

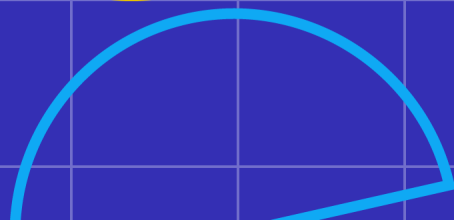
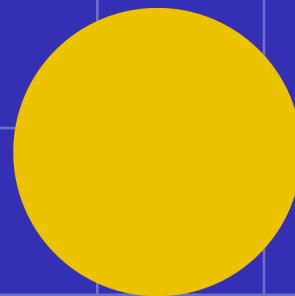
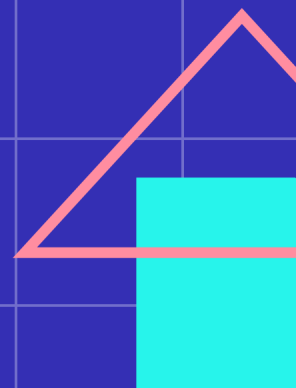


云原生社区
Cloud Native Community

云原生社区 MEETUP

Pipy - 新一代高性能Web

彭柳 Flomesh





全栈云之Flomesh网络解决方案

1. 全栈云

- 1.1 最佳实践/参考思路
 - 1.1.1 多供应商 ④
- 1.2 信创+国产化 ④
- 1.3 方案选择
 - 1.3.1 IaaS + PaaS(容器)
 - 1.3.2 容器+裸金属
- 1.4 多云、混合云、多集群
 - 1.4.1 服务注册、服务发现 ①⑦
 - 1.4.2 网络打通、访问控制 ②③
 - 1.4.3 高可用、容灾 ⑫

2. 低延迟网络

- 智能路由 (就近路由、最短路径)
- 智能DNS (就近解析)
- eBPF加速网络
- 🎧 高性能代理和web--Pipy ** (6.3下午讲)
- 网络可观测、网络性能监控

3. 高并发网络 (互联网/物联网)

- 高性能web
- 高性能代理
- 软件负载均衡
- IoT
 - 单机连接数
 - MQTT
 - 长链接
 - mux/demux
 - 隧道



flomesh-io / pipyPublic

Watch24

Fork58

Star597

<> Code

Issues1

Pull requests

Discussions

Actions

Projects

Wiki

Security

Insights

main3 branches36 tags

Go to fileAdd fileCode

pajama-coder [core] Remove ContextGroup all together with the usage of wait() fi...

✓ a029457 4 days ago1,496 commits

.github

[ci] build x86 binary on lower version OS3 weeks ago

cmake

Updated cmake version and added modules2 years ago

deps

[core] Bump Asio version from 1.24.0 to 1.28.03 weeks ago

docs

[core] Terminology change: muxer group parameter changed to session...3 months ago

gui

[sample] Skip markdown files other than README.md when building the...last month

include/pipy

[nmi] Add pipy_hold/pipy_free/pipy_schedule to NMI for multi-threadin...5 months ago

rpm

[ci]: enable llvm-toolset-7.0 in .spec6 months ago

samples

Merge pull request #154 from naqvis/samples_readme last month

src

[core] Remove ContextGroup all together with the usage of wait() filt...4 days ago

test

[test] Add benchmark test for HTTP/2last week

tutorial

[doc] Add TLS support to the sample and delete old tutorial scripts last year

.dockerignore

[build] Add Dockerfile for building image on Loongarch.last year

.gitignore

[build] Add build option PIPY_RFC89982 months ago

About

Pipy is a programmable proxy for the cloud, edge and IoT.

flomesh.io/pipy

api-gatewayproxyingressgatewayhttp-proxyload-balancerreverse-proxysocks-proxyforward-proxyservice-meshsidecarprogrammable-proxynetwork-proxyprogrammable-networkegresssidecar-proxyingress-gatewayegress-gatewaysoftware-loadbalancer

