



OpenResty XRay

产品简介

提供世界一流的解决方案，即刻驾驭错综复杂的现代 Web 挑战

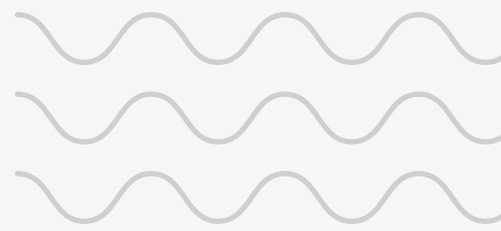
Powered by OpenResty Inc.

目录

1. 业界痛点
2. 产品简介
3. 核心优势与亮点
4. 底层实现
5. 客户案例
6. 产品愿景
7. 更多信息



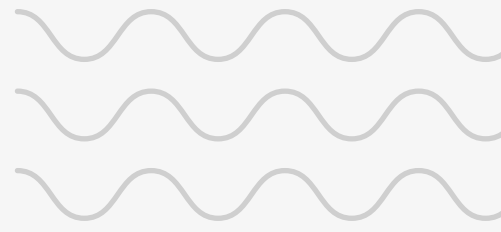
现代软件世界的挑战



人们对软硬件系统的洞察力和控制力逐渐降低

- 软硬件系统规模越来越庞大、业务逻辑越来越复杂，计算机算力和用户数量指数增长，往往导致问题定位难度上升，性能瓶颈、可靠性问题和安全漏洞频频出现
- 大部分问题线上独有，且难以复现，下线检查又严重影响软件系统的正常运行和用户体验

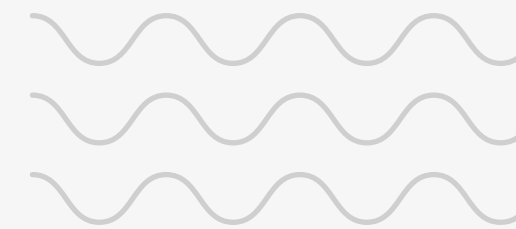
拥抱云原生的困扰



随着容器技术和微服务架构的普及，企业面临着前所未有的复杂性挑战

- 技术栈激增：容器化部署导致应用数量倍增，各类技术栈和发行版并存，系统复杂度指数级提升
- 可观测性缺失：精简容器镜像往往缺乏必要的调试工具，问题排查难度大幅提高
- 问题难追溯：容器的快速销毁与重建机制虽提高了系统韧性，却也容易掩盖潜在的软件缺陷，让根因分析变得更加困难

业界痛点



- 复杂性困境：传统 APM 工具需要在应用中植入探针代码，不仅增加了系统复杂度，还可能引入新的不稳定因素
- 资源浪费：大量的数据采集、传输和存储不仅带来高昂的成本，还会影响生产系统性能。同时，复杂的处理流程也容易出现误报和漏报
- 洞察力不足：表层的监控指标仅能反映问题现象，无法深入分析根本原因。缺乏对全技术栈的深度理解，使得问题定位往往停留在表面
- 响应滞后：从发现异常到确认问题再到实施解决方案，过程耗时冗长，难以满足快速变化的业务需求

OpenResty XRay



适用于开源软件和 Web 应用的故障排查和性能优化工具

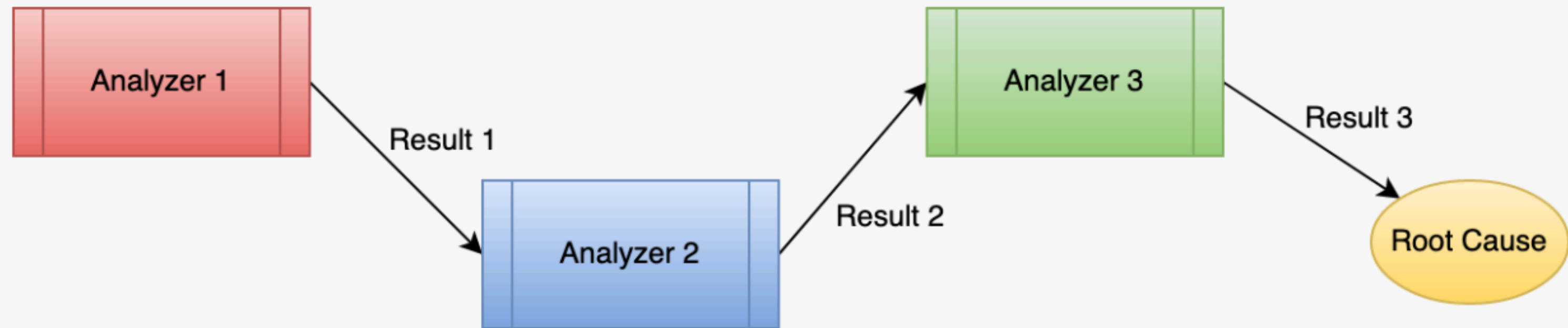
- OpenResty XRay 是新一代零侵入式动态追踪产品，专为解决企业日益复杂的运维挑战而生
 - 它能够全天候实时监测系统运行状态，无需修改任何代码，就能对云端及服务器应用进行深度分析
 - 系统自动处理所有诊断数据，生成专业的分析报告，并提供精准的优化建议，让复杂的问题诊断变得简单直观



