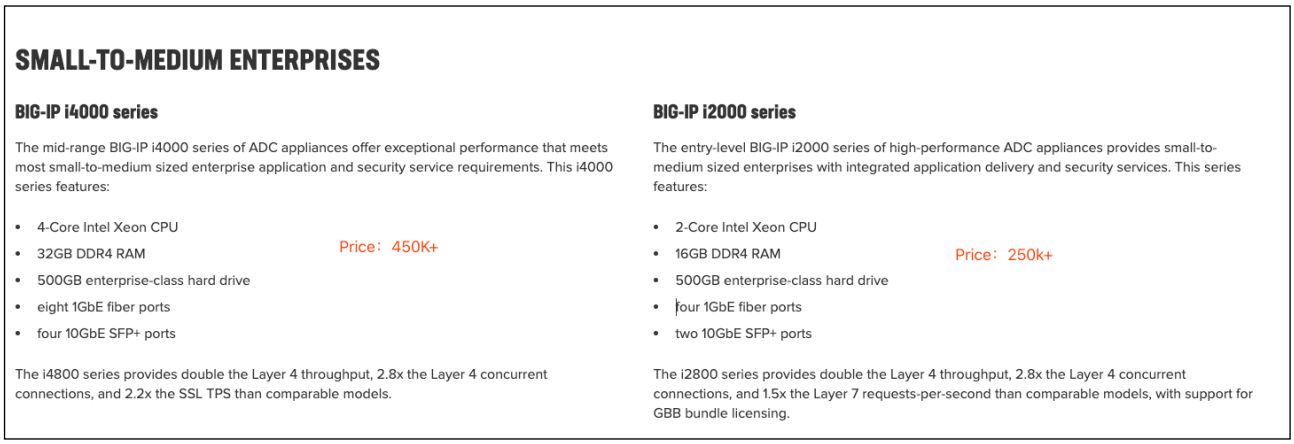
下一代负载均衡的思考与实践

当下，web 3.0大火，各种炒币、非同质化代币NFT，元宇宙等应用模式纷纷粉墨登场。这些场景有一个共同的点就是——上”链”（这里所说都是指公链。我个人认为联盟链都是假的，巨头的跑马圈地，最多就是web 2.0重来一遍。只有公链才是真的，所有场景都值得重塑一遍）。上”链”有一点难度的，一是因为应用场景可能受政治因素影响，另外一个是技术也是影响场景落地的一个关键要素。我认为技术关键点可能有三个，计算、存储和网络。这其中，又以网络因素影响最大。为什么呢？无限计算和存储的基础是通过网络连接构成一个”超级计算机”，使其拥有海量算力和存储。所以如何提高网络”力量”是根本，如何无”墙”连接？如何提速？如何合理流量调度？都是需要我们解决的问题。今天，咱们来一起聊下网络的小模块——流量调度。那就不得不提网络流量大闸——负载均衡。

一. 背景与现状

主流的负载均衡器有很多，大致分为两类，商业和开源。商业负载均衡代表产品有F5和A10，它们的典型特征是运行稳定，功能齐全，但价格偏高，性能有限。例如F5中端产品配置4core CPU-32G mem-1G net bandwidth的，应对如今的互联网的海量数据转发场景，根本无法应对。价格又偏高，低端BIG-IP I2000系列是25万+，中端BGP-IP i4000系列是45万+。性价比极低。



SMALL-TO-MEDIUM ENTERPRISES

BIG-IP i4000 series

The mid-range BIG-IP i4000 series of ADC appliances offer exceptional performance that meets most small-to-medium sized enterprise application and security service requirements. This i4000 series features:

- 4-Core Intel Xeon CPU
- 32GB DDR4 RAM
- 500GB enterprise-class hard drive
- eight 1GbE fiber ports
- four 10GbE SFP+ ports

Price：450K+

The i4800 series provides double the Layer 4 throughput, 2.8x the Layer 4 concurrent connections, and 2.2x the SSL TPS than comparable models.

