



障害を防げるか、障害対応に追われるか…
その勝敗は『事前テスト』で決まる。



保野 高之
ほの たかゆき

株式会社バイキューブ
CTO室

<http://x.com/hono>

経歴



趣味

居合・素潜り・山歩き

ピークキューブ
powered by V-cube

数々の大規模イベントを
支えてきたノウハウで、
確かな負荷テストを。

“実際のブラウザ”による数万人規模のラッシュテストサービスのご案内

なぜユーザー体験を含めた負荷テストが重要か

多くの企業では「負荷テスト = サーバーにリクエストを投げて応答を見るもの」

従来の負荷テスト

- APIレスポンスが「OK」でも...
- サーバーに問題がなくても...
- ログにエラーがなくても...

実際のユーザー離脱が発生

- 「画面遷移などの動作が遅い」
- 「操作が効かない」
- 「UI要素が正しく表示されない」

リクエストベースの負荷テストの強みと限界

✂️ **JMeter など従来の負荷テストツール**は多くの企業で活用されています
サーバー側の応答性能を検証する上で効果的なアプローチです

✔️ **強み：サーバーの応答速度を効率的に測定できる**
リクエストをシミュレーションし、バックエンドの安定性を確保

❗ **限界：UI レベルの問題を検出できない**

⚠️ **API が OK を返していても、ユーザーが見ている画面は白のまま**
フロントエンドのJavaScriptエラーやレンダリング問題は検知できません

⚠️ **レスポンスが速くても、JavaScript エラーでボタンが押せない**
UIの操作性やインタラクションの問題は従来型テストでは見逃されがちです

「API 正常 = ユーザー満足」ではないというのが実情です



ユーザーの実体験

サーバーログ 200 OK ✓

応答時間 0.8秒 ✓

データ取得 完了 ✓

ユーザー画面 真っ白 ✕

ブラウザコンソール

```
Uncaught TypeError: Cannot read property  
'data' of undefined  
at renderPage.js:42
```

従来型負荷テストの落とし穴

JMeterなどの従来型テストではAPIレスポンスを確認するのみ
サーバー応答が正常でもユーザー体験に問題が発生するケースが発生



API 応答は 200 OK
でも...

- ✕ ユーザーの画面は白のまま
- ✕ 画像やアイコンが読み込まれない
- ✕ レイアウトが崩れて操作不能



ユーザーからの問い合わせ

- ✕ 「ログインできない」
- ✕ 「ボタンが押せない」
- ✕ 「動画が30分で止まる」

API テスト だけでは 不十分。 UI の挙動 も検証が必要

ピークキューブとは？

🖥️ 実ブラウザを活用した負荷テストサービス

実際のブラウザ操作を再現し、高負荷時のユーザー体験を検証します

3つの特徴

特徴	詳細
👤 ユーザー操作をそのまま再現	ログイン、検索、購入、動画再生など実際の操作フローを忠実に再現
👁️ UIレベルの検証	画面描画の遅延、表示崩れ、ボタンの反応も検証可能
🔄 リアルタイム検証で即時 課題特定	お客様のシステム管理者様と負荷テストを共に実施し、テスト実行中に画面やエラーを「その場」で即時共有

「Evenな社会の実現」のために、サーバー健全性ではなく **ユーザー体験を守るテスト** を提供します

従来型テストとピークキューブの役割の違い

- ▶ 従来型の負荷テスト（JMeter）は、サーバー側のパフォーマンス測定に優れたツールです。
- ▶ ピークキューブは、ユーザーレベルまで踏み込み、画面の遅延や不具合を見つけられます。

項目	JMeter	ピークキューブ
テスト対象	 APIリクエスト 効率的なサーバー検証	 実ブラウザ+UI操作 ユーザー視点の検証
計測範囲	 HTTPレスポンス 基本性能の確認	 画面描画速度・UI反応 体感品質の確認
負荷再現	 単純リクエスト 大量アクセスの効率的再現	 実ユーザー操作の組み合わせ 実際の利用パターン検証
可視化	 サーバログ中心 データ分析に優れる	 実行中の画面とエラーを共有 直感的な問題把握

