Big Data Processing (IE494)

Stage-1 Report

T-44

Divyesh Ramani - 202201241

Manan Patel - 202201310

10 October 2024

Distributed Graph Processing Using Apache Spark

Objective:

The objective of this project is to implement a distributed algorithm using Apache Spark that accepts a graph as a query and checks if it is a subgraph of a larger graph. If a match is found, the algorithm will return the nodes of the subgraph, leveraging Spark's parallelism for efficient processing of large-scale graph data.

Plan:

Data Ingestion & Preprocessing

- Load the main graph and the query graph into Apache Spark as distributed datasets.
- Preprocess graphs (e.g., normalizing node labels, sorting edges) to ensure consistency during matching.

Graph Representation

- Convert the graphs into a suitable format. For this project we will be using the Property Graph representation for distributed processing using Spark's RDDs.

Subgraph Isomorphism Algorithm

- The graph given to us as query will be broken down into n segments, where n is the number of nodes in the query graph. After which each node will be matched in the main graph.

Subgraph Matching & Verification

- We will use the node and edge properties stored to match the candidate nodes in the main graph to nodes provided in the query.
- Filter and collect valid subgraphs across the cluster.

Result Collection

- Gather and return the nodes of the matched subgraph(s) from the distributed results.

Work Done So Far:

- Data ingestion and preprocessing of both the main graph and the query graph have been successfully completed.
- The graphs have been transformed into the desired Property Graph representation.
- The query graph has been divided into n segments.
- These segments will be used to probe and match against the main graph during the subgraph detection process.

Reference:

Balaji, Janani, and Rajshekhar Sunderraman. "Distributed graph path queries using spark." 2016 IEEE *40th Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 2. IEEE, 2016.