

# 微博平台架构练级之 高可用

新浪微博 平台架构

姚四芳(@icycrystal4)



## Temporary Error (500)

We're sorry, but your Gmail account is temporarily unavailable. We apologize for the inconvenience.  
status of the service.

If the issue persists, please visit the [Gmail Help Center »](#)

[Try Again](#) [Sign Out](#)

[Show Detailed Technical Info](#)

- 容量规划
  - 超级热点 #周一见# #且行且珍惜#
  - 有限的资源
- 资源QoS
  - 光纤被挖断
  - 交换机故障&网络拥塞
  - 磁盘故障等
- 软件质量
  - 迭代速度小于一周
- 模块依赖
  - 一次请求数十个服务与模块

# 高可用影响微博IPO



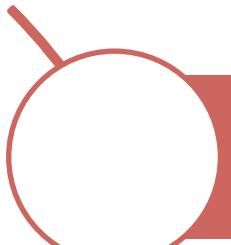
SLA:99.99%



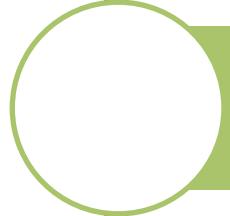
PV/UV



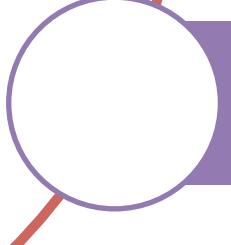