OpenResty XRay



全自动采样 无人值守模式



定时采样



事件驱动

CPU 变化, 内存变化, IO 变化, 异常错误

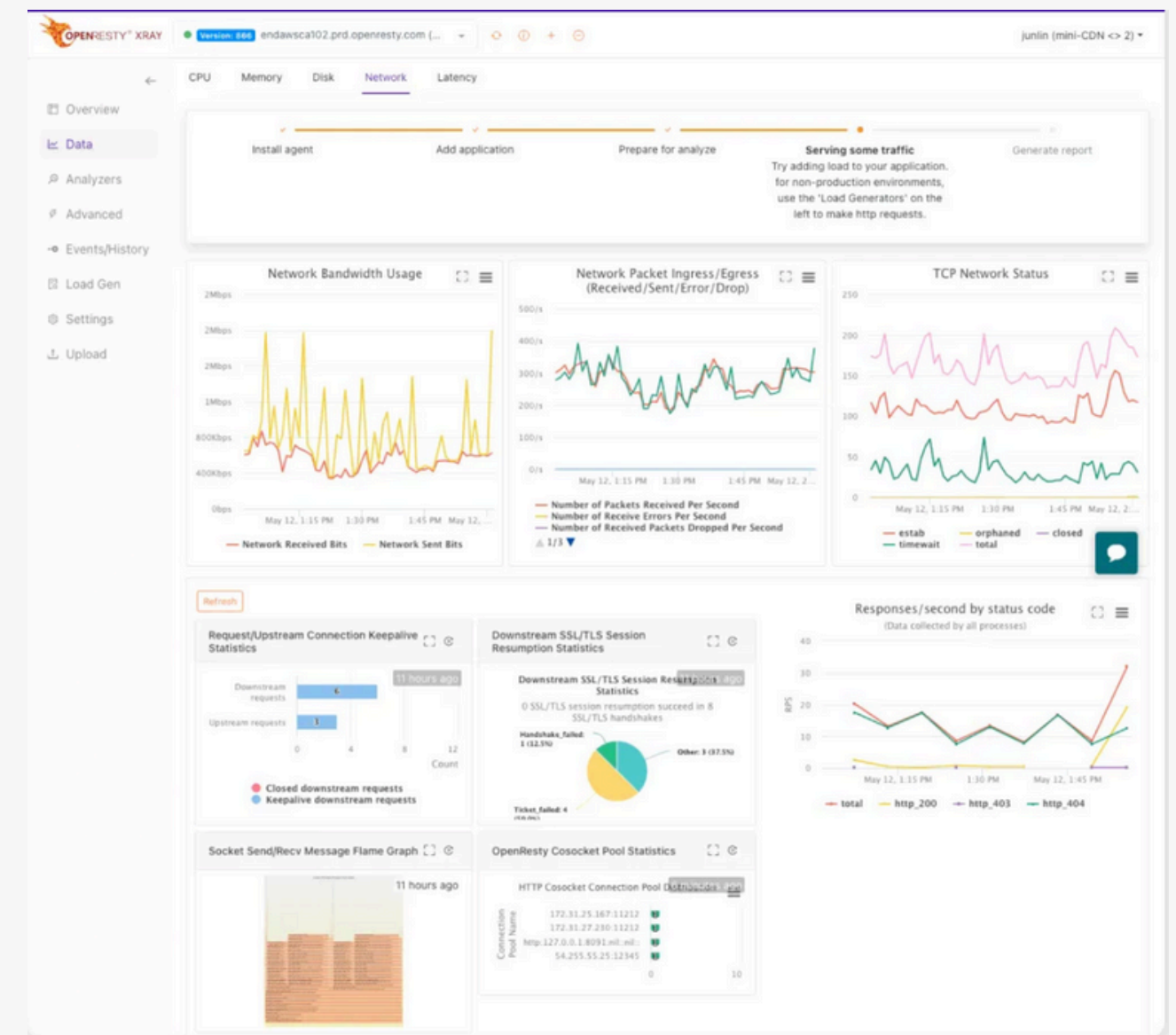


链式推理

实时监控各类系统性能指标

可视化监控面板 全方位展现系统运行状态

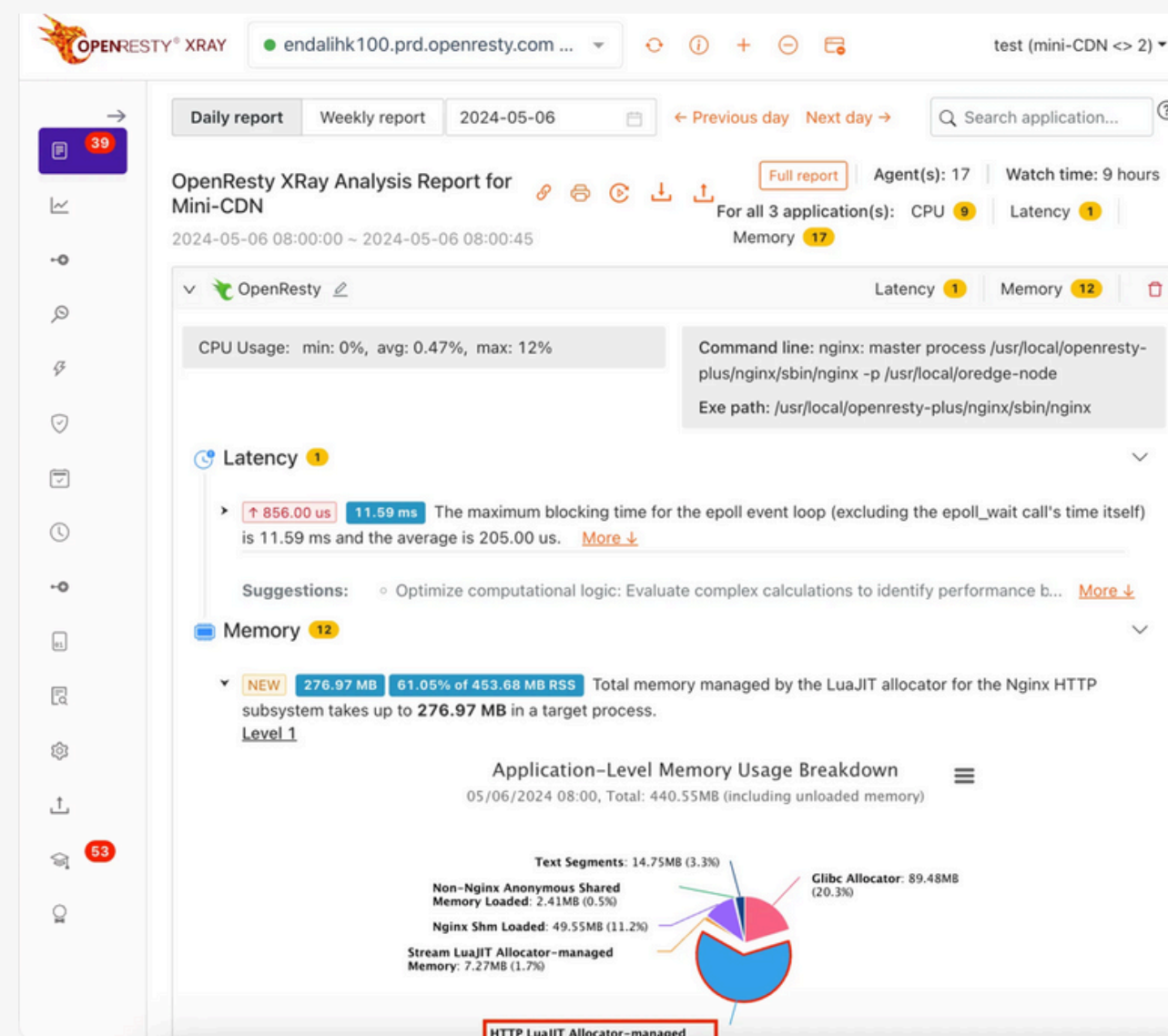
- 各类系统指标的采样结果将汇总在可视化仪表盘，实时掌握应用运行状况与性能的变化趋势
- 可以自动分析和定位不同应用中、不同的语言级别的各类系统性能问题