BIG-IP i2000 series

The entry-level BIG-IP i2000 series of high-performance ADC appliances provides small-to-medium sized enterprises with integrated application delivery and security services. This series features:

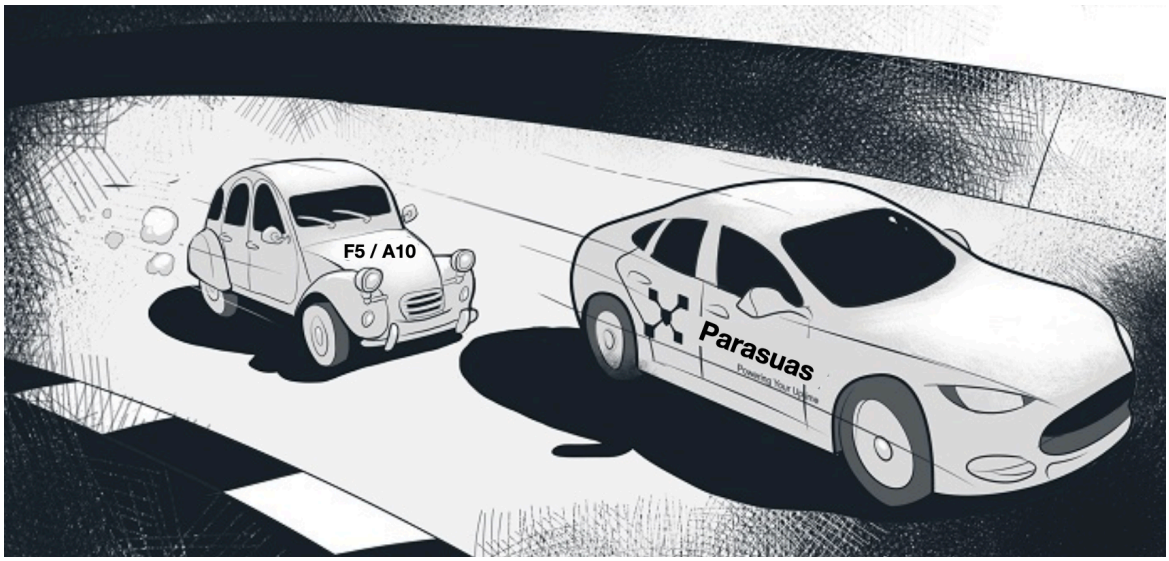
- 2-Core Intel Xeon CPU
- 16GB DDR4 RAM
- 500GB enterprise-class hard drive
- four 1GbE fiber ports
- two 10GbE SFP+ ports

Price：250k+

The i2800 series provides double the Layer 4 throughput, 2.8x the Layer 4 concurrent connections, and 1.5x the Layer 7 requests-per-second than comparable models, with support for GBB bundle licensing.

开源负载均衡又可以分为两类，一类是成熟的开源负载均衡，例如nginx、haproxy等。典型特征是功能基本满足，灵活，但缺少vip管理，热加载功能会造成业务抖动，配置数据基于文件存储，管理困难，对于企业级应用，不成熟。另一类是新兴开源负载均衡，例如iqiyi dpvs、openELB和metaLB等，特点是要么架构复杂，依赖特别多，dpvs是典型代表。或者，社区不活跃，维护比较困难，有问题基本靠自己解决，难上加难。

综述，我们是否能有一个负载均衡器？它是一种全新设计模式的，分布式的，性能强的，云原生的，它是为未来海量流量调度而生的。它能充分解决上述问题，设计简单，成本低又易用？——是的，是有的，那就是负载均衡器-合页（Parasaus）。



