

2020/2/28 - 10:00 - 1小时25分钟

题目1 - 接口限流器

2020/02/28 10:00 - 11:00

代码面试要求

1. PC钉钉共享屏幕
2. 拿到题目后先仔细理解题目，有不懂的可以提问
3. (预计15分钟) 完整理解设计后给面试官讲解下预备的思路
4. (预计40分钟) 完成题目，用自己熟悉的方式和语言在自己电脑上完成
5. (预计5分钟) 总结

接口限流实现

有一个API网关，出于对API接口的保护，需要建立一个流控功能，根据API名称，每分钟最多只能请求指定的次数（如1000次），超过限制则这分钟内返回错误，但下一分钟又可以正常请求。

环境：自己电脑(共享屏幕)，语言：C++，编译器：g++，IDE：Vscode，输入输出：无

解题思路：

- 数据结构：用 `map`，`key` 为请求API的URL，`value` 为一个结构体，包含：请求次数、“第一次”请求时间 两个属性。
- 算法：每请求一次，就用当前时间与对应 `value` 的时间属性比较，如果相差小于60秒，且请求次数小于1000，则请求次数属性加一；反之，请求次数属性设为1，时间属性改为当前时间。

题目2 - 接口限流器+滑动窗口

题目1是固定窗口，现改为滑动窗口，每过一秒，窗口滑动一次。即，0~59、1~60 内的请求次数不能大于1000。

刚开始可能没听清，不太理解题目。然后他举了几个例子，我才明白，这个每过一秒滑动一次，是指过了60秒之后，每过一秒滑动一次。比如，59秒的时候，只算0~59 内的请求次数；60秒的时候，只算1~60 内的请求次数。

解题思路：

- 数据结构：`map` 的 `value` 改为滑动窗口结构体。滑动窗口结构体中包含两个属性：当前窗口的总请求次数 和 一个队列。队列的基本单元也为一个结构体，与上题一样，包含：请求次数、“第一次”请求时间 两个属性。
- 算法：每请求 `url` 一次，就与队列的最后一个元素比较，如果当前时间与之差值小于1秒，则最后一个元素的请求次数加一、窗口总次数加一；反之，则从头开始遍历队列，如果队列的头元素与当前时间相差大于60秒，则 `pop()` 该头元素，同时总次数减去头元素的次数，直到头元素与当前时间相差小于60秒。

此题一开始用链表，后来想想不需要中间插入元素，就改成队列。结果，使用 `queue` 的时候，却爆出了一个bug，死活调不好。。。我看时间不够了，就跟面试官讲了一下思路，就结束了。。。

总结

阿里的题目很偏向于应用场景，曾遇见过就简单，第一次就感觉挺难的。

事后，谷歌一下，才发现有专门的限流器算法，是我孤陋寡闻了:(