Rate Limit Algorithms

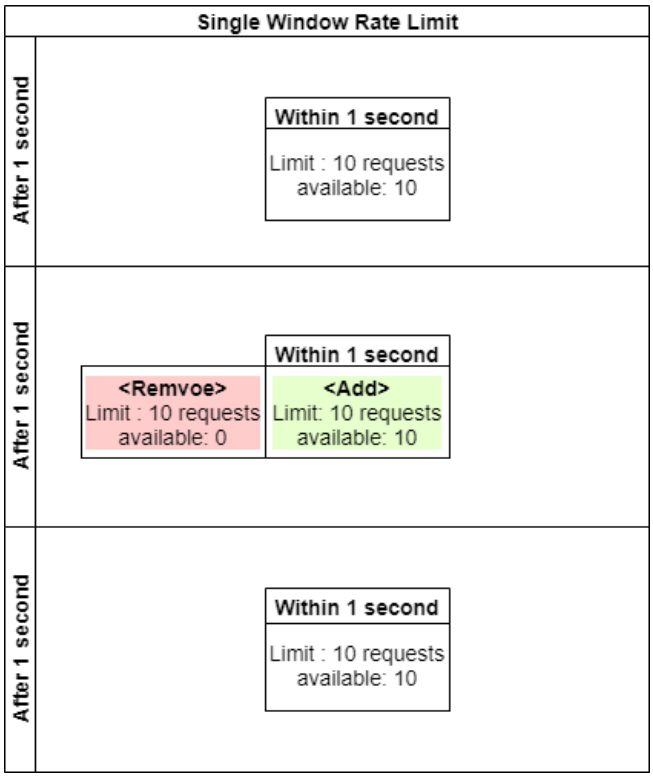
Date: 2020-09-16 00:25:22

In the development of the high availability system, we usually protect the system by different kinds of ways, such as Cache, Rate Limit, and Downgrade. And this topic will be talking about the Rate Limit. The purpose of the Rate Limit is to prevent the system not to be destroyed by large amounts of requests to the system in a short period.

The most common Rate Limit Algorithms are:

1. Fixed-windows counter Algorithm
2. Sliding windows counter Algorithm
3. Token Bucket Algorithm

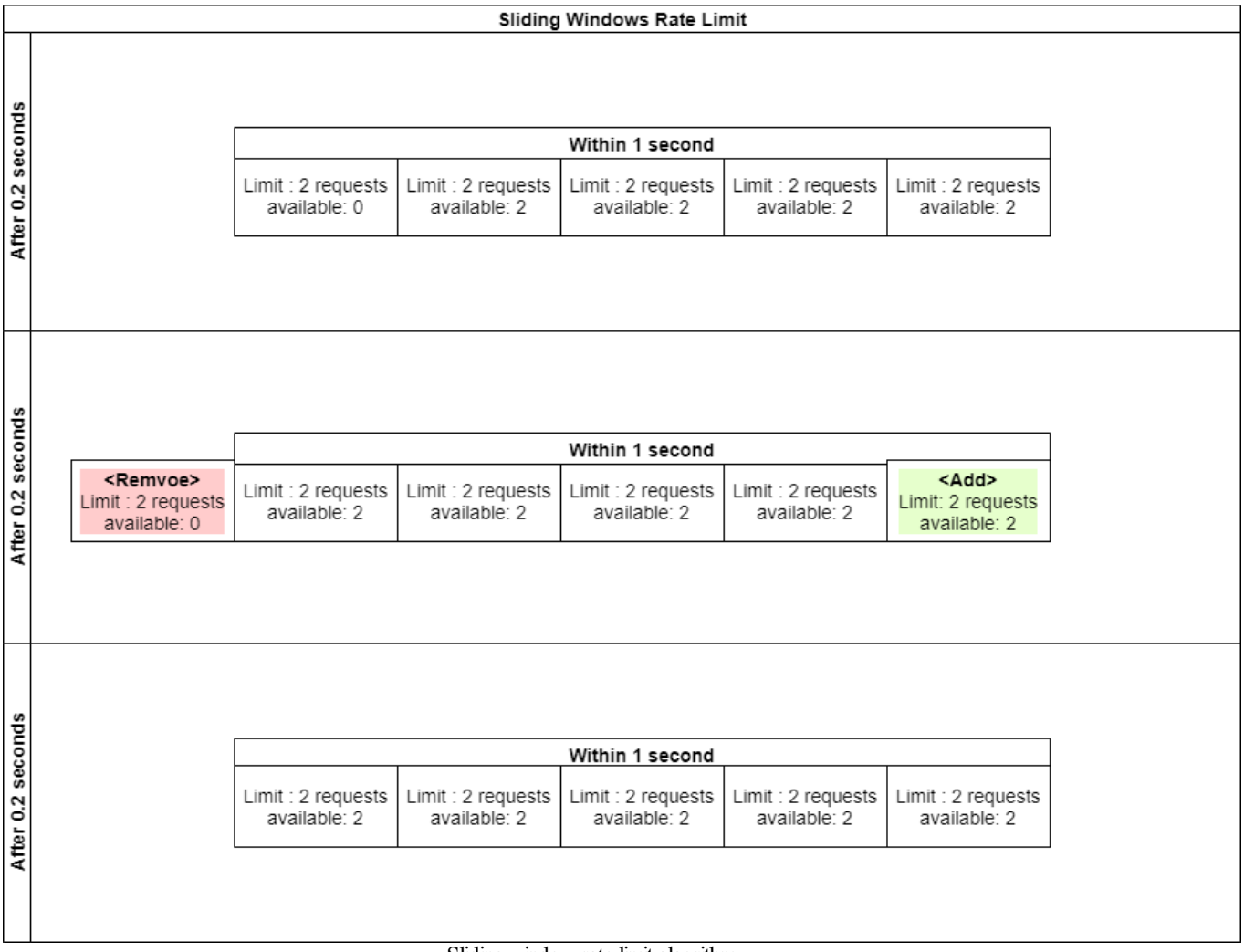
Fixed-Windows Counter Algorithm



The simplest Rate Limit implementation is to maintain a counter and check the total number of requests within a time slot. The system will reject all HTTP requests when the counter is greater than the total number of HTTP requests in the timeslot until the next timeslot starts (the Counter will initialize to ZERO at the same time).
The disadvantage of the Fixed-Windows Counter Algorithm is the maximum requests can probably be higher than the threshold.
Assume the threshold for maximum requests is 10 per second, but what if ten requests send in the last few milliseconds within a second? We will probably have a total of 20 HTTP requests per 1 second if there are other ten HTTP requests sent to the server after 1 second. How could we solve it?

https://github.com/life-is-curiosity/rate-limit-algorithms/blob/main/SingleWindowRateLimit.java

