Product Introduction of OpenResty XRay

Troubleshooting and Performance Optimization Tool for Open Source Software and Web Applications

Powered by OpenResty Inc.

Table of Contents

1. Industry Pain Points
2. Product Introduction
3. Core Advantages & Highlights
4. Underlying Implementation
5. User Case
6. Product Vision
7. Learn More

Challenges in the Modern Software World

The declining insight and control people have over hardware and software systems

- As hardware and software systems grow increasingly vast in scale and complex in business logic, computing power and user numbers experience exponential growth. This often leads to difficulties in problem identification, resulting in performance bottlenecks, reliability issues, and security vulnerabilities.
- Most problems occur exclusively in production environments and are difficult to reproduce. Taking systems offline for inspection seriously impacts normal operations and degrades the user experience.

The Challenges of Embracing Cloud Native

As container technologies and microservice architectures become widespread, enterprises face unprecedented complexity challenges

- Technology Stack Explosion: Containerized deployments have led to exponential growth in application numbers, with various technology stacks and distributions coexisting, dramatically increasing system complexity.
- Observability Gaps: Streamlined container images often lack necessary debugging tools, significantly increasing the difficulty of troubleshooting issues.
- Difficult Problem Traceability: While containers' rapid destruction and rebuilding mechanisms improve system resilience, they can also mask potential software defects, making root cause analysis increasingly challenging.

Industry Pain Points

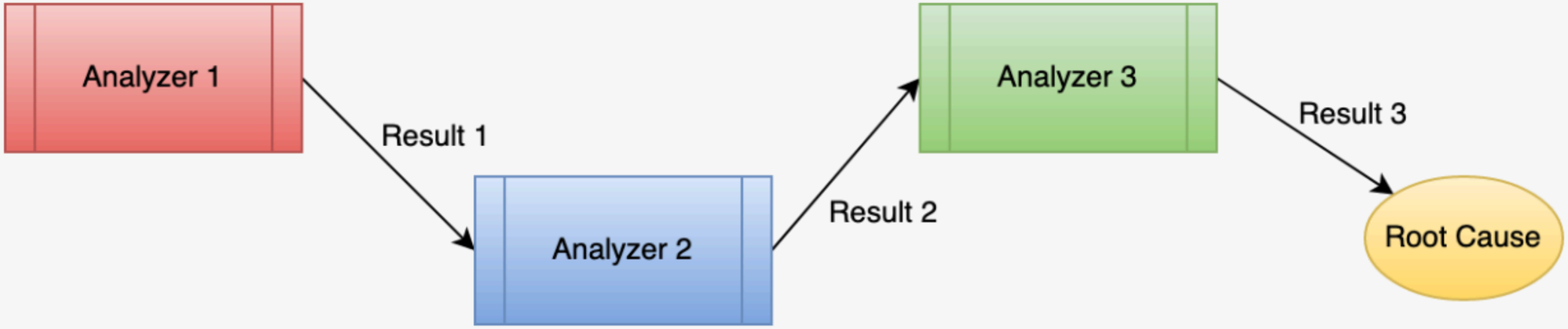
- Complexity Dilemma: Traditional APM tools require embedding probe code into applications, which not only increases system complexity but may also introduce new instability factors.
- Resource Waste: Extensive data collection, transmission, and storage lead to high costs and impact production system performance. Meanwhile, complex processing workflows often result in false positives and missed alerts.
- Insufficient Insights: Surface-level monitoring metrics only reflect symptoms of problems without analyzing root causes. The lack of deep understanding across the full technology stack means problem identification often remains superficial.
- Delayed Response: The process from anomaly detection to problem confirmation to solution implementation is time-consuming and cumbersome, making it difficult to meet rapidly changing business requirements.

What is OpenResty XRay

- OpenResty XRay is a next-generation non-intrusive dynamic tracing product, specifically designed to address the increasingly complex operational challenges faced by enterprises.
  - It provides 24/7 real-time system monitoring, providing in-depth analysis of cloud and server applications without requiring any code modifications.
  - The system automatically processes all diagnostic data, generates professional analysis reports, and provides precise optimization recommendations, transforming complex diagnostics into actionable insights.

