



# OpenResty XRay 製品紹介

---

オープンソースソフトウェアと Web アプリケーションの  
トラブルシューティングおよびパフォーマンス最適化ツール

Powered by OpenResty Inc.

# 製品紹介

1. 業界の課題
2. 製品概要
3. 主な強みとハイライト
4. 基盤技術
5. 導入事例
6. 製品ビジョン
7. 詳細情報



# 現代ソフトウェア世界の課題

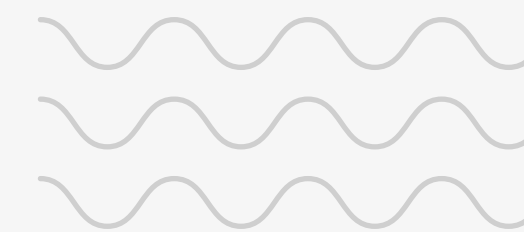


## ソフトウェア・ハードウェアシステムに対する洞察力と制御力の低下

---

- ソフトウェア・ハードウェアシステムの規模拡大、ビジネスロジックの複雑化、計算能力とユーザー数の指数関数的増加により、問題特定の難易度が上昇し、パフォーマンスのボトルネック、信頼性の問題、セキュリティの脆弱性が頻繁に発生しています。
- ほとんどの問題は本番環境特有であり、再現が困難です。オフラインでの検査はソフトウェアシステムの正常な運用とユーザーエクスペリエンスに深刻な影響を与えます。

# クラウドネイティブ導入の課題



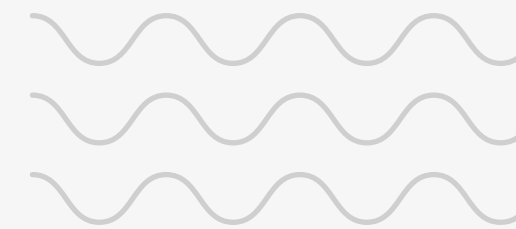
コンテナ技術とマイクロサービスアーキテクチャの普及に伴い、企業は前例のない複雑性に直面

---

- 技術スタックの急増：コンテナ化によりアプリケーション数が倍増し、様々な技術スタックとディストリビューションが共存することで、システムの複雑性が指数関数的に増加しています。
- 可観測性の欠如：最小化されたコンテナイメージには必要なデバッグツールが不足しており、問題の診断が著しく困難になっています。
- 問題の追跡困難：コンテナの迅速な破棄と再構築メカニズムはシステムの回復力を高める一方で、潜在的なソフトウェアの欠陥を隠蔽し、根本原因分析をより困難にしています。

# 業界の課題

---



- 複雑性のジレンマ：従来の APM ツールはアプリケーションにプローブコードを埋め込む必要があり、システムの複雑さを増すだけでなく、新たな不安定要素をもたらす可能性があります。
- リソースの浪費：大量のデータ収集、転送、保存は高コストをもたらすだけでなく、本番システムのパフォーマンスにも影響します。また、複雑な処理プロセスは誤検知や見落としを引き起こしやすくなります。
- 洞察力の不足：表面的な監視指標は問題の現象を反映するだけで、根本原因を深く分析することができません。技術スタック全体への深い理解の欠如により、問題の特定は表面的なレベルにとどまることが多いです。
- 対応の遅延：異常の発見から問題の確認、解決策の実施まで、プロセスが冗長で時間がかかり、急速に変化するビジネスニーズに対応することが困難です。

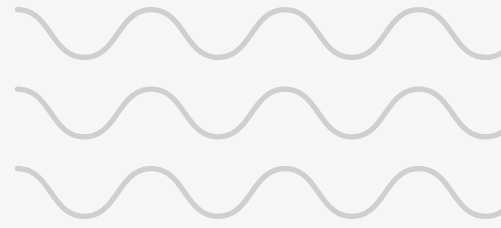
# OpenResty XRay

オープンソースソフトウェアと Web アプリケーションの  
トラブルシューティングおよびパフォーマンス最適化ツール

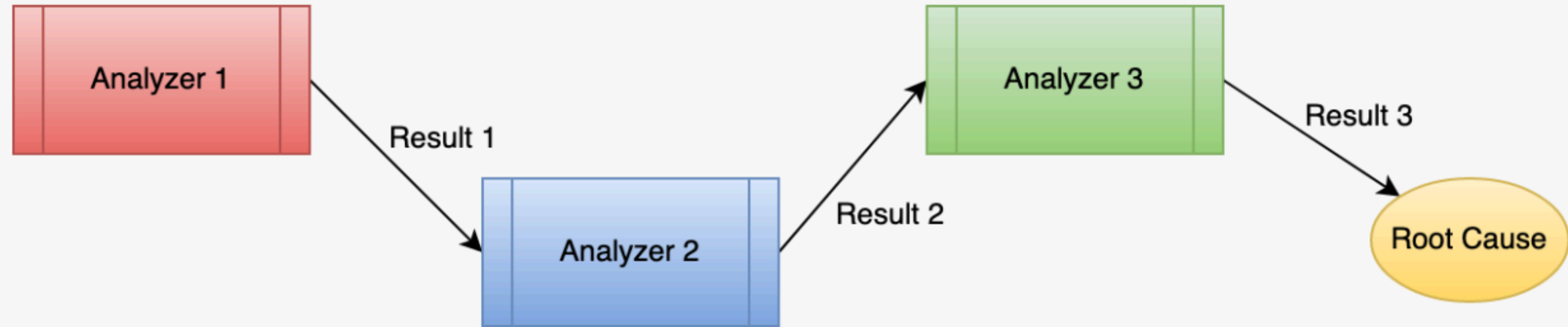
- OpenResty XRay は、企業の日々複雑化する運用課題を解決するために開発された、次世代の非侵入型動的トレースプロダクトです。
- コードを一切変更することなく、システムの稼働状態をリアルタイムで 24 時間監視し、クラウドおよびサーバーアプリケーションの深層分析を実現します。
- システムは自動的にすべての診断データを処理し、専門的な分析レポートを生成し、正確な最適化提案を提供することで、複雑な問題診断をシンプルで直感的なものにします。



