



OpenResty XRay

產品簡介

提供世界一流的解決方案，即刻駕馭錯綜複雜的現代 Web 挑戰

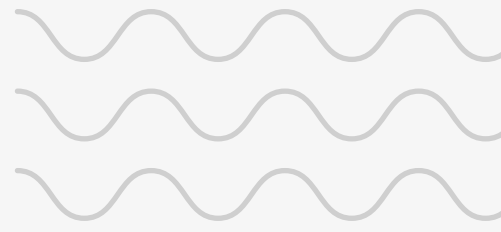
Powered by OpenResty Inc.

目錄

1. 業界痛點
2. 產品簡介
3. 核心優勢與亮點
4. 底層實現
5. 客戶案例
6. 產品願景
7. 更多信息



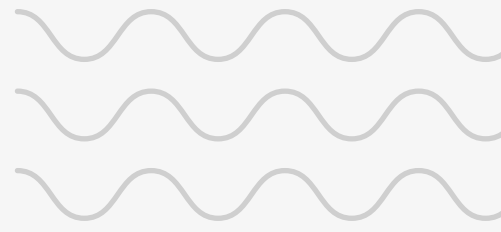
現代軟件世界的挑戰



人們對軟硬件系統的洞察力和控制力逐漸降低

- 軟硬件系統規模越來越龐大、業務邏輯越來越複雜，計算機算力和用戶數量指數增長，往往導致問題定位難度上升，性能瓶頸、可靠性問題和安全漏洞頻頻出現
- 大部分問題線上獨有，且難以復現，下線檢查又嚴重影響軟件系統的正常運行和用戶體驗

擁抱雲原生的困擾



隨着容器技術和微服務架構的普及，企業面臨前所未有的複雜性挑戰

- 技術棧激增：容器化部署導致應用數量倍增，各類技術棧和發行版並存，系統複雜度指數級提升
- 可觀測性缺失：精簡容器鏡像往往缺乏必要的調試工具，問題排查難度大幅提高
- 問題難追溯：容器的快速銷燬與重建機制雖提高了系統韌性，卻也容易掩蓋潛在的軟件缺陷，讓根因分析變得更加困難

業界痛點



- 複雜性困境：傳統 APM 工具需要在應用中植入探針代碼，不僅增加了系統複雜度，還可能引入新的不穩定因素
- 資源浪費：大量的數據採集、傳輸和存儲不僅帶來高昂的成本，還會影響生產系統性能。同時，複雜的處理流程也容易出现誤報和漏報
- 洞察力不足：表層的監控指標僅能反映問題現象，無法深入分析根本原因。缺乏對全技術棧的深度理解，使得問題定位往往停留在表面
- 響應滯後：從發現異常到確認問題再到實施解決方案，過程耗時冗長，難以滿足快速變化的業務需求

OpenResty XRay

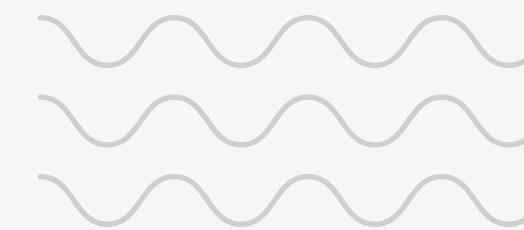


適用於開源軟件和 Web 應用的故障排查和性能優化工具

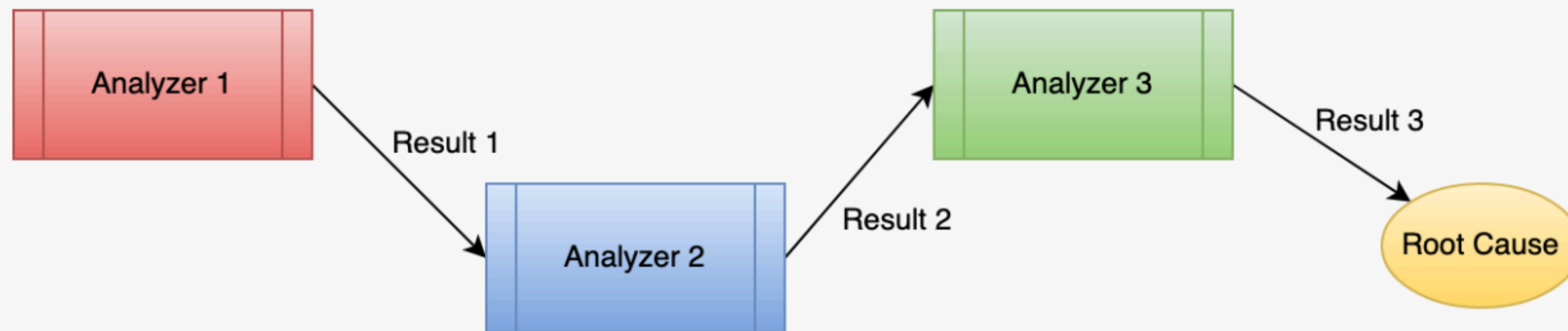
- OpenResty XRay 是新一代零侵入式動態追蹤產品，專為解決企業日益複雜的運維挑戰而生
- 它能夠全天候實時監測系統運行狀態，無需修改任何代碼，就能對雲端及服務器應用進行深度分析
 - 系統自動處理所有診斷數據，生成專業的分析報告，並提供精準的優化建議，讓複雜的問題診斷變得簡單直觀



