



Open Talk API 网关与高性能服务最佳实践
HANGZHO 2019

阿里七层流量入口负载均衡算法演变之路

王发康

主办方：



CONTENTS 目录

- ① 统一接入架构介绍
- ② 负载均衡改造背景
- ③ VNSWRR演进过程
- ④ 效果及总结



高性能

C1000K不在是问题

生态丰富

HTTP、Mail、TCP/UDP

K8s、Docker、Mesh

Lua、JS

模块化

可扩展、安全

Tengine介绍



兼容并包

100%继承于NGINX

高级特性

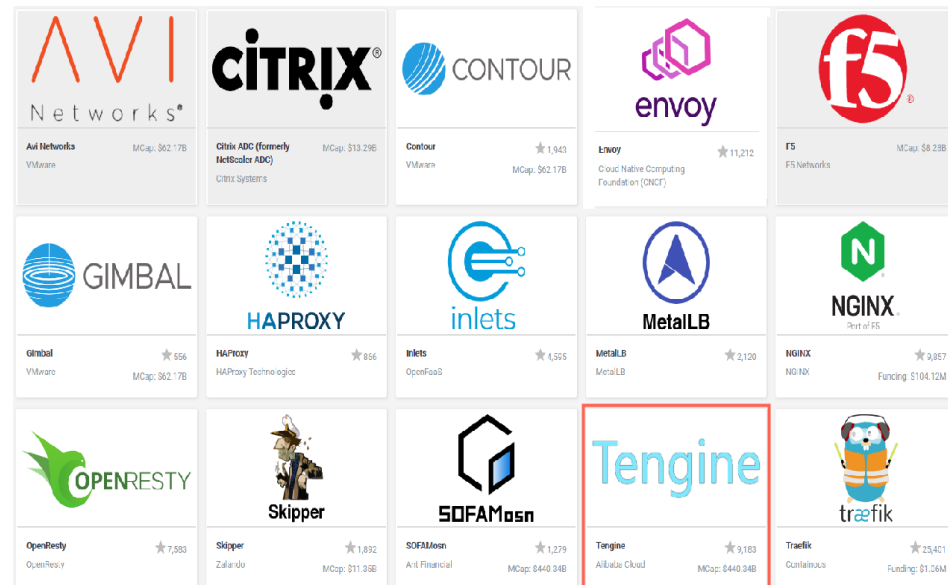
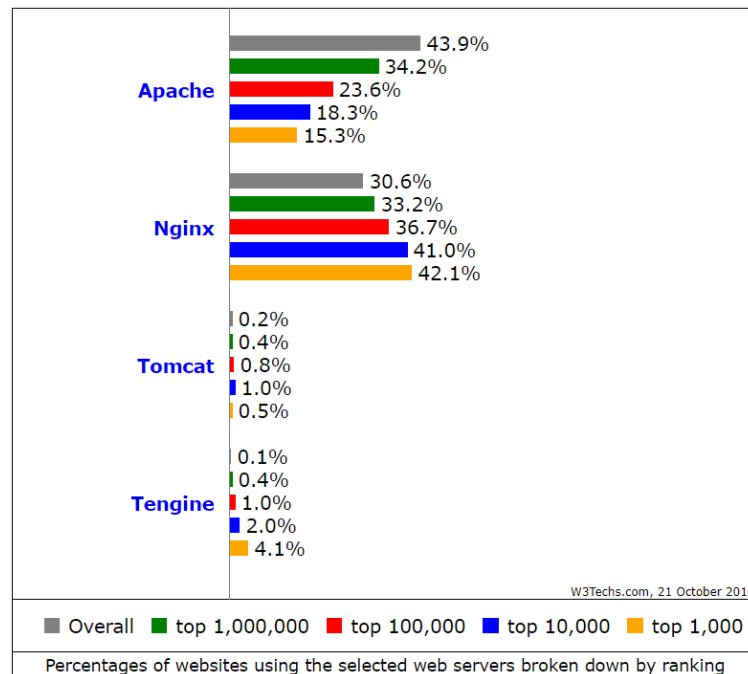
可监控、可维护、高可用、

