Microsoft Azure

intel

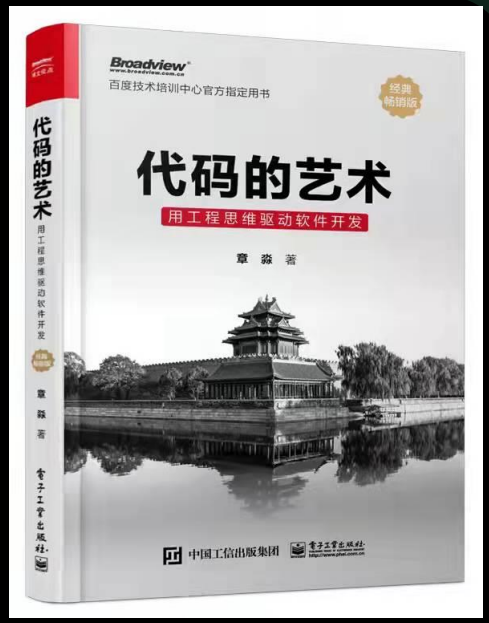
CLOUD NATIVE COMPUTING FOUNDATION

从 负载均衡 到 面向云原生的流量管理平台

章淼
百度 资深研发工程师
2023.4.22

个人简介

- 1994-2004，清华大学计算机系，博士
- 2004-2006，清华大学网络中心，助理研究员
- 1997-2006，清华大学，互联网协议 / 网络体系结构研究
  - 曾参与中国第一代核心路由器研发工作
- 2006-2012，多家公司（搜狗、腾讯等），用户产品研发
- 2012 - , 百度，网络基础架构，软件工程
  - 2012-2020，运维部BFE团队技术负责人
  - 2020-, 百度智能云智能负载均衡团队负责人
  - 2018.1- 2021.10, 百度代码规范委员会主席
  - 2020.10 - , 信通院金融行业开源技术应用社区技术专家



BFE历史背景

百度统一的七层流量转发平台**开始建设** **2012**

BFE => **Baidu Front End**

基于**Go语言**重构，2015年1月在百度**全量上线** **2014 - 2015**

BFE亮相美国**Velocity**大会，成为Go领域**标杆项目**

顺利完成对百度**春晚红包项目**的支持 **2019**

核心转发引擎**对外开源，**被**央视网、360**选用

BFE成为**网络方向中国首个CNCF**官方开源项目 **2020**

BFE => **Beyond Front End，**被**招商银行**选用

每日转发请求**超1万亿**，日峰值超过1000万QPS

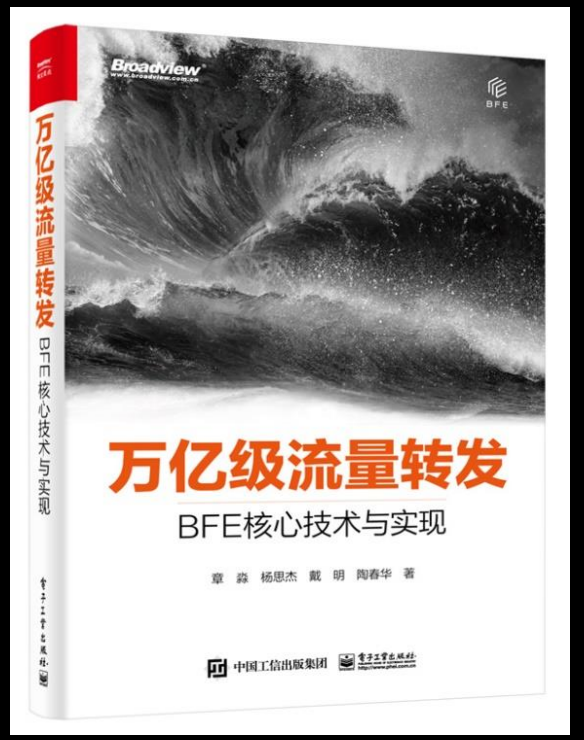
《**万亿级流量转发**：BFE核心技术与实现》正式出版 **2021**

BFE控制面和**BFE ingress**开源

在招商银行**大规模**部署 **2022**

被**广发银行**选用

BFE开源项目被**海外(北美，非洲)用户**采用

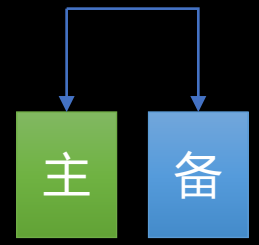


https://github.com/bfenetworks/bfe

负载均衡的现状和问题

基于硬件设备，体系结构落后

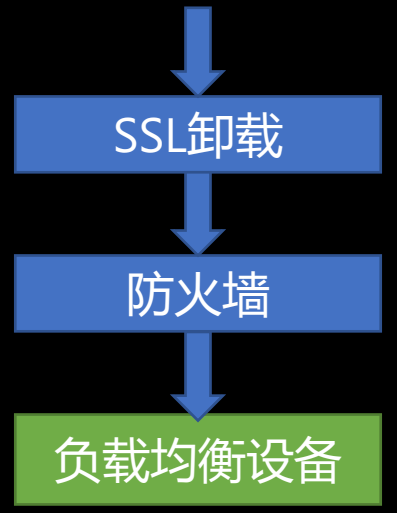
- 购置**成本**高，功能**升级**困难，动态**扩缩容**困难
- **集群化**能力弱，只能支持**主备**模式
- 四七层功能都支持，**七层处理功能/性能弱**
- **数据面和控制面耦合**，扩容和升级困难



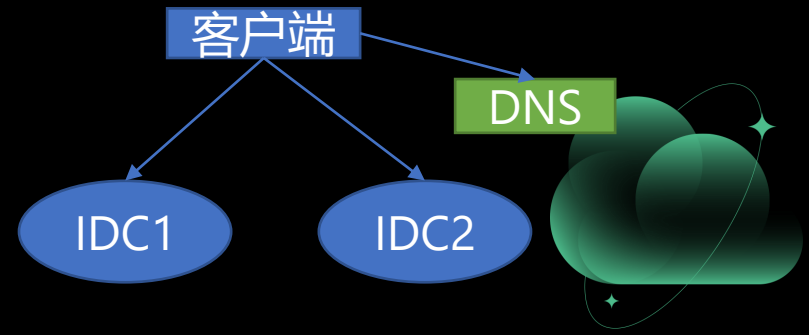
扩容困难，资源浪费

功能落后，无法满足新一代云服务要求

- **多租户**能力弱，**配置变更**困难
- 流量**统计能力**弱，难以支持**精细运营**
- 缺乏**安全能力**，组网难度高，风险大
- **多数据中心**的流量调度能力弱，速度慢，精度低
- 仍然基于SNMP、命令行等**传统管理方式**



