홍성민 | 기술적 판단으로 끝까지 해결하는 백엔드 개발자

인적사항

2001년 9월 13일

주소 : 서울특별시 강북구 삼양로 141 길 27

학력사항

한림대학교 소프트웨어학부

4학년 2학기 (3.97 / 4.5)

자격증

- AWS Solutions Architect – Associate
- 리눅스마스터 2급
- 네트워크관리사 2급

연락처

010-2209-2664

hskhsmmm@gmail.com

https://github.com/hskhsmm

https://hskhsmm.tistory.com

대규모 트래픽 환경에서의 동시성 문제와 시스템 병목을 분석하고,
실험과 수치를 기반으로 아키텍처를 설계한 경험이 있습니다.

6인 팀 단위 프로젝트에서 기획·와이어프레임 설계부터 백엔드·인프라 구축, 성능 실험,
최종 발표까지 프로젝트 전 과정을 리딩하며 팀의 기술적 의사결정을 주도했습니다.

단순 구현에 그치지 않고, 병목 원인을 정량적으로 검증하고
트레이드오프를 명확히 한 뒤 합리적인 기술 선택을 내리는 개발을 지향합니다.

대표 프로젝트

내 맘대로 캠페인 (2025.12~ 2026.01)

소개 | Kafka와 배치를 활용한 고성능 선착순 시스템 (백엔드 2인 협업 프로젝트)

깃 허브 | https://github.com/hskhsmm/event-driven-batch-kafka-system

기술 스택 | Java 25 · Spring Boot · Redis · MySQL · AWS · Github Actions · Docker · Kafka · k6 · Spring Batch

- 대규모 동시 요청 환경에서 공정한 선착순 처리를 목표로 시스템 설계
- Redis 기반 재고 차감과 DB 배치 저장 구조로 성능 병목을 분리
- 트랜잭션 커밋 기준 TPS 측정 및 파티션 수별 성능 실험을 통해 아키텍처 의사결정 수행

[문제 정의]

- 10만 건 동시 트래픽 환경에서 MySQL의 재고 차감 로직이 Row Lock 경합을 유발하면서, Kafka 파티션 수를 증가시켜도 전체 **처리량(TPS)이 정체**되는 문제가 있었습니다.(285 -> 288)
- 파티션을 늘릴 경우 처리량은 증가했지만, 멀티 스레드 환경에서 poll 및 commit 과정이 병렬로 수행되면서 **전역적인 처리 순서가 보장되지 않는** 트레이드오프가 확인되었습니다.

[해결 과정 및 설계 판단]

1. DB 병목 해결

- 기존: MySQL에서 current_stock – 1 방식의 원자적 UPDATE 쿼리를 통해 재고를 차감하고 있었으며, 이 과정에서 다수의 트랜잭션이 동일 Row에 접근하면서 심각한 Row Lock 경합이 발생하였습니다.
- 개선: 재고 차감 로직을 **Redis Lua 스크립트로 분리**하여 인메모리에서 원자적으로 처리하도록 변경하고, MySQL은 최종 결과를 저장하는 역할만 수행하도록 책임을 재구성하였습니다.
- 결과: 파티션 확장 시 TPS가 선형적으로 증가할 수 있는 구조를 확보했습니다.
  Kafka 메시지 소비속도는 2000TPS, 비즈니스 무결성이 완료된 **DB 트랜잭션 커밋 기준의 TPS는 285에서 595**로 약 **2.1배 증가**하였습니다.

2. 순서 보장과 처리량 간 트레이드오프 정량화

- 파티션 1·3·5개 환경에서 **k6 부하 테스트**를 수행하여 TPS와 switch_ratio(파티션 전환율)를 측정하고, 파티션 수 증가에 따른 **CPU 부하 시점과 부하 지속 시간**을 비교 분석했습니다.
- 파티션 수를 늘려도 TPS가 유의미하게 증가하지 않고 CPU 부하 지속 시간이 오히려 증가하는 현상을 통해, 시스템 병목이 Kafka가 아닌 DB 레이어에 있음을 **정량적으로 확인**했습니다.

3. 장애 격리 및 안정성 확보

- 메시지 처리 실패가 전체 파이프라인에 영향을 주지 않도록 **Dead Letter Queue(DLQ)를 구현**하여 **실패 이벤트를 별도로 격리**했습니다.
- 실시간 처리 로직과 통계 연산 간의 부하 간섭을 줄이기 위해 Spring Batch 기반 일일 집계 작업을 분리 구성했습니다. 단순 @Scheduled로도 가능했지만 배치 실행 이력 자동 기록과 DB 장애 시 재시도 정책을 확인하여 안정성을 확보하고자 Spring Batch를 선택했습니다.
- 결과적으로 10만 건 부하 테스트 환경에서 재고 초과 판매 없이 **데이터 정합성을 안정적으로 유지**할 수 있었습니다.

