

## 企业API网关热插拔插件的设计与实现

王盛义

引用本文

王盛义. 企业API网关热插拔插件的设计与实现[J]. 计算机科学, 2024, 51(6A): 230800176-7.

WANG Shengyi. Design and Implementation of Hot-swappable Plugin for Enterprise API Gateway[J].

Computer Science, 2024, 51(6A): 230800176-7.

---

## 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[一种易部署的Android应用程序动态监测方案](#)

Easy-to-deploy Dynamic Monitoring Scheme for Android Applications

计算机科学, 2020, 47(2): 262-268. <https://doi.org/10.11896/jsjcx.190100117>

[从Smalltalk 语言的结构特征看“软插件”的形成](#)

计算机科学, 1990, 17(3): 19-23.

[插件式网络安全集成防护框架](#)

计算机科学, 2005, 32(4): 149-150.

[基于XSLT的XML安全的应用研究](#)

计算机科学, 2004, 31(B09): 105-108.

[基于DirectUI可扩展应用程序架构的设计与实现](#)

Expanded Application Framework Based on DirectUI

计算机科学, 2012, 39(Z11): 295-300.

# 企业 API 网关热插拔插件的设计与实现

王盛义

同济大学电子与信息工程学院 上海 201804

**摘要** 为了解决微服务架构下传统 API 网关扩展能力弱、无法热更新等问题,对 API 网关扩展性进行研究分析,引入热插拔机制来实现企业 API 网关热插拔插件,同时,提出了一种企业 API 网关热插拔插件解决方案。实验结果表明,所提出的 API 网关的热插拔插件方案在支持网关插件热更新的同时不会对 API 网关的整体性能造成影响,也不会对业务功能的稳定性造成影响。目前,企业 API 网关已经在几十家大型企业得到应用,提供了身份鉴权、限流限速、协议转换、请求改写等 30 余种热插拔插件。通过企业 API 网关彻底解决了原有 API 网关无法热更新、热部署、难扩展等问题,减少了 40% 的重复开发工作,节省了 30% 运维成本,为企业 API 网关的进一步发展和应用提供了有益的参考,也为构建高效、安全、可扩展的企业 API 网关提供了新的思路。

**关键词:** API 网关;插件;热插拔;OpenResty;Nginx

**中图分类号** TP311.1

## Design and Implementation of Hot-swappable Plugin for Enterprise API Gateway

WANG Shengyi

College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China

**Abstract** Order to solve the problems of traditional API gateways with weak scalability and inability to hot update under the microservice architecture, this paper studies and analyzes the scalability of API gateways and introduces a hot-swappable mechanism to realize hot-swappable plugins for enterprise API gateways. At the same time, a hot-swappable plugin solution for enterprise API gateways is proposed. Experimental results show that the proposed solution will not affect the overall performance of the API gateway or the stability of business functions when supporting hot update of gateway plugins. At present, the enterprise API gateway has been applied in dozens of large enterprises, providing more than 30 kinds of hot-swappable plugins such as identity authentication, current and speed limiting, protocol conversion, and request rewriting. The enterprise API gateway completely solves the problems of the original API gateway being unable to be hot updated, hot deployed, and difficult to expand, reducing 40% of repeated development work, and saving 30% of operation and maintenance costs. It provides a useful reference for the further development and application of enterprise API gateways. It also provides new ideas for building efficient, secure, and scalable enterprise API gateways.

**Keywords** API gateway, Plugin, Hot-swappable, OpenResty, Nginx

### 1 引言

随着当今信息技术快速发展,微服务架构作为一种细粒度、灵活、可扩展的解决方案,被越来越多的企业所采用。在微服务架构中,API 网关<sup>[1]</sup>作为系统内外部交互的核心组件<sup>[2]</sup>以及流量入口<sup>[3]</sup>,承载着 API 全生命周期管理的重要作用<sup>[4]</sup>。早期的 API 网关只有简单的路由转发、负载均衡等基础功能。随着企业 API 数量不断增多以及用户需求日益复杂化,需要在不同的上游服务中重复实现限流、鉴权、协议转换等通用功能。这不仅导致功能难以重用,且一旦功能发生变化,所有的上游应用都需要随之变更<sup>[5]</sup>,这些问题已经严重影响了业务系统运行的稳定性。

针对以上问题,本文提出了一种针对企业级 API 网关的热插拔插件解决方案。通过引入热插拔插件机制,将业务系统

所需的通用功能以插件的方式集成到 API 网关中<sup>[6]</sup>。通过热插拔插件方案,可以在不影响现有系统稳定运行情况下,为企业 API 网关动态更新或移除某些功能模块,从而实现对 API 网关的灵活扩展。本文通过提升传统 API 网关的扩展性和灵活性,为企业构建高效、安全、可扩展的 API 网关提供新的思路,同时,有望为企业 API 网关的应用和发展提供有益参考。

### 2 相关工作

Netcraft 公司于 1994 年底在英国成立,多年来一直致力于互联网市场以及在线安全方面的咨询服务。根据 Netcraft 发布的 2023 年 12 月份全球 Web 服务器调查报告<sup>[7]</sup>,Apache 的市场份额位居第一,Nginx 以低于 Apache 0.29% 的市场份额居第二位,Cloudflare 和 OpenResty 分别位居第三和第四,结果如图 1 所示。

基金项目:国家重点研发计划(2022YFB3305700)

This work was supported by the National Key Research and Development Program of China(2022YFB3305700).