全自动分析报告

不仅仅是追踪与分析 深度分析解采样数据

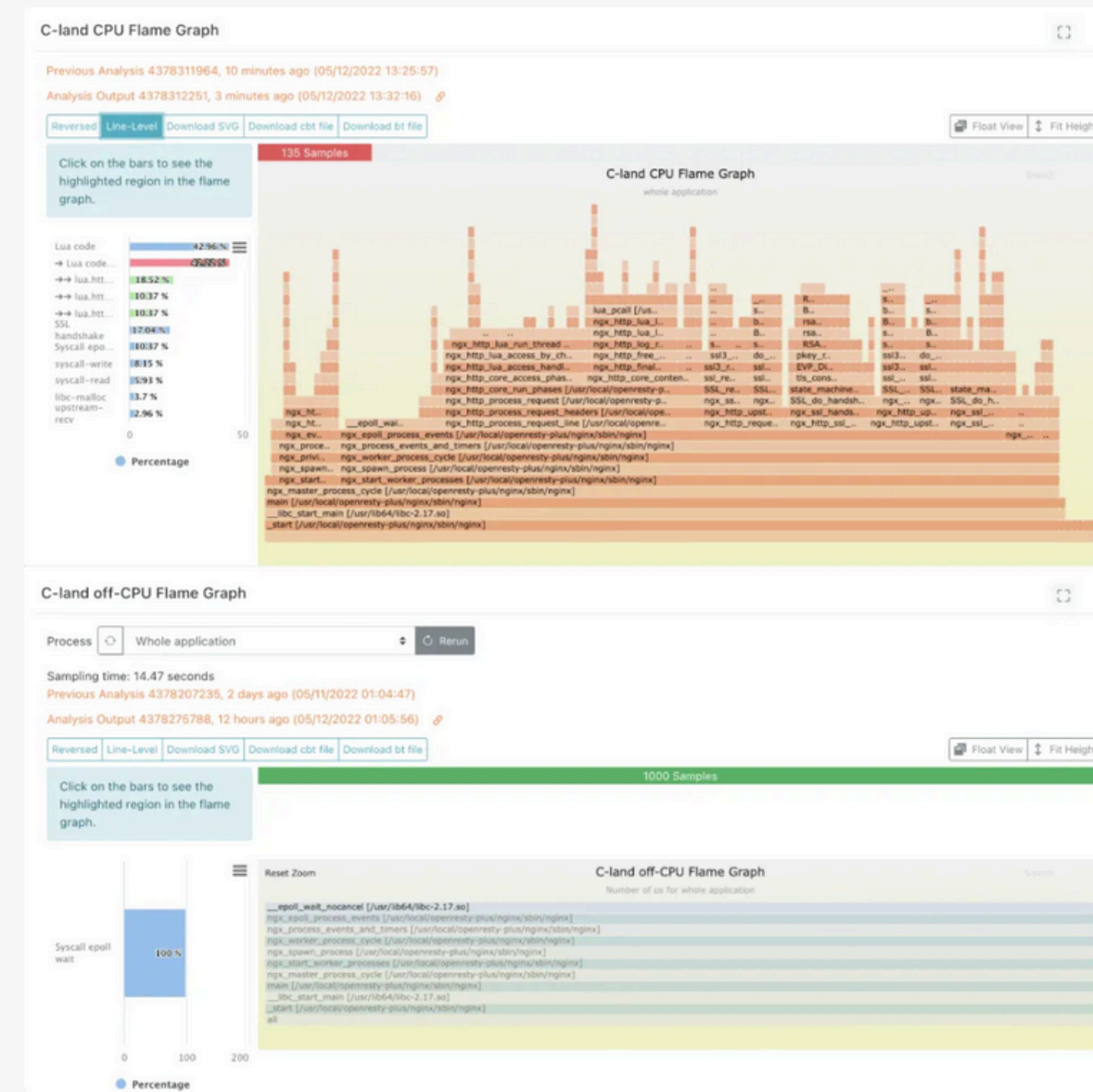
- 针对性能瓶颈、资源消耗等关键问题，XRay 将提供专业的根因分析和优化建议
- OpenResty 的专家团队会帮助您深入分析，精准定位线上问题的根因，让复杂问题诊断变得简单高效