事例① 一斉回答型システム

- 👤 通常の3倍の同時アクセスが発生するリリース前負荷テスト
- ✔️ APIレベルではすべてレスポンスはOKが返り「問題なし」と判断
- ⚠️ **ピークキューブでの実ブラウザ検証で発見した問題**
回答送信後のページの表示に**30秒の遅延**が発生
- 🔍 **原因はサーバーCPUの高負荷による処理遅延**
- 📄 **事前修正でサービス堅牢性と信頼を守る**
本番でのユーザー離脱を防ぐことができた

ユーザーの実体験

バックエンド応答	200 OK ✓
データ取得	完了 ✓
ブラウザ表示	30秒遅延 ✗
ユーザー行動	離脱 ✗

ブラウザコンソール

```
Uncaught TypeError: Cannot read property 'length' of undefined
    at renderProductList (productList.js:142)
```

負荷テストの結果

修正後：ユーザー離脱を防止

事例② オンラインイベント

👥 全国から同時接続する5,000人規模のオンラインカンファレンス

✔️ **事前準備：5,000名規模での実環境検証を実施**

十分な負荷状況を再現し、潜在的な問題を発見

🔍 **原因はプレイヤーの描画遅延とバッファ処理の不具合**

長時間視聴時にメモリ使用量が増加し、フロント側のプレイヤー描画が追いつかず
「動画が止まる／要素が表示されない」現象を発見

🔧 **本番前にプレイヤーの最適化を実施**

メモリ管理の改善とバッファリング処理の調整で安定性向上

🏆 **具体的な成果：視聴継続率の向上**

配信が途中で止まる問題を事前検知

総合的な品質検証

大規模負荷テスト **実施済✓**

パフォーマンス検証 **実施済✓**

実環境検証 **5,000名で実施✓**

発見された課題

長時間視聴時のメモリ使用量増加ネットワーク変動時の再生安定性

負荷テストの結果

修正後：サービス安定性の向上

なぜ UI テストが重要なのか

- 👤 ユーザーは「画面が速く、確実に動くか」で満足度を判断します
- ⚠️ API が正常でも、UI が止まれば不満・離脱・クレームにつながります

UI 検証を含めることで得られる3つの価値：

- 📦 **機会損失を防止** 売上・契約機会を守り、ビジネス成果を最大化
- 🛡️ **信頼性を担保** 顧客からの信頼維持と競争優位性の確保
- ★ **ブランド価値を守る** 「使いやすい」「安定している」というブランド評価を保護

- 🔍 **ピークキューブは、ユーザー目線で本当に使えるかを確認できる方法です**
あらゆるブラウザ環境で、実際のユーザー体験を事前に検証し、安心を提供します

意図した箇所に負荷が正しくかかっていることを視覚的に確認

JMeterでは、意図した箇所に正しく負荷がかかっているかを確認するのが難しいツールです。
 ピークキューブは、負荷テスト中の各ブラウザの動作をサムネイル形式でリアルタイムに表示し、
 意図した箇所に負荷が正しくかかっていることを視覚的に確認が可能になります。



まとめ と ご案内

🔄 **負荷テストは「サーバーを見る」から「ユーザーを見る」へ**
従来のサーバーメトリクス中心から、ユーザー体験を重視するテストへのパラダイムシフト

❗ **API テストだけでは不十分。UI も含めてこそ本番で安心**
ユーザーの画面が正しく表示されなければ意味がありません

🖥️ **ピークキューブはリアルな利用環境を再現**
「本当にユーザーが使えるか」を実ブラウザで事前に確認し、本番リスクを最小化します

💡 **本日のキーメッセージ**
「API が正常でも安心できない。ユーザー体験を確認してこそ真の品質検証」



ピークキューブ

ユーザー体験を守る負荷テスト

公式サイト

<https://jp.vcube.com/load-test>

導入事例ダウンロード

業種別の成功事例を公開中

無料相談・お問い合わせ

まずはお気軽にご連絡ください

「Evenな社会の実現」のために
品質検証の革新に取り組んでいます。