ALIBABA CLOUD

阿里云

应用高可用服务

产品简介

文档版本：20220304

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.什么是应用高可用服务AHAS 05
2.名词解释 09
3.基础知识 11
3.1. 基本概念 11
3.2. 流量控制 13
3.3. 熔断降级 16
3.4. 系统保护 16
4.AHAS 使用指引 17

1.什么是应用高可用服务AHAS

应用高可用服务（Application High Availability Service）是一款专注于提高应用高可用能力的SaaS产品，包含架构感知、流量防护、故障演练和功能开关四大独立的功能模块。

- 架构感知：
  - 自动感知应用的拓扑结构。
  - 以可视化的方式直观呈现应用对基础架构的依赖关系和组件间的依赖关系。
  - 持续记录上述依赖关系。
- 流量防护：
  - 专业化多样化的限流手段。
  - 实时秒级的监控。
  - 立即生效的规则管理。
- 故障演练：
  - 提供基于真实线上故障的高可用能力演练服务。
  - 根据您的应用架构智能推荐故障演练场景。
- 功能开关：
  - 按照统一的方式定义、管理业务运行时功能切换。
  - 支持定义任意类型的配置项，且自动根据类型做自动拆装，代码层无需关注类型信息，简化运行时配置项使用流程。

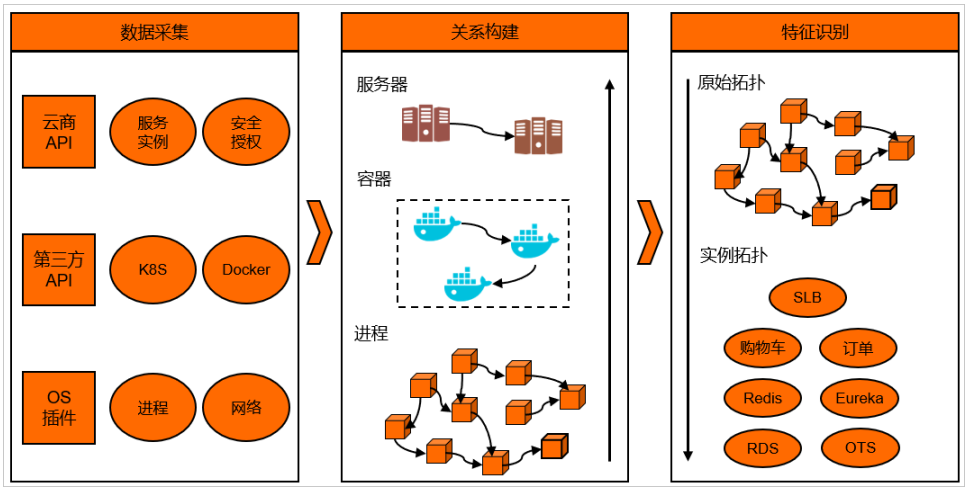
架构感知

- **功能**
  服务器、存储、网络是现代云平台的基础设施。随着上云战略的推进，越来越多的大型企业将业务、服务、系统构建在云平台上。开源软件和云服务的多样性，开发语言的异构性，以及企业IT团队的组织和能力差异，都提高了标准化的复杂性。
  在此背景下，架构感知功能应运而生。它会采集和分析操作系统及第三方标准接口，捕捉进程级的调用关系，并使用特征库算法识别进程所使用的技术组件，最后在服务器、容器和进程这三个维度上以可视化的方式展示应用架构。
- **AHAS支持的数据源**

	服务器	容器		进程		
	服务器	第三方容器服务	云容器服务	客户业务服务	第三方服务	云 SaaS
特征数据	Cloud OpenAPI	Docker OpenAPI	Docker OpenAPI	探针	探针	Cloud OpenAPI
	探针	K8s OpenAPI	K8s OpenAPI			
			Cloud OpenAPI			
依赖数据	计算	计算	Cloud OpenAPI	探针	探针	Cloud OpenAPI
			探针			
运行时数据	Cloud OpenAPI	Docker OpenAPI	Docker OpenAPI	探针	探针	Cloud OpenAPI
	探针	K8s OpenAPI	K8s OpenAPI			