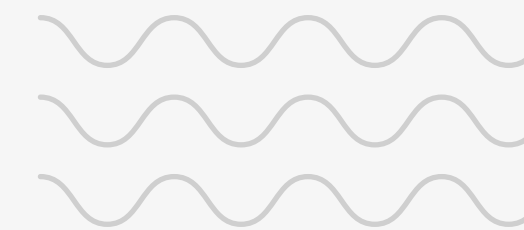
引导式分析

主动分析 实时调试

- 交互式故障排查工具，支持对正在运行的应用进行实时观测和分析
- 通过精准引导，帮助您快速验证优化方案，实现动态调试和性能调优，无需重启即可查看修改效果



产品核心优势

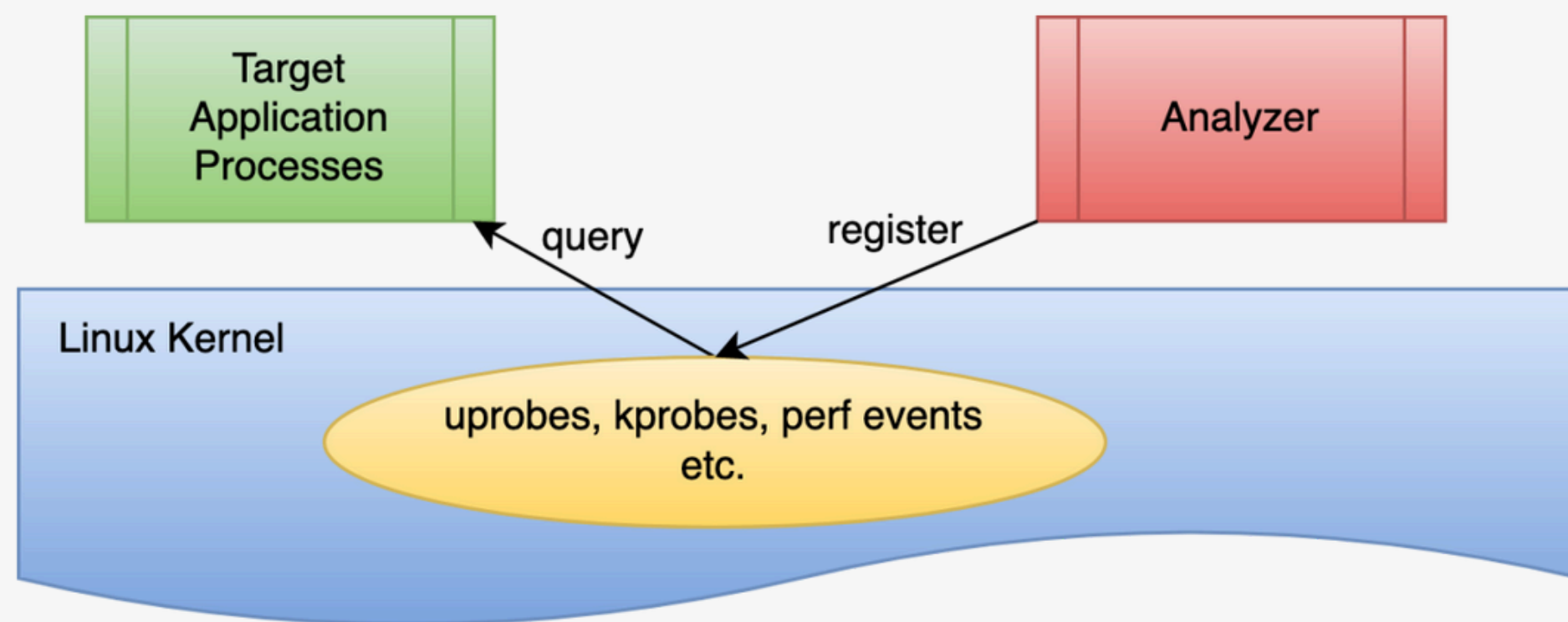


为什么选择我们

1. 新一代动态追踪技术驱动 100% 非侵入
2. 创新的自动采样方法 高效轻量低开销
3. 全栈、全方位无死角分析 涵盖各类系统性能问题
4. 无需调试符号也能分析应用
5. 具备容器透明化能力
6. 自动分析安全问题

新一代动态追踪技术驱动 100% 非侵入

动态追踪技术的优势



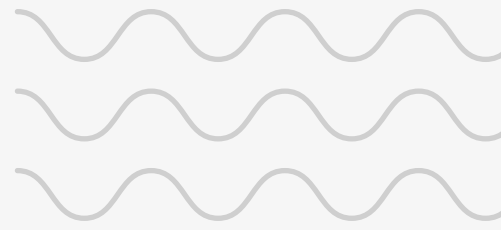
- 热插拔，无需修改应用，无需安装新插件、补丁、模块，无需重启应用，无需特殊启动或编译选项，无需重建应用或软件包，无需任何代码注入目标进程空间
- 运行中的进程和容器视为只读数据库，并提取必要的信息来解决性能问题、异常、错误和安全漏洞
- 能在不访问用户源代码的情况下，深入洞察在线或离线软件系统，找出源码层面的问题，准确定位到系统问题发生的源码行

创新的自动采样方法 高效轻量低开销

动态追踪技术的优势

1. 精准定向追踪，只追踪与之相关的代码路径和函数
2. 按需采样，信息采集量比传统 APM 产品少很多，可以在数据源进行聚合汇总
3. 未采样时的性能开销严格为 0，采样期间开销通常也低至无法测量

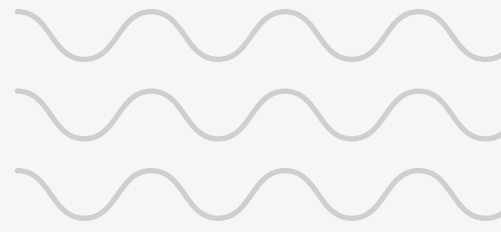
全栈、全方位无死角分析



涵盖各类系统性能问题

1. 覆盖不同软件层面的代码路径：业务编程语言层面（Lua/Python/PHP/Perl/Go/等），系统编程语言层面（C/C++/Rust），操作系统内核层面（网络协议栈/进程调度器/内存管理/系统调用）
2. 线程阻塞和锁
3. CPU 问题分析：CPU 使用率过高或阻塞、不同代码上 CPU 时间分布、常见的 CPU 瓶颈
4. 内存占用大或内存泄漏：定量分析内存分布、内存分配器的使用情况，内存泄漏、内存碎片、延迟释放等
5. Disk I/O 性能分析
6. 请求长延时、长尾请求：分解不同应用操作、处理阶段的延时，异步非阻塞 IO 的延时统计
7. 智能抓包和异常捕捉：只抓实际有异常的网络连接上的包
8. Core Dump 文件及进程崩溃

无需调试符号也能分析应用



OpenResty XRay 突破了生产环境调试符号缺失的难题

1. OpenResty XRay 建立了业界领先的中央符号数据库，已积累数百 TB 的公开包调试信息，并仍在快速扩充
2. 借助自主研发的机器学习算法，即使面对完全没有调试信息的二进制程序，OpenResty XRay 也能实现符号的自动重建
3. 只要调试符号曾被 OpenResty XRay 中央包数据库索引，目标机器无需安装或保存调试符号。客户无需在生产环境部署庞大的调试文件，就能获得完整的程序运行时信息，真正实现了轻量化部署与深度分析的完美结合

