Systems for Data Science - HW9

SparkSQL - Paper Review

Shubham Shetty

March 15 2021

1 Summary

Spark SQL is a module built over Apache Spark which combines relational processing with Spark's functional programming API. Big Data processing systems like MapReduce were powerful analytical engines, but had additional overheads such as managing various disparate data sources, storage formats, and manual optimizations, which led them to be difficult to program. The Spark SQL abstraction helps in providing a more intuitive and understandable API to ease Big Data development from a programmer's perspective.

Two important components of Spark SQL are -

- DataFrame API - It allows users to intermix procedural and relational code, and also allows users to perform advanced algorithms such as machine learning through UDFs.
- Catalyst - It is an extensible query optimizer. It makes it easier for the developer to add data sources, optimization rules, and data types for domains such as machine learning.

The main goals behind developing Spark SQL were -

- Support relational processing both on Spark internal datasets and external datasets.
- Provide high performance using established DBMS techniques.
- Allow advanced algorithms like graph processing and machine learning.
- Allow Spark to easily connect with external data sources including semi-structured data.

The DataFrame is the basic unit of Spark SQL. Spark runs on RDDs, DataFrames are an additional layer of abstraction over RDDs. A DataFrame is a distributed collection of rows with same schema, which is analogous to a relational database. SparkSQL can then perform basic actions and transactions on these dataframes like it does on RDDs. Operations on dataframes are lazily evaluated.

Spark SQL's extensible query optimizer 'Catalyst' is another important component. It supports both rule-based and cost-based optimization through a tree made up of node objects. Each node has a type and zero or more children. These trees are manipulated using rules, which are functions from one tree to another. Tree transformation generally includes steps like logical plan optimization, physical planning and code generation to compile parts of the source to Java bytecode.

2 Strengths

Following are the strengths of the SparkSQL module -

- The DataFrame API provides a very intuitive and programmer friendly abstraction.
- Supports advanced algorithms such as graph processing and machine learning algorithms.
- Allows to connect with external datasets and perform Spark operations.
- High performance using established DBMS techniques.

3 Unclear Aspects

Following points were unclear with respect to SparkSQL -

- How does the module abstract the underlying datasets? How is semi-structured data supported?
- More details regarding the DataFrame abstraction.
- How SparkSQL behaves with real-life data at a production scale?

4 Limitations / Areas for Improvement

Following are the limitations and areas of improvement for SparkSQL -

- Does not support transactional tables.
- Should provide greater support to load non-relational datasets.
- Should provide more advanced security features.