OpenResty XRay

Automated Sampling in Unattended Mode



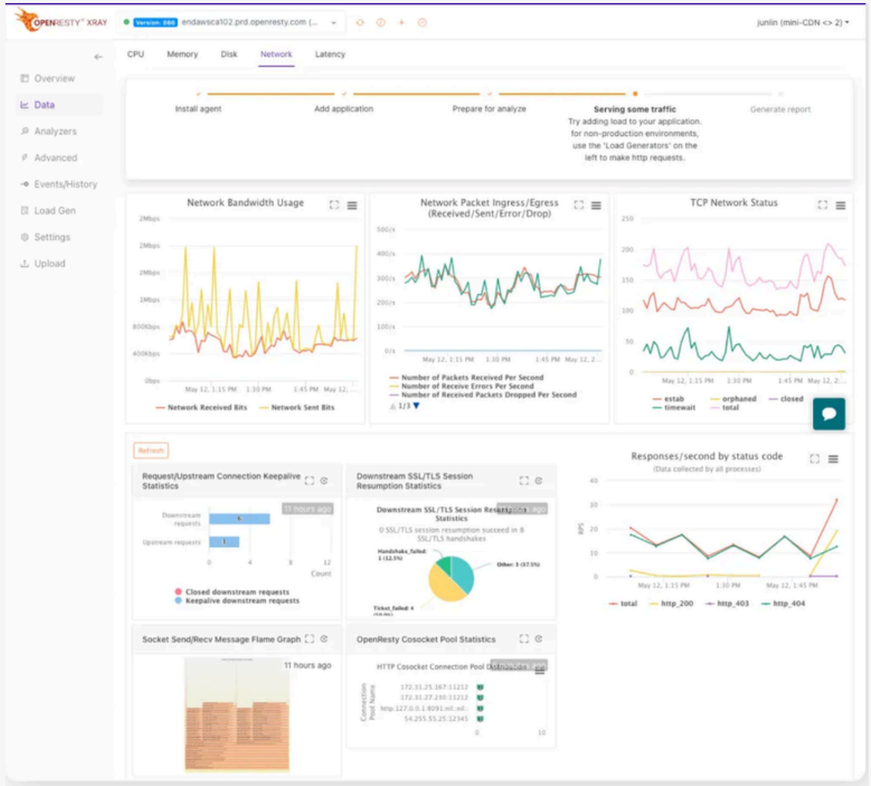
Scheduled sampling → **Event-driven monitoring** → Chain of Reasoning

Detection of changes in CPU, memory, I/O and errors/exceptions

Real-time Monitoring of System Performance Metrics

Visualization dashboard providing a comprehensive view of the system operational status

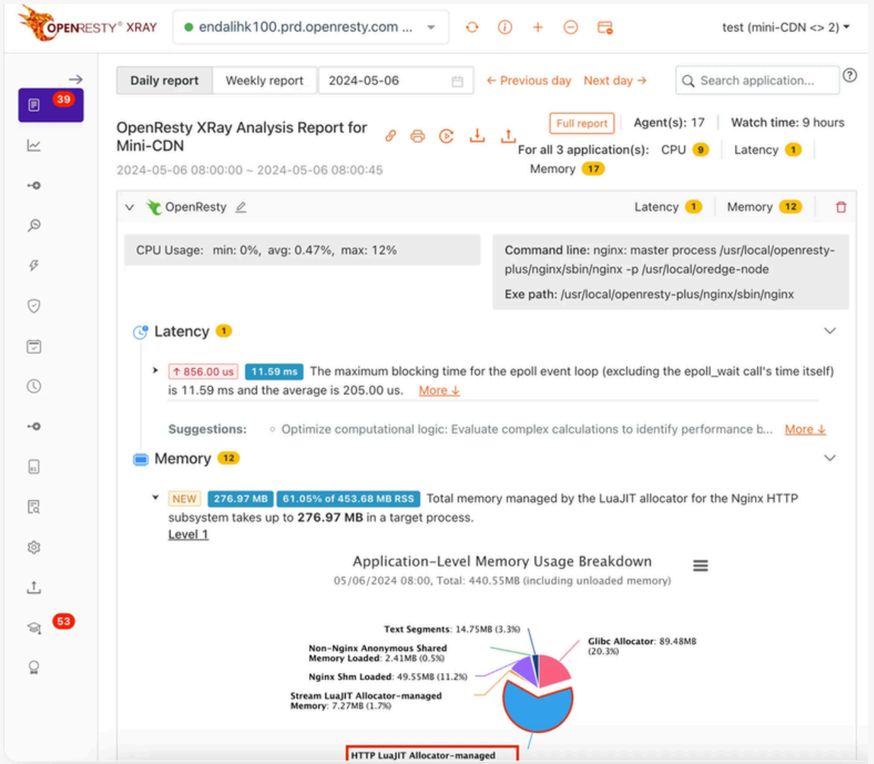
- Sampling results of various system metrics consolidated in real-time dashboards.
- Track application performance and analyze trends.
- Automatic analysis and identification of system performance issues across different applications and language levels.



