

실전! Redis 활용

신동현

섹션 0. Orientation

강좌 소개

강의 목적

실무에 적용 가능한 Redis 활용 방법을 익힌다

- 캐싱(Caching) 기법
- Redis의 다양한 Data type을 사용한 문제 해결법
 - String, List, Set, Hash, Sorted Set, Stream, Bitmap, Geospatial, etc.
- 대용량 트래픽 처리를 위한 Redis 활용 사례

수강 대상

백엔드 신입 개발자(또는 취준생)

Redis를 제대로 사용하는 방법이 궁금하신 분

대용량 트래픽 처리를 위한 Redis 활용법이 궁금하신 분

실전! Redis
활용

섹션 0. Orientation

커리큘럼

1. Redis 알아보기
2. Redis 설치
3. 데이터 타입 알아보기
String, List, Set, Hash, ZSet, Streams, Geospatial, Bitmaps, HyperLogLog, BloomFilter
4. Redis 특수 명령어
SET NX/XX, Pub/Sub, Pipeline, Transaction
5. 데이터 타입 활용
임시 비밀번호(One-Time Password) / 분산 락(Distributed Lock) / 비율 제한기(Rate Limiter)
소셜 피드(SNS Activity Feed) / 중복 제거(Unique Items) / 로그인 세션(Session)
온라인 상태 표시(User Online Status) / 반경 탐색(Geofencing) / 방문자 수 계산(Visitors
Count)
6. Redis 사용시 주의사항

섹션 0. Orientation

강사 소개

신동현

- 백엔드 개발자
 - 에이슬립 슬립루틴 서비스 개발
 - (전) 에이블리 광고 사업팀
- PyCon Korea 21/22/23 Speaker
- 3x AWS Certified
 - AWS StartUp Security GameDay 1등(2023.04)
 - AWS Certified Database – Specialty
 - AWS Certified Solutions Architect – Associate
 - AWS Certified Developer – Associate
- 기술 블로그: qu3vipon.com
- 깃허브: github.com/qu3vipon

실전! Redis
활용

섹션 1. Redis 알아보기

Redis 소개

Redis

Remote Dictionary Server

Open Source In-Memory Data Store written in ANSI-C

DB Ranking

Total 6위 <https://db-engines.com/en/ranking>

Key-value 1위 <https://db-engines.com/en/ranking/key-value+store>

실전! Redis
활용

섹션 1. Redis 알아보기

Redis 특징

특징

In-Memory	모든 데이터를 RAM에 저장 (백업 / 스냅샷 제외)
Single Threaded	단일 thread에서 모든 task 처리
Cluster Mode	다중 노드에 데이터를 분산 저장하여 안정성 & 고가용성 제공
Persistence	RDB(Redis Database) + AOF(Append only file) 통해 영속성 옵션 제공
Pub/Sub	Pub/Sub 패턴을 지원하여 손쉬운 어플리케이션 개발(e.g 채팅, 알림

등)

섹션 1. Redis 알아보기

Redis 장점

장점

높은 성능 ★★

모든 데이터를 메모리에 저장하기 때문에 매우 빠른 읽기/쓰기 속도 보장

Data Type 지원 ★

Redis에서 지원하는 Data type을 잘 활용하여 다양한 기능 구현

클라이언트 라이브러리

Python, Java, JavaScript 등 다양한 언어로 작성된 클라이언트 라이브러리 지원

다양한 사례 / 강한 커뮤니티

Redis를 활용하여 비슷한 문제를 해결한 사례가 많고, 커뮤니티 도움 받기 쉬움

실전! Redis
활용

섹션 1. Redis 알아보기

Redis 사용 사례

Caching	임시 비밀번호(One-Time Password) 로그인 세션(Session)
Rate Limiter	Fixed-Window / Sliding-Window Rate Limiter(비율 계산기)
Message Broker	메시지 큐(Message Queue)
실시간 분석 / 계산	순위표(Rank / Leaderboard) 반경 탐색(Geofencing) 방문자 수 계산(Visitors Count)
실시간 채팅	Pub/Sub 패턴

섹션 1. Redis 알아보기

Persistence

Persistence(영속성)

Redis는 주로 캐시로 사용되지만 데이터 영속성을 위한 옵션 제공
SSD와 같은 영구적인 저장 장치에 데이터 저장

RDB(Redis Database)

Point-in-time Snapshot -> 재난 복구(Disaster Recovery) 또는 복제에 주로 사용
일부 데이터 유실의 위험이 있고, 스냅샷 생성 중 클라이언트 요청 지연 발생

AOF(Append Only File)

Redis에 적용되는 Write 작업을 모두 log로 저장
데이터 유실의 위험이 적지만, 재난 복구시 Write 작업을 다시 적용하기 때문에 RDB 보다 느림

RDB + AOF 함께 사용하는 옵션 제공

실전! Redis
활용

섹션 1. Redis 알아보기

Caching

캐싱(Caching)