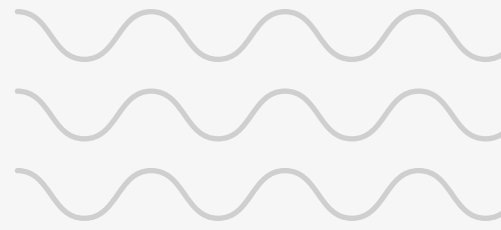
OpenResty XRay 容器观测能力



OpenResty XRay 突破性地实现了对容器边界的无缝穿透，只需在宿主机部署一个 Agent，就能自动发现并采样分析同一主机上其他的 Docker 容器和 Kubernetes Pod 中运行的应用

1. 无需修改容器镜像
2. 零配置即可实现跨容器监控
3. 不影响容器的资源限制和安全策略
4. 支持动态扩缩容场景下的自动发现

自动化安全风险检测

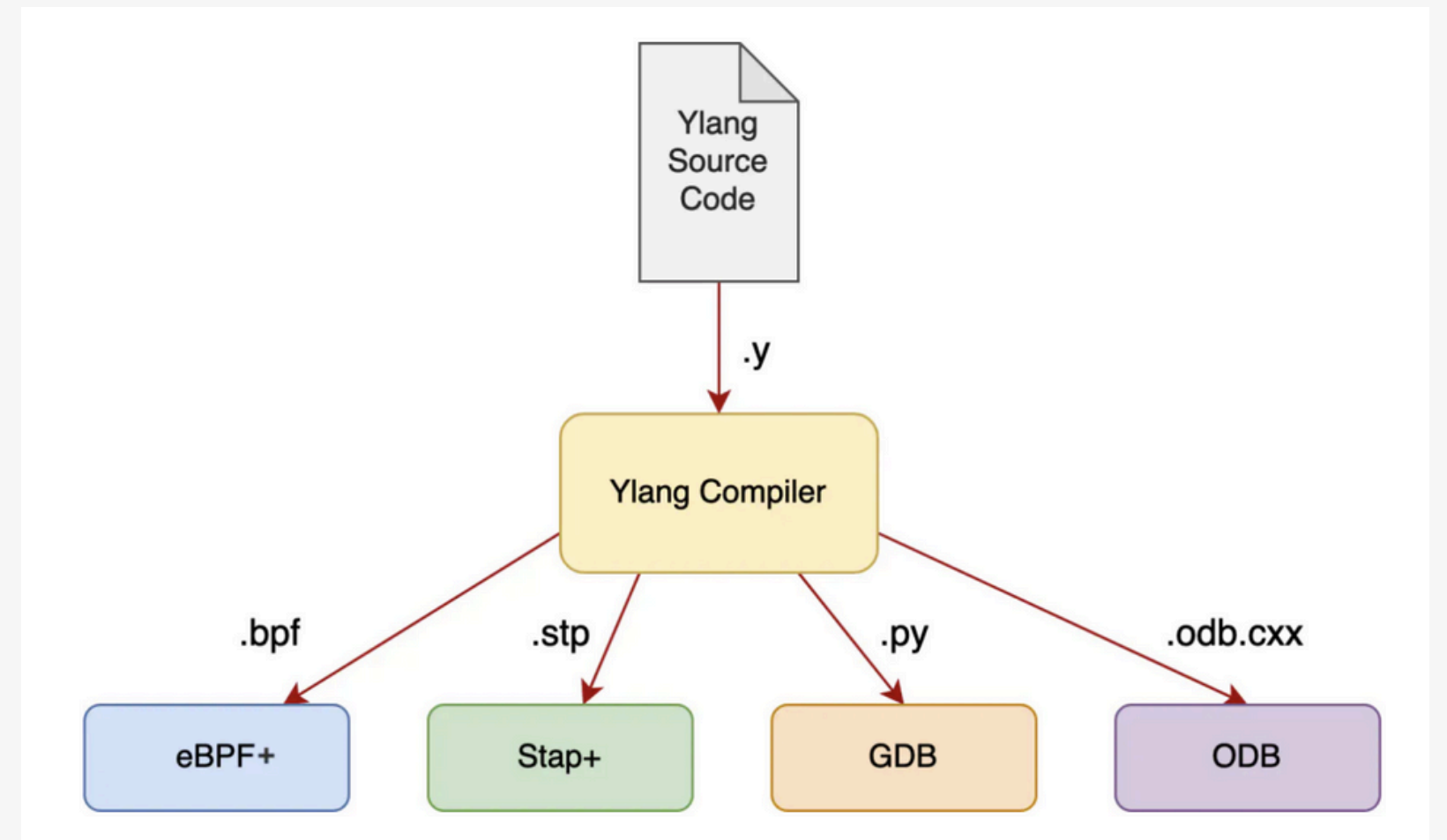


OpenResty XRay 提供全方位的安全漏洞扫描能力

1. 系统实时自动识别传输层安全风险，包括未加密通信、TLS 证书校验缺失以及过时的加密协议
2. 深度扫描与分析代码执行行为，定位潜在的远程 shell 命令注入漏洞