Fully Automated Analysis Reports

Beyond tracking and analysis - Deep insights from sampling data

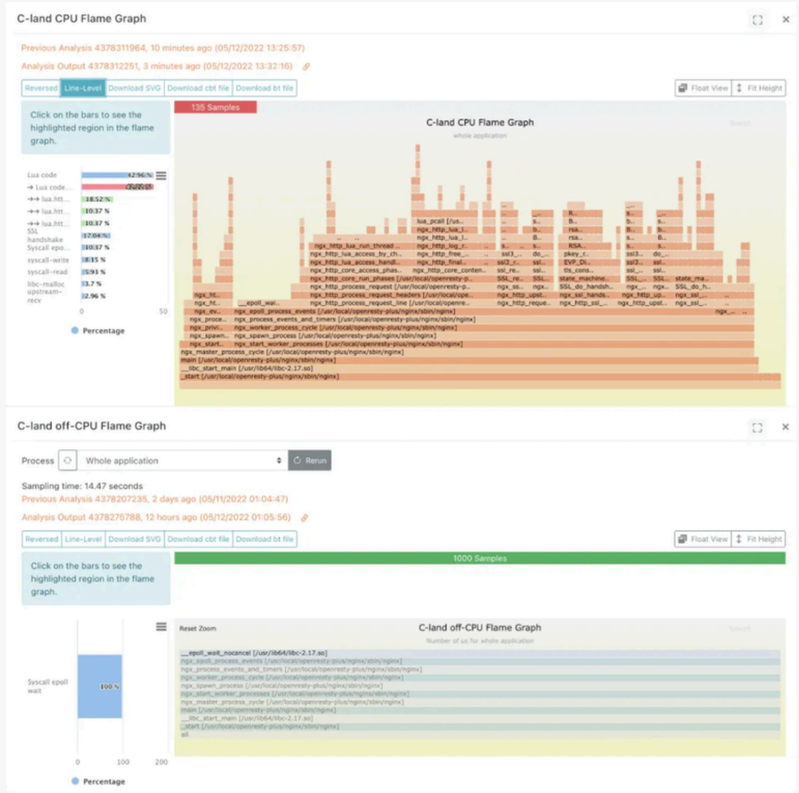
- Professional root cause analysis and optimization recommendations for performance bottlenecks and resource consumption issues.
- OpenResty's expert team helps you thoroughly analyze and precisely identify the root causes of production issues, making complex problem diagnosis simple and efficient.



Guided Analysis

Active Analysis | Real-time Debugging

- An interactive troubleshooting platform that enables real-time observation and analysis of production applications.
- Our solution empowers you to validate optimization strategies instantly, perform dynamic debugging and performance tuning, and implement changes with immediate feedback—all without system restarts.



Our Competitive Edge

Why Choose Us

1.Next-generation dynamic tracing technology - 100% non-intrusive

2.Innovative automatic sampling method - Efficient, lightweight, low overhead

3.Full-stack, comprehensive analysis covering all types of system performance issues

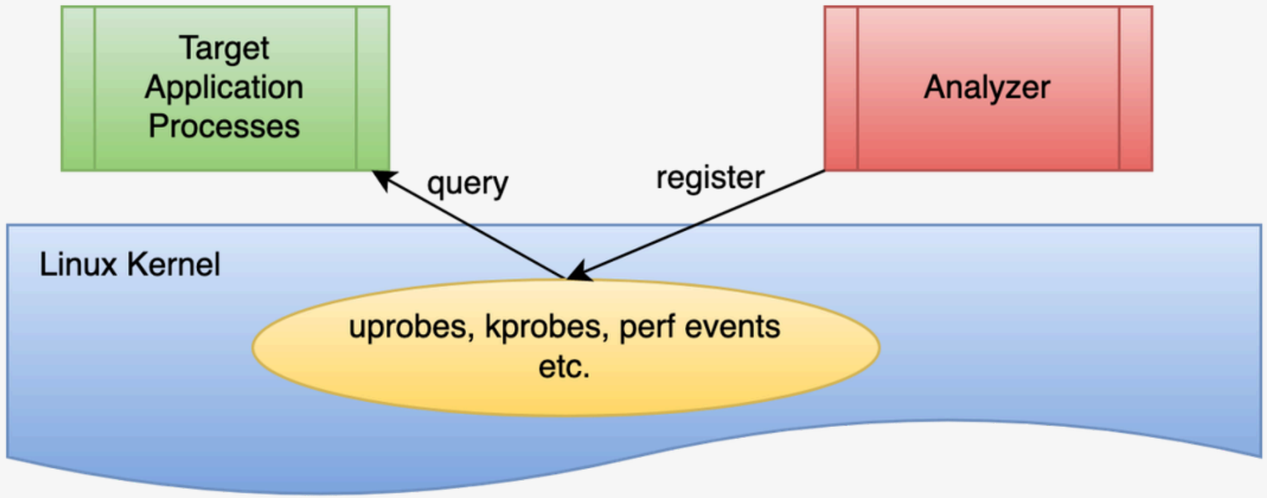
4.Ability to analyze applications without debug symbols

