

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.

```
70
71 RC InsertNode(const void* aKey, const RID & aRid);
72 RC DeleteNode(const void* aKey, int kpos);
73 RC MergeNode(IX_BTNode* aNode);
74 RC SplitNode(IX_BTNode* aNewNode);
75 RC DistributeNode(IX_BTNode * aLeftNode);
```

2) ix_btnode.cc

Distribution 을 위한 함수를 정의 합니다.

```
//*****
// If this method is executed, redistribution is checked valid.
RC IX_BTNode::DistributeNode(IX_BTNode * aLeftNode){
    if (keysNum > order) return -1;

    RID rid = *(rids);
    void * k;
    GetKey(0,k);

    RC rc = aLeftNode->InsertNode(k, rid);
    if (rc != 0) return rc;

    void * keytodel;
    GetKey(0, keytodel);
    rc = DeleteNode(keytodel, -1);
    if (rc == -1) return rc;

    return OK_RC;
}
//*****
```

- 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, 같

은 parent 밑의 sibling 이어야 함)을 만족시킨 상태 이어야 합니다.

3) ix_indexhandle.cc

Insert 후 distribution 조건을 확인하고, distribution 을 시도 합니다.

```
// *****
// Failed to insert node.
// Try redistribution.

if(result == -1){
    IX_BTNode* leftNode = FetchNode(node->GetLeft());
    if (leftNode != NULL){
        IX_BTNode* parentNode = path[level -1];

        void * leftLargest = leftNode->LargestKey();
        // Find left path.
        FindLeaf(leftLargest);
        IX_BTNode* leftParent = path[level -1];
        int posAtLeftParent = pathP[level -1];

        if ((leftParent == parentNode)
            && (leftNode->GetKeysNum() < leftNode->GetOrder())){
            // Able to Redistribute
            // Left is not full.

            // change path to inserting node
            FindLeaf(pData);

            //Distribute
            result = node->DistributeNode(leftNode);

            // update old key - keep same addr
            leftParent->DeleteNode(NULL, posAtLeftParent);
            leftParent->InsertNode((const void*) leftNode->LargestKey(),
                                  leftNode->GetNodeRID());

            if (result == 0){
                // Insert data to node.
                if (node->CompareKey(pData, leftNode->LargestKey()) >= 0) {
                    node->InsertNode(failedKey, failedRid);
                } else { // <
                    leftNode->InsertNode(failedKey, failedRid);
                }
            }
        }
    }
}
// Recover path.
FindLeaf(pData);
// *****
```

- ① Distribution 을 하기 위한 조건은 다음과 같습니다.
 - i. insert 하려는 node가 가득 차있어 insert 가 실패 (result == -1) 해야 합니다.
 - ii. leftNode 가 존재 해야 합니다. (leftNode != NULL)
 - iii. leftNode 와 insert 하려는 node의 parent 가 같아야 합니다.
 - iv. leftNode 에 insert 하려는 공간이 비어 있어야 합니다. (GetKeysNum < GetOrder)
- ② 위 조건을 확인하기 위해 필요한 것은 path 와 pathP, FindLeaf() 를 사용해야 합니다.
- ③ 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.
- ④ 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.
- ⑤ 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.