데이터를 빠르게 읽고 처리하기 위해 임시로 저장하는 기술

계산된 값을 임시로 저장해두고, 동일한 계산 / 요청 발생시 다시 계산하지 않고 저장된 값 바로 사용

캐시(Cache) = 임시 저장소

사용 사례

CPU 캐시 CPU와 RAM의 속도 차이로 발생하는 지연을 줄이기 위해 **L1, L2, L3** 캐시

사용

웹 브라우저 캐싱 웹 브라우저가 웹 페이지 데이터를 로컬 저장소에 저장하여 해당 페이지

재방문시 사용

DNS 캐싱 이전에 조회한 도메인 이름과 해당하는 **IP** 주소를 저장하여 재요청시 사용

데이터베이스 캐싱 데이터베이스 조회나 계산 결과를 저장하여 재요청시 사용

CDN 원본 서버의 콘텐츠를 **PoP** 서버에 저장하여 사용자와 가까운 서버에서 요청

처리

어플리케이션 캐싱 어플리케이션에서 데이터나 계산 결과를 캐싱하여 반복적 작업

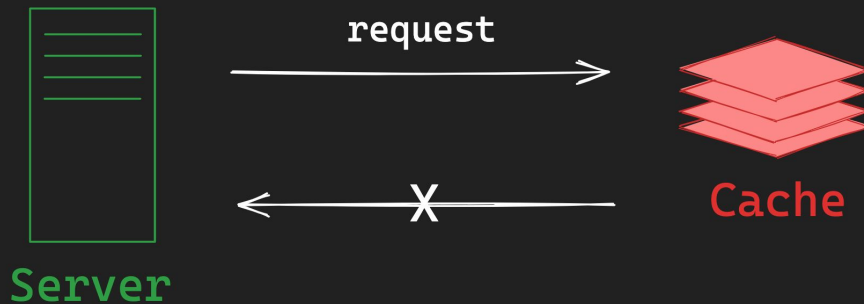
섹션 1. Redis 알아보기

Cache Hit / Miss

✓ Cache Hit



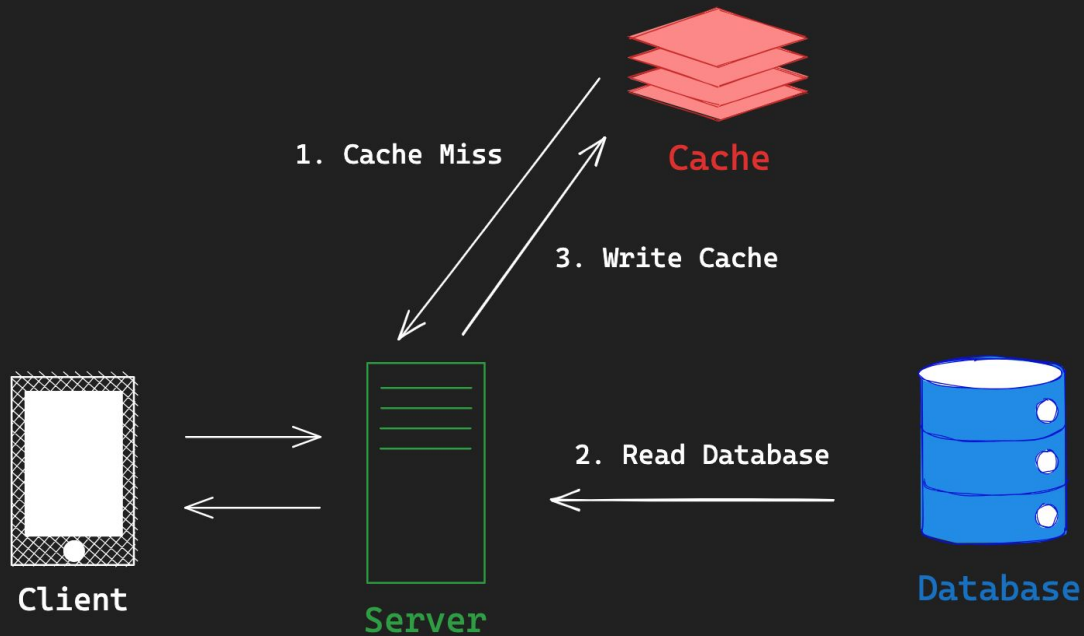
✗ Cache Miss



실전! Redis
활용

섹션 1. Redis 알아보기

Cache-Aside pattern



실전! Redis
활용

섹션 2. Redis 설치

Redis 설치 / 실행

Redis 설치

MacOS <https://redis.io/docs/getting-started/installation/install-redis-on-mac-os/>

Windows <https://redis.io/docs/getting-started/installation/install-redis-on-windows/>

Redis 실행

\$ redis-cli

\$ ping

실전! Redis
활용

섹션 2. Redis 설치

데이터 저장/조회/삭제

저장

\$ SET lecture inflearn-redis

조회

\$ GET lecture

삭제

\$ DEL lecture

섹션 3. 데이터 타입 알아보기

Strings

Strings 문자열, 숫자, serialized object(JSON string) 등 저장

명령어 \$ SET lecture inflearn-redis

\$ MSET price 100 language ko

\$ MGET lecture price language

\$ INCR price

\$ INCRBY price 10

\$ SET '{"lecture": "inflearn-redis", "language": "en"}'

\$ SET inflearn-redis:ko:price 200

섹션 3. 데이터 타입 알아보기

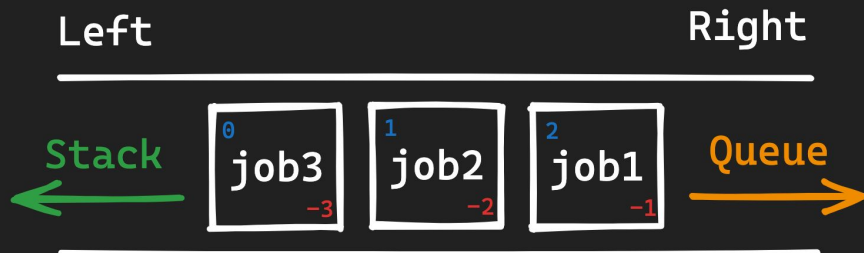
Lists

Lists String을 Linked List로 저장 -> push / pop에 최적화 $O(1)$
Queue(FIFO) / Stack(FILO) 구현에 사용

명령어 \$ LPUSH queue job1 job2 job3
 \$ RPOP queue

\$ LPUSH stack job1 job2 job3
\$ LPOP stack

\$ LPUSH queue job1 job2 job3
\$ LRANGE queue -2 -1
\$ LTRIM queue 0 0



실전! Redis
활용

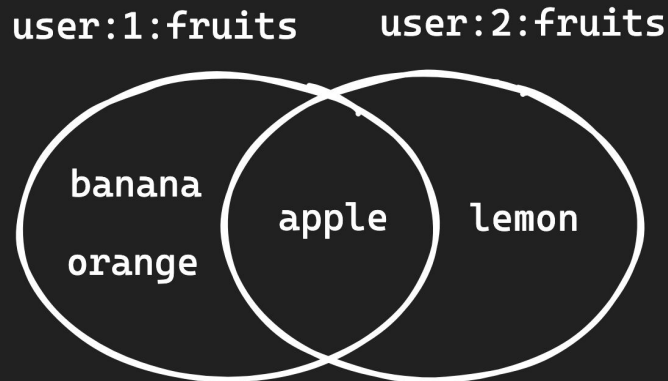
섹션 3. 데이터 타입 알아보기

Sets

Sets Unique string을 저장하는 정렬되지 않은 집합
Set Operation 사용 가능(e.g. intersection, union, difference)