Way To Earth (2025.08 ~ 2025.12)

소개 | 러닝 기록을 가상 여정·커뮤니티로 확장한 소셜 러닝 플랫폼 (팀장 / 캡스톤 3인 팀 프로젝트)

깃 허브 | https://github.com/WayToEarth-Team/WayToEarth_BE

프로젝트 홍보영상 | https://www.youtube.com/watch?v=e45tMPZ9_9M

기술 스택 | Java 21 · Kotlin · Spring Boot · Spring Security · JPA · QueryDSL · Redis · MariaDB · WebSocket · AWS · Docker · GitHub Actions · Firebase(FCM)

- 달린 거리로 가상 여정을 여행하고 크루와 함께 성장하는 게임화된 러닝 플랫폼
- 27개 테이블 규모의 복합 도메인 설계 및 DB·인프라 아키텍처 총괄
- 러닝, 실시간 크루 채팅, 랭킹 시스템, AI 러닝 코칭 등 핵심 기능 구현

[문제 정의 및 해결]

1. JWT 로그아웃 시 토큰 무효화 불가

- 문제: JWT는 Stateless 특성상 서버에서 토큰을 강제 만료시킬 수 없어, 로그아웃 후에도 탈취된 토큰으로 API 접근이 가능한 보안 취약점이 존재했습니다.
- 해결: Redis 기반 토큰 블랙리스트를 구현하여 로그아웃 시 액세스 토큰을 블랙리스트에 등록하고, 모든 API 요청 시 블랙리스트 여부를 검증하도록 처리했습니다.
  또한 토큰의 만료 시간을 TTL로 설정하여 만료된 토큰은 자동 삭제되도록 구성, 메모리 효율성을 확보하였습니다.
- 결과: 로그아웃 이후 토큰 재사용을 차단하여 인증 흐름의 보안성을 강화했으며 추가적인 DB 조회 없이 Redis 기반 검증만으로 보안 요구사항을 충족하였습니다.

2. 러닝 세션 동시 수정으로 인한 데이터 정합성 문제

- 문제: 러닝 일시정지/재개/완료 요청이 동시에 들어올 경우, 세션 상태가 비정상적으로 변경되는 문제가 발생하였습니다.
- 해결: findBySessionIdWithLock 메서드에 Pessimistic Locking(SELECT ... FOR UPDATE)을 적용하여 동시 요청 시 순차 처리를 보장했습니다.
  또한 러닝 완료 시 사용자 통계 업데이트는 updateRunningStatsAtomic 메서드로 원자적 Update 쿼리를 사용하여 동시성 문제를 해결하였습니다.
- 결과: 동시 요청 환경에서도 러닝 세션 상태와 사용자 통계가 일관되게 유지되었으며 중복 완료나 잘못된 상태 전이로 인한 데이터 오류를 방지할 수 있었습니다.

3. 회원 탈퇴 시 외래 키 제약 조건으로 인한 삭제 실패 문제

- 문제: 크루장 시스템을 도입하면서 크루 MVP 및 가입 신청 처리자가 사용자를 외래 키로 참조하고 있어, 회원 탈퇴 시 다수의 외래 키 제약 조건 에러가 발생했습니다. 참조 필드를 NULL로 설정하여 삭제를 가능하게 했으나, 이는 임시방편에 불과하다고 판단했습니다.
- 해결: 도메인 관점에서 승인/거절 상태의 가입 신청은 더 이상 유지할 수 없다고 판단하여 승인/거부 시 가입 신청 데이터를 즉시 삭제하도록 설계를 변경하였습니다.
- 결과: 불필요한 참조 자체를 제거함으로써 회원 탈퇴 시 외래 키 제약 문제를 원천 차단했고, 데이터 무결성과 DB 관리 측면 모두에서 안정적인 구조로 개선하였습니다.

4. S3 Presigned URL 만료로 인한 이미지 접근 불가 문제

- 문제: 이미지 파일을 S3에 저장하고 Presigned URL로 조회하는 구조에서, URL 만료로 인해 일정 시간이 지나면 프로필·피드·크루 이미지가 깨지는 문제가 발생했습니다.
- 해결: CloudFront를 도입하여 S3를 Origin으로 설정하고, 이미지 조회는 CloudFront CDN URL 업로드만 Presigned URL을 사용하는 구조로 분리했습니다. S3는 OAC(Origin Access Control) 방식으로 비공개 유지하여 보안을 강화했습니다.
- 결과: 이미지 만료 문제를 근본적으로 해결했으며, **CDN 캐싱을 통해 이미지 응답 속도 개선과 전송 비용 절감 효과**를 함께 확보했습니다.