通信作者:王盛义

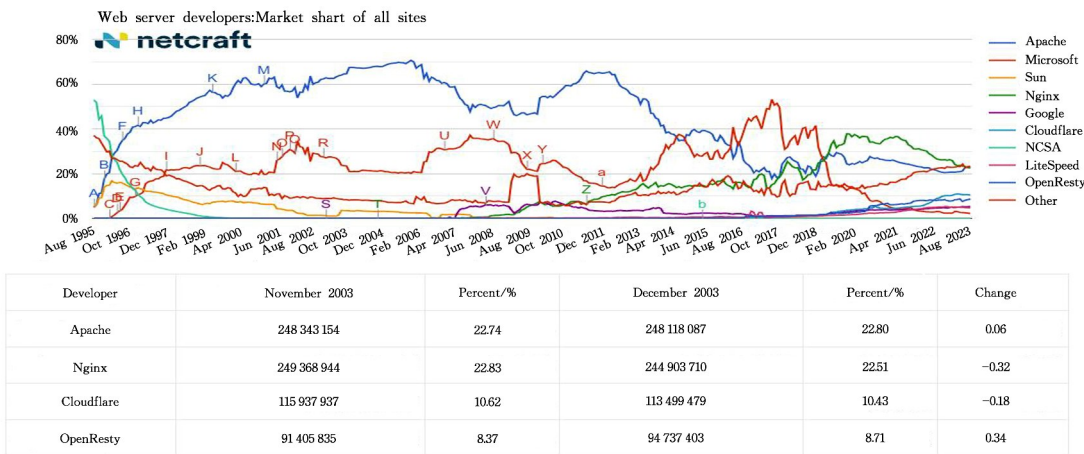


图1 Nercraft 2023年12月份全球Web服务器调查报告

Fig. 1 Global Web server survey report of Nercraft in December 2023

Apache HTTP Server 是 Apache 软件基金会开源的一款 Web 代理服务器。Apache 官方所支持的功能,更多是侧重于通用的业务场景,一些个性化的业务需求无法得到满足。自定义功能模块的开发门槛较高,且开发周期较长,不符合当下的短、平、快的软件交付模式。

Nginx 是一款基于事件驱动的高性能、低内存、热部署、可扩展的纯异步实现的 Web 代理服务器<sup>[1]</sup>。Nginx 自身虽然是支持可扩展的,但是门槛却很高,存在明显短板。虽然可以通过重新编译 Nginx 为其功能带来扩展,但 Nginx 的运行仍然受制于声明式的配置文件,对 Nginx 配置的任何更改都必须通过重启服务才能生效,无法满足业务连续性要求较高的系统。

Cloudflare 在 2022 年 9 月份宣布使用内部自研的 Pingora 取代 Nginx 作为内部的 HTTP 代理服务器,由于其极致的性能和超高的稳定性,Cloudflare 在 Nercraft 2023 年 1 月份全球 Web 服务器调查报告中超过了 Nginx 和 Apache,登上了榜首的位置,但是由于 Pingora 是 Cloudflare 的内部闭源项目,因此还无法在国内的互联网行业广为应用。

OpenResty 是一款集成 Nginx 核心和 LuaJIT 的 Web 代理服务器以及动态网关,支持开发者通过编写 Lua 模块对功能进行扩展<sup>[8]</sup>。Lua 的脚本语言更加简单易懂,因此在扩展性方面会比 Nginx 和 Apache 的门槛低很多。由于 OpenResty 底层是基于 Nginx 的,加载新模块时需要重启,与 Nginx 同样存在无法热更新的问题。

综上所述,通过对几款 Web 代理服务器的充分分析和研究发现,近几年比较流行的 Web 代理服务器在扩展性上都有一定的弊端,无法满足快速动态扩展的需求。本文将针对现有 Web 代理服务器在扩展性上的不足,提出了一种企业 API 网关的热插拔插件的解决方案。

### 3 方案描述

企业 API 网关热插拔插件方案主要包括两个方面:首先是插件本身的热插拔,即在不干扰现有业务的前提下,实现对网关插件的动态更新和无感知扩展;其次是插件配置的热加载,使得在网关运行时能够即时、动态地更新插件配置。OpenResty 作为 Nginx 的卓越延伸,继承了 Nginx 的高性能,同时融合了强大的扩展能力。本方案选用 OpenResty 作为底

