윤영빈 경력기술서

안녕하세요 토스 러너스 하이 1기 지원자 윤영빈입니다.

24.12.17 ~ 25.01.16 한달동안 토스 온보딩 영상을 보고 직접 실무에서 개선해보았던 경험들을 적어보았습니다.

저는 **효율적인 시스템과 성능 최적화**를 좋아하는 개발자입니다. 메모리 관리, 동시성 처리, 네트워크 I/O 개선 등 **기능적인 완성도뿐만 아니라 성능과 안정성까지 고려한 개발**을 중요하게 생각합니다. 특히 **비효율적인 프로세스를 분석하고 개선하는 과정**에서 큰 보람을 느끼며, 보다 **빠르고 견고한 서비스**를 만드는 데 집중하고 있습니다.

---

1. 소켓 서버 메모리 누수 문제 해결

- **이슈:** 소켓 서버에서 store 사용 시 GC가 정상적으로 동작하지 않는 문제 발생
- **분석:**
  - heap dump 파일 분석 결과, client 관련 map 데이터가 지속적으로 유지됨
  - 중복 로그인 시 클라이언트에게 연결해제 이벤트를 발생시키나, socket out이 아니기 때문에 강한 참조로 인해 GC가 되지 않음
  - netty socket.io 라이브러리 내부에서 데이터를 완전히 해제하지 않아 메모리 누수 발생
- **개선 조치:**
  - 현재로서는 따로 클라이언트 정보관련된 세션관리를 하지 않고 있기 때문에 store 사용 제거
  - 메트릭 및 로깅 정보 추가 (개발/QA 환경 적용)
  - **결과:** 메모리 누수 방지 및 안정적인 소켓 서버 운영 가능

---

2. 주문 이벤트 분산락 적용 및 성능 개선

- **이슈:**
  - 주문 이벤트 발생시 동시성 문제 및 DB 부하 발생
  - 2주에 1번꼴로 동시성 이슈 관련 운영 대응 요청
- **분석:**
  - 오라클 데이터베이스 트리거의 락에 동시성 제어를 의존하고 있었기 때문에 100퍼센트 동시성 제어를 하기 어려움
  - 데이터베이스 레벨에서 락을 기대하게 되면 경합과 데드락이 발생할 수 있고 애플리케이션에서는 DB Exception 을 받아서 처리해야하기 때문에 DB 에 부하를 줄 수 있음
- **개선 조치:**
  - 애플리케이션 레벨에서 선행 분산락 적용 (DB 트리거 의존 제거)
  - **결과:**
    - 동시성 문제 해결 및 DB 부하 감소

---

3. 알림 이벤트 실시간 처리 개선

- **이슈:**
  - 기존 구조는 알림 로그 생성 후 폴링 방식으로 외부 인프라 호출 → 최대 5초 지연 발생
- **개선 조치:**
  - Spring Event를 활용하여 이벤트 생성 즉시 앱으로 알림을 발송하고 알림로그는 따로 저장하도록 개선
  - **결과:**
    - 알림 지연 시간 **100% 제거**
    - 실시간 알림 전송 가능

---

4. 트랜잭션 최적화 및 데드락 위험 제거

- **이슈:**
  - 기존 방식: 다건 업데이트 시 1+N 쿼리 발생 → 서브 데이터 개수 제한 없음 → 최대 15초 지연
  - 데드락 위험 존재
- **개선 조치:**
  - 단건 트랜잭션 분리하여 독립적으로 커밋 처리
  - 업데이트 쿼리에 인덱스 조건 추가
  - **결과:**
    - **트랜잭션 처리 속도 개선 (최대 15s → 수ms 단위)**

        - 데드락 발생 가능성 제거

---

6. API 엔드포인트 이관 후 성능 저하 해결

- **이슈:**
    - 기존 프로시저 → API 기반 엔드포인트로 변경 후 운영 배포
    - APM 분석 결과, 불필요한 대량 데이터 조회로 네트워크 I/O 지연 발생
- **개선 조치:**
    - 공통 메서드 제거 후, 최소 데이터 조회 로직으로 수정
    - **결과:**
        - 불필요한 네트워크 I/O 제거
        - API 응답 속도 **600ms → 120ms (80% 성능 개선)**

---

7. 지속적인 네트워크 통신 최적화 및 아키텍처 개선

- **이슈:**
    - 서버 간 의존성이 높은 구조 (MSA 차용했으나 완전한 MSA 아님)
    - 불필요한 네트워크 통신으로 인한 성능 저하
- **개선 방향:**
    - 아키텍처 재설계 진행 중 (네트워크 통신 효율 최적화)