

# 엣지 디바이스 상에서의 업데이트 가능한 런드 인덱스 성능 평가

박태운, 김옥희  
건국대학교

gspear@konkuk.ac.kr, wookhee@konkuk.ac.kr

## Performance Evaluation of Updatable Learned Indexes on Edge Device

Taeyoon Park, Wook-Hee Kim  
Konkuk Univ.

### 요약

런드 인덱스는 데이터의 분포로 학습된 여러 개의 모델이 계층을 이루도록 구성된 인덱스 구조이다. 학습된 모델이 예측한 위치를 바탕으로 탐색을 진행하기 때문에 전통적 인덱스 구조보다 좋은 탐색연산 성능을 보이나, 삽입연산과 같은 업데이트에는 취약하다. 최근 이러한 문제를 해결하기 위하여 다양한 업데이트 가능한 런드 인덱스가 개발되고 있으나, 대부분의 런드 인덱스는 x86 시스템을 기반으로 개발되었으며, 엣지 디바이스에서의 연구는 아직 활발하지 않은 것으로 보인다. 따라서, 본 연구에서는 엣지 디바이스에서의 업데이트 가능한 런드 인덱스들을 분석하고 성능 평가를 진행하였다.

### I. 서론

런드 인덱스는 데이터의 값과 위치정보로 학습한 모델을 활용해 데이터를 관리하는 인덱스 구조이다. 모델의 오차로 인해 실제 데이터의 위치와 예측한 데이터의 위치의 차이가 발생하기 때문에 예측한 위치에 대해 일정 범위의 탐색을 진행하여 실제 위치를 찾게 된다. 특히, 현실 세계 데이터는 복잡한 분포를 갖기 때문에 단일 모델로 전체 데이터를 학습하게 될 경우 모델의 오차가 증가하여 탐색 성능이 저하될 수 있다. 따라서 런드 인덱스는 유사한 분포를 갖는 데이터들끼리 나누어 모델을 학습하도록 하고, 학습한 모델들이 계층 구조를 이루도록 구성하여 모델이 높은 정확도를 유지해 좋은 탐색 성능을 갖도록 하였다. 그러나, 새로운 데이터가 추가되면 데이터의 분포가 모델이 학습한 분포와 달라져 삽입 연산 등에 취약하기 때문에, 업데이트 가능한 런드 인덱스 연구가 활발히 진행되고 있다. 또한, 현재 대부분의 런드 인덱스는 x86 아키텍처 기반의 서버급 성능을 갖는 머신을 기반으로 연구가 진행되었다. 따라서 본 연구에서는 서버급 머신 보다는 성능이 낮은 엣지 디바이스에서 런드 인덱스의 성능 평가를 진행하였다.

### II. 업데이트 가능한 런드 인덱스

실험에 앞서 먼저 본 연구에서 활용한 업데이트 가능한 런드 인덱스에 대해 설명하고자 한다.

#### II-1. ALEX

ALEX [1]은 B+-Tree 와 같이 실제 데이터를 키-밸류 형태로 저장하는 리프 노드와 하위 노드 정보를 저장하는 내부 노드로 구성된 계층 구조의 런드 인덱스이다. 모든 노드는 자신의 학습된 모델을 통해 노드 내 키의 위치를 예측하며, 모델의 정확도를 유지하며 삽입연산을 처리하기 위한 Gap 이라는 버퍼 구조를 사용한다. 키 탐색 시 내부 노드는 리프 노드에 도달할 때까지 모델이 예측한 위치 정보를 통해 하위 노드로 이동한다. 리프 노드 내 탐색은 모델이 예측한 위치에서 Exponential Search 를 통해 Gap 구조로 인해 발생한 빈 공간을 빠른 속도로 탐색한다.

#### II-2. PGM-index

PGM-index [2]은 사전 정의된 오차한계를 만족하도록 학습된 모델들로 이루어진 계층 구조의 런드 인덱스이다. 가장 낮은 계층의 모델들은 키와 키의 위치정보로 학습하며, 나머지 계층은 하위 계층 모델의 키 범위 내 최소 키와 해당 최소 키의 위치정보로 학습하며, 이는 최상위 계층에 한 개의 모델만 생성될 때까지 반복한다. 키 탐색은 최상위 계층에서 시작해 키를 찾을 때까지 모델이 예측한 위치에서 이진 탐색을 통해 올바른 하위 계층 모델로 이동을 반복한다.

#### II-3. LIPP

LIPP [3]은 노드에 실제 키-밸류와 하위 노드 정보를 함께 저장하는 계층 구조의 런드 인덱스이다. LIPP 은 Faster Minimum Conflict Degree (FMCD)알고리즘을 사용해 모델을 학습하여 모델이 항상 정확한 키의

위치를 예측할 수 있다. 모델 학습 과정에서 모델이 두 개 이상의 키에 대해 같은 위치 정보를 예측하는 경우, 하위 노드를 생성해 새로운 모델로 해당 키들을 학습하여 모델의 오차를 제거하였다. 키 탐색 시 모델이 예측한 위치에서 키를 찾으면 탐색이 종료되며, 그렇지 않은 경우 하위 노드로 이동하여 이를 반복한다.