명령어

```
$ SADD user:1:fruits apple banana orange orange  
$ SMEMBERS user:1:fruits  
$ SCARD user:1:fruits  
$ SISMEMBER user:1:fruits banana  
  
$ SADD user:2:fruits apple lemon  
  
$ SINTER user:1:fruits user:2:fruits  
$ SDIFF user:1:fruits user:2:fruits  
$ SUNION user:1:fruits user:2:fruits
```



실전! Redis
활용

섹션 3. 데이터 타입 알아보기

Hashes

Hashes field-value 구조를 갖는 데이터 타입
다양한 속성을 갖는 객체의 데이터를 저장할 때 유용

명령어 \$ HSET lecture name inflearn-redis price 100 language ko

\$ HGET lecture name

\$ HMGET lecture price language invalid

\$ HINCRBY lecture price 10

섹션 3. 데이터 타입 알아보기

Sorted Sets

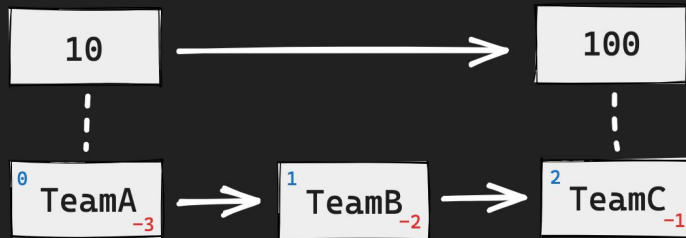
ZSets Unique string을 연관된 score를 통해 정렬된 집합(Set의 기능 + 추가로 score 속성 저장)
내부적으로 Skip List + Hash Table로 이루어져 있고, score 값에 따라 정렬 유지
score가 동일하면 lexicographically(사전 편찬 순) 정렬

명령어 \$ ZADD points 10 TeamA 10 TeamB 50 TeamC

\$ ZRANGE points 0 -1

\$ ZRANGE points 0 -1 REV WITHSCORES

\$ ZRANK points TeamA



섹션 3. 데이터 타입 알아보기

Streams

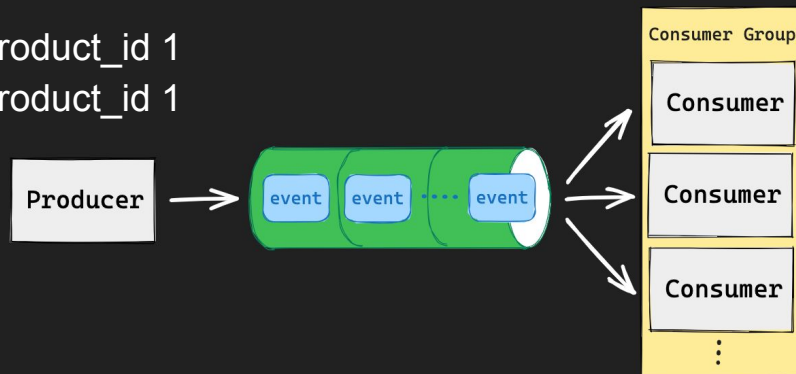
Streams append-only log에 consumer groups과 같은 기능을 더한 자료 구조

추가 기능 unique id를 통해 하나의 entry를 읽을 때, $O(1)$ 시간 복잡도
Consumer Group을 통해 분산 시스템에서 다수의 consumer가 event 처리

명령어 \$ XADD events * action like user_id 1 product_id 1
\$ XADD events * action like user_id 2 product_id 1

\$ XRANGE events - +

\$ XDEL events ID



실전! Redis
활용

섹션 3. 데이터 타입 알아보기

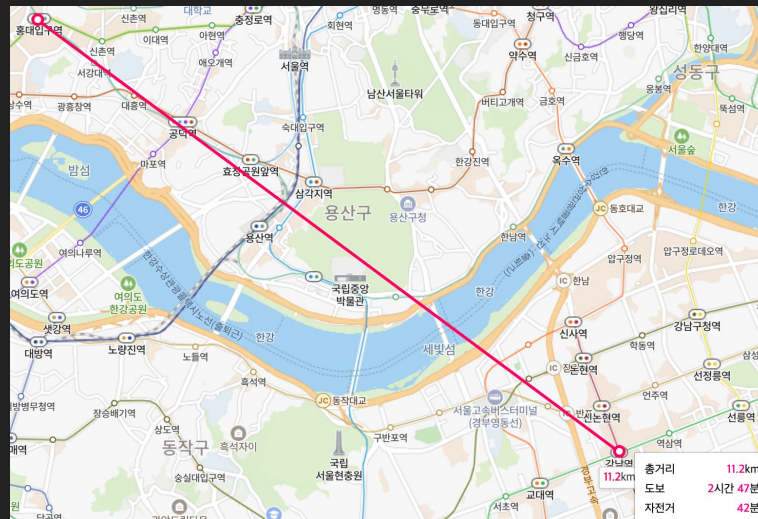
Geospatials

Geospatial Indexes 좌표를 저장하고, 검색하는 데이터 타입
거리 계산, 범위 탐색 등 지원

명령어

```
$ GEOADD seoul:station  
126.923917 37.556944 hong-dae  
127.027583 37.497928 gang-nam
```

```
$ GEODIST seoul:station hong-dae gang-nam KM
```



실전! Redis
활용

섹션 3. 데이터 타입 알아보기

Bitmaps

Bitmaps 실제 데이터 타입은 아니고, String에 binary operation을 적용한 것
최대 42억개 binary 데이터 표현 = $2^{32}(4,294,967,296)$

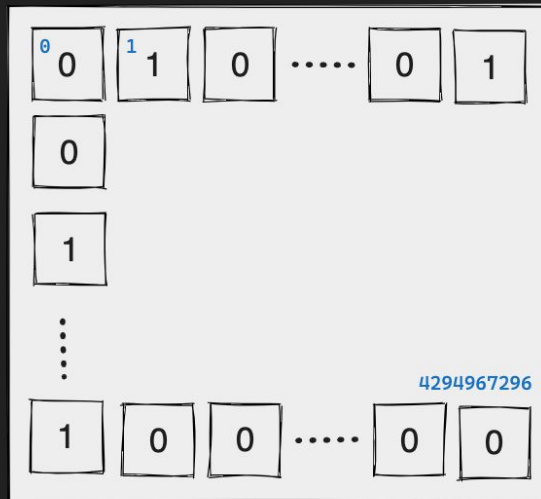
명령어

```
$ SETBIT user:log-in:23-01-01 123 1
$ SETBIT user:log-in:23-01-01 456 1
$ SETBIT user:log-in:23-01-02 123 1

$ BITCOUNT user:log-in:23-01-01

$ BITOP AND result
  user:log-in:23-01-01 user:log-in:23-01-02

$ GETBIT result 123
```