# 架构思路



容量规划与流量控制

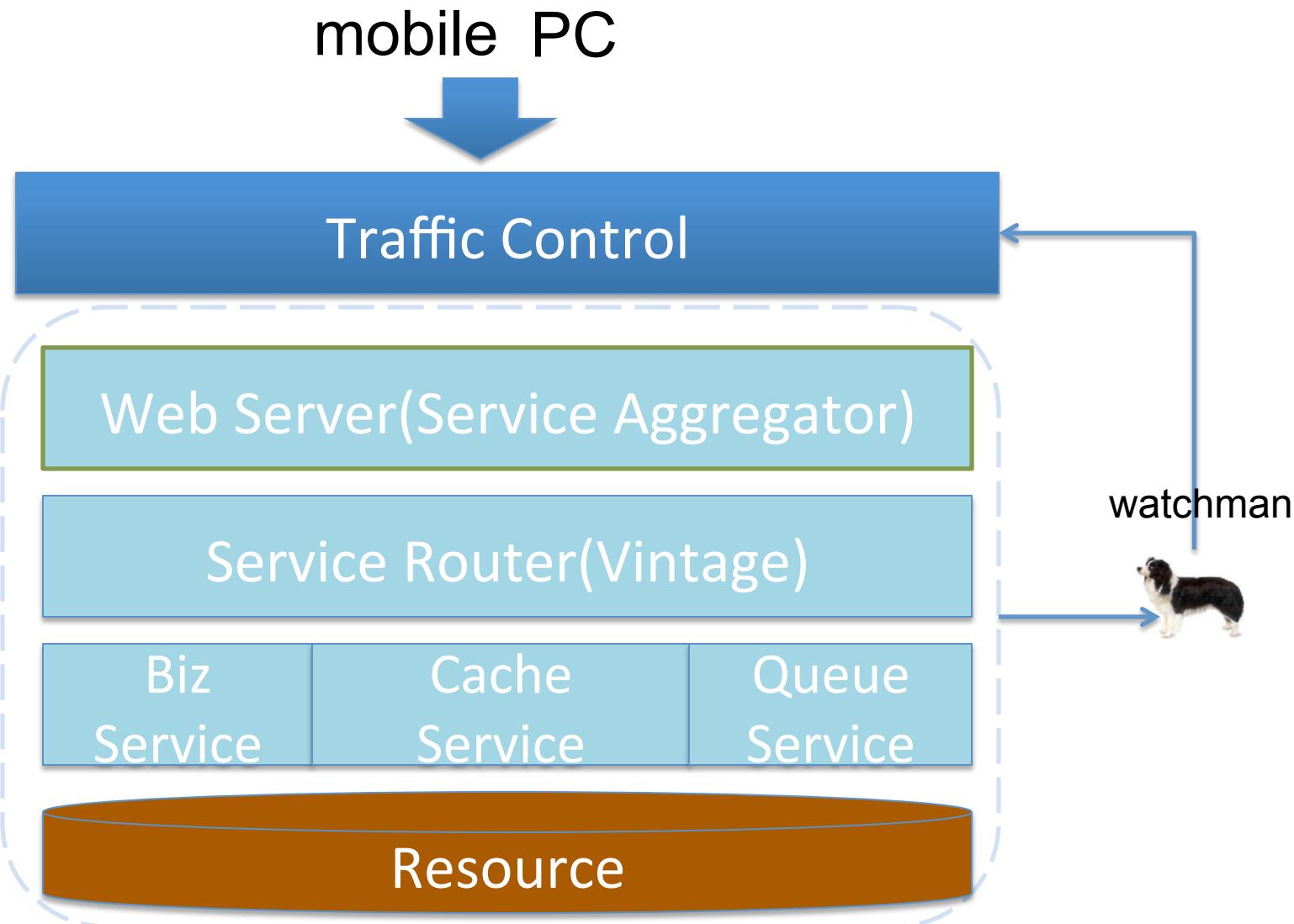


系统架构容错



问题快速定位与响应

# 微博平台架构示意图



# 容量规划与流量控制

容量预估

快速扩容

流量控制

## ➤ 目标

- 不容量与压力对可用性的影响
- 99.99%可用性下不同压力需要的容量

## ➤ 微博压力测试的特点

- 模拟用户访问行为、生产的内容成本高
- 峰值测试时客户端成为瓶颈-百万QPS
- 沙盒环境成本高
- 资源之间存在依赖关系，相互影响

# 压力测试



真实流量峰值

touchstone



