



【稳定性】稳定性建设之依赖设计

京东云开发者

2024-03-21

 1,545

 阅读11分钟

智能总结

复制

重新生成

这篇文章围绕分布式微服务中强弱依赖治理展开，介绍了依赖的概念和分类，包括最强、强、弱、最弱依赖等，以及业务域、系统启动、基础技术服务等依赖分类。还阐述了强弱依赖治理的目标、工具扫描和人工梳理等实施步骤，指出治理需确定关系、分析数据、制定方案并持续改进，以提升系统稳定性。

关联问题: [如何确定依赖强弱?](#) [弱依赖如何容错?](#) [怎样优化依赖关系?](#)

基于该文章内容继续向AI提问

背景

随着分布式微服务的发展，一个普通的应用可能会依赖于许多其他服务，这给系统的限流降级、优化改造等操作带来了困难。在没有明确强弱依赖关系的情况下，我们很难有效地进行这些操作。为了解决这个问题，强弱依赖治理成为了一种科学的手段。通过强弱依赖治理，我们可以持续稳定地获取应用间的依赖关系、流量以及强弱等数据。这样，我们可以提前发现由于依赖问题可能导致的系统稳定性故障。

一、依赖概念

依赖原则是去除依赖、弱化依赖、控制依赖。多一个依赖多一分风险。能不依赖则不依赖，能异步弱依赖不要同步强依赖。

(1) 最强依赖

当所依赖的服务不可用时，服务不可用，且造成系统崩溃。对于所有的依赖，不建议最强依赖。

(2) 强依赖

假定服务A依赖于服务B，服务B出现故障不可用时，服务A也不可用，通常服务A会返回错误信息，且当所依赖的B服务恢复后自动恢复，我们称这种依赖为强依赖。服务只可强依赖于同等级或高等级的服务与资源。

案例：商详结算下单服务对于库存服务的依赖就属于强依赖，下单时必须校验是否有库存。


(3) 弱依赖

假定服务A依赖于服务B，服务B出现故障不可用时，服务A仍然可用，通常服务A会返回正确信息，只是与服务B相关的信息会不返回或者做默认处理，损失一些次级功能，我们称这种依赖为弱依赖。

案例：下单服务对于话术的依赖就属于弱依赖。

(4) 最弱依赖

当所依赖的服务不可用时，服务继续可用，无任何功能损失。在成本可控情况下，推荐采用最弱依赖的方式。

 京东云

京东云开发者

技术运营 @京东科技信息技术...

作者榜No.1 > 优秀作者

1.8k

文章

3.0m

阅读

19k

粉丝

关注

私信

目录

收起

- (1) 治理目标
- (2) 工具扫描

2.1) Pfinder应用拓扑图

2.2) API接口的链路跟踪环节
- (3) 人工梳理

3.1) JSF-API接口依赖梳理

3.2) UMP采集点突出依赖关系
- (4) 降级时机

总结:

相关推荐

- 流量录制回放平台探索
558阅读 · 1点赞
- 深入探秘OpenTelemetry Agent奇特的...
1.4k阅读 · 4点赞
- gRPC学习之-三种代理方式
152阅读 · 0点赞
- 10秒钟，想想如何回答好面试官口中的C...
130阅读 · 2点赞
- 揭秘研发效率提升的秘诀：日志管理的...
1.1k阅读 · 1点赞

精选内容

- 鸿蒙轻内核A核源码分析系列七 进程管...
别说我什么都不会 · 16阅读 · 0点赞
- OpenHarmony（鸿蒙南向开发）——小...
塞尔维亚大汉 · 30阅读 · 0点赞
- 边缘计算框架HonoJs的最佳实践
Y11_推特同名 · 123阅读 · 0点赞
- spring学习笔记
big男孩 · 34阅读 · 1点赞
- Shell 脚本快速入门
freejackman · 84阅读 · 0点赞

找对属于你的技术圈子
回复「进群」加入官方微信群



AI
助手

案例：商详评论，在大促高峰期如有必要是可以进行降级的。

二、依赖分类

分布式系统下的各资源依赖，按类型和层次提炼出来会有如下几种分类。

(1) 业务域依赖原则

建议上层业务域可以依赖下层业务域，整体的依赖原则受到系统依赖原则的控制，必须首先遵守应用系统之间的依赖原则，而下层业务域不允许依赖上层业务域。

核心是输出系统核心功能场景的流程图、时序图、架构图，用例图，领域模型等，需要结合业务来进行梳理。

(2) 系统启动依赖

系统启动只允许依赖数据库、应用服务器本地资源（如本地文件）、公共存储，不允许有其它基础技术服务、内部服务或外部服务依赖。消除启动依赖可以支持当发生大规模故障后的快速恢复。

