Mining Frequent Queries for Big Data

Keywords: distributed data mining, mapreduce; hadoop.

Abstract

The problem of mining frequent queries in a relational database defined over a star schema is not easy even when we deal with only one table, because, on the one hand, the size of the search space is huge (because encompassing all possible queries that can be addressed to a given database), and on the other hand, testing whether two queries are equivalent (which entails redundant support computations) is NP-Complete. Therefore, the problem is even more difficult when they are applied to Big Data.

In previous work [5, 6, 4, 8, 7, 3], we defined a pre-ordering \((q \leq q')\) on star schema's queries that satisfies the following basic properties:

1. The support measure is anti-monotonic with respect to \(\leq\), and
2. Defining \(q \equiv q'\) if and only if \(q \leq q'\) and \(q' \leq q\)

all equivalent queries have the same support. The anti-monotonicity of pre-ordering defined on star schema's queries allow to apply level-wise algorithm like Apriori [1, 2] and we define an algorithm named FQF (Frequent Query Finder) to find Frequent Queries in database with small size defined over a Star Schema. However, FQF become problematic when they are applied to huge database (BigData). Fortunately, recent improvements in the field of distributed datamining provide good tools to resolve these issues.

The main goal of this thesis is, on the one hand, to investigate the applicability of FQF on the MapReduce platform, and on the other hand, to propose, implement and test an algorithm based on the MapReduce Platform to mine frequent queries in relational database with huge size defined over a star schema.

References


