System Design: The Rate Limiter

Let's understand the basic details to design a rate limiter.

We'll cover the following

What is a rate limiter?

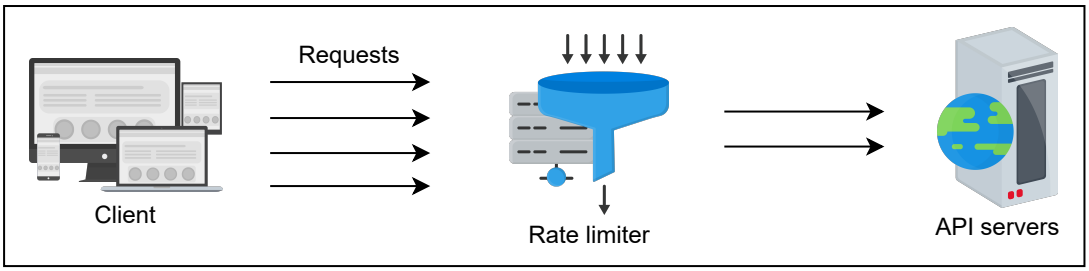
A **rate limiter**, as the name suggests, puts a limit on the number of requests a service fulfills. It throttles requests that cross the predefined limit. For example, a client using a particular service's API that is configured to allow 500 requests per minute would block further incoming requests for the client if the number of requests the client makes exceeds that limit.

Why do we need a rate limiter?

A rate limiter is generally used as a defensive layer for services to avoid their excessive usage, whether intended or unintended. It also protects services against abusive behaviors that target the application layer, such as **denial-of-service (DOS)**↴ attacks and brute-force password attempts.

Below, we have a list of scenarios where rate limiters can be used to make the service more reliable.

- **Preventing resource starvation:** Some denial of service incidents are caused by errors in software or configurations in the system, which causes resource starvation. Such attacks are referred to as friendly-fire denial of service. One of the common use cases of rate limiters is to avoid resource starvation caused by such denial of service attacks, whether intentional or unintentional.
- **Managing policies and quotas:** There is also a need for rate limiters to provide a fair and reasonable use of resources' capacity when they are shared among many users. The policy refers to applying limits on the time duration or quantity allocated (quota).
- **Controlling data flow:** Rate limiters could also be used in systems where there is a need to process a large amount of data. Rate limiters control the flow of data to distribute the work evenly among different machines, avoiding the burden on a single machine.
- **Avoiding excess costs:** Rate limiting can also be used to control the cost of operations. For example, organizations can use rate limiting to prevent experiments from running out of control and avoid large bills. Some cloud service providers also use this concept by providing freemium services to certain limits, which can be increased on reques by charging from users.



Throttling the number of requests to API servers via a rate limiter

How will we design a rate limiter?

In the following lessons, we will learn about the following:

1. **Requirements:** This is where we discuss the functional and non-functional requirements of the rate limiter. We also describe the types of throttling and locations where a rate limiter can be placed to perform its functions efficiently.
2. **High-level design:** In this section, we look at the high-level design to provide an overview of a rate limiter.
3. **Detailed design:** In this section, we discuss the detailed design of a rate limiter and explain various building blocks involved in the detailed design.
4. **Rate limiter algorithms:** In this lesson, we explain different algorithms that play a vital role in the operations of a rate limiter.
5. **Quiz:** To assess your understanding of rate limiters, we've provided a quiz at the end of this chapter.

In the next lesson, let's start by understanding the requirements and design of a rate limiter.

Back

Design of a Pub-sub System

Next

Requirements of a Rate Limiter's Design

Mark as Complet