云商 API	开源 API	AHAS 能力	无

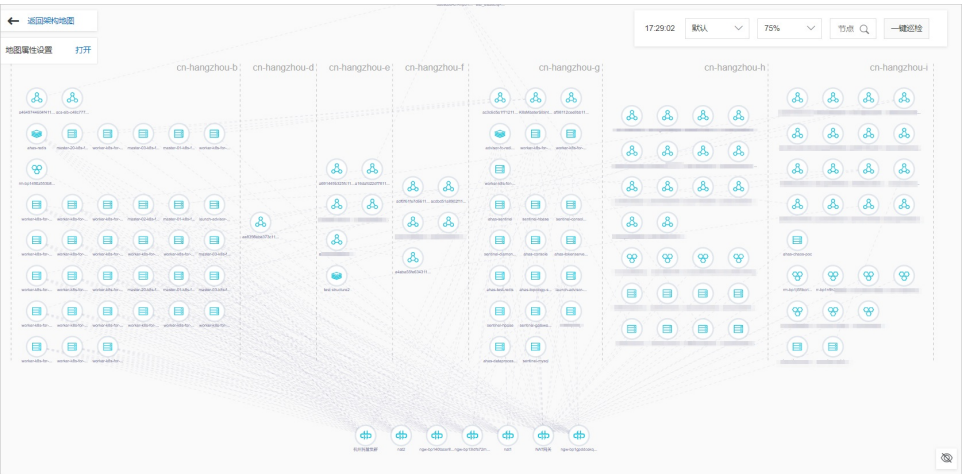
- **工作流程**
  架构感知工作流程



AHAS架构感知的工作流程包括四个步骤：

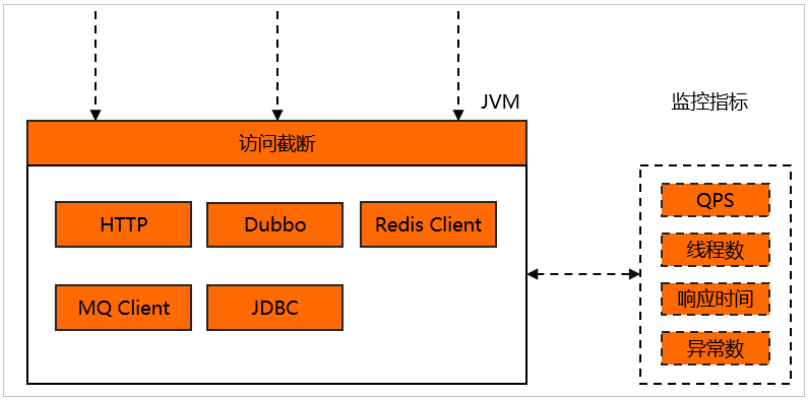
i. 数据采集
ii. 关系构建
iii. 特征识别
iv. 架构可视化

- **界面展现**
  架构可视化



流量防护

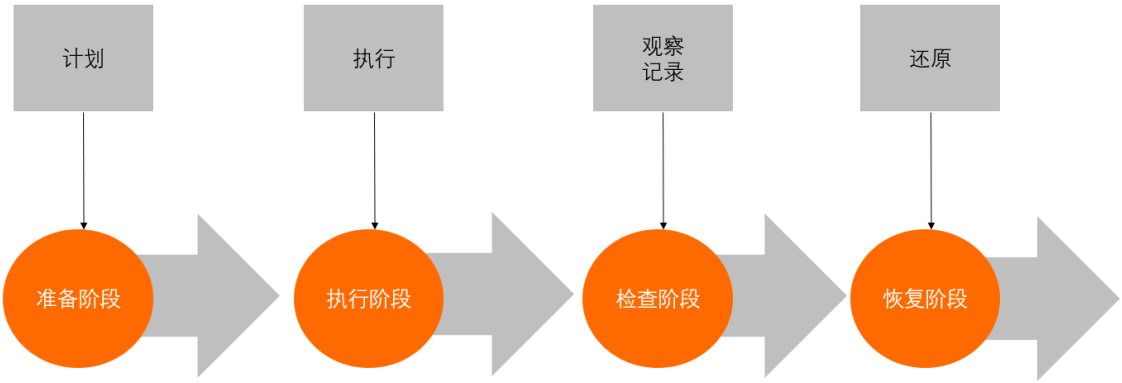
- **功能**
  在当今日益复杂的应用环境下，应用设计应该遵循“面向失败”的设计原则，对上下游的依赖零信任。借助流量控制、熔断降级模块，您的应用有能力对流量采取限流控制，并对下游依赖进行降级处理。
- **工作流程**
  流量防护模块支持主流的Java框架，包括HTTP和Dubbo。该模块可以实时监控框架的QPS（Queries per Second，每秒查询数）、线程数、响应时间、异常数等指标，并有选择地截断对这些框架的访问，从而保护应用的可用性。此外，利用AHAS提供的SDK，您还可以采取更细粒度的代码级限流降级防护措施。
  流量防护工作流程