架构复杂
增加延迟
易出事故



流量管理平台的需求和发展趋势

负载均衡要为 现代应用 服务

现代应用(Modern App) 的特征

- **Scalability**: 容量可扩展
- **Portability**: 支持多云和混合云
- **Resilience**：高可用，快速恢复
- **Agility**：敏捷迭代，快速更新

为了支持现代应用，需要 **新一代** 的

面向 **云原生** 的 **流量管理平台**

新一代流量管理平台的特征

① **软件化** （容量可扩展，支持多云部署）

② **四七层分离**（容量可扩展）

③ **多主**集群（容量可扩展）

④ **数据平面和管理平面分离**（容量可扩展）

⑤ **多云/多集群调度**能力（多云，高可用，敏捷）

⑥ **强大的路由管理**能力（微服务，敏捷）

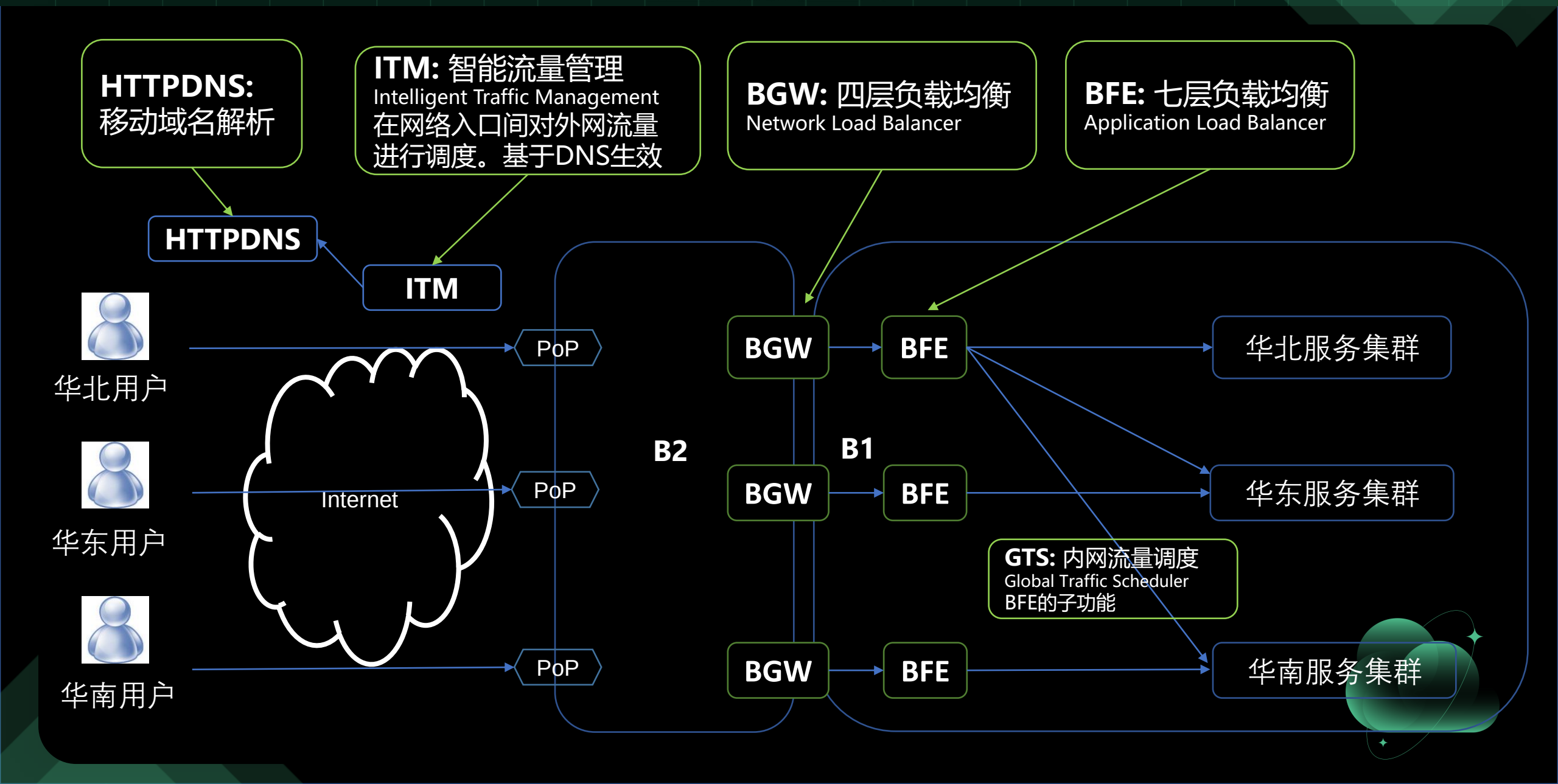
⑦ **流量洞察**能力（高可用，敏捷）

⑧ **安全能力**（高可用）

⑨ **多租户** 能力（敏捷）

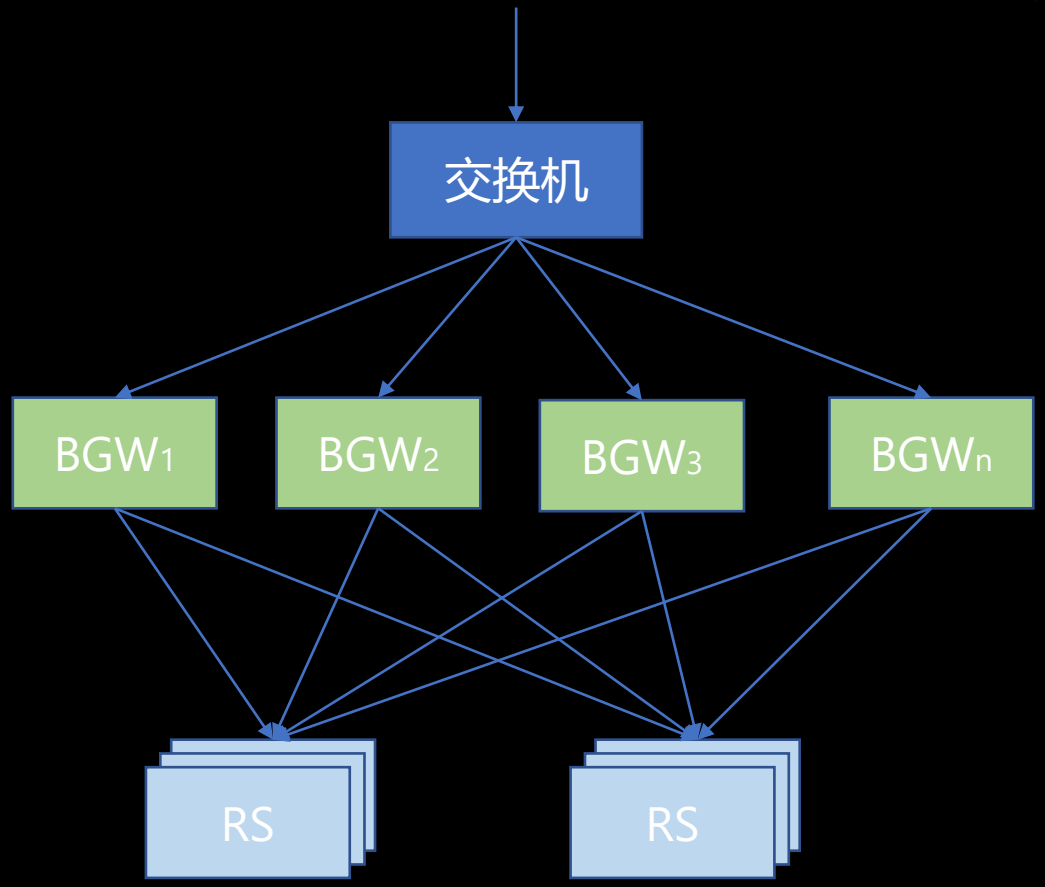
⑩ **平台化 / API接口** (高可用，敏捷)

百度流量管理平台的总体架构



网络负载均衡（BGW）

- **全功能支持**
  - 多种协议支持(TCP/UDP/FTP/QUIC)
  - 灵活的调度算法，多种转发模式，支持复杂组网环境
- **高性能、水平扩容**
  - 基于DPDK，单机容量50G
  - 最大可水平扩容至64台
- **IPv6支持**：NAT66/NAT64
- **安全能力**
  - **防DDoS攻击**：synflood/ackflood/icmpflood
  - **限速**：带宽、每秒新建连接速、每秒数据包限速
- **数据报表**
  - **流量采集**：端口、服务等多维度流量数据采集