层基础,通过 Lua 实现 API 网关热插拔插件的运行框架,使企业 API 网关具备热插拔插件的扩展能力。

#### 3.1 架构设计

企业 API 网关整体架构由控制平面、数据平面、共享存储和配置中心 4 部分组成,其逻辑架构如图 2 所示。

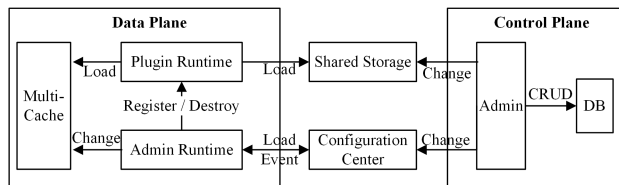


图2 API网关逻辑架构

Fig. 2 API gateway logical architecture

网关控制平面负责将变更的插件配置持久化到数据库后推送到配置中心,并在网关插件发生变更后,将最新的插件源代码更新到共享存储,同时还提供网关插件配置的可视化管理能力。

网关数据平面基于 OpenResty 构建,其主要任务是根据网关控制平面传递的插件配置,在请求处理时运行相应的网关插件。网关数据平面的核心构成包括插件运行时、Admin 运行时和多级缓存。插件运行时为网关插件提供统一的运行环境,管理插件生命周期,包括注册、卸载、执行以及协作等动态能力;Admin 运行时监听配置中心的变更事件<sup>[2]</sup>以及管理网关配置,同时,插件运行时也会通过 Admin API 实时加载最新的插件,以响应插件变更。多级缓存用于存储所有配置信息。引入多级缓存的目的是提升配置检索效率,避免频繁查询配置中心,从而降低 IO 操作带来的处理延迟。

共享存储用于共享网关插件的源代码文件,而配置中心主要负责配置统一管理以及配置信息的传输。

##### 3.1.1 Plugin Runtime

网关插件运行时主要由插件控制器、插件过滤器以及插件执行器 3 部分构成,它们共同构建了插件的运行环境,提供插件运行态的生命周期管理。插件控制器负责整体协调和管理,包括插件配置检索、实例创建、链式编排等;插件过滤器通过准入判断确保只有符合条件的插件被执行;插件执行器负责实际执行、初始化、资源回收和日志记录。将插件运行时嵌入 OpenResty 的生命周期中,通过这种模块化、可扩展的设计提高了系统的灵活性、可维护性和性能,使插件开发、配置和

管理更加高效、精确和可控。其整体架构如图 3 所示。

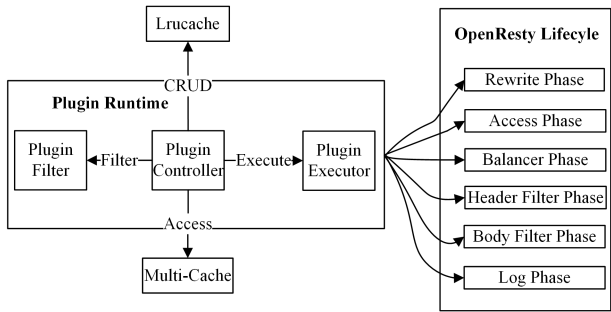


图 3 插件运行时系统架构

Fig. 3 Plugin runtime system architecture

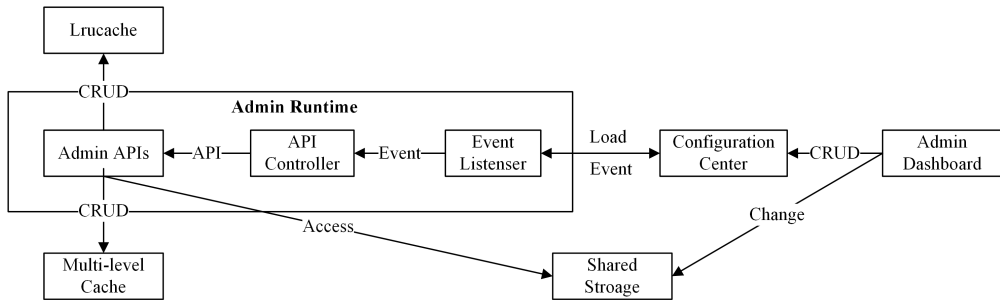


图 4 Admin 运行时系统架构

Fig. 4 Adminruntime system architecture

### 3.2 插件的内部结构

网关的热插拔插件基于 OpenResty 实现,插件的运行主要是依赖插件本身的钩子函数。网关插件的钩子函数和 OpenResty 中的生命周期是一一对应的,每个网关插件的钩子函数会挂载在同名的 OpenResty 生命周期中。

将网关插件的执行与 OpenResty 生命周期相结合具有以下优势:首先,网关插件可以在合适的阶段执行业务逻辑,使得插件能够在请求处理的不同阶段完成特定任务,从而满足多样化的需求;其次,插件的钩子函数与 OpenResty 生命周期一一对应,确保插件的执行与网关请求处理完全同步,避免了不必要的延迟或错位;此外,这种设计允许灵活的插件组合,使得多个插件能够协同工作,实现更复杂的功能。其内部结构图如图 5 所示。

