**LOTUS SQL**

**ABSTRACT**

The rapid development of information technology has brought significant progress to human society, and the amount of data that computer system needs to deal with has increased accordingly. SQL is a common choice for data analysis in many scenarios. In recent years, Apache Spark has become the accepted standard for big data processing. SparkSQL is a module offering support for relational analysis on Spark with SQL. It provides convenient data processing interfaces. Despite its efficient optimizer, SparkSQL still suffers from the inefficiency of Spark resulting from JVM and the unnecessary data serialization and deserialization. Adopting native languages such as C++ could help to avoid such bottlenecks. Lotus is a high-performance data parallel computing engine built with C++. Lotus gains high performance because of its bare-mental runtime, compact storage strategy, coarse-grained function call, and memory-efficient design. Although C++ brings significant benefits for Lotus, it also entails extensive programming and debugging efforts. Features in modern C++, such as template usage and automatic type deduction are widely used in Lotus. Although APIs are carefully designed, users might encounter confusing problems when constructing compex applications. Thus, a high-level interface such as SQL is desired. LotusSQL adopts a cost-based optimizer of Calcite and offers cost evaluation methods for Lotus operators to guide the optimization. In addition, LotusSQL extends Calcite’s optimization process to produce good execution plans. This seminar gives an introduction about LotusSQL, Background-Lotus and Calcite, Workflow, Physical operators, Query optimization and Evaluation

Submitted by,

Bibin P Varghese

HKC18CS021

S7 CSE