Report: ACMDB LAB4

Jiasen Li
LIJIASEN00921@GMAIL.COM
ACM HONOR CLASS

Design Decisions

IntHistogram

I use a Segment Tree with a small constant. I know that the number of buckets may be 100 or more. This is larger than a Typical Log(N) for Segment Tree. I implement my Segment Tree with the Non-Recursive Method.

I use Float64 (double) Border. This is much easier to implement than Integer Border. And I used the Small-Constant Binary Search to get the integral Starting Point and the End Point of a bucket.

StringHistogram

I enlarge the possible input of a Segment Tree in the IntHistogram to “Long”, so that the range does not exceed “Int” when doing Binary Search and translating Index to BucketNumber.

TableStats

I create a “MinMaxNode” that counts the largest and the smallest item in each Int Field. This costs 1 scan. Then, I use Histogram (int / string) to do Stats for the Table. This costs another scan.

The 2-Scan Method can estimate more precisely than scanning it just once.

JoinOptimizer

I implement a more efficient method to get the subset of a Vector of length n. I count the number from 0 to (2^n)-1. Each number means a subset. If Long.bitcount(state) is the length it requires, construct a subset according to state.

It runs Long.bitcount(state) for at most (n*2^n) times. It constructs the subset for at most (2^n) times, when each construction costs at most n. So the time complexity is O(n*2^n) in all, finding subsets of size 1 to n. So in the average case, it is O(2^n) per call, with a small constant.

I estimate the Card and Cost according to the hints in the repo, which is simple. “Compute Cost And Card Of Subplan” has already considered left join and right join in the left deep tree. I manually update the PlanCache After calling the “Compute...” function.

I find something very tricky. The name of the field has not been tested in the previous unit tests and system tests. And each Field Name Should be checked carefully.

I changed the Field Name In SeqScan according to BTreeScan. And there was a check of Tuple Description in the Insert.java. I changed it into only checking the type of the TD. I added the Field Name Checking in the BufferPool.insertTuple.

API Change?

No, I did not change the API in this Lab.

Missing or Incomplete Elements?

The Lock and Transaction will be implemented in the Next Lab.

I have passed the **ant test** and **ant systemtest.**

I think that the field name may be updated in the next lab. I am not sure.

How Long Did I implement this?

I do not remember the exact time, but it is longer than I expected. I found some problems in Field Name, and some problems in its structure when I saw it at first sight. I solved them.

With the help of the TAs, I learned a lot about the Principle of Database.