# 第七章系统优化项之限流与稳定

如何保障系统稳定和高可用？

Sentinel 介绍

Springboot 整合 Sentinel

Sentinel 和秒杀业务结合

防爬虫与恶意刷请求

## 1. 如何保障系统稳定和高可用？

### 1. 面临的问题

问题1：当秒杀的用户量超过预计，请求量超过服务器最大承载压力怎么办？问题2：当有服务出现故障，不可用时如何应对？

系统保护措施

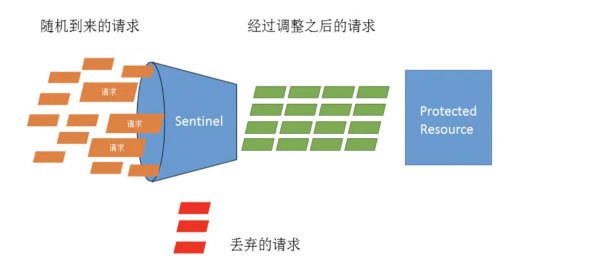
1.限流（Ratelimit），流量控制（flow control）2.服务熔断（Circuit-breaker）

阿里巴巴开源框架 Sentinel， Sentinel 是面向分布式服务架构的轻量级流量控制框架

### 2. 流量控制介绍

对于秒杀系统来说，秒杀时请求瞬间的秒杀流量是不可控的，没办法准确预估多少，但是秒杀系统的处理能力是有限的，当流量过大时服务器会被打挂

流量控制（flow control），其原理是监控应用流量的 QPS 或并发线程数等指标，当达到指定的阈值时对流量进行控制，以避免被瞬时的流量高峰冲垮，保护系统不会被压垮，从而保障应用的高可用性。



### 3. 熔断 & 降级介绍

一旦下游服务C因某些原因变得不可用，积压了大量请求，服务B的请求线程也随之阻塞。线程资源逐渐耗尽，使得服务B也变得不可用。紧接着，服务A也变为不可用，整个调用链路被拖垮。

像这种调用链路的连锁故障，这种不可用像滚雪球一样逐渐放大的时候，叫做雪崩效应。

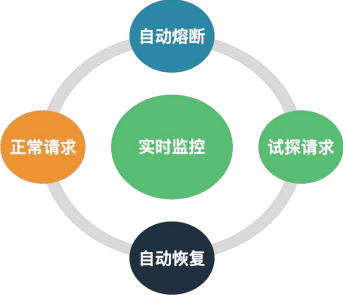
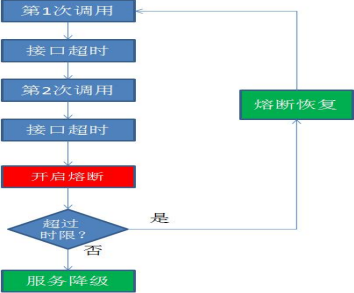
#### 什么是服务熔断？

熔断这一概念来源于电子工程中的断路器（Circuit Breaker）。在互联网系统中，当下游服务因访问压力过大而响应变慢或失败，上游服务为了保护系统整体的可用性，可以暂时切断对下游服务的调用。

这种牺牲局部，保全整体的措施就叫做熔断

如果不采取熔断措施，我们的系统会怎样呢？我们来看一个例子 当前系统中有A，B，C三个服务，服务A是上游，服务B是中游，服务C是下游

熔断器，如同电力过载保护器，在一段时间内侦测到许多类似的错误，就会强迫其以后的调用该服务的快速失败，不再访问该服务，从而防止应用程序不断地尝试执行可能会失败的操作。当检测到该节点微服务响应正常后恢复调用链路。



