Design API Rate Limiter

Description

What is Rate Limiter?

Rate Limiter restricts the number of client requests received for a particular API endpoint, for a specific duration or time frame. If the number of client requests exceeds the set threshold, those requests will be ignored or dropped. We can also define rate limiter as the mechanism used to define the rate and speed at which the consumers can access the APIs. Below are some of the example use cases for rate limiter.

1. Allowing only 100 tweets allowed per hour by Twitter. Tweets beyond 100 count will be blocked or discarded with some user friendly error saying “Only 100 tweets allowed per hour”.
2. Viewing only 20 LinkedIn profiles from a specific IP Address without login. When trying to access 21st time, users are required to login.
3. Online PDF editing service allowing only 10 pdf’s per day from an IP address for their free tier service.

Why Rate Limiting is used?

Preventing DoS Attack

Denial of service attack is one of the reasons rate limiting is required. In DoS attack, a particular server or resource is bombarded with huge number of requests. By doing this the server or the system will crash, making the application inaccessible to the legitimate users. The request seems to be coming from a legitimate user, but in reality they can be bots triggering the requests intentionally.

Preventing excess server load

Some organizations have fewer resources. Due to low capacity server's, the load on the server has to be limited or minimal. Rate limiter will filter out the excess requests.

Cost Reduction

When you are using third party APIs which are charged for every call you make, in this case you want to limit the number of requests made to paid third party APIs.

Functional Requirements

1. Lets say we want to limit the request to 20 requests per minute duration.
2. The user should get an error message when the threshold of 20 requests are completed.

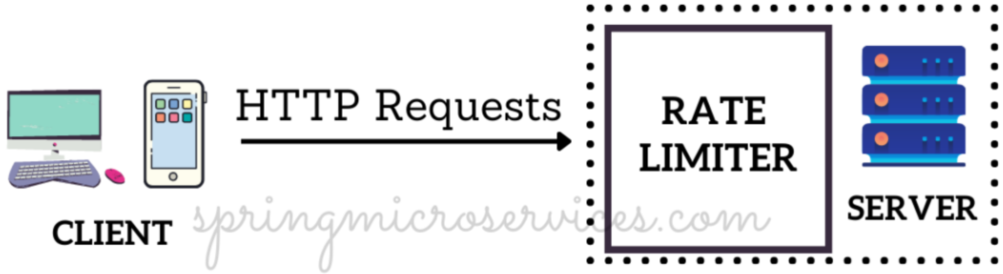
Non-Functional Requirements

1. The system should be available and accurately rate limit the user.
2. After introducing the rate limiter, the system should not slow down or the performance should not be impacted.
3. Rate limiter should work correctly with multiple servers.
4. Entire system should not go down when there are issues in API rate limiter .

High Level System Design for Rate Limiter

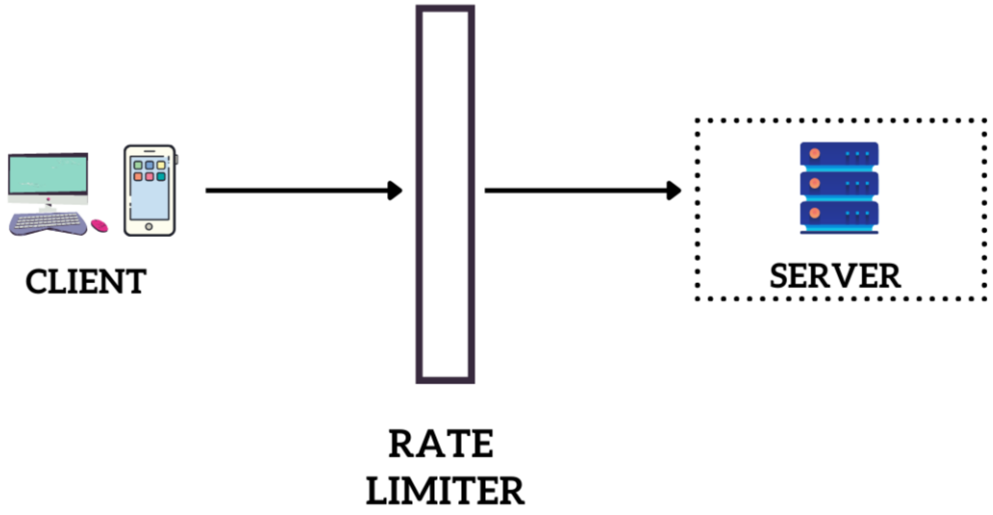
There are 3 places where we can keep the API rate limiter as shown below.

