

정수범.

(SooBeom Jeong)

BackEnd Developer

“ 기술적 견고함,
비즈니스 가치를
동시에 고민하는,
'기획하는 개발자'
정수범입니다. ”

Mail

snowdrop6342@gmail.com

Github

<https://github.com/Jsim6342>

Introduce

[여러 선택지를 고려하여 개발 방향을 주도적으로 기획합니다]

단순히 코드를 작성하는 '코더'를 넘어, 서비스 확장성, 구현 속도 사이에서 개발 방향을 기획합니다.

- 흩어져 있는 서비스들의 트래픽을 분산 및 통합하여 관리할 수 있는 이벤트 아키텍처 설계 및 개발 주도
- 사내 코어 라이브러리를 구축하며 불필요한 반복 작업을 제거, 팀 전체의 생산성 향상

[원인 파악에 능하며, 안정적인 개발을 추구합니다]

서비스에서 발생하는 복잡한 에러 이슈와 병목 지점을 빠르게 파악하고 해결하는 데 강점이 있습니다.

- Redis 메모리 캐시 문제를 빠르게 파악, Kafka DLQ를 활용해 손실된 통계 데이터를 100% 복구
- 비효율 스케줄러로 인한 CPU 급증 현상을 이벤트 리스너 방식으로 개선하여 사용량을 30% 이상 절감

[기술의 끝은 결국 '수익화'와 '사용자 가치'임을 이해합니다]

과거 기획자, 마케팅 프리랜서, 그리고 국비 지원 백엔드 강사로서 쌓아온 폭넓은 경험은 다른 개발자 대비 저만의 차별화된 무기입니다. 덕분에 기획과 마케팅의 의도를 정확히 파악하며, 타 직군과의 원활한 커뮤니케이션을 통해 프로젝트가 실질적인 비즈니스 성과로 이어지도록 기여합니다.

- 단순 트래픽 분산 요구사항을 사내 확장성과 장기 유지보수를 고려한 MSA 구조로 재설계 제안 및 개발 주도
- 인프런 코딩테스트 강의 기획·마케팅 전담, 수강생 900명 유치. 기술 외 비즈니스 영역에서의 직접적인 성과 보유

Skills

- Backend: Java, Spring, JPA, QueryDSL, MyBatis, JUnit5
- Database: MySQL, MSSQL, Redis
- Messaging & Infra: Kafka / AWS(EC2, RDS, S3 등) / Kubernetes, Docker / Jenkins, ArgoCD
- Frontend: Vue.js, React (기본 페이지 구현 및 AI 활용 서비스 수준)

Careers

- 메가스터디(24.02~재직 중): 이벤트 아키텍처 설계 및 개발, 단축 URL 서비스 개발, 사내 라이브러리 개발, Ebook 서비스 개발, 사내 Java Spring 템플릿 구축, Kubernetes 기반 인프라 운영 및 개선
- 소프트스퀘어드(22.08~24.02): 외주 서비스(애플리 앱) 서버 개발, 롯데카드 사내 서비스 개발 및 백엔드 강의 진행

Project Solution Summary

- 이벤트 중계 MSA 아키텍처 설계를 주도 및 개발. 이를 통해 사내 여러 사업부들의 공통 기능(알림, 파일, 인증, 결제 등)들을 공통 서비스로 묶고, 다양한 트래픽 이슈 등을 효율적으로 분산 처리할 수 있는 기반 구축
- 사내에서 공통 라이브러리(인증, 로깅 기능) 개발을 통한 개발 시간 단축 및 이슈 분석 시간 단축
- Redis 메모리 풀 이슈 원인 분석 및 캐시 구조 개선. 메모리 사용량·CPU·네트워크 트래픽 모두 현저히 감소
- Kafka 메시지 복원 로직을 통해 운영 중 누락된 통계 데이터 6일치 100% 복구, 장애 대응 역량 입증
- DB Buffer 캐시 설정 최적화를 통해 100만 건 통계 페이지 조회 속도 10초 → 1초 이하로 개선 (10배 향상)
- 알림 스케줄러를 이벤트 기반 리스너로 개선하여 서버 CPU 사용률 30% 이상 감소
- AOS 단말 중복 요청 방지 로직 및 유니크 키 적용으로 중복 데이터 저장 및 CS 이슈 제거

