链滴

流量防护框架 Sentinel 初体验

作者：flowaters

原文链接：https://ld246.com/article/1535768344830

来源网站：链滴

背景

本文介绍的Sentinel是阿里巴巴最近开源的一个高可用防护的流量管理框架

特点

Sentinel的特点是，本文简单归纳了一下，如下：

1. 流量控制：支持多种流控策略，如线程数 / QPS / RT / 异常
2. 熔断降级：下游应用不可用时，支持实时熔断
3. 系统负载保护：突发流量时，只放过系统可以承受的流量，拦截多余的流量
4. 实时监控和控制台：通过SpringBoot开发控制台，支持单台和集群的实时监控

稳定性

Sentinel 承接了阿里巴巴近 10 年的双十一大促流量的核心场景，有阿里巴巴背书，所以可以放心的。

Dubbo生态

Dubbo已经通过投票，进入了Apache 基金会孵化器。

Sentinel的适配器已经捐给了Dubbo，可以在Dubbo中直接使用Sentinel。

活跃度

Sentinel最近在全国各地举办meetup，而且有钉钉群/微信群可以交流讨论，开发者非常活跃，版本新也比较快。

初体验

仅仅需要三步，五分钟，不需要额外的依赖。

pom.xml

引入Sentinel的依赖

```
<dependency>
    <groupId>com.alibaba.csp</groupId>
    <artifactId>sentinel-core</artifactId>
    <version>0.1.1</version>
</dependency>
```

业务代码

将业务代码用SphU包起来，如

```
    public static void main(String[] args) {
        initFlowRules();
        while (true) {
            Entry entry = null;
            try {
                entry = SphU.entry("HelloWorld");
                System.out.println("hello world");
            } catch (BlockException e1) {
                System.out.println("block!");
            } finally {
                if (entry != null) {
                    entry.exit();
                }
            }
        }
    }
```

设置流控策略

流按策略和业务代码是分离的，如下：

```
    private static void initFlowRules() {
        List<FlowRule> rules = new ArrayList<FlowRule>();
        FlowRule rule = new FlowRule();
        rule.setResource("HelloWorld");
        rule.setGrade(RuleConstant.FLOW_GRADE_QPS);
        // Set limit QPS to 20.
        rule.setCount(20);
        rules.add(rule);
        FlowRuleManager.loadRules(rules);
    }
```

而且流控策略支持动态下发。这个本文还没有尝试。

运行效果

输出了20个hello world，剩下的请求会被block。

```
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
hello world
```

```
hello world
hello world
hello world
hello world
hello world
hello world
block!
block!
block!
block!
block!
block!
block!
```

查看日志

Sentinel在本地~/logs/csp目录下，已经保存了一份运行日志。所以看到本地多了一个logs文件夹千不要惊讶，我已经习惯了。

日志内容如下：

```
tail testSentinel.MainSentinel-3577-metrics.log.2018-09-01

|--timestamp-|------date time----|-resource-|p |block|s |e|rt
1535765758000|2018-09-01 09:35:58|HelloWorld|20|36859|20|0|1
1535765759000|2018-09-01 09:35:59|HelloWorld|20|111935|20|0|0
```

其中各列的含义

- p: 通过的请求
- block: 被阻止的请求
- s: 成功处理的请求个数
- e: 用户自定义的异常
- rt: 平均响应时长

Sentinel 与 Hystrix 的对比

传统的熔断库是 Netflix Hystrix，我在前面也写过一篇Hystrix初体验(一): 原理。

Sentinel 和 Hystrix 有什么异同呢?

本文偏使用，就不多介绍啦，有兴趣的可以看Sentinel 与 Hystrix 的对比。

后记

微服务的生态还在快速发展，对于我们使用者，多一个选择，总是好事情。

参考

- Sentinel github

- Sentinel 与 Hystrix 的对比

原文链接：流量防护框架 Sentinel 初体验