YOLO Architecture

Motivation

Yolo service is dependant on an unreliable WTF service, the aim of this architecture is to make Yolo service more reliable than WTF.

YOLO architecture should deal with all the traffic without overloading the serving with a huge message queue.

The architecture should let YOLO know when to stop issuing calls to WTF until WTF is available again.

Constraints

- YOLO only waits up to 30 seconds for each call made to WTF.
- YOLO's server has a limited capacity of 10 simultaneous requests.
- YOLO has at least 5 requests per second.
- YOLO is using the best server in the market.
- YOLO needs to make synchronous calls to WTF in each request.

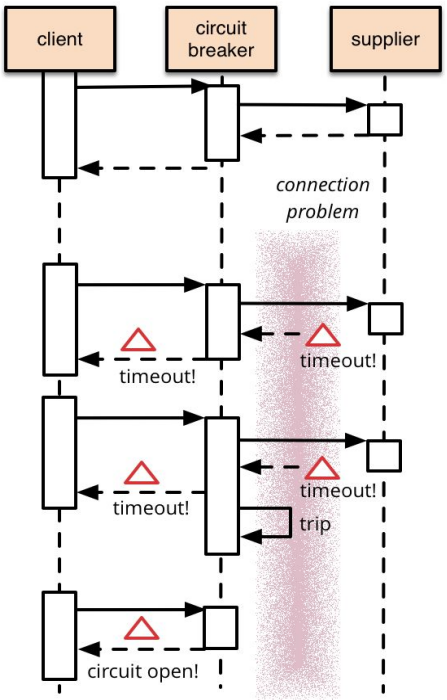
Architecture decisions

Yolo TPS is at least 5tr/s, considering that only 10 transactions can be handled on simultaneos therefore the maximum time a transaction can take is 2s to avoid queueing.

As the max elapsed time per transaction is way smaller than WTF timeout 30s, the architecture should monitor the execution time and ensure it doesn’t go above the max of 2s.

Also the system should monitor calls to WTF and stops calling it if too many calls fail, it should also restore communication after WTF is back.

A known pattern to deal with this type of constraints is a **circuit breaker**:



The circuit breaker should be parameter with the following parameters:

- Failure threshold, the number of fail request to open the circuit (stop calling WF)
- Success threshold, the number of successful trail request to close back the circuit (resume calling WTF).
- Timeout, the max time that a request take. If the request timeouts it is considered as a failure.

Since there is no condition of what YOLO should respond when WTF is down, the most simplistic answer is to send back an error message.

Going further

If data requested from WTF is not volatile (it doesn't change very fast), we could consider creating a proxy of WTF with cached data. This cache can be synchronised with WTF whenever a successful WTF request is processed. Also this cache can serve as failover data source for when the circuit is open.