OpenResty XRay



全自動採樣 無人值守模式



定時採樣



事件驅動

CPU 變化, 內存變化, IO 變化, 異常錯誤

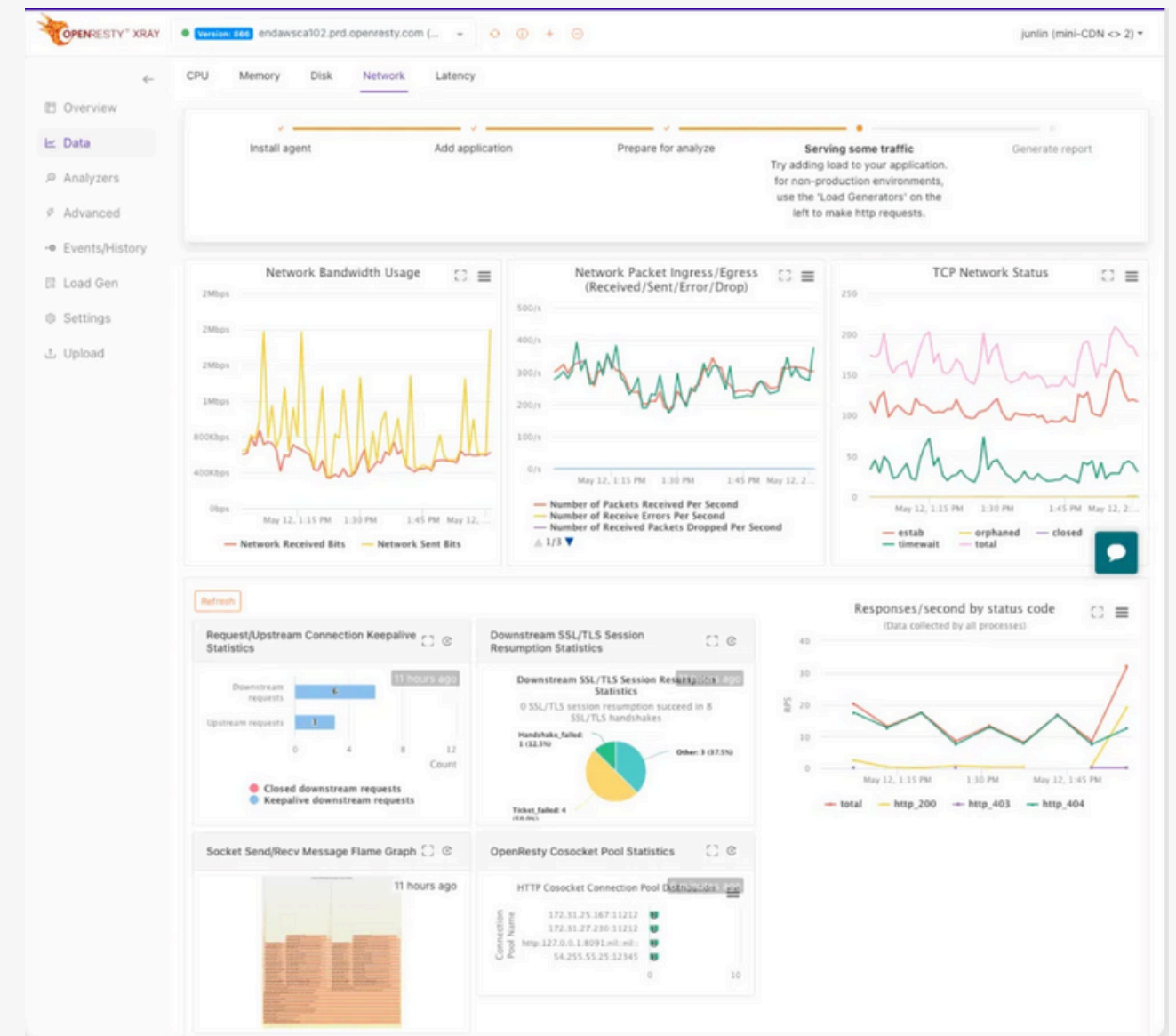


鏈式推理

實時監控各類系統性能指標

可視化監控面板 全方位展現系統運行狀態

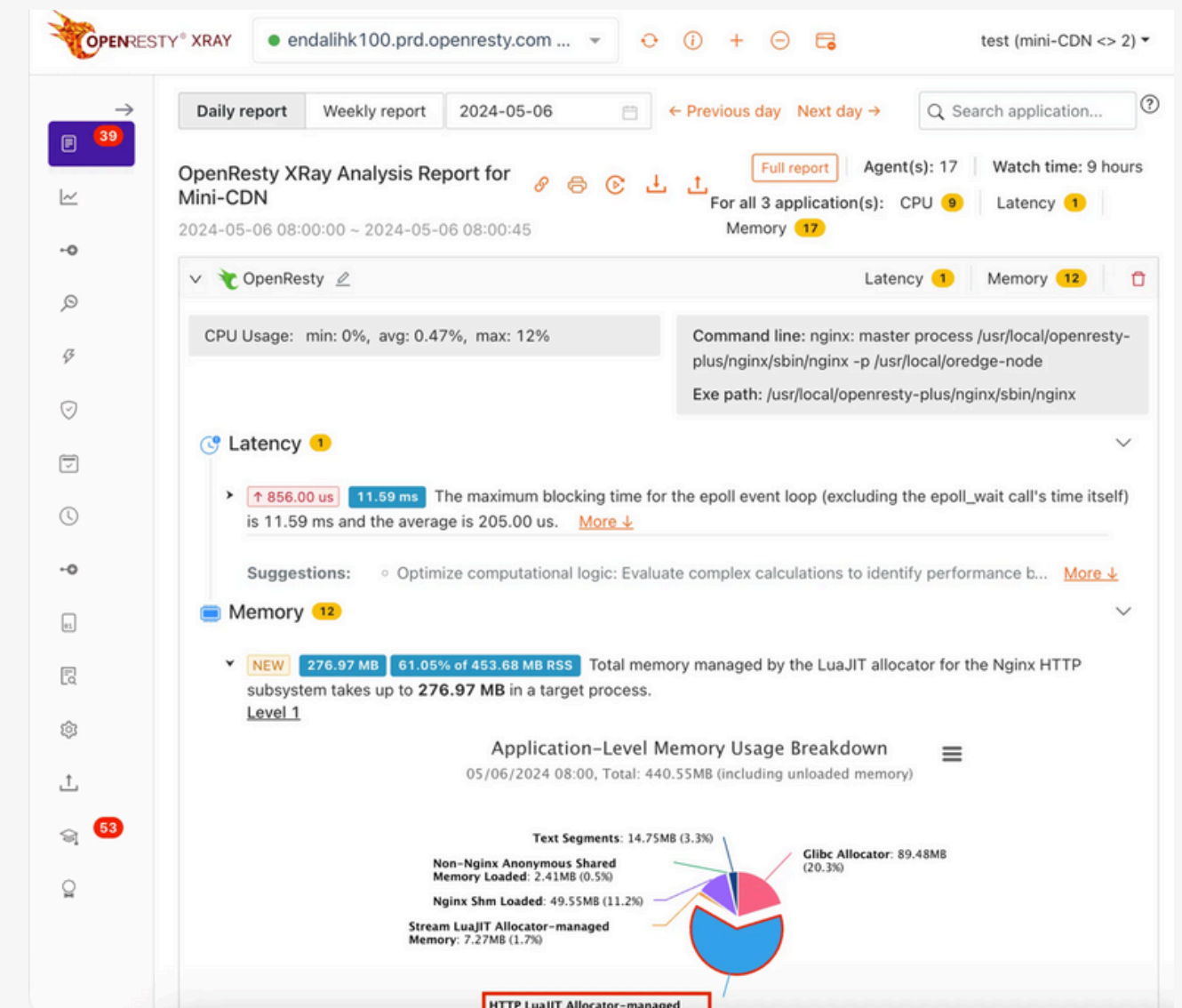
- 各類系統指標的採樣結果將彙總在可視化儀表板，實時掌握應用運行狀況與性能的變化趨勢
- 可以自動分析和定位不同應用中、不同的語言級別各類系統性能問題