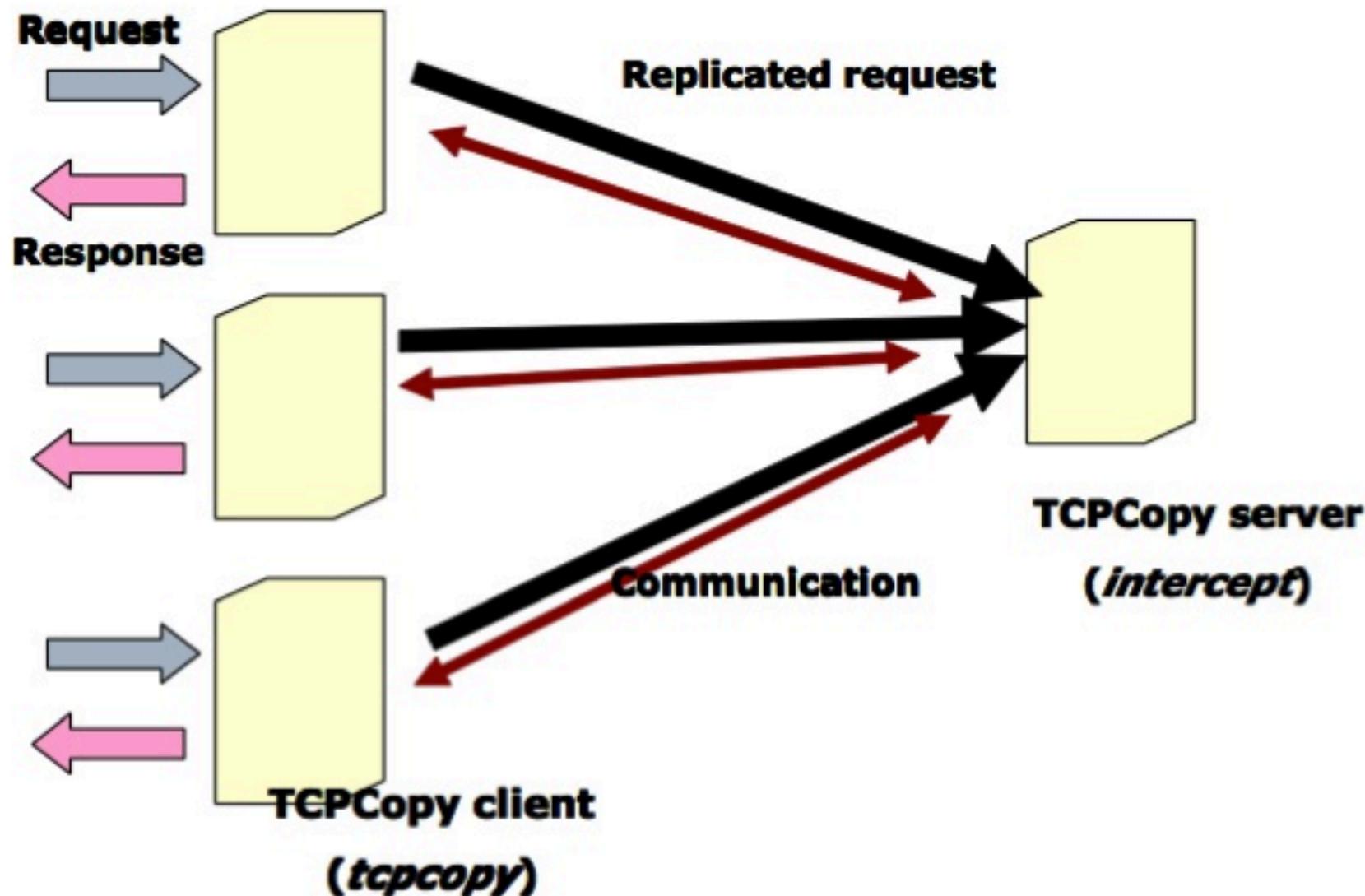
消除依赖资源影响

JMeter

9999下压力对容量的影响



# tcpcopy引流

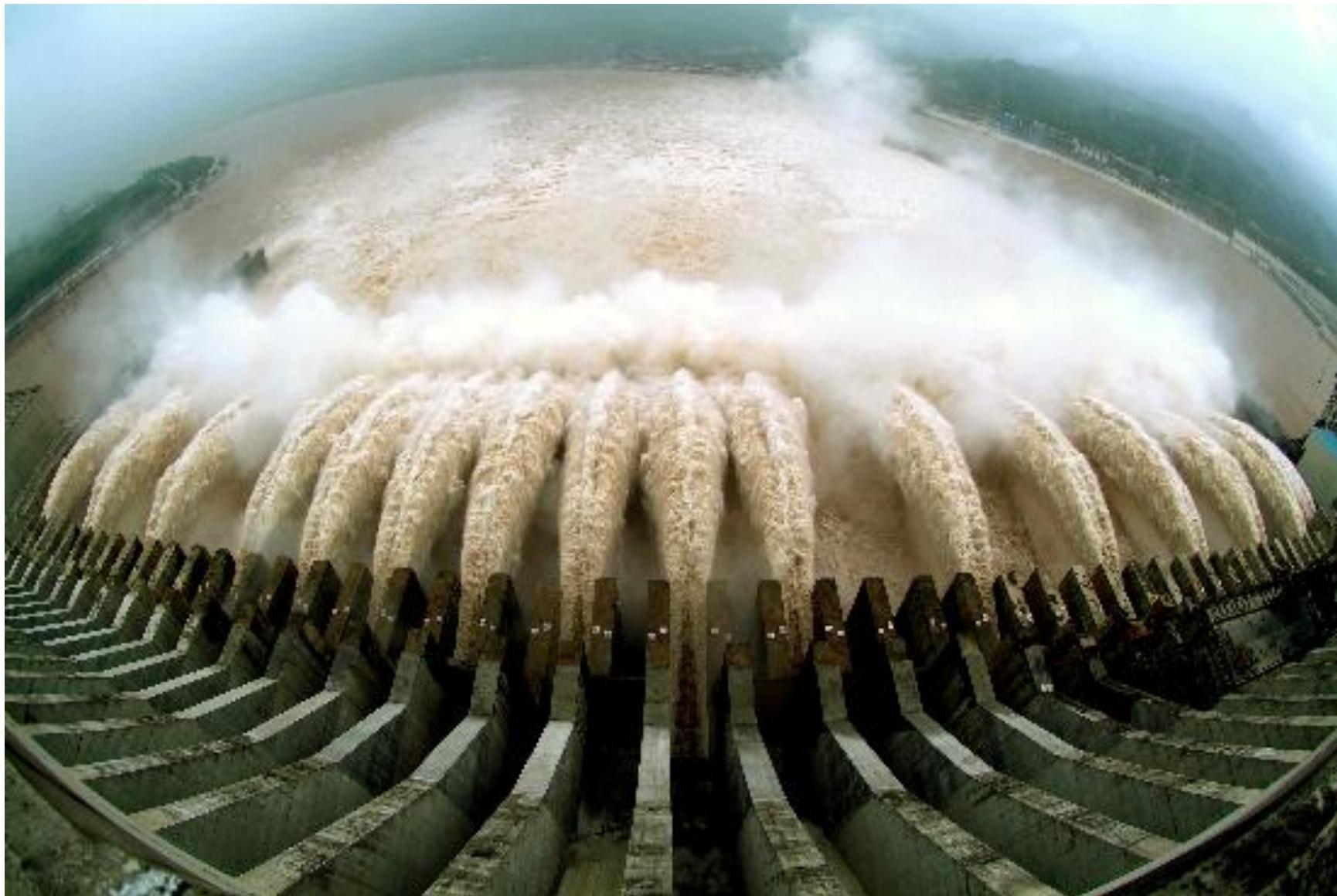


# 蓄水池:消息堆积

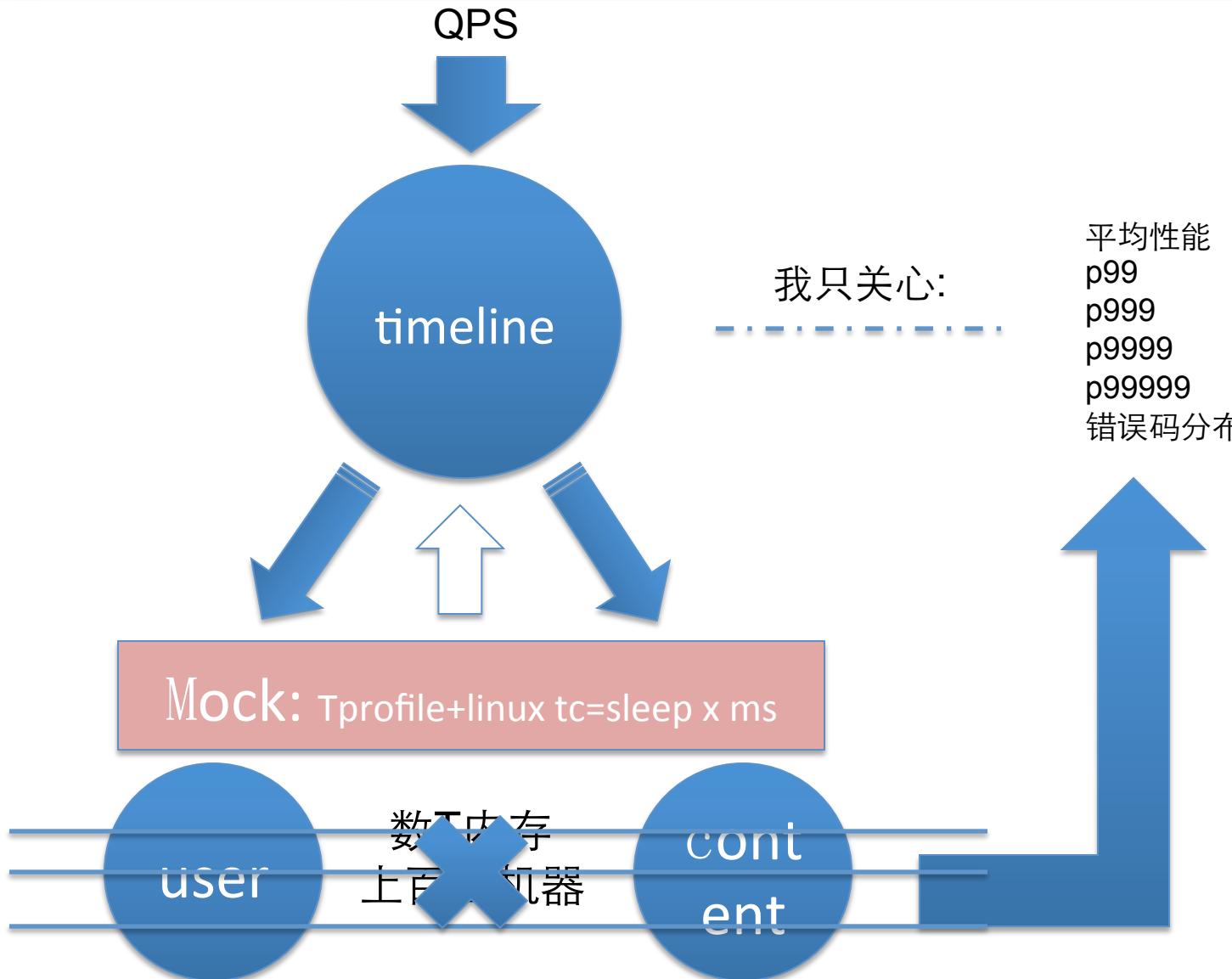


NO:20100320002650355159

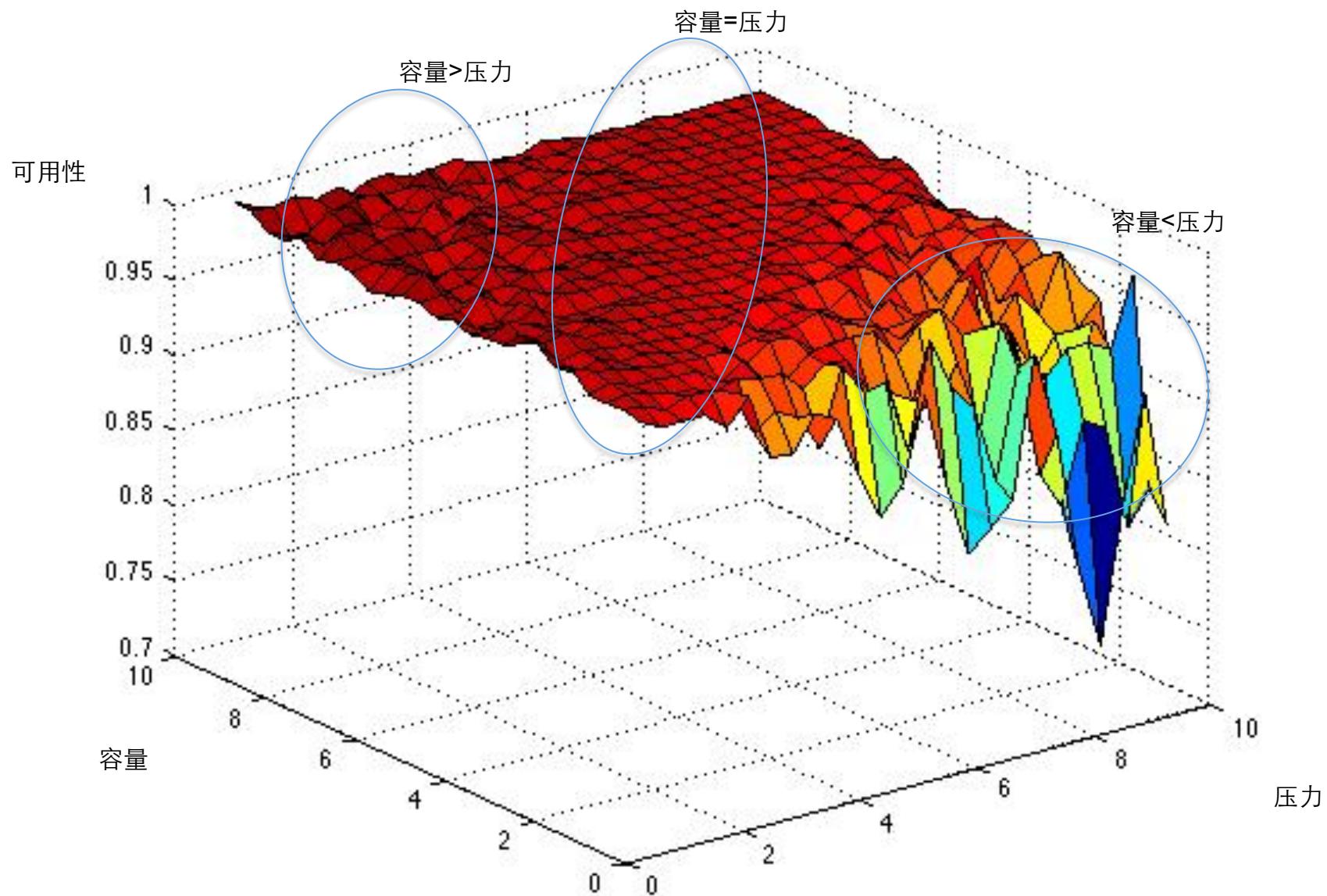
# 蓄水池:瞬间泄洪峰值



# Mock依赖资源

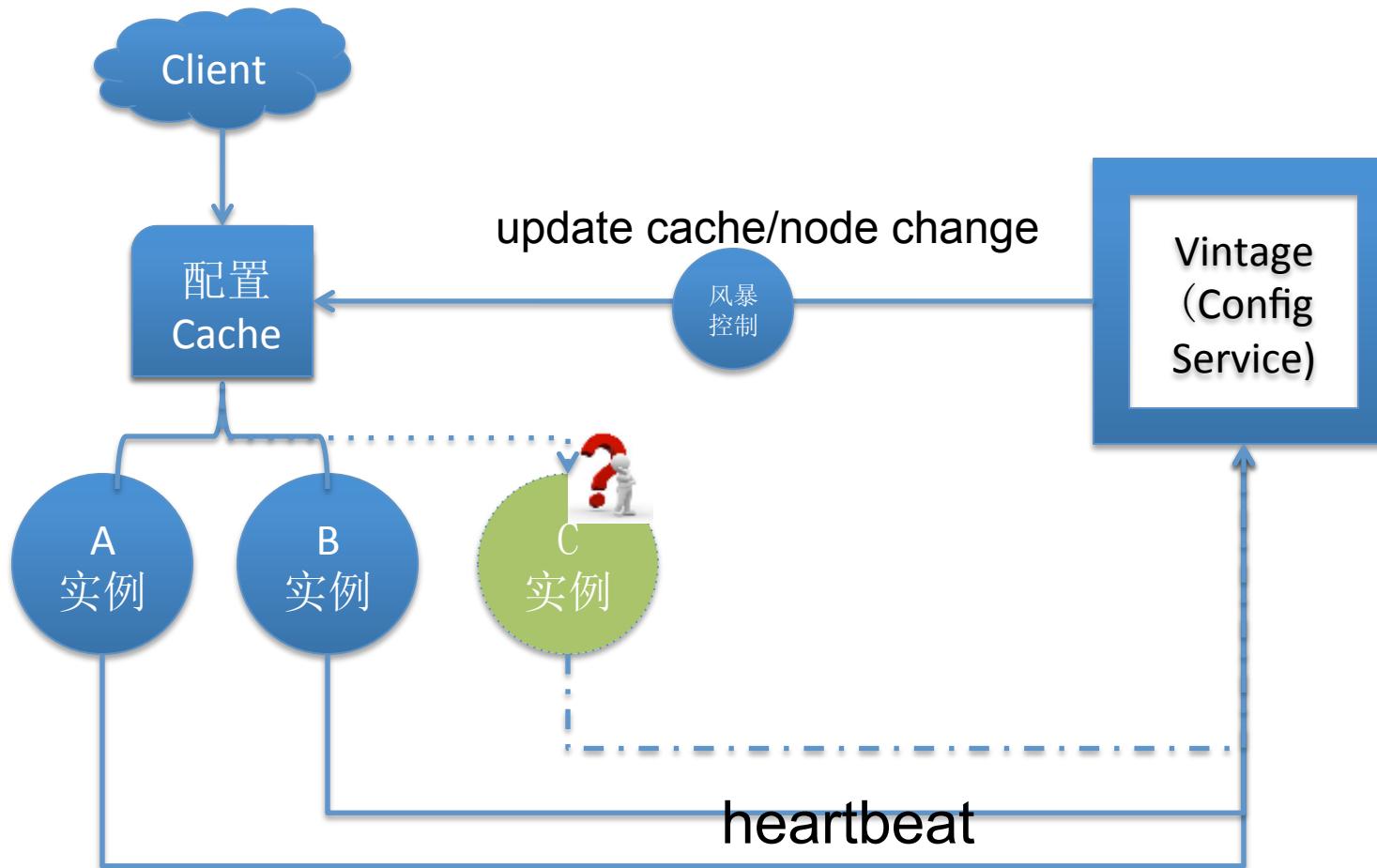


# 容量、压力与可用性三维关系



快速  
扩容

# Vintage与快速扩容



流量  
控制

# 五层流量控制保护



Req

区域热点、光缆挖断

跨机房流量迁移

DDOS

7层频次控制(用户、IP...)

分流

核心池

非核心池

降级

Service: switch=on|off

隔离

Tomcat线程池隔离调度

- DNS跨机房流量迁移
  - 地域热点：春晚、雅安
- 7层流量控制
  - DDOS攻击
  - 核心池非核心池分流
- 降级控制
  - switch=on|off
- 线程池隔离
  - 扩展TOMCAT线程调度机制