赋能业务优化能力、软硬件

结合

规模化

双11全链路验证

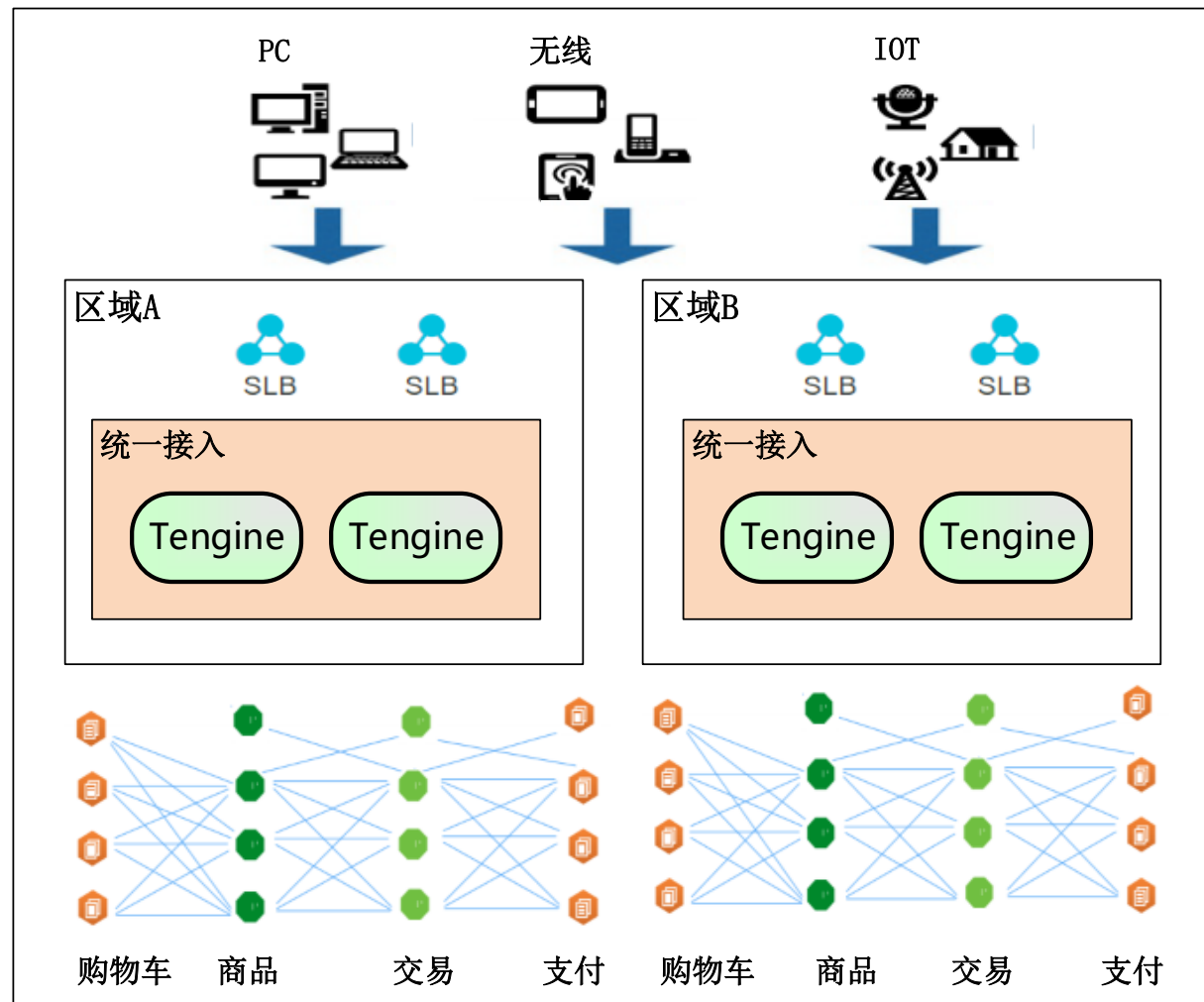


统一接入层架构介绍

Tengine作为阿里集团七层流量入口核心系统，支撑着阿里巴巴历年双11等大促活动平稳度过；

提供智能的流量转发策略、HTTPS加速卸载、动态服务发现、系统限流保护、动态开关、链路追踪等众多高级特性；

秉着软硬件结合的性能优化思想，Tengine在功能、性能、高可用等方面均获得充分验证；



NGINX SWRR算法描述

假设有 3 台机器 a、b、c 权重分别为 5、1、1，CW(current weight)、EW(effective Weight)、TW (total weight)。

请求编号	选前权重值	被选中server	选后权重值
1	{5, 1, 1}	a	{-2, 1, 1}
2	{3, 2, 2}	a	{-4, 2, 2}
3	{1, 3, 3}	b	{1, -4, 3}
4	{6, -3, 4}	a	{-1, -3, 4}
5	{4, -2, 5}	c	{4, -2, -2}
6	{9, -1, -1}	a	{2, -1, -1}
7	{7, 0, 0}	a	{0, 0, 0}

其选中的顺序为：{a, a, b, a, c, a, a}

普通WRR选择的顺序为：{c, b, a, a, a, a, a}

特点：平滑、分散

```
for(i = random() % n; i != i || flag; i = (i + 1) % n) {  
    s[i].cw += s[i].ew;  
    tw += s[i].ew;  
    if(best == NULL || s[i].cw > best->cw) {  
        best = &s[i];  
    }  
    flag = 0;  
}
```

```
best->cw -= tw;  
return best;
```