# OpenResty XRay



完全自動サンプリング 無人監視モード



定期サンプリング



イベント駆動型

CPU 変動、メモリ変動、IO 変動、異常エラー

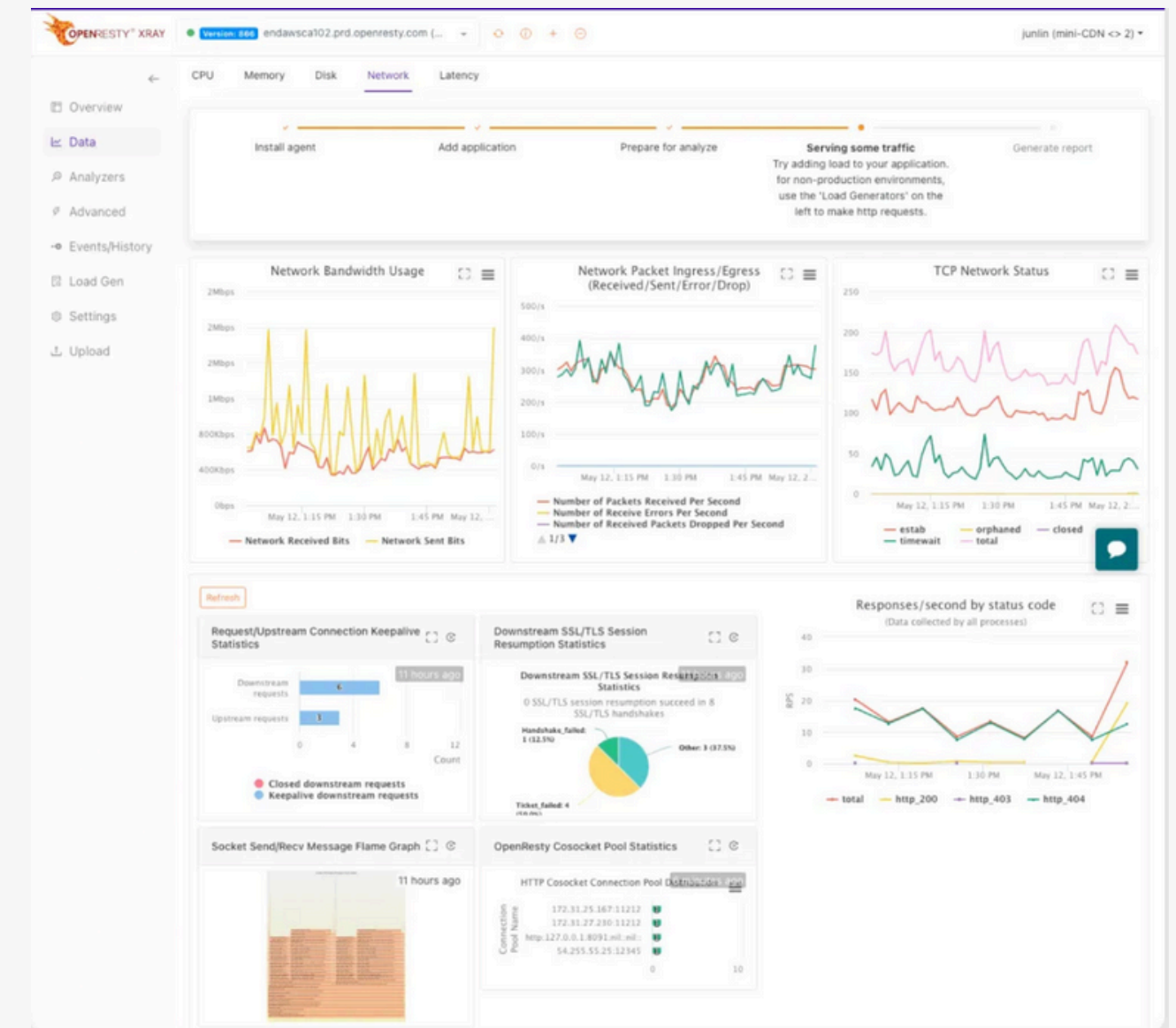


連鎖推論

# 各種システムパフォーマンス指標のリアルタイム監視

可視化モニタリングダッシュボードでシステム稼働状況を全方位から表示

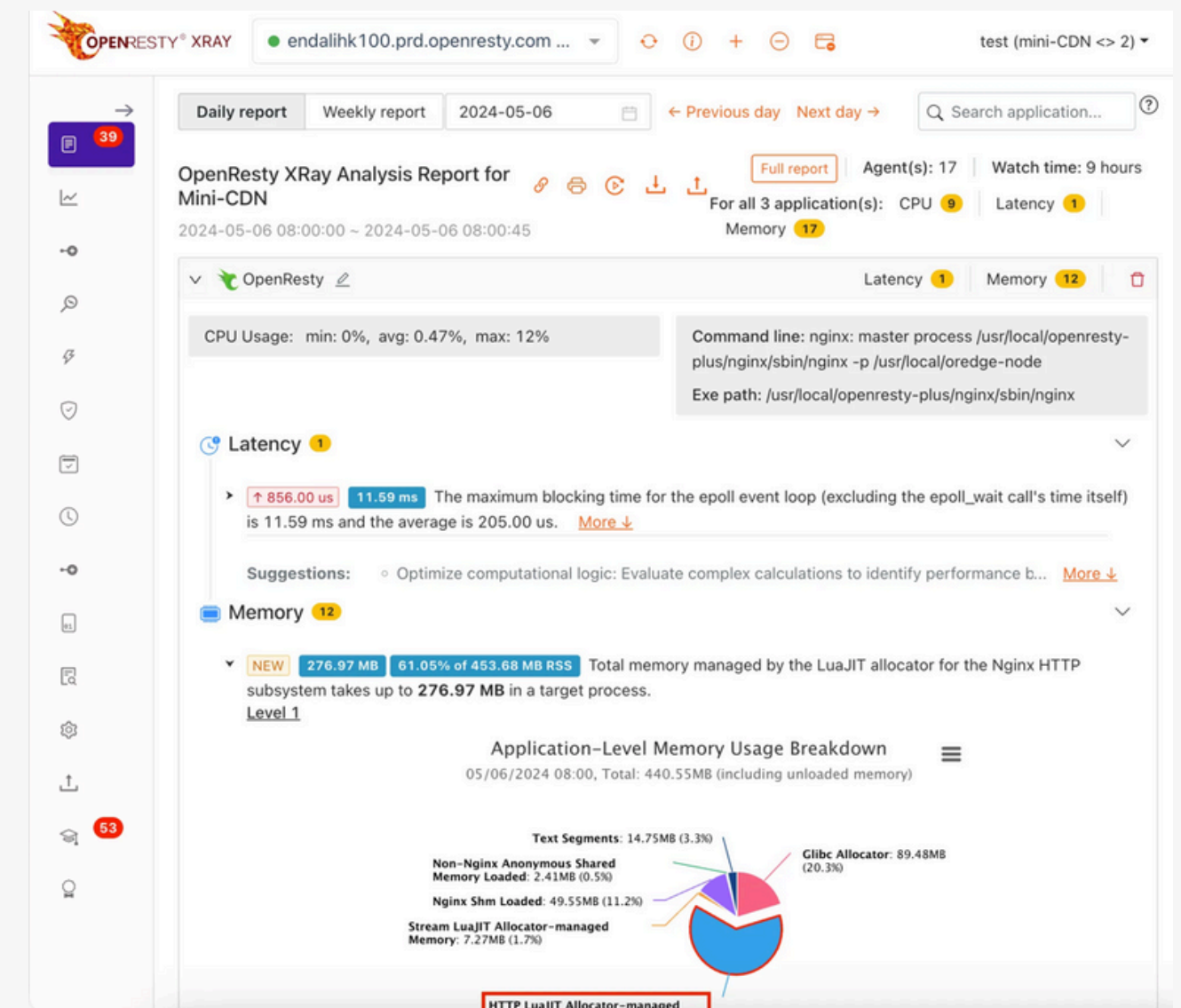
- 様々なシステム指標のサンプリング結果は可視化ダッシュボードに集約され、アプリケーションの稼働状況と性能の変化傾向をリアルタイムで把握できます。
- 異なるアプリケーション内や様々なプログラミング言語レベルにおける各種システム性能問題を自動的に分析し特定することが可能です。