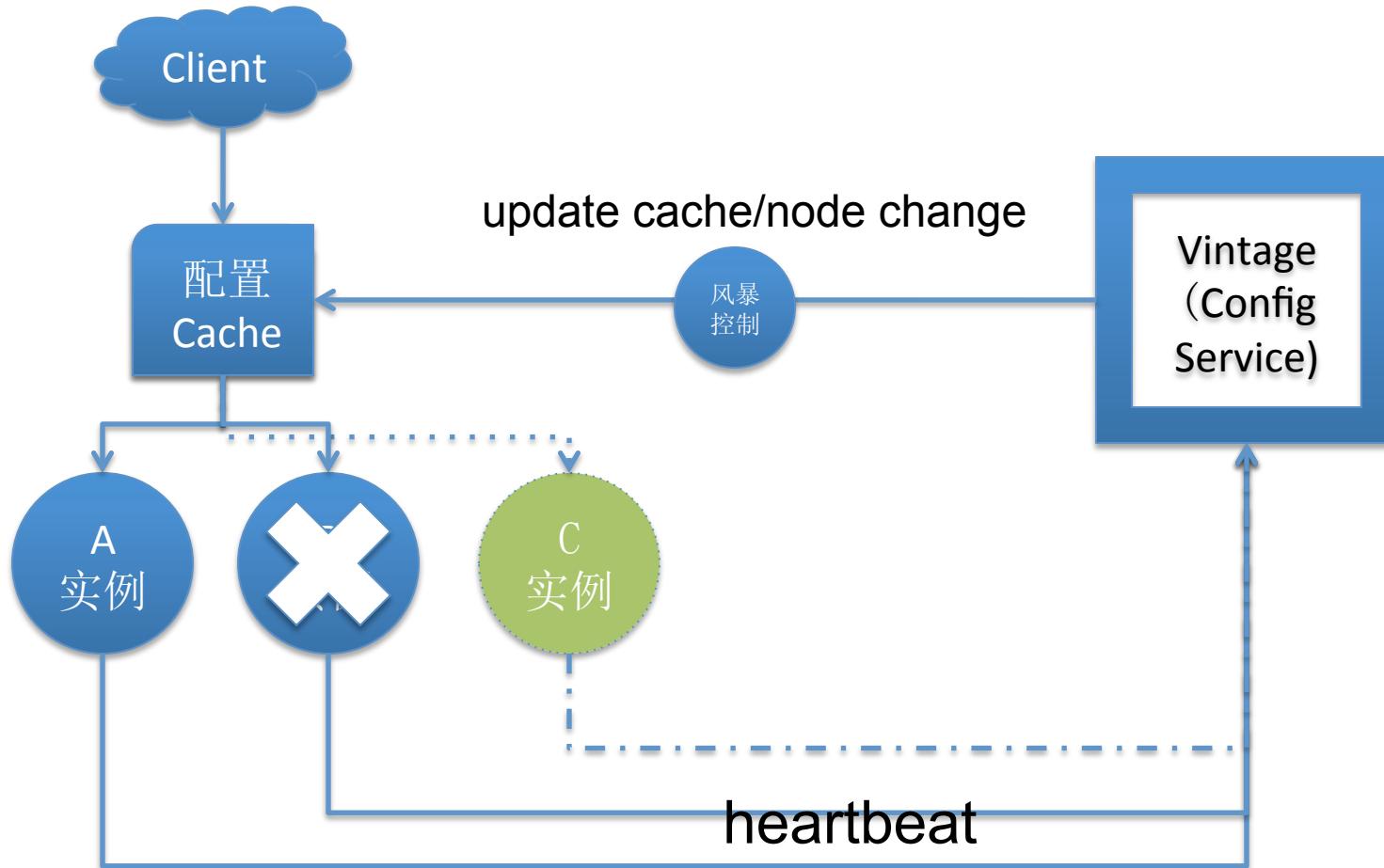
touchstone 压力测试&容量预估

vintage 快速扩容

所有资源服务均可进行流控

系统容错

# 计算资源容错



# 代码容错

## ➤ 快速回滚

- 新老版本同时存在
- 开关控制运行版本

## ➤ 灰度发布

- 按流量灰度
- 按用户灰度
- 按地域灰度

# 缓存资源容错

Cache现状

>1.5MQPS

99%命中率

超级热点

scale up/out

CacheService

twemproxy

宕机后miss

扩容复杂

影响命中率

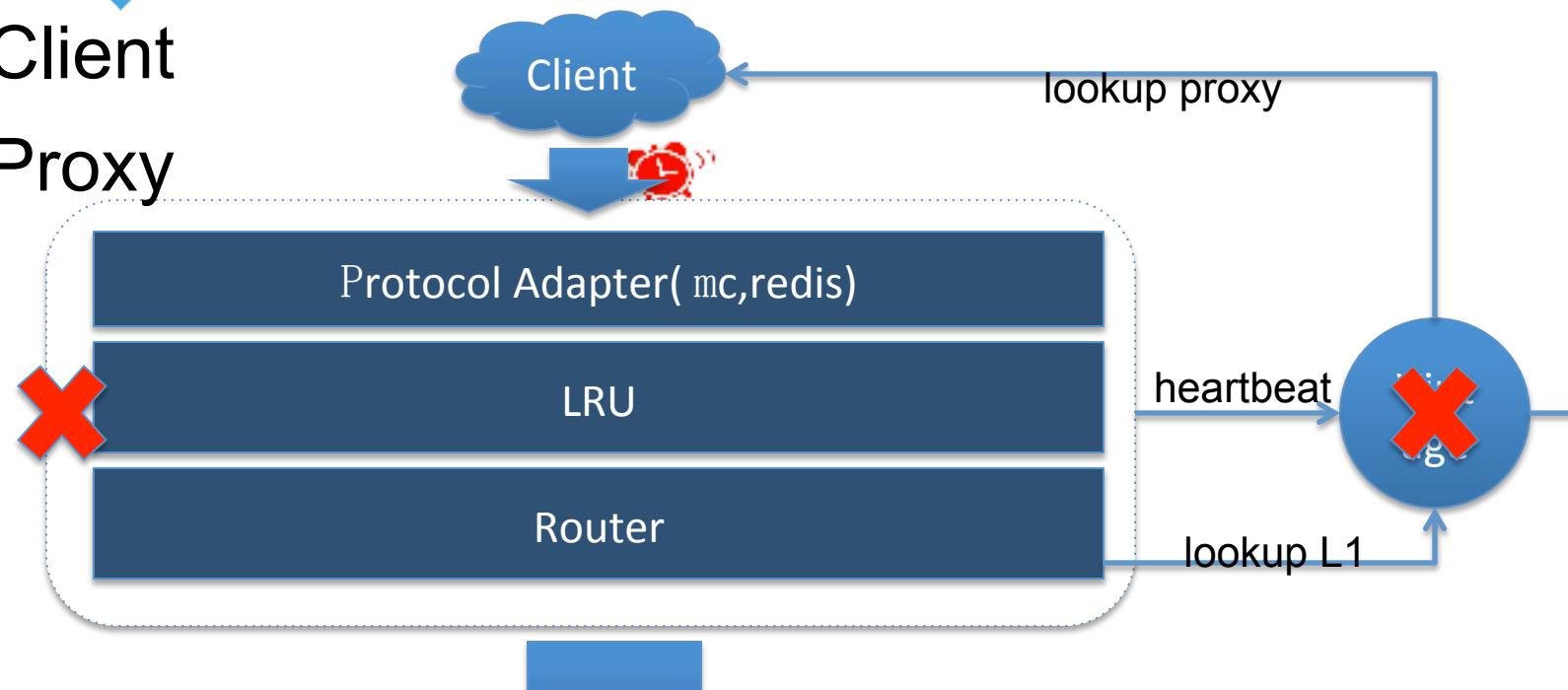
节点摘除

# 资源容错: CacheService

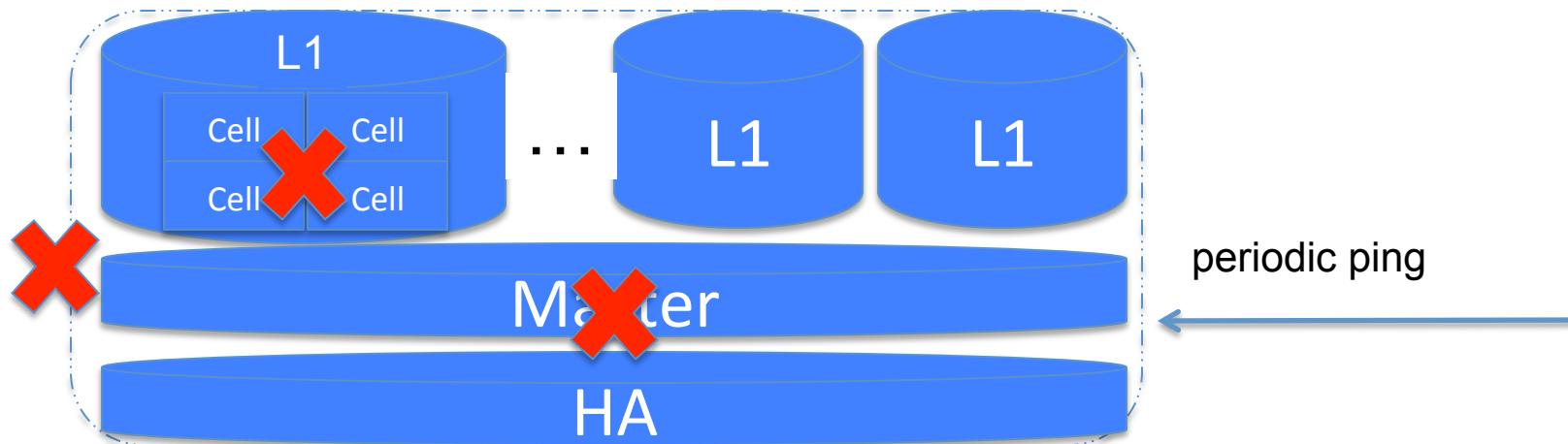


Client

Proxy



Cache



## ➤ Client

- 超时控制 & 重试

## ➤ Proxy层

- LRU (超级热点)
- Router: L1 cache路由

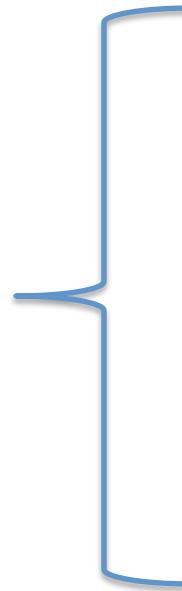
## ➤ Cache

- L1: 水平扩展解决访问峰值
- Master: L1穿透
- HA: Master的高可用

## ➤ Vintage

- Proxy&Cache节点的状态管理

服务化  
服务化



解耦合

整体高可用

