#ASLI ENGINEERING

API Retries - Thundering Herd Problem

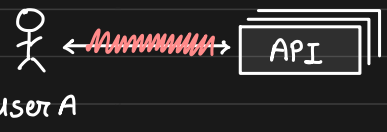
SWIPE

BY

ARPIT BHAYANI

Thundering Herd Problem

Say user A makes an API call to the backend and there was some issue with network and hence the call failed!



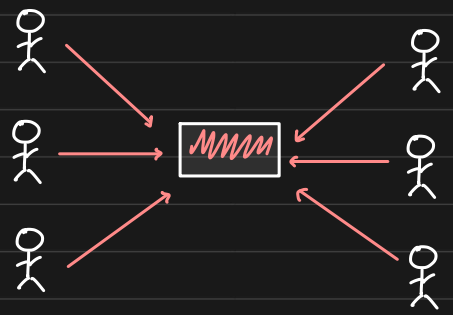
What do we do ?

We retry * assuming our APIs are idempotent

But, how should we retry?

Simplest way of doing it

```
For i = 1 to 3 ;
    doIt();
```



Simple enough solution, but it falters at scale... imagine the server is overwhelmed because of which API call failed and every single connected client retries!!

Things just got worse

The already overwhelmed server

- more request due to retries
- new requests

No room to recover

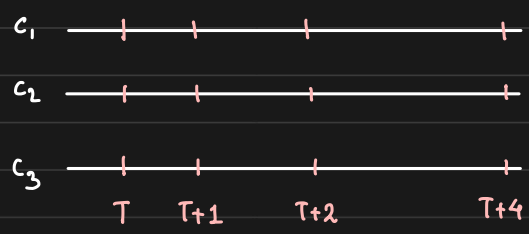
Exponential Backoff

Instead of repeating immediately back to back we retry with a backoff

This gives server a breathing space and the much needed time to recover.

But, the retries still coincide!!

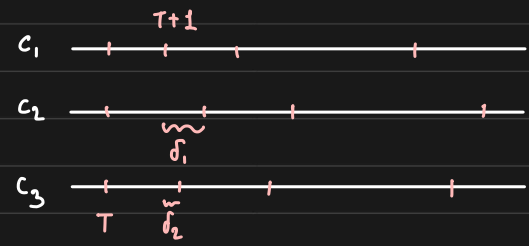
↳ puts pressure on server



You see this in action on Gmail and Slack when internet goes away

Exponential Backoff + Jitter

Instead of retrying at immediate instant, we add some random Jitter and this randomness (delay) reduces the coincidences during retries.



This ensures retries are distributed and does not add to the problem.

While implementing retries ensure

1. you add random jitter
2. retries are exponentially spaced