

최우녕

Backend Engineer / Portfolio woonyong.contact@gmail.com GitHub



저장 구조의 병목을 찾고 개선한 SQL 엔진과, 장애 대응의 증거·변경·검증을 연결한 K8s Clue를 소개합니다.

SQL 엔진은 개인 주도 구현, Clue는 팀 설계 기여와 종료 후 개인 검토를 구분해 정리했습니다.

구현 사례

SQL 엔진

개인 주도 구현 · 정글 팀 과제

C11 · 저장 구조 · 성능 개선

- SQL 한 문장을 파싱하고 실행 계획을 세워 행을 읽고 쓰는 학습용 엔진. Slotted Page·B+Tree·Buffer Pool을 연결하고 행·범위 잠금을 구현
- id 범위 조회가 전체 행을 훑는 문제를 B+Tree 리프 순회로 변경. 일치하는 행 위치를 모은 뒤 힙을 읽어, 인덱스와 힙의 래치를 동시에 잡지 않도록 처리
- 100만 행으로 규모를 늘리자 INSERT 처리량이 급감. 스택 샘플링과 실행 경로 대조로 빈 슬롯 반복 탐색과 문장 종료 후 잠금 누적을 병목으로 확인
- 빈 슬롯이 있는지 기록하는 힌트를 추가해 불필요한 페이지 재탐색을 생략. DELETE 시 힌트를 갱신하고 DB 재열기 시 복원하며, REPL에서도 문장 종료 직후 잠금을 해제하도록 수정
- 동일 엔진을 수정 전후로 측정한 결과, INSERT 처리량 **2,751 → 591,429 ops/sec** (약 215배). PostgreSQL 대비 우수성이나 운영 트래픽 처리 성과가 아닌, 학습 엔진의 병목 개선 결과
- ASAN·UBSAN 빌드·테스트 CI 통과 기록과 측정 조건을 함께 공개. 범위 조회 개선은 id 인덱스에 한정하며 비-id 조건은 전체 스캔을 사용

측정: 같은 엔진의 수정 전후, 100만 행·단일 연결·합성 데이터·캐시 상주, 3회 중앙값. WAL·MVCC·다중 문장 트랜잭션은 미지원.

K8s Clue

5인 팀장 · 설계 / 종료 후 개인 검토

Kubernetes 장애 진단·변경 제안 도구

- 장애 관측 시점의 상태, 원인 판단, 변경과 배포 결과가 서로 다른 도구에 흩어지는 문제를 다룬 팀 프로젝트. 전체 아키텍처·장애 파이프라인·서비스 간 인터페이스 설계 담당
- 증거 수집 → 원인 판정 → 변경 제안 → 사후 검증을 같은 사건 ID로 연결. 단계가 실패해도 원본 증거와 실패 원인을 따라갈 수 있도록 인터페이스 설계
- 자동화가 클러스터를 직접 수정하지 않도록 읽기 전용 수집과 변경 제안을 분리. 변경은 허용된 manifest 필드의 GitHub Draft PR로 제한하고 자동 merge는 제외
- 프로젝트 종료 후 대표 경로를 ImagePullBackOff 하나로 좁혀 검토. 허용 필드·base SHA·Draft PR 조건을 코드와 테스트에 대조
- 대표 계약에서 사건 ID의 전파, 허용되지 않은 필드 변경·직접 commit 거부, 실패 시 원본 증거 참조 보존을 확인할 수 있도록 근거 연결
- 기여 범위는 팀 전체 코드의 개인 구현이 아니라 파이프라인·인터페이스 설계와 종료 후 검토. 로컬 계약 테스트와 실제 클러스터·GitHub를 연결한 외부 검증을 구분

대표 검증 범위: ImagePullBackOff 1개 시나리오. 실사용 트래픽이 없는 데모이며 외부 클러스터 E2E는 미수행.

#Python #Kubernetes #PostgreSQL #NATS