A Study on Query Optimization

Anjali(anjali@wisc.edu) and Sakshi(sbansal8@wisc.edu)

Abstract

Query optimization is the part of the query process in which the database system compares different query strategies and chooses the one with the least expected cost. The query optimizer, which carries out this function, is a key part of the relational database and determines the most efficient way to access data. It makes it possible for the user to request the data without specifying how these data should be retrieved.

The cost of accessing a query is a weighted combination of the I/O and processing costs. The I/O cost is the cost of accessing index and data pages from disk. Processing cost is estimated by assigning an instruction count to each step in computing the result of the query.

In this project, we study the optimization done by PostgreSQL and SQLite by running the queries present in the SSB and TPCH benchmarks. The results obtained will help us to further investigate the query optimization techniques implemented by the two database systems. SQLite is called zero-conf because it does not require service management or access control based on password and GRANT.

1 Introduction

Query optimization is part query processing in many relational database management systems. It determines the most efficient way to execute a particular query by evaluating the various query plans for it. Once the query is submitted to the database server, it is then passed to the parser and then passed to the query optimizer.

2 Databases

2.1 Postgresql

PostgreSQL is a relational database management system following a client-server architecture. At the server side the PostgreSQL's processes and shared memory work together and build an instance, which handles the access to the data. Client programs connect to the instance and request read and write operations.

2.2 SQLite

SQLite is a relational database management system embedded in a C library which can be linked statically or dynamically with the application program. It does not have a client-server database engine which makes the SQLite applications require less configuration.

SQLite is ACID (atomicity, consistency, isolation, durability) compliant and implements the SQL standard. It stores the entire database as a single cross platform file on a host machine.

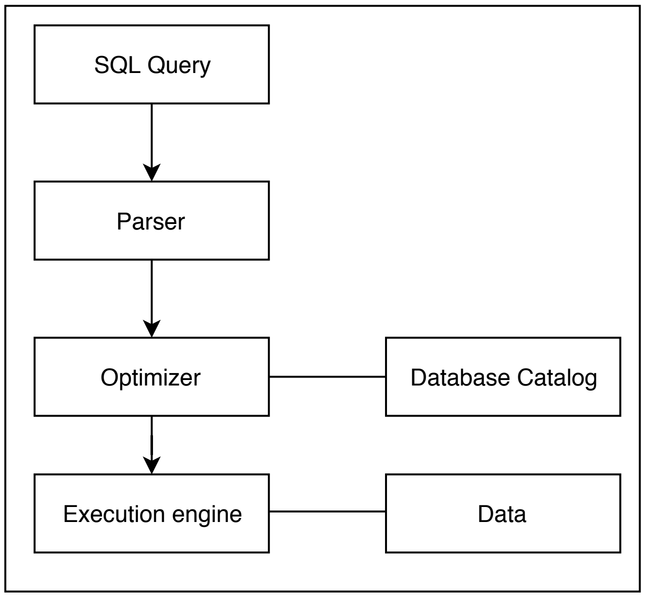

Figure 1. Stages of Query processing)

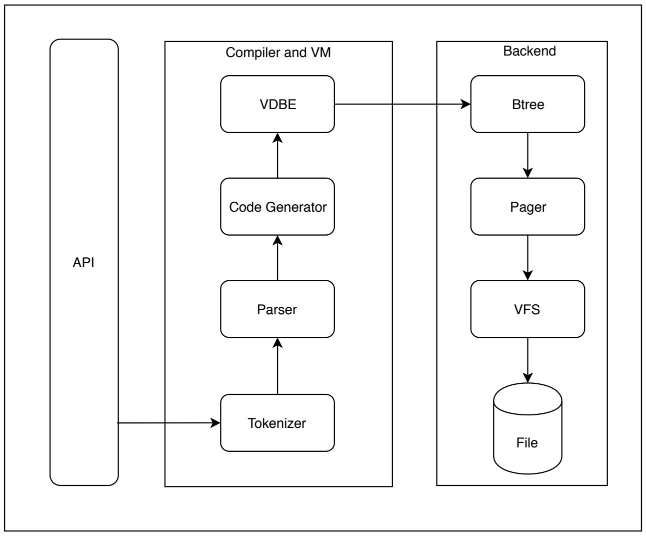

Figure 2. SQLite Architecture)

SQLite database architecture split into two different sections named as core and backend. Core section contains Interface, Tokenizer, Parser, Code generator, and the virtual machine, which create an execution order for database transactions. Backend contains B-tree, Pager and OS interface to access the file system. Tokenizer, Parser and code generator altogether named as the compiler which generates a set of opcodes that runs on a virtual machine.

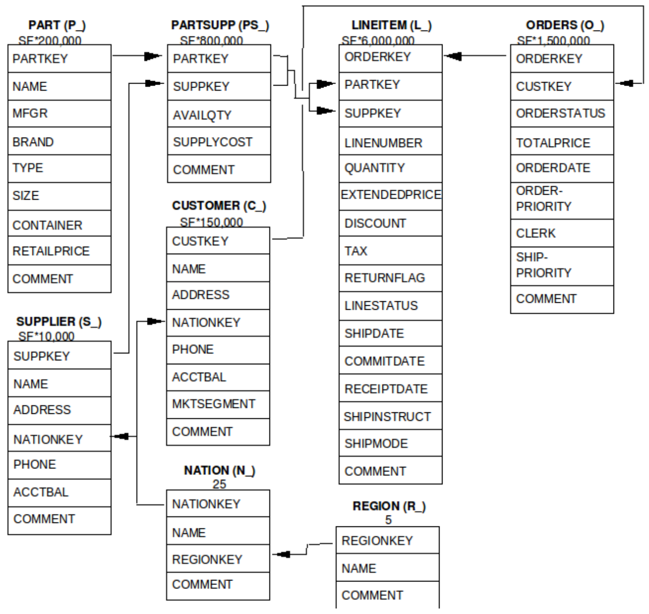

Figure 3. TPCH schema

3 Benchmarks

3.1 TPCH

3.2 SSB

The Star Schema Benchmark (SSB) was designed to test star schema optimization to address the issues outlined in TPC-H with the goal of measuring performance of database products and to test a new materialization strategy. The SSB is a simple benchmark that consists of four query flights, four dimensions, and a simple roll-up hierarchy . The SSB is significantly based on the TPC-H benchmark with improvements that implements a traditional pure star-schema and allows column and table compression.

The SSB is designed to measure the performance of database products against a traditional data warehouse scheme. It implements the same logical data in a traditional star schema whereas TPC-H models the data in pseudo 3NF schema.

Schema modifications were made to the TPC-H schema to transform it into a star schema form. The TPC-H tables LINEITEM and ORDERS are combined into one sales fact table named LINEORDER. The PARTSUPP table is dropped. The comment attributes for LINEITEMS, ORDERS, and shipping instructions are also dropped as a data warehouse does not store such information in a fact table, they canâĂŹt be aggregated and take significant storage space. A dimension table called DATE is added to the schema as is in line with a typical data warehouse. LINEORDER serves as a central fact table. Dimension Tables are created for CUSTOMER, PART, SUPPLIER and DATE.

SSBM concentrates on queries that select from the LINEORDER table exactly once. It avoids the use of self-joins or subqueries as well as or table queries also involving LINEORDER. The classic warehouse query selects from the table with restrictions on the dimension table attributes. SSBM supports queries that appear in TPC-H.SSB consists of one large fact table (LINEORDER) and four dimensions tables (CUSTOMER, SUPPLIER, PART and DATE).