

背景：

微服务运行时稳定性的场景

激增流量



- 激增流量导致系统 CPU / Load 飙高，无法正常处理请求
- 激增流量打垮冷系统（数据库连接未创建，缓存未预热）
- 消息投递速度过快，导致消息处理积压

不稳定服务依赖

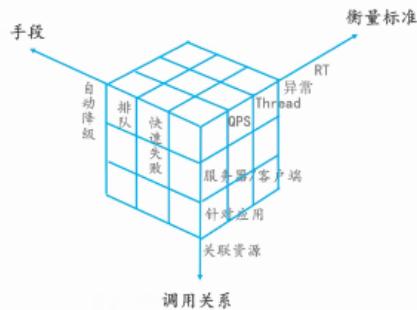


- 慢 SQL 查询卡爆连接池
- 第三方服务不响应，卡满线程池
- 业务调用持续出现异常，产生大量的副作用

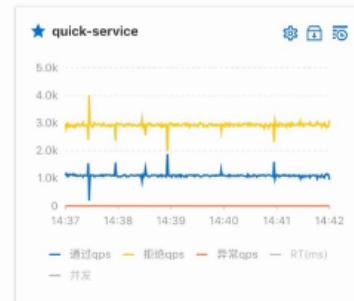
Reliability && Resilience



Sentinel 是阿里巴巴开源的、面向分布式服务架构的流量控制组件，主要以流量为切入点，从流控、流量整形、熔断降级、系统自适应保护、热点防护等多个维度来帮助开发者保障微服务的稳定性，历经阿里双十一10年大促场景沉淀，全面覆盖微服务、网关、Service Mesh 多种运行时稳定性场景。



