**消息队列背后的设计思想**

<https://mp.weixin.qq.com/s/k8sA6XPrp80JiNbuwKaVfg>

消息队列也通常称为消息中间件，提到消息队列，大部分互联网人或多或少都听过该名词。对于后端工程师而言，更是日常开发中必备的一项技能。随着大数据时代的到来，apache 旗下的 kafka 一度成为消息队列的代名词，提起消息队列大家自然而然就想到了 kafka。近而网上有太多太多介绍消息队列 kafka 功能或者内部实现的文章。

然而消息队列本身是工程领域内一种解决问题的通用方案。它的背后有着一些通用的设计思想和经典模型，这些是消息队列的精髓和灵魂。它们独立于任何一种消息队列的具体实现(例如 kafka)，但每种消息队列(除了 kafka 外，还有 rocketMQ、pulsar 等)的实现中到处体现着这些设计思想。本文主要从抽象层面来简单谈谈消息队列背后的一些设计思想，辅助理解消息队列这一类组件。

本文主要解决三个问题：

1. 消息队列适合什么场景？
2. 消息队列有哪些主流产品、各自的优缺点？
3. 消息队列背后的设计思想(整体核心模型、数据存储考量、数据获取方案对比、消费者消费模型)

**1.消息队列适合哪些场景？**

**消息队列：它主要用来暂存生产者生产的消息，供后续其他消费者来消费。它的功能主要有两个：a.暂存(存储)、b.队列(有序：先进先出)。其他大部分场景对数据的消费没有顺序要求，主要用它的暂存能力** 。从目前互联网应用中使用消息队列的场景来看，主要有以下三个：  
**1. 异步处理数据**  
**2. 系统应用解耦**  
**3. 业务流量削峰**