Experience & Awards

- SKT에서 주관하는 T아카데미의 ASAC 과정 교육 커리큘럼 개발 및 백엔드 전담 강사 [Link]
- 유데미 X 소프트스퀘어드에서 주관하는 라이징캠프 교육 서비스 강사
- 900명 이상이 수강중인 인프런 코딩테스트 강의 기획 & 운영 [Link]
- 공기업 aT 주관 빅데이터 해커톤 '대상' (서비스 개발 및 MVP 구현 부문) [Link]
- 개인 SNS, 숨고, 외주 채널 등에서 백엔드 개발, 자동화 프로그램 등 다양한 외주 개발 경험

Education

- 정보처리기사 (2022.11 취득)
- 소프트스퀘어드, 라이징캠프 부트캠프 2기 (2021.09 ~ 2021.11, 우수 수료)
- 빅데이터 기반 SW 개발자 국비 교육과정 (2020.06 ~ 2020.11, 우수 수료)

이벤트 아키텍처 설계 및 개발(25.12 ~ 진행중)

사내 여러 사업부의 대용량 트래픽 분산 및 공통 기능 통합을 위한 이벤트 기반 MSA 아키텍처를
주도적으로 설계 및 개발

사용 기술: Spring Eureka, Spring Gateway, Kafka, Redis, CDC, MSSQL

이벤트 아키텍처 설계

[개발 배경]

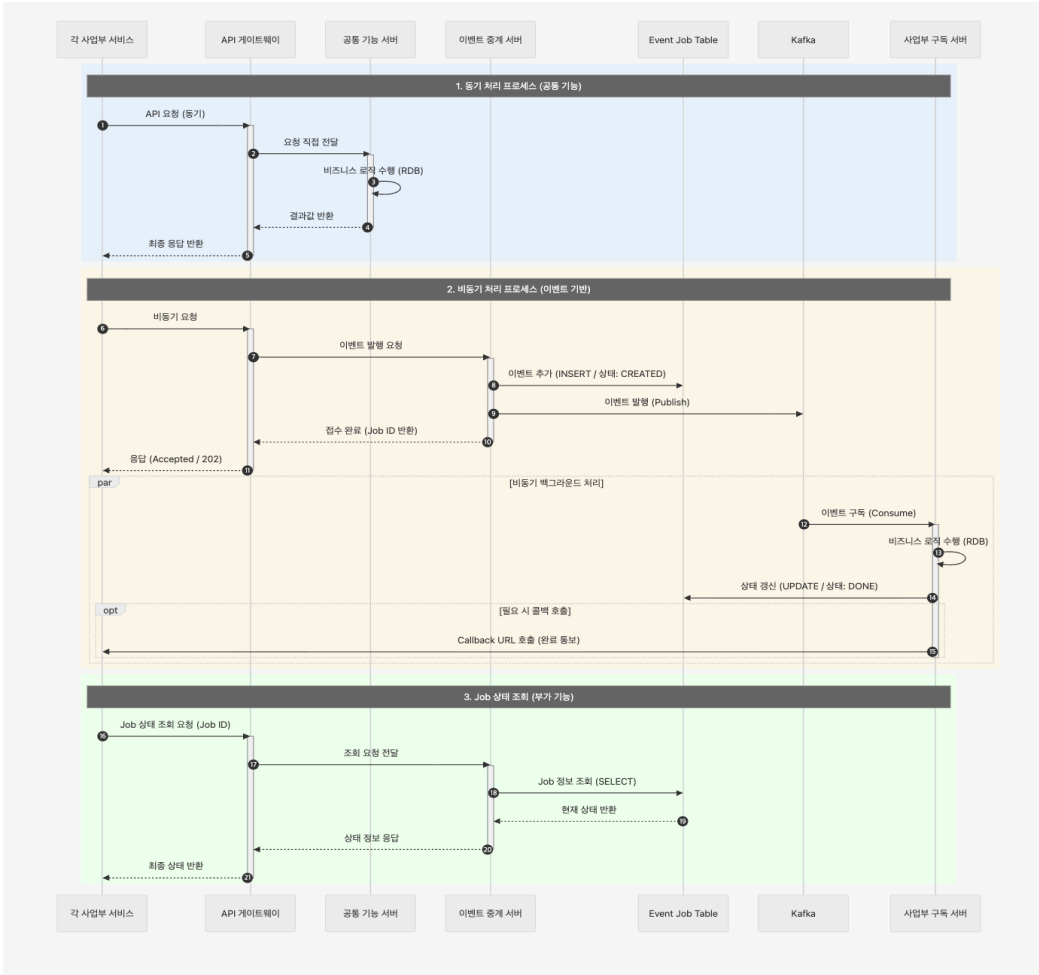
- ASP, Java, Node 등 다양한 기술 스택이 혼재하는 사내 환경에서, 사업부 마다 공통 기능(알림, 인증, 결제 등)을 반복 개발하는 중복 공수 문제 발생
- 고등 사업부의 경우 대용량 트래픽 발생 시 데드락 등 서버 장애가 빈번히 발생하는 상황