#### II-4. XIndex

XIndex [4]은 멀티 스레드 환경을 고려하여 설계된 두 개의 계층 구조로 이루어진 런드 인덱스이다. 상위 계층인 루트 노드는 Recursive Model Index (RMI)구조를 통해 하위 계층인 그룹 노드의 정보를 저장한다. 그룹 노드는 실제 키-밸류와 학습된 모델, 새로 삽입된 키-밸류를 저장하는 버퍼로 구성되어 있다. 키 탐색 시 루트 노드의 RMI 구조를 통해 그룹 노드를 선택하고, 해당 그룹 노드 내 모델이 예측된 위치에서 탐색을 통해 키의 실제 위치를 찾는다.

### III. 실험

#### III-1. 실험환경 및 런드 인덱스 동작 여부 확인

실험은 6 코어의 Arm Cortex-A78AE v8.2 64-bit CPU 와 8GB 메모리를 탑재한 NVIDIA Jetson Orin Nano 에서 진행하였다.

표 1. 런드 인덱스 동작 여부 및 실패 원인

런드 인덱스	동작 여부	실패 원인
ALEX	O	-
PGM-index	O	-
LIPP	X	Out of Memory
XIndex	X	x86 에서만 사용가능한 라이브러리

엡지 디바이스 내 런드 인덱스 동작 여부를 확인해본 결과, 표 1 과 같이 LIPP 와 XIndex 는 동작하지 않음을 확인하였다. LIPP 는 모델 학습 시 사용하는 FMCD 알고리즘으로 인해 반복적으로 노드가 생성되어 인덱스 구조의 크기가 증가하였고, 이로 인해 Out of Memory 가 발생하였다. XIndex 는 모델 학습 과정에서 x86 에서만 사용 가능한 라이브러리를 활용하는 관계로 ARM 아키텍처 기반인 엡지 디바이스에서 정상적으로 컴파일 되지 않았다.

#### III-2. YCSB Benchmark

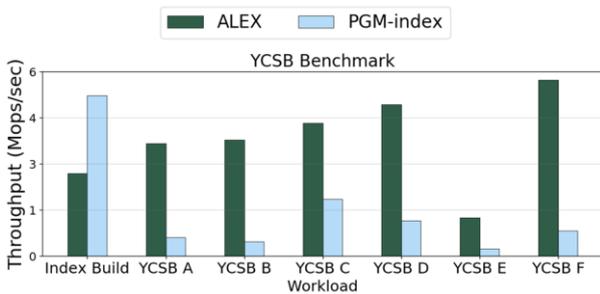


그림 1. YCSB Benchmark 측정 결과

본 실험에서는 동작하지 않는 LIPP, XIndex 를 제외한 ALEX, PGM-index 를 5 천만개의 YCSB [5]데이터로 구축한 뒤, 인덱스 구조 구축, Workload A (Read 50%, Update 50%), Workload B (Read 95%, Update 5%), Workload C (Read 100%), Workload D (Read 95%, Insert 5%), Workload E (Range Query 95%, Insert 5%), Workload F(Read-Modify-Write)에 대한 성능을 측정하였다. 그림 1 과 같이 ALEX 는 PGM-Index 대비 인덱스 구축 과정에서 모델의 높은 정확도를 위해 오랜 시간이 걸렸고, 이로 인해 데이터의 위치를 낮은 오차로 예측할 수 있게 되어 모든 Workload 에서 PGM-index 보다 좋은 성능을 보였다.

### IV. 결론

본 논문은 엡지 디바이스 환경에서 런드 인덱스 ALEX, PGM-index, LIPP, XIndex 의 성능을 평가하였다. ALEX 는 높은 정확도를 갖는 모델을 통해 좋은 탐색성능을 보였다. PGM-index 는 정의된 오차한계로 모델을 학습하여 넓은 범위의 고정된 탐색 범위를 갖기 때문에, ALEX 보다 낮은 탐색 성능을 보였다. 또한 LIPP 와 Xindex 는 x86 아키텍처에 최적화되어 엡지 디바이스에서는 정상적으로 동작하지 않음을 확인하였다. 런드 인덱스는 전통적인 인덱스 대비 자원을 적게 쓸 수 있을 것으로 기대되므로, 한정된 자원을 갖는 엡지 디바이스 환경에서 런드 인덱스를 활용하기 위한 연구가 필요하며, 우수한 인덱스 구축 성능과 탐색 성능을 가질 수 있는 런드 인덱스 연구가 필요하다.

### ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원·학·석사연계 ICT 핵심인재양성 지원을 받아 수행된 연구임(IITP-2024-RS-2020-II201834)

### 참 고 문 헌

- [1] Ding, J., Minhas, U. F., Yu, J., Wang, C., Do, J., Li, Y., Zhang, H., Chandramouli, B., Gehrke, J., Kossmann, D., Lomet, D., Kraska, T. 2020. ALEX: An Updatable Adaptive Learned Index. Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20), pp.969- 984.
- [2] Ferragina, P., Vinciguerra, G. 2020. The PGM-index: a fully-dynamic compressed learned index with provable worst-case bounds. PVLDB, 13(8): pp.1162- 1175
- [3] Wu, J., Zhang, Y., Chen, S., Chen, Y., Wang, J., Xing, C. 2021. Updatable Learned Index with Precise Positions. PVLDB, 14(8): pp.1276- 1288
- [4] Tang, C., Wang, Y., Dong, Z., Hu, G., Wang, Z., Wang, M., Chen, H. 2020. XIndex: a scalable learned index for multicore data storage. PPOPP. ACM, pp.308- 320.
- [5] Cooper, B., Silberstein, A., Tam, E., Ramakrishnan, R., Sears, R. 2010. Benchmarking Cloud Serving Systems with YCSB. In Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC '10). ACM, New York, NY, USA, pp.143- 154.