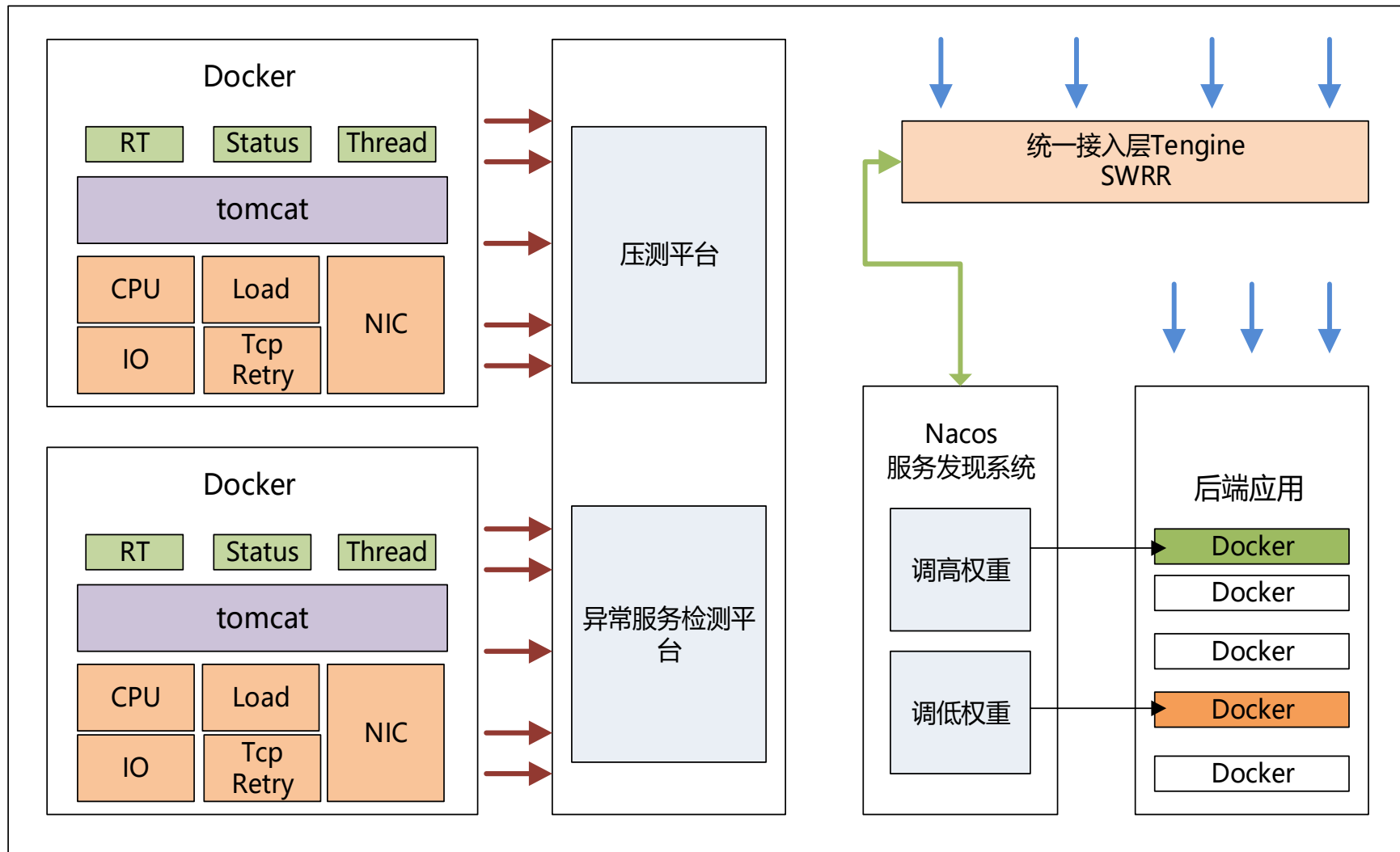
流量调度

压测平台

通过调高机器权重，从而达到线上机器真实引流目的。

异常检测平台

通过实时监测各应用机器及服务状态，按照一定的算法降低异常机器的权重，从而规避一些异常问题。



接入层VNSWRR算法改造背景

压测平台和流量调度平台会动态调整接入层后端机器的转发权重，原生NGINX的负载均衡算法会导致调高权重的机器瞬间请求量暴涨，导致压测平台和流量调度平台不可用。



当机器的权重从1调整为2，被调高权重机器的流量瞬间冲高50倍左右；

为何瞬间冲高？ 为何流量冲高的不是预期的2倍？

问题分析

如下是调高权重后对应机器的QPS变化特征：

- 1、QPS持续上涨峰值是1300左右；
- 2、QPS持续上升7s左右后开始下降，随后稳定在平均值的2倍左右；
- 3、权重被调高机器在A机房，接入层Tengine是优先同机房转发、其CPU是32核；
- 4、当时该应用A机房的QPS是K左右，接入层A机房N台；