Readme

View license

Code of conduct

Security policy

Activity

597 stars

24 watching

58 forks

<https://github.com/flomesh-io/pipy>



代理的发展

配置文件时代



第一代代理

配置语言时代



第二代代理

脚本语言时代



第三代代理

集群时代



第四代代理

云时代



linkerd



第五代代理





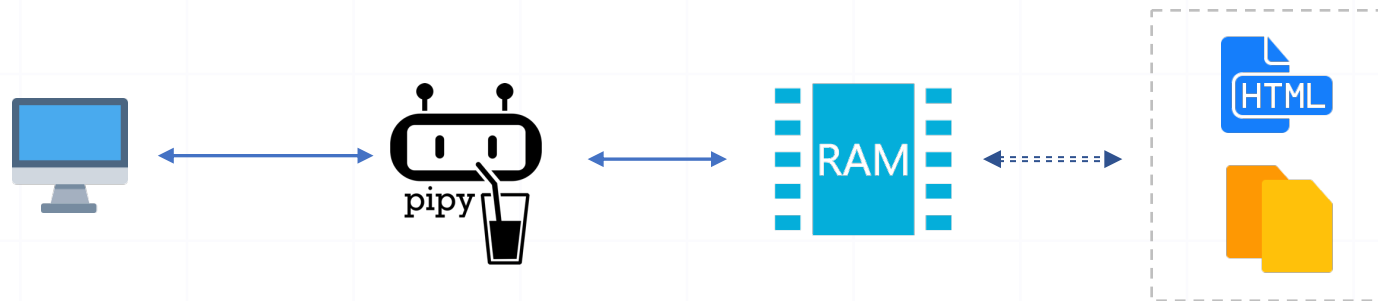
静态内容

Pipy 可以实现如静态站点、CDN 的静态内容处理。

当客户端请求静态内容时，Pipy 会先检查资源如果已经在内存中且有效，会直接返回内容；否则，将读取资源写入内存并记录时间，然后返回给客户端。

更可提供 HTTPS、限流限速等策略。

Name	RPS	P50	P75	P90	P99
Nginx	42675.37	2.17ms	2.52ms	2.89ms	4.15ms
Pipy	56477.58	1.40ms	1.80ms	2.37ms	3.35ms