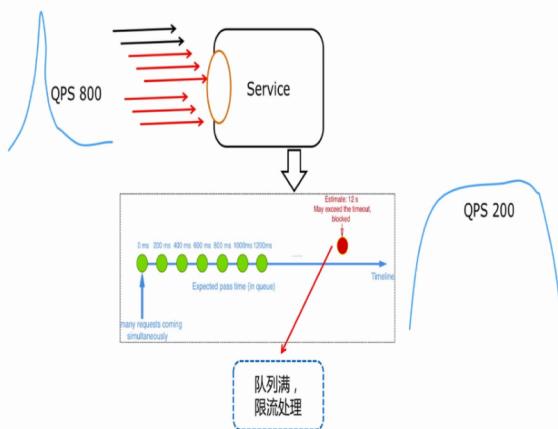
多维度的高可用流量防护能力
自适应保护系统



秒级实时监控与动态规则管理

场景：

流量控制



最普适的场景：

- RPC provider 端控制脉冲流量
- 针对不同调用来源进行流控
- Web 接口流控

如何配置规则：

1. 梳理核心接口
2. 通过**事前压测**评估核心接口的容量，配置 QPS 阈值
3. 注意流控处理逻辑

流量不大，万事无忧？

ms级流量的统计,基于统计进行流量控制

流量低不代表不需要做流控,每个机器都有一个可以承受的流量上限.只要系统有超过阈值的风险,就有必要进行配置

熔断降级与隔离



熔断降级与隔离

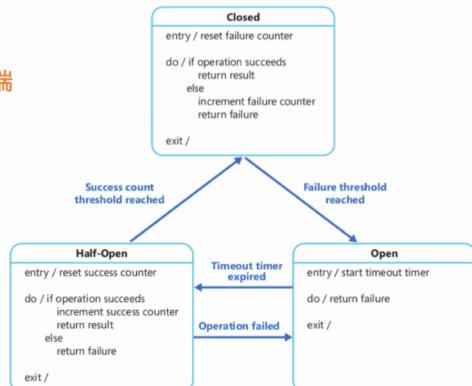
保护自身的手段

- 并发控制（信号量隔离）
- 基于慢调用比例熔断
- 基于异常比例熔断

通常在 consumer 端
组合配置

触发熔断后的处理逻辑示例

- 提供 fallback 实现（服务降级）
- 返回错误 result
- 读缓存（DB 访问降级）



熔断降级的前提：梳理强弱依赖，只有弱依赖才能降级

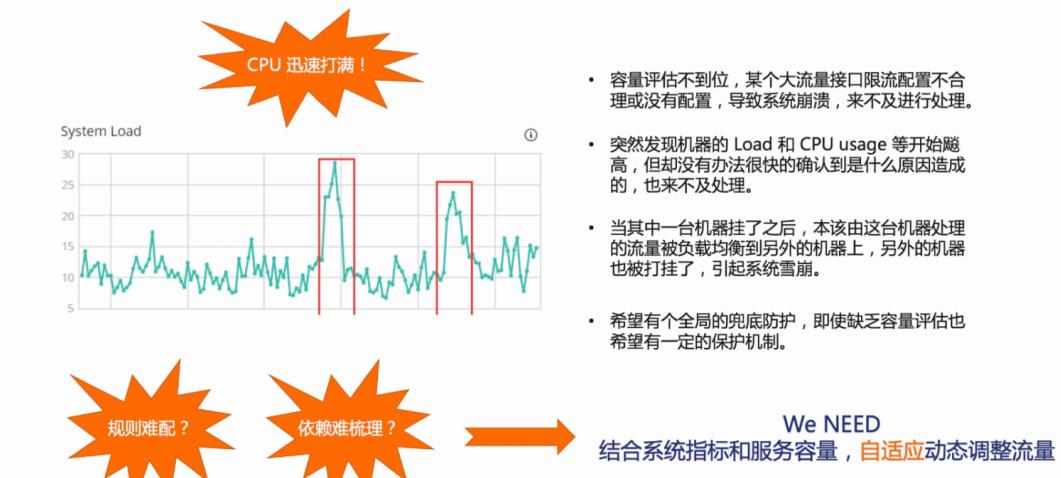
hystric : 线程池隔离 : 硬隔离 ,

问题 :

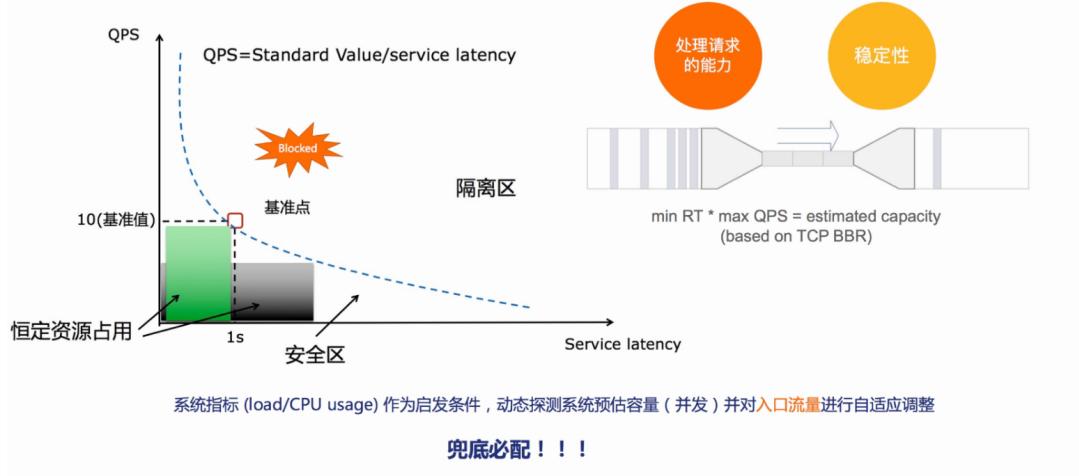
- 增加线程消耗,增加线程上下文切换消耗
- 业务需要线程上下文,threadlocal中信息,例如spring 事务,在隔离之后会因为跨线程导致获取不到.

估算并发 : rt (响应时间) 和 qps (使用较长的响应时间和较高的qps来进行评估)

系统自适应保护



系统自适应保护



基于 tcp 的 bbr 的自适应算法

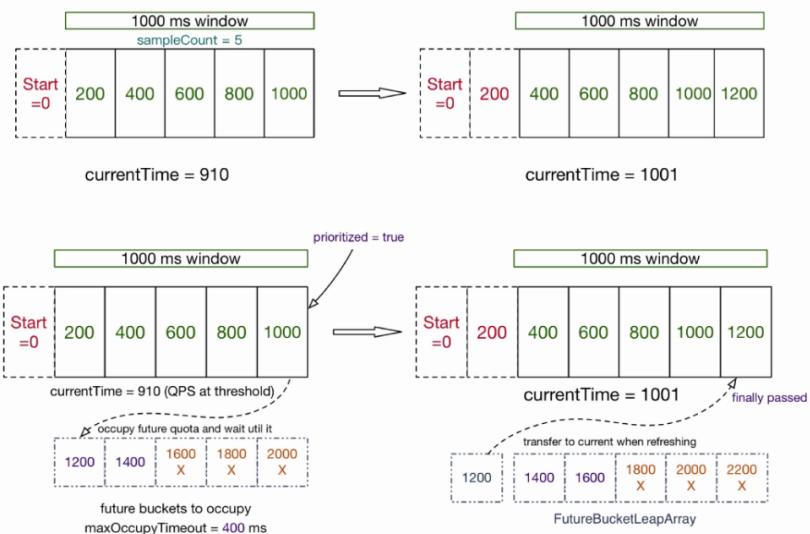
https://www.cnblogs.com/x_wukong/p/7752558.html