$$(N * 32 / K) * 1 / 2 \text{ 约等于 } 7.68s$$

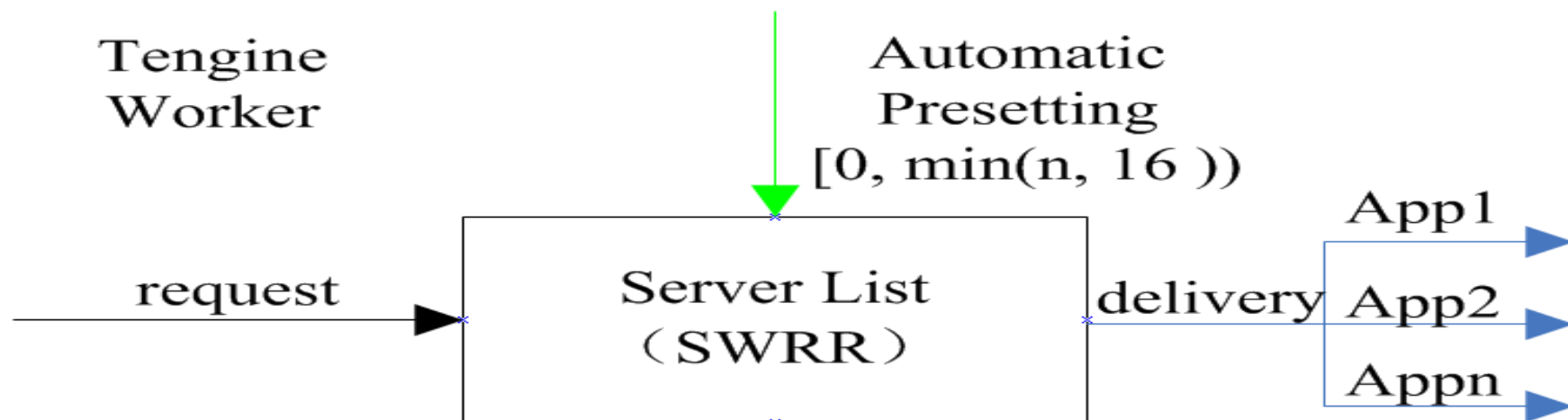


当机器的权重从1调整为2，被调高权重机器的流量瞬间冲高50倍左右；

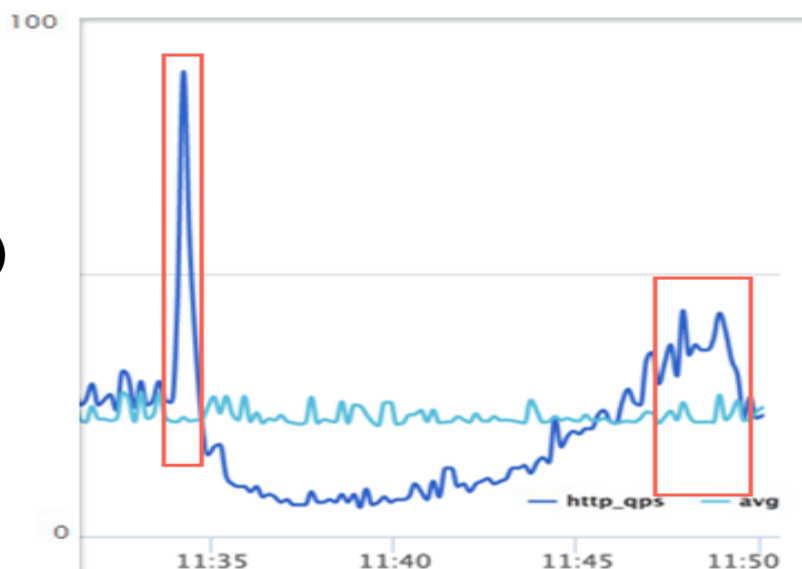
为何瞬间冲高？ 为何流量冲高的不是预期的2倍？

接入层VNSWRR算法 (V1)

算法模型 (V1)



对应效果 (V1)