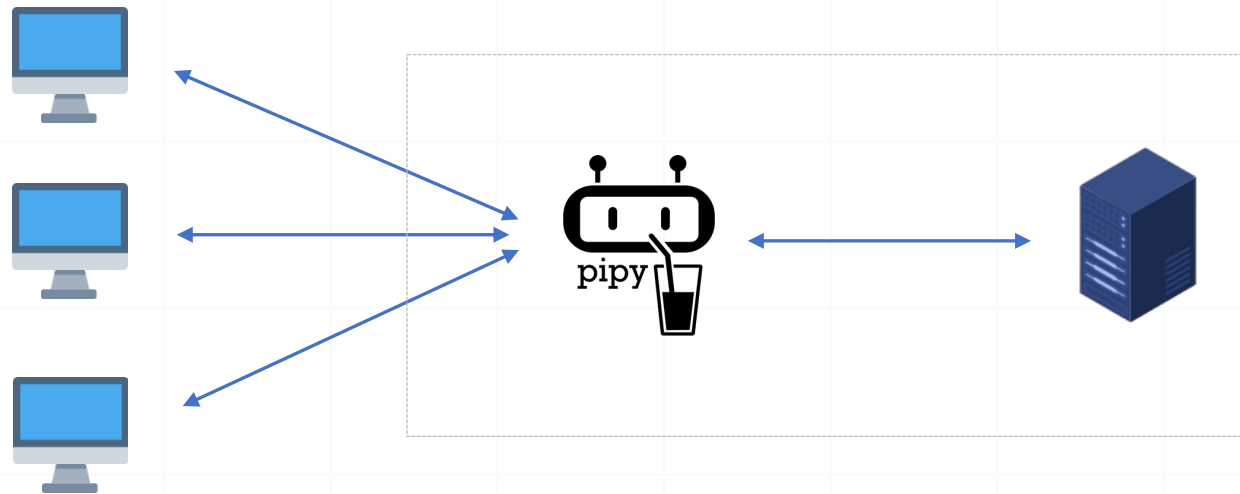




反向代理

反向代理客户端和后端服务器之间的中间层，接收来自客户端的请求，转发给后端服务器。
代表后端服务器提供服务，并隐藏后端服务器的细节。

可以实现负载均衡、SSL/TLS 终止、缓存、访问控制等功能。





L4 负载均衡：NAT 模式

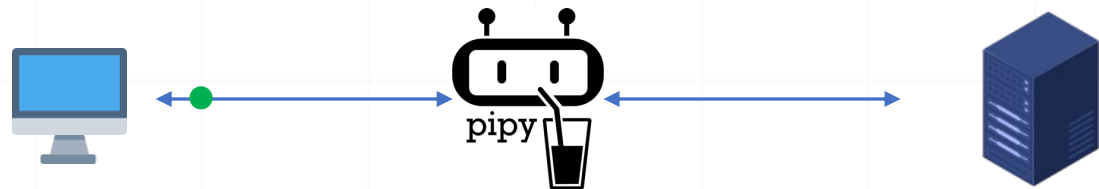
常见的代理模式，客户端请求和服务端的响应都会经过代理。

优点：

- 简单易用，负载均衡器作为中介进行地址转换。
- 安全性好，提供一定的网络隔离，不会暴露后端地址。

缺点：

- 性能较低，负载均衡器需要进行数据包的转发和地址转换，带来额外的开销。
- 扩展性受限，负载均衡器需要处理和转发所有请求和响应数据，性能和容量可能会成为瓶颈。





L4 负载均衡：DR 模式

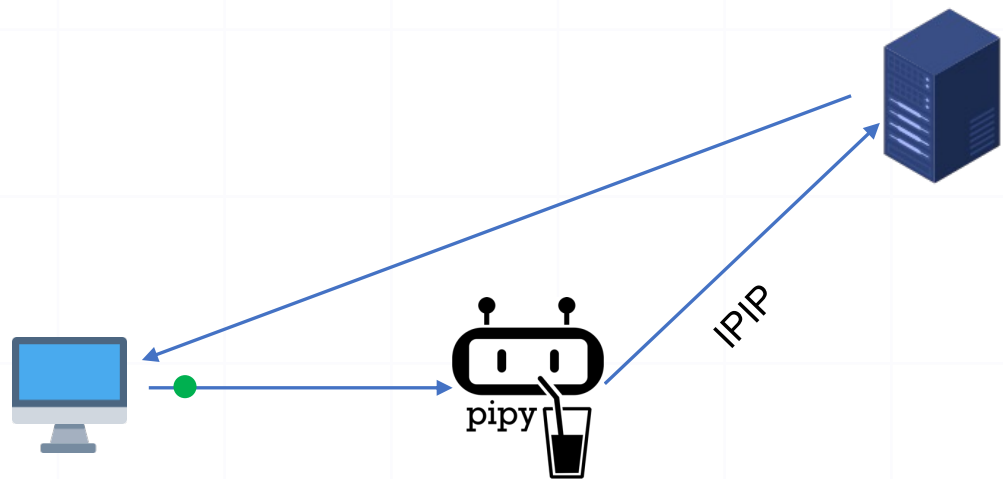
适用于高性能和扩展的场景。

优点：

- 性能高，负载均衡器只会处理请求，服务端将响应
- 会直接发给客户端。
- 扩展性好，负载均衡器无需处理响应，不会成为瓶
- 颈。后端的扩展性和容量决定整理的扩展性和容量。

