20. 分布式环境下流控技术汇总_V20240122

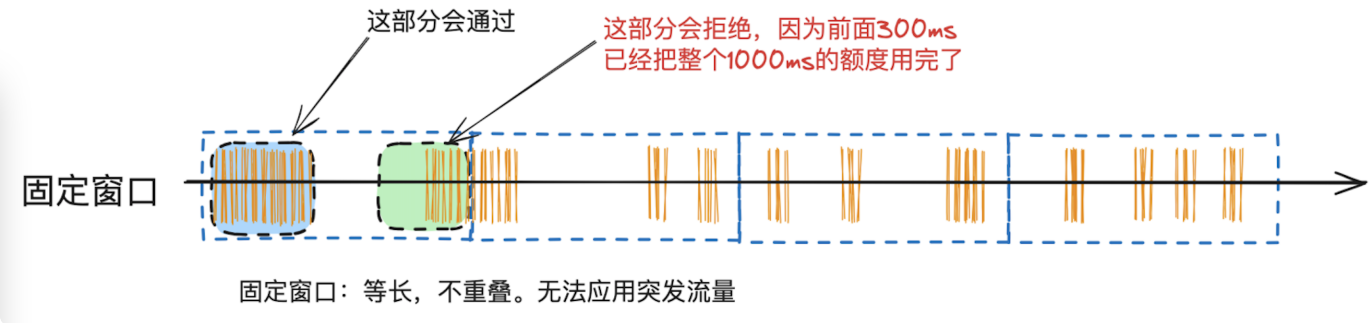
本篇主要是对分布式环境流控技术及使用场景做个简要的汇总，包括：固定时间窗口算法，滑动时间窗口算法，漏桶算法，令牌桶算法，分布式消息中间件，流控与熔断利器Sentinel。

1. 前言

在流量控制系列文章中的前六篇，分别介绍了固定时间窗口算法、滑动时间窗口算法、漏桶原理、令牌桶、消息中间件、Sentinel如何应用到分布式环境下的流量与并发控制。