[개발 내용]

- 공통 기능을 별도 서버로 분리. 알림·파일은 비동기, 인증·결제는 동기 방식으로 구분 처리
- 다양한 기술 스택이 범용적으로 사용할 수 있도록 이벤트 중계 서버 설계. 공통 API 스키마 기반의 비동기 요청으로 트래픽 분산
- 기존 사업부 DB 공유가 필요한 경우 Interface 테이블 + CDC 기반 CQRS 패턴 적용으로 조회 성능 향상 및 트래픽 분산
- Circuit Breaker, DLQ, ShedLock, Prometheus/Grafana를 활용한 안정적인 운영 및 모니터링 환경 구축. 장애 발생 시 즉각 알림 및 복구 체계 마련

[결과]

- 모의고사 시즌 수강 이력 API 트래픽 급증으로 반복적으로 다운되던 고등 서버 이슈를 이벤트 중계 서버로 트래픽 분산하여 해소
- 실제 데드락으로 미처리된 메시지가 DLQ에 적재된 케이스에서 빠른 복구 성공
- 범용적인 구조 덕분에 여러 사업부에서 공통 기능 통합 및 트래픽 분산 목적으로 활용 가능한 사내 인프라 제공



단축 URL 서비스 개발 리드 (25.6 ~ 25.12)

Bitly를 대체하기 위한 사내 전용 단축 URL 서비스 개발

사용 기술: Java 17, Spring Boot 3.0, MyBatis, MSSQL, Redis, Kafka, Docker, Kubernetes