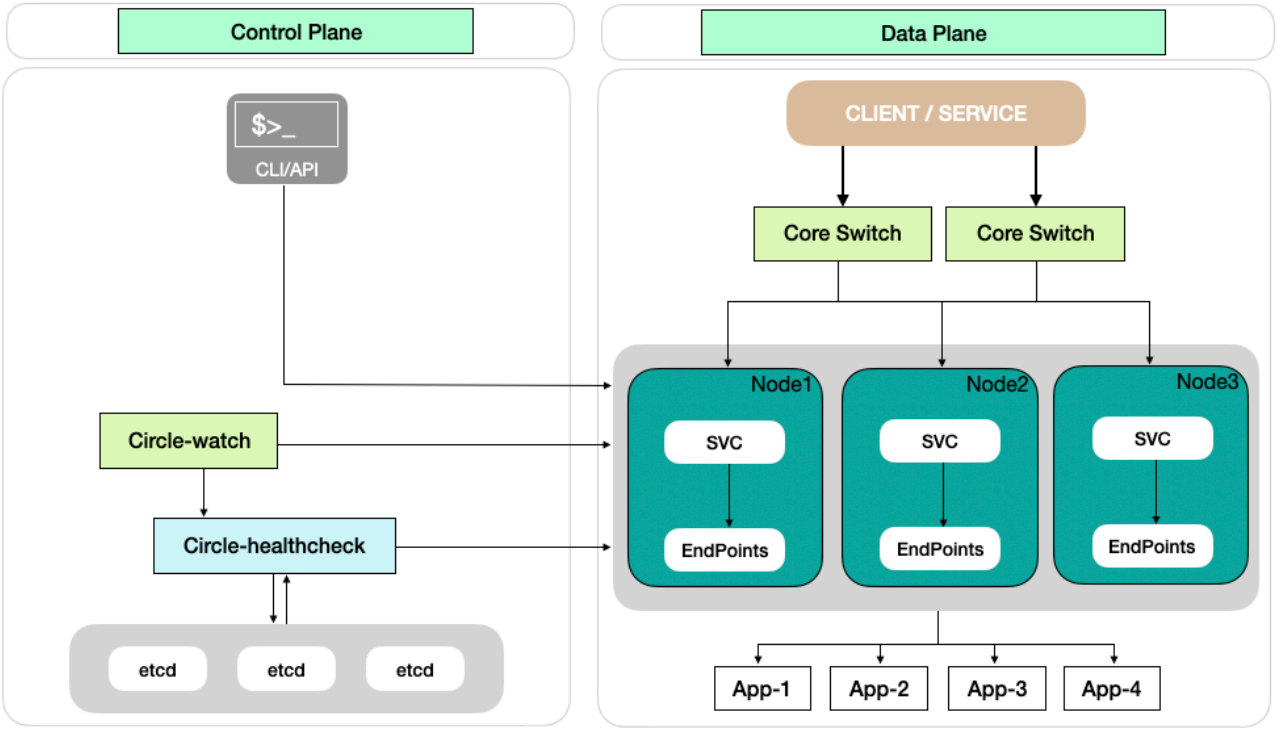
二. 原理：设计架构与实现特性

合页（Parasaus）按照数据面和控制面分离的架构而设计，数据面只负责数据转发，控制面负责watch条目的变化、管理员的增删改查及节点探活，架构简单清晰。

1. 数据面的流量转发节点无状态，支持横向扩展，最多支持254个节点。
2. 流量转发基于lvs实现，流量转发在内核态完成，性能优异，非常高效。
3. 整个负载均衡维护一个ip地址池和端口池，容量有35000个地址。可以为每个服务都创建一个连接地址。
4. 控制面比较轻量，功能简洁，circle-watch模块观察到条目变化，触发一个hook，在circle-healthcheck添加探活条目，用以及时监测到服务是否可用。circle-healthcheck探测到有不可用服务条目或服务恢复条目时，触发一个hook，在数据面对转发流量进行调整，使流量能全部转发到所有服务健康节点上。
5. 服务健康检查方式支持三种，分别是L7 http、L4 tcp、mysql。
6. 支持容器部署，产生的元数据存储到etcd中，易于管理，并保证高可用。
7. 数据面的实现借鉴了kubernetes service的架构，支持多种负载均衡策略，例如RR、LC、WLC、WRR、source hash等。
8. 配置的增删改均支持热加载，即使是业务高峰期，对业务也是零影响。

