



下载APP

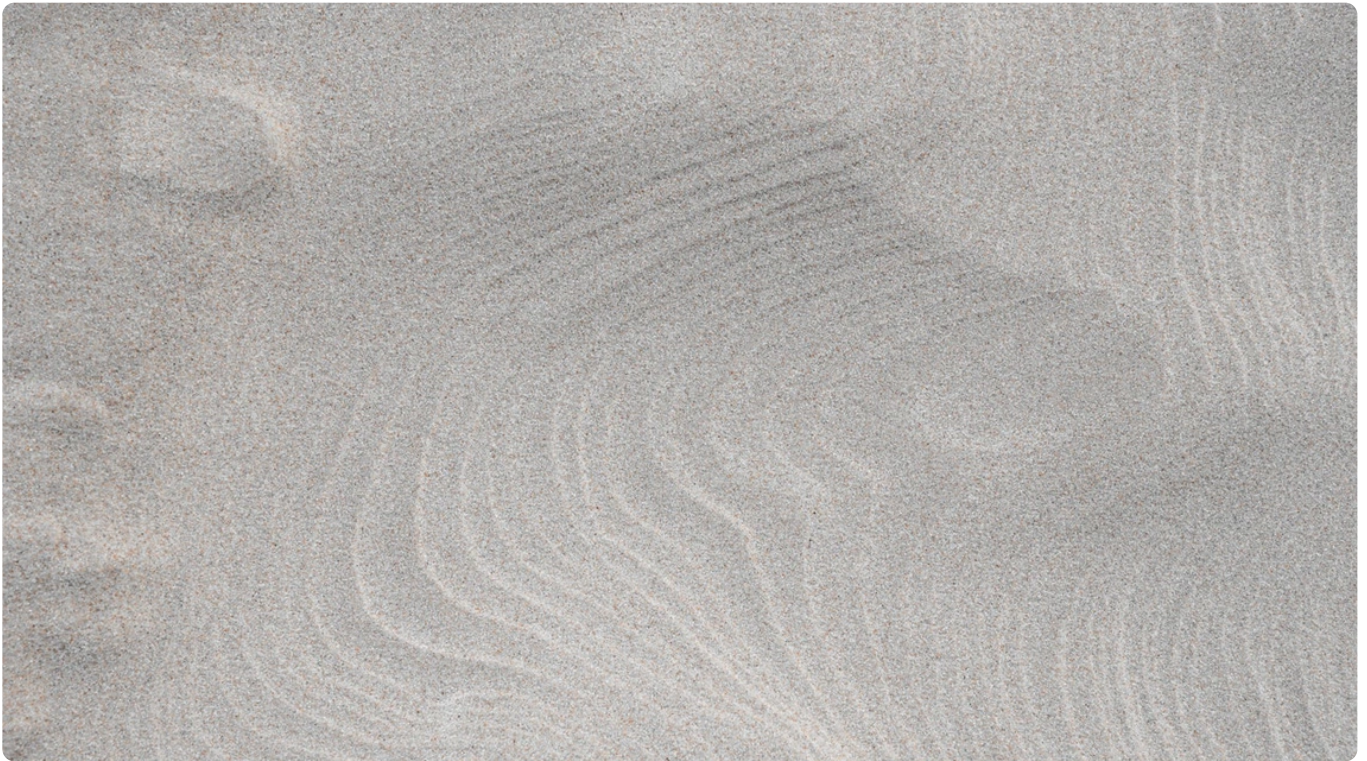


28 | 消息驱动：谁说消息队列只能削峰填谷？

2022-02-16 姚秋辰

《Spring Cloud 微服务项目实战》

课程介绍 >



讲述：姚秋辰

时长 12:53 大小 11.81M



你好，我是姚秋辰。

说起消息驱动，它可是一个有点年头的老技术了，如今借着微服务架构这股春风可谓是混得春风得意。但凡你去大厂面试，被问到三高系统架构的问题，高低得整两句：消息驱动是如何“削峰填谷”来解决高并发的。

领资料

要知道，消息驱动可不仅仅停留在面试环节，它的用武之地也不只局限于“削峰填谷”。今天我就带你了解一下，消息驱动技术在微服务系统中有哪些常用场景。这节课我会基于过去开发过的实际项目，来一一列举各种应用场景，加深你的学习体感。