# 全自動分析レポート

## 単なる追跡と分析だけではなく - サンプルングデータの深層分析

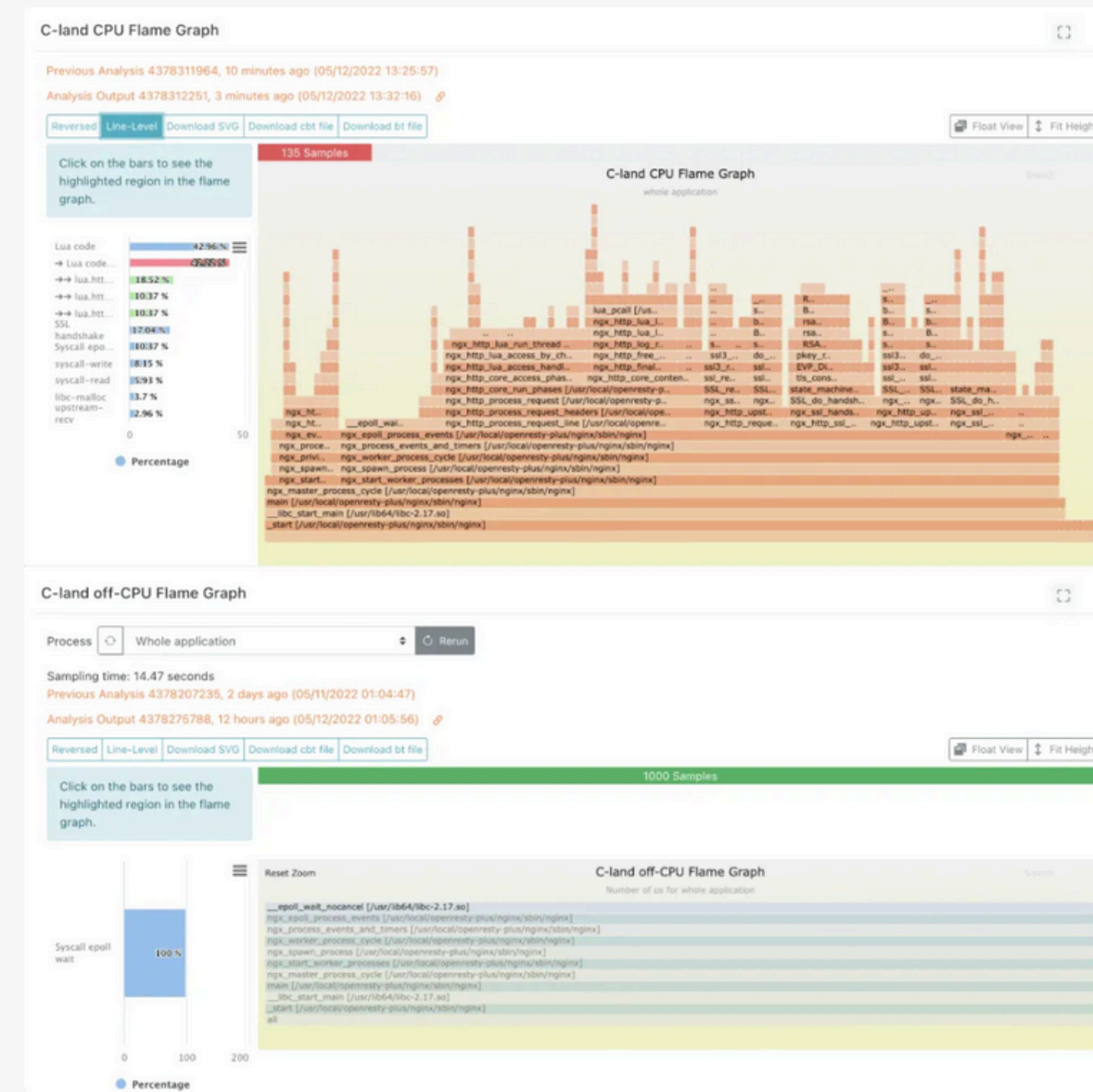
- パフォーマンスのボトルネック、リソース消費などの重要な問題に対して、XRay は専門的な根本原因分析と最適化提案を提供します。
- OpenResty の専門家チームがお客様の深層分析をサポートし、オンライン問題の根本原因を正確に特定し、複雑な問題診断をシンプルかつ効率的にします。



# ガイド付き分析

## 能動的分析 | リアルタイムデバッグ

- インタラクティブなトラブルシューティングツールで、実行中のアプリケーションをリアルタイムで観測・分析できます。
- 精密なガイダンスにより、最適化プランを迅速に検証し、再起動せずに動的デバッグとパフォーマンスチューニングを実現し、修正効果をすぐに確認できます。



# 主な強み



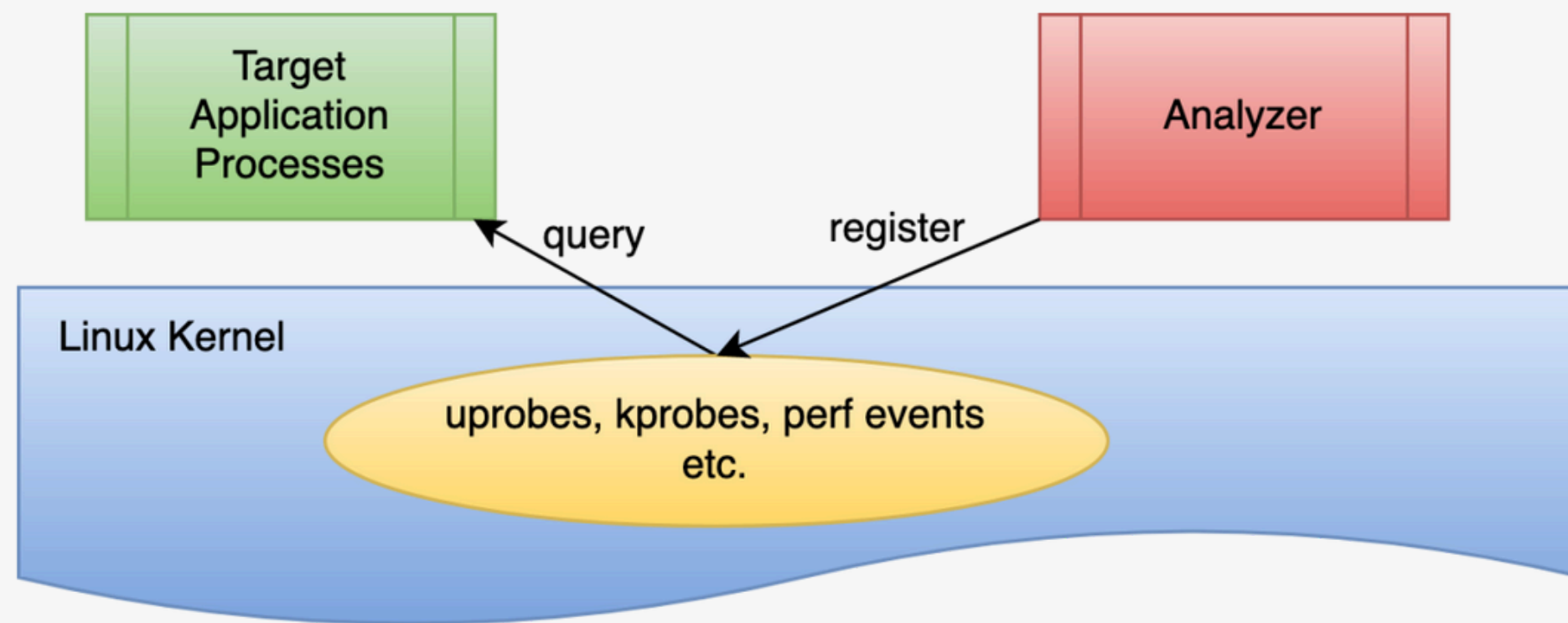
## お客様に選ばれる理由

---

1. 新世代の動的トレース技術による 100% 非侵入型システム
2. 革新的な自動サンプリング方法 - 高効率・軽量・低オーバーヘッド
3. フルスタック・全方位の死角なし分析 - あらゆるシステムパフォーマンス問題に対応
4. デバッグシンボルなしでもアプリケーション分析が可能
5. コンテナ透過化能力を搭載
6. セキュリティ問題の自動分析機能