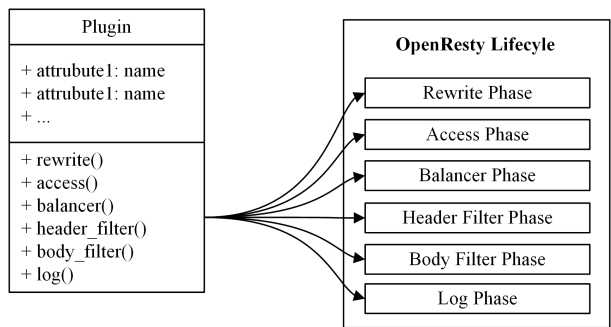


图 5 插件内部结构图

Fig. 5 Diagram of plugin internal structure

### 3.3 插件的生命周期管理

插件的生命周期管理包括插件运行态管理和插件管理态管理。插件的运行态管理由插件运行时全权负责,插件运行时为网关插件提供统一的运行环境,主要负责插件的加载、运行等运行态功能;插件的管理态管理负责插件

### 3.1.2 Admin Runtime

网关 Admin 运行时(Admin Runtime)内部主要分为网关资源 API、API 控制器和事件监听器 3 部分,实现了资源管理和插件管理的有效分离。事件监听器负责监听配置中心的变更事件,触发配置更新,并通过 API 控制器协调处理;API 控制器则负责解析事件消息、调用资源管理 API,并记录执行结果以进行运营审计;资源管理 API 定义了资源变更的逻辑处理,通过适配不同资源类型和执行动作,实现了资源配置的变更。网关 Admin 运行时架构设计提供了高效、可控的 Admin 运行时,使得资源管理和插件管理的操作模块化、更灵活并易于维护,同时记录日志信息,有助于问题追溯和系统审计,提升了整体系统的可靠性和可维护性。其整体架构如图 4 所示。

的注册、注销、更新、配置管理等管理态功能,由网关的 Admin 运行时与插件运行时协同处理。

### 3.3.1 插件加载

插件加载是网关插件在运行态的情况下通过插件配置和请求的上下文信息一步一步动态加载网关插件的过程,主要涉及插件运行时中插件控制器和插件过滤器。其时序图如图 6 所示。

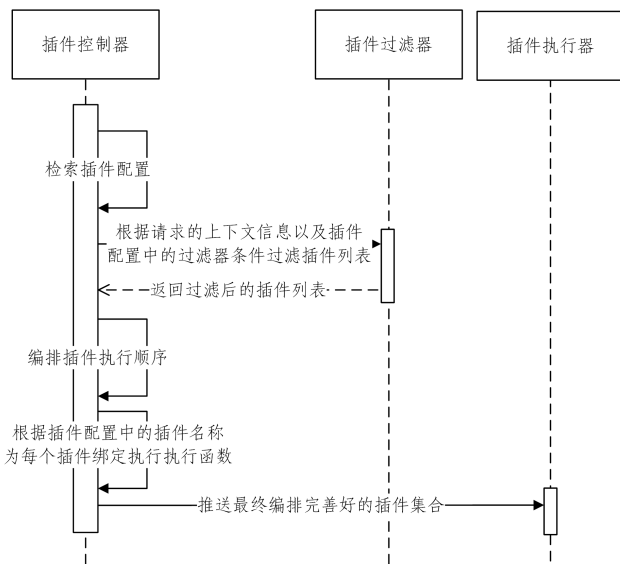


图 6 插件加载时序图

Fig. 6 Sequence diagram of plugin loading

### 3.3.2 插件执行

插件执行是插件运行时完成网关插件的加载和过滤后,针对真实被执行的插件在插件执行器中一步一步执行的过程。此步骤主要涉及网关插件运行时中的插件执行器。其时序图如图 7 所示。