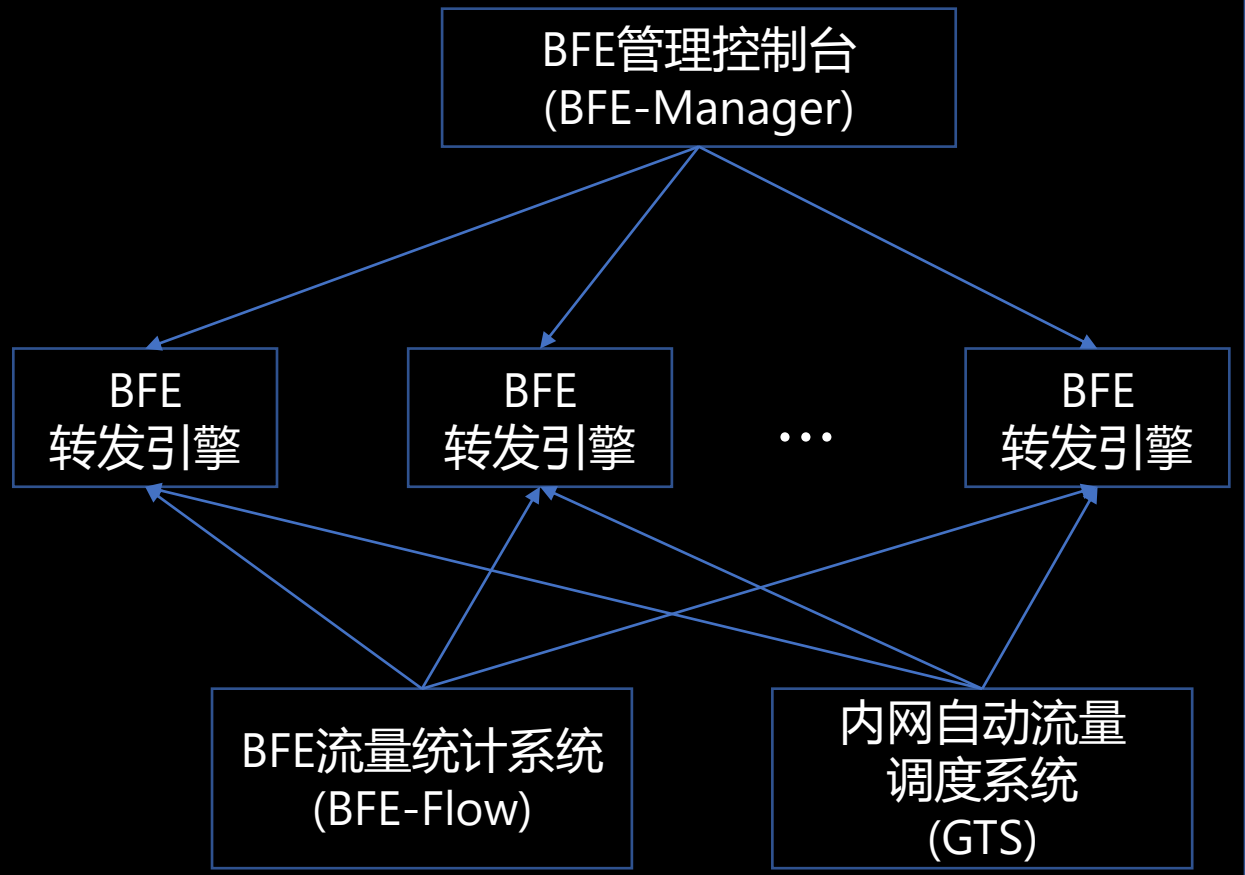


七层负载均衡生态对比

生态	代表项目	说明	性能	安全性/稳定性	开发效率	开源生态	转发延迟
Nginx / OpenResty 生态	Nginx, OpenResty, Kong APISIX	• OpenResty是对Nginx的一种扩展，可以利用Lua语言对Nginx功能做扩展。OpenResty开源项目由中国工程师章亦春创建。 • Kong和APISIX均为API网关开源项目，详情见GitHub	高	低	低	强	低
Envoy 生态	Envoy	• Envoy是基于C++开发的七层开源软件。最早由美国Lyft公司技术团队开发并开源，后Google加入。目前Envoy已经成为服务网格(Service Mesh)中Sidecar网关的重要候选系统。	高	低	低	强	低
Go语言 生态	BFE, Traefik, Tyk	• Traefik为一家法国创业公司推出的七层负载均衡开源软件，详情见 https://github.com/traefik/traefik。 • Tyk为一家英国创业公司推出的API网关开源软件，详情见 https://github.com/TykTechnologies/tyk。	低	高	高	强	低，但有少量长尾
Rust语言 生态	Linkerd	• Linkerd为一家美国创业公司，专注于服务网格方向。其中包含一个使用Rust语言开发的七层负载均衡软件。	高	高	低	弱	低

应用负载均衡（BFE）

- **流量接入**：
  - 多种协议支持。HTTP，HTTPS，HTTP/2，QUIC等
- **流量分发**：
  - 基于HTTP Header、支持丰富语义、各种规则组合的流量分发机制
- **内网流量调度**：
  - 支持跨机房集群级别的自动流量调度
- **安全防攻击**：
  - 应用层精细限流
- **数据分析**：
  - 提供多维度的数据报表，用于分析用户分布、服务状态、网络状态
- **多租户管理**：
  - 可以基于租户粒度对配置和权限进行隔离



应用层路由 - 条件表达式

- **技术**

  -对每个租户提供独立的分流转发表

  -其中采用自研的**条件表达式**描述转发条件

  -内置40多种条件原语，可支持 与/或/非 组合

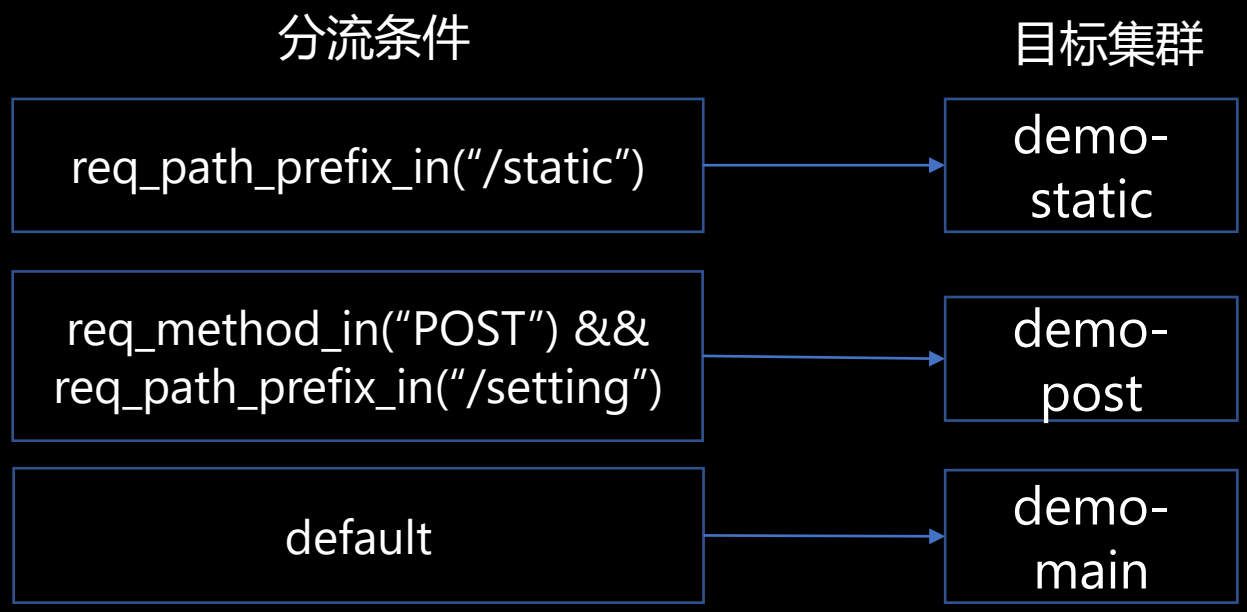
- **优势**

  - 具有强大的转发条件描述能力

  - 相比**正则表达式**，

    (1) 具有更好的**可维护性**

    (2) **无性能退化（恶性回溯）**的隐患



超大规模路由表的支持

基础转发表

匹配条件	目标集群
www.a.com/a/*	Demo-A
www.a.com/a/b	Demo-B
*.a.com/	Demo-C
www.c.com	ADVANCED_MODE

