Automatic Data Analysis of RDF Datasets Using Apache Spark GraphX

Tigran Shahinyan
Institute for Informatics and Automation Problems of NAS RA
Yerevan, Armenia
e-mail: tigran.shahinyan@gmail.com

***Abstract* — RDF semantic data description framework is widely used to describe data in various fields from social networks to different scientific fields. It allows defining information resources and describing knowledge about them. There are many large datasets described in this framework but working with them requires preliminary knowledge about the nature of the data and the techniques for retrieving and reasoning on that data. But to start working with a dataset it would be useful to know about some important properties of it, e.g., the most influential resources, connected components of the graph, etc. To allow this analysis we transform the graph of the dataset into a new one with homogenous resources and import into property graph in GraphX. After that it is possible to run out-of-box or custom algorithms on the property graph. The algorithms are run in distributed environment allowing processing of huge datasets in reasonable time.**

***Keywords* — Distributed computing, semantic web, RDF, Spark.**

I. INTRODUCTION

Graph data structures are one of the widely used data representation approaches. It has some benefits over other ways of describing datasets.

One of the most frequently used standards for describing data as graph structure is Resource Description Framework (RDF) [1].

Linked data is a distributed structured data, which uses HTTP, URI, and RDF for describing it.

There are many providers of linked data datasets who publish their data publicly. The datasets are from various fields from social networks and Wikipedia to biology and health sciences.

One of the main obstacles of intensively using these datasets is their complexity. To retrieve data in RDF datasets, a query language called SPARQL is usually used [2].

SPARQL is a declarative query language, which has many implementations. A query in SPARQL distantly resembles SQL queries in relational databases. It is a powerful tool for working with RDF data.

But working with multiple datasets has a serious burden. The user must have robust knowledge about the schema of the data, which frequently is very complex.

It would be useful to have some initial knowledge about new datasets before diving deeper into their schemas. We provide a solution to process datasets using a distributed graph processing tool GraphX in order to run some basic graph algorithms like PageRank, connected components and triangle count.

II. RDF DATASETS

Authors are encouraged to use LaTeX to prepare their extended abstracts, using the style file and example on the conference web-site. Authors using other means to prepare their abstracts should attempt to duplicate the style of the example as closely as possible.

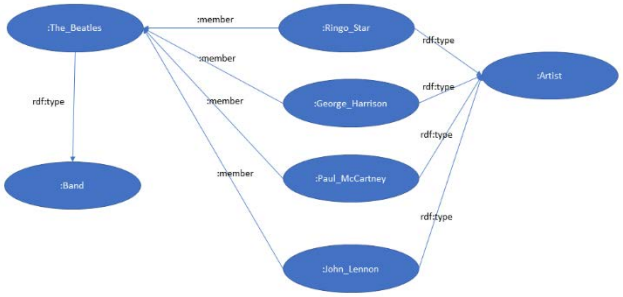
Authors are invited to submit papers in PDF format (template available on the Conference website) by submission system. Accepted papers will be published (up to 4 pages in length) in the Conference Proceedings.

The Resource Description Framework (RDF) is a framework for representing information in the Web. The core structure of the abstract syntax is a set of triples, each consisting of a subject, a predicate, and an object. A set of such triples is called an RDF graph. An RDF graph can be visualized as a node and directed-arc diagram, in which each triple is represented as a node-arc-node link. [1].

Currently there are thousands of open RDF datasets, e.g., DBPedia, Uniprot, etc.

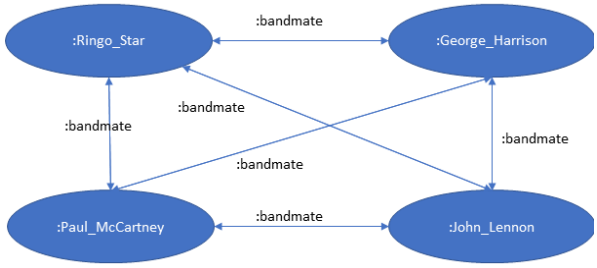
RDF graphs represent data about objects of various types and their relationships. To run the graph algorithms on such graphs we need to do some transformations either by getting rid of nodes “unimportant” nodes or transforming them into edges between nodes of interesting type.