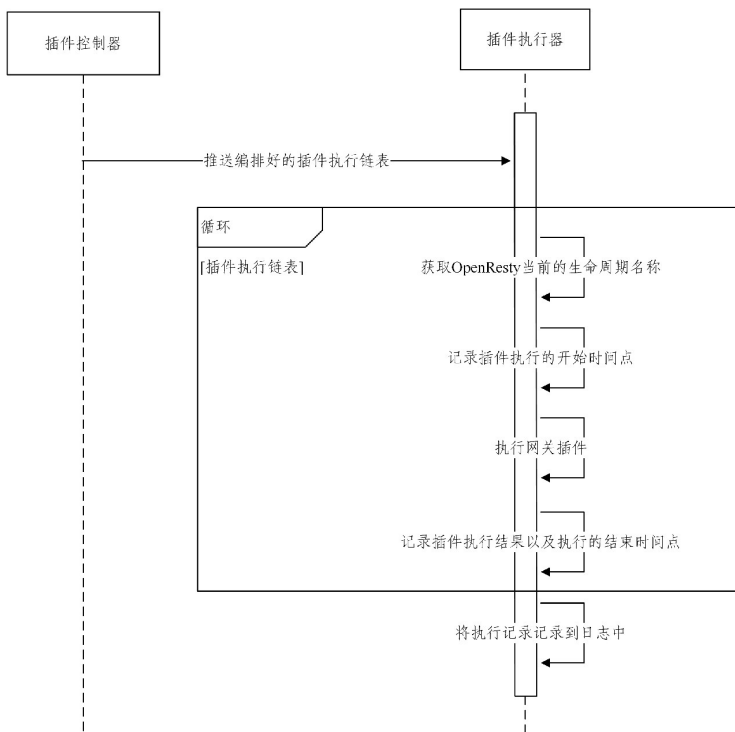


图7 插件执行时序图

Fig. 7 Sequence diagram of plugin execution

3.3.3 插件注册

插件注册是将新的网关插件增加到 API 网关时, API 网关中的各个组件协同工作来达到网关插件动态新增的

过程。此步骤涉及网关控制平面和网关数据平面中的各个组件。

其时序图如图 8 所示。

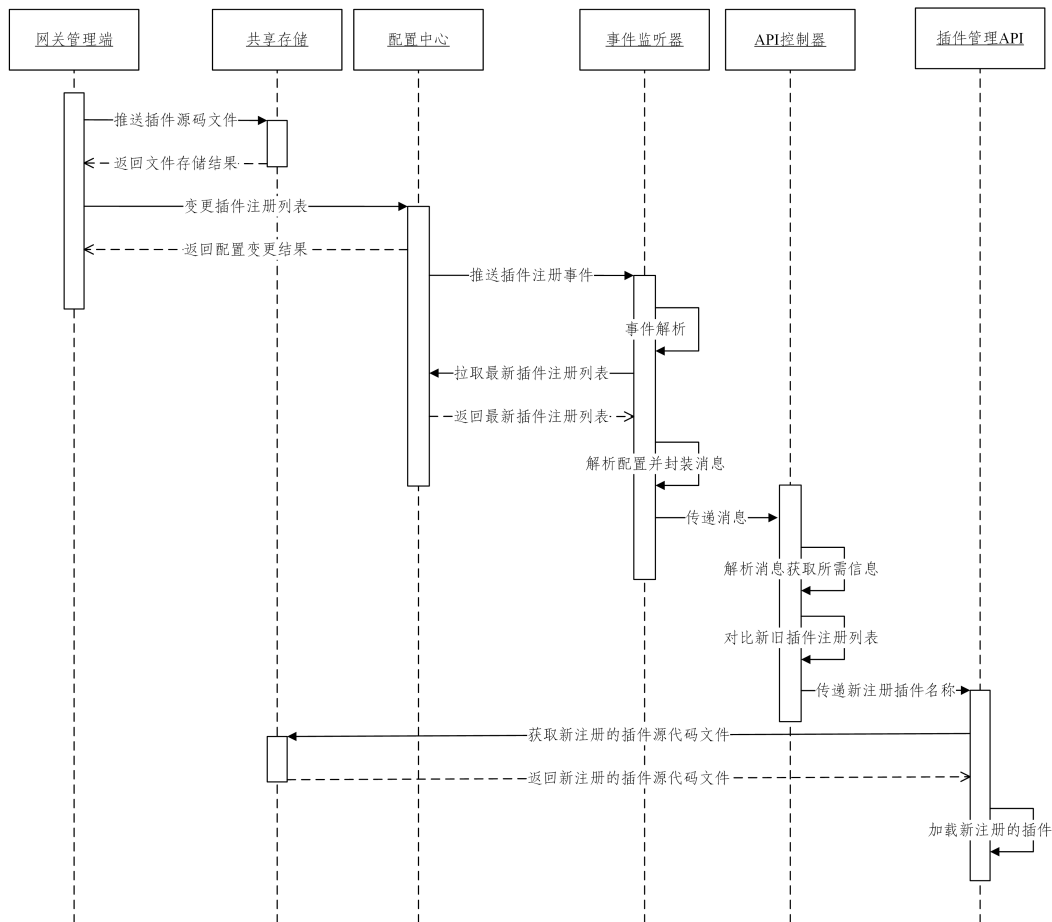


图8 插件注册时序

Fig. 8 Sequence diagram of plugin register

### 3.3.4 插件注销

插件注销是当不再需要某个网关插件时,API 网关中的各

个组件协同工作来动态删除网关插件的过程。此步骤涉及网关控制平面和网关数据平面中的各组件。其时序图如图 9 所示。

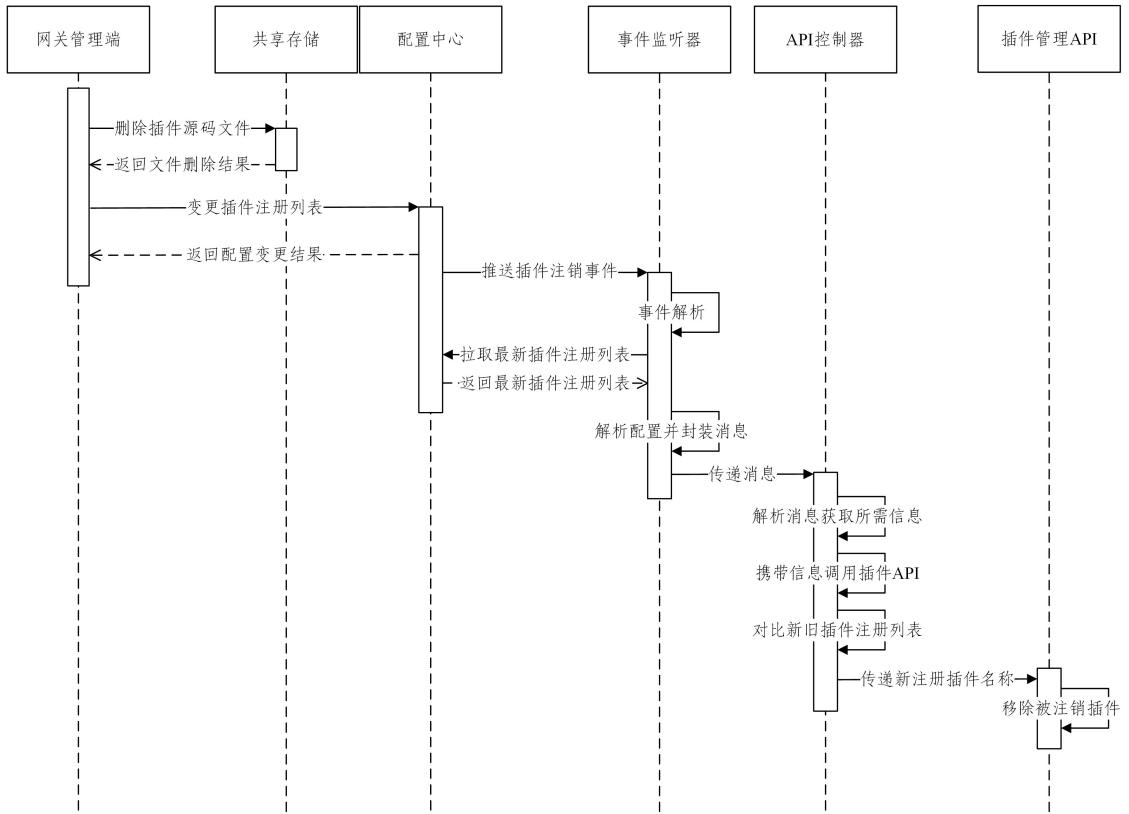