Server side rate limiter



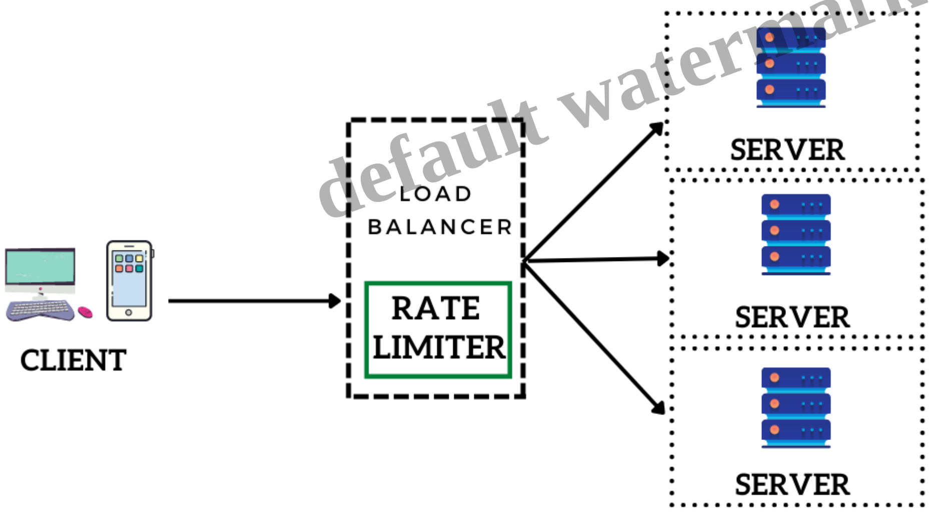
Server side rate limiter

Rate Limiter middleware



Rate Limiter middleware

Rate limiter within gateway



Rate limiter inside load balancer

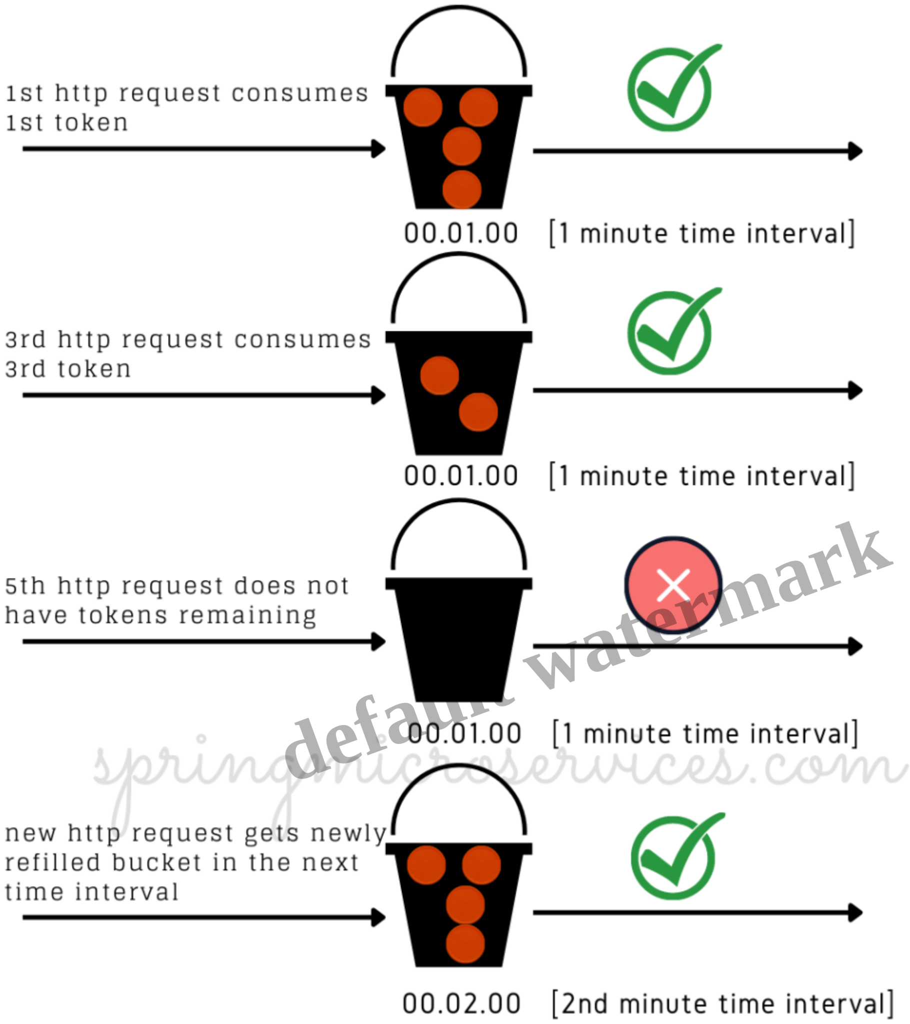
Algorithms for Rate Limiter

In this section we will discuss about some of the most popular algorithms used for implementation of rate limiter.