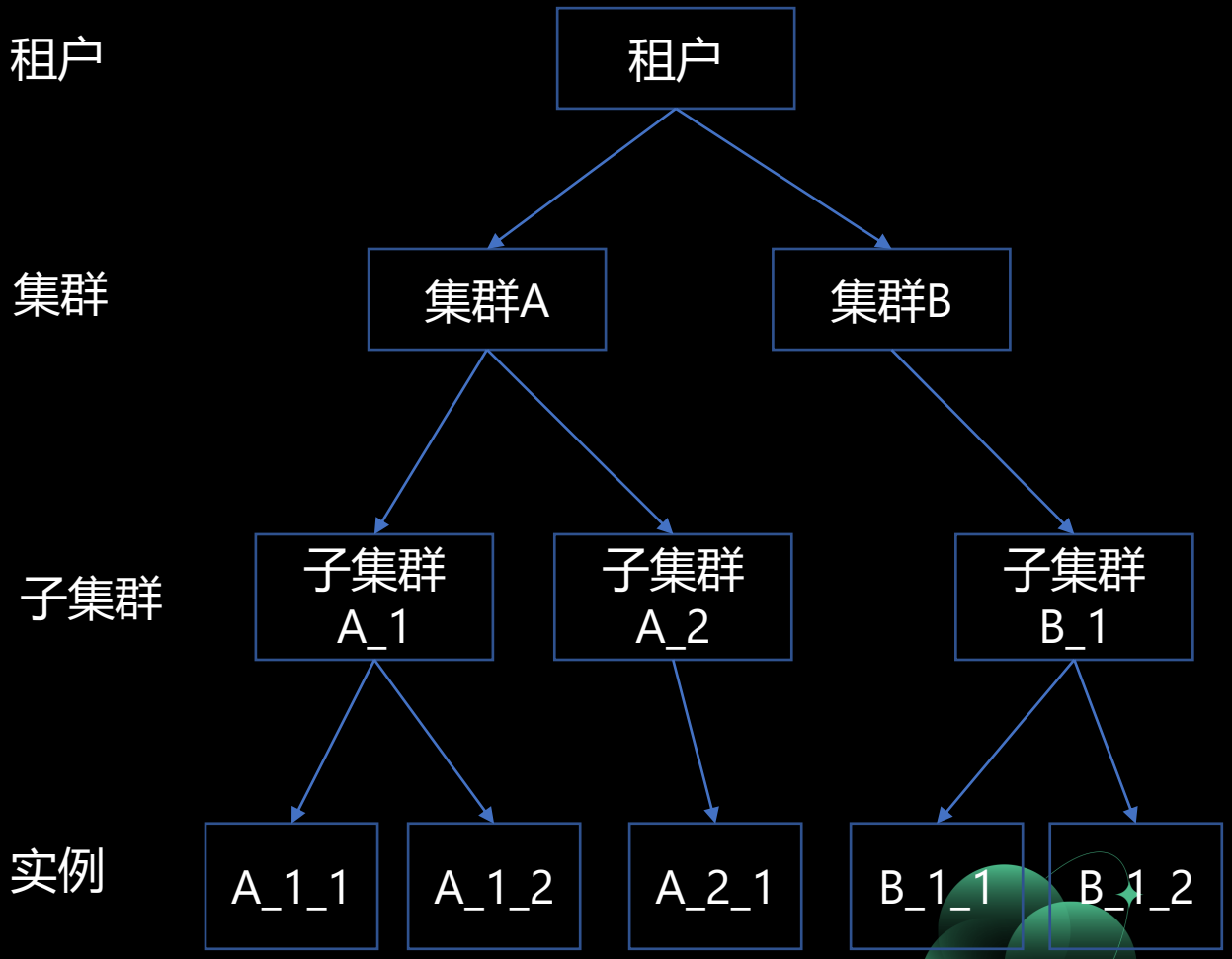
高级转发表

匹配条件	目标集群
req_host_in(“www.c.com”) && req_cookie_value_prefix_in("deviceid", "x", false)	Demo-D1
req_host_in(“www.c.com”)	Demo-D

默认集群 Demo-E

多租户的支持

- **Nginx在多租户支持方面的问题**
  - Nginx引擎未提供多租户支持
  - 配置热加载 => 长连接中断，配置加载开销大
  - 正则表达式的性能隐患 => 单租户配置风险
- **BFE的相关机制**
  - 内置多租户模型
  - 配置热加载不影响长连接
  - 多模块可单独动态加载配置
  - 条件表达式机制 避免 正则表达式的性能隐患



内网自动流量调度(GTS)