缺点：

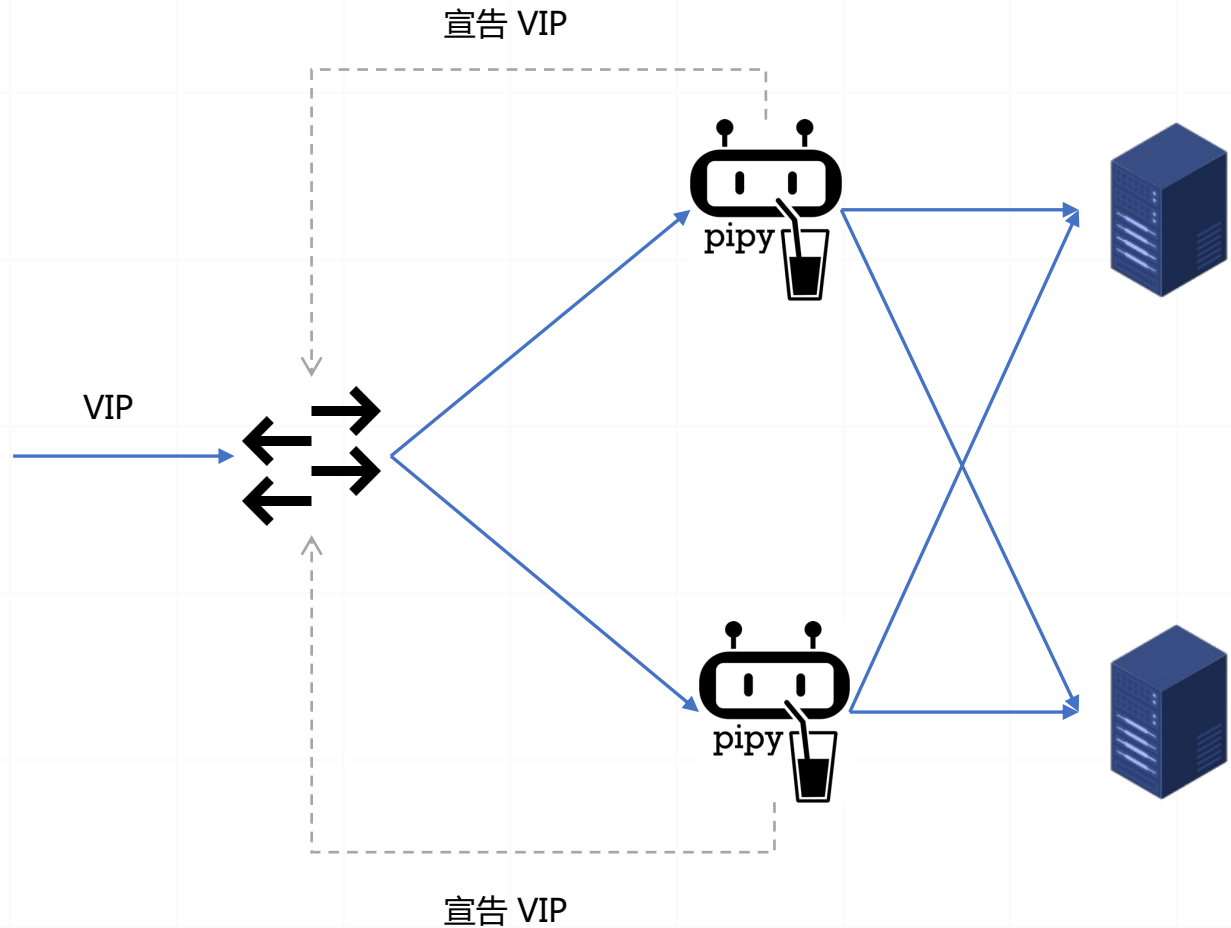
- 网络配置复杂，负载均衡器与后端之间需要特定的
- 网络配置，增加了部署和维护的复杂性。
- 网络隔离问题，负载均衡器无法跟踪响应情况，且
- 会暴露后端服务的地址和端口。





L4 负载均衡：BGP+ECMP

4LB 采用集中部署模式时，一个 Pipy 4LB 集群会通过 BGP 协议向外宣告相同的 VIP，路由器根据宣告信息使用 ECMP 策略将路由包转发到 4LB 集群的不同节点，Pipy 会将请求转发到后端服务器。



