Report for Software Engineer Intern at FutureWei Technologies

Dakai Du

Abstract

This report describes the internship I spent at FutureWei Technologies from 7/18/2016 to 9/30/2016. In this internship report I will describe my experiences during my internship period. The internship report contains an overview of the internship activities, tasks and projects that I have worked on during my internship. Writing this report, I also will describe and reflect my learning objects and personal goals that I have set during my internship period.

Internship Overview

1. Introduction

The main goal of my internship is develop a database optimizer. The project is based on the open source project GPORCA, the Greenplum Next Generation Query Optimizer. GPORCA is a modular, cost-based query optimizer. Simply put, it takes in a parsed query statement and returns what it considers to be the fastest execution plan for the database. It combines the query with metadata (e.g. statistics, schemas, etc.) and information about the database cluster to generate the execution plan.

Historically, every database has shipped with its own optimizer. That means every software developer spent valuable R&D cycles building one, and maintaining it. This is not a scalable solution, nor does it foster collaborative research. GPORCA is built as an external plugin, which makes it the perfect test bed for database research that can benefit a wide variety of databases. GPORCA is portable and modular because it can work with more than one database and can easily support new database operators. Currently it is possible to run GPORCA with open source projects Greenplum Database and Apache HAWQ (incubating).

2. Task Overview

The GPORCA is written by C++. My job is convert core part of the GPORCA into java so that the optimizer can work on multiplatform. Also, I need to rewrite the relate programs and subroutines to make the database optimizer more efficiency in the right environment.

First of all, I need to understand the general principle of how database optimizer is established and manage to reproduce the procedures of the optimization. After reading the related papers, I find that the biggest difference between GPORCA and other database optimizer is that is is running outside the core database system. Furthermore, the primary optimization techniques are componentized, allowing new operators, transformation, statistical/cost models and other techniques to be added with ease. There are two important aspects are primarily responsible for GPORCA's performance gains: query optimization strategy and common table expression.

Therefore, I split my project into three phases. First, based on the GPORCA tests, capture the main process of the program: extracting query statements, translate into DXL query, pass through pseudo-optimization, translate DXL back into query statement, return result. The second phase is complete the core optimization part. The goal of this part is add those optimization operators, strategies, transformations, statistical models, and other techniques. The last phase is the test and examination. Based on the GPORCA's build-in test samples and newly added samples to test the main function of the database optimizer.

3. Specific Works

Since I have never write java project before and I am not familiar with software development in Linux environment, the first task was to read up and learn about the basic commands and operations of java and Linux. I spent lots of time on learning java language and try to figure out those advanced programming techniques in the original GPORCA. When I look back to the test samples in the original project, I found that I cannot extract the proper context between files. My supervisor Mr. Qingqing Zhou introduced me an important command statement called *strace*, he also showed me a test example to demonstrate it. With the help of him, I have got the basic order of GPORCA's workflow.

During the workflow research, I always stuck with some detail problem which leads me to more problems, especially in the original C++ programs. At the beginning, my idea is try to understand

each pointer, each parameter, each function and each file. Clearly I have underestimate the complexity of the project. Thanks to Mr. Zhou, he told me that it is nearly impossible to totally understand a project like this, what I need to focus on is the what files, functions have been used in each procedure, and ignore the details in those functions, just make sure the input and the output's correctness. His words gave me a new aspect to analyze the test sample procedure and I realized that those problems I have met are not important since I will meet them when I start to convert the original project. Trying to understand all the details in those files have no help with my current goal.

Since I have never write a complete java file before, the first version of my phase 1's result is totally crush. Mr. Zhou recommend me a support software which can convert small C++ program into java version. At first, I do not think that this software is helpful because some simple program it cannot convert correctly, nor to this project with hundreds and thousands of files. Then I found that the software is not as bad as I think. I understand the reason why he introduced this software to me. Clearly this simple software cannot do all the jobs. However, it can provide a lot of useful information. The basic idea of program conversion, the reasons why some statements cannot be translated, to achieve one's function what new features should be introduced and what should be abandoned and so on. The software can help reduce the workload and reveal the problems, traps before they actually affect the project.

4. Summary

Due to my unskilled java coding ability and low efficiency, during the internship, I just nearly complete phase one goal. This was the first time I have experienced using java. Although the syntax is unique, the general layout is similar to those programming language I have used before. What I have mainly done is rebuild the basic procedure of the GPRCA's workflow, including extract query statement, rewrite as DXL format, call the main dump function, pass though the pseudo-optimization program and cost model, rewrite to SQL, then print out the optimized query. The most difficult part of the task was that in order to complete the replacement I needed to avoid the disadvantages in the C++ language and create new functions, ideas. Maybe each problem I have met in the conversion is small, but combine together is considerable. This project tested my understanding of java as well as my understanding of the optimization of database.

Future Works

For continuing this project, once I have got the main procedure of the optimization, I followed the order of each process and signals received from the system, and try to convert the program one file by one file. The disadvantage is that I always met new references and new references leads to more references. For example, the main test program is EresSubtest file, it contain the working environment statements *GPOS*, extraction function EresRunMinidump, different Template functions and inline functions have been referenced in those files. I found that follow one trunk and try to fill up its branches is a available way, but the result is that my memorandum work is bad. I wasted too much time on matching new functions / classes / templates / pseudo-memorypool managements with old finished files. So for people continue my work, he may take lots of time on recording which parts has been finished and which parts needs to be completed.

The future work should be more well-planned. I have put all my works in one single file, including the *GPOS* running environment, IOStream, various templates functions with its inline functions, main test program. They should have been sorted in well in different subdirectories.

Also, try more unit tests. I used to think finish the truck and do the test. However, I should have done more small unit test. This technique is very helpful especially in converting pointers which is a complex problem in conversion since different pointer has different conversion.

Last, always connect with original files. I am still not familiar with Linux commands. Set different stop flag or print command in the original files, check the result with the converted works. It won't save time, but it will be much more easy to find out problems and traps that I have not noticed.

Outcomes

It is really sorry that my final project did not reach the supervisor's expectation. I was not able to complete the build of an available database optimizer. The prototype I have created is not as consummate as I expected and the basic features of GPORCA and suitable running environment/platform are still in developing.

The biggest skill that was enhanced during this internship was the ability to adapt and learn. There were a lot of things that I needed to learn to complete my daily tasks such as learning different programming language and coding skills. I also learned the importance of report. I always try to overcome the problem by myself which leads to the delay of my schedule. In fact, if I submit those problems to the supervisor earlier, some can be resolved easily, some are unnecessary to consider about, some already have solving plan, only small part of them really need me to conquer.

From this internship I gained a lot of information about how software development is used to complete projects. There was a lot of software and command line commands that I used which were new to me. Besides the knowledge that was learned from completing my assigned tasks I gained a lot from talking to my supervisor and fellow workers. They gave a lot of good advice that I will take with me as I prepare to enter the workforce after graduation. I had a great semester working at FutureWei Technologies.

Acknowledge

I would like to thank:

- Qingqing Zhou, Head of Big Data Platform for allowing me to work for him and being a great boss.
- Ziang Hu, for helping me on my work.
- All of the colleagues at FutureWei for being very friendly and helping me on my work.