图 9 插件注销时序

Fig. 9 Sequence diagram of plugin destroy

### 3.3.5 插件更新

插件更新是当某个插件的功能逻辑发生变更时,API 网关中的各个组件协同工作来动态更新网关插件的过程。这一

过程尤为关键,处理不当可能会导致插件的执行结果和预期不符。此步骤涉及网关控制平面和网关数据平面中的各组件。其时序图如图 10 所示。

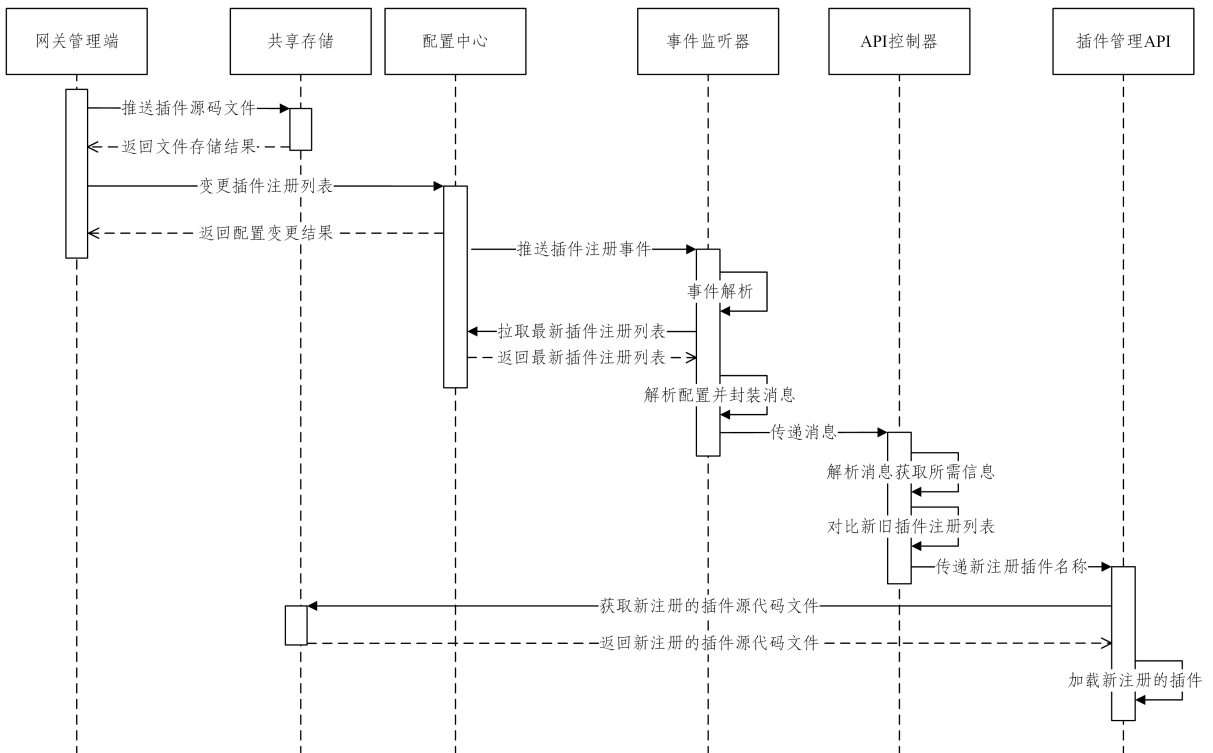


图 10 插件更新时序图

Fig. 10 Sequence diagram of plugin update

3.3.6 配置管理

插件配置管理主要负责运行态插件配置动态更新。插件配置热更新是网关插件在网关日常运行态的情况下最普遍的操作，

网关运维人员需要根据业务变化不定时地调整网关插件的运行配置以达到预期。网关插件配置的热更新涉及网关数据平面和网关控制平面的多个组件协同工作。其时序图如图 11 所示。