全自動分析報告

不僅僅是追蹤與分析 深度分析解採樣數據

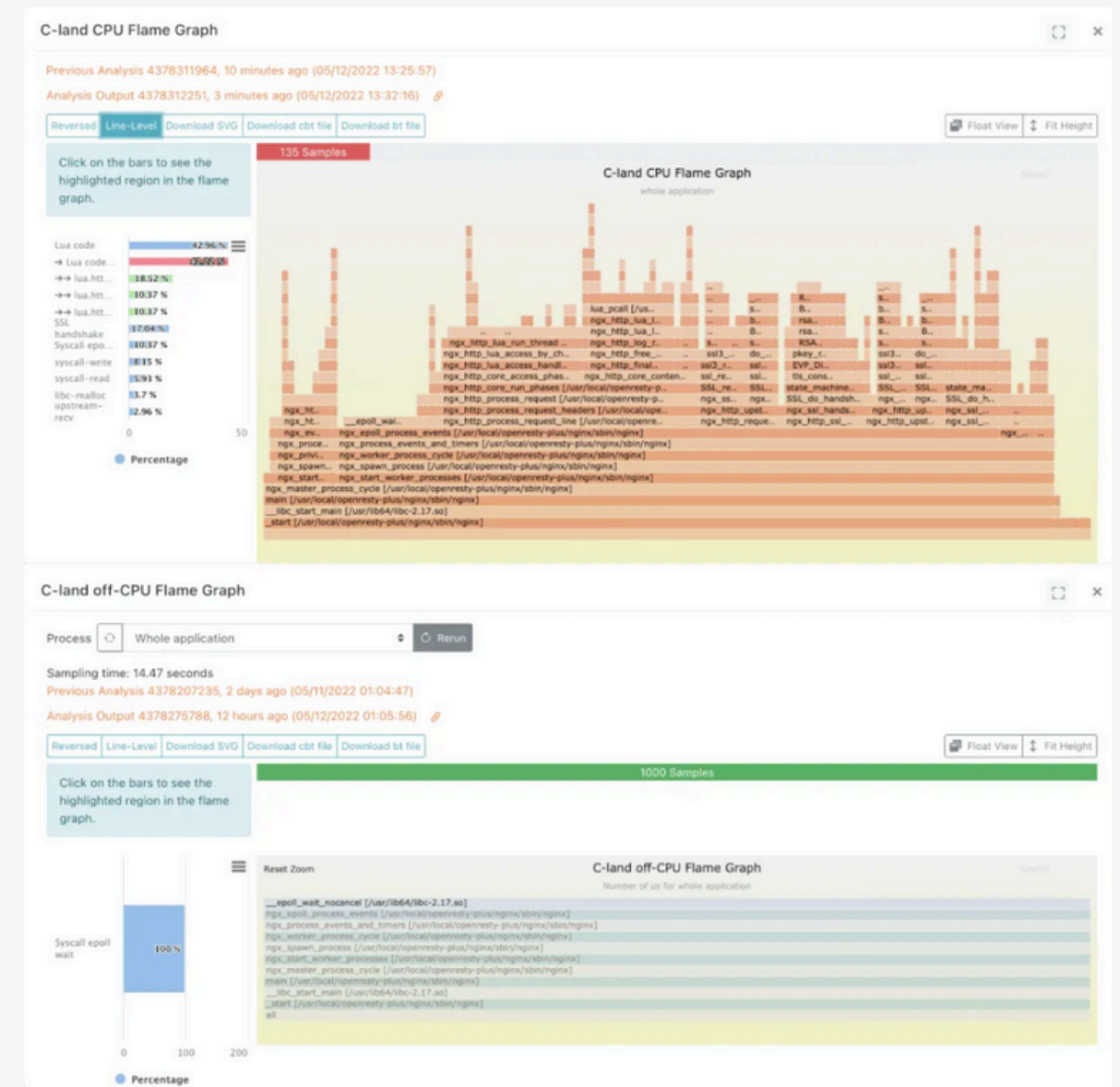
- 針對性能瓶頸、資源消耗等關鍵問題，XRay 將提供專業的根因分析和優化建議
- OpenResty 的專家團隊會幫助您深入分析，精準定位線上問題的根因，讓複雜問題診斷變得簡單高效