案例：OPS-Review会上很多团队系统启动需要加载缓存，通过获取Redis读取数据到本地缓存，这需要注意一点在大促期间如大批量扩容，需要考虑Redis同时的容量规划

(3) 基础技术服务依赖

基础软件依赖主要包括信息中心以及数据缓存依赖，同时还应考虑系统软件及其第三方包依赖，应用系统若无特殊情况不应依赖底层操作系统或JVM特定版本。

- 缓存设计：缓存过期时间是多少？对应key范围，set入口等
- 消息依赖：系统发布了哪些消息，订阅了哪些消息，什么时机发送的，核心的消费者有哪些，消息是否需要开并行，消息下游依赖是什么？如果出现问题对自身系统和下游的核心影响是什么？
- 定时任务：有哪些定时任务，是什么业务需要？定时任务执行的时间，是否会跟双11大促高峰期冲突？

(4) 数据库依赖原则

把数据库按照数据等级进行分级，不同等级的数据库的数据保护和业务连续性保证都不一样。高优先级应用系统不能够强依赖于次优先级的数据库，以此类推各级应用系统不允许强依赖低于自己等级的数据库服务。

数据库依赖（强弱依赖、依赖权重）可能很多简单系统都只有一个数据库，数据库挂了整个系统就挂了，实际上很多重要的复杂系统都会同时具有多个数据源，将核心业务从数据源层面隔离开，哪怕有天数据库挂了，也不是业务全挂。核心是输出业务与数据库依赖关系，数据库的部署架构，如果能输出慢sql治理方案，画出数据库表ER图。

(5) 部署依赖原则

应用系统自身的网络的依赖需求：包括跨机房的网络依赖、外网访问、防火墙等。原则上日常态不建议有跨机房的服务调用网络需求(特殊情况如数据复制、容灾等除外)，实现单机房内自闭环。汇天机房调用汇天机房，廊坊调用下游的廊坊机房，宿迁调用下游的宿迁机房