# 新世代の動的トレース技術による駆動

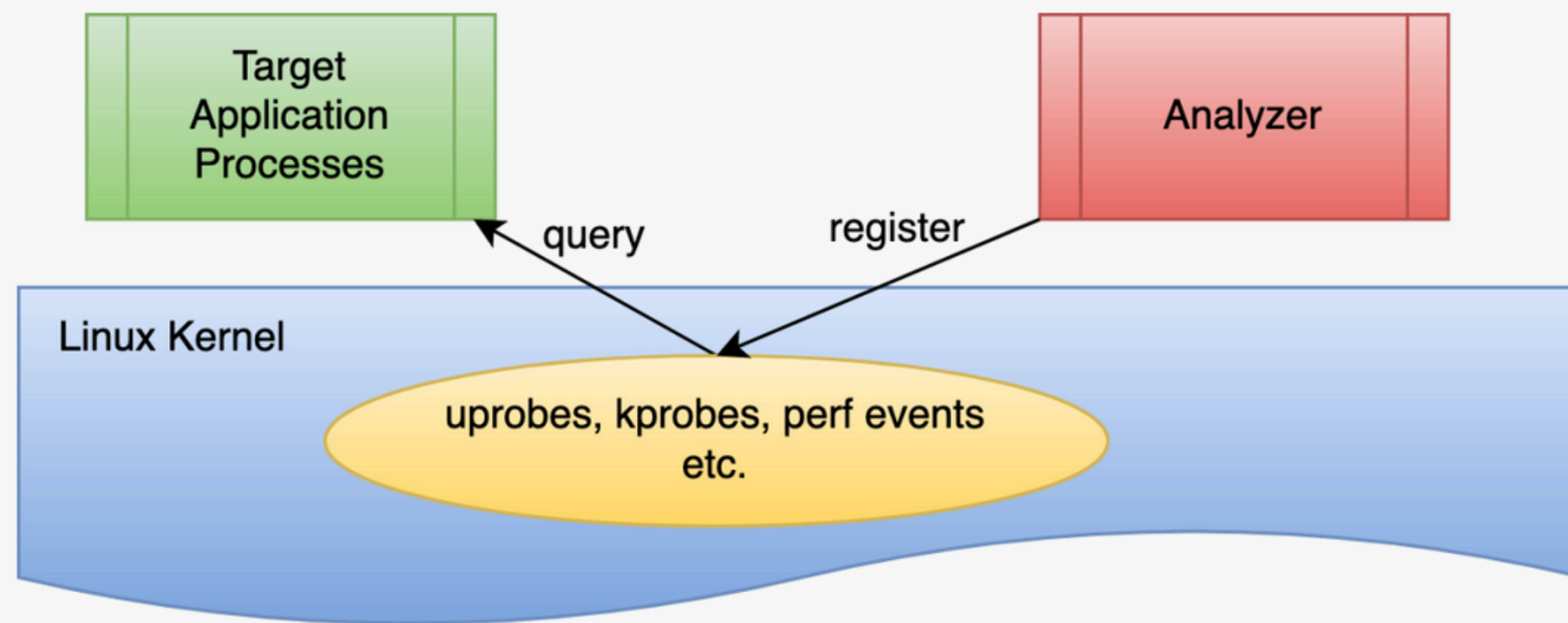
## 100% 非侵入型



- ホットプラグ対応：  
アプリケーションの変更や新しいプラグイン、パディング、モジュールのインストールは不要で、再起動や特別な起動・コンパイルオプションも必要ありません。また、アプリケーションやソフトウェアパッケージの再構築や、ターゲットプロセス空間へのコード注入も不要です。

# 新世代の動的トレース技術による駆動

## 100% 非侵入型

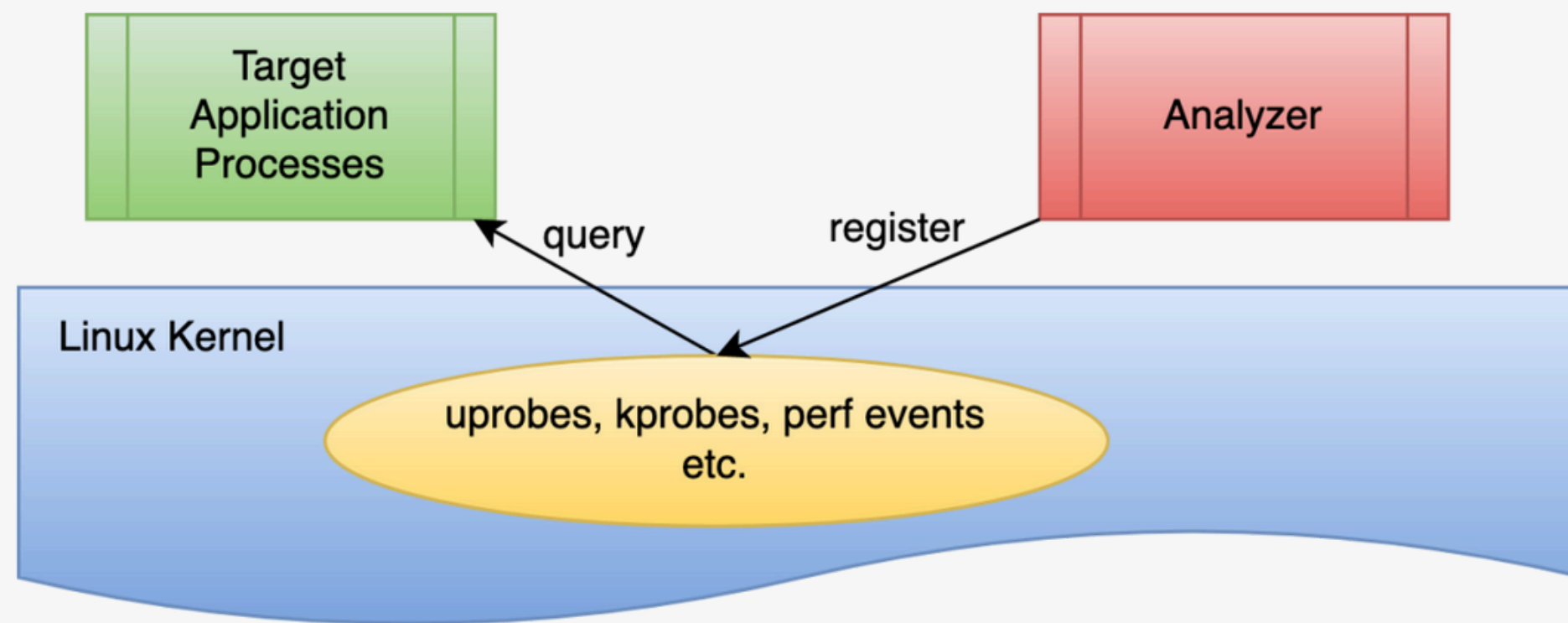


- 実行中のプロセスやコンテナを読み取り専用データベースとして扱い、パフォーマンスの問題、異常、エラー、セキュリティ脆弱性を解決するために必要な情報を抽出します。