- ⑥ 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.

```
(root@f7fe5515782f:/home/edubase/src# ./ix_test
Starting IX component test.

Test 1: create, open, close, delete an index...
IX_CreatIndex test OK!
IX_OpenIndex test OK!
IX_CloseIndex test OK!
Doing "ls -l *testrel*"
-rw-rw-r-- 1 root root 12288 Dec 22 04:00 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 1

Test2: Insert a few integer entries into an index...
Adding 20 int entries
Verifying index contents
Verifying index contents
State:Index is open.
Index Root Page number:1
Index Smallest Page number:1
Index #Pages:2
Index BTreeNode order:340
Index B+ Tree Height:1
Doing "ls -l *testrel*"
-rw-rw-r-- 1 root root 12288 Dec 22 04:00 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
```

```
Test3: Delete many integer entries from an index...
Adding 57971 int entries
root split happened
root split happened
State:Index is open.
Index Root Page number:344
Index Smallest Page number:1
Index #Pages:345
Index BTreeNode order:340
Index B+ Tree Height:3
Verifying index contents
Verifying index contents
*****After Deleteing*****
Deleting 46376 int entries
100%
State:Index is open.
Index Root Page number:344
Index Smallest Page number:274
Index #Pages:72
Index BTreeNode order:340
Index B+ Tree Height:3
Verifying index contents
Verifying index contents
Doing "ls -l *testrel*"
-rw-rw-r-- 1 root root 1417216 Dec 22 04:00 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 3

Test4: Inequality scans...
insert finir.....
insert finir.....
Adding 57971 int entries
root split happened
root split happened
insert finir.....
open less than scan for 28985.....
Found 28984 entries in <-scan.
open less equal scan for 28985.....
Found 28985 entries in <=scan.
open larger than scan for 28985.....
Found 28986 entries in >-scan.
Found 28987 entries in >=scan.
Doing "ls -l *testrel*"
-rw-rw-r-- 1 root root 1417216 Dec 22 04:00 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 4
```

```
Test5: Insert a large integer entries into an index...
Adding 57971 int entries
root split happened
root split happened
State:Index is open.
Index Root Page number:344
Index Smallest Page number:1
Index #Pages:345
Index BTreeNode order:340
Index B+ Tree Height:3
Verifying index contents
Verifying index contents
Doing "ls -l *testrel*"
-rw-rw-r-- 1 root root 1417216 Dec 22 04:00 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 5

Test6: Insert a large float entries into an index...
Adding 57971 float entries
0% root split happened
99% root split happened
100%
Doing "ls -l *testrel*"
-rw-rw-r-- 1 root root 1417216 Dec 22 04:00 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 6

Test7: Insert a large String entries into an index...
Adding 57971 string entries
0% root split happened
99% root split happened
100%
Doing "ls -l *testrel*"
-rw-rw-r-- 1 root root 1417216 Dec 22 04:00 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 7

Ending IX component test.
```

② Distribution 을 사용 했을 때

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

```
root@f7fe5515782f:/home/edubase/src# ./ix_test
Starting IX component test.

Test 1: create, open, close, delete an index...
IX_CreatIndex test OK!
IX_OpenIndex test OK!
IX_CloseIndex test OK!
Doing "ls -l *testrel*"
-rw----- 1 root root 12288 Dec 22 03:57 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 1

Test2: Insert a few integer entries into an index...
Adding 20 int entries
Verifying index contents
Verifying index contents
State:Index is open.
Index Root Page number:1
Index Smallest Page number:1
Index #Pages:2
Index BTreeNode order:340
Index B+ Tree Height:1
Doing "ls -l *testrel*"
-rw----- 1 root root 12288 Dec 22 03:57 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 2
```

```
Test3: Delete many integer entries from an index...
Adding 57971 int entries
root split happened
State:Index is open.
Index Root Page number:3
Index Smallest Page number:1
Index #Pages:173
Index BTreeNode order:340
Index B+ Tree Height:2
Verifying index contents
Verifying index contents
*****After Deleteing*****
Deleting 46376 int entries
100%
State:Index is open.
Index Root Page number:3
Index Smallest Page number:138
Index #Pages:37
Index BTreeNode order:340
Index B+ Tree Height:2
Verifying index contents
Verifying index contents
Doing "ls -l *testrel*"
-rw----- 1 root root 712704 Dec 22 03:58 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 3

Test4: Inequality scans...
insert finir.....
insert finir.....
Adding 57971 int entries
root split happened
insert finir.....
open less than scan for 28985.....
Found 28984 entries in <-scan.
open less equal scan for 28985.....
Found 28985 entries in <=scan.
open larger than scan for 28985.....
Found 28986 entries in >-scan.
Found 28987 entries in >=scan.
Doing "ls -l *testrel*"
-rw----- 1 root root 712704 Dec 22 03:58 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 4
```

```
Test5: Insert a large integer entries into an index...
Adding 57971 int entries
root split happened
State:Index is open.
Index Root Page number:3
Index Smallest Page number:1
Index #Pages:173
Index BTreeNode order:340
Index B+ Tree Height:2
Verifying index contents
Verifying index contents
Doing "ls -l *testrel*"
-rw----- 1 root root 712704 Dec 22 03:58 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 5

Test6: Insert a large float entries into an index...
Adding 57971 float entries
0% root split happened
100%
Doing "ls -l *testrel*"
-rw----- 1 root root 712704 Dec 22 03:58 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 6

Test7: Insert a large String entries into an index...
Adding 57971 string entries
0% root split happened
1% root split happened
100%
Doing "ls -l *testrel*"
-rw----- 1 root root 1404928 Dec 22 03:58 testrel.0
-rw-rw-r-- 1 1001 1001 8192 Apr 21 2014 testrel.10
-rw-rw-r-- 1 1001 1001 139264 Apr 21 2014 testrel.25
Passed Test 7

Ending IX component test.

root@f7fe5515782f:/home/edubase/src#
```