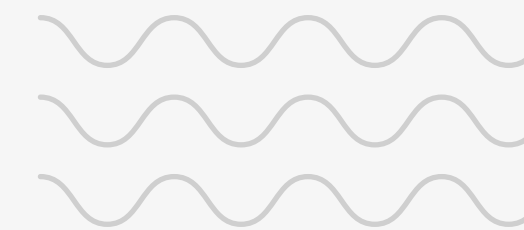
引導式分析

主動分析 實時調試

- 交互式故障排查工具，支持對正在運行的應用進行實時觀測和分析
- 通過精準引導，幫助您快速驗證優化方案，實現動態調試和性能調優，無需重啓即可查看修改效果



產品核心優勢

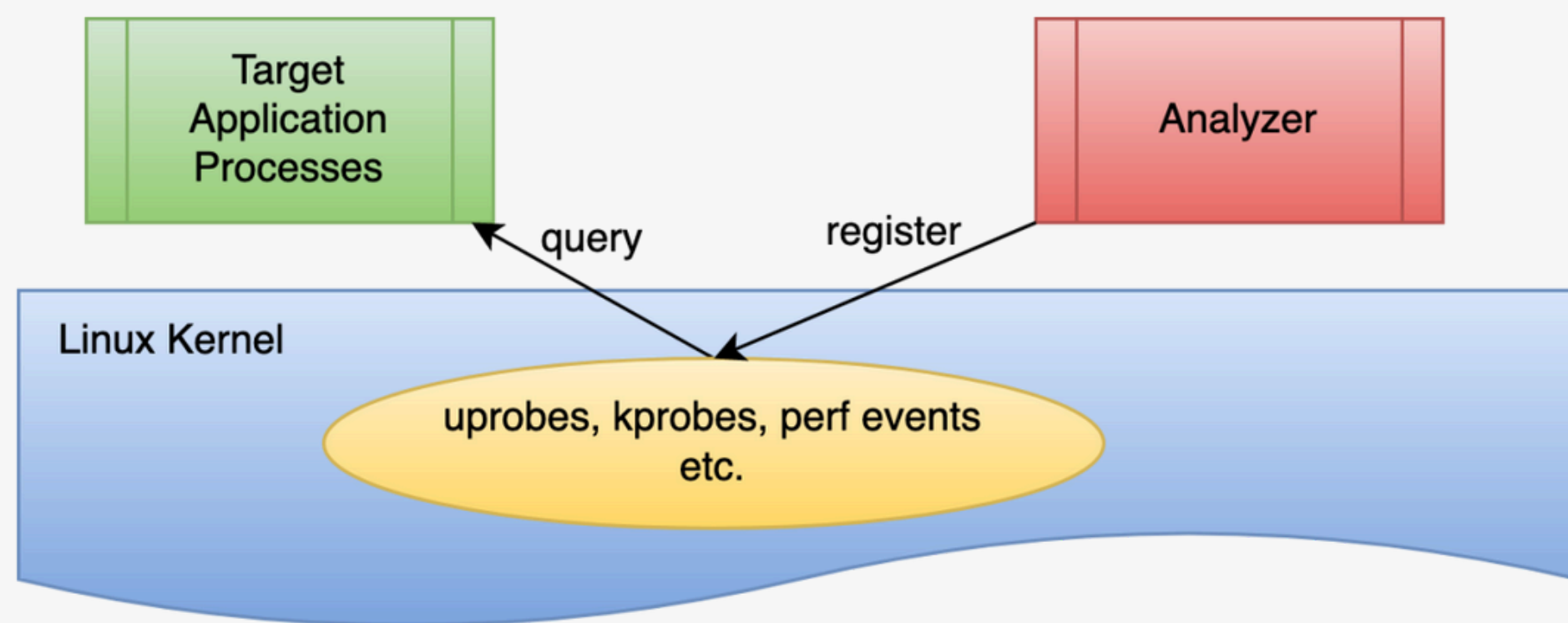


爲什麼選擇我們

1. 新一代動態追蹤技術驅動 100% 非侵入
2. 創新的自動採樣方法 高效輕量低開銷
3. 全棧、全方位無死角分析 涵蓋各類系統性能問題
4. 無需調試符號也能分析應用
5. 具備容器透明化能力
6. 自動分析安全問題

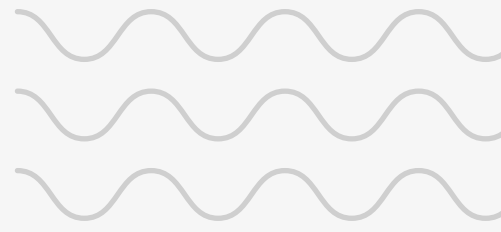
新一代動態追蹤技術驅動 100% 非侵入

動態追蹤技術的優勢



- 熱插拔，無需修改應用，無需安裝新插件、補丁、模塊，無需重啓應用，無需特殊啓動或編譯選項，無需重建應用或軟件包，無需任何代碼注入目標進程空間
- 運行中的進程和容器視爲只讀數據庫，並提取必要的信息來解決性能問題、異常、錯誤和安全漏洞
- 能在不訪問用戶源代碼的情況下，深入洞察在線或離線軟件系統，找出源碼層面的問題，準確定位到系統問題發生的源碼行

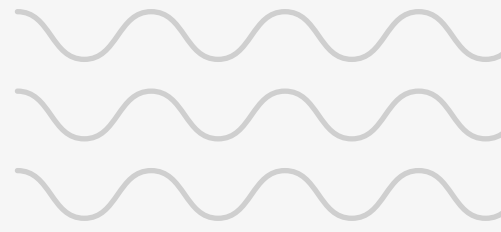
創新的自動採樣方法 高效輕量低開銷



動態追蹤技術的優勢

1. 精準定向追蹤，只追蹤與之相關的代碼路徑和函數
2. 按需採樣，信息採集量比傳統 APM 產品少很多，可以在數據源進行聚合彙總
3. 未採樣時的性能開銷嚴格為 0，採樣期間開銷通常也低至無法測量

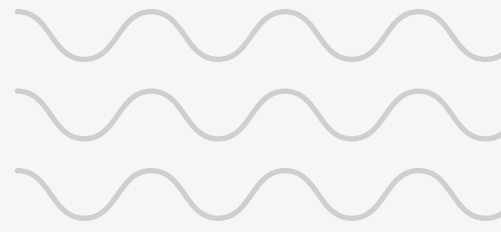
全棧、全方位無死角分析



涵蓋各類系統性能問題

1. 覆蓋不同軟件層面的代碼路徑：業務編程語言層面（Lua/Python/PHP/Perl/Go/等），系統編程語言層面（C/C++/Rust），操作系統內核層面（網絡協議棧/進程調度器/內存管理/系統調用）
2. 線程阻塞和鎖
3. CPU 問題分析：CPU 使用率過高或阻塞、不同代碼上 CPU 時間分佈、常見的 CPU 瓶頸
4. 內存佔用大或內存泄漏：定量分析內存分佈、內存分配器的使用情況，內存泄漏、內存碎片、延遲釋放等
5. Disk I/O 性能分析
6. 請求長延時、長尾請求：分解不同應用操作、處理階段的延時，異步非阻塞 IO 的延時統計
7. 智能抓包和異常捕捉：只抓實際有異常的網絡連接上的包
8. Core Dump 文件及進程崩潰