# 新世代の動的トレース技術による駆動

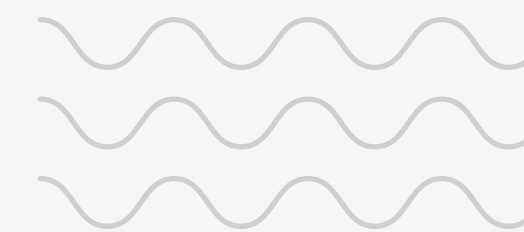


## 100% 非侵入型



- ユーザーのソースコードにアクセスすることなく、オンラインまたはオフラインのソフトウェアシステムを深く洞察し、ソースコードレベルの問題を特定し、システム問題が発生したソースコード行を正確に特定可能です。

# 革新的な自動サンプリング方法



効率的・軽量・低オーバーヘッド

---

1. 精密な指向性トレース、関連するコードパスと関数のみを追跡
2. 必要に応じたサンプリング、従来の APM 製品よりもはるかに少ない情報収集量、データソースでの集約が可能
3. サンプリングしていない時のパフォーマンス オーバーヘッドは厳密に 0、サンプリング中のオーバーヘッドも通常測定不能なほど低い

# フルスタック・全方位の分析

## あらゆるシステムパフォーマンス問題を網羅

---

1. 多層ソフトウェアのコードパスを完全カバー：ビジネスプログラミング言語（Lua/Python/PHP/Perl/Go など）、システムプログラミング言語（C/C++/Rust）、OS カーネル層（ネットワークプロトコルスタック/プロセススケジューラ/メモリ管理/システムコール）
2. スレッドのブロッキングとロック状態の解析
3. CPU 問題の詳細分析：CPU 高負荷やブロッキング、コード別 CPU 時間分布、一般的な CPU ボトルネックの特定
4. メモリ使用量過多やメモリリークの解決：メモリ分布の定量分析、メモリアロケータの使用状況把握、メモリリーク・フラグメンテーション・遅延解放などの検出
5. ディスク I/O パフォーマンスの最適化分析
6. 長時間レイテンシー・ロングテールリクエストの解析：アプリケーション処理段階別の遅延分解、非同期ノンブロッキング I/O のレイテンシー統計
7. インテリジェントパケットキャプチャと異常検知：問題のあるネットワーク接続のみを選択的に捕捉
8. コアダンプファイルとプロセスクラッシュの徹底解析

# デバッグシンボルなしでもアプリケーション分析を実現



OpenResty XRay は本番環境におけるデバッグシンボル不足の課題を解決

---

1. OpenResty XRay は業界をリードする中央シンボルデータベースを構築し、数百 TB におよぶ公開パッケージのデバッグ情報を蓄積しており、現在も急速に拡充を続けています。
2. 独自開発の機械学習アルゴリズムにより、デバッグ情報が完全に欠如したバイナリプログラムに対しても、OpenResty XRay はシンボルの自動再構築を実現します。
3. デバッグシンボルが一度 OpenResty XRay の中央パッケージデータベースにインデックス化されていれば、対象マシンにデバッグシンボルをインストールまたは保存する必要はありません。お客様は本番環境に大規模なデバッグファイルをデプロイすることなく、プログラムの実行時情報を完全に把握でき、軽量デプロイと深度分析の理想的な融合を実現します。

# OpenResty XRay コンテナ監視能力



OpenResty XRay は、コンテナ境界を革新的に透過し、ホストマシンに Agent を一つ配置するだけで、同一ホスト上の他の Docker コンテナや Kubernetes Pod で実行されているアプリケーションを自動的に検出し、サンプリング分析することができます。

1. コンテナイメージの修正が不要
2. 設定不要でコンテナ間監視を実現
3. コンテナのリソース制限やセキュリティポリシーに影響を与えない
4. 動的スケーリング環境での自動検出に対応

# 自動化セキュリティリスク検出

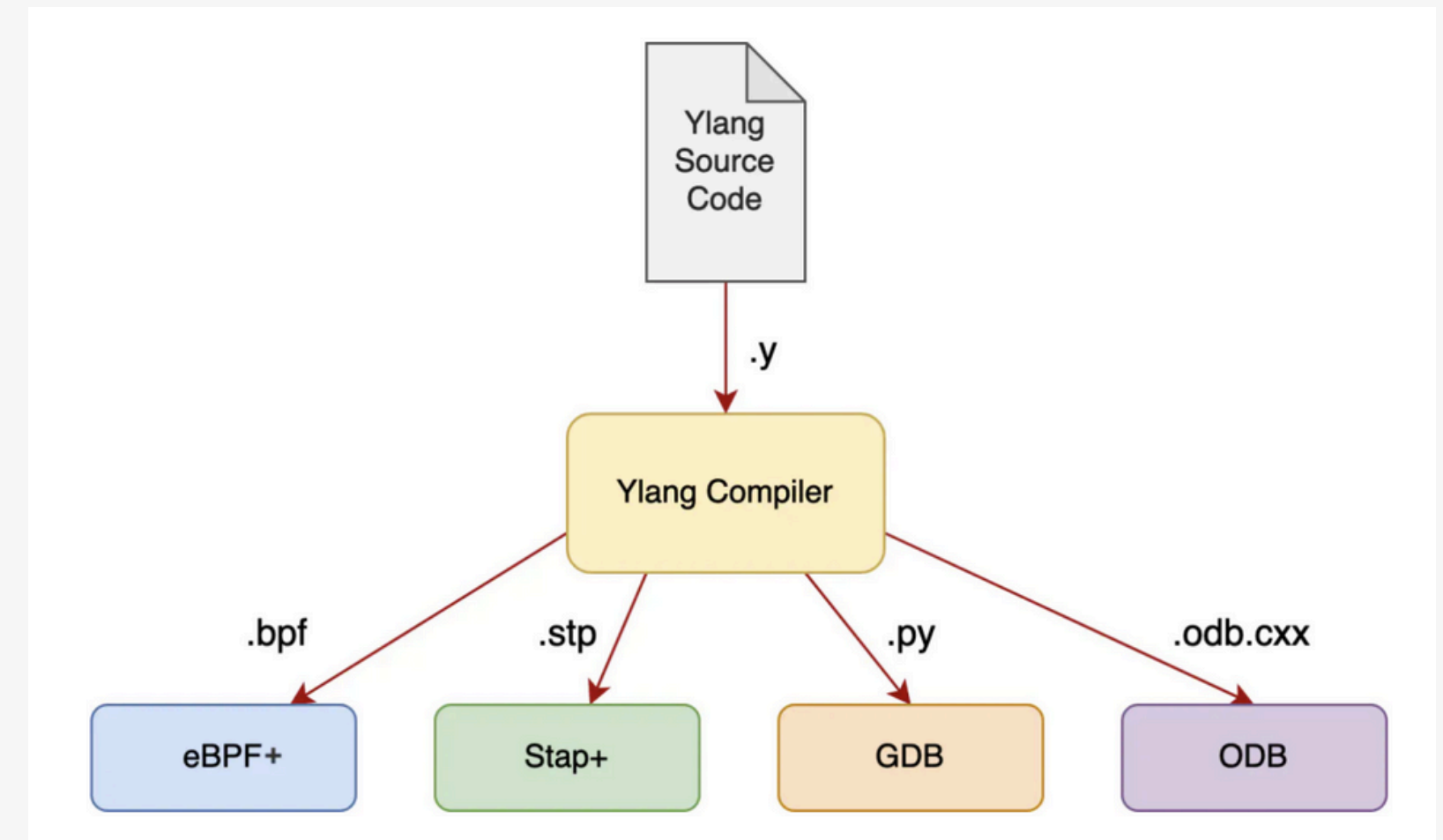


OpenResty XRay は包括的なセキュリティ脆弱性スキャン機能を提供する

1. システムはリアルタイムで転送層のセキュリティリスクを自動的に識別します。これには非暗号化通信、TLS 証明書検証の不備、および旧式の暗号化プロトコルが含まれます。
2. コード実行の挙動を詳細にスキャン・分析し、潜在的なリモート shell コマンドインジェクションの脆弱性を特定します。