故障演练

故障演练是一款遵循混沌工程实验原理并融合了阿里巴巴内部实践的产品，提供丰富故障场景实现，能够帮助分布式系统提升容错性和可恢复性。

- **流程**
  故障演练建立了一套标准的演练流程，包含准备阶段、执行阶段、检查阶段和恢复阶段。通过四阶段的流程，覆盖用户从计划到还原的完整演练过程，并通过可视化的方式清晰的呈现给用户。



- **适用场景**
  故障演练可适用于以下典型场景：
  - 衡量微服务的容错能力。
  - 验证容器编排配置是否合理。
  - 测试PaaS层是否健壮。
  - 验证监控告警的时效性。
  - 定位与解决问题的应急能力。

  更多信息，请参见什么是故障演练。

功能开关

功能开关是一个轻量级的框架，既可用于集中管理各应用的业务开关，也可以管理应用的运行时基础配置。运行时动态配置修改

- 功能描述
  - 业务开关管理：按照统一的方式定义、管理业务运行时功能切换。
  - 运行时配置管理：支持定义任意类型的配置项，且自动根据类型做自动拆装，代码层无需关注类型信息，简化运行时配置项使用流程。

- 使用场景
  - 业务开关管理。一个业务经常由多个系统、多个功能模块组成，为保证某些业务的动态性，后端程序通常会用开关来控制程序的逻辑，以达到在系统运行时切换运行逻辑的目的。
  - 运行时配置管理。无需写死的URL、接口名、阈值和读取文件用的编码、黑白名单等，直接使用功能开关即可方便快速创建运行时能覆盖的动态配置。

AHAS使用场景

- **微服务高可用**
  如今，Spring Cloud、Kubernetes的微服务架构对服务的高可用性提出了前所未有的挑战。借助AHAS，您可以在零代码改动的前提下，快速使用SaaS化的高可用服务，包括架构可视化、架构变化追踪、故障演练和流量防护的保护。
  AHAS支持Spring Cloud组件的快速发现、故障演练能力，以及SpringBoot应用的一键流量防护。
  AHAS支持阿里云容器服务Kubernetes应用的快速接入。
- **传统应用高可用**
  由于缺乏高可用能力，对于传统的Java单体应用和分布式应用而言，如何保障稳定性和连续性向来是个难题。
  AHAS提供了应用高可用保障所必需的架构实时展现与追踪、架构高可用性故障演练，以及Java应用零代码改动接入流量防护的能力。即便是已上线的应用，也无需升级改造。此外，对于引入的开源或第三方组件，AHAS可以进行自动化智能评估，并提供优化使用方面的指导。