9999

# 快速问题定位

# 痛苦的线上问题定位



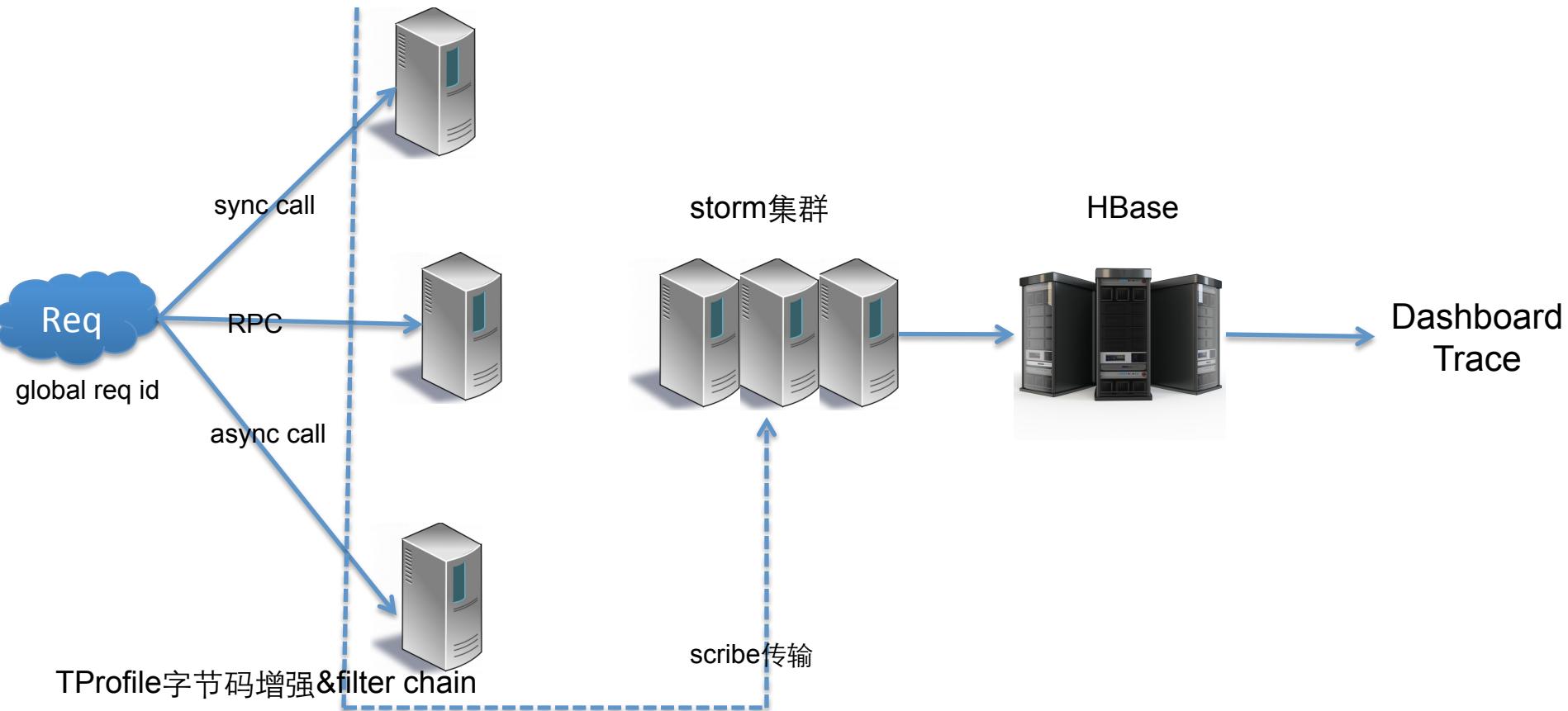
- 报警只是表象
- 海量的日志 (T级别)
  - grep&awk
- 日志离散在不同机器
- 没有完整的请求日志链

# 其实可以非常简单



- 4Ws
    - 谁(who)
    - 在什么时候(when)
    - 做了什么事情--API(what)
    - 调用了哪些方法(where)
  - Resource
    - 涉及的服务与资源(IP&Port)
  - 问题
    - 访问量
    - 性能
- 
- Why

# Watchman:分布式trace系统



数据采集

分析

存储

展现

- 全局唯一的req ID
- 无侵入拦截方法调用
  - 字节码增强
  - filter chain(RPC)
- trace
  - 用户维度、API维度、资源维度
  - 分布耗时、请求量
- dashboard 图形化
- 秒级数据延迟

# 示例

trace    Profile    monitor

Error:

Overview    Dependencies

Expand all

802 ms

about 2 hours ago



## service node summary

node	CPU	Wait
10.75.25.65:880	877.000	552.000