## 2. Sentinel介绍

git地址： https://github.com/alibaba/Sentinel

### 1. Sentinel 概述

Sentinel 是阿里巴巴出品的面向分布式服务架构的流量控制组件。

背景：

主要以流量为切入点，从限流、流量整形、熔断降级、系统负载保护、热点防护等多个维度来帮助开发者保障微服务的稳定性。



特征

| 丰富的应用场景 | 完备的实时监控 | 广泛的开源生态 |
| --- | --- | --- |
| Sentinel 承接了阿里巴巴  近 10 年的双十一大促流  量的核心场景,例如秒杀  (即突发流量控制在系统  容量可以承受的范围)、  消息削峰填谷、集群流量  控制、实时熔断下游不可  用应用等。 | Sentinel 同时提供实时  的监控功能。您可以在  控制台中看到接入应用  的单台机器秒级数据,  甚至 500 台以下规模  的集群的汇总运行情况。 | Sentinel 提供开箱即用  的与其它开源框架/库  的整合模块。您只需要  引入相应的依赖并进行  简单的配置即可快速地  接入 Sentinel。 |

### 2. Sentinel 的相关概念

#### 1. 资源

资源是 Sentinel 的关键概念

它可以是Java 应用程序中的任何内容。例如,由应用程序提供的服务,或由应用程序调用的其它应用提供的服务,甚至可以是一段代码。

只要是通过Sentinel API定义的代码,就是资源,能够被 Sentinel保护起来。大部分情况下,可以使用方法签名、URL、甚至服务名称作为资源名来标示资源。

#### 2. 规则

围绕资源的实时状态设定的规则，可以包括流量控制规则、熔断降级规则以及系统保护规则。所有规则可以动态实时调整。

#### 3. Sentinel 的优势

Sentinel 对比Hystrix



3. Sentinel 的优势

Sentinel 的优势

控制台 ：开箱即用，可配置规则、查看秒级监控、机器发现等

限流： 支持QPS限流，线程数限流，多种限流策略

降级： 支持按平均返回时间降级，按多种异常数降级，按异常比率降级

负 载保 护： 支持链路的关联，按链路统计限流，系统保护，热门资源保护等等

### 3. SpringBoot整合Sentinel

1. pom 依赖导入

2.创建测试TestController

3.测试

正常访问时候

当快速刷新，使QPS大于2时

浏览器响应结果 控制台打印结果

### 4. Sentinel和秒杀业务结合

#### 代码实践

以查询秒杀活动的列表为例，写代码验证一下

需要新增一个页面 wait.html， 当被限流时候跳转到这个页面

### 5. 防爬虫与恶意刷请求

1. 解决方案

限流机制 Ratelimit Mechanism

验证码机制 Verification Code Mechanism

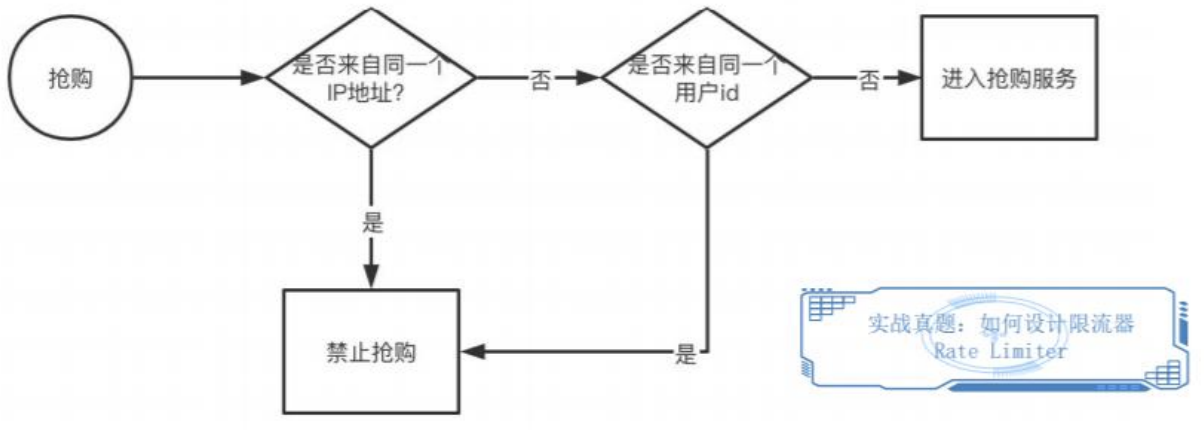
黑名单机制 Blacklist Mechanism

2. 验证码机制

Verification Code Mechanism



3. 限流机制 Ratelimit Mechanism



4. 黑名单机制

Blacklist Mechanism

1 黑名单 IP 地址

2 黑名单用户ID