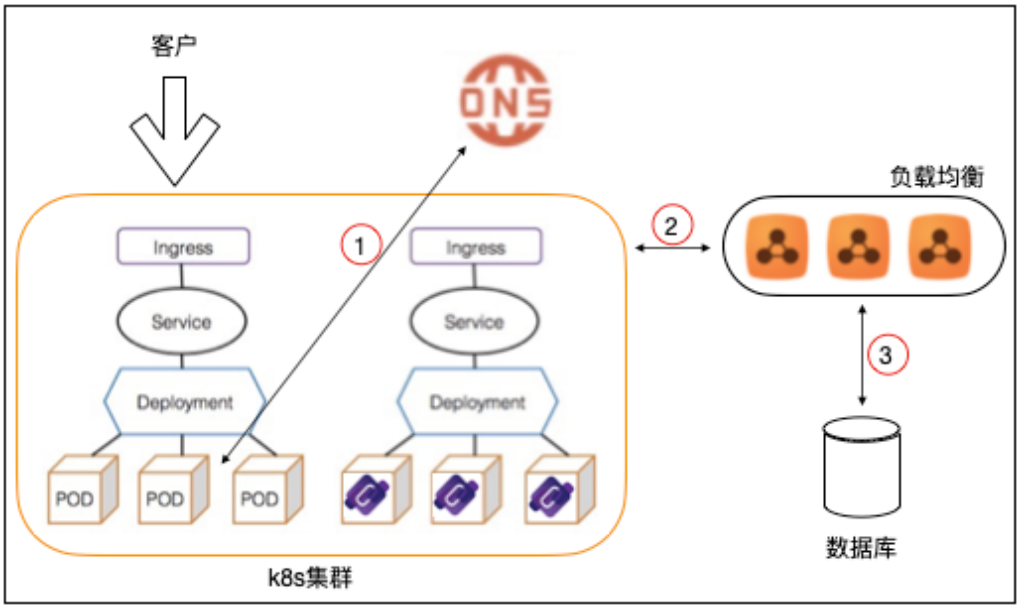
9. 硬件均采用普通x86服务器，无需任何特殊配置（例如dpvs的dpdk驱动）或硬件，成本极低。自主可控，支持信创。