相关文档

- 第三方组件和云服务支持列表
- 开源Sentinel与商业化AHAS流量防护对比

2.名词解释

本文列出了应用高可用服务 AHAS 的主要术语。

故障演练相关术语

标签

即用户对演练自定义的内容。一个演练可以有多个标签，便于用户对演练进行分类和管理。

场景

即活动的分类，例如磁盘满载和磁盘IO高都属于磁盘这个场景。

草稿

即未配置完成的演练或修改后未保存的演练。

活动

即演练流程中的一个可运行的节点，一个演练由多个活动组成。

演练

即一个完整的、可执行的流程配置。

演练任务

即一次演练的执行记录，每执行一次演练就产生一个任务。

演练报告

即一个演练任务的结果，一个演练任务对应一份报告。

演练参数

即一个活动运行时所需要的参数，参数决定了活动执行的结果。

演练对象

即实施演练的目标主机或设备。目前以活动为最小颗粒度，不同活动可以对不同的主机或设备执行。

流量防护相关术语

规则

即围绕应用资源的实时状态，设定规则（Rule）控制资源调用。规则类型包括：流控规则、降级规则、隔离规则。

流量防护

AHAS 流量防护（限流降级）提供专业多样化的流量控制、熔断降级手段，并提供强大的监控信息，以及实时的动态规则推送能力，为您的应用和服务保驾护航。

流量控制

即监控应用流量的 QPS 或并发线程数等指标，控制达到阈值的流量，避免应用被瞬时的流量高峰冲垮，从而保障应用的高可用性。

熔断降级

即限制不稳定资源的调用，让请求快速失败，避免影响到其它的资源而导致级联错误。

系统保护

即从整体维度对应用入口流量进行控制，结合应用的 Load、总体平均 RT、入口 QPS 和线程数等维度的监控指标，让系统的入口流量和系统的负载达到一个平衡，让系统尽可能跑在最大吞吐量的同时保证系统整体的稳定性。

说明 Sentinel 基于 TCP BBR 拥塞控制算法实现流量塑形，对入口流量进行系统保护。

3.基础知识

3.1. 基本概念

本文介绍了应用高可用服务AHAS涉及的基本概念和四大子模块的基本概念，了解这些基本概念有助于您更好地使用本产品。

- 通用概念：
  - 什么是应用高可用服务环境（AHAS 环境）？
  - 什么是探针？
- 架构感知相关概念：
  - 什么是架构和架构组件？
- 流量防护相关概念：
  - 什么是资源？
  - 什么是规则？
  - 什么是流控（限流）？
  - 什么是熔断降级？
  - 什么是系统防护？

  关于流量防护的详细说明，请参见什么是应用防护和网关防护概述。
- 故障演练相关概念：
  - 什么是故障演练？
- 功能开关相关概念：
  - 什么是功能开关？
  - 功能开关的使用场景？

什么是应用高可用服务环境（AHAS 环境）？

通常情况下，用户有生产、测试、开发环境，各环境的资源是隔离的。AHAS通过“环境”划分各环境下的应用高可用服务作用范围，包含一个环境下的应用高可用服务资源。

AHAS环境按地域划分，每个地域会有一个默认（Default）环境，您也可以添加自定义环境。

什么是探针？

应用高可用服务AHAS包括以下两种探针：

- 应用高可用探针（即AHAS探针）：这是安装在操作系统上的独立进程，用来采集架构信息。使用架构感知、故障演练功能前需安装此探针。
- 应用高可用Java探针：这是针对JVM的Java探针，通过字节码增强技术进行实时监控和流量防护。如果需要使用流量防护功能，可安装此探针。

什么是架构和架构组件？

- 架构分为水平和垂直两个维度：
  - 水平架构：进程拓扑、容器拓扑、主机拓扑
  - 垂直架构：进程、容器和主机之间的依赖关系
- 架构组件是指架构的组成部分，包含进程（应用进程、第三方组件进程、云服务）、容器、主机。

什么是故障演练？

故障演练是一款遵循混沌工程实验原理并融合了阿里巴巴内部实践的产品，提供丰富故障场景实现，能够帮助分布式系统提升容错性和可恢复性。

更多名词解释如下：

名词	名词解释
演练	一个完整的、可执行的流程配置。
草稿	未配置完成的演练或修改后未保存的演练。
演练任务	一次演练的执行记录，每执行一次演练就产生一个任务。
演练报告	一个演练任务的结果，一个演练任务对应一份报告。
阶段	活动的集合，一个演练共分4个阶段，每个阶段包含多个活动，演练会按照阶段顺序执行。
活动	演练流程中的一个可运行的节点，一个演练由多个活动组成。
演练参数	一个活动运行时所需要的参数，参数决定了活动执行的结果。
演练对象	实施演练的目标主机或设备。目前以活动为最小颗粒度，不同活动可以对不同的主机或设备执行。
场景	活动的分类，例如磁盘满载和磁盘IO高都属于磁盘这个场景。
标签	用户对演练自定义的含义。一个演练可以有多个标签，便于用户对演练进行分类和管理。