실전! Redis
활용

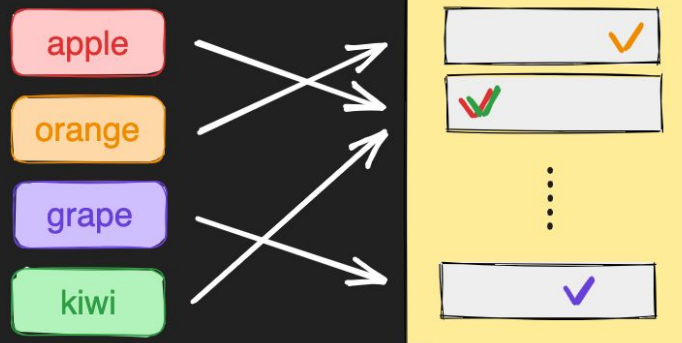
섹션 3. 데이터 타입 알아보기

HyperLogLog

HyperLogLog 집합의 **cardinality**를 추정할 수 있는 확률형 자료구조
정확성을 일부 포기하는 대신 저장공간을 효율적으로 사용(평균 에러 0.81%)

vs. Set 실제 값을 저장하지 않기 때문에 매우 적은 I/O

명령어 `$ PFADD fruits apple orange grape kiwi`
`$ PFCOUNT fruits`



실전! Redis
활용

섹션 3. 데이터 타입 알아보기

BloomFilter

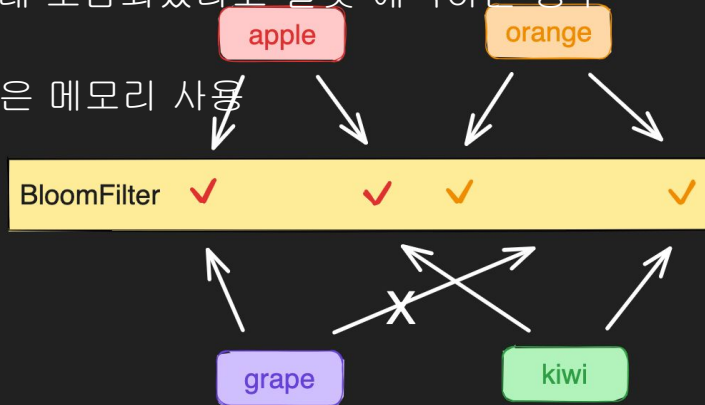
BloomFilter element가 집합 안에 포함되었는지 확인할 수 있는 확률형 자료 구조
(=membership test)

정확성을 일부 포기하는 대신 저장공간을 효율적으로 사용

false positive element가 집합에 실제로 포함되지 않는데 포함되었다고 잘못 예측하는 경우

vs. Set 실제 값을 저장하지 않기 때문에 매우 적은 메모리 사용

명령어 \$ BF.MADD fruits apple orange
 \$ BF.EXISTS fruits apple
 \$ BF.EXISTS fruits grape



실전! Redis
활용

섹션 4. Redis 특수 명령어

데이터 만료

Expiration 데이터를 특정시간 이후에 만료 시키는 기능

TTL(Time To Live) 데이터가 유효한 시간(초 단위)

특징 데이터 조회 요청시에 만료된 데이터는 조회되지 않음
데이터가 만료되자마자 삭제하지 않고, 만료로 표시했다가 백그라운드에서 주기적으로
삭제

명령어 \$ SET greeting hello
 \$ EXPIRE greeting 10
 \$ TTL greeting
 \$ GET greeting

 \$ SETEX greeting 10 hello

실전! Redis
활용

섹션 4. Redis 특수 명령어

SET NX/XX

NX 해당 **Key**가 존재하지 않는 경우에만 **SET**

XX 해당 **Key**가 이미 존재하는 경우에만 **SET**

Null Reply **SET**이 동작하지 않은 경우 (**nil**) 응답

명령어 \$ SET greeting hello NX
 \$ SET greeting hello XX

섹션 4. Redis 특수 명령어

Pub/Sub

Pub/Sub Publisher와 Subscriber가 서로 알지 못해도 통신이 가능하도록 decoupling 된 패턴
Publisher는 Subscriber에게 직접 메시지를 보내지 않고, Channel에 Publish
Subscriber는 관심이 있는 Channel을 필요에 따라 Subscribe하며 메시지 수신

vs. Stream 메시지가 보관되는 Stream과 달리 Pub/Sub은 Subscribe 하지 않을 때 발행된 메시지 수신 불가



실전! Redis
활용

섹션 4. Redis 특수 명령어

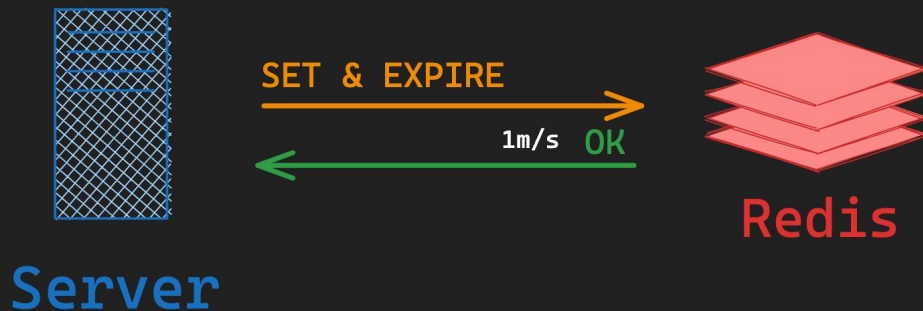
Pipeline

Pipelining

다수의 `commands`를 한 번에 요청하여 네트워크 성능을 향상 시키는 기술
Round-Trip Times 최소화
대부분의 클라이언트 라이브러리에서 지원

