Index B+tree 응용

Distribution Code Tuning

2015004493

김형순

0. 개발/테스트 환경

1. 제공된 docker 이미지를 이용하여 개발/테스트 하였습니다.

1. 프로그램 설명 및 사용방법

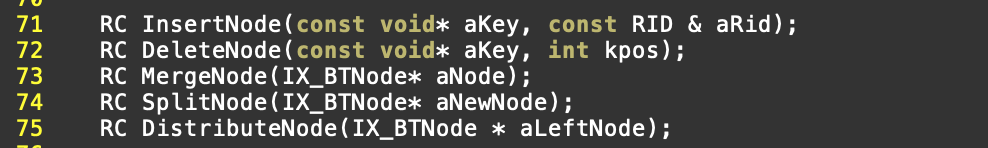
1. edubase git lab 에 Assignment3 디렉토리에 과제를 하며 수정 한 파일 3개 (ix.h, ix\_btnode.cc, ix\_indexhandle.cc) 가 있습니다.
2. 기존 edubase 의 ix.h, ix\_btnode.cc, ix\_indexhandle.cc 파일을 제출된 파일을 교체 합니다.
3. make 를 이용해 데이터베이스를, make testers 를 이용해 테스트 파일들을 컴파일 합니다.

2. 구현 방법, 함수 별 코드 설명

1) ix.h

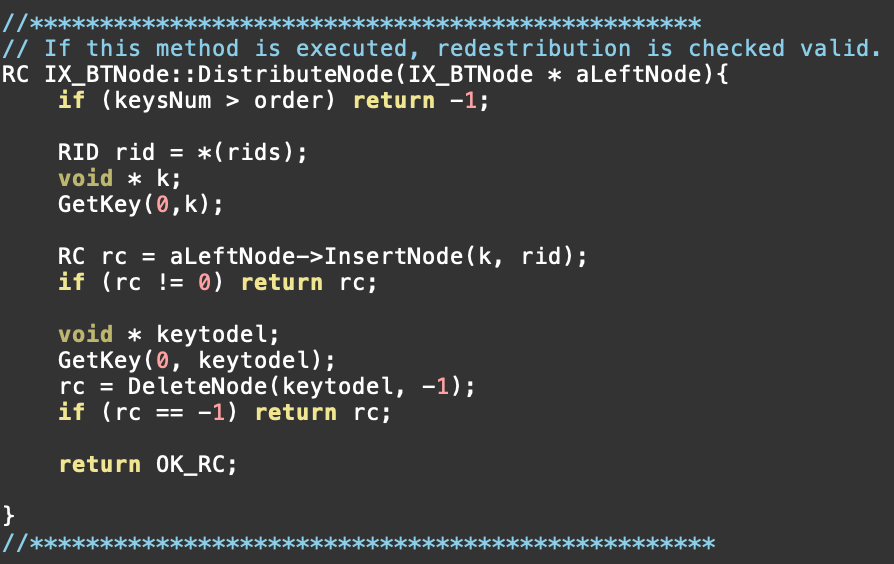
1. index 관리를 위한 헤더 파일 입니다.
2. b+ tree 의 노드에 해당하는 클래스 IX\_BTNode 클래스에 distribution 을 위한

RC DistributeNode(IX\_BTNode \* aLeftNode) 메소드를 선언 합니다.



2) ix\_btnode.cc

Distribution 을 위한 함수를 정의 합니다.



1. 0번째 key(가장 왼쪽의 key) 에 해당하는 데이터를 가져옵니다.
2. 가져온 데이터를 자신의 왼쪽 노드에 insert 합니다.
3. 그 데이터를 현재 노드에서 제거 합니다.
4. 이 메소드가 실행 되기 전에 distribution 가 가능한 조건 (왼쪽 노드가 있어야 하고, 같은 parent 밑의 sibling 이어야 함)을 만족시킨 상태 이어야 합니다.

3) ix\_indexhandle.cc

Insert 후 distribution 조건을 확인하고, distribution 을 시도 합니다.