# OpenResty XRay の基盤技術

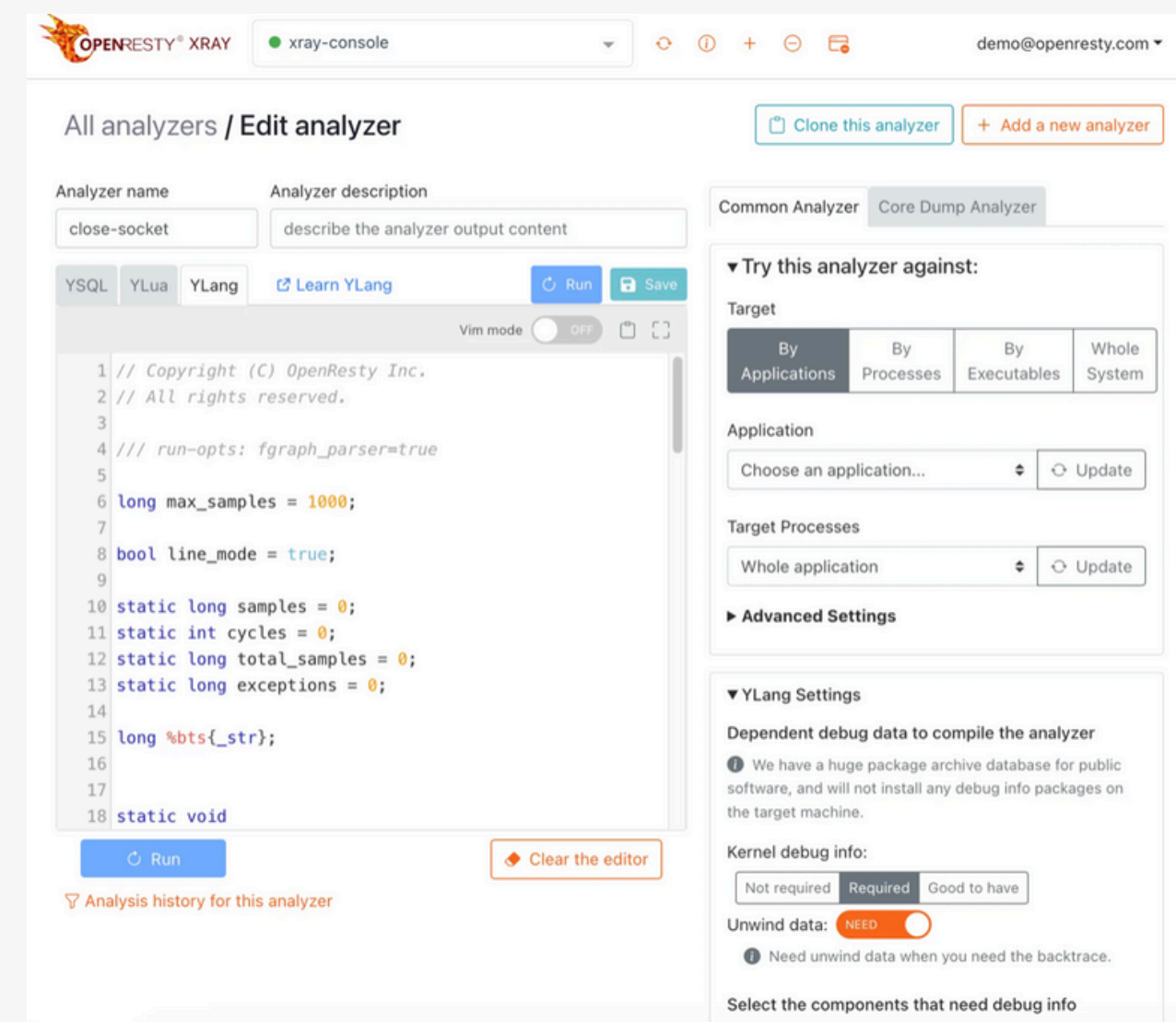
1. OpenResty XRay は、OpenResty Inc. が独自開発した Y 言語を基盤技術として採用しています。
2. Stap+、eBPF+、GDB、ODB など多様なデバッグフレームワークに対応しています。
3. ylua、ysql、y++ などの高度なツールキットを搭載し、優れたプログラム性と拡張性を実現しています。これにより、低コストかつ柔軟で効率的に複雑なシステム課題を解決することが可能です。



一度書けば、どこでも実行可能

# 数百種の標準アナライザー

1. 包括的なカバレッジ - 様々なタイプのパフォーマンス問題や異常状態を分析・診断する能力
2. 正確な原因特定 - 専用アナライザーによる問題の根本原因を迅速に特定
3. 高度な自動化 - 人的介入なしに複雑な分析作業を完了可能
4. 時間の大幅削減 - 問題診断にかかる時間を数日から数分へと短縮
5. 予兆検知 - 潜在的な問題を早期に発見し、未然に防止
6. カスタマイズ対応 - 専門家チームがお客様の要件に応じたアナライザーを提供



# 多言語パフォーマンスアナライザー

主要な Linux ディストリビューションとコンテナデプロイメント方式をサポート

---

1. Nginx、Envoy、Kong などの主要 Web サーバーとプロキシをサポート
2. LuaJIT、Python、PHP、Go、Java、Erlang、Perl、Ruby および Rust など多様なプログラミング言語とランタイム環境に対応
3. 主要 Linux ディストリビューションに適応：Ubuntu、Debian、Fedora、Red Hat Enterprise Linux (RHEL)、CentOS、openSUSE、AlmaLinux、Amazon Linux、Alibaba Cloud Linux、Rocky Linuxなどを網羅
4. コンテナデプロイメント方式をサポート：Docker、Kubernetes

