが必要とする数のスマートチップとランチャーを作成します。

最適化されたエクスペリエンスを確保するには、これらのガイドラインに従ってください。

- 同じセット内のすべてのスマートチップは、同じページに表示されます。スマートチップ設定表示ルールは、ページ読み込み時に1回のみ評価され、個々のスマートチップ表示条件は、特定の設定と選択したエレメントに従って評価されます。
- スマートチップとランチャーの実際のルールを最適化することは、スマートチップとランチャー自体の数よりも重要です