Usability, Performance and Scalability for Expressive Data Languages via Cardinality-Based Aggregates

Carlo Zaniolo

UCLA Computer Science Department
Los Angeles, CA
USA
Email: zaniolo@cs.ucla.edu

Abstract. In theory, the combined use of recursion and negation allows the declarative expression of very powerful algorithms in languages such as Datalog or SQL. However, the diffculty of writing such programs and proving that they satisfy formal non-monotonic semantics makes such an approach totally impractical for software developers. Thus, we propose a new general solution to the problem of developing advanced applications in logic-based languages. Our approach is based on the combined use of recursion and aggregates endowed with a unified definition of their semantics based on cardinality. In fact, we show that this entails the expression on a wide range of algorithms used in ML, data-mining and graph applications, which can be very appealing for practitioners because of (i) the compactness of their code, (ii) a simplified proof or verification for their stable-model semantics, (iii) their effcient implementations via a max-optimized semi-naive fixpoint algorithm, and (iv) their superior scalability via Stale-Synchronous Parallelism. To demonstrate and further enhance the usability of our proposed framework, including the several efficient and scalable applications we developed, we provide a Logical Algorithm Library (Llib) and a Logical Data-Frame System (LFrame). By integrating access to Llib libray with other Apache Spark libraries, and supporting the interoperability of our BigDatalog, RaSQL, and Datalog-ML systems with Scala, Java and Python, LFrame turns Datalog into a powerful and attractive tool for advanced application development in the Spark ecosystem.