这里再次对这几个做一个简单回顾，知道工具箱里面的不同工具的特性，才能更好更快地干活。

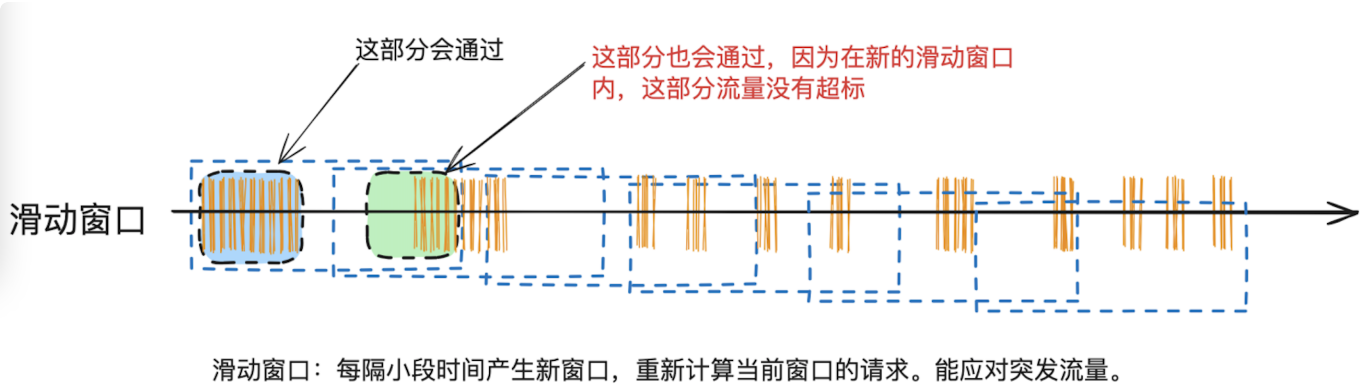
2. 固定时间窗口算法



固定窗口：算法简单，对突然流量响应不够灵活。超过流量的会直接拒绝，通常用于限流。

详见：《精确掌控并发：固定时间窗口算法在分布式环境下并发流量控制的设计与实现》

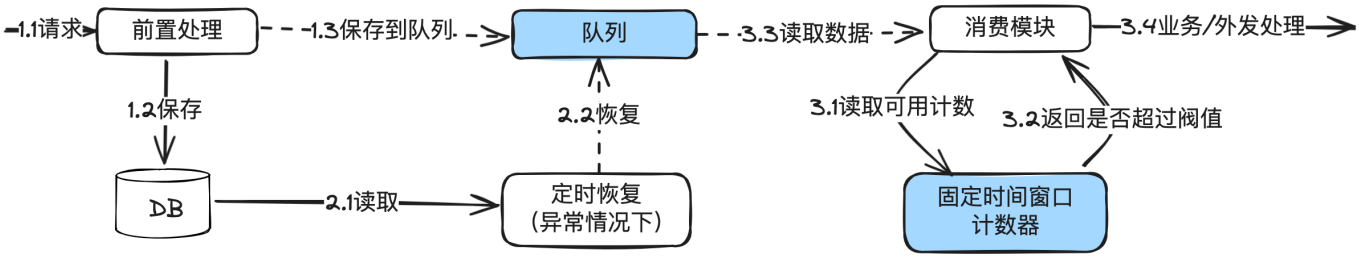
3. 滑动时间窗口算法



滑动窗口： 算法简单，对突然流量响应比固定窗口灵活。超过流量的会直接拒绝，通常用于限流。

详见：《精确掌控并发：滑动时间窗口算法在分布式环境下并发流量控制的设计与实现》

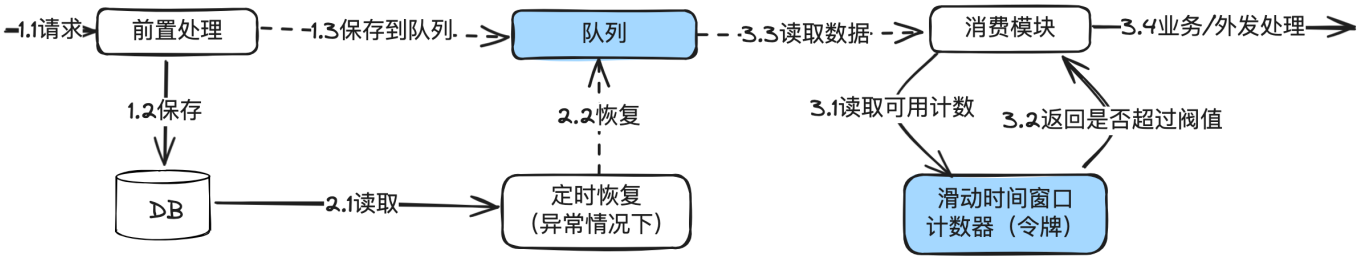
4. 漏桶算法



漏桶算法：在固定窗口的基础之上，使用队列缓冲流量。提供了稳定的流量输出，适用于对流量平滑性有严格要求的场景。

详见：《精确掌控并发：漏桶算法在分布式环境下并发流量控制的设计与实现》

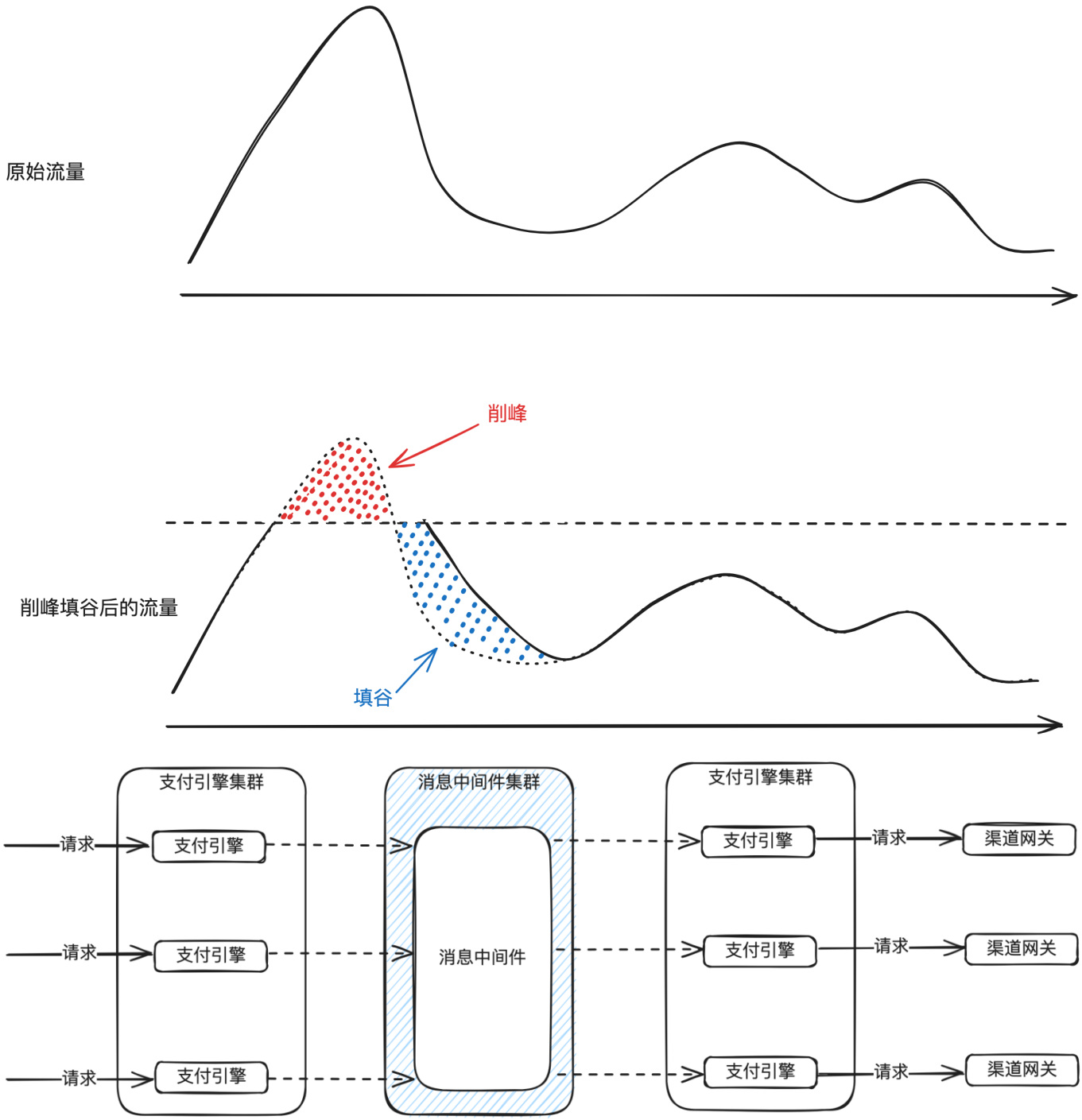
5. 令牌桶算法



令牌桶算法：在滑动窗口的基础之上，使用队列缓冲流量。提供了稳定的流量输出，且能应对突发流量。

详见：《精确掌控并发：令牌桶算法在分布式环境下并发流量控制的设计与实现》

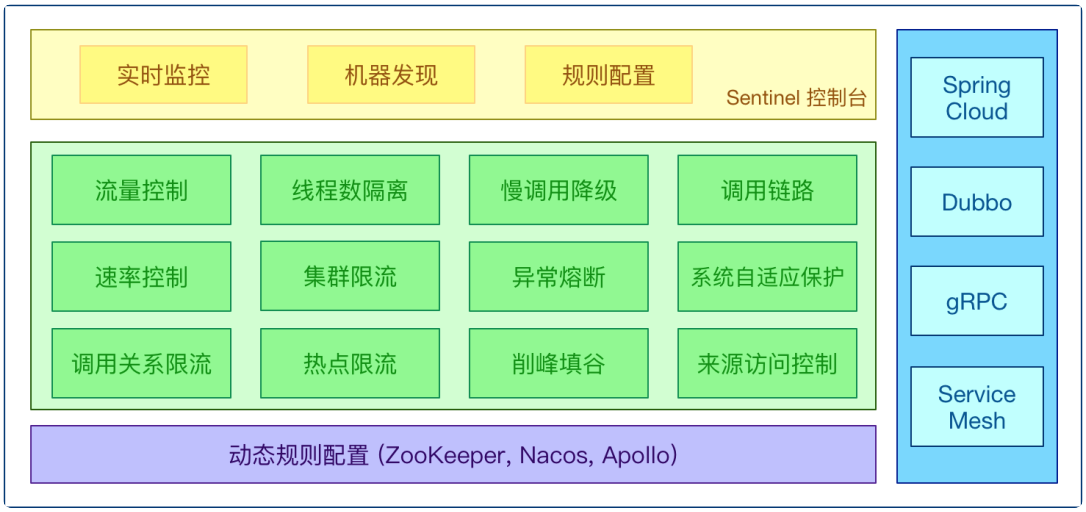
6. 分布式消息中间件