OpenResty XRay 底层实现

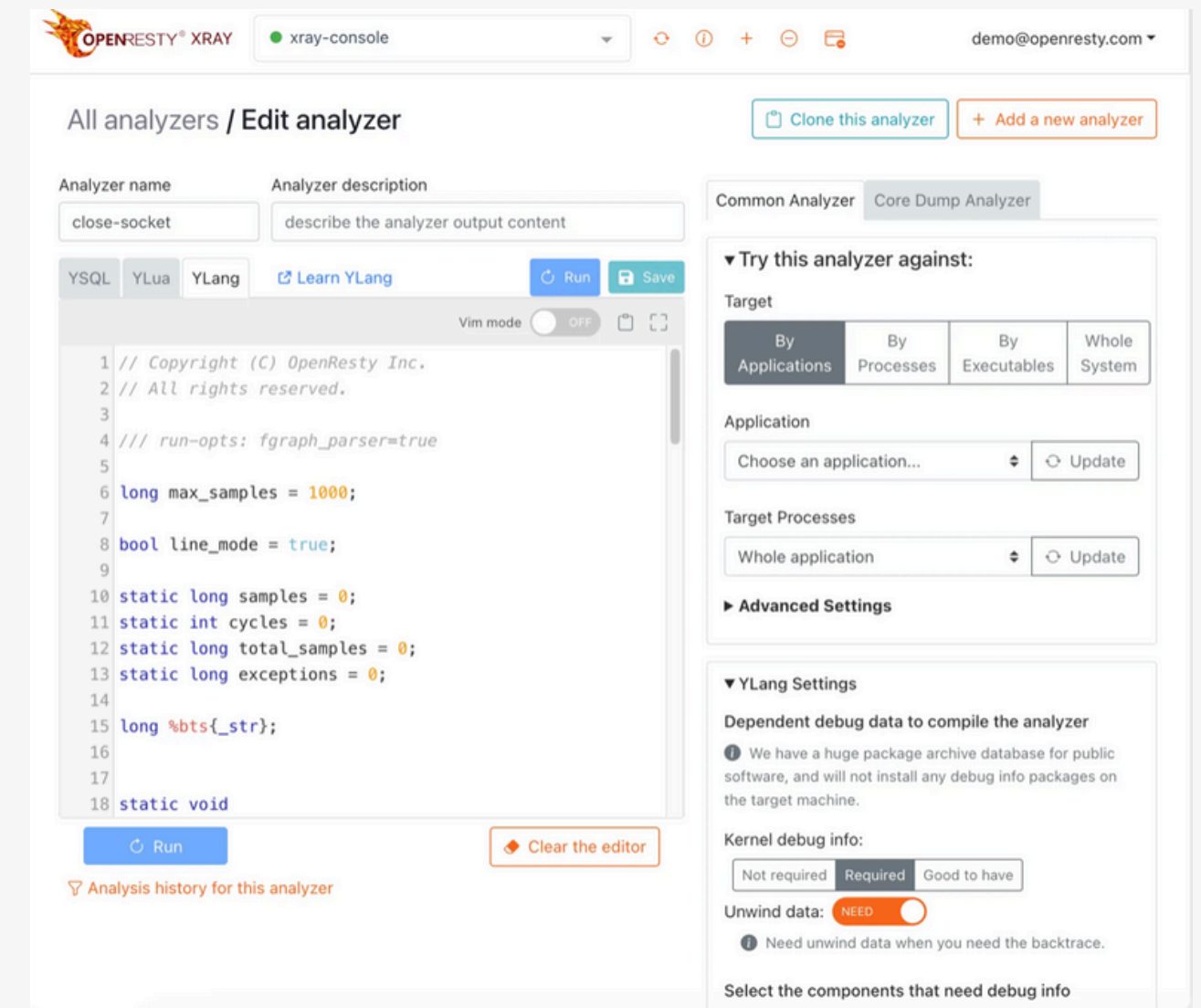
1. 在底层实现上，OpenResty XRay 由 OpenResty Inc. 开发的 Y 语言驱动
2. 支持多种不同的调试框架，如 Stap+, eBPF+, GDB和 ODB
3. ylua、ysql、y++ 等高级工具包，使 XRay 具有更高的可编程、可扩展能力。因此能以更低的开销成本、更灵活和高效地排查复杂的系统问题



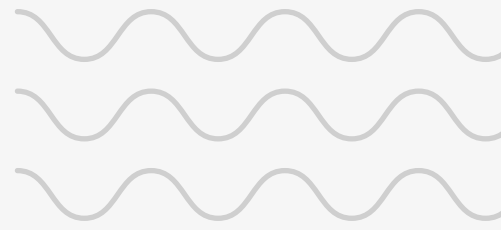
一次编写 到处运行

数百种标准分析器

1. 全面覆盖 - 能够分析和诊断各种不同类型的性能问题和异常情况
2. 精准定位 - 通过专门的分析器快速锁定具体问题根源
3. 自动化程度高 - 无需人工介入就能完成复杂的分析工作
4. 节省时间 - 问题诊断从可能需要数天缩短到分钟级
5. 提前预警 - 能够及早发现潜在问题，防患于未然
6. 支持自定义 - 专家团队能够针对您的需求，定制分析器



多语言性能分析器



OpenResty XRay 支持主流 Linux 发行版和容器部署方式

1. 支持 Nginx、Envoy 和 Kong 等主流 Web 服务器和代理
2. 兼容 LuaJIT、Python、PHP、Go、Java、Erlang、Perl、Ruby 以及 Rust 等多种编程语言和运行时环境
3. 适配主流 Linux 发行版：涵盖 Ubuntu、Debian、Fedora、Red Hat Enterprise Linux (RHEL)、CentOS、openSUSE、AlmaLinux、Amazon Linux、阿里云 Linux、Rocky Linux、腾讯 Linux 以及麒麟等
4. 支持容器部署方式：Docker、Kubernetes