스프링 이벤트, 비동기 처리를 활용한 트래픽 고려 개발

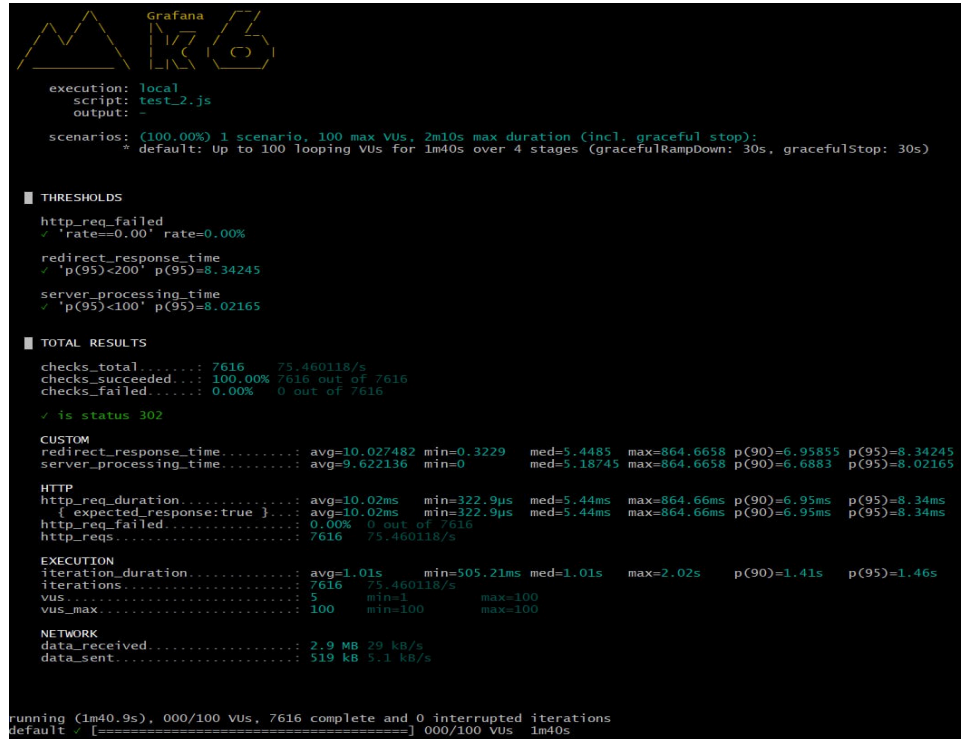
[개발 내용]

- 트래픽 병목 지점인 클릭 통계 저장에 Spring Event + @Async 적용. 리다이렉트는 동기, 통계 저장은 비동기로 분리하여 성능과 유연성 확보

- 과거 최대 일일 이용 건수(3,000건) 기준 K6 스트레스 테스트 진행 및 비동기 스레드풀 튜닝

[결과]

- 에러율 0%, 평균 응답속도 10ms, 처리량 75.4 req/s — 0.008초 내 리다이렉트 보장



Redis 성능 이슈 파악 및 개선

[문제]

- Redis 캐시보다 RDB 조회가 더 빠른 현상 발생

[원인 분석]

- Redis 스토리지가 사내 NFS 원격 스토리지를 사용 중이어서 네트워크 지연 발생

[해결]

- Redis를 메모리 스토리지로 변경, 배치성 데이터는 Kafka 구독 방식으로 전환하여 Redis는 순수 캐시 용도로만 유지

[결과]

- 쓰기 약 8배, 읽기 약 6배 성능 향상

데이터 기반 비교 분석			
비교 항목	스토리지 연결 (NFS)	스토리지 미연결 (Memory Only)	차이 (배수)
IP 주소			-
SET (쓰기)	5,745 req/sec	45,495 req/sec	약 7.9배 차이 🚀
GET (읽기)	6,585 req/sec	40,966 req/sec	약 6.2배 차이 🚀
안정성 (Avg Latency)	4ms ~ 41ms (들쭉날쭉)	0.7ms (매우 일정)	불안정함

사내 코어 라이브러리 개발 (25.2 ~ 25.7)

사내 Spring 기반 서비스에 공통 적용할 인증·로깅 라이브러리 개발
사용 기술: Java 17, Spring Boot 3.0, JWT, Spring Security

공통 기능(로깅, 인증) 개발

[개발 내용]

- 인증·인가 필터를 라이브러리로 제공하되, 빈 오버라이딩 및 인터페이스 확장으로 프로젝트별 커스텀 가능하도록 설계
- 요청마다 requestId 부여로 로그 추적 용이화, yaml 설정 기반으로 인증 방식(session/header/jwt)·민감정보 마스킹 등 환경별 커스터마이징 지원

[결과]

