链滴

限流算法：令牌桶与漏桶

作者：DongXiaokai0819

原文链接：https://ld246.com/article/1582879775824

来源网站：链滴

限流算法

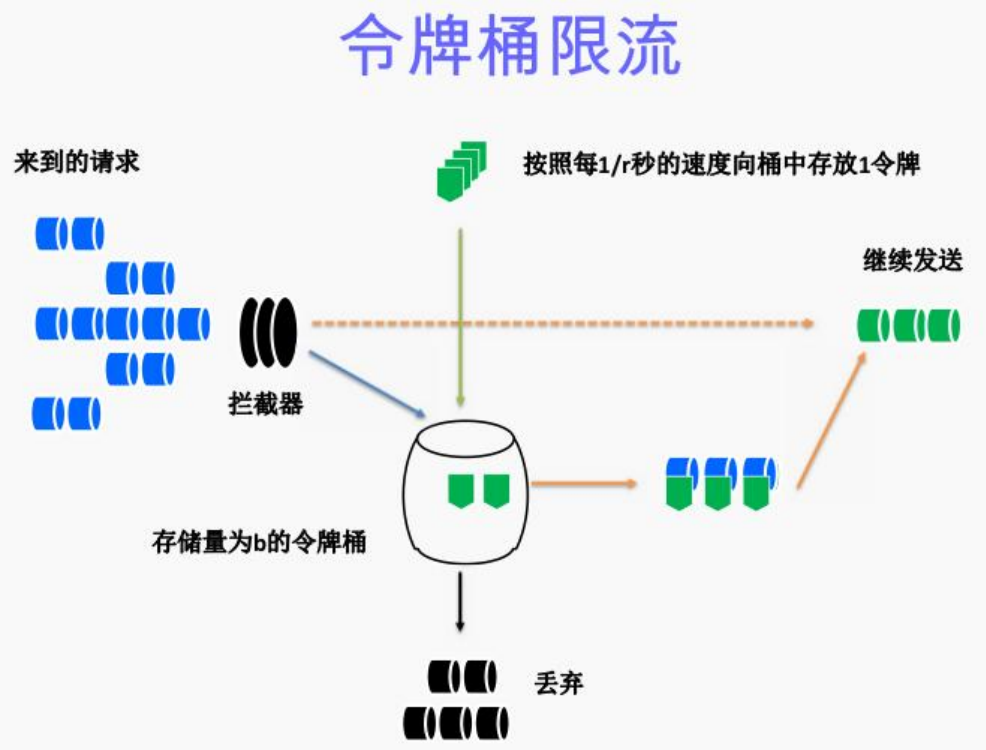
    高并发系统中有三把
器用来保护系统：**缓存**、**降级**和**限流**。限流的目的是为了保护系统不被大量的请求冲垮，通过限制请的速度来保护系统。在电商的秒杀活动中，限流是必不可少的一个环节。

    限流的方式有很多种
可以在**Nginx层面**限流，也可以在应用当中限流，比如在**API网关**中。

令牌桶

令牌桶算法是一个存放固定容量令牌的桶，按照固定速率往桶里添加令牌。可以控制流量也可以控制发量，假如我们想要控制API网关的并发量最高为1000，可以创建一个令牌桶，以固定的速度往桶里加令牌，超出1000则不添加。