前面提到了削峰填谷，但我还偏就不从这老掉牙的话题开场，这种“面试宝典”里的经典问题，就是在校生成都能答出个一二三来。我这里要为你介绍的第一个消息驱动场景，就是

和微服务架构最贴合的“服务间解耦”。

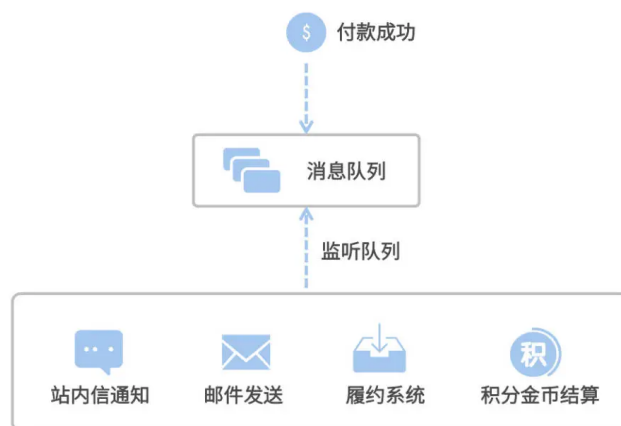
服务间解耦

如果你认为把服务拆分成微服务就叫做服务间解耦，那咱对微服务的认知还停留在第一层。在很多场景中，我们还需要借助消息驱动组件对业务场景做进一步解耦。

举个例子，在我们网购下单完成付款之后，有一系列的后续业务流程会被执行。比如买家短信和邮件通知、IM 和站内信推送、金币和积分结算、卖家端履约流程等等。有时候搞线上活动，还会在付款完成之后触发赠券服务。

我们假设这些业务场景都被拆分成了微服务，从业务完整性的角度来讲，为了实现付款后自动触发完整链路，交易服务的回调接口必须挨个调用我前面提到的各个业务系统。如果未来需要接入新的业务场景，你还得往回调接口里加上一个新的系统集成点。再从从服务容错的角度考虑，你还得兼顾关键场景（如金币 + 积分结算）的失败重试，这些逻辑都掺和到了支付成功的回调接口里。

所以，即便我们的业务系统是微服务架构，上下游之间的调用还是跑不掉，这种代码中的调用关系也是一种“耦合”。在真实业务中这类场景很普遍，它的特点是通过某个事件（如支付成功）触发多个下游业务场景，像这类场景就特别适合使用消息驱动技术做解耦。



比如，我可以将付款成功的信息连同当前订单信息放入一个消息队列中，让所有的下游服务监听这个队列，通过这种“**断直连**”的方式，我们就将上下游服务之间的耦合间接地解

除了。不管以后下游服务要添加什么新场景，对上游服务都几乎是无感知的，因为新的业务场景只要对接消息队列就好，并不需要对上游服务发起调用。

说完了服务解耦，我们接下来再聊一聊消息广播，它也是消息驱动的一个重要应用。

消息广播

消息广播是相对于单播来讲的。单播是指在同一个消费组里，最多只有一个消费者实例可以去消费消息，而广播则是说，一个消费组里所有的消费者都会对消息做一次消费。

消息广播的一个常用场景是热点数据的处理，啥是热点？比如我某一天被明星出轨了，那我的微博就会成为一个“热点数据”，咱前面在 Sentinel 的课程里了解过，热点数据是高可用破防能手。各个大厂都有自己的热点侦测方案，这个不展开说了，就说一旦某个资源被甄别成热点数据之后，是不是要通知各个服务“小心防范”？碰到热点资源的访问请求，直接打到专门的热点集群上做处理。

那么这里的“通知”动作，就特别适合使用消息广播的方式来处理，我们只要在侦测到热点数据之后，发送一个消息到特定的消息队列，让各个有可能接收到热点请求的应用服务接入这个队列，执行相应的热点逻辑。

还有一个和热点数据相类似的场景：本地缓存构建。你一定知道通过 Redis 和 Tair 这类缓存系统来抗 QPS，但对于一些访问频次比较高的资源，我们会倾向于在 Client 本地构建一个“本地缓存”，一来堆内缓存一定是访问速度最快的缓存（绝对比外部缓存快），二来可以降低外部缓存的 QPS，毕竟缓存也是能被压崩盘的。

和热点数据同理，这个例子中的“本地缓存”也是可以通过消息广播来构建的。比如在网关或者 RPC 链路上，我通过一些流技术对实时调用情况进行聚合分析，将访问频次比较高的资源标记为临时热点，并通过消息驱动推送到各个消费者节点。这样，我们就借助消息广播场景实现了资源标记的推送。

那接下来就让我们去了解第三个场景，延迟类业务吧。

延迟业务

你可以把这类业务理解为一个闹钟，它是在未来某个时间会被执行的业务逻辑。最常见的一类延迟业务就集中在网购中的订单模块，我这里举两个例子。

订单确认：下了单付了款收了货，就是不点确认收货，没关系，7 天之后系统会自动确认。

取消订单：下单之后在 30 分钟时间内没有付款，自动取消订单。

上面这两个场景都可以借助延迟消息来实现，不过在具体实现的时候，你还需要借助消息分区等功能降低消息的积压量。

我还实现过一些延迟类业务：批量改价单和批量库存发布。改价单的业务需求是给商品设置一个新的价格，指定特定时间生效。批量库存的需求是设置一个补货时间，在指定的时间点修改单品 SKU 库存数量。这些场景我也是构建在延迟消息之上搭建的。