(6) 对外API&MQ与访问量依赖原则

核心是根据访问模式和访问量可以推算出未来的访问量，并进行容量分析和规划。

(7) 硬件依赖原则

硬件这个方面，就交给硬件运维吧，专业的事情交给专业的人来做。

三、强弱依赖治理

(1) 治理目标

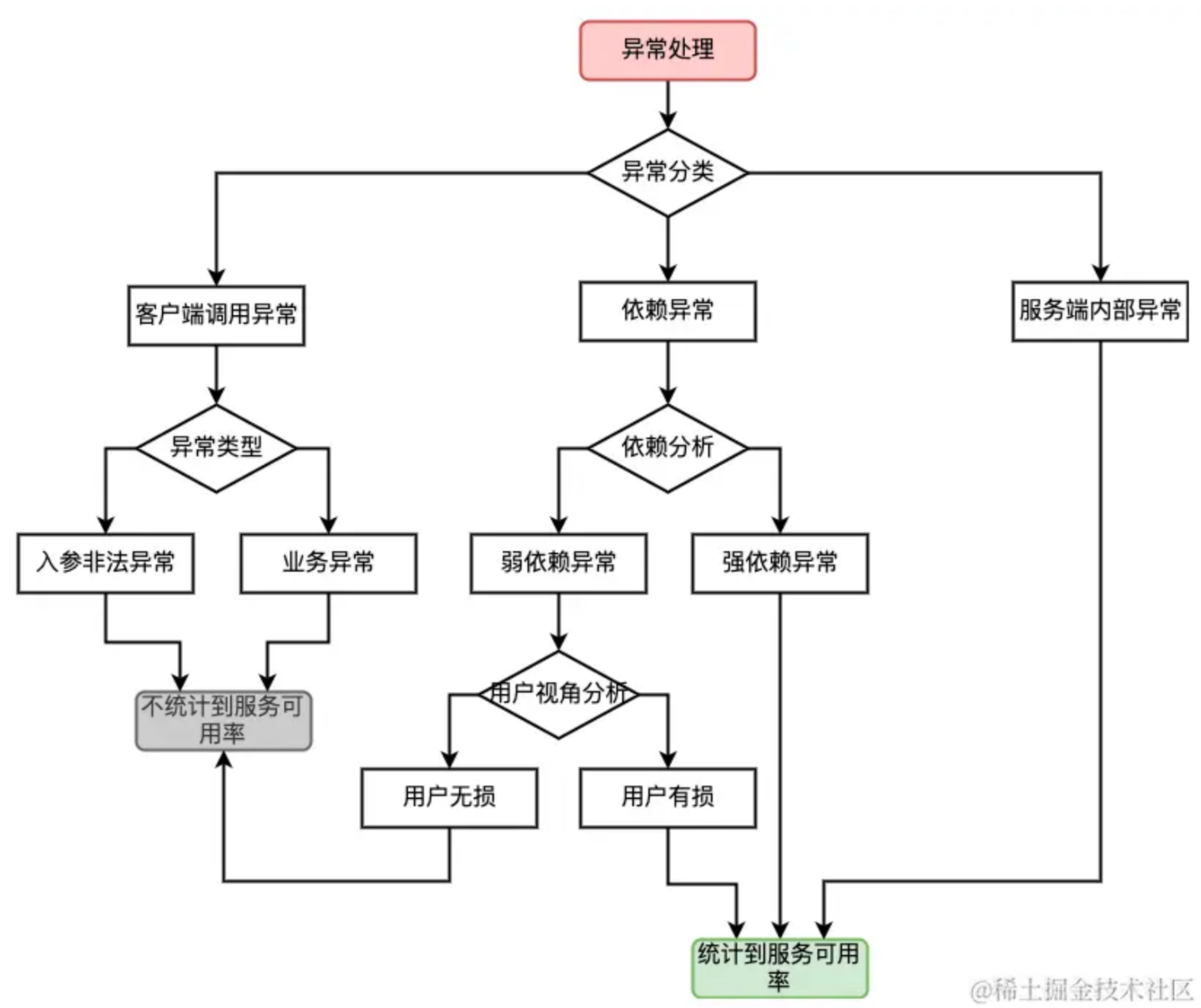
通过对核心链路内外部服务依赖治理，我们的目标是实现以下两个关键目标：

- 1.非核心业务故障不影响核心业务：通过优化服务依赖关系，确保非核心业务的故障不会对核心业务造成影响。这可以通过输出服务、应用及场景的依赖关系来实现，包括强弱依赖关系的明确划分。同时，我们会定期进行全量强弱依赖验证，以确保核心服务、应用及场景相关上下游依赖的强弱合理清晰。
- 2.提高系统的稳定性：通过弱依赖出现各类异常（包括但不限于超时、失败等）场景时的容错逻辑和应急预案，有效避免弱依赖故障对核心业务的影响。

为了达到以上目标，我们将采取以下措施：

- 1.输出应用及API场景的依赖关系：通过对系统全面的分析，我们将输出完整的服务、应用及场景的依赖关系图。这将帮助我们了解各个组件之间的关系，并确定强弱依赖关系。
- 2.弱依赖容错逻辑和应急预案：针对弱依赖出现的各类异常情况，我们将制定相应的容错逻辑和应急预案。这些预案将经过验证，以确保其能有效避免弱依赖故障对核心业务的影响。

附：依赖关系和服务可用率关系图



(2) 工具扫描

分析服务实现流程中所依赖的所有应用系统（以及这些系统提供的服务）。对一个应用系统而言，将它提供的每一个服务所依赖的应用系统汇总起来，可以构成应用依赖总体结构图。

2.1) Pfinder应用拓扑图

可看出应用的上游调用方和下游依赖方列表，可按TP99、调用量维度排序。

2.2) API接口的链路跟踪环节

通过Pfinder的调用链跟踪，可梳理对应的依赖关系以及对应的耗时统计：

(3) 人工梳理

在前期，我们通过投入相当人力，通过代码走读的形式将用车核心链路上的所有依赖及依赖强弱进行梳理。对每一个依赖，需要识别该依赖的以下属性：

依赖强弱： 强依赖是指必须的依赖，弱依赖是指可选的依赖；

同步或异步： 同步表示需要等待返回，异步指调用发生后无需等待立即返回；比如Promise发送全程跟踪MQ原先是同步发送MQ（强依赖）改成异步(弱依赖)发送方式。

依赖权重： 一次服务过程中依赖的次数，即访问的次数。

针对具体的服务类型，需要针对性地开展依赖分析，如：Redis依赖：服务实现流程中所依赖的所有缓存数据，将它提供的每一个服务所依赖的缓存数据汇总起来，可以构成该应用对Redis的依赖总体结构图。

3.1) JSF-API接口依赖梳理

案例：Promsie提供了80+JSF接口，针对这些接口进行了依赖关系的梳理，把依赖关系的UMP打点统一采集点到一个URL，并且整理为joyspace文档。 这样也是为了方便快速定位TP99毛刺高是哪个依赖，然后快速采取对应的应急预案。