- 사내 Spring 프로젝트 10개 이상에 적용. 반복 개발 제거 및 이슈 분석 시간 단축, 개인정보 보호 기준 강화

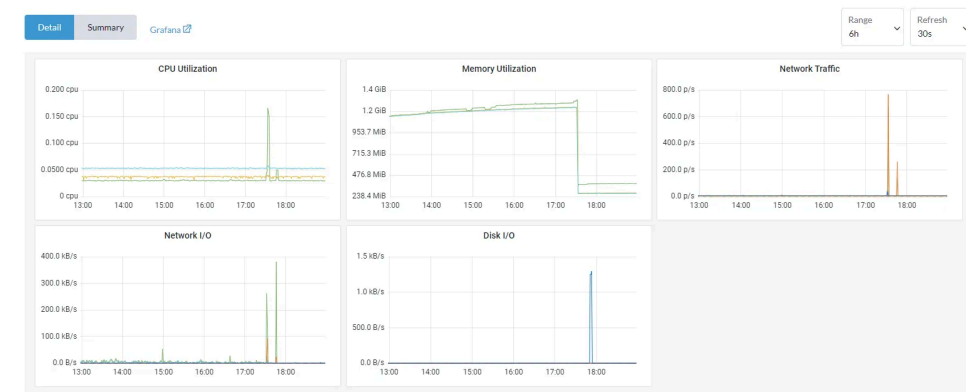
Ebook 서버 개발 & 유지보수 (24.2 ~ 25.5)

신사업 Ebook 서비스의 관리자·API·쇼핑몰 서버 3종 개발 및 운영

사용 기술: Java 17, Spring Boot 3.0, MyBatis, MySQL, Redis, Kafka, Docker, Kubernetes

Redis Memory Full 이슈 해결

- **문제:** 서비스 오픈 다음날 Redis Connection Timeout 반복 발생, Redis 관련 기능 전체 장애
- **원인 분석:** 프로모션 캐시 키가 중복 생성되어 약 500MB를 점유 (키 1개당 131,500 byte × 4,000개)
- **해결:** 프로모션 캐시를 단일 키로 변경, 기존 중복 데이터 삭제
- **결과:** Redis 메모리·CPU·네트워크 트래픽 현저히 감소



Kafka 통계 데이터 누락 이슈 해결

- **문제:** 쿼리 오작상으로 운영 서버 통계 데이터 6일치 유실
- **원인 파악:** Consumer에서 DB INSERT 쿼리 오작상으로 데이터 유실 확인
- **해결:** 평소 Exception 발생 시 별도 Error Topic에 메시지를 적재해두던 구조 활용. 임시 Consumer를 생성하고 offset을 유실 시점으로 초기화하여 데이터 재삽입
- **결과:** 약 1분 내 6일치 데이터 100% 복구

통계 데이터 조회 성능 개선

- **문제:** 주간 앱 실행 통계(50~100만 건) 조회에 10초 이상 소요
- **원인 파악:** innodb_buffer_pool_size가 기본값으로 설정되어 있어 캐싱 사이즈가 매우 작은 상태였음
- **해결:** innodb_buffer_pool_size 기본값 상향 조정
- **결과:** 조회 시간 10초 → 1초 이하 (10배 향상)

애플리 서버 개발 & 유지보수 (22.08 ~ 22.12)

누적 사용자 약 10만 명의 헬스케어 앱 백엔드 개발

사용 기술: Java 8, Spring Boot, MySQL, JPA, QueryDSL, AWS(EC2, RDS, S3)

주요 이슈 해결 정리

- **CPU 급증 개선:** 2분 간격 알림 스케줄러를 EntityListeners 기반 이벤트 리스너로 전환 → CPU 사용량 30% 이상 감소

- **중복 요청 방지:** AOS 네트워크 불안정으로 인한 중복 POST 요청 문제를 DB 유니크 키 + 예외 처리로 해결, 관련 CS 이슈 전체 제거

- **데드락 및 커넥션 풀 관리:** 사용자 급증(2만→10만명)에 따른 Too Many Connection·데드락 이슈를 RDS 커넥션 풀 조정 및 트랜잭션 호출 순서 정렬로 해결