- **技术**

  BFE支持按照给定的权重在多个子集群间分配流量

  GTS支持根据流量、容量、机房间距离等因素持续自动计算分流权重

- **优势**

  在流量、容量等发生变化后，可在**20秒内**完成调整

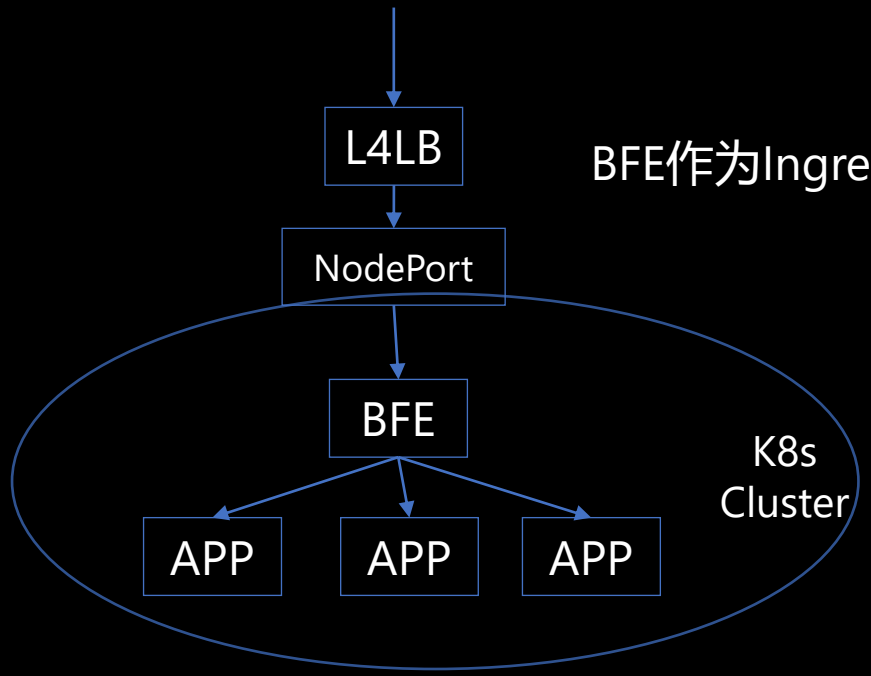
  和传统的**DNS**的方案相比

  (1) 实现子集群间流量的**精确分配**

  (2) 调整时间**大幅缩短**（10分钟 => 20秒）

对Kubernetes的支持

场景1: 在K8s集群之内

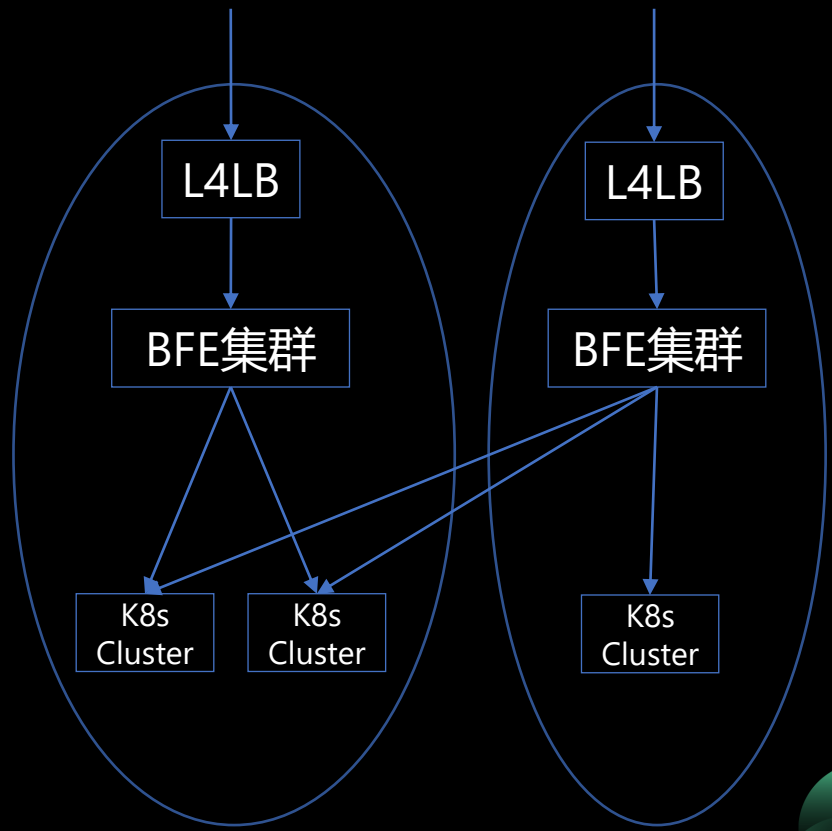


BFE作为Ingress

趋势:

- Ingress => Gateway API
- 增强安全、流量洞察能力

场景2: 在K8s集群之外



BFE作为
全局负载
均衡器

大流量高维度实时报表

