背景

每一个对外提供的API接口都是需要做流量控制的，不然会导致系统直接崩溃。很简单的例子，和保险丝的原理一样，如果用电符合超载就会烧断保险丝断掉电源以达到保护的作用。API限流的意义也是如此，如果API上的流量请求超过核定的数值我们就得对请求进行引流或者直接拒绝等操作。

1、限流算法

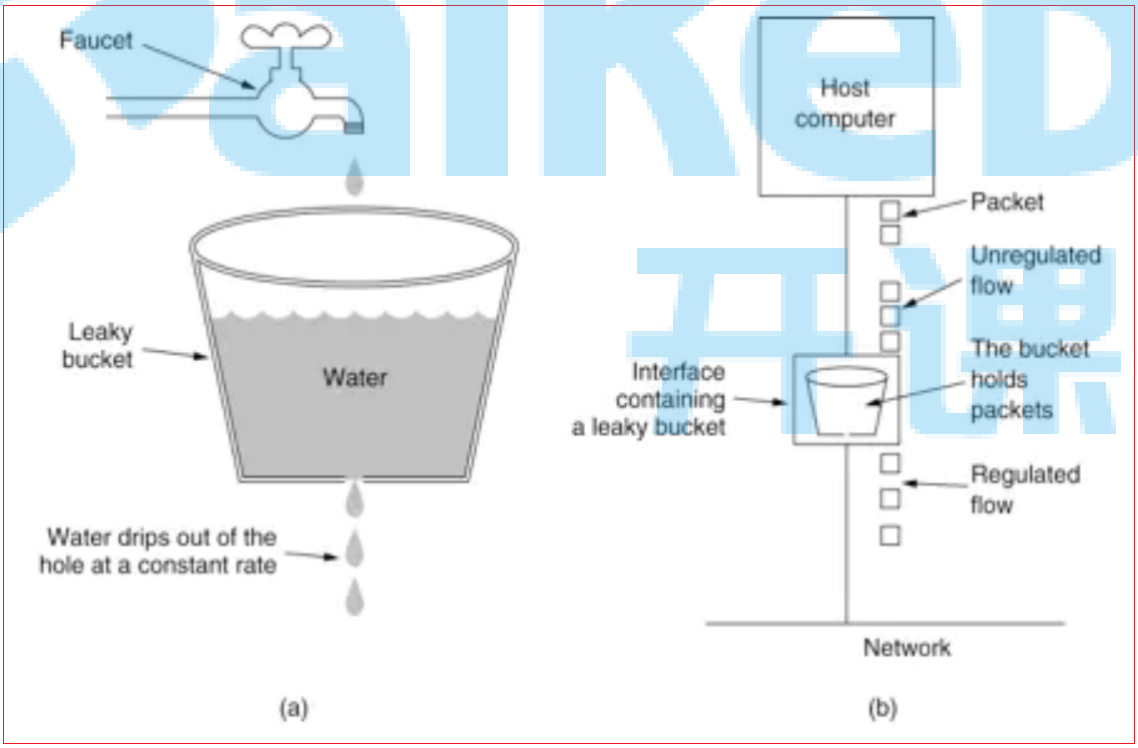
既然要限流，就得提到限流算法了，一般有漏桶算法和令牌桶算法两种限流算法。

1.1 漏桶算法

漏桶算法(Leaky Bucket)是网络世界中流量整形（Traffic Shaping）或速率限制（Rate Limiting）时经常使用的一种算法，它的主要目的是控制数据注入到网络的速率，平滑网络上的突发流量。漏桶算法提供了一种机制，通过它，突发流量可以被整形以便为网络提供一个稳定的流量。

漏桶可以看作是一个带有常量服务时间的单服务器队列，如果漏桶（包缓存）溢出，那么数据包会被丢弃。 在网络中，漏桶算法可以控制端口的流量输出速率，平滑网络上的突发流量，实现流量整形，从而为网络提供一个稳定的流量。