当一个请求到达之后就从桶里获取一个令牌，如果能获取到令牌就可以继续往下请求，获取不到就说令牌不够，并发量达到了最高，请求就被拦截。



算法描述：

1. 假如用户配置的平均发送速率为r，则每隔1/r秒一个令牌被加入到桶中（每秒会有r个令牌放入桶中；

2. 假设桶中最多可以存放b个令牌。如果令牌到达时令牌桶已经满了，那么这个令牌会被丢弃；

3. 当一个n个字节的数据包到达时，就从令牌桶中删除n个令牌（不同大小的数据包，消耗的令牌数量一样），并且数据包被发送到网络；

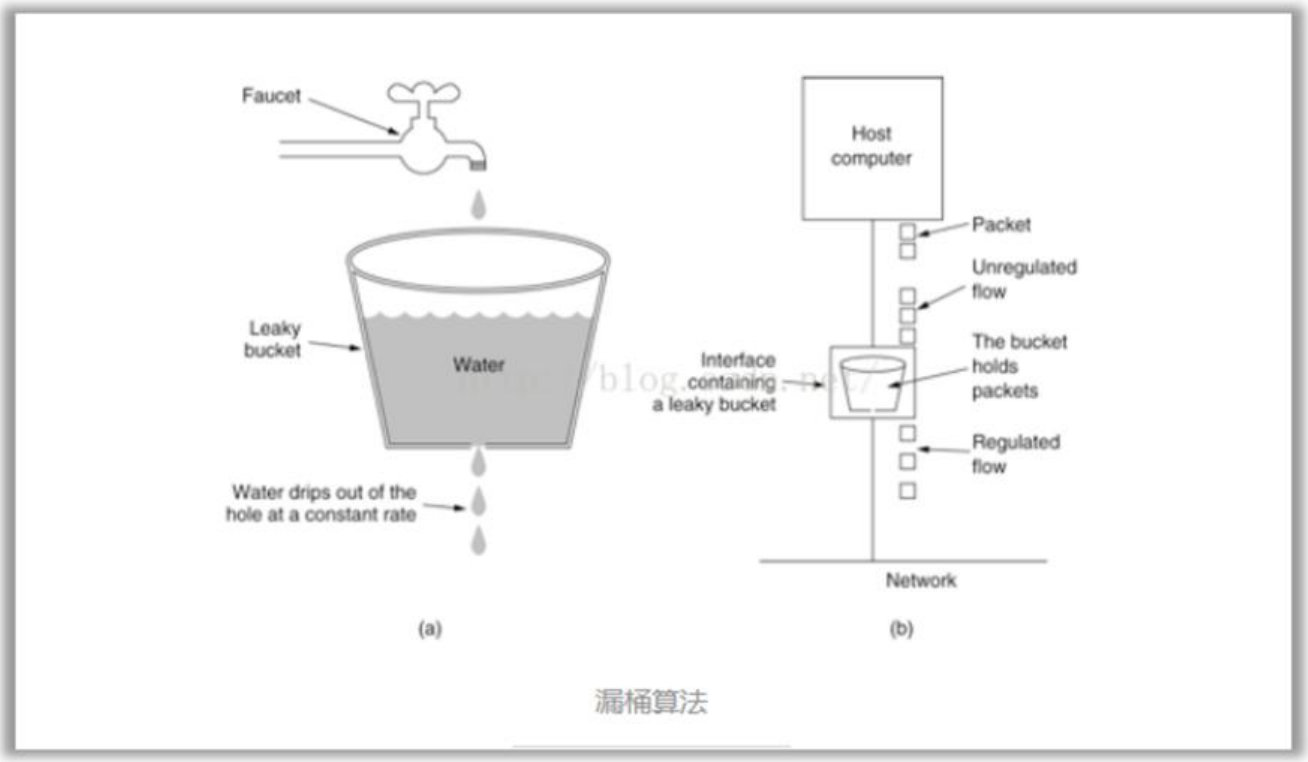
4. 如果令牌桶中少于n个令牌，那么不会删除令牌，并且认为这个数据包在流量限制之外（n个字节，要n个令牌。该数据包将被缓存或丢弃）；

5. 算法允许最长b个字节的突发，但从长期运行结果看，数据包的速率被限制成常量r。对于在流量限外的数据包可以以不同的方式处理：（1）它们可以被丢弃；（2）它们可以排放在队列中以便当令牌

中累积了足够多的令牌时再传输；（3）它们可以继续发送，但需要做特殊标记，网络过载的时候将些特殊标记的包丢弃。

漏桶

漏桶是一个固定容量的桶，按照固定的速率流出，可以以任意的速率流入到漏桶中，超出了漏桶的容就被丢弃，总容量是不变的。但是输出的速率是固定的，无论你上面的水流入的多块，下面的出口只这么大，就像水呗开闸放水一样。



漏桶的伪代码

```
long timeStamp = getNowTime();
int capacity = 10000;// 桶的容量，即最大承载值
int rate = 1;//水漏出的速度，即服务器的处理请求的能力
int water = 100;//当前水量，即当前的即时请求压力

//当前请求线程进入漏桶方法，true则不被拒绝，false则说明当前服务器负载水量不足，则被拒绝
public static bool control() {
long  now = getNowTime();//当前请求时间
//先执行漏水代码
//rate是固定的代表服务器的处理能力，所以可以认为“时间间隔*rate”即为漏出的水量
    water = Math.max(0, water - (now - timeStamp) * rate);//请求时间-上次请求时间=时间间隔
    timeStamp = now;//更新时间，为下次请求计算间隔做准备
    if (water < capacity) { // 执行漏水代码后，发现漏桶未满，则可以继续加水，即没有到服务器可承担的上线
        water ++;
        return true;
    } else {
        return false;//水满，拒绝加水，到服务器可以承担的上线，拒绝请求
   }
```

原文链接：限流算法：令牌桶与漏桶

```
}
```

令牌桶与漏桶的区别

这两种算法的主要区别在于**漏桶算法能够强行限制数据的传输速率**，而**令牌桶算法在能够限制数据的均传输速率外，还允许某种程度的突发传输。**在令牌桶算法中，只要令牌桶中存在令牌，那么就允许发地传输数据直到达到用户配置的门限，因此它适合于具有突发特性的流量。

参考：https://blog.csdn.net/SunnyYoona/article/details/51228456