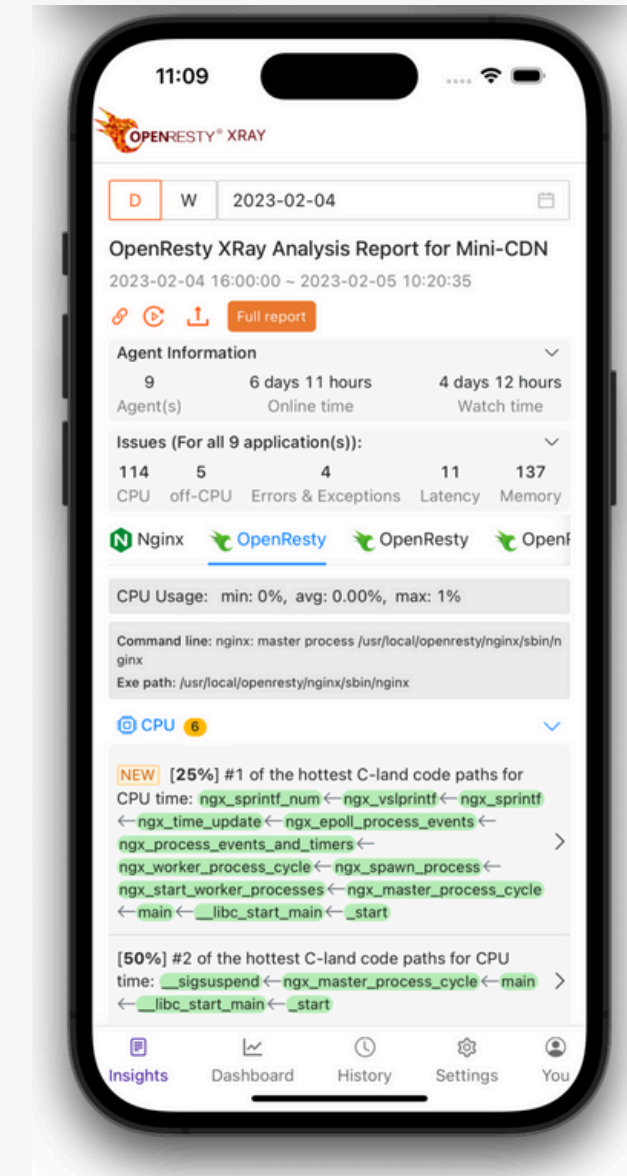
未来支持计划

- nodejs
- pytorch、langchain AI 软件栈等

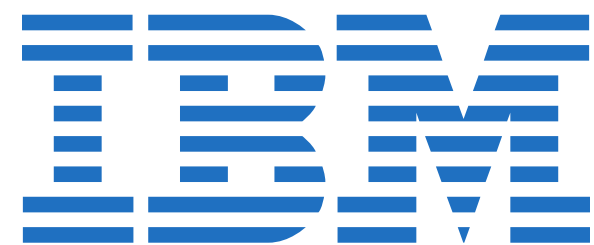
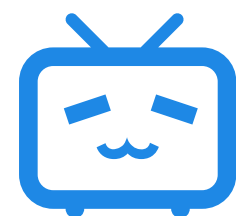
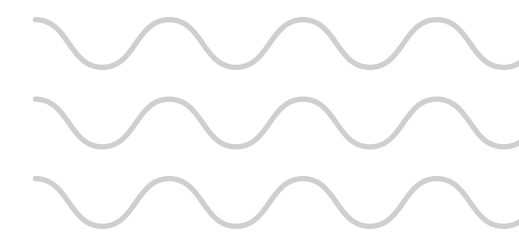
OpenResty XRay 移动端应用

随时随地 察看您的在线应用

- Android / Google Play
- iOS / 苹果商店



为众多企业客户所信任



OpenResty XRay 分析和解决 B 站重大线上事故

用户案例：<https://blog.openresty.com.cn/cn/bilibili-xray-incident/>

长期战略合作伙伴



Bilibili

线上服务不可用重大事故

事故

- B 站基于 OpenResty 开发的网关系统出现全线故障
- 所有服务器 OpenResty 进程 CPU 100%，无法处理请求
- 重启和代码回滚都无效

使用 OpenResty XRay 排查

1. 使用 OpenResty XRay 生成 C 语言级 CPU 火焰图，定位到问题集中在 Lua 代码执行
2. 进一步通过 Lua 级 CPU 火焰图，锁定具体代码路径
3. 最终发现根因：业务配置中写入了字符串 "0" 作为权重值，而系统期望是数值类型的 0，导致无限递归

使用成果

- XRay 采样分析仅花费几十秒到几分钟
- 100% 非侵入式动态追踪，无需修改目标进程
- 快速精准定位到问题代码路径

解决自定义 Kong 插件中的 Lua 异常所引起的 CPU 瓶颈

用户案例 : <https://blog.openresty.com.cn/cn/lua-str-lower-excep/?q=kong>

长期战略合作伙伴



Kong

CPU 性能瓶颈

问题

即使在流量不大的情况下，客户的 Kong 服务器 CPU 使用率异常高(80%)

使用 OpenResty XRay 排查

- 发现热点代码路径中 `string.lower` 函数抛出异常
- 定位到具体错误: 在自定义 `auth` 插件中错误地向 `string.lower` 传入了 `nil` 值
- 精确到源文件位置: `.../kong/plugins/auth/handler.lua` 第 35 行

解决方案

修复 `auth` 插件中的 bug，确保不向 `string.lower` 传入 `nil` 值

使用成果

- CPU 使用率从 80% 降至 50%
- 性能提升 37.5%
- 无需复杂的代码重构
- 快速解决问题，为客户节省大量排查时间

OpenResty XRay 如何洞穿 Lua 与 C 的边界，找回 45% 的 CPU 算力

用户案例：<https://blog.openresty.com.cn/cn/xray-rsa-cpu/>



问题

某金融行业核心业务网关集群高峰时段 CPU 使用率持续 100% 饱和，P99 延迟（99%分位延迟）急剧抖动，服务质量严重下降

使用 OpenResty XRay 排查

- 发现惊人热点：C 函数 `pkey_rsa_decrypt` 消耗了高达 44.8% 的 CPU 时间
- 根因定位：精准追溯到 Lua 肇事者：`cb-session-validation.lua` 插件的第 93 行
- 调用链还原：完整绘制了从 `lua 插件` -> `JWT 库 (Lua)` -> `libcrypto.so (C 库)` 的调用路径

性能优化效果

- 释放了近 45% 的 CPU 算力，系统吞吐能力几乎翻倍
- 高峰期 CPU 不再触顶，P99 延迟恢复平稳
- 直接获得从 Lua 到 C 的完整调用栈，锁定问题根源
- 极大缩短了跨语言边界性能问题的排查时间，为客户挽回了因性能瓶颈造成的损失

OpenResty XRay 实现 DNS 服务性能 60% 优化的案例解析



用户案例 : <https://blog.openresty.com.cn/cn/xray-customer-casestudy-dns/>

问题

客户的 DNS 服务系统面临严重的性能问题

- CPU 负载不均: 部分 Nginx worker 进程 CPU 占用率过高, 而其他进程相对空闲。
- 响应延迟增加: 系统整体响应慢, 在高负载情况下尤其明显。