什么是资源？

只要通过应用防护SDK定义的代码，即资源，就能够被应用防护保护起来。

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称来标识资源。资源可以是Java应用程序中的任何内容，例如应用程序提供的Dubbo服务或HTTP API，或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。

示例：

通过应用防护SDK定义的资源如下所示。

```
Entry entry = null;
try {
    entry = SphU.entry("HelloWorld");
    // 正常的业务逻辑
    System.out.println("hello world");
} catch (BlockException e1) {
   // 处理资源被限制的情况
    System.out.println("block!");
} finally {
    if (entry != null) {
         entry.exit();
    }
}
```

然后调用此资源，那么在AHAS控制台就可以看到 `HelloWorld` 这个资源了。

什么是规则？

在AHAS中围绕应用或应用资源的实时状态来设定规则，包括流控规则、降级规则和系统保护规则。

所有规则都会推送至该应用或应用资源所在的每台机器上，并针对单台机器生效。您可以通过控制台动态实时调整规则。

什么是流控（或称限流）？

流控，即流量控制，是AHAS提供的一种应用高可用防护能力。流量控制包括限流、冷启动、削峰填谷等各种流量塑形手段。限流是根据流量，当达到您指定的阈值时立即拦截流量，进行限流，以避免被瞬时的流量高峰冲垮，从而保障您的应用高可用性。

什么是熔断降级？

熔断降级是分布式服务中不可缺少的一种防护手段。其原理是当应用不稳定时，通常表现为响应时间或异常比例升高，则将不稳定应用降级甚至熔断，防止关联应用被不稳定的应用拖垮。

什么是系统防护？

系统防护是从系统表现，例如Load、RT、QPS和线程数四个维度出发，对应用的**入口流量**进行控制，让系统尽可能跑在最大吞吐量的同时保证系统稳定性。

什么是功能开关？

功能开关是一个轻量级的运行时动态配置修改框架，既可用于集中管理各应用的业务开关，也可以管理应用的运行时基础配置。

功能开关的使用场景？

- 业务开关管理。一个业务经常由多个系统、多个功能模块组成，为保证某些业务的动态性，后端程序通常会用开关来控制程序的逻辑，以达到在系统运行时切换运行逻辑的目的。
- 运行时配置管理。无需写死的URL、接口名、阈值和读取文件用的编码、黑白名单等，直接使用功能开关即可方便快速创建运行时能覆盖的动态配置。

3.2. 流量控制

流控，即流量控制（Flow Control），根据流量、并发线程数、响应时间等指标，把随机到来的流量调整成合适的形状，即流量塑形。避免应用被瞬时的流量高峰冲垮，从而保障应用的高可用性。
流量控制有以下几个角度：

- 运行指标，例如QPS、并发线程数等。
- 资源的调用关系，例如资源的调用链路，资源和资源之间的关系，调用来源等。
- 控制效果，例如排队等待、直接拒绝、Warm Up（预热）等。

一条流控规则主要由下面几个因素组成，我们可以组合这些元素来实现不同的限流效果：

- 资源名：即流控规则的作用对象。
- 阈值：流控阈值。
- 阈值类型：流控阈值类型（QPS或并发线程数）。
- 来源应用：流控规则针对的调用来源，若为 `default` 则不区分调用来源。
- 流控方式：调用关系限流策略。
- 流控模式：即流控效果，包括快速失败、Warm Up、排队等待模式。

