

**High-Level Programming Abstractions for Distributed Graph Processing**

**Abstract:**

Efficient processing of large-scale graphs in distributed environments has been an increasingly popular topic of research in recent years. Inter-connected data that can be modeled as graphs appear in application domains such as machine learning, recommendation, web search, and social network analysis. Writing distributed graph applications is inherently hard and requires programming models that can cover a diverse set of problems, including iterative refinement algorithms, graph transformations, graph aggregations, pattern matching, ego-network analysis, and graph traversals. Several high-level programming abstractions have been proposed and adopted by distributed graph processing systems and big data platforms. Even though significant work has been done to experimentally compare distributed graph processing frameworks, no qualitative study and comparison of graph programming abstractions has been conducted yet. In this survey, we review and analyze the most prevalent high-level programming models for distributed graph processing, in terms of their semantics and applicability. We review 34 distributed graph processing systems with respect to the graph processing models they implement and we survey applications that appear in recent distributed graph systems papers. Finally, we discuss trends and open research questions in the area of distributed graph processing.

**Existing System:**

Writing distributed graph analysis applications is inherently hard. Computation parallelization, data partitioning, and communication management are major challenges of developing efficient distributed graph algorithms. Furthermore, graph applications are highly diverse and expose a variety of data access and communication patterns. For example, iterative refinement algorithms, like PageRank, can be expressed as parallel computations over the local

neighborhood of each vertex. Graph traversal algorithms produce unpredictable access patterns, while graph aggregations require grouping of similar vertices or edges together.

**Proposed System:**

We present an overview of high-level programming abstractions for distributed graph processing and analyze their execution semantics and user-facing interfaces. We further consider performance limitations of some graph programming models and we summarize proposed extensions and optimizations.

For each model, we identify classes of graph algorithms that can be intuitively and easily expressed. Additionally, we look into examples of algorithms that are hard or problematic to express with some graph processing abstractions.

We categorize 34 distributed graph processing systems, with reference to the graph programming models they expose. We discuss execution models and communication mechanisms used.