無需調試符號也能分析應用



OpenResty XRay 突破了生產環境調試符號缺失的難題

1. OpenResty XRay 建立了業界領先的中央符號數據庫，已積累數百 TB 的公開包調試信息，並仍在快速擴充
2. 藉助自主研發的機器學習算法，即使面對完全沒有調試信息的二進制程序，OpenResty XRay 也能實現符號的自動重建
3. 只要調試符號曾被 OpenResty XRay 中央包數據庫索引，目標機器無需安裝或保存調試符號。客戶無需在生產環境部署龐大的調試文件，就能獲得完整的程序運行時信息，真正實現了輕量化部署與深度分析的完美結合

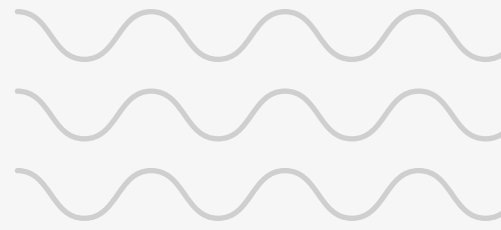
OpenResty XRay 容器觀測能力



OpenResty XRay 突破性地實現了對容器邊界的無縫穿透，只需在宿主機部署一個 Agent，就能自動發現並採樣分析同一主機上其他的 Docker 容器和 Kubernetes Pod 中運行的應用

- 無需修改容器鏡像
- 零配置即可實現跨容器監控
- 不影響容器的資源限制和安全策略
- 支持動態擴縮容場景下的自動發現

自動化安全風險檢測

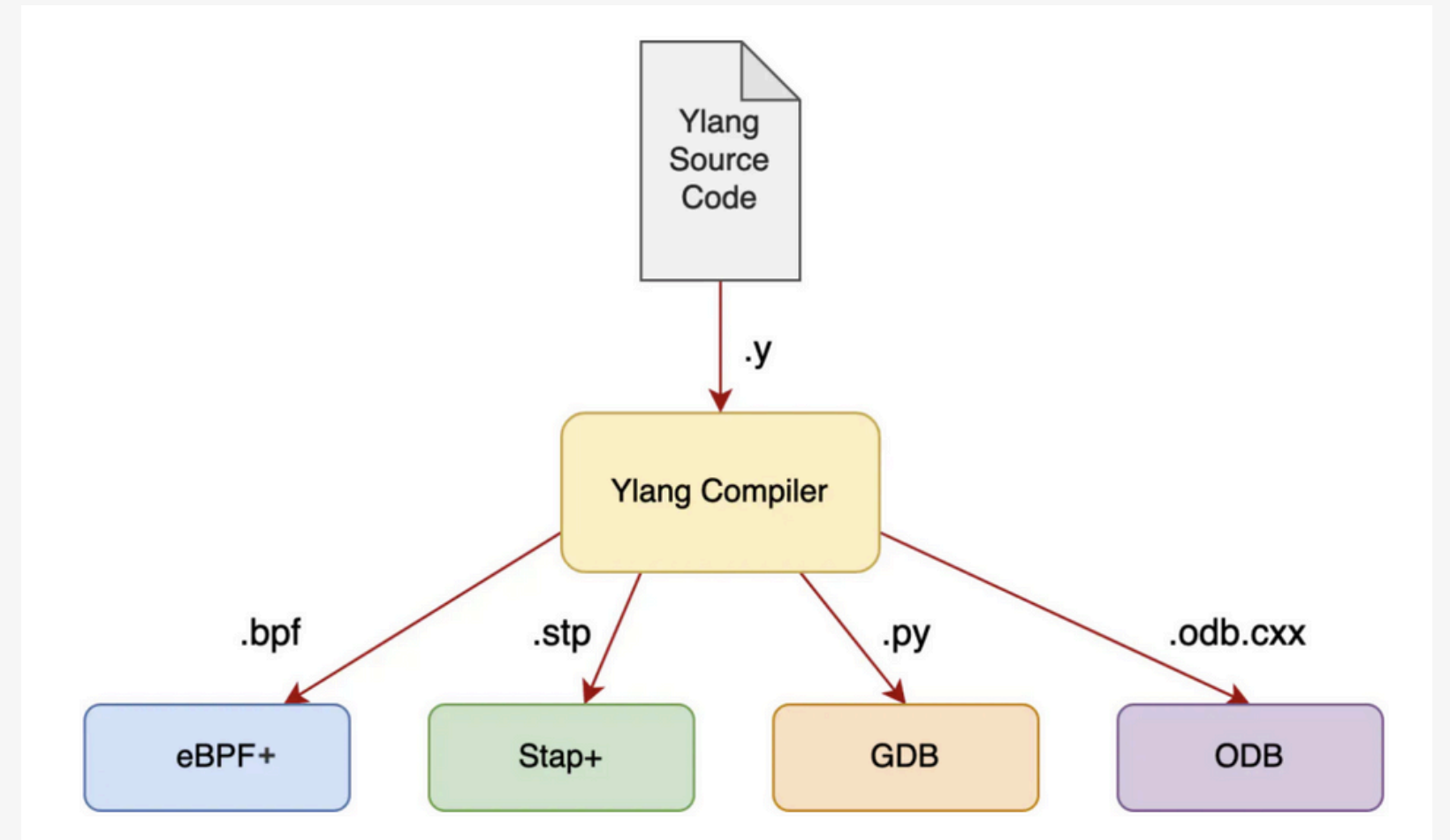


OpenResty XRay 提供全方位的安全漏洞掃描能力

1. 系統實時自動識別傳輸層安全風險，包括未加密通信、TLS 證書校驗缺失以及過時的加密協議
2. 深度掃描與分析代碼執行行爲，定位潛在的遠程 shell 命令注入漏洞