运行指标

流量控制支持通过以下运行指标进行流量塑形。

- 基于QPS
  QPS模式根据资源的实时QPS进行流量控制。当QPS超过指定阈值时采取相应的控制策略，更多信息，请参见控制效果。
- 基于线程数
  线程数模式按照资源的并发线程数（即该资源正在执行的线程数）进行流量控制。
  并发线程数限流用于保护业务线程数不被耗尽。例如，当应用所依赖的下游应用由于某种原因导致服务不稳定、响应延迟增加，对于调用者来说，意味着吞吐量下降和更多的线程数占用，极端情况下甚至导致线程池耗尽。为应对太多线程占用的情况，业内有使用隔离的方案，比如通过不同业务逻辑使用不同线程池来隔离业务自身之间的资源争抢（线程池隔离）。这种隔离方案虽然隔离性比较好，但是代价就是线程数目太多，线程上下文切换的overhead比较大，特别是对低延时的调用有比较大的影响。Sentinel并发线程数限流不负责创建和管理线程池，而是简单统计当前请求上下文的线程数目，如果超出阈值，新的请求会被立即拒绝，效果类似于信号量隔离。

资源的调用关系

调用关系包括调用方和被调用方。一个函数可能会去调用另一个函数，形成一个调用链路的层次关系。Sentinel会记录调用链路的实时统计信息，因此可以衍生出多种按调用关系进行流量控制的手段。

流控模式支持以下几种资源的调用关系：直接模式（按调用方限流）、关联模式（关联资源限流）和链路模式（调用链路限流）。

- 根据调用来源限流
  直接模式下会针对流控规则中的来源应用进行流量控制。
  流控规则中的 `limitApp` 字段用于根据调用方进行流量控制。该字段的值有以下三种选项，分别对应不同的场景：
  - `default` ：表示不区分调用来源，来自任何调用者的请求都将进行限流统计。如果这个资源名的调用总和超过了这条规则定义的阈值，则触发限流。
  - `{some_origin_name}` ：表示针对特定的调用者，只有来自这个调用者的请求才会进行流量控制。例如 `NodeA` 配置了一条针对调用者 `caller1` 的规则，那么当且仅当来自 `caller1` 对 `NodeA` 的请求才会触发流量控制。
  - `other` ：表示针对除 `{some_origin_name}` 以外的其余调用方的流量进行流量控制。例如，资源 `NodeA` 配置了一条针对调用者 `caller1` 的限流规则，同时又配置了一条调用者为 `other` 的规则，那么任意来自非 `caller1` 对 `NodeA` 的调用，都不能超过 `other` 这条规则定义的阈值。
    直接模式下直接控制来自调用来源的访问流量；如果来源应用为default则不区分调用来源。
- 关联限流

关联模式会控制当前资源的关联资源的流量。
当两个资源之间具有资源争抢或者依赖关系的时候，这两个资源便具有了关联。比如对数据库同一个字段的读操作和写操作存在争抢，读的速度过高会影响写得速度，写的速度过高会影响读的速度。如果放任读写操作争抢资源，则争抢本身带来的开销会降低整体的吞吐量。可使用关联限流来避免具有关联关系的资源之间过度的争抢，举例来说，`read_db` 和 `write_db` 这两个资源分别代表数据库读写，我们可以给 `read_db` 设置限流规则来达到写优先的目的：设置**关联资源**为 `write_db` 。这样当写库操作过于频繁时，读数据的请求会被限流。

- 链路限流
  链路模式下会控制该资源所在的调用链路入口的流量。需要在规则中配置**入口资源**，即该调用链路入口的上下文名称。
  一棵典型的调用树如下所示：

```
machine-root
                    /        \
                   /          \
             Entrance1      Entrance2
                /              \
               /                \
      DefaultNode(nodeA)    DefaultNode(nodeA)
```

上图中来自入口 `Entrance1` 和 `Entrance2` 的请求都调用到了资源 `NodeA` ，Sentinel允许只根据某个调用入口的统计信息对资源限流。比如链路模式下设置入口资源为 `Entrance1` 来表示只有从入口 `Entrance1` 的调用才会记录到 `NodeA` 的限流统计当中，而不关心经 `Entrance2` 到来的调用。