10. 管理接口支持多种方式，例如接口、yaml文件和web界面，简单易操作。



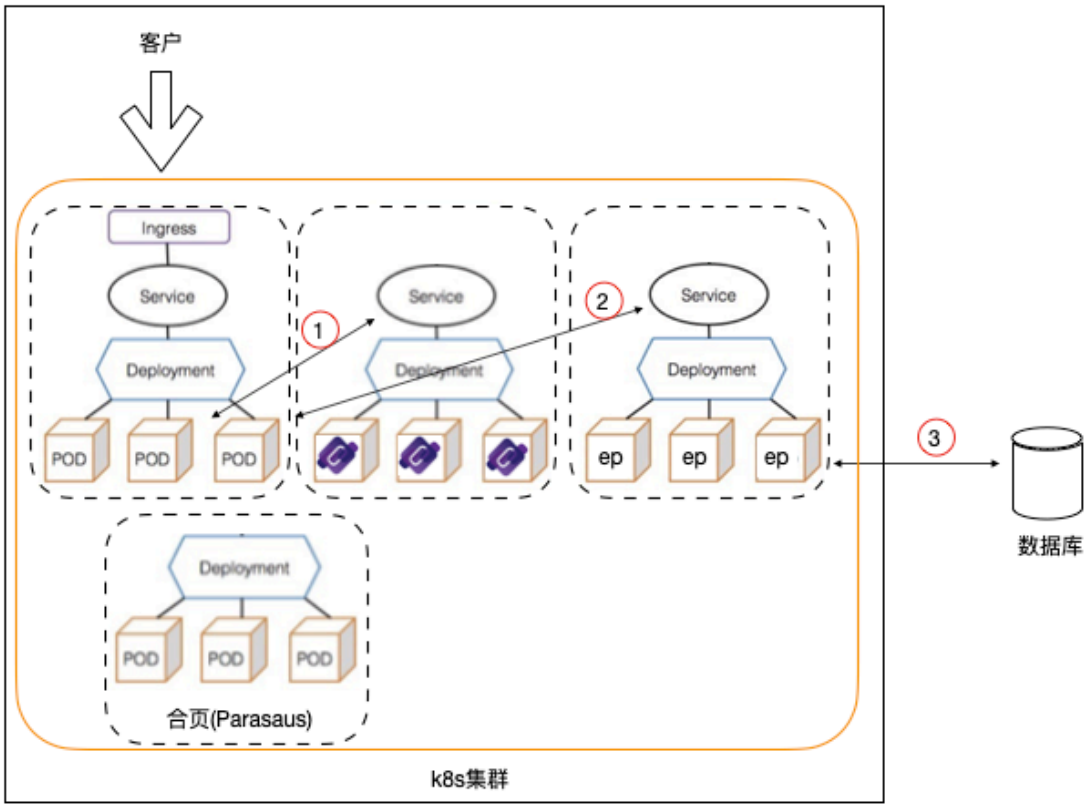
三. 应用场景：为云原生而生

场景1：完全云原生，是cloud-native，更是kubernetes-native的，可大幅缩短通讯链路，有效降低延迟。举例：在传统场景中，k8s中pod应用对外部服务的访问必须流出容器集群虚拟网络而进入数据中心物理网络，每次进出其他子网和经过转发节点（例如dns、负载均衡等），都会耗费时间，且增加故障域，如下：

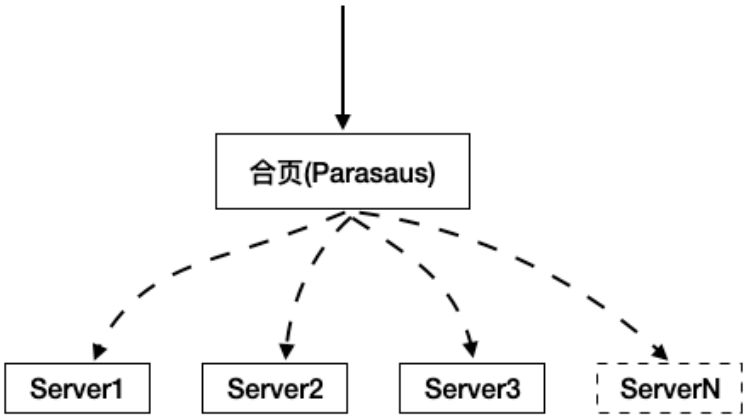


但

在新场景中，借助合页（parasaus）的强大功能，可实现流量转发操作全部在一个区域内完成。针对以上案例，负载均衡和探活功能由合页完成，且是在容器集群内全部完成，大大缩短链路和减少故障域。

