Spark SQL: Relational Data Processing in Spark

wbg231

January 2023

1 abstract

- new spark module to work with SQL
- kind of bets of both worlds between spark and sql
- makes a tighter integration between Relational and procedural processing through a declarative data frame api that integrates with spark code

introduction

- early cluster computing frameworks like map reduce were lowlevel newer ones are trying to work with relational interfaces
- the Relational model alone is inseficent for big data applications as users may need to write custom code for more complex tasks
- so in practice a mix of self writen code and declerative can be best
- spark sql allows for seemless mixing of relational and procedural paradigms
- spark data frames can preform Relational operations on both external and sparks built in colection
- still keeps lazzy evaluation form RDD
- spark sql also adds a new optimizer called catalyst
- data frame offers rich procedural and declarative integration within spar programs
- data frames are collections of records that can be manipulated using sparks procedural api, and can be built in spark allowing relational processing within spark programs
- can use sql commands on spark sql data frames which saves time if need to do certain tasks like group by
- catalyst uses trees from code optimization and generation

background and goals

spark overview

- rdds are lazy. each rdd is a lgoical plan to compute a data set but spark wiats until certain output operterations are hit

goals for spark sql

- support relational processing within spark (on native rdd) and external data sources
- provide high preformance DBMS techniques
- easily support varied data types
- enable extensions to build off it

programming interface

- spark sql is built ontop of spark

data frame api

- dataframes are a distributed collection of rows with a heterogneous schema
- unlike rdd data frames keep track of there shcema allowing them to be more optimized
- spark data frames like RDD are alos lazy
- data frames can be viewed as an rdd of row objects so rdd methods work on data frames

data model

- spark sql uses a nested data model based on hive
- supports all major sql data type as well as complex data types from spark
- can work with abstract or even use defined data types so very flexible in what data can be models

data frame operations

- users can preform relational operations on data frames using a domain specific language
- data frames only support pre defind functions but in doing so are able to be well optimzed for what they do

data frames vs relational query languages

- while on the surface data frames provide the same operations as relational query languages like sql ,they can be a lot easier to work with since they are already integrated into a full progrmaing language

querying native datasets

- real world pipelines often extract data from heterogneous data sources adn run a wide variety of algorthms on them
- spark data fram can infer schema from rdd objects of arbitary data types and use that for sql queries

user defined functions

- spark sql allows user defined functions to be written in line with the code

catalyst optimizer

- the rest of this seems to get a bit in the weeds so i am going to transition to the lecture