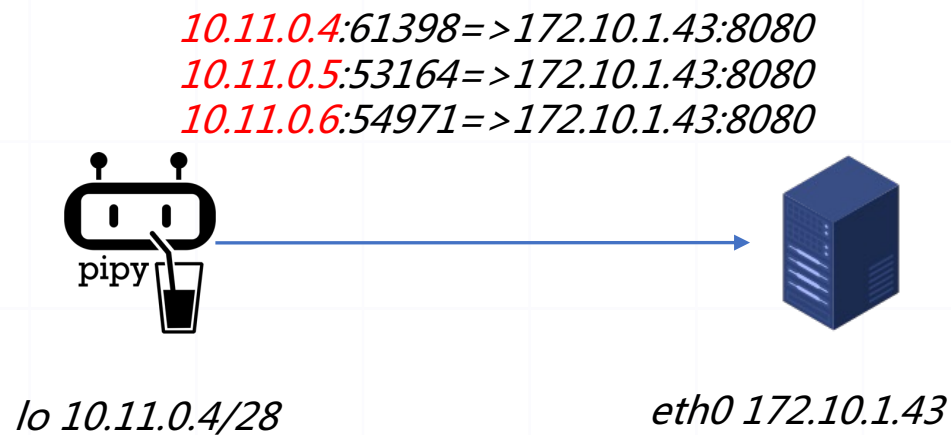


L4 负载均衡：源地址池

每个源 IP 地址允许的并发连接受限，影响性能

- 操作系统限制
- 网络设备限制
- 网络环境
- 应用程序限制

为负载均衡器分配源地址池，最大并发连接数 = 单个 IP 最大连接数 * IP 个数

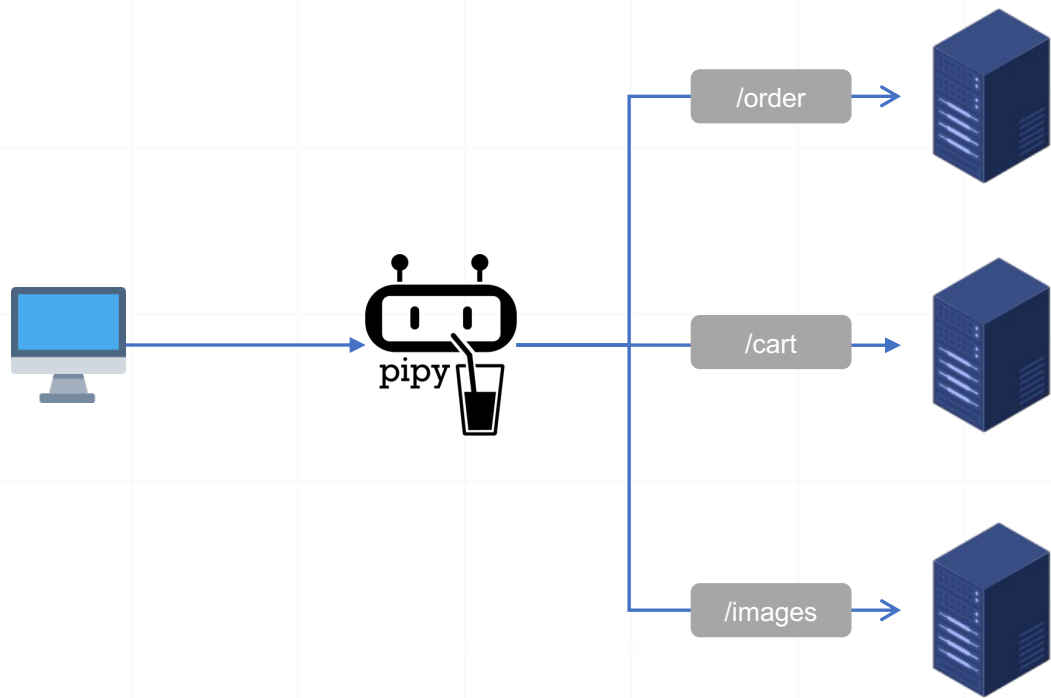




L7 负载均衡：HTTP

负载均衡 HTTP 请求，并支持策略配置：

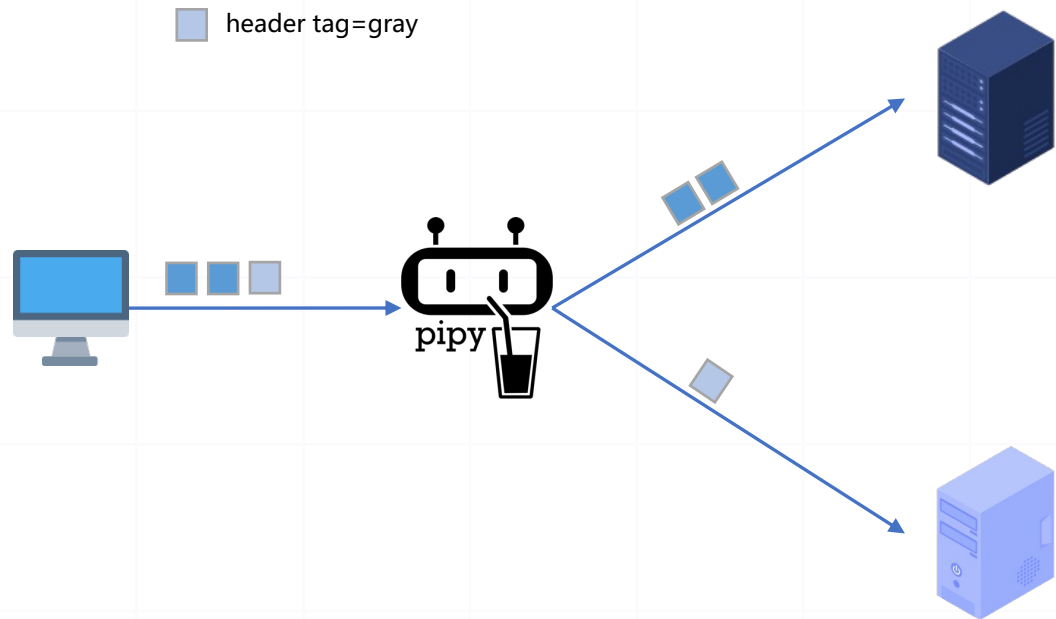
- 负载均衡算法
- 固定/动态路由
- 限速限流
- 认证授权
- 黑白名单
- 健康检查
- SSL/TLS
- 缓存压缩
- 监控
- 日志
- ...