5.Container transparency capabilities

6.Automatic security issue analysis

Next-Generation Dynamic Tracing Technology

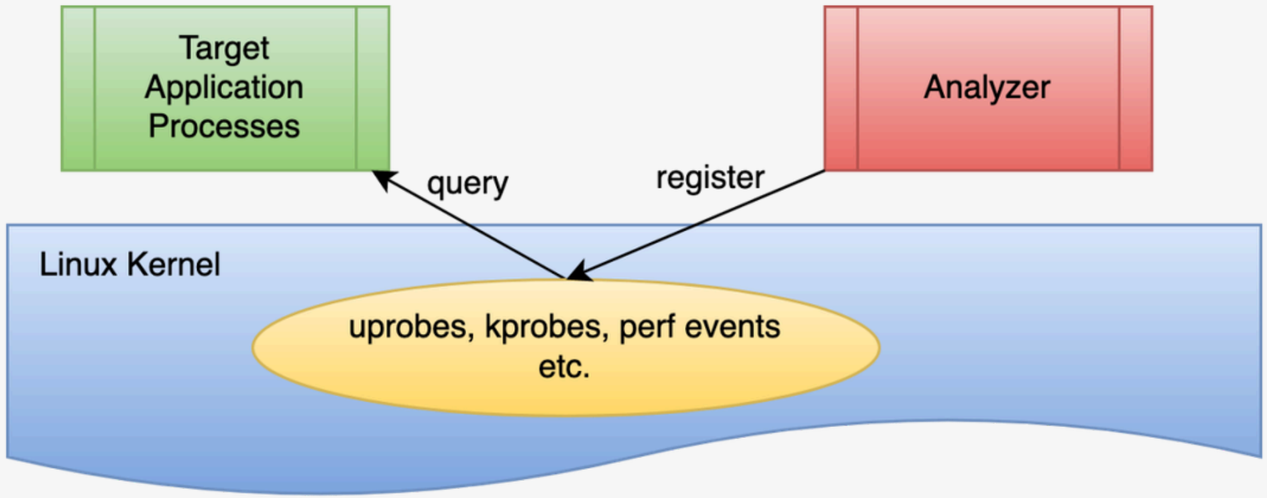
100% Non-intrusive



- Hot-pluggable: No need to modify applications, install new plugins, patches, or modules, restart applications, use special startup or compilation options, rebuild applications or packages, or inject any code into the target process space.

Next-Generation Dynamic Tracing Technology

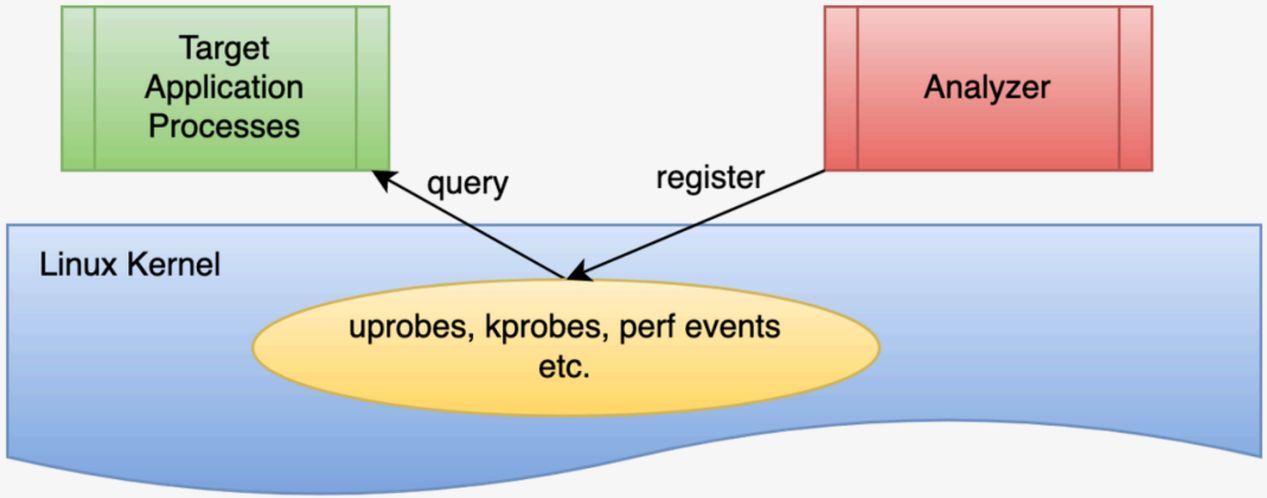
100% Non-intrusive



- Treats running processes and containers as read-only databases: extracting necessary information to solve performance issues, anomalies, errors, and security vulnerabilities.

Next-Generation Dynamic Tracing Technology

100% Non-intrusive



- Provides deep insights into online or offline software systems: without accessing user source code, identifying source code-level issues, and precisely locating the source code line where system problems occur.

