

Email | choi20014830@gmail.com

Github | <https://github.com/barabobBOB>

Phone | 010-4830-6738

Blog | <https://velog.io/@chltpdus48>

최세연

Summary

- 동시 1만대 이상의 IoT 장비로 부터 데이터를 실시간으로 수집하는 모니터링 서비스를 개발한 경험이 있습니다.
- Spring Boot, Django, Go를 활용해 서버를 개발하고 GCP, AWS를 통해 배포할 수 있습니다.
- MySQL, PostgreSQL에서 Explain를 이용해 쿼리를 분석하고 쿼리튜닝을 진행해 성능 개선한 경험이 있습니다.
- End-to-End 프로젝트를 통해 Backend, Frontend, DevOps까지 이해하고 있는 개발자입니다.

Experience

유니로보틱스 2022.03 - 2022.07, Server Engineer / 인턴

Okomo 2022.03 - 2022.07 / 프로젝트 인원 6명

약 1만대의 IoT 장비에서 데이터를 수집하는 파이프라인을 구축하고 모니터링을 제공하는 프로젝트를 진행하였습니다.

사용 기술: Spring Boot, Spring Data JPA, QueryDSL, JUnit5, Mockito, MySQL, GCP(Compute Engine, BigQuery, Pub/Sub), Gitlab CI, Jacoco, Docker, Prometheus, Grafana, k6

대표 문제 해결 성과

개요

10초에 한 번씩 데이터를 전송하는 가상 IoT 장비 1천대와 가상 유저 100명을 바탕으로 K6를 활용해 모니터링 딜레이 테스트를 진행하였습니다. 서버에서 데이터를 수신한 시점부터 사용자 조회 시점까지 10초 이상의 딜레이가 발생하는 문제를 확인하였습니다. GCP Cloud Monitoring 트레이싱 분석 결과, 권한과 조건에 따라 IoT 장비 고유 번호를 조회하는 로직 중에 Slow Query 문제가 발생함을 알게 되었습니다.

해결 과정

1. N+1 쿼리 문제 해결: Fetch Join 적용

관련 엔티티를 한 번의 쿼리로 조회할 수 있도록 Fetch Join을 적용하여 N+1 쿼리 문제를 해결하였습니다.

2. 쿼리 최적화: Explain 분석 및 복합 인덱스 적용

JPA에서 실행 중인 쿼리를 분석하기 위해 SQL의 Explain 함수를 사용하여 실행 계획을 검토했습니다. 이를 통해 cost가 높은 테이블을 식별하고, 해당 테이블의 카디널리티를 분석하여 복합 인덱스를 적용하였습니다.

결과: API Latency 2s → 600ms

3. BigQuery 테이블 파티셔닝 정책 변경: Monthly에서 Daily로

추가적으로, BigQuery 내 IoT 센서 데이터 테이블의 파티션 정책이 월별(Monthly)로 설정되어 있음을 발견했습니다. 이는 데이터 검색 시 검색 범위가 넓어져 성능 저하를 일으키는 주요 원인 중 하나였습니다. 따라서, 파티션 정책을 일별(Daily)로 변경하여 검색 대상 row의 수를 줄이고 쿼리 성능을 극대화하였습니다.

결과: API Latency 600ms → 200ms

기여

- IoT 센서 데이터 시각화 API 개발하였습니다.
- BigQuery와 GCP Pub/Sub을 활용한 IoT 센서 디바이스 데이터 수집 파이프라인을 개발하였습니다.
- JaCoCo를 사용해 코드 커버리지를 분석하여 테스트의 안정성을 보장하는 CI와 GitLab CI와 Docker을 활용해 브랜치의 변경 사항이 자동으로 서버에 반영되는 CI/CD 파이프라인을 구축하였습니다.
- 코드 신뢰성을 위해 JUnit5, Mockito를 이용한 단위테스트와 TestContainer를 활용한 통합테스트를 작성하였습니다.

Project

이웃집명냥 2023.11 - 진행중 / 프로젝트 인원 6명 [Github Link](#) [API Link](#)

반려동물 돌봄 커뮤니티 앱 서비스이며, 서비스 런칭 예정입니다.

사용 기술: Go, PostgreSQL, Firebase, Fly.io, Github Actions

문제 해결

개요

시스템의 트랜잭션 관리 로직이 저장소(repository) 계층에만 관리되고 있었습니다. 이로 인해 서비스(service) 계층에서 처리 중인 작업이 실패할 경우, 이미 실행된 데이터베이스 작업에 대한 rollback이 불가능한 문제가 발생했습니다. 이는 데이터 일관성을 보장할 수 없는 심각한 문제로, 시스템의 안정성과 신뢰성에 큰 문제가 되었습니다.

해결 과정

데이터베이스 트랜잭션을 안전하게 관리하기 위해, 컨텍스트 관리와 콜백 패턴을 사용한 범용 트랜잭션 실행 함수를 구현하였습니다. 데이터베이스 트랜잭션을 동적으로 처리하고, 오류 발생 시 자동 롤백을 실행합니다.