L7 负载均衡：动态路由

- 根据请求的内容进行动态路由：
 - 请求头：host、path、header、params
 - 请求体
- 根据服务器状态
 - 健康情况
 - 负载





L7 负载均衡：多协议

除支持 HTTP 协议外，还原生支持：

- Protobuf
- Thrift
- Dubbo
- BGP
- MQTT
- DNS

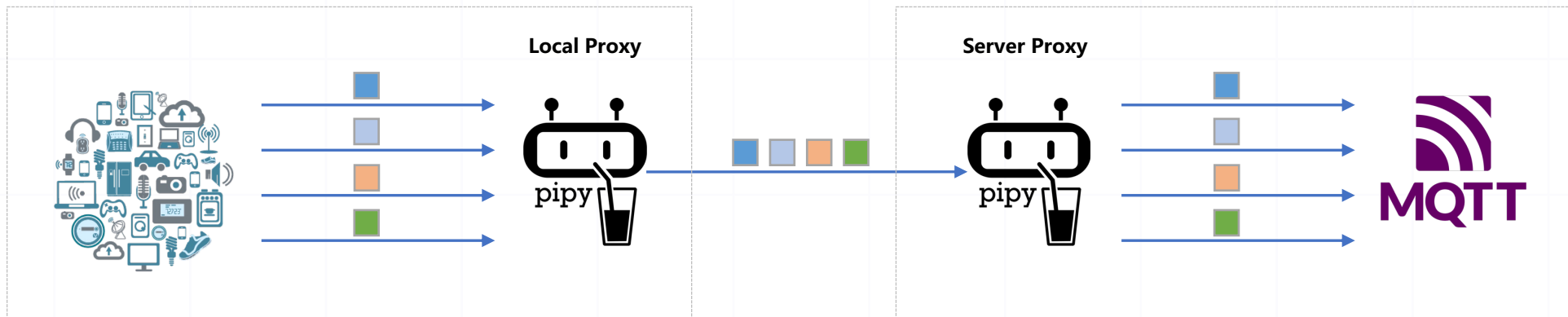
更可通过 [Pipy NMI 特性](#)来扩展支持更多协议。



多路复用