Innovative Automatic Sampling Method

Efficient, Lightweight, Low Overhead

1. Precise targeted tracing: Only traces relevant code paths and functions
2. On-demand sampling: Collects significantly less information than traditional APM products, enabling aggregation and summarization at the data source
3. Zero performance overhead when not sampling, with overhead during sampling typically too low to measure

Full-Stack, Comprehensive Analysis

Addressing All System Performance Issues

- Coverage Across Software Layers:

- Application Programming Languages (Lua/Python/PHP/Perl/Go/etc.)
- Systems Programming Languages (C/C++/Rust)
- Operating System Kernel (Network Protocol Stack/Process Scheduler/Memory Management/System Calls)

- Thread Blocking and Locking Issues
- CPU Analysis:

- High CPU Utilization or Blocking
- CPU Time Distribution Across Different Code Paths
- Common CPU Bottlenecks

- Memory Management:

- Quantitative Analysis of Memory Distribution
- Memory Allocator Usage Patterns
- Memory Leaks, Fragmentation, and Delayed Release

Full-Stack, Comprehensive Analysis

Addressing All System Performance Issues

- Disk I/O Performance Analysis
- Latency Optimization:

- Long Request Latency and Tail Requests
- Breakdown of Delays Across Application Operations and Processing Stages
- Latency Statistics for Asynchronous Non-Blocking I/O

- Intelligent Packet Capture and Exception Detection:

- Targeted Capture of Network Traffic on Connections with Anomalies

- Core Dump Analysis and Process Crash Resolution

Analyze Applications Without Debug Symbols

OpenResty XRay Overcomes the Challenge of Missing Debug Symbols in Production Environments

1. OpenResty XRay has built an industry-leading central symbol database holding petabyte-scale debug information for public packages, and it continues to grow.
2. Through proprietary machine learning algorithms, OpenResty XRay can automatically reconstruct symbols even when facing binary programs completely lacking debugging information.
3. As long as debug symbols have been indexed by the OpenResty XRay central package database, target machines don't need to install or store debug symbols. Customers can obtain complete program runtime information without deploying large debug files in production environments, truly achieving the perfect combination of lightweight deployment and deep analysis.

OpenResty XRay Container Observability

OpenResty XRay has achieved a breakthrough in seamlessly penetrating container boundaries. With just a single Agent deployed on the host machine, it can automatically discover and analyze applications running in Docker containers and Kubernetes Pods on the same host.

1. No container image modification required
2. Zero-configuration cross-container monitoring
3. No impact on container resource limits or security policies
4. Automatic discovery in dynamic scaling scenarios

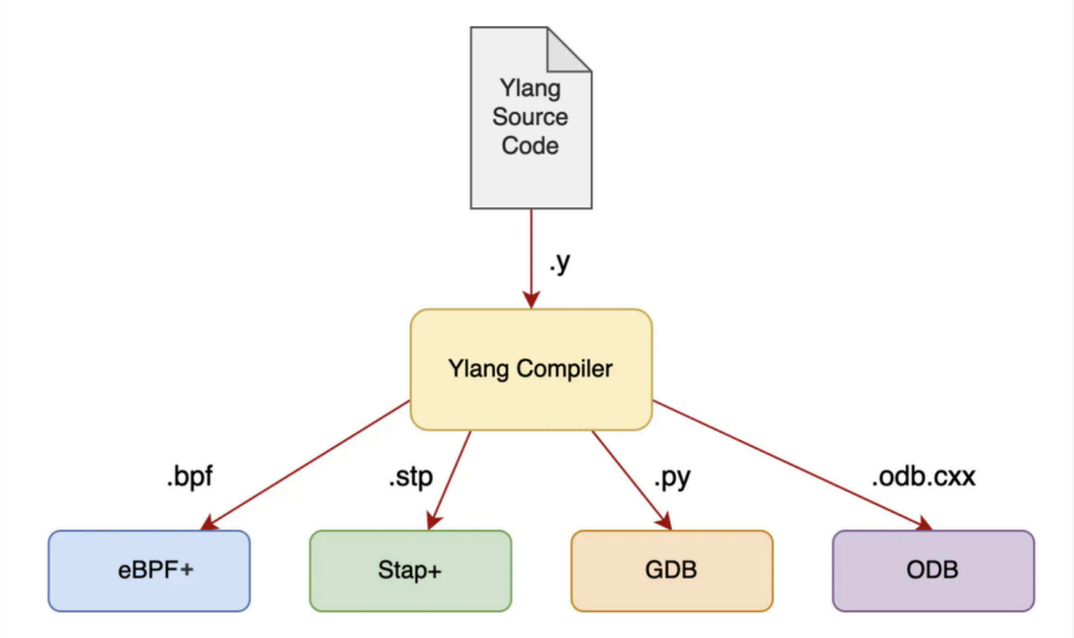
Automated Security Risk Detection

OpenResty XRay provides comprehensive security vulnerability scanning capabilities

1. Real-time automatic identification of transport layer security risks, including unencrypted communications, missing TLS certificate validation, and outdated encryption protocols.
2. Deep scanning and analysis of code execution behavior to locate potential remote shell command injection vulnerabilities.