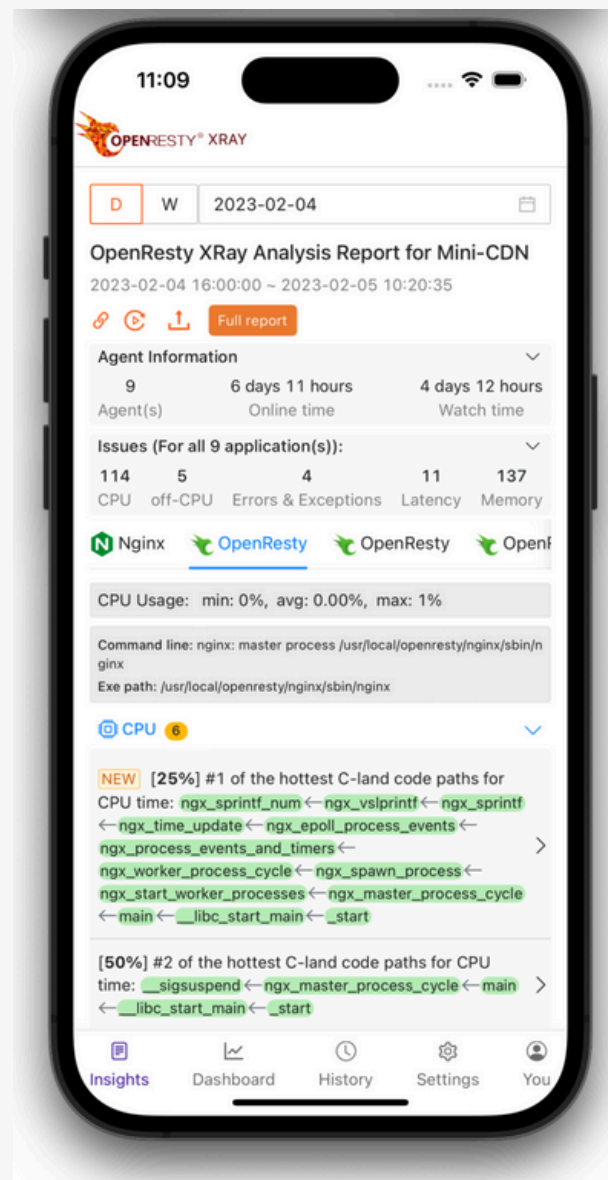
## 将来のサポート計画

- nodejs
- pytorch、langchain AI stack, etc.

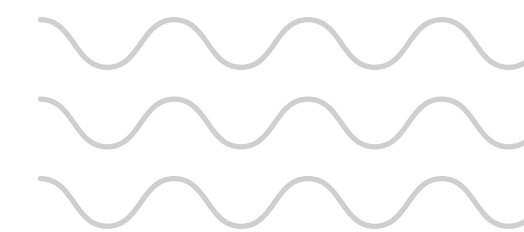
# OpenResty XRay モバイルアプリケーション

いつでもどこでもオンラインアプリケーションを監視

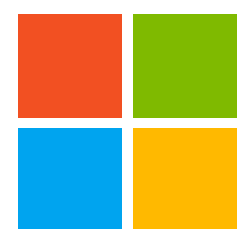
- Android / Google Play
- iOS / app store



# お客様から信頼される



GitLab



Microsoft

TED

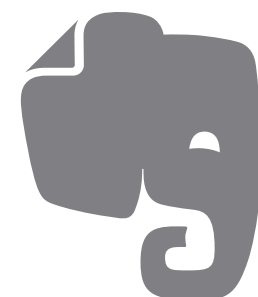
ADP



bilibili

zoom

IBM



EVERNOTE

vimeo

# OpenResty XRay によるビリビリの重大なオンライン障害の分析と解決

<https://blog.openresty.com/jp/bilibili-xray-incident/>

---

## 障害内容

- ビリビリの OpenResty ベースのゲートウェイシステムで全面的な障害が発生
- すべてのサーバーの OpenResty プロセスが CPU 使用率 100% となり、リクエスト処理が不可能な状態に
- 再起動やコードのロールバックも効果がなかった

## OpenResty XRay による調査

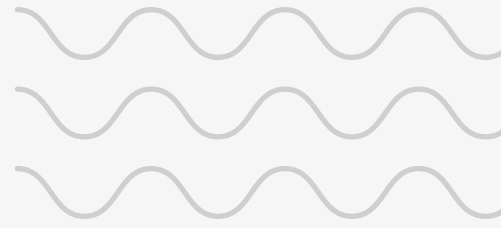
1. OpenResty XRay を使用して C 言語レベルの CPU フレームグラフを生成し、問題が Lua コード実行に集中していることを特定
2. さらに Lua レベルの CPU フレームグラフにより、具体的な問題コードパスを特定
3. 最終的に根本原因を発見：業務設定において文字列「0」が重み値として設定されていたが、システムは数値型の 0 を期待しており、無限再帰が発生していた

## 導入効果

- XRay によるサンプリング分析はわずか数十秒から数分で完了
- 100% 非侵入型の動的トレースにより、対象プロセスの変更が一切不要
- 問題のコードパスを迅速かつ正確に特定することに成功



# カスタム Kong プラグインにおける Lua 例外起因の CPU ボトルネックを解決



<https://blog.openresty.com/jp/lua-str-lower-excep/>

---

## 課題

トラフィック量が少ないにもかかわらず、お客様の Kong サーバーの CPU 使用率が異常に高く（80%）なっていた

## OpenResty XRay による調査

- ホットコードパスで string.lower 関数が例外をスローしていることを発見
- 具体的な原因を判明：カスタム認証プラグインで誤って nil 値を string.lower に渡していた
- ソースファイルの正確な位置：.../kong/plugins/auth/handler.lua の 35 行目を特定

## 解決策

認証プラグイン内の不具合を修正し、string.lower に nil 値が渡されないように対応

## 導入効果

- CPU 使用率が 80% から 50% へ大幅に低減
- パフォーマンスが 37.5% 向上
- 複雑なコードリファクタリングを必要とせず
- 問題を迅速に解決し、お客様の貴重な調査時間を大幅に節約することができました

# プラグインが CPU の 45% を食い潰した際、ソースコードなしで Lua の 93 行目を特定した方法？

<https://blog.openresty.com/jp/xray-rsa-cpu/>

## 課題

ある金融業界の基幹業務ゲートウェイクラスターでは、ピーク時に CPU 使用率が継続的に 100% に達し、P99 レイテンシ（99 パーセントイル遅延）が急激に悪化し、サービス品質が著しく低下していました。

## OpenResty XRay による調査

- ある金融業界の基幹業務ゲートウェイクラスターでは、ピーク時に CPU 使用率が継続的に 100% に達し、P99 レイテンシ（99 パーセントイル遅延）が急激に悪化し、サービス品質が著しく低下していました。
- OpenResty XRay による調査
- 驚くべきホットスポットを特定：C 関数 `pkey\_rsa\_decrypt` が CPU 時間の最大 44.8% を消費していました。
- 根本原因の特定：Lua の原因箇所である `cb-session-validation.lua` プラグインの 93 行目を正確に突き止めました。
- 呼び出しチェーンの可視化：`Lua プラグイン` -> `JWT ライブラリ (Lua)` -> `libcrypto.so (C ライブラリ)` へと続く呼び出しパスを完全に可視化しました。

## 導入効果

- 約 45% の CPU リソースを解放し、システムのスループットはほぼ倍増しました。
- ピーク時の CPU は上限に張り付くことがなくなり、P99 レイテンシは安定を取り戻しました。
- Lua から C への完全な呼び出しスタックを直接取得し、問題の根源を特定できました。
- 多言語間の性能問題の調査時間を大幅に短縮し、お客様の性能ボトルネックによる損失を回避することができました。

# OpenResty XRay で DNS 性能を 60% 向上させた徹底解析

<https://blog.openresty.com/jp/xray-customer-casestudy-dns/>

あるお客様の DNS サービスシステムは、深刻な性能問題を抱えていました。

- CPU 負荷の不均衡: 一部の Nginx worker プロセスの CPU 占有率が過度に高く、他のプロセスは比較的アイドル状態でした。
- 応答遅延の増大: システム全体の応答が遅延しており、高負荷時には特に顕著でした。