引入本地代理和云端的服务端代理。

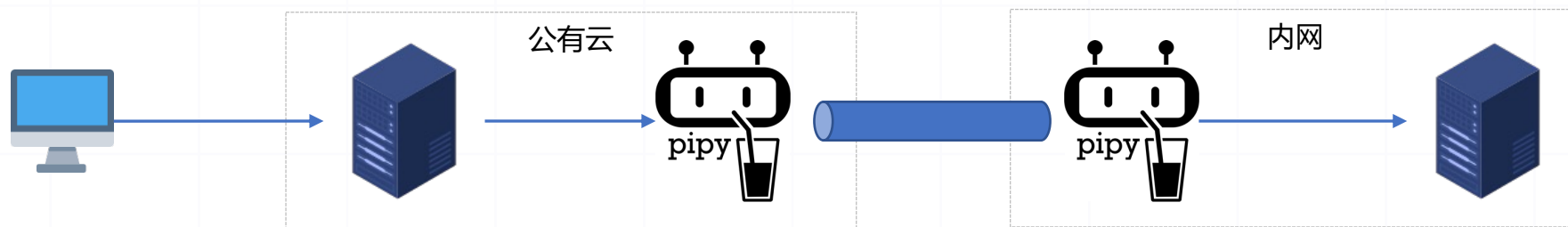
- 在本地使用代理接受 IoT 设备的请求，使用少量的连接发送到服务端
- 在云端，服务端代理根据消息中的客户端 ID 对消息进行分路处理，保证会话状态一致。





隧道

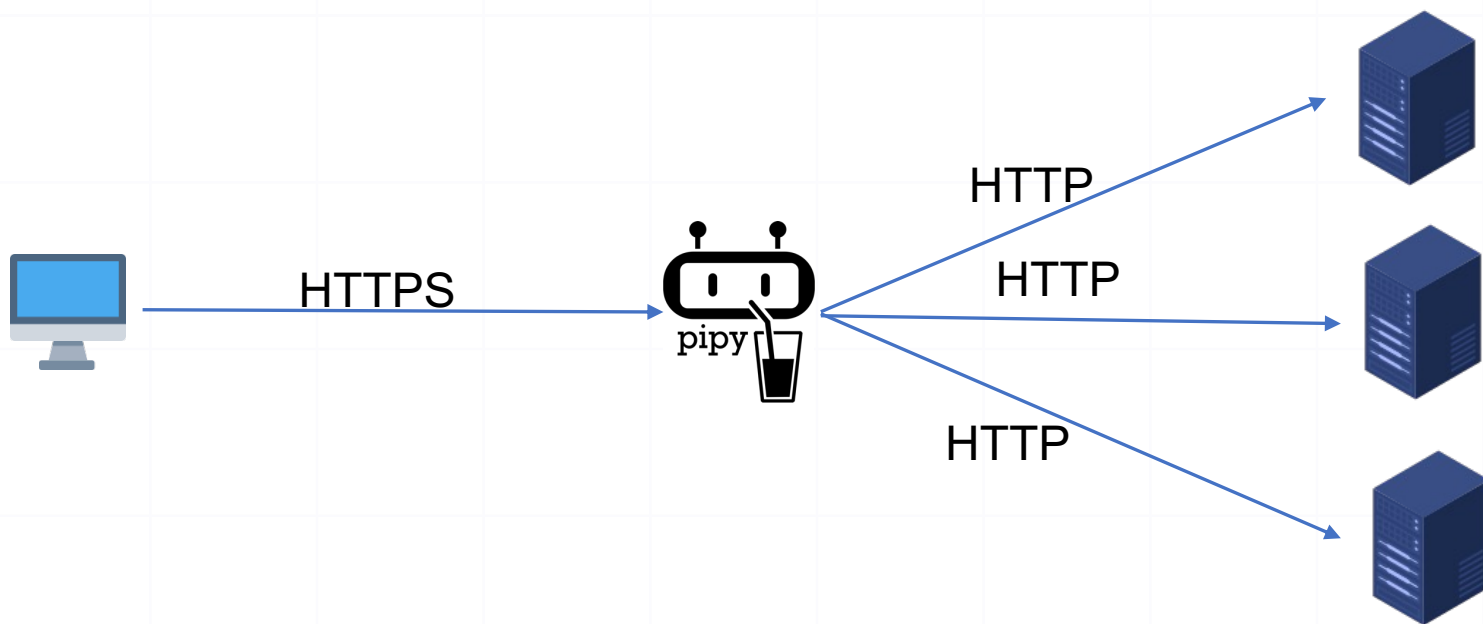
- 简化网络规则配置：打通复杂网络
- 协议转换：传输非 HTTP 协议
- 加密和安全性增加：使用加密的隧道传输非加密的数据
- 跨网络传输：通过互联网连接到企业内部