使用 OpenResty XRay 排查

- 发现请求分配不均: 定位到所有监听端口均未启用 `reuseport` 选项, 导致 worker 间请求分配不平衡。
- 通过 C 语言级别的火焰图分析, 发现 `cjson` 模块 (JSON 解析) 消耗了约 60% 的 CPU 时间, 是系统的绝对瓶颈。
- 核心业务逻辑 `dns_server.lua` 仅占约 5% CPU。
- `cosocket` 接收操作消耗了约 16% 的 CPU 时间, 是另一个显著的消耗点。

性能优化效果

- 核心瓶颈 (JSON 解析) 的 CPU 消耗降低 60% 以上。
- 仅通过配置调优 (启用 `reuseport`), 就使整体性能提升 20-30%。
- 系统吞吐量获得显著提升。
- 精准定位多个性能问题, 为客户提供了清晰、多层次的优化路径, 节省了大量分析和试错时间。

从堆内存异常到精准定位：OpenResty XRay 在 Java 内存问题诊断中的实践

用户案例：<https://blog.openresty.com.cn/cn/xray-java-mem/>

问题

客户订单系统在性能回归测试中，堆内存持续上升，即使在多次 Full GC 后也无法回收，系统存在严重的内存泄漏风险。

使用 OpenResty XRay 排查

定位泄漏根源: 使用 GC 对象引用分析，快速找到非预期的内存驻留对象及其完整的引用链。

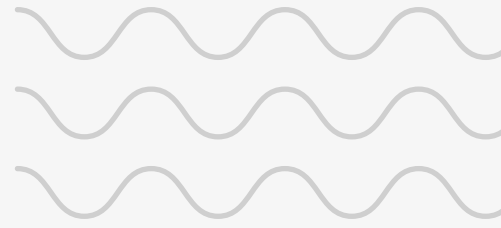
排查内存抖动: 通过 GC 对象分配次数分析，发现高频创建临时对象的代码路径。

定位大对象: 利用 GC 对象分配大小分析，精确定位“大对象”的创建源头。

性能优化效果

- 成功解决内存泄漏问题，保障了系统的稳定性。
- 无需重启服务或修改代码，对生产环境透明无影响。
- 不依赖 JVM 安全点 (Safepoint)，分析过程无 STW (Stop-the-World) 暂停。
- 直观的可视化图表（火焰图、引用图）极大降低了内存问题的分析门槛，快速定位根源。

OpenResty XRay 如何精准定位 Gzip 配置的隐性成本



用户案例：<https://blog.openresty.com.cn/cn/xray-gzip-cpu/>

问题

在高吞吐量的金融级业务场景中，系统表面指标（QPS、延迟）看似达标，但高昂的 CPU 成本成为吞吐量扩展的瓶颈。

使用 OpenResty XRay 排查

火焰图清晰显示，CPU 资源被压缩与解压过程大量占据，而非核心业务逻辑。

- * Gzip 压缩占用 54% 的 CPU 时间。
- * Gzip 解压占用 61% 的 CPU 时间。
- * 即便是 Brotli 压缩也消耗了 30.9% 的 CPU。
- * 将宏观的 CPU 消耗，直接追溯到微观的 `gzip_level` 等配置参数上。

基于 OpenResty XRay 提供的数据洞察，优化 Gzip 相关配置后

- * 系统整体 CPU 使用率显著下降。
- * 核心业务线程的 CPU 时间得到释放。
- * QPS 获得实质性提升。
- * 请求延迟显著改善。

在 OpenResty 或 Nginx 进程中追踪最慢的 PCRE 正则表达式



用户案例：<https://blog.openresty.com.cn/cn/nginx-slowest-pcre-regexes/>

问题

OpenResty/Nginx Lua 应用效率低下，导致服务器 CPU 使用率持续达到 100%。

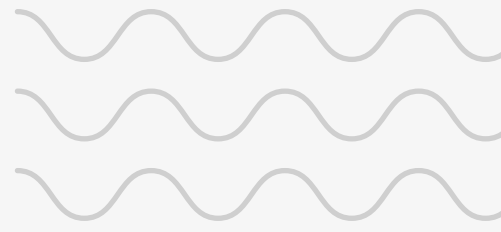
使用 OpenResty XRay 排查

- 初步诊断: C 语言级别的 CPU 火焰图显示，绝大部分 CPU 时间消耗在执行 Lua 代码中的 PCRE 正则匹配上。
- 深入分析: Lua 级别的 CPU 火焰图进一步确认，性能瓶颈集中在单一代码路径的 `ngx.re.match` API 调用上。
- 精准定位: 使用 `lj-slowest-ngx-re` 分析器，精准定位到最慢的正则表达式为 `(?:.)*css`，其执行延迟超过 20.6 毫秒，原因是贪婪量词 `.*` 导致了大量回溯。

优化正则表达式，避免了引擎的大量回溯

- * CPU 瓶颈消除：CPU 不再消耗于 Lua 正则表达式计算，热点转移到正常的 I/O 写操作（`writev` 系统调用）。
- * 性能极大提升：最慢正则表达式的最大延迟从 20.6 毫秒降至 59 微秒，性能提升了数百倍。
- * 精准定位问题：无需猜测或逐一排查，直接定位到有问题的正则表达式。
- * 无需修改复杂的业务逻辑，快速定位并解决了深层次的性能问题。

OpenResty XRay - 产品愿景



让复杂系统透明可控

- OpenResty XRay 致力于解决现代计算世界最具挑战性的难题：通过零侵入式技术，将任何黑盒系统变为白盒，实现从硬件到业务层的全栈透视。为我们的用户提供前所未有的系统洞察能力。
- 我们正在打造业界最强大的自动化问题诊断引擎。无需修改目标系统、无需重启进程，仅通过轻量级 Agent 就能对整个软件栈进行全方位扫描，自动发现并定位各类性能瓶颈、安全隐患和稳定性问题。
- 我们正在开拓突破性的实时热补丁技术。未来，OpenResty XRay 不仅能发现问题，还能自动为运行中的系统打上修复补丁，让诊断和修复变得简单而优雅。

了解更多

欢迎试用 OpenResty XRay.



官网：<https://openresty.com.cn/cn/>

博客：<https://blog.openresty.com.cn/cn/>

视频教程：<https://space.bilibili.com/457424101>

Q&A：<https://openresty.com.cn/cn/xray/faq/>