- **技术**

  支持分钟级实时流量报表

  业务相关的报表：流量变化(分地域、域名、...)

  下游服务的健康状态：错误率，响应延迟，...

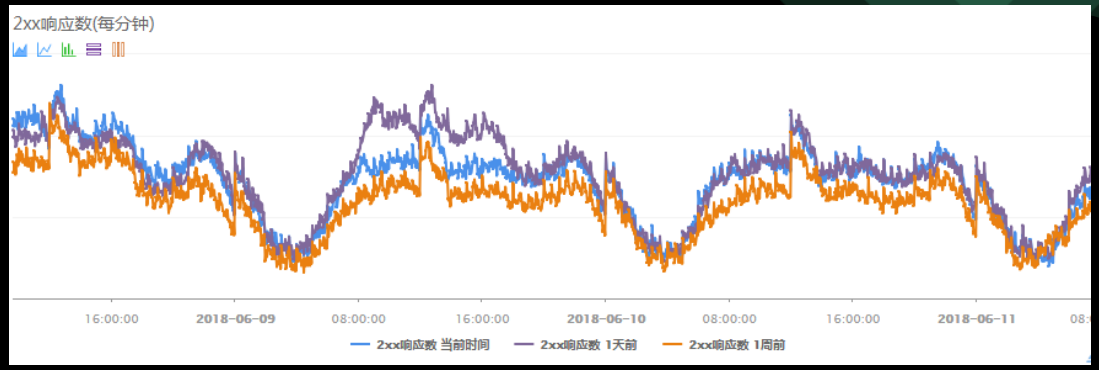
- **优势**

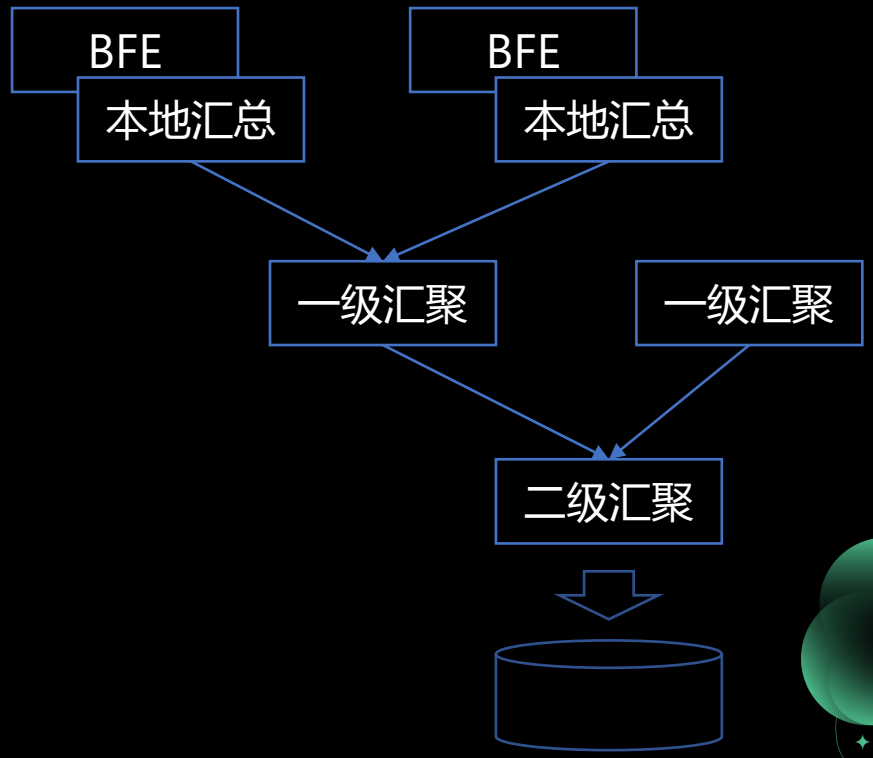
  提升**流量洞察**能力，辅助提升服务质量

  通过配置文件即可增加新报表，**无需开发**

  支持大流量和异地多机房场景，**内网带宽消耗小**

  和**Prometheus**对接，易于能力扩展






```
{" name" : "wan-traffic-by-src" ,
 "dimension" :[ "product" ," region" ],
  "metrics" :[ "req" , "2xx" , " 5xx" ]}
```