能力：

秒级监控的流量，可以导出，需要业务根据需定制展示，例如把数据接入普罗米修斯

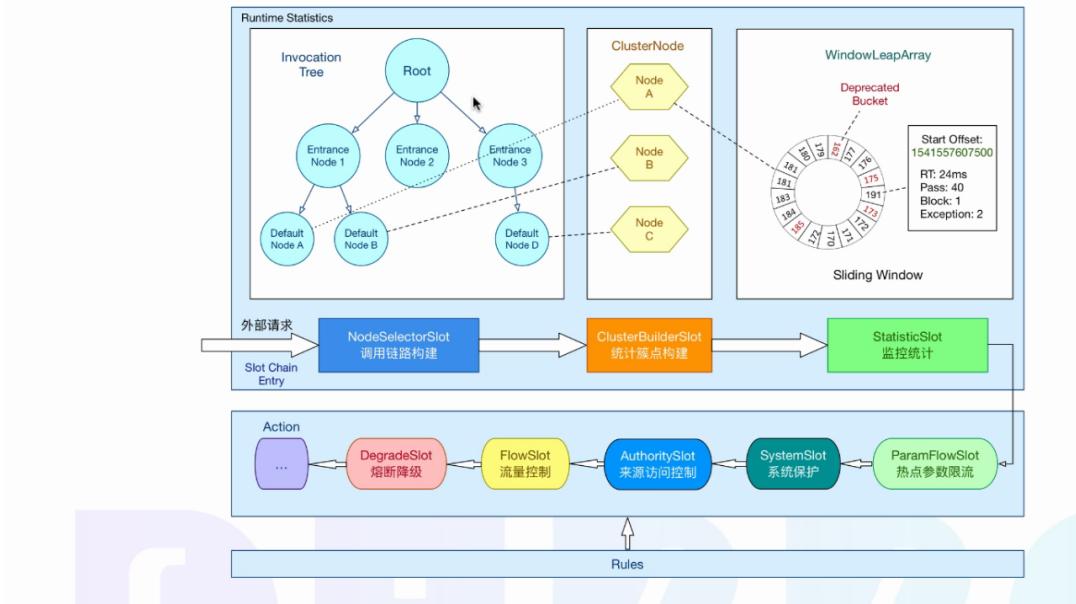
配置动态修改：可以把配置接入阿波罗

Sentinel 核心技术-毫秒级实时统计、



借助滑动窗口,把1s划分更加细粒度,避免1s内的高流量把系统打垮

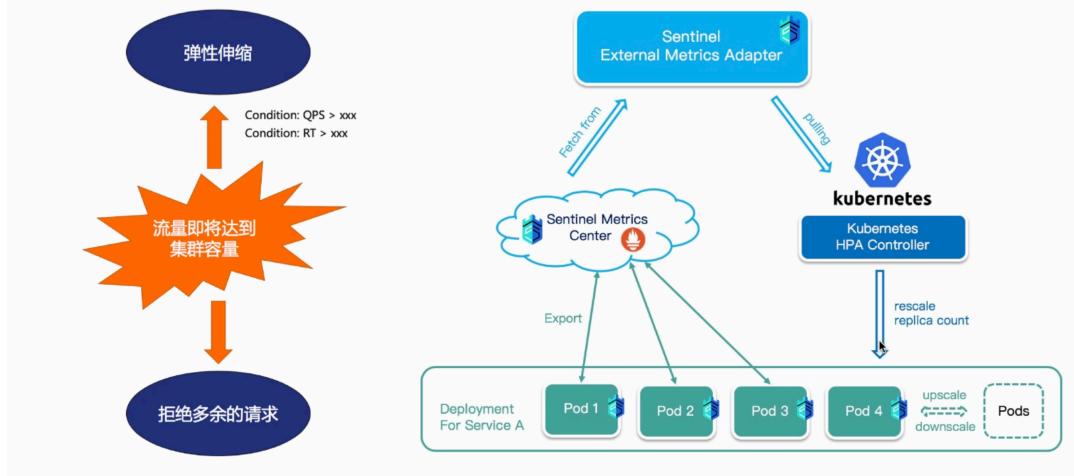
Sentinel 核心技术-整体骨架



	Sentinel	Hystrix	resilience4j
隔离策略	信号量隔离（并发线程数限流）	线程池隔离/信号量隔离	信号量隔离
熔断降级策略	基于响应时间、异常比率、异常数	基于异常比率	基于异常比率、响应时间
实时统计实现	滑动窗口（LeapArray）	滑动窗口（基于RxJava）	Ring Bit Buffer
动态规则配置	支持多种数据源	支持多种数据源	有限支持
扩展性	多个扩展点	插件的形式	接口的形式
基于注解的支持	支持	支持	支持
限流	基于 QPS，支持基于调用关系的限流	有限的支持	Rate Limiter
流量整形	支持预热模式、匀速器模式、预热排队模式	不支持	简单的 Rate Limiter 模式
系统自适应保护	支持	不支持	不支持
控制台	提供开箱即用的控制台，可配置规则、查看秒级监控、机器发现等	简单的监控查看	不提供控制台，可对接其它监控系统

高可用的另一种思路

Kubernetes HPA base on Sentinel metrics



通过sentinel的监控来帮助系统进行水平扩展