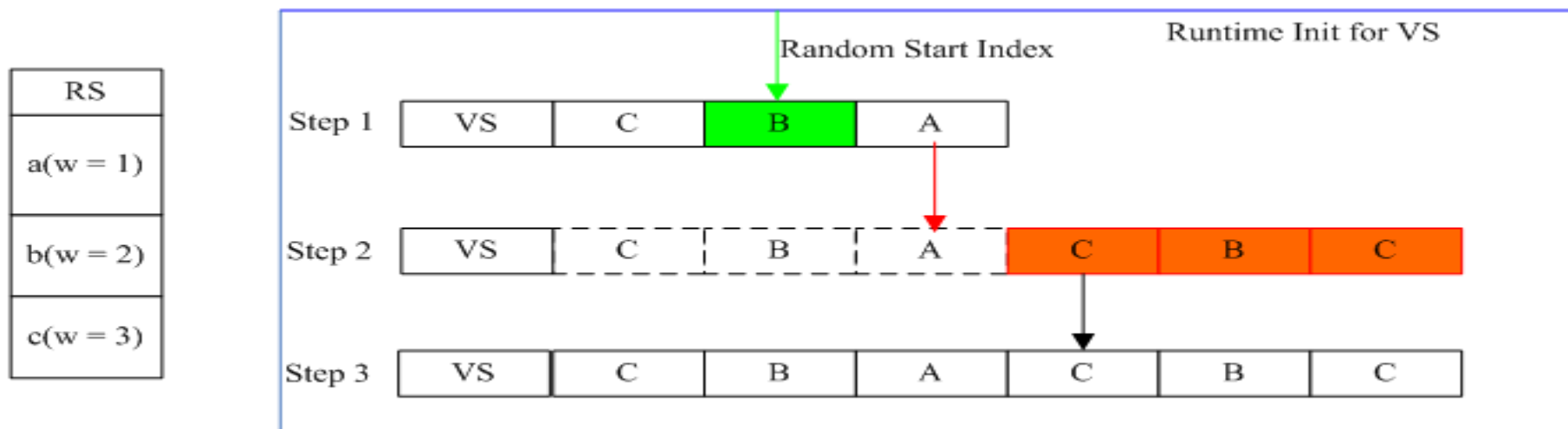


当机器权重从**1**调整为**2**，流量瞬间冲高**3**倍左右，随后**15**分钟左右达到**2**倍左右

为何流量还瞬间标高3倍？
为何流量稳定达到2倍需要长达15分钟左右？