新一代的安全架构

- **现有问题**

  SSL**易成为瓶颈**，易受到攻击

  WAF**难扩容**，**变更**易导致**性能下降**、甚至**服务中断**

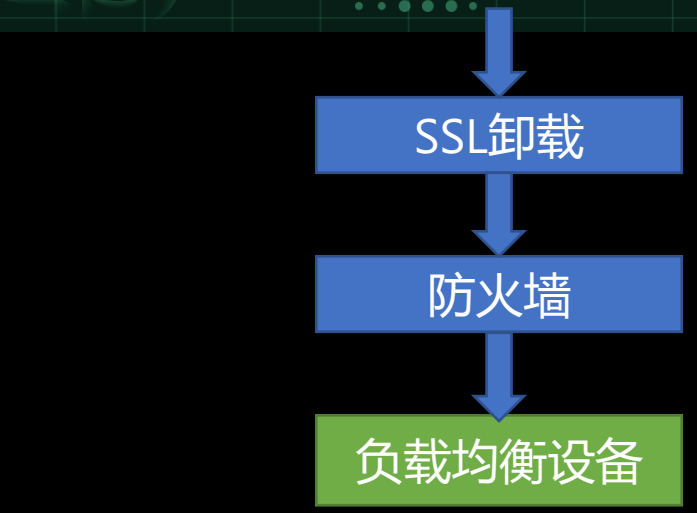
  多层转发**延迟大**

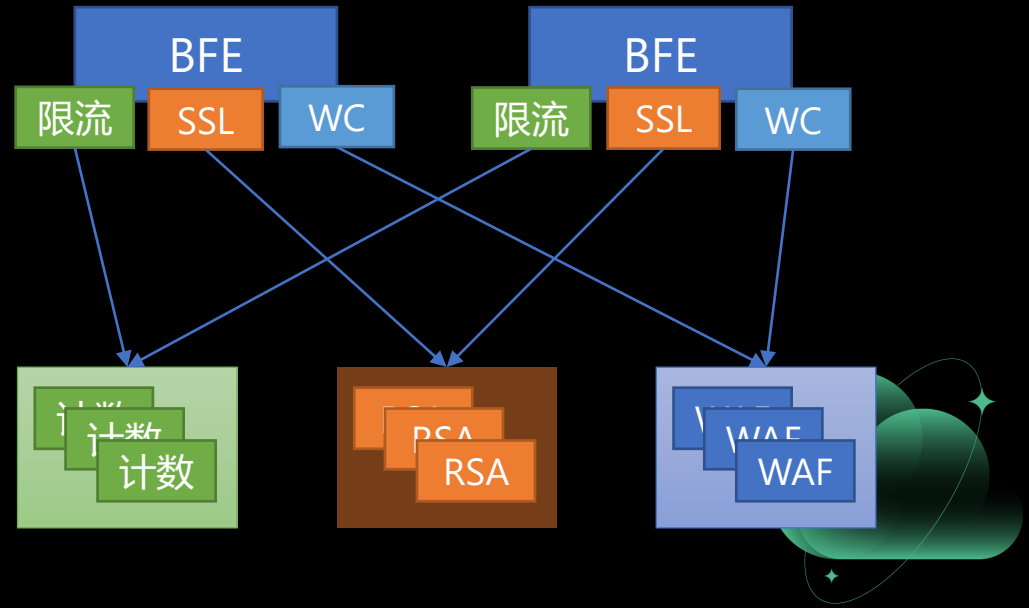
- **优化思路**

  将**安全功能整合**至BFE

  形成**外部资源池**（计数，RSA，WAF）

  WAF处理成为**外部调用** => 保证转发延迟和可靠性





在金融场景的使用

- **需求场景**

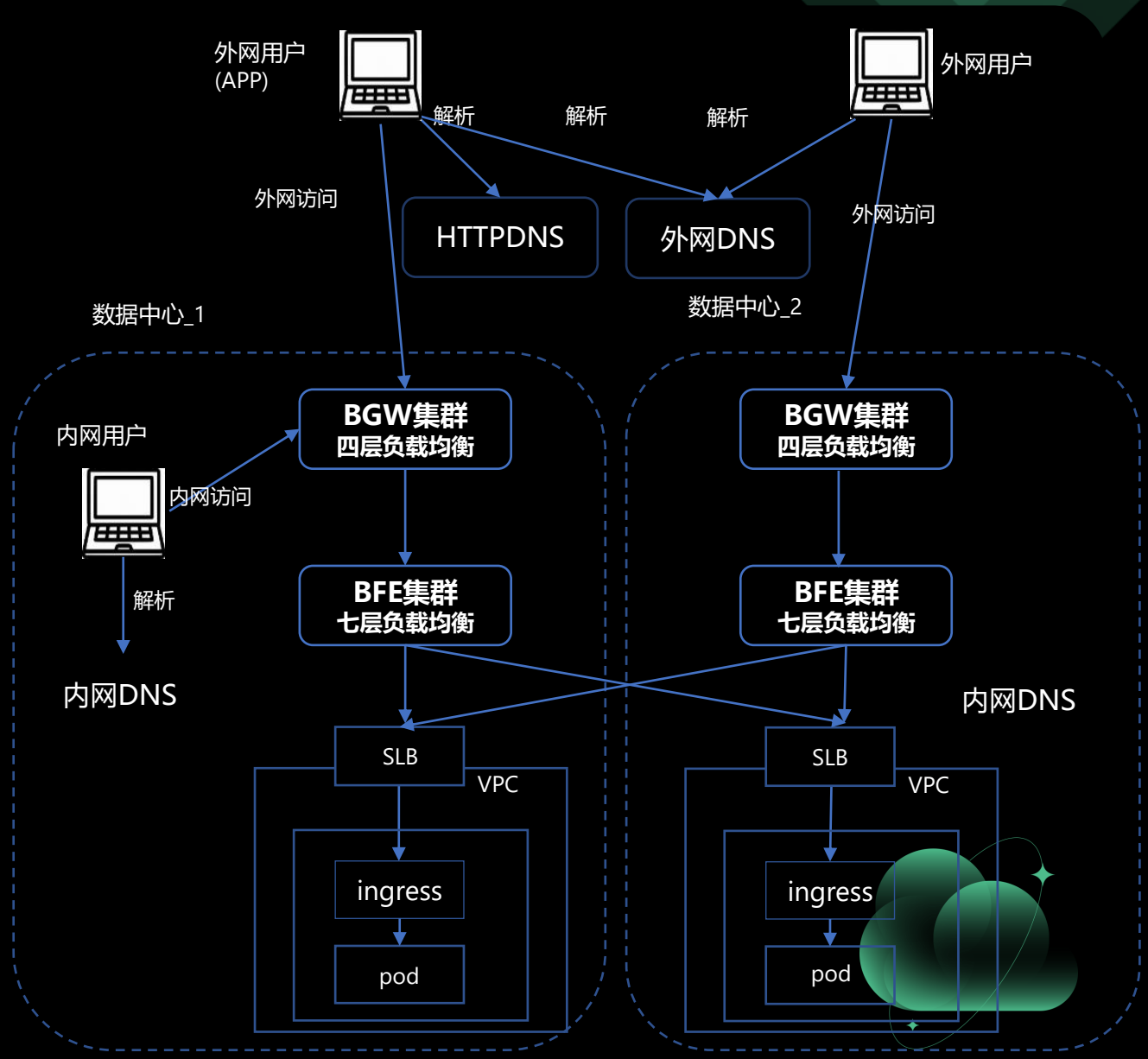
  同城双活，多容器云集群调度

- **收益**

  成本：负载均衡成本降低，应用部署资源节省

  效率：调度能力增强，路由变更速度增强

  可用性：加快止损速度，流量洞察能力增强



延伸：对服务网格的思考

- 服务网格的基本思路

