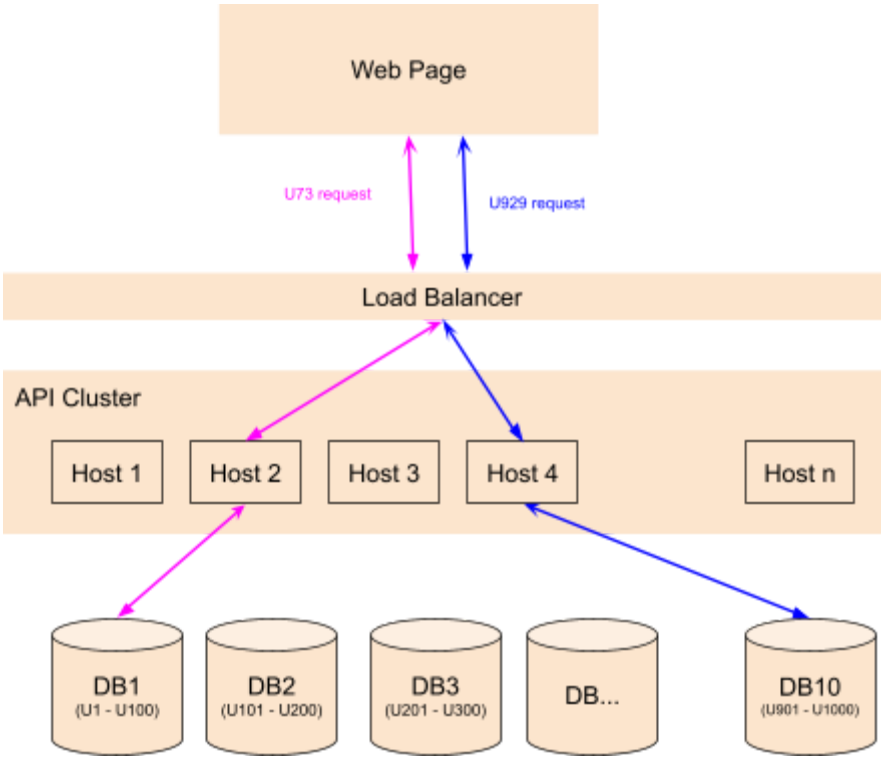
Dynamic SafeGuard Mechanism against slow responses

Problem Statement

Setup

We have client server architecture (as shown in the diagram)

- There are a cluster of Web Services/API hosts serving user requests.
- This cluster is behind a single load balancer.
- We have a cluster of 10 database instances hosting 1M users each.
- User logs into a web page and makes a call to get a list of tasks assigned to him/her which are stored in the corresponding DB.
- Each API request takes total 1 sec (800 ms in API host and 200ms in DB query)
- In normal load there are a total 100 RPS (10 each for each DB user). But sometimes the split of the number of requests for different DB users can vary but never goes above 100 RPS total.
- There are total 120 workers/threads setup across API hosts to serve these requests
- Assume each DB belongs to a different department in the company
- Each API request has the information about which DB the user belongs to



Problem Scenarios

- Problem1: Once in a while assume that randomly any DB call can take upto 10sec instead of 200ms due to some reason but other calls go fine.
- Problem 2: One particular DB can go slow fully and all calls to that start to take 9.8 sec each. It is possible that the DB can go bad for few minutes/hours but can recover on its own or with manual intervention

Expected solution

- (Problem 1 scenario) Device a solution which serves all users (even the users whose DB calls take much longer) until there are empty threads in the system to serve without impacting the response time for all other users.
- (Problem 1,2 scenarios) Extend the solution so that beyond a point if the system can't serve all users due to slow requests, it can start to penalize/block requests only of the bad performance DB (hence department) without impacting users of other departments.
- The system should degrade the service (when DB/request performance degrades) or recover (when DB performance improves) automatically without any manual intervention.