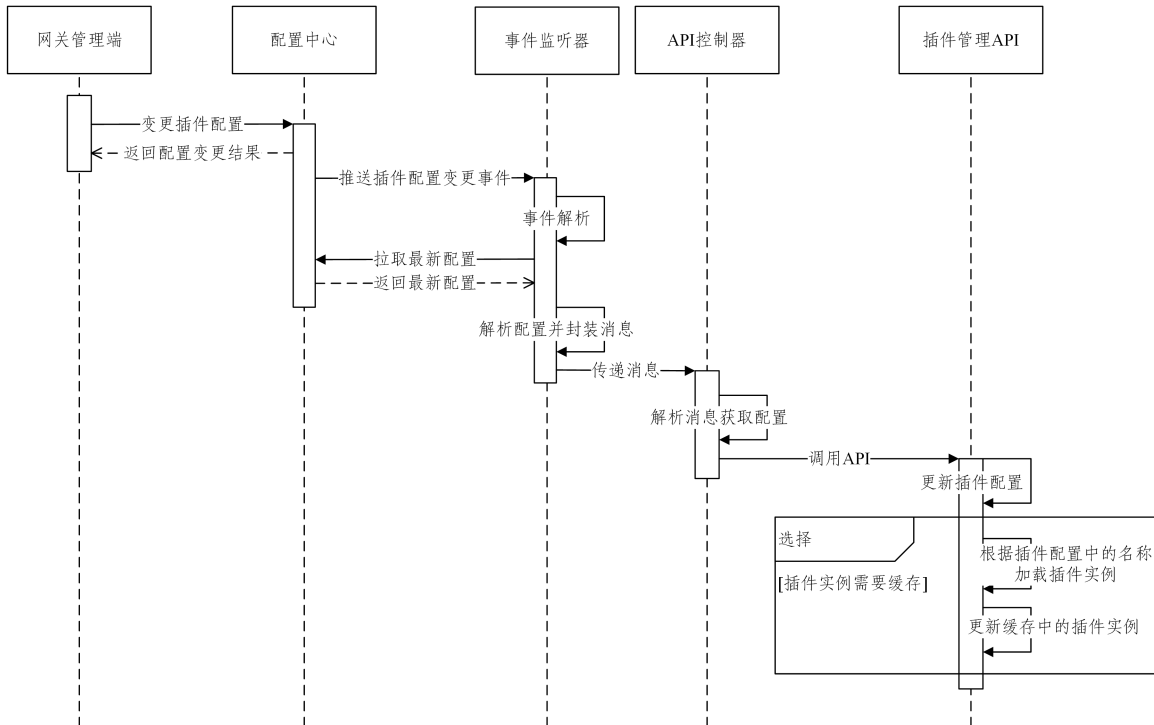


图 11 插件配置管理时序图

Fig. 11 Sequence diagram of plugin configuration management

4 实验验证

4.1 实验设计

在 Kubernetes 环境中部署 API 网关以及使用 Go 语言编写一个 Web 服务, API 网关代理 Web 服务后, 使用 WRK 工具执行基准测试, 对比 API 网关在不同资源配额下开启网关插件前后的 QPS 变化。网关插件选择 Proxy-Rewrite 插件, 该插件主要是对原始请求的请求信息进行变更, 该实验的主要目的在于验证网关的热插拔插件在实际运行时是否会对 API 网关的整体性能造成一定影响。

在 API 网关中为特定 API 绑定好网关插件后, 使用 WRK 工具执行基准测试, 持续访问该 API, 同时, 为绑定在 API 上的网关插件准备 10 份不同逻辑的代码实现, 使用 Python 编写的自动化脚本不断对已经绑定在 API 上的网关插件进行变更, 验证网关插件热更新是否会对实际业务的稳定性造成影响。

4.2 结果分析

4.2.1 实验 1: API 网关热插拔插件的运行机制对 API 网关整体性能的影响评估

通过调整 API 网关在 Kubernetes 中的资源配额, 分别在 1C2G, 2C4G, 4C8G, 8C16G 资源下进行了持续 10 min 的实验, 对比 API 网关在单核 CPU 以及多核 CPU 下, 开启网关插件前后的 QPS 变化。实验结果如图 12 所示。

由实验结果可以看出, 在不同资源下, API 网关的插件开启和关闭对 API 网关的整体性能并无明显影响, 这是由于 API 网关的热插拔插件在 API 网关的控制平面进行变更时, API 网关数据平面会将热插拔插件加载到 OpenResty 的 VM

