Report

Jiasen Li
ACM HONOR CLASS 2017
SHANGHAI JIAO TONG UNIVERSITY
LIJIASEN00921@GMAIL.COM

1. Design Decisions

Join, Filter, HashJoin

In the Hash Join, I use a HashMap <Field, Vector <Tuple> > to store the Tuples instead of a MultiMap. The java will help me deal with the structure in the Memory. In addition, I use Vectors with an initial size of 1 to save the space. I add the **"buildHashMap"** function **(API added)**.

In the normal Join, there are no tricky ideas.

Notice that the interface of Tuple does not have "merge". Why does Tuple Desc need "merge"? Tuple Desc should not be modified after it is created, but the Tuple can be modified with its interface, so merging can be implemented with modification functions in Tuple. I also add the **getLength** to the TupleDesc. **(API added)**

In the Filter, there are no tricky ideas.

Aggregator Aggregate

I use a class AggregatorNode that counts the number, max, min, sum of a sequence. HashMap <Field (Group Field), AggregatorNode > saves the counters.

There is a tricky thing that Aggregator should also be able to return the **Tuple Description**, so that it can be used in Aggregate. **(API added)**

HeapFile Mutability

I implement it in lab2, so I just modify it a little bit and pass the test. Notice that there are also tests for Invalid Deletion.

Insertion and Deletion

Normal non-tricky Implementation. Notice that there can be empty child Iterator.

BufferPool

After I implemented lab3.1 to lab3.4, I happily ran "ant test" and "ant systemtest".

However there was a test case in the folder not mentioned in the readme.md.

It is tricky that a test case in it does not use BufferPool.getpage. I manage all the dirty pages with File.getPage, which calls the BufferPool.getPage. A New Page is added when it is needed in File.insertTuple.

I use this implementation because I see the idea in BTreeFile and find it very cool.

So I additionally handle the case that the Users call the Page.insertTuple without using File.insertTuple.

Evict Strategy

I use the LRU and maintain two LRUs. One stores normal pages, the other one stores dirty pages. I know that the dirty pages owned by insertTuple will not be released even if I evict it because it is still in the DirtyPage List and will be returned to the insertTuple. So when I evict a page, if it is in the DirtyPage List, I evict it into the Victim LRU. Otherwise, I evict it into the disk. After the Insertion, I evict those who should have been evicted but cancelled due to the "lock". Pages in Victim LRU can also be put back to Normal LRU if it is used.

2. Review

I spent 2 days approximately implementing Lab3. I understand a lot with the help of TAs. I implemented all the required codes and I passed all the tests and system tests.

Jiasen Li
ACMDB LAB3