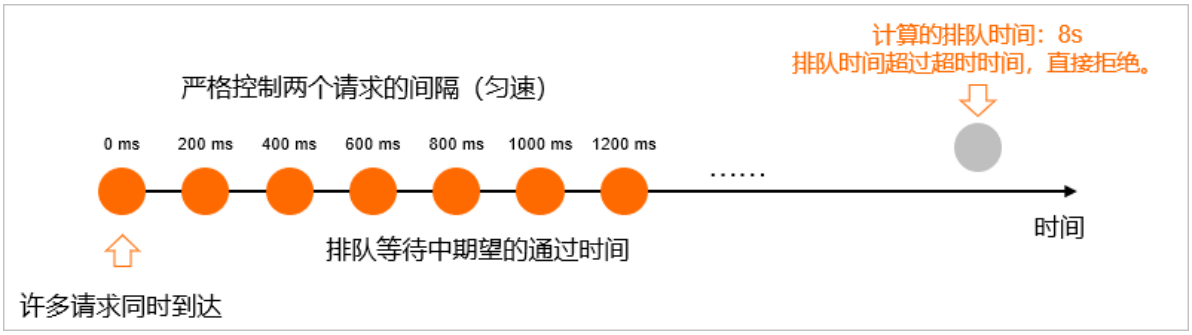
调用链的入口（上下文）是通过API方法 `ContextUtil.enter(contextName)` 定义的，其中contextName即对应调用链路入口名称。

控制效果

流控方式（控制效果）指的是，当阈值类型为QPS时，选择如何控制流量，处理被拦截的流量，实现流量塑形。流量控制的手段包括以下几种：快速失败、Warm Up、排队等待。

> **说明** 控制效果仅对QPS限流生效。

- 快速失败方式
  快速失败方式是默认的流量控制方式，当QPS超过规则的阈值后，新的请求就会被立即拒绝，拒绝方式为抛出 `FlowException` 。这种方式适用于对系统处理能力确切已知的情况下，比如通过压测确定了系统的准确水位时。
- Warm Up模式
  当系统长期处于空闲的情况下，当流量突然增加时，直接把系统拉升到高水位可能瞬间把系统压垮。通过Warm Up模式（预热模式），让通过的流量缓慢增加，经过设置的预热时间以后，到达系统处理请求速率的设定值。
  Warm Up模式默认会从设置的QPS阈值的1/3开始慢慢往上增加至QPS设置值。
- 排队等待模式
  排队等待模式下会严格控制请求通过的间隔时间，即请求会**匀速**通过，允许部分请求排队等待，通常用于消息队列削峰填谷等场景。需设置具体的超时时间，当计算的等待时间超过超时时间时请求就会被拒绝。例如，QPS配置为5，则代表请求每200 ms才能通过一个，多出的请求将排队等待通过。超时时间代表最大排队时间，超出最大排队时间的请求将会直接被拒绝。排队等待模式下，QPS设置值不要超过1000（请求间隔1 ms）。



3.3. 熔断降级

除流量控制以外，对调用链路中不稳定的资源进行熔断降级也是保障高可用的重要措施之一。
由于调用关系的复杂性，如果调用链路中的某个资源不稳定，最终会导致请求发生堆积。**熔断降级**会在调用链路中某个资源出现不稳定状态时（例如调用超时或异常比例升高），对这个资源的调用进行限制，让请求快速失败，避免影响到其它的资源而导致级联错误。

Sentinel 熔断降级支持以下几种策略：

- **秒级 RT 模式**：若持续进入 5 个请求，它们资源的平均响应时间都超过阈值（秒级平均 RT，以 ms 为单位），资源调用会被熔断。在接下的降级时间窗口（在降级规则中配置，以 s 为单位）之内，对这个方法的调用都会自动地返回（抛出 `DegradeException` ）。
- **秒级异常比例模式**：当资源的每秒异常数占通过量的比值超过阈值之后，资源进入降级状态，即在接下的降级时间窗口（在降级规则中配置，以 s 为单位））之内，对这个方法的调用都会自动地返回。异常比率的阈值范围是 `[0.0, 1.0]` ，代表 0% - 100%。
- **分钟级异常数模式**：当资源最近 1 分钟的异常数目超过阈值之后会进行熔断。

