Liu, Cong, A53207007 Xie, Xianda, A53218476
Department of Electrical Engineering and Computer Science
University of California, San Diego

Mar 16, 2017

MILESTONE3 — Applying Database Optimization Techniques on XQuery

1 Project Overview

This project is based on the previous two milestones. The XQuery engines built in the first part of the project miss the query optimization opportunities we know from the database literature. The input is the query, according to the method in the previous two milestones, the join process will be conducted first and then the selection process. And that will be too time-consuming. If we do the selection process on the individual tuple set first, then join the tuple sets according to the where conditions, the time complexity will be much smaller.

2 Implementation

Our strategy for implementation is intuitive. Firstly, we rewrite the input query. Then we do join on the tuple sets.

2.1 Rewrite

When rewriting the query, we basically need to make changes(re-sort) to the "for" clause, "where" clause, and "return" clause.

The text can be extracted by the XQueryParser.FLWRContext class. The interesting part is that the return class of forClause(), whereClause(), returnClause() are different. You have to look into the "black box" to see how to use the method in the returning classes to extract the text string.

To sort the for clauses into different parts,(they are for different tuple sets), we used a list of HashSet to make the classification process more fast. To analyze the "where" clause, and make them corresponding to its tuple set , we used the previous list of HashSet again. The rewrite process on the "return" clause is basically some string operation, which is relatively easy.

2.2 Join

For the join process, we used the methods built in the previous milestones. We used hash join. The key of the hash join is a text string converted from the Node(we've tried to use the node itself as a key, but failed). If the joins are on pairs of attributes,we concatenate their text string as the key.

3 Summary

The best part of the three milestones is that it breaks a large project into small parts and we can learn about how database is working by building the project step by step.

Besides, one rewarding thing is that we have seldom have chances to write codes on a large scope. When we learn algorithms, for example, we write individual code to solve one theory problem. But when you are facing a big project, like the three milestones in this class, you need to consider how to organize your codes in a more efficient and elegant way(or you may get confused when adding different features and functions), and how to make it more readable for your teammates.

Moreover, we have a better understanding of the features of Java language by practicing it on such a project.

Submitted by Liu, Cong, A53207007 Xie, Xianda, A53218476 on Mar 16, 2017.