中, 后续在某条 API 使用该插件时, API 网关只需要单纯地从 VM 中执行这个插件即可。其次, 热插拔插件的插件配置也是直接存储在 API 网关数据平面的内存中的, 在执行网关插件时, 加载插件配置的操作是内存层面的操作。因此, API 网关在开启和关闭网关插件时, 不会影响 API 网关的整体性能。

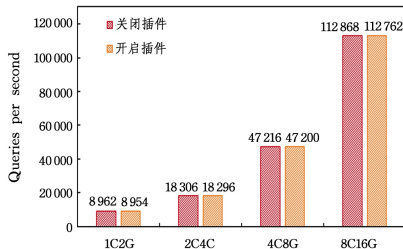


图 12 不同资源下 API 网关的插件开启前后的 QPS 分布情况

Fig. 12 QPS distribution before and after API gateway plugin is enabled under different resources

4.2.2 实验 2: API 网关热插拔插件在频繁变更插件逻辑时对服务稳定性的评估

实验模拟 API 网关在高频运行期间, 使用 Python 自动化脚本频繁更新插件版本的操作。实验结果如表 1 所列。

表 1 模拟频繁更新 API 网关插件版本的测试结果统计

Table 1 Statistics of test results of simulating frequent updates of API gateway plugin versions

实验	实验持续时间/s	版本变更次数	总请求数	请求错误数	正确率/%
第一次	30	10	559 761	0	100
第二次	30	20	539 201	0	100
第三次	30	30	550 772	0	100

变更网关热插拔插件属于 API 网关控制平面的工作,而 API 网关的数据平面和控制平面被设计为相互独立解耦的架构,因此 API 网关在运行态时,频繁变更网关插件并不会对 API 网关的数据面的运行造成影响。实验也证明了上述结论。

综上所述,本文提出的 API 网关的热插拔插件方案在支持网关插件热更新时不会对 API 网关的整体性能造成影响,也不会对业务功能的稳定性造成影响。

**结束语** 目前企业 API 网关已经在几十家大型企业得到应用,提供了身份鉴权、限流限速、协议转换、请求改写等 30 余种热插拔插件。通过企业 API 网关彻底解决了原有 API 网关无法热更新、热部署、难扩展等问题,减少了 40% 的重复开发工作,节省了 30% 运维成本。

API 网关的插件热更新方案仍存在一些不足,目前网关插件实现仅支持 Lua 语言,后续考虑结合 WASM 生态,实现 API 网关的多语言插件扩展,如 Python,Java,C++,Go 等主流开发语言,进一步提升网关扩展能力。WASM 的全称是 WebAssembly,是一种基于堆栈虚拟机的二进制指令格式。WASM 被设计为编程语言的可移植编程目标,目前主流的浏览器都可以基于 WASM 通过高级语言完成对功能的扩展。这项技术目前不仅可以在 Web 端使用,也可以在服务器端运行 WASM 指令。目前 Proxy-WASM 这个开源项目已经开源了对 OpenResty 的 SDK 支持,可以方便地在 OpenResty 中运行由 Go,C++ 等高级语言编译好的 WASM 格式的字节码文件。WASM 不仅仅可以支撑多语言插件,也可以提高 API 网关的安全性,因为 WASM 在执行 WASM 字节码时是在 WASM Runtime 独立的虚拟机中执行的,即使在运行插件时出现问题,也不会影响 API 网关的正常运行。后续的网关插件多语言扩展的研究将进一步促进 API 网关插件的发展,使其在企业级应用中发挥更大的作用。

## 参考文献

- [1] TAO H. In-depth understanding of Nginx: module development and architecture analysis [M]. Mechanical Industry Press, 2016: 2-5.
- [2] ZHOU C, ZHOU X, DU J, et al. System Design and Implementation Based on K8s Microservice Architecture [C] // 2022 4th International Conference on Applied Machine Learning (ICAML). IEEE, 2022: 433-438.
- [3] SONG M, ZHANG C, HAIHONG E. An Auto Scaling System for API Gateway Based on Kubernetes [C] // 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS). Beijing, China; 2018: 109-112.
- [4] MOREIRA P, RIBEIRO A, SILVA J M. AGE: Automatic Performance Evaluation of API Gateways [C] // 2023 IEEE Symposium on Computers and Communications (ISCC). Gammarth, Tunisia; 2023: 405-410.
- [5] ZHAO J T, JING S Y, JIANG L Z. Management of API Gateway Based on Micro-service Architecture [J]. Journal of Physics: Conference Series, 2018, 1087: 032032.
- [6] ZUO X, SU Y, WANG Q, et al. An API gateway design strategy optimized for persistence and coupling [J]. Advances in Engineering Software, 2020, 148: 102878.
- [7] December 2023 Web Server Survey [EB/OL]. (2023-11-14) [2024-01-12]. <https://www.netcraft.com/blog/december-2023-web-server-survey>.
- [8] LI X, FAN J W, WANG F Q. Design and implementation of API gateway system based on OpenResty platform [J]. Informatization Research, 2020, 46(3): 62-68.



**WANG Shengyi**, born in 1981, master, senior engineer. His main research interests include industrial software and industrial Internet platform.