OpenResty XRay Implementation

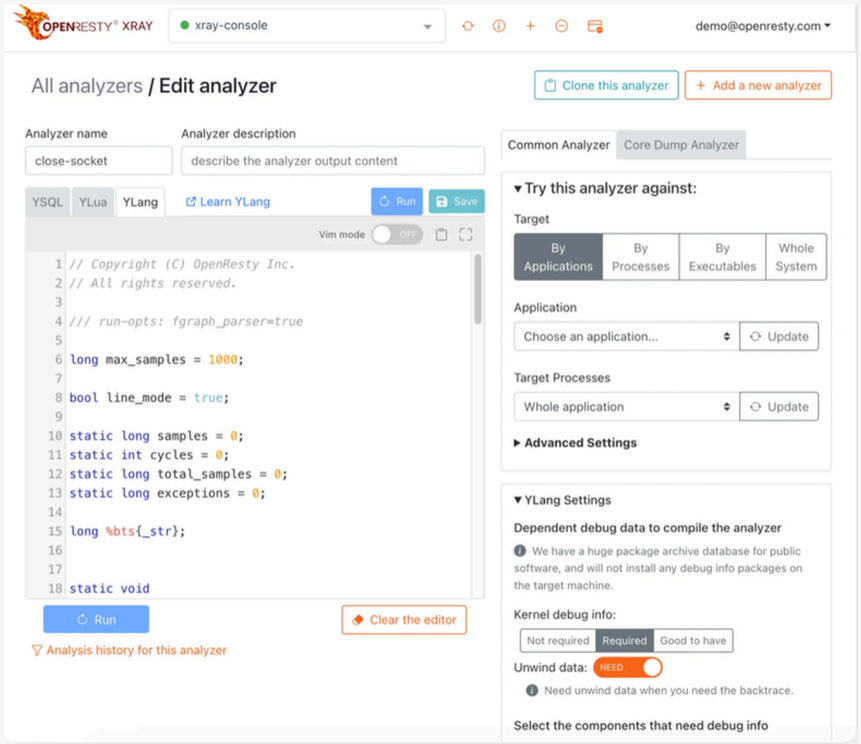
1. At the core implementation level, OpenResty XRay is powered by the Y language developed by OpenResty Inc.
2. Supports multiple debugging frameworks such as Stap+, eBPF+, GDB, and ODB.
3. Advanced toolkits (ylua, ysql, y++) enhance XRay's programmability and extensibility, enabling cost-effective and efficient troubleshooting of complex system issues.



Write Once, Run Anywhere Philosophy

Hundreds of Standard Analyzers

1. Comprehensive Coverage - Capable of analyzing and diagnosing various types of performance issues and anomalies
2. Precise Identification - Quickly pinpoint specific root causes through specialized analyzers
3. High Degree of Automation - Complete complex analyses without human intervention
4. Time-Saving - Reduce problem diagnosis from potentially days to minutes
5. Early Warning - Detect potential issues in advance, preventing problems before they occur
6. Customization Support - Our expert team can tailor analyzers to your specific requirements



Multilingual Performance Analyzer

OpenResty XRay supports mainstream Linux distributions and container deployment methods

1. Supports leading web servers and proxies including Nginx, Envoy, and Kong
2. Compatible with multiple programming languages and runtime environments: LuaJIT, Python, PHP, Go, Java, Erlang, Perl, Ruby, and Rust
3. Adapted for mainstream Linux distributions: Ubuntu, Debian, Fedora, Red Hat Enterprise Linux (RHEL), CentOS, openSUSE, AlmaLinux, Amazon Linux, Alibaba Cloud Linux, Rocky Linux, Tencent Linux, and Kylin
4. Supports container deployment platforms: Docker, Kubernetes

Future Support Plans

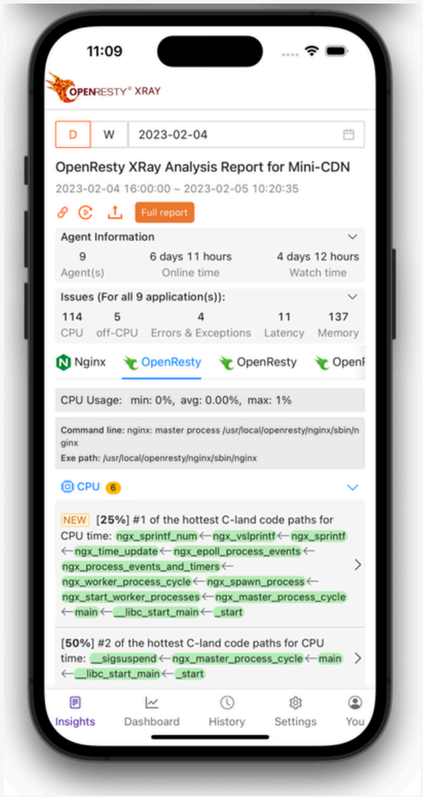
- nodejs
- pytorch、langchain AI stack, etc.