Sliding Windows Algorithm

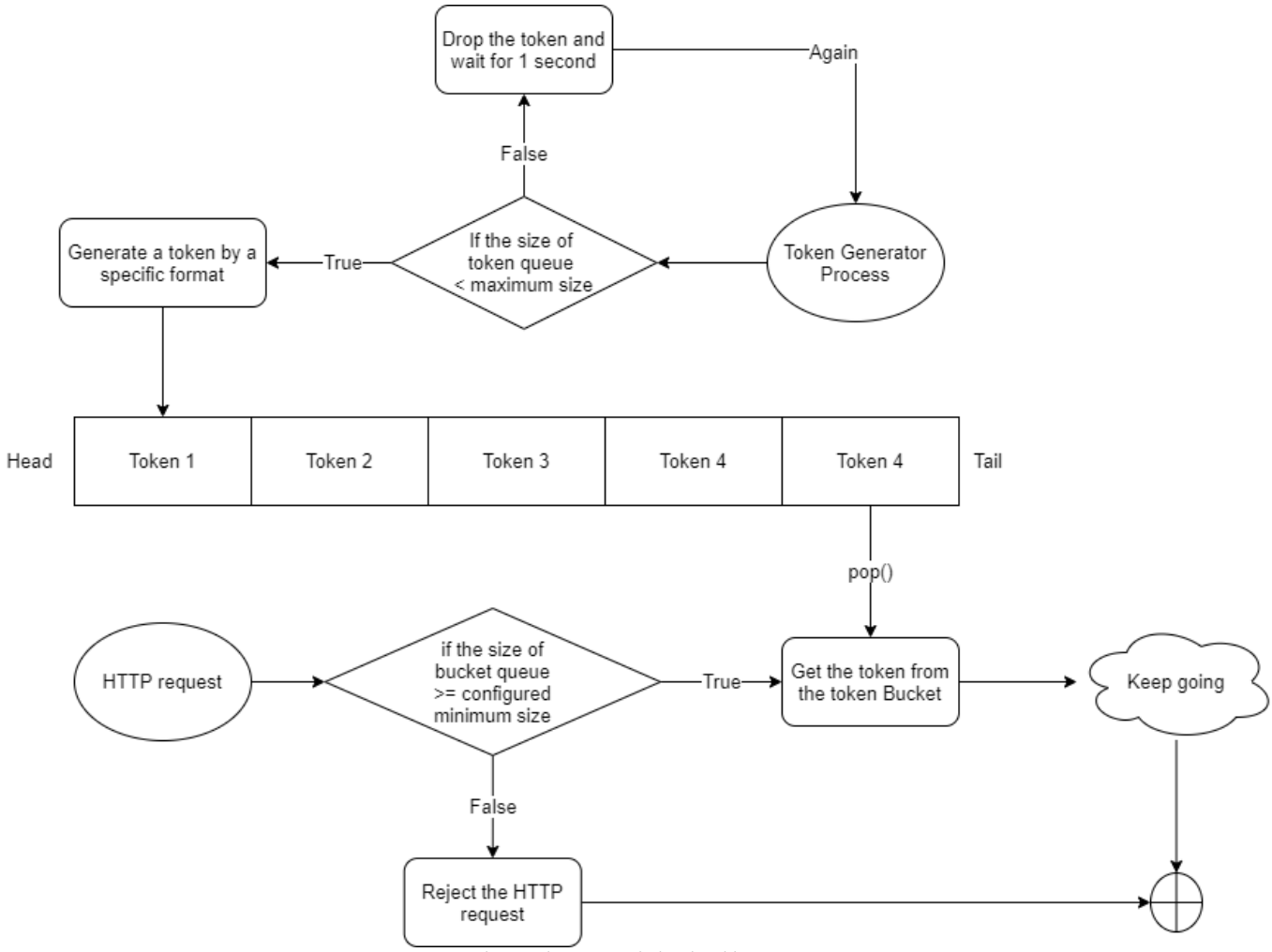


Sliding window rate limit algorithm

Obviously, only one window can not solve the problem of the threshold boundary issue then the sliding Windows Algorithm helps to solve the kind of issue. Assume the threshold for maximum requests is 10 per second, but we are going to divide it into five timeslots, every timeslot has a maximum in 2 requests per 0.2 seconds. Every 0.2 seconds passed then the timeslot will be slide to the next new window. In addition, dividing more timeslots can also be smoother for the user experience.

https://github.com/life-is-curiosity/rate-limit-algorithms/blob/main/sliding-windows.py

Token Bucket Algorithm



Leaky Bucket Rate Limit Algorithm

Another rate limit is the Token Buck Algorithm which can be used on sudden large amounts of requests situations at some special moments as long as the bucket does not be empty. At the beginning of the process, the Token Bucket is empty and the system will be adding a token to the bucket frequently. When the requests are coming, they need to take a token from the token bucket. Once the request can not get the token or the bucket is empty, the system will reject it immediately.

Conclusion

Nevertheless, every rate limit algorithm has at least a unique feature and a unique suitable situation. The simplest way to implement is the Fixed-Windows Counter Algorithm and the Token Bucket Algorithm can have a high resource usage rate and higher availability and the Sliding Windows Algorithm should have a reasonable arrangement of the timeslot.

https://github.com/life-is-curiosity/rate-limit-algorithms/blob/main/leaky-bucket.py