

Teaching Compiler SQL

Shirin Bhosale, Siddhika More, Seethal Paul, Sebastian Schaefer, Shaheer

July 2024

Contents

1	Introduction	1
2	Methodology	2
3	Literature	2
4	Discussion	2
5	Future Work	2
6	Conclusion	2

Abstract

The abstract will be written at the end when we have written the literature of the paper

1 Introduction

In the modern age, despite the advancements in all aspects of technology, query compilers still face lots of issues. Generally query compilers are very complex and normally there is little to no room for optimizations and flexibility as they are not made to be extensible, with only performance and speed kept in mind. SQL has passed the test of time and still remain a foundational tool for creating and managing relational databases but in today's data driven world these relational databases have increased complexity, therefore there is a growing need for modern compilers that can handle these complexities and also integrate seamless machine learning workflows. This project aims to address this problem by developing a method for modern compilers to understand traditional database operations and modern computation models.

IREE (intermediate Representation Execution Environment) is considered a modern compiler system that is built on MLIR (Multi-Level Intermediate

Representation), which is used to provide a unified, extensible framework for representing and transforming code across abstraction levels. MLIR is also designed to be extensible and customizable, it offers flexibility to run efficiently on different hardware platforms such as CPUs, GPUs and specialized accelerators, this modular nature of makes it more adaptable and achieve high performance. IREE takes advantage of MLIR and supports machine learning models across different platforms, which makes IREE an obvious choice for our project.

Our project starts with parsing the SQL queries with the Pglask libraries which converts the SQL code into an abstract syntax tree (AST), which can then be considered a structured representation of the query. Parsing the SQL is considered a decomposition of the code into fundamental components, and is a crucial step that enables detailed analysis and manipulation of the code to be then used in the subsequent translation into the IREE compatible code.

Lastly, the crucial steps of our project start with parsing the SQL code to generate AST. Next, we aim to convert this parsed SQL into IREE compatible code, we plan to leverage IREE's support for multiple input models including JAX, PyTorch, TensorFlow, ONNX and TOSA. Although it is possible to achieve the results from other input models, but we will be primarily focusing on PyTorch due to its widespread adoption, rich features and robust community support. Finally, our goal is the seamless integration of SQL queries with advance machine learning workflows, our motivation for this project is to enhance data processing and analysis in modern computational environments along with optimizing the performance and scalability of SQL operations.

2 Methodology

3 Literature

4 Discussion

5 Future Work

6 Conclusion