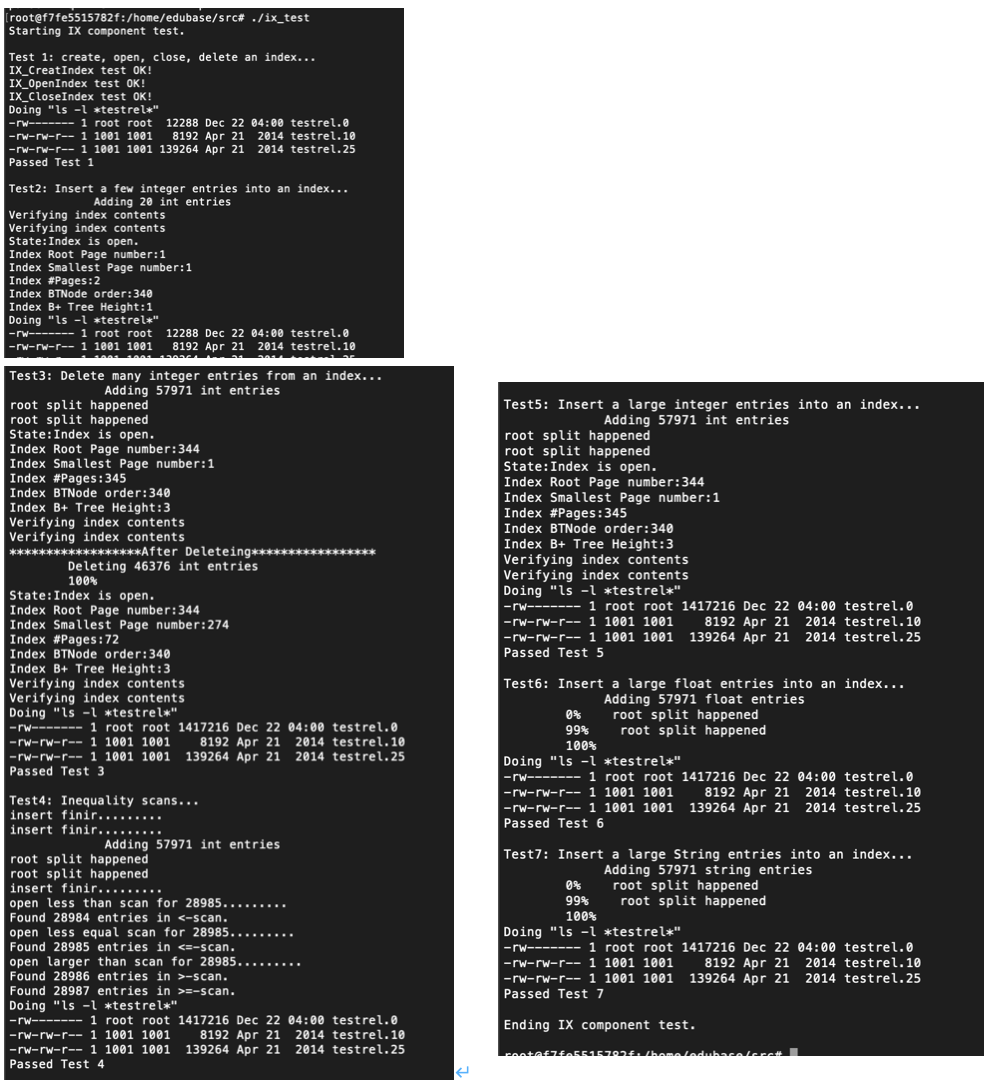
1. Distribution 을 하기 위한 조건은 다음과 같습니다.
2. insert 하려는 node가 가득 차있어 insert 가 실패 (result == -1) 해야 합니다.
3. leftNode 가 존재 해야 합니다. (leftNode != NULL)
4. leftNode 와 insert 하려는 node의 parent 가 같아야 합니다.
5. leftNode 에 insert 하려는 공간이 비어 있어야 합니다. (GetKeysNum < GetOrder)
6. 위 조건을 확인하기 위해 필요한 것은 path 와 pathP, FindLeaf() 를 사용해야 합니다.
7. path 는 root 부터 insert node 로 올 때까지의 경로에 해당하는 node 를 저장하고 있습니다. 이것을 이용하여 leftNode 와 현재 node 의 parent 가 같은지 확인 합니다.
8. FindLeaf 는 path와 pathP 를 바꿔줍니다. 찾고 싶은 node 의 데이터를 인자로 주어 path 를 바꿀 수 있습니다. 주의 할 점은 FindLeaf 함수는 변수 node 도 바꾸기 때문에 왼쪽 node 의 path 를 찾은 후 다시 node 를 사용하기 전에 복구시켜 주어야 합니다.
9. Distribute node 가 끝나면 parent node 의 key 도 바꾸어 주어야 합니다. pathP 를 이용해 leftNode 로의 pathP, 즉 leftNode 와 현재 node 를 구분하는 key 값을 바꾸어 주어야 합니다. parentNode 의 key 를 삭제하고 insert 하여 key 를 바꿉니다.
10. 마지막으로 distribution 이 끝난 후 compare key 를 통해 leftNode, 혹은 원래 insert 하려 했던 node 에 data 를 다시 insert 합니다.

3. 실행(테스트 결과)

b+ 트리의 distribution이 잘 실행 되는 것을 확인 하기 위해 ix\_test 를 실행 해 보았다.

1. Distribution 을 하지 않았을 때

- 테스트 3, 5 를 보면 57971 개의 int entry 를 넣었을 때 345 개 의 page 가 indexing 에 사용되고, b+ tree 의 height 가 3 인 것을 확인 할 수 있다.

­­

1. Distribution 을 사용 했을 때

- 테스트 3, 5 를 보면 57971 개의 int entry 를 넣었을 때 173 개 의 page 가 indexing 에 사용되고, b+ tree 의 height 가 2 로 줄어 memory 활용 효율이 높아진 것을 확인 할 수 있다.