3.4. 系统保护

系统保护机制从整体维度对应用入口流量进行控制，结合应用的Load、总体平均RT、入口QPS和线程数等几个维度的监控指标，让系统的入口流量和系统的负载达到一个平衡，对多余的流量直接拒绝，让系统尽可能维持在最大吞吐量的同时保证系统整体的稳定性。

> **说明** 系统保护规则是整体维度的，而不是资源维度的，并且**仅对入口流量生效**。入口流量指的是进入应用的流量，比如Web服务或Dubbo服务端接收的请求，都属于入口流量。

对于一个应用来说，每种相同的系统保护规则最多只能存在一条，即一个应用最多配置五条系统保护规则。

系统规则支持五种阈值类型：

- **Load**（仅对Linux、Unix-like机器生效）：当系统load1超过阈值，且系统当前的并发线程数超过预计的系统容量时才会触发系统保护。预计的系统容量由系统的 `maxQps * minRt` 计算得出。
- **CPU使用率**：当系统CPU使用率超过阈值（0.0~1.0）即触发系统保护。
- **RT**：当单台机器上所有入口流量的平均RT达到阈值即触发系统保护，单位是毫秒。
- **线程数**：当单台机器上所有入口流量的并发线程数达到阈值即触发系统保护。
- **入口QPS**：当单台机器上所有入口流量的QPS达到阈值即触发系统保护。

4.AHAS 使用指引

本文列出了AHAS的使用指引，帮助您快速学习和使用AHAS。

开通 AHAS

架构感知

AHAS的架构感知模块能够自动感知您的系统架构，以可视化的方式直观呈现应用对基础架构的依赖关系，以及组件间的依赖关系。

应用防护

AHAS应用防护以流量为切入点，从流量控制、熔断降级、系统保护等多个维度来帮助您保障服务的稳定性。

网关防护

AHAS网关防护可以对网关进行流量控制，从流量入口处拦截激增的流量，防止下游服务被冲垮。

Nginx防护

通过Nginx Sentinel模块可以快速接入到Nginx防护中。

故障演练

故障演练是一款遵循混沌工程实验原理并融合了阿里巴巴内部实践的产品，提供丰富故障场景实现，能够帮助分布式系统提升容错性和可恢复性。

功能开关

功能开关是一个轻量级的运行时动态配置修改框架，既可用于集中管理各应用的业务开关，也可以管理应用运行时的基础配置。

接入
接入ECS主机
快速接入容器服务Kubernetes
手动接入容器服务Kubernetes

普通Linux主机接入

接入
接入Java应用
接入Java SDK应用
接入Go应用
接入PHP应用

接入
通过Agent接入
接入Spring Cloud Gateway应用
接入Spring Cloud Zuul应用
接入容器服务 Kubernetes 版应用

接入
接入Nginx

接入
接入演练应用
安装故障演练探针

接入
使用 SDK 接入
使用 Spring Boot Starter 接入

控制台指南
云资源视图
Kubernetes资源视图
应用视图

风险视图
Kubernetes监控视图

控制台指南
配置流控规则
配置隔离规则
配置降级规则
配置系统规则
配置热点规则
集群流控
应用概览

控制台指南
接口详情
API管理
机器监控
API流控规则

控制台指南
Nginx Sentinel模块配置

控制台指南
创建演练
执行演练
停止演练
强弱依赖治理方案

控制台指南
新增功能开关
查看功能开关

设置开关推送
查看历史记录

相关信息
第三方组件和云服务支持列表
探针安装常见问题

相关信息
流量防护原则
开源Sentinel与商业化AHAS流量防护对比
支持列表
流量控制
熔断降级

相关信息
触发网关防护规则后的限流策略

相关信息
演练场景说明

最佳实践
运行时动态调整日志级别
主动降级业务功能
快速实现黑白名单功能

管理
管理环境

管理探针

访问控制概述

RAM 主子账号授权

跨云账号授权

AHAS服务关联角色