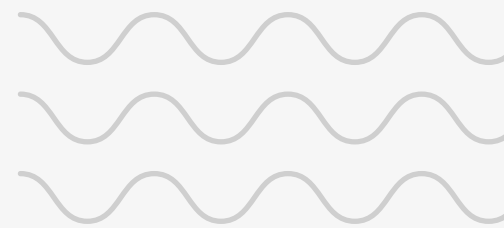
## OpenResty XRay による調査

- リクエスト割り当ての不均衡を発見: すべてのリスニングポートで `reuseport` オプションが有効になっていないことを特定しました。これが worker 間でのリクエスト割り当ての不均衡を引き起こしていました。
- C 言語レベルのフレームグラフ分析を通じて、`cjson` モジュール (JSON 解析) が CPU 時間の約 60% を消費しており、システムの絶対的なボトルネックであることが判明しました。
- コアビジネスロジックである `dns\_server.lua` は、CPU の約 5% しか占めていませんでした。
- `cosocket` 受信操作は CPU 時間の約 16% を消費しており、これもまた顕著な消費要因でした。

## 性能最適化の効果

- 主要なボトルネック (JSON 解析) における CPU 消費量を 60% 以上削減しました。
- 設定チューニング (`reuseport` の有効化) のみで、システム全体の性能が 20~30% 向上しました。
- システムスループットが著しく向上しました。
- 複数の性能問題を正確に特定し、お客様に明確で多層的な最適化の道筋を提供することで、分析と試行錯誤に要する時間を大幅に節約できました。

# OpenResty XRay による Java メモリ問題診断の実践



<https://blog.openresty.com/jp/xray-java-mem/>

お客様の注文システムでは、性能回帰テスト中にヒープメモリが継続的に上昇し続け、複数回の Full GC 後もメモリが解放されないため、深刻なメモリリークのリスクを抱えていました。

## OpenResty XRay による調査

- \* リークの根本原因を特定：GC オブジェクト参照分析を活用し、予期せずメモリ上に残存するオブジェクトとその完全な参照チェーンを迅速に特定します。
- \* メモリフットプリントの変動を調査：GC オブジェクト割り当て回数分析を通じて、一時オブジェクトを高頻度で生成しているコードパスを発見します。
- \* ラージオブジェクトを特定：GC オブジェクト割り当てサイズ分析を利用し、「ラージオブジェクト」の生成元を正確に突き止めます。

## 性能最適化効果

- \* メモリリーク問題の解決に成功し、システムの安定性を確保しました。
- \* サービスの再起動やコードの変更は不要で、本番環境に透過的に動作し、影響を与えません。
- \* JVM セーフポイント (Safepoint) に依存せず、分析プロセス中に STW (Stop-the-World) 一時停止は発生しません。
- \* 直感的な可視化グラフ (フレームグラフ、参照グラフ) により、メモリ問題分析のハードルを大幅に低減し、根本原因を迅速に特定できます。

# Gzip 設定の潜在的なコストをいかに正確に特定するか

<https://blog.openresty.com/jp/xray-gzip-cpu/>

---

## 課題

高スループットの金融機関レベルのビジネスシナリオにおいて、表向きのシステム指標（QPS、遅延）は基準を満たしているように見えますが、多大な CPU コストがスループット拡張のボトルネックとなっています。

## OpenResty XRay による調査

フレームグラフは、CPU リソースがコアビジネスロジックではなく、圧縮と解凍プロセスに大量に消費されていることを明確に示しています。

- \* Gzip 圧縮が CPU 時間の 54% を占めています。
- \* Gzip 解凍が CPU 時間の 61% を占めています。
- \* Brotli 圧縮でさえ CPU の 30.9% を消費しています。
- \* マクロな CPU 消費を、ミクロな `gzip\_level` などの設定パラメータに直接関連付けています。

## OpenResty XRay が提供するデータインサイトに基づき、Gzip 関連設定を最適化した結果

- \* システム全体の CPU 使用率が著しく低下しました。
- \* コアビジネススレッドの CPU リソースが解放されました。
- \* QPS が実質的に向上しました。
- \* リクエスト遅延が著しく改善されました。

# OpenResty または Nginx プロセスにおける最も遅い PCRE 正規表現の特定

<https://blog.openresty.com/jp/nginx-slowest-pcre-regexes/>

OpenResty/Nginx Lua アプリケーションの非効率性により、サーバーの CPU 使用率が常に 100% に達していました。

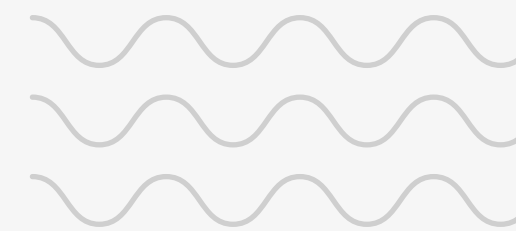
## OpenResty XRayによる調査

- 初期診断: C レベルの CPU フレームグラフから、CPU 時間の大部分が Lua コード内の PCRE 正規表現マッチングの実行に費やされていることが示されました。
- 詳細分析: Lua レベルの CPU フレームグラフにより、パフォーマンスのボトルネックが単一のコードパスにおける `ngx.re.match` API 呼び出しに集中していることがさらに確認されました。
- 正確な特定: `lj-slowest-ngx-re` アナライザーを使用し、最も遅い正規表現が `(?:.\*)\*css` であることが正確に特定されました。その実行遅延は 20.6 ミリ秒を超え、貪欲な量指定子 `.\*` が大量のバックトラックを引き起こしていたことが原因でした。

## 正規表現の最適化により、エンジンの大量のバックトラックを回避

- \* CPU ボトルネックの解消: CPU が Lua 正規表現の計算に費やされることがなくなり、ホットスポットは通常の I/O 書き込み操作 (`writev` システムコール) に移行しました。
- \* パフォーマンスの大幅な向上: 最も遅い正規表現の最大遅延は 20.6 ミリ秒から 59 マイクロ秒に減少し、パフォーマンスは数百倍向上しました。
- \* 問題の正確な特定: 推測や個別の調査を行うことなく、問題のある正規表現を直接特定できました。
- \* 複雑なビジネスロジックを変更することなく、根本的なパフォーマンス問題を迅速に特定し解決しました。

# OpenResty XRay - プロダクトビジョン



## 複雑なシステムの透明化と制御性の実現

---

- OpenResty XRay は、現代コンピューティング世界における最も困難な課題に挑戦しています：ゼロ侵襲技術により、あらゆるブラックボックスシステムをホワイトボックス化し、ハードウェアからビジネス層まで全スタックの可視化を実現します。お客様には、かつてない深いシステム洞察力を提供いたします。
- 当社は業界最高峰の自動問題診断エンジンを構築しています。対象システムの改変不要、プロセス再起動不要で、軽量 Agent のみで全ソフトウェアスタックを総合的に分析し、各種パフォーマンスのボトルネック、セキュリティリスク、安定性の問題を自動的に検出・特定します。
- さらに、革新的なリアルタイムホットパッチ技術の開発も進めています。将来的に OpenResty XRay は問題の発見だけでなく、稼働中のシステムに自動的に修正パッチを適用し、診断と修復をシンプルかつエレガントに実現します。

# 詳細はこちら

[OpenResty XRay](#) を無料でトライアル

公式サイト：<https://openresty.com/jp/>

ブログ：<https://blog.openresty.com/jp/>

X：<https://x.com/OpenResty>

ビデオチュートリアル：<https://www.youtube.com/channel/UCXVmwF-UCScv2ftsGoMqxhwr>