Round-Trip Times

Request / Response 모델에서 발생하는 네트워크 지연 시간



실전! Redis
활용

섹션 4. Redis 특수 명령어

Transaction

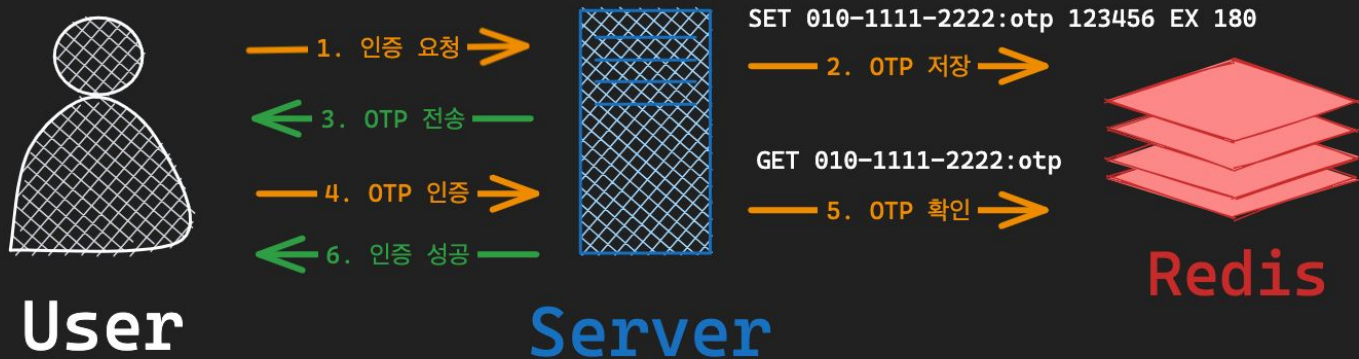
Transaction	다수의 명령을 하나의 트랜잭션으로 처리 -> 원자성 (Atomicity) 보장 중간에 에러가 발생하면 모든 작업 Rollback 하나의 트랜잭션이 처리되는 동안 다른 클라이언트의 요청이 중간에 끼어들 수 없음
Atomicity	All or Nothing / 모든 작업이 적용되거나 하나도 적용되지 않거나
vs. Pipeline	Pipeline은 네트워크 퍼포먼스 향상을 위해 여러개의 명령어를 한 번에 요청 Transcation은 작업의 원자성을 보장하기 위해 다수의 명령을 하나처럼 처리하는 기술
	Pipeline과 Transaction을 동시에 사용 가능
명령어	\$ MULTI \$ INCR foo \$ DISCARD \$ EXEC

실전! Redis
활용

섹션 5. 데이터 타입 활용

One-Time Password

One-Time Password 인증을 위해 사용되는 임시 비밀번호(e.g. 6자리 랜덤 숫자)



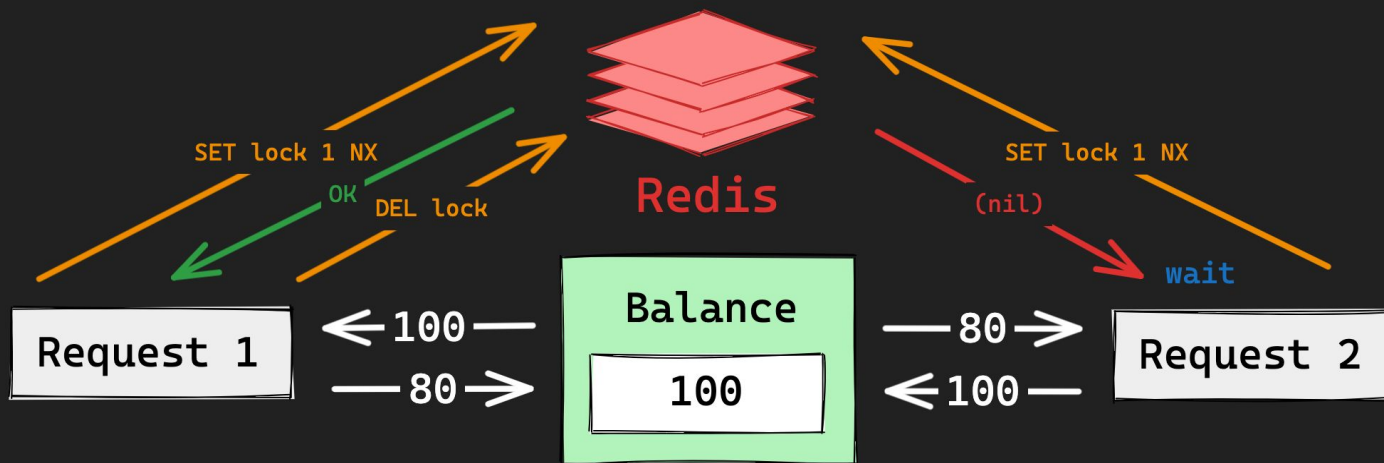
실전! Redis
활용

섹션 5. 데이터 타입 활용

Distributed Lock

Distributed Lock
해결

분산 환경의 다수의 프로세스에서 동일한 자원에 접근할 때, 동시성 문제



실전! Redis
활용

섹션 5. 데이터 타입 활용

Rate Limiter

Rate Limiter 시스템 안정성/보안을 위해 요청의 수를 제한하는 기술
IP-Based, User-Based, Application-Based, etc.