SSL卸载

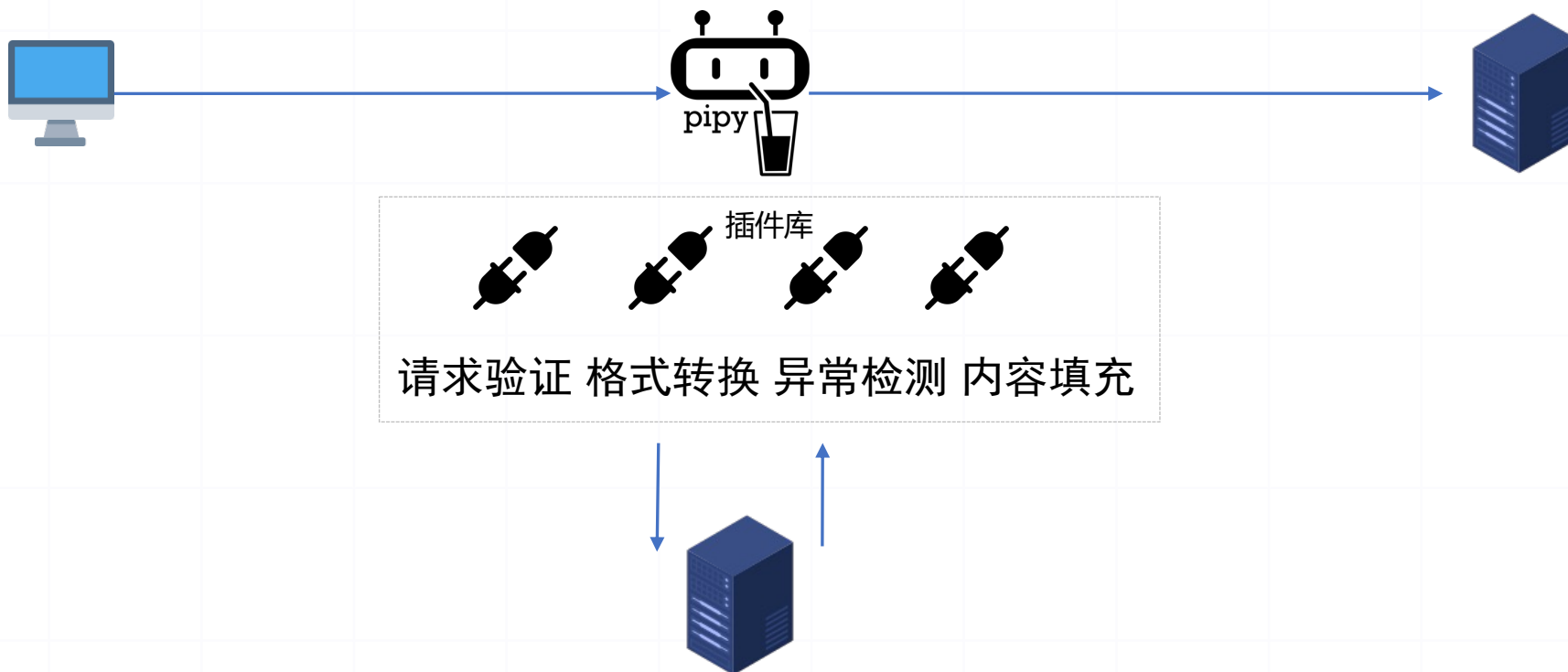
- 提高性能：SSL加解密是CPU密集型的，在网关、代理处卸载SSL，可以使后端专注于业务。
- 提高SSL证书管理能力：统一管理。
- 提高安全性：结合身份验证、IP白名单和SSL可以实现更好的安全性。

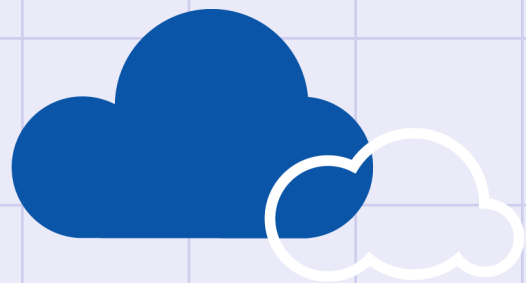




+AI ?

- API请求验证
- 动态格式转换
- 个性化
- 异常检测
- 内容填充





感谢观看



云原生社区
Cloud Native Community

