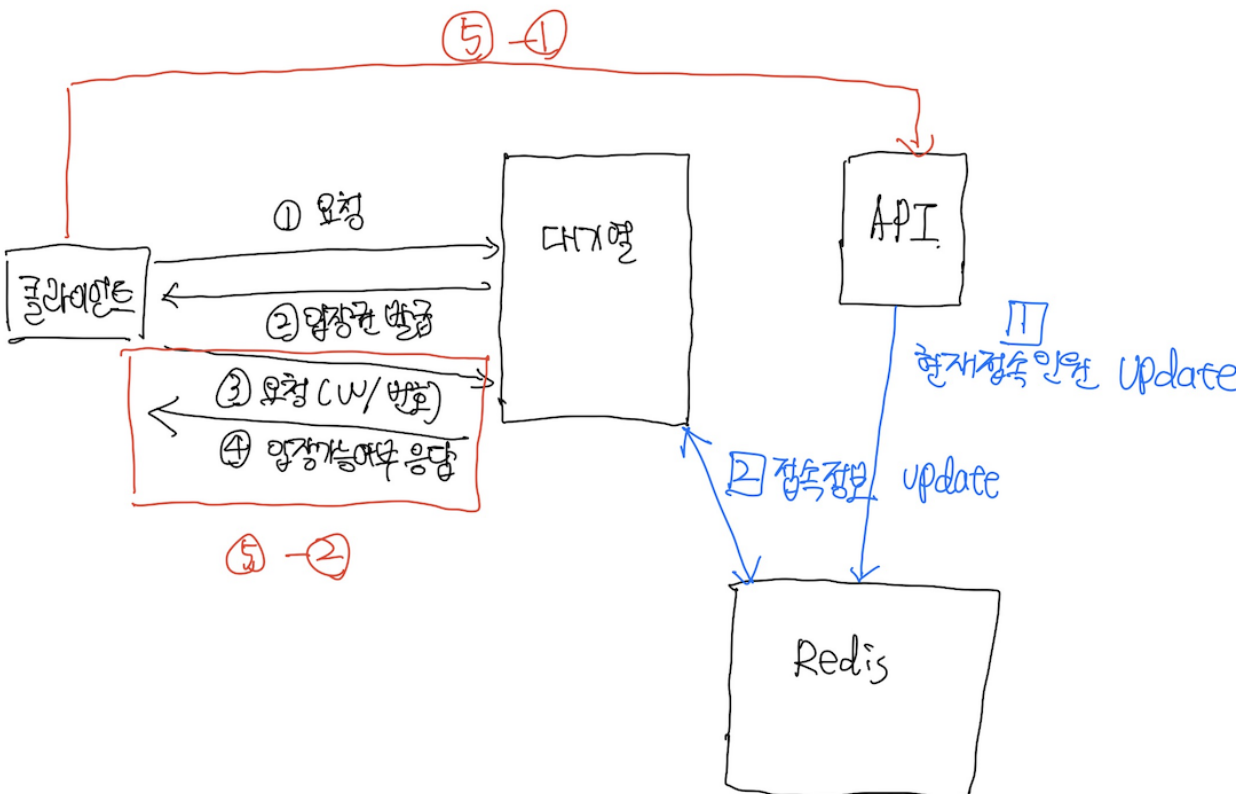


모바일 앱 - 대기열 구현 : 이호경

요구사항

- 모바일 어플리케이션 최대 동시접속 인원 제한 (유동적으로 변경할 수 있어야 함)
- 특정 유저는 인원 제한과 상관없이 통과할 수 있어야 함
- 대기하는 사람은 몇 명 인원이 남았는지 알 수 있어야 함

개략적인 구조



클라이언트 - 대기열

- ① 최초 접속시 입장이 가능한지 대기열 서버에 별도의 파라미터 없이 요청
- ② 대기열 서버는 파라미터가 없다면, 입장 번호를 발급 (기존 입장 발급번호 + 1)

→ 이 때, 입장이 가능하다면 바로 입장 처리 ⑤-①로 바로 이동

③ 발급 받은 번호로 입장이 가능한지 대기열에 요청

④ 입장이 불가능할 경우, 몇명이 대기중인지 응답, 입장 가능시, 마지막 입장 시간 업데이트)

⑤-① 입장이 가능할 경우 API 요청

⑤-② 입장이 불가능할 경우, 5초마다 재요청하여 확인

대기열 - REDIS - API

① 현재 접속 인원 REDIS에 Update (CurrentUsserSize)

② 최대 입장 가능 수 : 2000

현재 발급된 입장 번호

현재 입장 가능 번호

마지막 입장 번호 (만료 시간 추가)

생각해볼 것

- 언제 입장 가능 인원을 늘려줄 것인가?
- 입장권을 발급 받고, 입장 가능 시간이 되었을 때, 입장하지 않았을 경우 어떻게 처리할지?

고민했던 것

- 컬렉션에 넣고 싶었으나, 조회시 $O(N)$ 이기 때문에 제외. 사실 특정 유저가 그렇게 많지 않으므로 크게 문제는 되지 않을 것으로 생각되나, request * $O(N)$ 이므로 변경
- 특정 유저는 FreePass:<DeviceId> 조합으로 키 저장.
 - 문제점 : 특정 유저를 관리하기 어려움. 클러스터 환경이 아니라면 가상 DB를 사용 해, 해당 DB에 따로 저장하면 되지만 클러스터 환경에서는 불가.(실제 stage 갔을 때야 알게 됨)
 - 클러스터 환경에서 사용할 수 없는 명령어들은 별도로 정리 필요

- 대기열이 무너진다면? API와 같은 쿠버네티스 환경에 있기 때문에, 대기열이 무너졌다면, 연쇄적으로 API도 정상적으로 작동하지 않음을 가정함. 오버엔지니어링으로 판단하여, 별도로 클라이언트에서 구현 안함.

참고자료

- http://ndcreplay.nexon.com/NDC2019/sessions/NDC2019_0069.html
- 가상 면접 사례로 배우는 대규모 시스템 설계 기초 - 알렉스 쉬(이병준 역), 인사이트, 2021