What is Token Bucket algorithm

Token bucket is one of the widely used rate limiting algorithm. Large companies like Amazon and Stripe use this algorithm. Lets understand how the token bucket works.

For every incoming request, a token is assigned from the bucket. Bucket will have a capacity which will hold ‘n’ number of tokens only, e.g. 10. Whenever the tokens are consumed by the incoming request, and the bucket is emptied, the bucket will be refilled with another 10 tokens when the next time interval starts. So for every time interval say 60 seconds , only 10 requests can be accepted by consuming 10 tokens from the bucket. Any future requests will dropped since the quota of 10 requests was served.

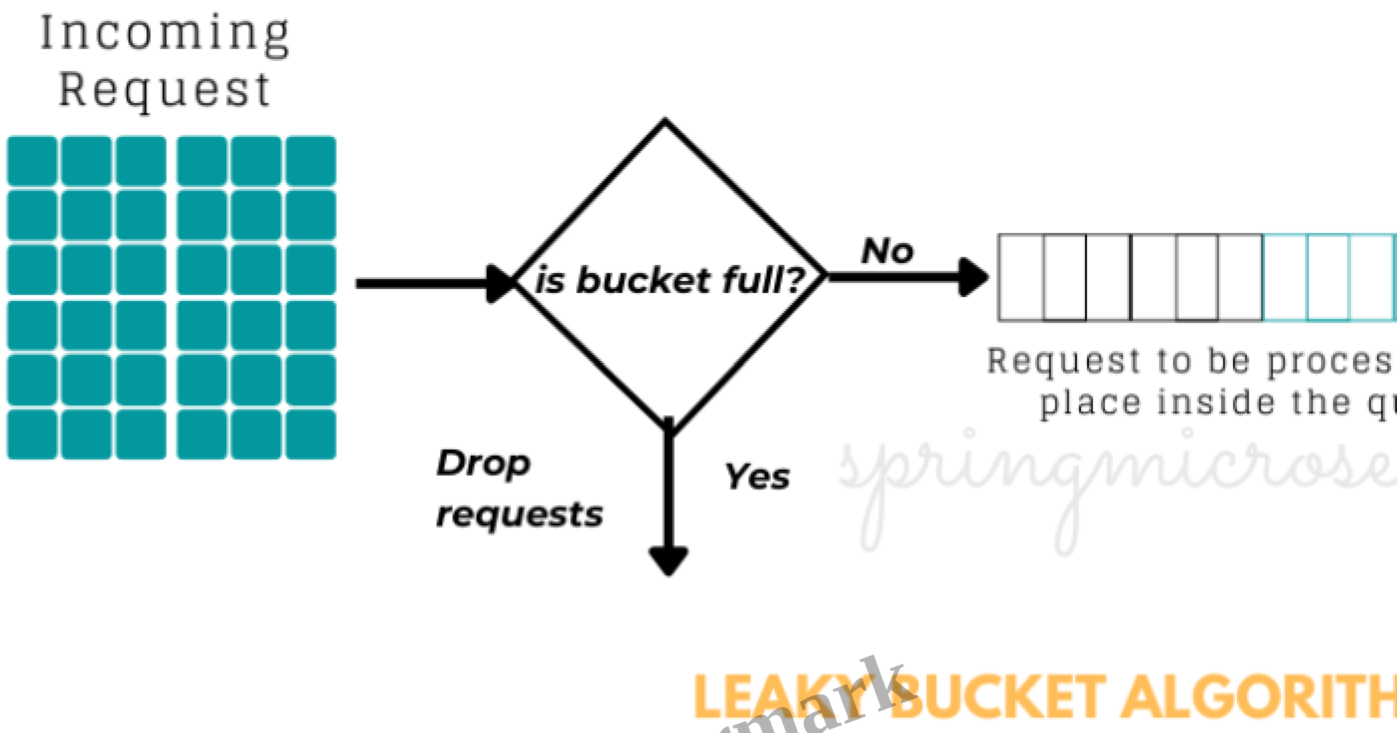

1st http request consumes 1st token
00.01.00 [1 minute time interval]
3rd http request consumes 3rd token
00.01.00 [1 minute time interval]
5th http request does not have tokens remaining
00.01.00 [1 minute time interval]
new http request gets newly refilled bucket in the next time interval
00.02.00 [2nd minute time interval]

Token bucket algorithm

What is Leaky Bucket algorithm

Leaky bucket algorithm is almost similar to token bucket algorithm, the only difference is here we have queue’s instead of bucket, and the requests are processed at a fixed rate (time interval).

