Chenyang Zhao cz17u22@soton.ac.uk

This report outlines two key optimization steps: logical and physical optimization [1].

Logical optimization mainly adjusts the structure of the syntax tree [2]. First, we need to think about the whole syntax structure with the data structure of the tree, like below.

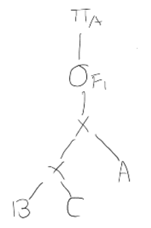

Figure1. Original Tree

The product may create a very huge relation with too many tuples. To solve this, select and project operations are pushed down to the sub-branches, which create a new tree like below.

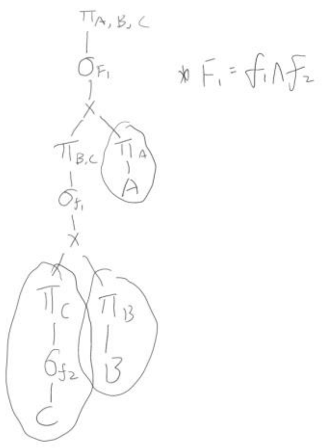

Figure2. New Tree

For the code, we can build a new plan according to the coursework requirement, instead of changing the old plan. All of the branches (sub-plan) in the circles in Figure 2 are the most basic part, they should be executed at first. We can create a new list to store these sub-plans, whose size is equal to the number of relations (scans). Then, we need to decide which select and project can be moved down. According to the following theorem and basic commutative Law of Projection, Selection, and Product, we can freely expand the attributes of the project that needs contains the original attributes and the attributes predicate needs, and move the select that only about one relation down [3].

$$\pi_{A_1,\dots,A_n}(\sigma_F(E)) \equiv \pi_{A_1,\dots,A_n}(\sigma_F(\pi_{A_1,\dots,A_n B_1,\dots,B_m}(E)))$$

Figure3. The commutative law of projection and selection

Then, the sub-plan is finished. The second step is to product the sub-plan together to form a new-level sub-plan and get a top-level plan finally. The following figure is an example of a second-level sub-plan.

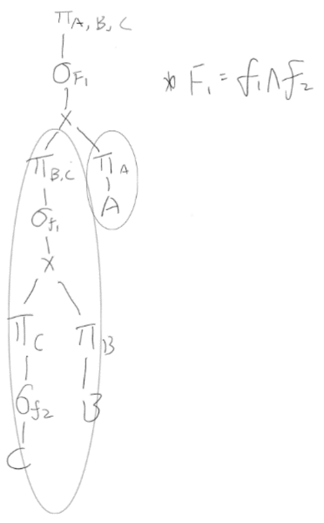

Figure4. Second Level Sub Trees

As for the physical optimization, it is mainly about the create join and the join order, because we cannot know which plan is the best only according to the trees, we need to use an estimator to get the cost and compare to get the fastest plan. If both of the left sub-trees and the right sub-trees of a product are not null, we can change it to a join operator which can move them faster. To determine the order, we need to obtain all combinations of products or join and calculate the cost. After choosing the cheapest cost plan, we got the final new plan.

One notable point is that projecting can increase tuple numbers in this algorithm. However, in real databases, content within blocks can be wasted if tuples are too long. Thus, pre-projection, which reduces tuple length and saves block space, is necessary.

Reference

1. Jarke, M., & Koch, J. (1984). Query optimization in database systems. ACM Computing surveys (CsUR), 16(2), 111-152.
2. Bennett, K. P., Ferris, M. C., & Ioannidis, Y. E. (1991). A genetic algorithm for database query optimization. University of Wisconsin-Madison Department of Computer Sciences.
3. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). Database system concepts.