

# Source Code Analysis

# RockDB 개선안

---

컴퓨터소프트웨어학부

김현정 박정호

# CONTENTS

---

**01** 스크립트 세팅  
컴파일용 스크립트, 실험용 스크립트

**02** 새로운 아이디어  
아이디어 및 질문

# 스크립트 세팅

---

컴파일용 스크립트

실험용 스크립트

# 컴파일용 스크립트

기존의 RocksDB 라이브러리와 수정된 RocksDB 라이브러리를 둘다 빌드하기 위해 따로 Makefile을 만들었다. RocksDB에서 제공하는 examples 를 빌드하는 Makefile을 참고했으나, 구조 상 두 라이브러리를 빌드하는 것이 어려워서 Shell Script까지 사용했다.

Shell Script를 사용해서 두 static librar를 모두 빌드할 수 있도록 했고, 현재는 무조건 두 라이브러리가 모두 다시 빌드되도록 설정되어있다. 이후 분기를 만들어서 상황에 따라 필요한 파일만 빌드할 수 있도록 할 생각이다.

Experiment 디렉토리의 compile.sh 를 직접 실행하거나, make 하는 것으로 빌드가 가능하다.

# 실험용 스크립트

Python으로 실험을 진행하는 스크립트를 구현해두었다.

옵션을 2가지 설정해두었고, 그 내용은 다음과 같다.

1. `--workload`  
실험에 사용할 workload의 형식에 대한 옵션이다.  
`both`, `skewed`, `uniform`의 총 3가지가 가능하며, default는 `both`이다.
2. `--rocksdb`  
실험할 RocksDB 라이브러리에 대한 옵션이다.  
`both`, `origin`(원래 RocksDB), `custom`(수정된 RocksDB)의 총 3가지가 가능하며, default는 `both`이다.

실행하게 될 경우 설정된 옵션에 해당하는 테스트 프로그램을 실행한다.

각 실행 후, 결과를 모아서 matplotlib을 이용해 그래프로 만든 후 저장한다.

단, 실행해야 할 테스트 프로그램이 빌드되지 않았다면 에러 메시지를 출력하도록 했다.

# 새로운 아이디어

---

아이디어 및 질문

# 아이디어들.

1. Index block의 building과정에서 user key의 shortest prefix를 구해서 key순(Increasing order)으로 block을 정렬하는 코드에서 착안하여 seq number에 해당하는 부분도 decreasing order로 정렬하는 것을 동시에 support를 할 수 있는지에 대한 고민을 하게 되었다.
2. 현재 index block을 구성하기 위해 key값에 접근하는 부분을 분석하고 있는데, internal key(user key와 seq number를 결합한 정보) 자체가 정렬되는 user key와 함께 seq number를 가지고 있다. 따라서 user key정렬과정에서 기존의 min-max 아이디어에 필요한 min, max값을 추출하면 어떨까하는 생각을 했다.
3. Snapshot의 seq number와 인접하여 자주 쓰일만한 값들을 예측할 수 있는 방법은 없을까. Seq number를 위한 cache같은 느낌으로.

# THANK YOU!

---

감사합니다!