分布式消息中间件：在支付场景的削峰填谷用得比较多，且对精度没有那么苛刻的场景。以及应用间的解耦。

详见：《削峰填谷与应用间解耦：分布式消息中间件在分布式环境下并发流量控制的应用》

7. 流控与熔断利器Sentinel



Sentinel：分布式场景下的流量控制和熔断机制利器。

详见：《流量控制与熔断利器：Sentinel介绍》

8. 方案选型

限流和熔断保护：Sentinel。

削峰填谷和应用间解耦：消息中间件。

极低并发要求：自己使用redis实现漏桶或令牌桶。

想手撸一段代码测试：固定时间窗口和滑动时间窗口。

实际上，自己实现的固定时间窗口或滑动时间窗口，还可以加上一些其它技术，解决一些其它的问题，比如渠道自动开关。这个后面单独开文章介绍。

9. 结束语

前面六篇文章对流控的原理、实现方案、应用场景分别做了详细的描述，应对绝大部分的支付系统，是绰绰有余的。哪怕中国TOP2的支付公司，内部的使用也差不多是这样，只是部署集群的规模更大，对稳定性的要求更高，对应地附加了很多其它保障手段。

这是《百图解码支付系统设计与实现》专栏系列文章中的第（20）篇。和墨哥（隐墨星辰）一起深入解码支付系统的方方面面。

欢迎转载。

Github（PDF文档全集，不定时更新）：https://github.com/yinmo-sc/Decoding-Payment-System-Book

公众号：隐墨星辰。

有个小群不定时解答一些问题或知识点，有兴趣的同学可先加微信（yinmo_sc）后进入，添加微信请备注：加支付系统设计与实现讨论群。