

The pros and cons of the Throttling architecture pattern

<https://www.redhat.com/en/blog/pros-and-cons-throttling>

As you browse redhat.com, we'll recommend resources you may like. For now, try these.

Share

Having a grasp of common architectural patterns is essential to designing software architecture at scale. Using them saves not only time but also ensures a reliable implementation of your design. There's no need to reinvent the wheel when there's an architectural pattern available that applies to an architecture you're developing.

The following is a brief overview of the Throttling architectural pattern.

The Throttling pattern, also known as Rate Limiting, is one in which a throttle is placed in front of a target service or process to control the rate at which data flow into the target. Throttling is a technique that ensures that the flow of data being sent into a target can be digested at an acceptable rate. Should the target become overwhelmed, the throttle will slow down or even stop calls to the target.

In addition to being used as a defensive measure, the Throttling pattern can be used by consumers of cloud providers as a way to control consumption costs. Consumers can use tools to set a rate limit on calls made to a specific third-party service. This limit is usually defined as calls per second. Thus, clients avoid incurring unanticipated costs.

Pros

Cons

The essential benefit of the Throttling pattern is that it allows a system to control both internal and external traffic that might compromise a system's ability to operate in a safe and predictable manner. Unanticipated bursts in call activity do happen; sometimes by accident; sometimes by nefarious intention. The Throttling pattern mitigates the impact of such bursts.

Client-side throttling will control the rate at which calls are made to commercial services that

calculate fees according to usage activity. Thus, the Throttling pattern is used as a cost-control mechanism.

Using the Throttling pattern internally on the server-side protects a system from the DDoS acts from bad actors. Also, internal throttling protects the system from unanticipated burst activity from other internal processes and services. The tradeoff is that when throttling is in effect, the system might slow down and thus overall system performance can be affected. Hence, the Throttling Pattern is often used in conjunction with the Circuit Breaker pattern.

Bob Reselman is a nationally known software developer, system architect, industry analyst, and technical writer/journalist. Over a career that spans 30 years, Bob has worked for companies such as Gateway, Cap Gemini, The Los Angeles Weekly, Edmunds.com and the Academy of Recording Arts and Sciences, to name a few. He has held roles with significant responsibility, including but not limited to, Platform Architect (Consumer) at Gateway, Principal Consultant with Cap Gemini and CTO at the international trade finance company, ItFex.

The latest on IT automation for tech, teams, and environments

Updates on the platforms that free customers to run AI workloads anywhere

Explore how we build a more flexible future with hybrid cloud

The latest on how we reduce risks across environments and technologies

Updates on the platforms that simplify operations at the edge

The latest on the world's leading enterprise Linux platform

Inside our solutions to the toughest application challenges

Entertaining stories from the makers and leaders in enterprise tech

We're the world's leading provider of enterprise open source solutions including Linux, cloud, container, and Kubernetes. We deliver hardened solutions that make it easier for enterprises to work across platforms and environments, from the core datacenter to the network edge.