As an example, let’s consider the transformation of the following graph representing rock bands:



It consists of artists and bands, and each artist can be a member of zero or more band. In our examples, there are four artists who are members of the same band.

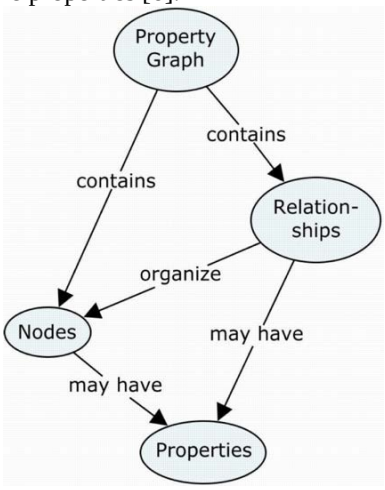
If we are interested in artists and their relationships, we need to get rid of other types, e.g., bands, leaving the relationships between artists, which we represent as a new predicate ":bandmate". After the transformation, our new graph will look like this:



III. Apache GraphX and Property Graphs

Apache GraphX is a distributed graph computation framework that unifies graph-parallel and data-parallel computation. GraphX provides a small, core set of graph-parallel operators expressive enough to implement the Pregel and PowerGraph abstractions, yet simple enough to be cast in relational algebra. GraphX uses a collection of query optimization techniques such as automatic join rewrites to efficiently implement these graph-parallel operators [3,4].

In GraphX data is represented as property graphs. A property graph is a type of graph model, where relationships not only are connections but also carry a name (type) and some properties [6].



Apache GraphX stores data in two immutable distributed RDDs: VertexRDD and EdgeRDD [6].

GraphX allows distributed storage and processing of graph data using Spark's RDD mechanisms.

There are several algorithms provided by GraphX out of the box, which can be applied to our data:

PageRank, which measures the importance of each vertex in a graph,

Connected Components, which labels each connected component of the graph with the ID of its lowest-numbered vertex, thus allowing to detect whether two vertices are in the same connected component,

Triangle Counting, where a vertex is part of a triangle when it has two adjacent vertices with an edge between them.

IV. Converting RDF Datasets into Property Graphs

Our solution provides a mechanism to convert the RDF dataset into a property graph by storing the data in VertexRDD and EdgeRDD.

After storing the graph data in GraphX we can run available algorithms to detect some general features of our homogeneous graph structure.

Having information about the most "important" nodes or connected components of the graph will help the user to have some preliminary knowledge about the dataset.

V. Conclusions and Future Work

The solution shows that it is possible to use distributed technologies to improve working with complex datasets like large RDF graph datasets.

The work is the first step to a general solution, which will allow loading any graph datasets into the distributed environment and run graph algorithms. In our work we use several simple algorithms like PageRank and Connected Components, but more algorithms will be added specified to the field of discourse.

References

[1] Richard Cyganiak, David Wood, Markus Lanthaler, RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation, 25 February 2014.

[2] S. Harris, A. Seaborne, SPARQL 1.1 Query Language, W3C Recommendation, 21 March 2013.

[3] Xin, R.S., Crankshaw, D., Dave, A., Gonzalez, J.E., Franklin, M.J., Stoica, I.: Graphx: Unifying data-parallel and graph-parallel analytics. CoRR arxiv:1402.2394, 2014.

[4] Reynold S. Xin, Joseph E. Gonzalez, Michael J. Franklin, Ion Stoica, GraphX: A Resilient Distributed Graph System on Spark, Proceedings of the First International Workshop on Graph Data Management Experience and Systems (GRADES), June 23, 2013, New York.

[5] What Is a Property Graph? Michelle Knight, DataVersity, April 28, 2021.

[6] Zaharia, Matei & Xin, Reynold & Wendell, Patrick & Das, Tathagata & Armbrust, Michael & Dave, Ankur & Meng, Xiangrui & Rosen, Josh & Venkataraman, Shivaram & Franklin, Michael & Ghodsi, Ali & Gonzalez, Joseph & Shenker, Scott & Stoica, Ion. Apache spark: A unified engine for big data processing. Communications of the ACM November 2016, vol. 59, no. 11, pp. 56-65.