  集中式网关**容量有限** => 使用**分布式**网关sidecar

- 服务网格的**假设存在问题**

  其实，集中式**网关集群**也是可以方便扩容的

- 服务网格带来的问题

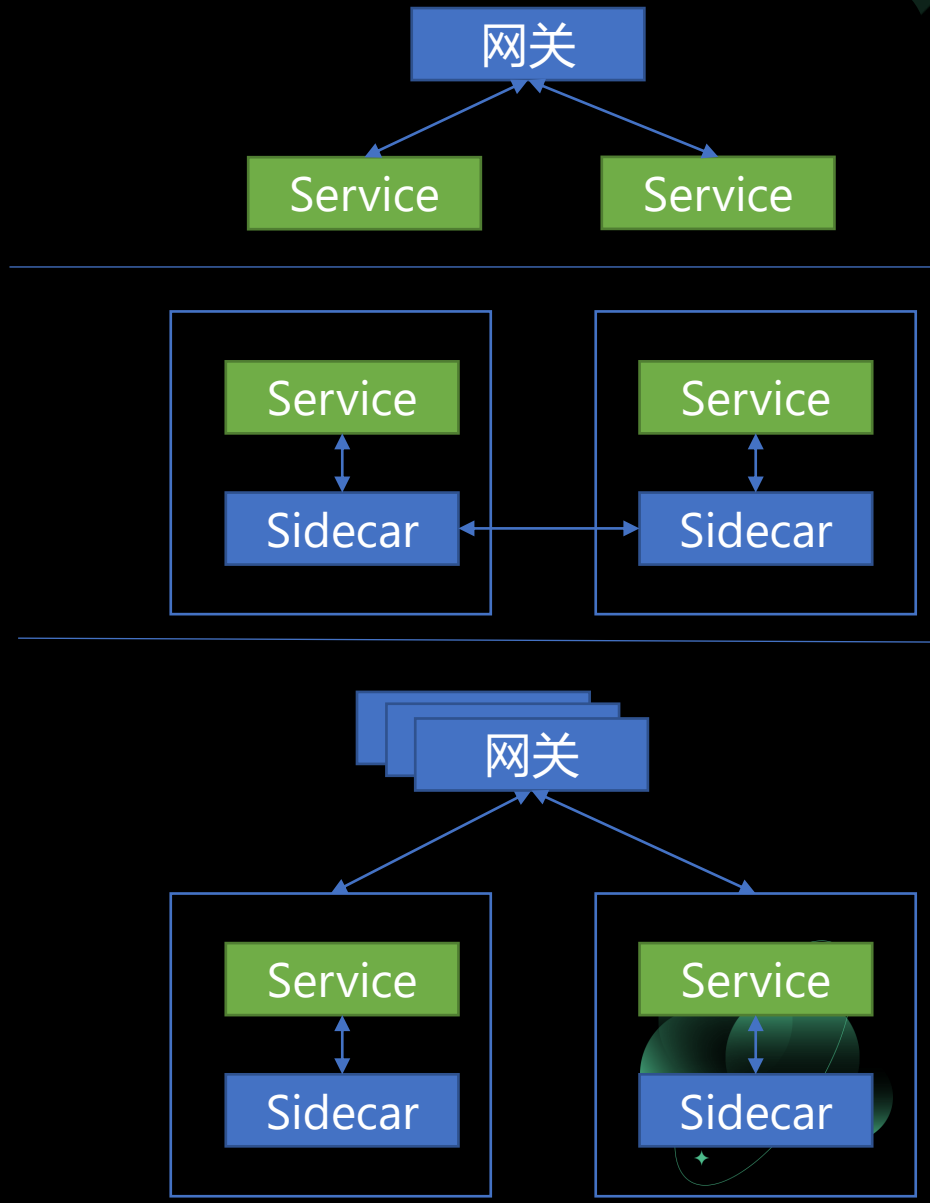
  数量庞大的sidecar，**运维管理**存在严重问题

  sidecar的**路由表**在**可扩展性**方面存在问题

- Sidecar的推荐定位

  不是用于替换集中式网关（路由继续由集中式网关管理）

  而是用于**替换RPC SDK**（相比SDK更容易升级）



总结

- 负载均衡要为 **现代应用** 服务
- 传统的负载均衡技术已**无法满足**现代应用的需求
- 为了支持现代应用，需要 **新一代** 的技术，即

  面向 **云原生** 的 **流量管理平台**

Thanks

Microsoft Azure

intel

CLOUD NATIVE COMPUTING FOUNDATION