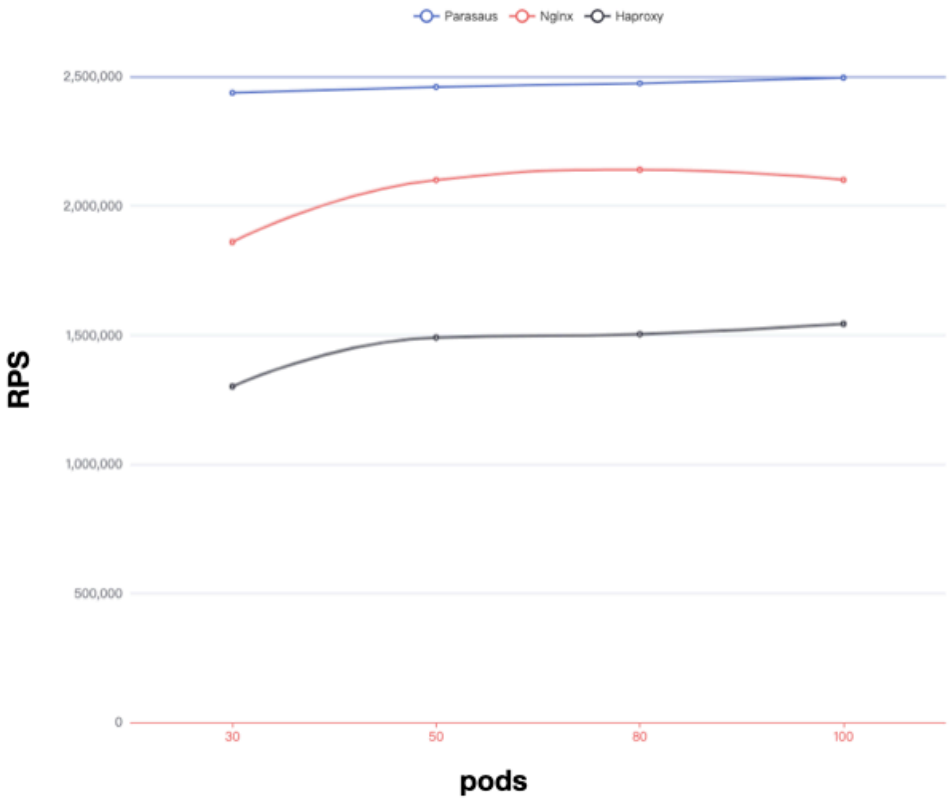


场景2：在数据中心内进行流量调度，用作于基础设施底层的一环，将流量在南-北向和东-西向进行合理分配。



四. 性能表现

我们针对当前比较主流的负载均衡开源产品进行了比较和测试，相较于nginx、haproxy，parasaus性能表现很好。

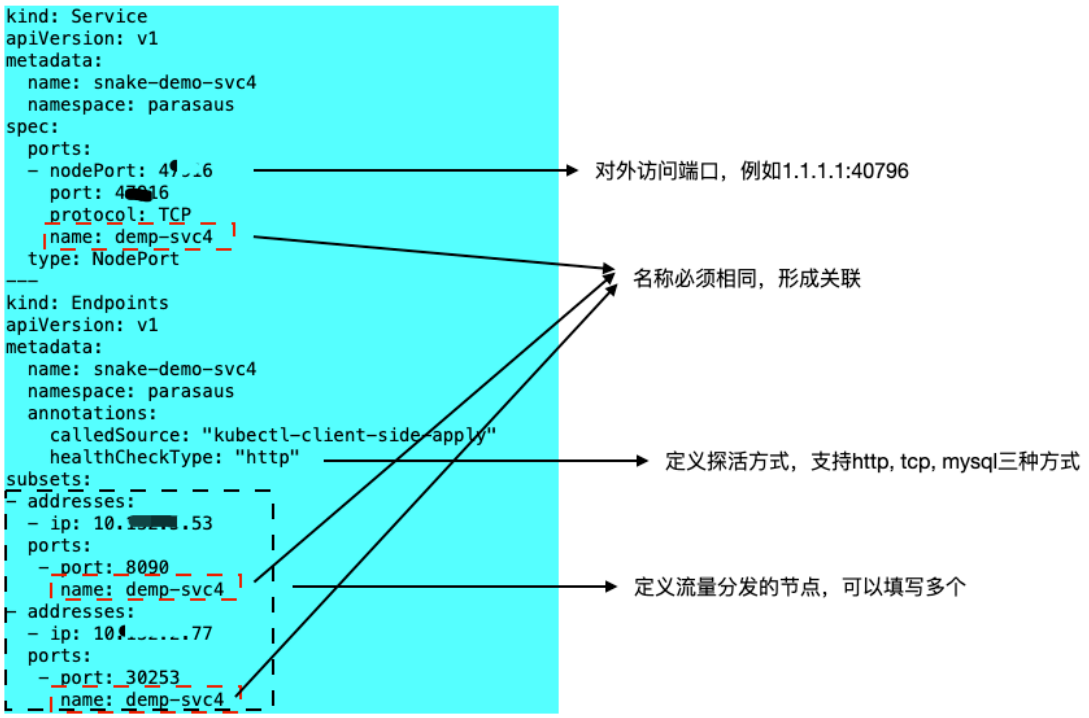


五. 使用方式

目前合页(Parasaus)负载均衡支持三种方式，分别是web页面、yaml文件和API接口方式，使用方便，对开发人员和运维人员都很友好。举例：

web页面方式：

Yaml文件方式：



六. 未来发展

合页(Parasaus)已经在一些公司试用，发挥价值。但为了让其更快成长和服务更多人，我们选择将其开源。开源是软件项目最大的混沌工程，也是最有效的反脆弱措施。基于此，我们将合页进行开源，并将其捐献给NextArch基金会，希望能够借助基金会和社区的力量发展壮大。同时，也希望有兴趣的同学联系我们（349317925@163.com），共同成长。