# 春晚大考

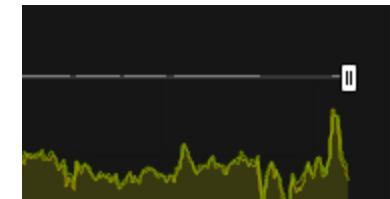
- 读写峰值压力大难以预估
  - 13年:10倍+写入&数倍读压力
  - 14年:CNTV深度合作、2次口播&二维码
- 峰值持续时间短
  - 数秒~数分钟
- 资源十分有限
  - 线性扩容成为不可能

10W/s的写入压力

- 全链路压测&Touchstone
  - 在线压测--洪水计划(蓄水池+瞬间泄洪)
  - 识别拐点与临界点
- 扩容核心服务与资源
- 应急预案
  - QPS超过X时进行流量迁移
  - 当DB的p999到达Xms时不写入DB
- 两轮次的在线压测演练
- dashboard & watchman

# 在春晚

- 喝着咖啡，看着春晚，盯着dashboard
- 流量迁移
  - 4次跨机房流量迁移，共计1/3
- 李明镐引发的降级
  - 13个服务降级
    - 最新一条微博
    - 短链
    - 提醒
- 零点
  - 第一分钟发表863408微博
  - 12倍+的峰值请求



# 总结

## ➤ 资源需要保护

- 容量规划
- 快速扩容
- 流量控制

## ➤ 资源与服务需要容错

- 禁止单点
- 异步&超时控制
- 降级：舍车保帅

## ➤ dashboard&watchman

- 图形化
- 自动化

# Q&A

欢迎加入新浪微博

简历私信至 @微博平台架构

邮件至 sifang@staff.sina.com.cn

以微博之力 让世界更美！

*weibo.com*