接入层VNSWRR算法 (V2)

算法模型 (V2)



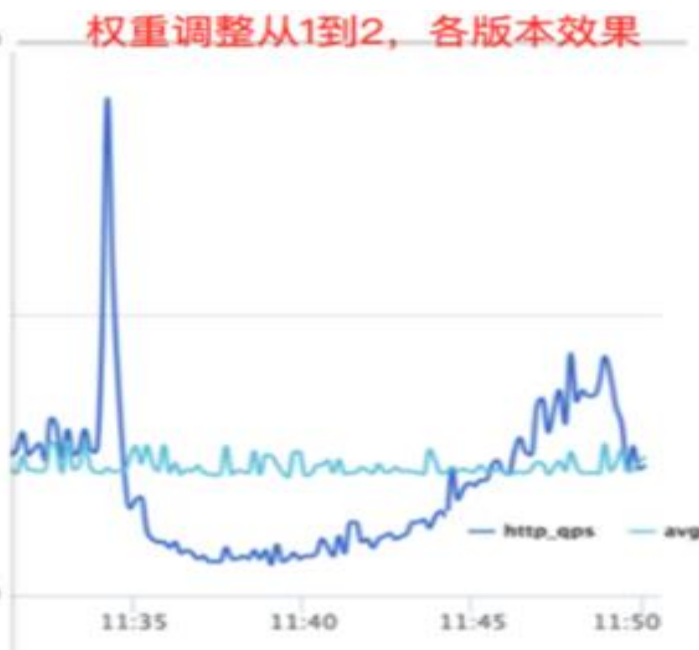
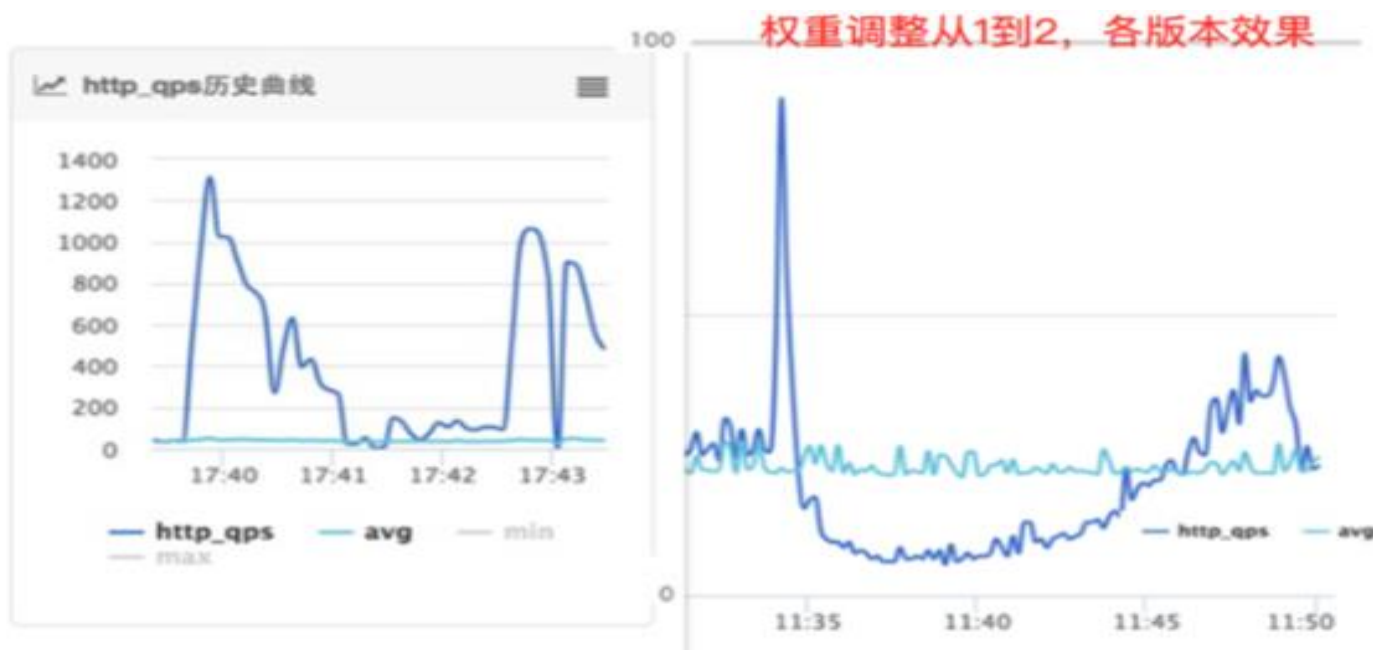
对应效果(V2)