기여

- 돌봄 게시글 도메인 API와 단위테스트를 구현하였습니다.
 - PostgreSQL의 테이블 상속을 활용하여, 베이스 게시글 구조를 기반으로 다양한 게시글 종류를 관리하는 유연한 데이터 모델을 설계하고 구현하였습니다.
 - 돌봄조건 데이터를 효율적으로 적재하기 위해 마이그레이션 스크립트를 개발하였습니다.
- WebSocket 활용해 실시간 다중 채팅과 메시지 브로드캐스팅을 구현하였습니다.

DevQuest 2023.03 - 2023.05 / 프로젝트 인원 4명 [Github Link](#)

개발자 기술 면접을 위한 모의 면접 서비스입니다.

사용 기술: Java, Spring Boot, Spring Data JPA, AWS(S3, EC2, Route 53), Nginx, MySQL, Github Actions, Prometheus, Grafana

문제 해결

개요

기존 시스템에서는 '기본 면접 질문' 도메인과 '개인 등록 면접 질문' 도메인이 분리되어 관리되었습니다. 이 두 도메인의 주요 차이점은 사용자가 직접 등록한 질문을 다루느냐, 아니면 시스템이 기본적으로 제공하는 질문을 다루느냐에 있었습니다. 이로 인해 두 도메인에서 유사한 기능이 중복 구현되어 코드의 중복이 발생했습니다. 이런 구조는 유지 보수의 효율성을 저하시키고, 코드 베이스의 크기를 불필요하게 증가시켰습니다.

해결 과정

1. 관리자 역할을 추가해 서비스에서 기본으로 제공하는 질문을 관리자가 제공하도록 하였습니다.
2. 기본 질문과 개인 질문 테이블을 병합하고 유저와 질문 테이블을 1:N 관계로 재정의 하였습니다.
3. 기본 질문 도메인과 개인 질문 도메인을 병합하여 코드 중복을 줄였습니다.

기여

- Question 도메인 CRUD REST API 개발하였습니다.
 - 질문 목록에 대한 응답 시간 최적화를 목표로 Redis를 활용해 캐싱 처리하였습니다. (Latency 약 50% 감소)
- 사용자 답변 채점을 위해 ChatGPT API 이용한 사용자 면접 답변 피드백 기능 구현하였습니다.
- Github Actions, AWS S3, Code Deploy, EC2를 활용해 CI/CD 파이프라인 구축하고 배포하였습니다.
- AWS Route 53과 ACM를 활용해 도메인을 적용하고 HTTPS를 구성하여 서비스의 안정성을 향상시켰습니다.

Lego2Me

2021.12 - 2022.01 / 프로젝트 인원 4명, 팀 리더

[Github Link](#)

2022 실리콘밸리 인턴십 프로그램에서 진행한 프로젝트로, 이미지의 상, 하의를 분석하여 이미지와 유사한 옷을 입은 LEGO 캐릭터를 제공하는 서비스를 개발하였습니다.

사용 기술: Django, RabbitMQ, Celery, GCP(BigQuery, Google Cloud Storage), Nginx, Docker

문제 해결

개요

서비스 운영 중 한 서버에서 이미지 분석(AI 모델)과 운영 로직을 함께 처리하고 있었습니다. 이 구조에서 이미지 분석 과정에서 오류가 발생하면 서버 전체가 응답하지 않는 현상이 발생했습니다.

해결 과정

이미지 분석 작업을 수행하는 서버와 서비스 운영 로직을 처리하는 서버를 분리함으로써, 하나의 작업에서 발생하는 문제가 전체 시스템에 영향을 미치지 않도록 하였습니다. RabbitMQ와 Celery를 이용한 비동기 처리 도입하면서 요청 처리 시간을 단축시키고 서버의 부하를 분산시켰습니다. 최종적으로 CPU 바운드 작업을 I/O 바운드 작업으로 전환하면서 전체적인 시스템의 부하를 줄였습니다.

기여

- Python, RabbitMQ, Celery를 활용해 이미지 분석 파이프라인 구현하였습니다.
- Django, BigQuery와 Google Cloud Storage를 활용해 사용자 피드백 REST API 구현하였습니다.
- GCP Compute Engine, Docker을 이용한 서버 배포 및 Nginx를 이용한 포트포워딩하였습니다.

Skills

Application : Java(Spring Boot, JPA), Python(Django, DRF, Flask), Go, React

Database : MySQL, PostgreSQL, RabbitMQ, Redis

Testing : JUnit5, Mockito, Testify(Go Testing)

Infra: AWS, GCP, Github Actions, Gitlab CI, Nginx, Airflow

Activity

- [Techeer x D.camp x 티타임즈] 2023 SW 부트캠프 멘토 (2023.12 - 2024.2)
- 2023 카카오톡모빌리티 넥스트모빌리티 해커톤: 네모톤 예선
- [성남시] 성남청년프리인턴십 4기 멘토 (2022.9 - 2022.10)
- 2022 실리콘밸리 인턴십 프로그램 (2021.12 - 2022.01)
- Techeer [테크어] (2021.9 ~ 현재)
 - 실리콘밸리 엔지니어의 SW 개발자 커리어 그룹
- [성남시, POSCO ICT] 성남청년프리인턴십 2기 (2021.05 - 2021.07)

Education

성결대학교 - 컴퓨터공학전공 (2020.03 ~ 2024.2)

- 수시 우수 장학생 선정 / 국가우수장학금(이공계), 한국장학재단