3.2) UMP采集点突出依赖关系

UMP打点目前是支持打点采集点比对功能，把接口的下游打点信息全链路进行比对，可快速的定位到tp99等耗时环节，提高了日常的值班效率尤其对于大促争分夺秒来说更是关键。

通过人工梳理发现，比如Promise的获取下传时效接口核心业务链路只有依赖JIMDB配置数据时效、产能状态接口、GIS经纬度获取围栏ID、GIS详细地址获取围栏ID、到家门店时效、发送时效全程跟踪MQ。

上图Promise接口经过梳理后发现其中只有JIMDB、到家门店时效是****强依赖 。GIS获取围栏ID（可降级到四级地址时效）、全程跟踪MQ是****弱依赖

(4) 降级时机

如果弱依赖服务发生问题,则降级的触发条件可分为主动降级和被动降级；

- 主动降级：一般在大型活动时产生流量尖峰，系统无法支撑，提前对非核心的业务进行了降级处理；
- 被动降级：一般是在发生故障时自动触发预设的降级策略。

总结：

强弱依赖治理的实施需要以下几个步骤：


- 1.确定依赖关系：首先，我们需要明确应用之间的依赖关系。这可以通过分析代码、配置文件等方式来实现。只有了解了应用之间的依赖关系，我们才能进行后续的治理工作。
- 2.分析依赖数据：接下来，我们需要收集应用间的依赖关系、流量以及强弱等数据。这可以通过监控工具、日志分析等方式来实现。通过收集这些数据，我们可以更好地了解系统的运行情况，发现潜在的依赖问题，并预测可能出现的故障。这样，我们可以及时采取措施，为后续的治理工作提供依据。
- 3.制定优化方案：根据数据分析的结果，我们可以制定相应的优化方案。这可能包括调整应用间的依赖关系、优化流量分配等措施。通过实施这些优化方案，我们可以提升系统的稳定性和性能。
- 4.持续改进：强弱依赖治理是一个持续的过程。我们需要不断地收集、分析和优化数据，以推动系统稳定性的提升。同时，我们还需要及时响应用户反馈和需求变化，不断改进我们的治理策略。

总之，强弱依赖治理是一种科学的手段，可以帮助我们应对分布式微服务的复杂性。通过持续稳定地获取应用间的依赖关系、流量以及强弱等数据，我们可以提前发现潜在的故障，避免依赖故障对用户体验的影响，并积累数据持续推进系统稳定性的提升。

参考：信通院稳定性建设指南

标签： 架构

评论 0



登录 / 注册

即可发布评论!



暂无评论数据

为你推荐

电商系列：本文以商品订单为例来讲述微服务之间的依赖如何处理？

天蓝色东 | 2年前 | 981 | 2 | 评论

Java 微服务

微服务之间的数据依赖问题，你知道怎么解决吗

我爱娃哈哈 | 3年前 | 1.9k | 3 | 评论

架构 设计

限界上下文依赖: 共享内核与防腐层怎么选？

简简单单 | 5月前 | 111 | 2 | 评论

领域驱... 后端 架构

微服务的设计模式

终码一生 | 2年前 | 331 | 3 | 评论

后端

手把手教你实现分布式服务的调用链跟踪实现

果果果果果 | 3年前 | 757 | 4 | 评论

Java 后端

7. Molinillo 依赖校验

土土Edmond木 | 3年前 | 1.3k | 9 | 1

Cocoa... 前端

SpringCloud之Hystrix服务降级

以范特西之名 | 1年前 | 884 | 2 | 评论

后端 Spring Cl...

微服务架构中，二次浅封装实践

七号楼 | 3年前 | 821 | 1 | 评论

微服务 分布式

通过降级手段保障业务稳定性的思考与总结

码农丁丁 | 7月前 | 260 | 点赞 | 评论

代码规范

探讨篇（一）：服务粒度的艺术 - 简化架构与避免服务泛滥

京东云开发者 | 9月前 | 399 | 2 | 评论

后端

微服务架构下，如何通过弱依赖原则保障系统高可用？

京东零售技术 | 10月前 | 874 | 4 | 评论

后端 架构

微服务架构下如何通过弱依赖原则保障系统高可用

京东云开发者 | 10月前 | 910 | 3 | 评论

后端

业务压力一大就宕机？一文带你搞懂限流熔断！

煎鱼eddcyj | 4年前 | 706 | 4 | 评论

Go

Enhancer-轻量化的字节码增强组件包

得物技术 | 1年前 | 8.6k | 21 | 1

Java 架构

服务架构的设计原则

王翊珩 | 8月前 | 978 | 3 | 评论

后端