OpenResty XRay 底層實現

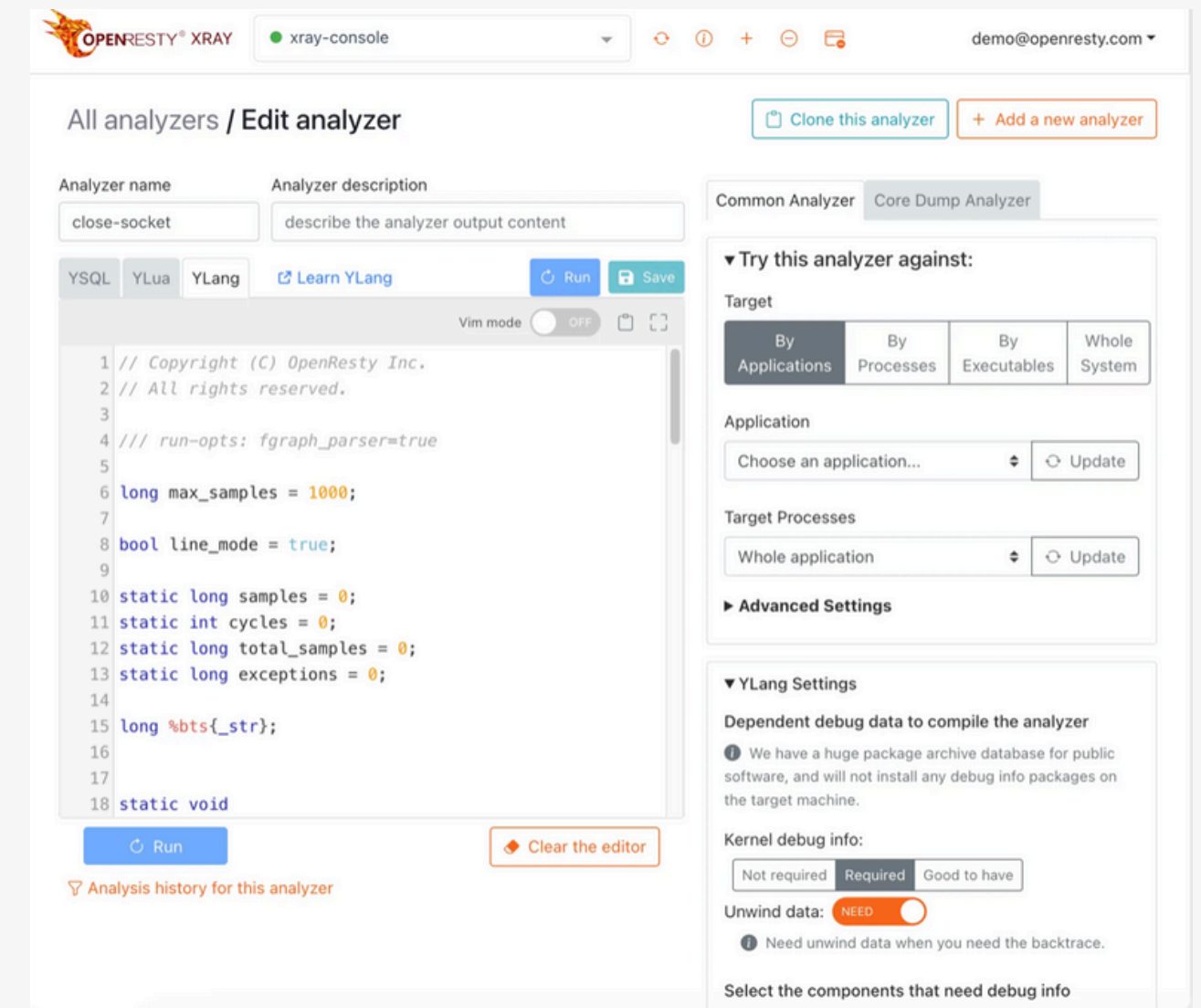
1. 在底層實現上，OpenResty XRay 由 OpenResty Inc. 開發的 Y 語言驅動
2. 支持多種不同的調試框架，如 Stap+, eBPF+, GDB和 ODB
3. ylua、ysql、y++ 等高級工具包，使 XRay 具有更高的可編程、可擴展能力。因此能以更低的開銷成本、更靈活和高效地排查複雜的系統問題



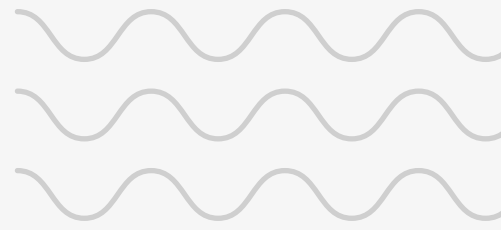
一次編寫 到處運行

數百種標準分析器

1. 全面覆蓋 - 能夠分析和診斷各種不同類型的性能問題和異常情況
2. 精準定位 - 通過專門的分析器快速鎖定具體問題根源
3. 自動化程度高 - 無需人工介入就能完成複雜的分析工作
4. 節省時間 - 問題診斷從可能需要數天縮短到分鐘級
5. 提前預警 - 能夠及早發現潛在問題，防患於未然
6. 支持自定義 - 專家團隊能夠針對您的需求，定製分析器



多語言性能分析器



OpenResty XRay 支持主流 Linux 發行版和容器佈署方式

1. 支持 Nginx、Envoy 和 Kong 等主流 Web 服務器和代理
2. 兼容 LuaJIT、Python、PHP、Go、Java、Erlang、Perl、Ruby 以及 Rust 等多種編程語言和運行時環境
3. 適配主流 Linux 發行版：涵蓋 Ubuntu、Debian、Fedora、Red Hat Enterprise Linux (RHEL)、CentOS、openSUSE、AlmaLinux、Amazon Linux、阿里雲 Linux、Rocky Linux、騰訊 Linux 以及麒麟等
4. 支持容器部署方式：Docker、Kubernetes