Fixed-window Rate Limiting 고정된 시간(e.g. 1분) 안에 요청 수를 제한하는 방법



실전! Redis
활용

섹션 5. 데이터 타입 활용

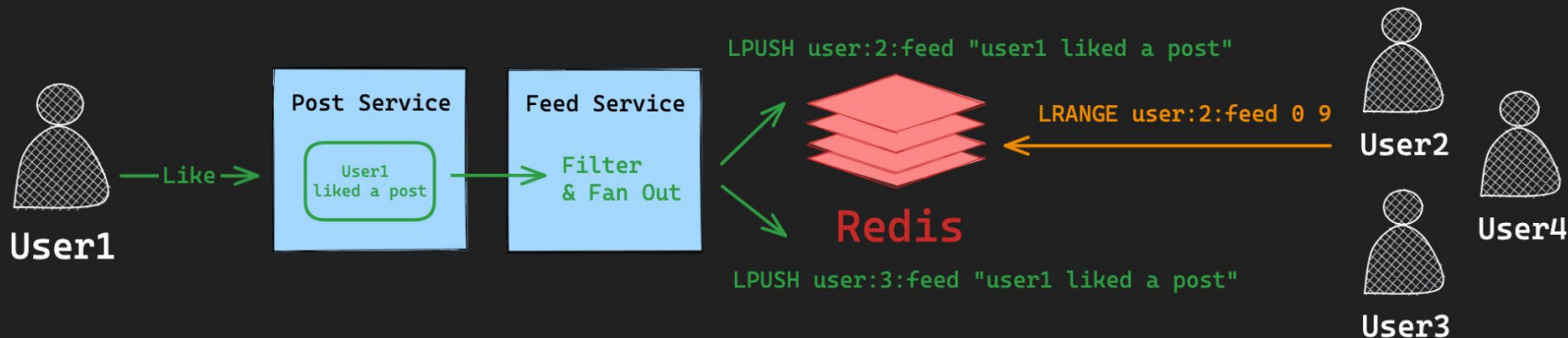
SNS Activity Feed

Activity Feed
보여주는 기능

사용자 또는 시스템과 관련된 활동이나 업데이트를 시간순으로 정렬하여

Fan-Out

단일 데이터를 한 소스에서 여러 목적지로 동시에 전달하는 메시징 패턴



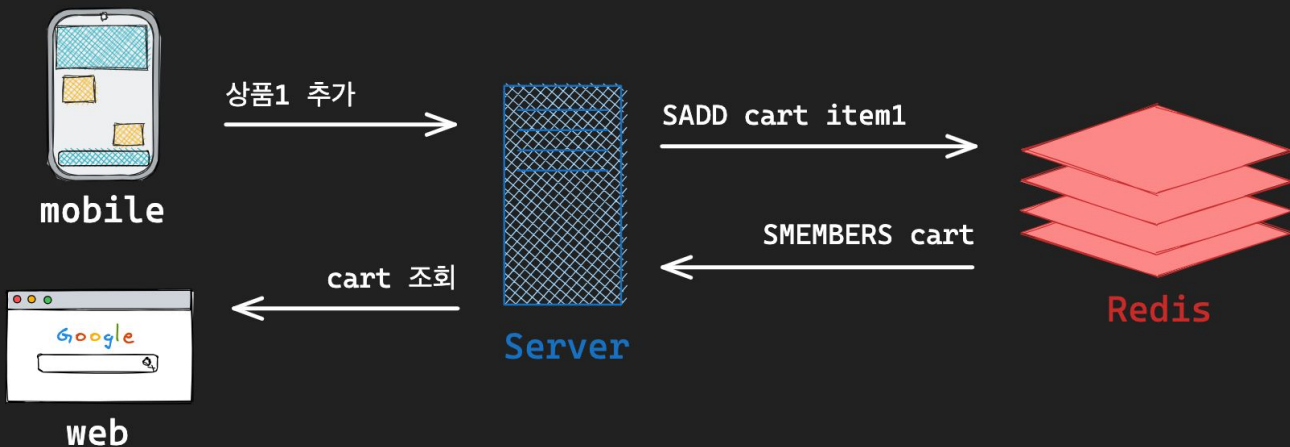
실전! Redis
활용

섹션 5. 데이터 타입 활용

Shopping Cart

Shopping Cart 사용자가 구매를 원하는 상품을 임시로 모아두는 가상의 공간

특징 수시로 변경이 발생할 수 있고, 실제 구매로 이어지지 않을 수도 있다



실전! Redis
활용

섹션 5. 데이터 타입 활용

Login Session

Login Session

사용자의 로그인 상태를 유지하기 위한 기술

동시 로그인 제한
제한

로그인시 세션의 개수를 제한하여, 동시에 로그인 가능한 디바이스 개수



실전! Redis
활용

섹션 5. 데이터 타입 활용

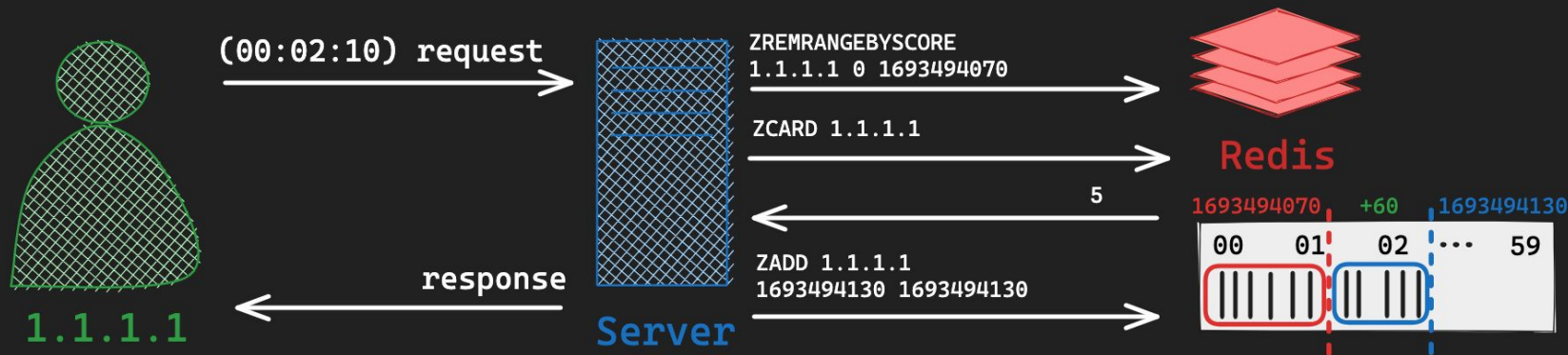
Sliding Window Rate Limiter

Sliding Window Rate Limiter
기술

시간에 따라 Window를 이동시켜 동적으로 요청수를 조절하는

vs. Fixed Window

Fixed Window는 window 시간마다 허용량이 초기화 되지만,
Sliding Window는 시간이 경과함에 따라 window가 같이 움직인다.