OpenResty XRay Mobile Application

Monitor your online applications anytime, anywhere

- Android / Google Play

- iOS / app store



Trusted by Customers

GitLab

Microsoft
TED
ADP

bilibili
ZOOM
IBM

EVERNOTE
vimeo

Resolving Bilibili’s major site incident with OpenResty XRay

https://blog.openresty.com/en/bilibili-xray-incident/

The Incident

- Bilibili's gateway system, developed with OpenResty, experienced a complete system outage
- All server OpenResty processes reached 100% CPU utilization, unable to process requests
- Both system restarts and code rollbacks proved ineffective

Troubleshooting with OpenResty XRay

1. Generated C-level CPU flame graphs using OpenResty XRay, identifying that the issue was concentrated in Lua code execution
2. Further analysis through Lua-level CPU flame graphs pinpointed the specific code path
3. Root cause discovery: a string-type zero value for a server weight (i.e., “0”) was inserted into the business logic’s configuration metadata, while the system expected a numerical weight value. It thus led to infinite recursion and infinite loops

Results

- XRay sampling analysis required only seconds to minutes to complete
- 100% non-intrusive dynamic tracing, without requiring any modifications to the target process
- Rapid and precise identification of the problematic code path

How we solved a CPU bottleneck caused by Lua exceptions in a custom Kong plugin

https://blog.openresty.com/en/lua-str-lower-excep/

The challenge

Even though the incoming API traffic was not very high, the customer's Kong server experienced abnormally high CPU usage (80%)

Investigation with OpenResty XRay

- Identified the string.lower function throwing exceptions in the hot code path
- Pinpointed the specific error: nil value incorrectly passed to string.lower in a custom auth plugin
- Located the exact source: .../kong/plugins/auth/handler.lua line 35

Solution

Fixed the bug in the auth plugin, ensuring nil values are not passed to string.lower

Results

- CPU usage reduced from 80% to 50%
- 37.5% performance improvement
- No complex code refactoring required
- Rapid problem resolution, saving the customer significant troubleshooting time

When a "Black Box" Plugin Consumes 45% CPU: How We Pinpointed Lua Line 93 Without Source Code

https://blog.openresty.com/en/xray-rsa-cpu/

A core business gateway cluster in the financial industry experienced continuous 100% CPU utilization during peak hours, with P99 latency (99th percentile latency) drastically fluctuating, leading to a severe decline in service quality.

Troubleshooting with OpenResty XRay

- Discovered a critical hotspot: The C function `pkey_rsa_decrypt` consumed up to 44.8% of CPU time.
- Root Cause Identification
- Precisely traced the issue back to the Lua source: line 93 of the `cb-session-validation.lua` plugin.
- Call Chain Visualization
- Comprehensively mapped the call path from `Lua plugin` -> `JWT library (Lua)` -> `libcrypto.so (C library)`.

Performance Optimization Results

- Nearly 45% of CPU capacity was freed up, almost doubling system throughput.
- CPU utilization no longer reached saturation during peak periods, and P99 latency stabilized.
- Directly obtained the complete call stack from Lua to C, pinpointing the root cause of the problem.
- Significantly shortened the troubleshooting time for cross-language performance issues, mitigating losses for the customer caused by performance bottlenecks.

Performance Breakthrough: 60% DNS Optimization Success

https://blog.openresty.com/en/xray-customer-casestudy-dns/

The customer's DNS service system was experiencing severe performance issues

- Uneven CPU load: Some Nginx worker processes exhibited high CPU utilization, while others remained relatively idle.
- Increased response latency: Overall system response was slow, particularly noticeable under high load conditions.

Troubleshooting with OpenResty XRay

- Uneven request distribution was identified: It was determined that all listening ports had not enabled the `reuseport` option, leading to an unbalanced distribution of requests across workers.
- Through C-level flame graph analysis, it was found that the `cjson` module (JSON parsing) consumed approximately 60% of the CPU time, representing the system's absolute bottleneck.
- The core business logic `dns_server.lua` utilized only about 5% of the CPU.
- `cosocket` receive operations consumed approximately 16% of the CPU time, another significant contributor to CPU consumption.

Optimization Results

- CPU consumption of the core bottleneck (JSON parsing) was reduced by over 60%.
- Configuration tuning alone (by enabling `reuseport`) led to an overall performance increase of 20-30%.
- System throughput saw significant improvement.
- By precisely identifying multiple performance problems, OpenResty XRay provided the customer with clear, multi-level optimization paths, saving significant analysis and trial-and-error time.