未來支持計劃

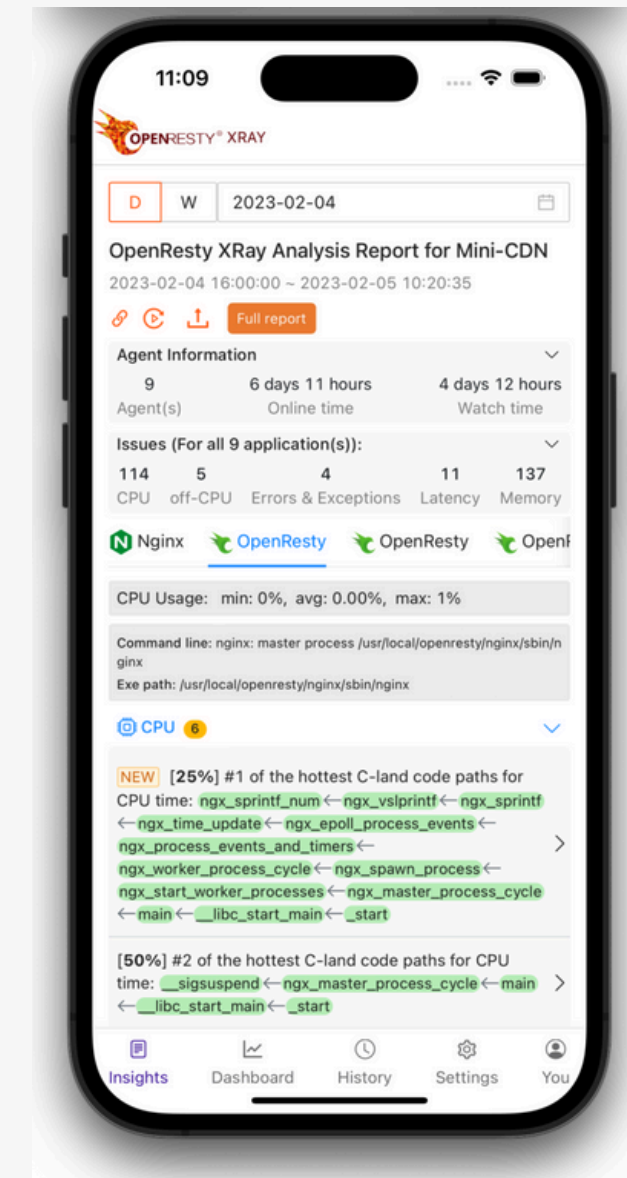
- nodejs
- pytorch、langchain AI 軟件棧等

OpenResty XRay 移動端應用

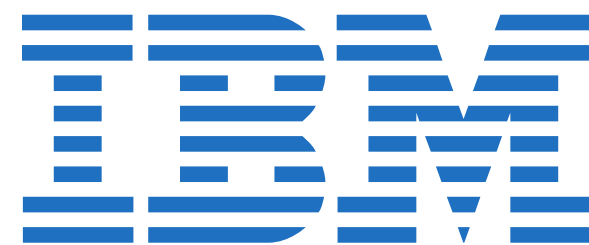
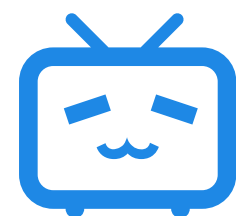
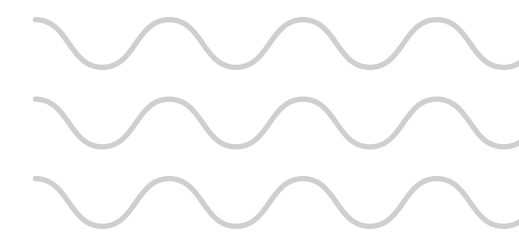
隨時隨地 察看您的在線應用

- Android / Google Play

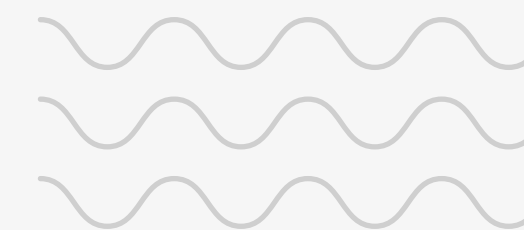
- iOS / 蘋果商店



為眾多企業客戶所信任



OpenResty XRay 分析和解決 B 站重大線上事故



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

長期戰略合作伙伴



Bilibili

線上服務不可用重大事故

事故

- B 站基於 OpenResty 開發的網關係統出現全線故障
- 所有服務器 OpenResty 進程 CPU 100%，無法處理請求
- 重啓和代碼回滾都無效

使用 OpenResty XRay 排查

1. 使用 OpenResty XRay 生成 C 語言級 CPU 火焰圖，定位到問題集中在 Lua 代碼執行
2. 進一步通過 Lua 級 CPU 火焰圖，鎖定具體代碼路徑
3. 最終發現根因：業務配置中寫入了字符串 "0" 作為權重值，而系統期望是數值類型的 0，導致無限遞歸

使用成果

- XRay 採樣分析僅花費幾十秒到幾分鐘
- 100% 非侵入式動態追蹤，無需修改目標進程
- 快速精準定位到問題代碼路徑

解決自定義 Kong 插件中的 Lua 異常所引起的 CPU 瓶頸

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



長期戰略合作夥伴



Kong

CPU 性能瓶頸

問題

即使在流量不大的情況下，客戶的 Kong 服務器 CPU 使用率異常高(80%)

使用 OpenResty XRay 排查

- 發現熱點代碼路徑中 `string.lower` 函數拋出異常
- 定位到具體錯誤: 在自定義 `auth` 插件中錯誤地向 `string.lower` 傳入了 `nil` 值
- 精確到源文件位置: `.../kong/plugins/auth/handler.lua` 第 35 行

解決方案

修復 `auth` 插件中的 bug，確保不向 `string.lower` 傳入 `nil` 值

使用成果

- CPU 使用率從 80% 降至 50%
- 性能提升 37.5%
- 無需複雜的代碼重構
- 快速解決問題，為客戶節省大量排查時間

OpenResty XRay 如何洞穿 Lua 與 C 的邊界，找回 45% 的 CPU 算力

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

問題

某金融行業核心業務網關集羣高峯時段 CPU 使用率持續 100% 飽和，P99 延遲（99%分位延遲）急劇抖動，服務質量嚴重下降

使用 OpenResty XRay 排查

- 發現驚人熱點：C 函數 `pkey_rsa_decrypt` 消耗了高達 44.8% 的 CPU 時間
- 根因定位：精準追溯到 Lua 肇事者：`cb-session-validation.lua` 插件的第 93 行
- 調用鏈還原：完整繪製了從 `Lua 插件` -> `JWT 庫 (Lua)` -> `libcrypto.so (C 庫)` 的調用路徑