실전! Redis
활용

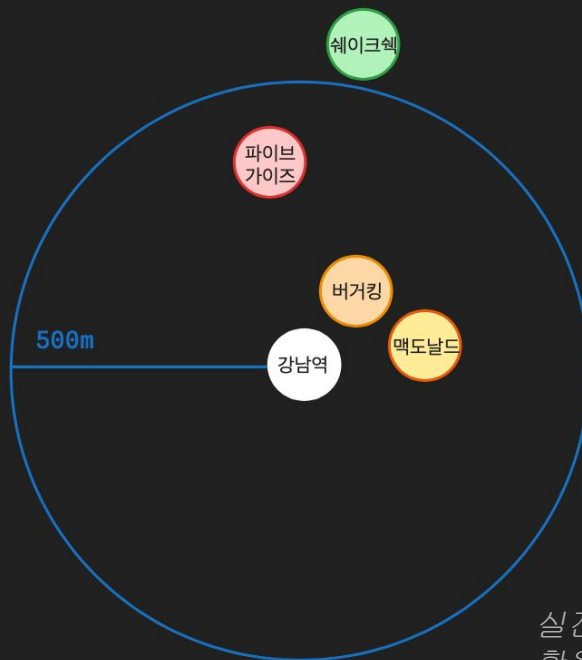
섹션 5. 데이터 타입 활용

Geofencing

Geofencing 위치를 활용하여 지도 상의 가상의 경계 또는 지리적 영역을 정의하는 기술

명령어 \$ GEOADD gang-nam:burgers
127.025705 37.501272 five-guys
127.025699 37.502775 shake-shack
127.028747 37.498668 mc-donalds
127.027531 37.498847 burger-king

\$ GEORADIUS gang-nam:burgers
127.027583 37.497928 0.5 km



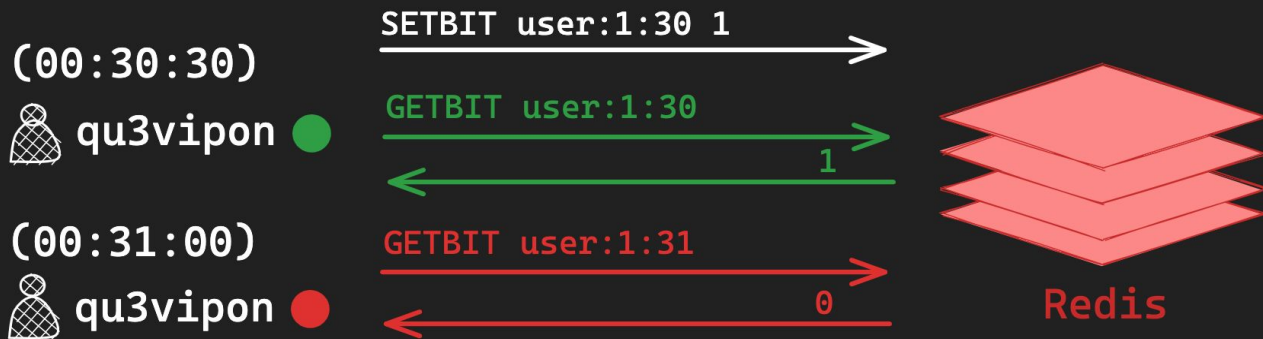
실전! Redis
활용

섹션 5. 데이터 타입 활용

Online Status

Online Status 사용자의 현재 상태를 표시하는 기능

특징 실시간성을 완벽히 보장하지는 않는다. 수시로 변경되는 값이다.



실전! Redis
활용

섹션 5. 데이터 타입 활용

Visitors Count

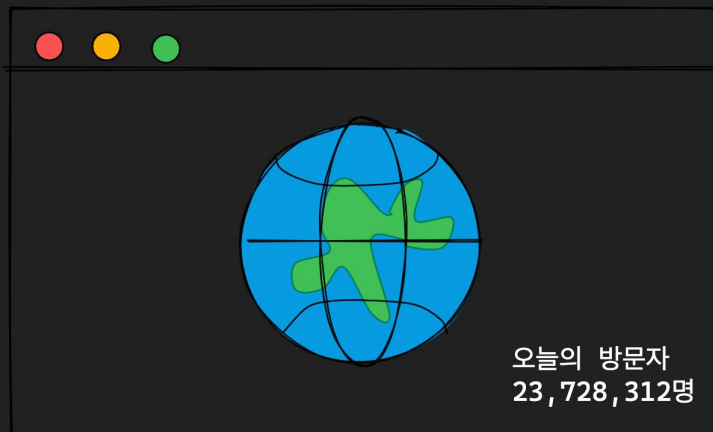
Visitors Count Approximation

경우

방문자 수(또는 특정 횟수)를 대략적으로 추정하는 경우
정확한 횟수를 셀 필요 없이 대략적인 어림치만 알고자 하는

명령어 \$ PFADD today:users
 user:1:1693494070
 user:1:1693494071
 user:2:1693494071

 \$ PFCOUNT today:users

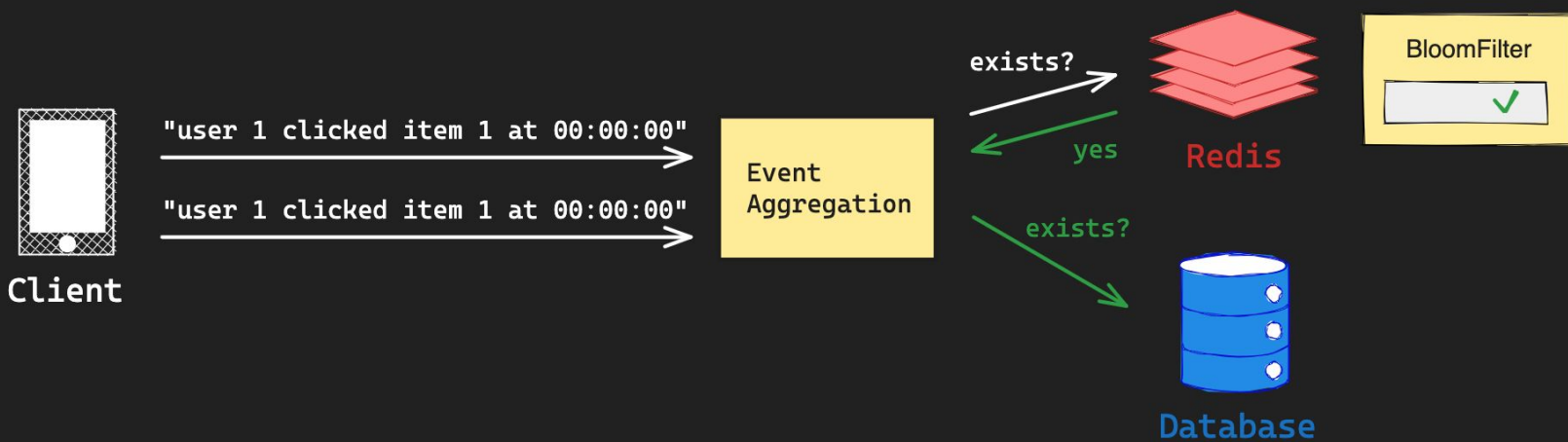


실전! Redis
활용

섹션 5. 데이터 타입 활용

Unique Events

Unique Events 동일 요청이 중복으로 처리되지 않기 위해 빠르게 해당 item이 중복인지 확인하는 방법



실전! Redis
활용

섹션 6. Redis 사용시 주의사항

O(N) 명령어

O(N) 명령어

대부분의 명령어는 O(1) 시간복잡도를 갖지만, 일부 명령어의 경우 O(N)