削峰填谷

最后就到了老生常谈的削峰填谷场景了，这个场景我们可以拆分为削峰和填谷这两段来看。

削峰就是指削减峰值流量，如果某个业务的峰值流量超过了系统吞吐量，并且这类业务又非常重要，不能简单粗暴地通过限流熔断把请求 cut 掉，那么你可以考虑把这些请求压入消息队列，让消费者根据自身的吞吐量从队列中获取消息并消费。

填谷就是指闲的没事儿干的时候让你忙起来，当业务峰值已经过去了，流量逐渐减少的时候，先前积压在消息队列中的请求就能被逐渐消化。

削峰填谷其实是一种平滑利用资源的手段，之所以我们能将大量消息压入消息队列，是因为目前主流的消息队列都有非常强大的消息堆积能力。当然了，MQ 组件的消息积压量也是有极限的，在真实的线上业务中，我们会为消息队列构建完善的监控指标，提前对消息积压进行预警。

削峰填谷这个用法适合用在一些实时性要求不高，但并发量比较高的业务中。我举一个自己曾参与搭建的电商业务场景，帮你加深对削峰填谷的理解，这个例子就是商品批量发布。

在新零售业务中，我们提供了一种“一键开店”的业务模式，即通过一个简单流程，一键将数十万 SKU 发布到新的门店中。商品发布是一个非常复杂的流程，它需要将商品元数据注册到商品中心、发布商品主副图、详情页 SKU、各类营销优惠信息的发布等等。尽管这是一个低频场景，但奈何一次发布的商品基数非常大，很容易形成一个流量洪峰冲击。我的做法就是借助淘系 MetaQ（RocketMQ 的前身），将商品发布请求压到 MQ 里，由下游集群不紧不慢地去消费。

到这里，我们对消息队列的几个常见场景都有了一定的了解。下面就让我来带你回顾下本节重点吧。

总结

在今天的课程里，我们先后介绍了服务间解耦、消息广播、延迟消息和削峰填谷这几个常见的消息队列场景。在纷繁复杂的电商业务中，消息组件还有各式各样的花式玩法，比如廉价好用的一致性保证措施“事务性消息”、处理顽固异常的“死信队列”、用来做消息路由的“一致性哈希 Routing”等等。随着工作经验的积累，你一定会接触到越来越多的消息组件花式玩法。

无论是 Kafka、RocketMQ 还是 RabbitMQ，每个消息组件都有自己拿手的领域和特性，而且有些中间件还提供了丰富的插件库，用来提供一些“外挂”功能。了解每个消息中间件的特点，可以帮助你在业务场景中做出更好的技术选型判断。

我这里想跟你推荐一个运行成本低又十分灵活的消息组件 Pulsar，它可以同时支持流和队列两种语义，在扩展性和可靠性方面也相当优秀。我在这里立一个 Flag，隐约之间 Pulsar 有成为下一代消息驱动王者的王霸之气，让我们拭目以待。

思考题

你能根据自己的项目，分享一些消息组件的实际应用场景吗？

好啦，这节课就结束啦。欢迎你把这节课分享给更多对 Spring Cloud 感兴趣的朋友。我是姚秋辰，我们下节课再见！

 生成海报并分享 赞 0  提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 27 | 微服务网关：如何借助 Nacos 实现动态路由规则？

下一篇 29 | 消息驱动：如何集成 Stream 实现消息驱动？

精选留言 (4)

 写留言

请叫我和尚

2022-02-16

分享一下消息队列在我们团队其中的一些应用：

延迟业务：

背景，记录用户的行为，当用户的行为 x s/m/h，未进行下一步行为的时候，我们就需要执行相关策略。

比如配置策略：当用户进入商品界面 30m 之后未点击购买，就给用户发送这个商品购买...

展开

共 1 条评论

 4



2022-02-16

电商OMS系统中，订单下载后处理过程涉及的业务比较多，如匹配仓库、匹配快递、扣减库存、打标签、匹配赠品规则、拆单、合单等等，如果从订单下载到处理整个流程同步进行将大大降低订单处理效率，下游的订单处理服务也需要更多的资源。

订单下载后入系统订单表，发送消息进行转单处理，再发送消息进行系统订单处理，大促期间只需要适当增加订单下载服务就可以扛得住大流量的冲击。

展开



 2



快乐的小土豆

2022-02-16

想问下一般都用消息队列做异步通知，而RPC框架同步获取结果吗

作者回复：有些rpc框架也支持异步调用，MQ的场景是“可靠消息投递”，用mq送达率比rpc要高

**peter**

2022-02-16

请教老师几个问题啊：

Q1 消息队列怎么实现延迟任务？

消息队列只能收发消息，不能执行任务啊。是消息队列定时发消息给具体业务来实现的吗？

Q2 用实时流技术来识别热点数据，一般是怎么做的？用大数据组件吗？比如spark。 ...

展开 ∨

作者回复: Q1：延迟消息后面会讲到

Q2：stream流分析rpc或网关日志，各个大厂有不同做法

Q3：消息分区相当于分库分表同理

Q4：有的消息组件有插件可以支持，有的cloud版都有云服务商提供了配置参数做预警

Q5：零售行业SKU数量多，线下卖场