性能優化效果

- 釋放了近 45% 的 CPU 算力，系統吞吐能力幾乎翻倍
- 高峯期 CPU 不再觸頂，P99 延遲恢復平穩
- 直接獲得從 Lua 到 C 的完整調用棧，鎖定問題根源
- 極大縮短了跨語言邊界性能問題的排查時間，為客戶挽回了因性能瓶頸造成的損失

OpenResty XRay 實現 DNS 服務性能 60% 優化的案例解析

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

問題

客戶的 DNS 服務系統面臨嚴重的性能問題

CPU 負載不均: 部分 Nginx worker 進程 CPU 佔用率過高，而其他進程相對空閒。

- 響應延遲增加: 系統整體響應慢，在高負載情況下尤其明顯。

使用 OpenResty XRay 排查

- 發現請求分配不均: 定位到所有監聽端口均未啓用 `reuseport` 選項，導致 worker 間請求分配不平衡。
- 通過 C 語言級別的火焰圖分析，發現 `cjson` 模塊（JSON 解析）消耗了約 60% 的 CPU 時間，是系統的絕對瓶頸。
- 核心業務邏輯 `dns_server.lua` 僅佔約 5% CPU。
- `cosocket` 接收操作消耗了約 16% 的 CPU 時間，是另一個顯著的消耗點。

性能優化效果

- 核心瓶頸（JSON 解析）的 CPU 消耗降低 60% 以上。
- 僅通過配置調優（啓用 `reuseport`），就使整體性能提升 20-30%。
- 系統吞吐量獲得顯著提升。
- 精準定位多個性能問題，爲客戶提供了清晰、多層次的優化路徑，節省了大量分析和試錯時間。

從堆內存異常到精準定位：OpenResty XRay 在 Java 內存問題診斷中的實踐

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

問題

客戶訂單系統在性能迴歸測試中，堆內存持續上升，即使在多次 Full GC 後也無法回收，系統存在嚴重的內存泄漏風險。

使用 OpenResty XRay 排查

定位泄漏根源: 使用 GC 對象引用分析，快速找到非預期的內存駐留對象及其完整的引用鏈。

排查內存抖動: 通過 GC 對象分配次數分析，發現高頻創建臨時對象的代碼路徑。

定位大對象: 利用 GC 對象分配大小分析，精確定位“大對象”的創建源頭。

性能優化效果

- 成功解決內存泄漏問題，保障了系統的穩定性。
- 無需重啓服務或修改代碼，對生產環境透明無影響。
- 不依賴 JVM 安全點 (Safepoint)，分析過程無 STW (Stop-the-World) 暫停。
- 直觀的可視化圖表（火焰圖、引用圖）極大降低了內存問題的分析門檻，快速定位根源。

OpenResty XRay 如何精準定位 Gzip 配置的隱性成本

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

問題

在高吞吐量的金融級業務場景中，系統表面指標（QPS、延遲）看似達標，但高昂的 CPU 成本成爲吞吐量擴展的瓶頸。

使用 OpenResty XRay 排查

火焰圖清晰顯示，CPU 資源被壓縮與解壓過程大量佔據，而非核心業務邏輯。

- * Gzip 壓縮佔用 54% 的 CPU 時間。
- * Gzip 解壓佔用 61% 的 CPU 時間。
- * 即便是 Brotli 壓縮也消耗了 30.9% 的 CPU。
- * 將宏觀的 CPU 消耗，直接追溯到微觀的 `gzip_level` 等配置參數上。

基於 OpenResty XRay 提供的數據洞察，優化 Gzip 相關配置後

- * 系統整體 CPU 使用率顯著下降。
- * 核心業務線程的 CPU 時間得到釋放。
- * QPS 獲得實質性提升。
- * 請求延遲顯著改善。

在 OpenResty 或 Nginx 進程中追蹤最慢的 PCRE 正則表達式



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

問題

OpenResty/Nginx Lua 應用效率低下，導致服務器 CPU 使用率持續達到 100%。

使用 OpenResty XRay 排查

- 初步診斷: C 語言級別的 CPU 火焰圖顯示，絕大部分 CPU 時間消耗在執行 Lua 代碼中的 PCRE 正則匹配上。
- 深入分析: Lua 級別的 CPU 火焰圖進一步確認，性能瓶頸集中在單一代碼路徑的 `ngx.re.match` API 調用上。
- 精準定位: 使用 `lj-slowest-ngx-re` 分析器，精準定位到最慢的正則表達式為 `(?:.*)*css`，其執行延遲超過 20.6 毫秒，原因是貪婪量詞 `.*` 導致了大量回溯。

優化正則表達式，避免了引擎的大量回溯

- * CPU 瓶頸消除：CPU 不再消耗於 Lua 正則表達式計算，熱點轉移到正常的 I/O 寫操作 (`writev` 系統調用)。
- * 性能極大提升：最慢正則表達式的最大延遲從 20.6 毫秒降至 59 微秒，性能提升了數百倍。
- * 精準定位問題：無需猜測或逐一排查，直接定位到有問題的正則表達式。
- * 無需修改複雜的業務邏輯，快速定位並解決了深層次的性能問題。

OpenResty XRay - 產品願景



讓複雜系統透明可控

- OpenResty XRay 致力於解決現代計算世界最具挑戰性的難題：通過零侵入式技術，將任何黑盒系統變為白盒，實現從硬件到業務層的全棧透視。為我們的用戶提供前所未有的系統洞察能力。
- 我們正在打造業界最強大的自動化問題診斷引擎。無需修改目標系統、無需重啟進程，僅通過輕量級 Agent 就能對整個軟件棧進行全方位掃描，自動發現並定位各類性能瓶頸、安全隱患和穩定性問題。
- 我們正在開拓突破性的實時熱補丁技術。未來，OpenResty XRay 不僅能發現問題，還能自動為運行中的系統打上修復補丁，讓診斷和修復變得簡單而優雅。

瞭解更多

歡迎試用 OpenResty XRay.



官網：<https://openresty.com/tw/>

博客：<https://blog.openresty.com/tw/>

影片教程：<https://www.youtube.com/channel/UCXVmwF-UCScv2ftsGoMqxhw>

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