当机器权重从1调整为2，被调高权重机器的流量随后平缓上涨，分钟级内达到2倍左右

接入层VNSWRR算法演进效果对比

不仅解决被调高权重机器的流量异常飙高问题，且新算法时间复杂度从 $O(n)$ 优化为 $O(1)$ 、整体性能提升**10%**左右。



思考&总结

解决问题优先性能优化

开始解决这个问题的时候就一头扎入性能优化的深坑里面（a.怎么去初始化这个虚拟节点？ b.怎么高效（ $O(n)$ ）的将虚拟节点打散等等），其实在解决问题的过程中就逐步演进到最优状态；

任何一个特殊现象，背后必定有原因

就譬如当我们看到权重调高后机器的QPS变化趋势图一样，每一个特殊峰值点以及变化趋势将会给我们挖掘这个问题的本质带来很大的帮助；

简单、高效、可靠

简单并不代表low，代码入侵越小、出问题就越少、同时也容易发现；

小程序大流量场景，兼容并包

大到一个设计方案、小到每一行代码都应该具有兼容并包的思想；

Tengine官网: <http://tengine.taobao.org>

Tengine Github: <https://github.com/alibaba/tengine>

Tengine Gzip 硬件加速: https://mp.weixin.qq.com/s/v_BsfSFp-LwiCOFKaMIS_g

Tengine Async OpenSSL 硬件加速: <https://mp.weixin.qq.com/s/5OZOURttVdP8ye2IBXwwzg>

Tengine VNSWRR算法介绍: <https://mp.weixin.qq.com/s/3KZ99d94yqRDxEByn7nGWa>

Tengine 开源交流群

Tengine

329人

Q?A:E



扫一扫群二维码，立刻加入该群。

THANKS!