**流量控制熔断降级**

# 介绍

微服务架构中，微服务之间相互调用，springcloud可以用feign方式和RestTemplate+ribbon方式来实现服务间的相互调用。但如果某一个服务出现问题，所有调用该出问题的服务都将出现阻塞，如果有大量的请求，则Servlet容器的线程资源会被消耗完毕，导致服务瘫痪。服务与服务之间的依赖性，故障会传播，会对整个微服务系统造成灾难性的严重后果，这就是服务故障的“雪崩”效应。为了解决这个问题，业界提出了断路器模型。

# 解决问题

实时熔断

秒杀场景

消息削峰填谷

集群流量控制

# 对比

主要有两个组件可选：netfix的Hystrix和Alibaba的sentinel。



结论：选择Sentinel。

秒级流量分析功能，动态规则实时推送。

专业多样化的防护手段：

入口流量控制：按照服务容量进行流量控制，常用于应用入口，例如： Gateway、前端应用、服务提供方等。

热点隔离：将热点和普通流量隔离出来，避免无效热点抢占正常流量的容量。

对依赖方隔离 / 降级：对应用和应用之间、应用内部采用隔离 / 降级手段，将不稳定的依赖的对应用的影响减至最小，从而保证应用的稳定性。

系统防护：流控降级可以根据系统的能力（例如 Load、CPU 使用率等）来动态调节入口的流量，保证系统稳定性。

实时的单机监控能力，强大的聚合监控和历史监控查询能力。

# 实战Demo

1.获取控制台

您可以从 [release 页面](https://github.com/alibaba/Sentinel/releases) 下载最新版本的控制台 jar 包。

2.引入依赖和配置

|  |
| --- |
| <!--nacos-sentinel--> <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-alibaba-sentinel</artifactId>  <version>0.9.0.RELEASE</version> </dependency> |

application.yml

|  |
| --- |
| spring:  cloud:  sentinel:  transport:  port: 8719  dashboard: localhost:8080 |

Sentinel 的理念是开发者只需要关注那些资源需要保护而不需要考虑如何保护。

|  |
| --- |
| @RequestMapping(value = "/hi", method = RequestMethod.POST)  @SentinelResource(value = "hi", blockHandler = "exceptionHandler")  public String hi(@RequestBody String name) throws InterruptedException {  Thread.sleep(90);  LOGGER.info("My name is " + name);  return "hi " + name + " " + port;  }  // 限流与阻塞处理  public String exceptionHandler(String str, BlockException ex) {  System.err.println("blockHandler：" + str + ex);  return "error";  } |

3.配置sentinel规则并使用nacos进行持久化。

推荐使用推模式：

推送规则正确做法应该是 配置中心控制台/Sentinel 控制台 → 配置中心 → Sentinel 数据源 → Sentinel，而不是经 Sentinel 数据源推送至配置中心。

方式一：sentinel控制台配置

优点：图形化配置方便

缺点：改动量大。

方式二：配置中心控制台配置

优点：改动量小

缺点：通过json方式配置麻烦。

**待解决**

参考https://blog.csdn.net/lilizhou2008/article/details/97075236

|  |
| --- |
| com.alibaba.csp.sentinel.slots.block.flow.FlowException流控  com.alibaba.csp.sentinel.slots.block.degrade.DegradeException 降级RT  com.alibaba.csp.sentinel.slots.block.flow.param.ParamFlowException 热点参数 |

# 参考资料

<https://help.aliyun.com/document_detail/101132.html>

<https://github.com/alibaba/spring-cloud-alibaba/wiki/Sentinel>

<https://cloud.tencent.com/developer/article/1409932>