

Project5 Report

20160463 성해빈

b+ tree로 구현했다.

Class

Node

InternalNode와 LeafNode의 parent class다.

둘 모두 공통으로 사용하는 keys vector와 is_leaf 변수가 있다.

key는 최대 order-1만큼 저장되며, order만큼 저장되는 순간 overflow로 간주해 split한다.

InternalNode

InternalNode는 Node*의 vector인 children이 있다.

children은 최대 order 개수만큼 저장된다.

LeafNode

LeafNode는 int8_t의 vector인 values이 있다.

values는 최대 order-1 개수만큼 저장된다.

BPlusTree

update, insert, search, scan operation을 구현했다.

insert의 경우 search 후 leaf insert를 진행한다. key의 개수가 order만큼이면 split을 진행하고, 그 결과 parent에도 split이 필요하다면 진행한다.

scan의 경우 search후 B+ tree의 leaf linked list를 사용한다.

Optimization

Order

Order을 정하는 기준으로 key값의 조회가 모조리 한 cache line에 있도록 해보는 방법이 있을 것이다.

하지만 cache line이 64 byte인데, 그에 해당하게 ORDER을 8로 지정해도 속도가 오히려 느리다.

cache로 보는 이득보다 높기로 손해보는 것이 더 큰 것 같다.

order이 어느 지점 이상을 넘어가면 또 오히려 느리다.

b+ tree 구조로 보는 이득을 얻지 못해서 인 것 같다.

Reserve

vector를 쓸때, 최대 개수인 order에 맞게 reserve함수로 초기화를 해주면 allocation 코스트를 최소화할 수 있다.

Node, InternalNode, LeafNode 모두의 constructor에 order가 있는 이유다.