如图所示，把请求比作是水，水来了都先放进桶里，并以限定的速度出水，当水来得过猛而出水不够快时就会导致水直接溢出，即拒绝服务。

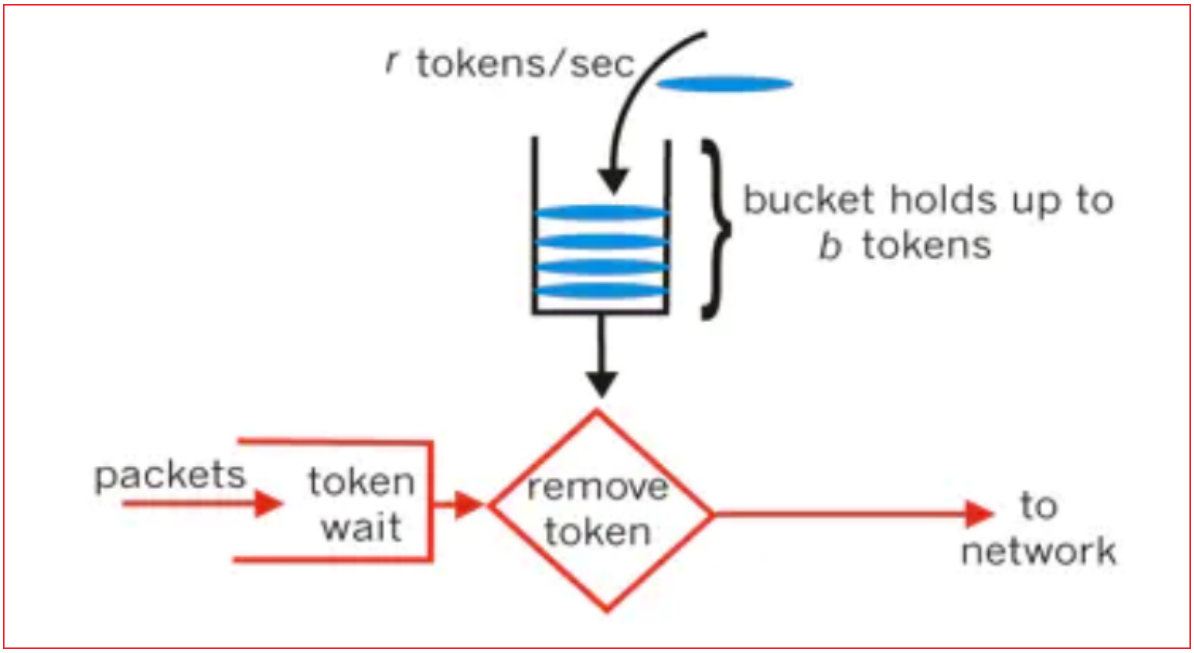


可以看出，漏桶算法可以很好的控制流量的访问速度，一旦超过该速度就拒绝服务。

1.2 令牌桶算法

令牌桶算法是网络流量整形（Traffic Shaping）和速率限制（Rate Limiting）中最常使用的一种算法。典型情况下，令牌桶算法用来控制发送到网络上的数据的数目，并允许突发数据的发送。

令牌桶算法的原理是系统会以一个恒定的速度往桶里放入令牌，而如果请求需要被处理，则需要先从桶里获取一个令牌，当桶里没有令牌可取时，则拒绝服务。从原理上看，令牌桶算法和漏桶算法是相反的，一个“进水”，一个是“漏水”。



技术上使用Google开源工具包Guava提供了限流工具类RateLimiter，该类基于令牌桶算法来完成限流，非常易于使用

2、漏桶算法和令牌桶算法的选择

漏桶算法与令牌桶算法在表面看起来类似，很容易将两者混淆。但事实上，这两者具有截然不同的特性，且为不同的目的而使用。

漏桶算法与令牌桶算法的区别在于：

漏桶算法：

水（请求）先进入到漏桶里，漏桶以一定的速度出水，当水流入速度过大会直接溢出（拒绝服务），可以看出漏桶算法能强行限制数据的传输速率

- **流入：以任意速率往桶中放入水滴。**
- **流出：以固定速率从桶中流出水滴。**

缺点：因为当流出速度固定，大规模持续突发量，无法多余处理，浪费网络带宽

优点：无法击垮服务，可以处理突发流量

令牌桶算法：

令牌桶算法：请求获取到token令牌之后采用可以访问具体服务。

令牌桶分为2个动作，动作1(固定速率往桶中存入令牌)、动作2(客户端如果想访问请求，先从桶中获取token)

- **流入：以固定速率从桶中流入水滴**
- **流出：按照任意速率从桶中流出水滴**

优点：支持大的并发，有效利用网络带宽

缺点：没有很强的突发流量处理能力