기타 프로젝트

MOVA (2025.10 ~ 2025.11)

소개 | AI 기반 맞춤형 영화 큐레이션 및 커뮤니티 서비스 (백엔드 리드 / 3인 팀 프로젝트)
깃 허브 | https://github.com/MOVA-Team/MOVA-BE
기술 스택 | NestJS · TypeScript · MongoDB · TMDB API

- NestJS + TypeScript 기반 RESTful 백엔드 API 개발

  (모듈화된 구조를 통해 짧은 기간 내 팀 단위 병렬 개발 및 유지보수성 확보)
- MongoDB 기반 데이터 모델링 및 API 구현

  (영화 메타데이터·사용자 활동 로그 등 비정형 데이터와 잦은 스키마 변경에 대응하기 위해 선택)
- 외부 영화 데이터인 TMDB API 연동을 통한 영화 검색·상세·출연진 정보 제공
- JWT 기반 인증과 북마크, 리뷰, 커뮤니티, 개인화 추천 기능 구현

HabiGlow (2025.08 ~ 2025.09)

소개 | 실패 경험까지 기록하는 공감 기반 루틴 관리 서비스(팀장 / 6인 팀 프로젝트)
깃 허브 | https://github.com/9oormthon-univ/2025_SEASONTHON_TEAM_90_BE
기술 스택 | Java 21 · Spring Boot · Spring Security · JPA · PostgreSQL · Docker · Firebase(FCM) · OpenAI API

- Spring Boot 기반 RESTful 백엔드 API 설계
- strategy 패턴을 활용한 OAuth2 소셜 로그인 구현 (Google, Naver, Kakao)
- OpenAI GPT-4o-mini 연동을 통한 AI 기반 주간 회고 피드백 기능 구현
- Firebase Cloud Messaging + @Scheduled를 활용한 일일 푸시 알림 배치 처리
- Guava RateLimiter 기반 API Rate Limiting 적용
- 복잡한 루틴 관리 로직을 파사드 패턴(RoutineManagementFacade)으로 통합하여 컨트롤러-서비스 간 의존성 단순화

HOW I WORK

Communication

- API 명세와 업무 정의를 Notion, Swagger로 문서화하여 개발 · 기획 간 커뮤니케이션 비용을 줄이고 협업 효율을 높였습니다.
- ApiResponse<T> 공통 응답 객체와 @RestControllerAdvice 기반 전역 예외 처리를 구성하여, 모든 API가 {success, message, errorCode, data} 형식으로 통일된 응답을 반환하도록 했습니다. 이를 통해 프론트엔드와의 에러 처리 협업을 단순화했습니다.
- 기술 규칙 및 업무 프로세스를 체계적으로 문서화하며 팀 내 지식 공유를 주도했고, 이에 대해 긍정적인 동료 평가를 받은 경험이 있습니다.
- 데일리 스크럼 문화를 주도하여 팀원들이 당일 진행 상황과 이슈를 공유할 수 있는 환경을 조성했습니다.

Code Review

- 모든 코드 변경 사항은 GitHub PR을 통해 공유하고, 최소 1명 이상의 리뷰어 승인(Approve)을 Merge 필수 조건으로 설정하여 팀 내 코드 품질 상향 평준화를 주도했습니다.
- 변수명 등 사소한 스타일 수정보다는 N+1 문제 사전 차단, 비효율적인 인덱스 조회 방지, 보안 취약점 점검 등 구조와 성능 위주의 피드백을 우선시하며 프로젝트 완성도를 높였습니다.

Scalability & Data Integrity Synergy

- 월 450억 건의 방대한 데이터를 처리하는 에어브릿지의 에어브릿지 환경에서, 대규모 트래픽이 시스템에 주는 영향을 직접 목격하고 배우고 싶습니다. 저는 10만 건 트래픽 실험을 통해 **Kafka 소비 속도와 DB 커밋 속도의 괴리를 분석하며 기술적 트레이드오프를 체감**한 바 있습니다.
- 단순 구현에 그치지 않고 **아키텍처 개선으로 TPS를 2.1배 향상시켰던 경험**은 저에게 '수치 기반 의사결정'의 중요성을 일깨워주었습니다. 이러한 분석 습관을 바탕으로, 에어브릿지의 고가용성 파이프라인이 어떻게 유지되는지 현업 선배님들 곁에서 깊이 있게 배우고 싶습니다.
- '매체 연동 10x' 및 'AI 마이그레이션 10x' 프로젝트에서 **Airflow 파이프라인 개선과 데이터 정합성 확보** 과정을 함께하며, 제가 배운 기본기가 실무의 복잡한 문제를 해결하는 데 어떻게 쓰일 수 있는지 증명하고 싶습니다.

외부활동 및 수상

- 구름톤 유니브 4 기 (2025.03 ~ 2025.11)
- 교내 알고리즘 스터디 '프루빗' 1 기 (2025.01 ~ 2025.12)
- SeSAC: AWS 클라우드 아키텍트 플러스 (2023.12 ~ 2024.05)
- 멋쟁이사자처럼 11 기 (2023.03 ~ 2023.12)
- 2025-2 한림대학교 캡스톤디자인 수상