주의사항

Redis는 Single Thread로 명령어를 순차적으로 수행하기 때문에,
오래 걸리는 O(N) 명령어 수행시, 전체적인 어플리케이션 성능 저하

예시

KEYS

지정된 패턴과 일치하는 모든 키 key 조회
Production 환경에서 절대 사용 금지 -> SCAN 명령어로 대체

SMEMBERS

Set의 모든 member 반환(N = Set Cardinality)

HGETALL

Hash의 모든 field 반환(N = Size of Hash)

SORT

List, Set, ZSet의 item 정렬하여 반환

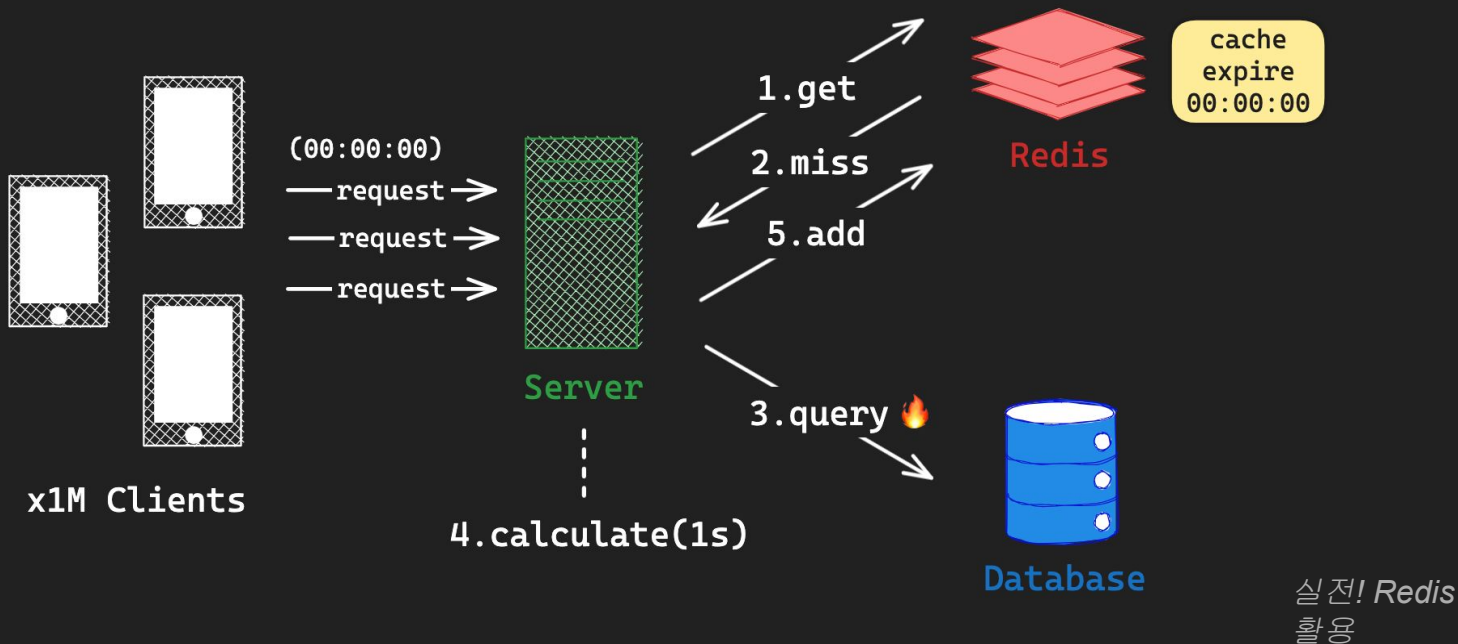
실전! Redis
활용

섹션 6. Redis 사용시 주의사항

Thundering Herd Problem

Thundering Herd

병렬 요청이 공유 자원에 대해서 접근할 때, 급격한 과부하가 발생하는 문제



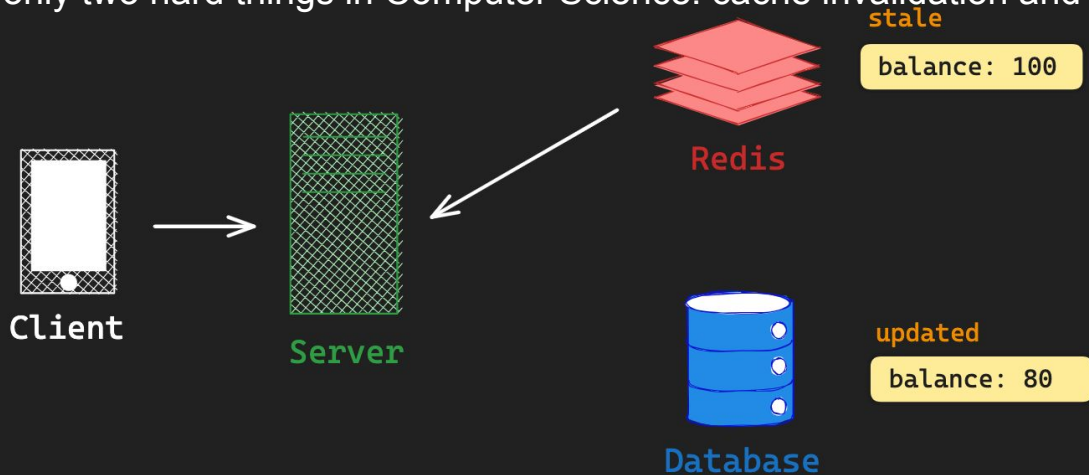
섹션 6. Redis 사용시 주의사항

Stale Cache Invalidation

Cache Invalidation 캐시의 유효성이 손실되었거나 변경되었을 때, 캐시를 변경하거나 삭제하는 기술

Two Hard Things

“There are only two hard things in Computer Science: cache invalidation and naming things”



실전! Redis
활용

강의를 마치며

실전! *Redis*
활용