How OpenResty XRay Pinpointed Java Memory Issues

https://blog.openresty.com/en/xray-java-mem/

During performance regression testing of a customer's order system, heap memory continuously rose and could not be reclaimed even after multiple Full GCs, indicating a serious memory leak risk within the system.

Troubleshooting with OpenResty XRay

- Identifying Leak Root Causes: Using GC object reference analysis, quickly find unexpected retained objects and their complete reference chains.
- Diagnosing Excessive Object Allocation (Memory Churn): Through GC object allocation count analysis, discover code paths that frequently create short-lived objects.
- Locating Large Objects: Utilizing GC object allocation size analysis, precisely pinpoint the creation source of "large objects."

Performance Optimization Benefits

- Successfully resolved memory leak issues, ensuring system stability.
- No need to restart services or modify code, operating transparently with no impact on the production environment.
- Does not rely on JVM Safepoints, and the analysis process incurs no STW (Stop-the-World) pauses.
- Intuitive visualization charts (flame graphs, reference graphs) significantly lower the barrier to analyzing memory problems, enabling quick root cause identification.

Pinpoint the Hidden Cost of Gzip Configuration with OpenResty XRay

https://blog.openresty.com/en/xray-gzip-cpu/

The Challenge

In high-throughput, mission-critical financial environments, while key performance indicators (QPS, latency) appeared to meet targets, the substantial CPU overhead became a critical bottleneck for scaling throughput.

Troubleshooting with OpenResty XRay

Flame graphs clearly revealed that CPU resources were heavily consumed by compression and decompression processes, rather than core business logic.

- Gzip compression accounted for 54% of CPU time.
- Gzip decompression accounted for 61% of CPU time.
- Even Brotli compression consumed a significant 30.9% of CPU.
- OpenResty XRay enabled us to trace high-level CPU consumption directly to granular configuration parameters, such as gzip_level.

Results After Optimizing Gzip Configurations Based on OpenResty XRay's Data Insights

- Overall system CPU utilization was significantly reduced.
- CPU time for core business threads was freed up.
- QPS saw a substantial increase.
- Request latency dramatically improved.

Tracing the Slowest PCRE Regular Expressions in OpenResty or Nginx Processes

https://blog.openresty.com/en/ngx-slowest-pcre-regexes/

An OpenResty/Nginx Lua application was experiencing severe inefficiency, leading to server CPU usage consistently hitting 100%.

Troubleshooting with OpenResty XRay

- Preliminary Diagnosis: A C-level CPU flame graph revealed that the majority of CPU time was consumed by the execution of PCRE regular expression matching within the Lua code.
- In-depth Analysis: A Lua-level CPU flame graph further confirmed that the performance bottleneck was localized to the ngx.re.match API call within a single code path.
- Precise Identification: Utilizing the lj-slowest-ngx-re analyzer, the slowest regular expression was precisely pinpointed as (?:.*)*css. Its execution delay exceeded 20.6 milliseconds, primarily due to the greedy quantifier .* causing extensive backtracking.

Optimizing the Regular Expression to Prevent Extensive Engine Backtracking

- CPU Bottleneck Elimination: CPU is no longer consumed by Lua regular expression computations; the performance hotspot shifted to normal I/O write operations (writev system call).
- Significant Performance Improvement: The maximum delay of the slowest regular expression decreased from 20.6 milliseconds to 59 microseconds, resulting in a hundred-fold performance improvement.
- Accurate Problem Pinpointing: Eliminated the need for guesswork or manual, step-by-step troubleshooting, directly identifying the problematic regular expression.
- Rapid Resolution: Deep-seated performance issues were quickly identified and resolved without requiring modifications to complex business logic.

OpenResty XRay - Product Vision

Making Complex Systems Transparent and Controllable

- OpenResty XRay is dedicated to solving one of the most challenging problems in the modern computing world: transforming any black-box system into a white-box through zero-intrusion technology, enabling full-stack visibility from hardware to business layers. We provide our users with unprecedented system insight capabilities.
- We have developed the industry's most powerful automated problem diagnosis engine. Our lightweight Agent can comprehensively scan the entire software stack without requiring system modifications or process restarts, automatically identifying and locating various performance bottlenecks, security vulnerabilities, and stability issues.
- We are pioneering breakthrough real-time hot patching technology that will transform system maintenance. In the future, OpenResty XRay will not only discover problems but also automatically apply repair patches to running systems, making diagnosis and remediation simple and elegant.

Learn More

OpenResty XRay

Official Website：https://openresty.com/en/
Blog：https://blog.openresty.com/en/
X：https://x.com/OpenResty
Youtube：https://www.youtube.com/channel/UCXVmwF-UCScv2ftsGoMqxhwr