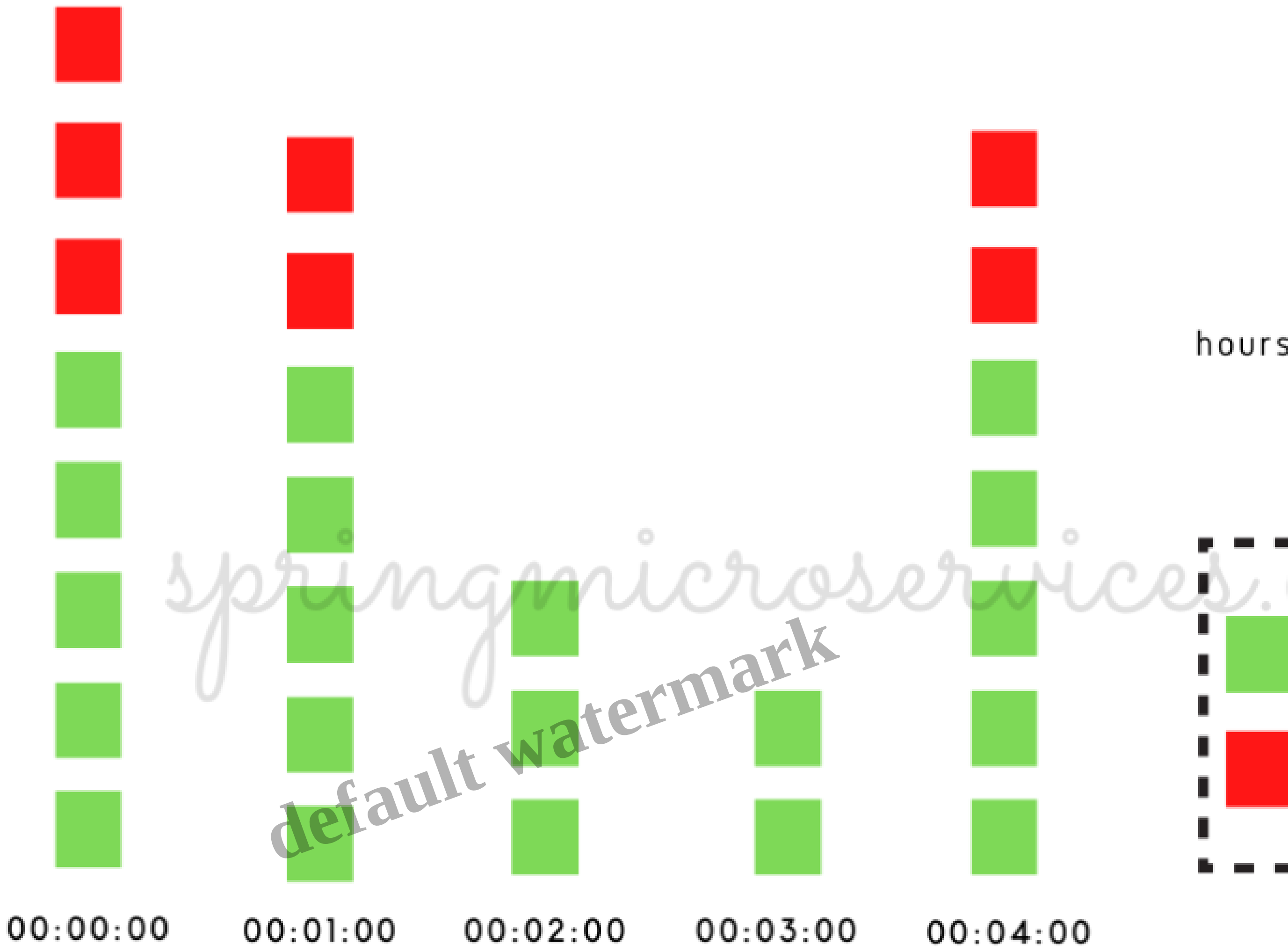
There are two parameters in leaky bucket algorithm
Bucket size – Bucket size is the size of the queue which holds the requests.
Outflow rate – It defines how many requests can be processed at a fixed rate (time parameter, i.e. seconds)



1. As shown in the above diagram, when a new request arrives, it is added in the queue to start processing.
2. In order the process a request, it is pulled from the queue.
3. If the queue is full, the requests are dropped.

What is Fixed window counter algorithm

In Fixed window counter algorithm, we only allow ‘n’ number of requests for a fixed duration of time. ‘n’ is the predefined threshold and fixed duration can be per seconds or per minute. To keep track of the number of requests processed, we have a counter which is incremented as the requests arrive, and only those are requests are allowed within that fixed window duration. E.g. ‘n’ requests per minute. Once the request count reaches the predefined counter, the excessive request are dropped.



Design Rate Limiter : Interview question

Candidates who are new to system design or not worked extensively on large applications are unaware of what is rate limiter and how to design rate limiter. This post is sufficient enough to get a good grasp on rate limiter system design. We have discussed three rate limiting algorithms in this post, you can explain all three or any two would be fine.

Category

1. Design

Date Created
August 5, 2022
Author
kk-ravi144gmail-com