Big graph analysis by visually created workflows

M. Ali Rostami[1] Eric Peukert[1] Moritz Wilke[1] Erhard Rahm[1]

Abstract: The analysis of large graphs has received considerable attention recently but current solutions are typically hard to use. In this demonstration paper, we report on an effort to improve the usability of the open-source system Gradoop for processing and analyzing large graphs. This is achieved by integrating Gradoop into the popular open-source software KNIME to visually create graph analysis workflows, without the need for coding. We outline the integration approach and discuss what will be demonstrated.

Keywords: Graph analysis; Workflow; Gradoop; KNIME; Visualization

1 Introduction

The analysis of big (graph) data becomes increasingly relevant for a variety of fields from bioinformatics to business intelligence. Big data processing frameworks make the computational part of this task possible but mostly require advanced software development and data engineering skills, which many data analysts and scientists lack. Workflow-oriented tools with graphical interfaces promise an easier usage for this group of users but often lack the ability to tackle large amounts of data. Therefore it is important to combine both approaches: user-friendly creation of data analysis workflows with the scalability of a big data processing framework in the background.

Gradoop [Ju15; Ju18] is an open-source framework for the analysis of large graphs that is developed by the Database Group of the University Leipzig and the Competence Center for Scalable Data Services and Solutions (ScaDS) Leipzig. It uses an extension of the flexible property graph model and provides a variety of graph operators and graph mining algorithms. It is build on top of Apache Flink for parallel processing and scalability to large data volumes. KNIME [Ba07], on the other hand, is a well-known open source software for data analysis for the creation and deployment of analytical workflows and applications. KNIME allows users with minimal experience in writing code to select a large set of different operators as nodes from a graphical interface and to combine them on a virtual canvas to define executable workflows.

To achieve both scalability and usability, we developed a KNIME integration for Gradoop operators within the BIGGR project [Ro19]. The aim of this demonstration paper is to show

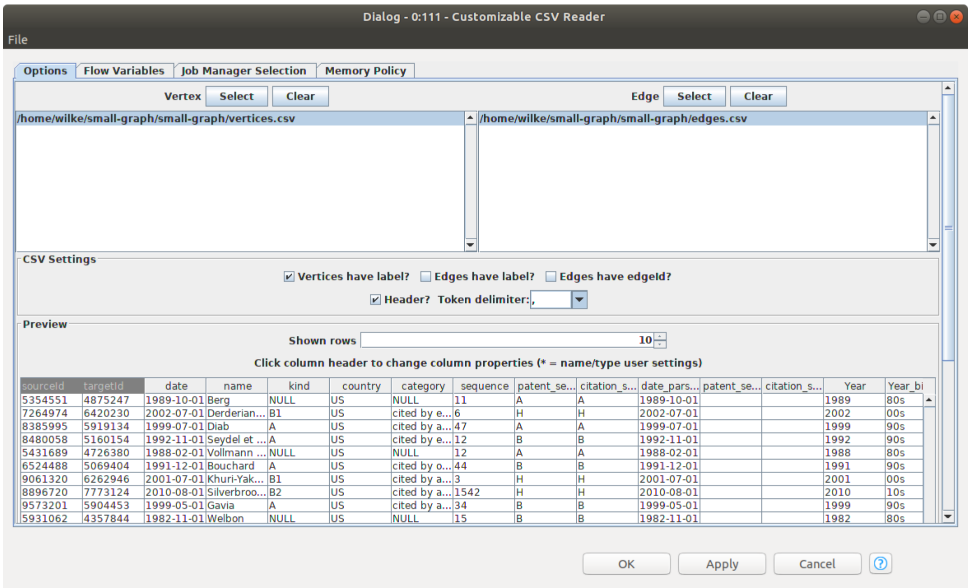
[1] University of Leipzig, ScaDS Dresden Leipzig, Augustusplatz 10, 04109 Leipzig, Germany, {rostami, peukert, wilke, rahm}@informatik.uni-leipzig.de

doi:10.18420/btw2019-45

how the tools KNIME and Gradoop can be used in combination to load, transform and analyze large graph data and to visualize the results.

2 Gradoop Integration in KNIME

The integration approach makes the Gradoop operators available as nodes within Knime and allows either the local or the remote execution of the Gradoop operators and workflows [Ro19]. The majority of real-word data is still stored in column based or unstructured formats, not in a property graph format which is directly readable with Gradoop. Hence the first task in a graph-based analysis is to transfer given data in a graph format, e.g. to specify which rows in a table are regarded as *vertices* and which relations qualify as *edges*. Knime supports various ways of reading and processing data and is build around a tabular data representation, so the data preparation step can be tackled with it. However, it requires knowledge about the internal data format of Gradoop which is a hurdle.



/home/wilke/small-graph/small-graph/vertices.csv

/home/wilke/small-graph/small-graph/edges.csv

CSV Settings

☑ Vertices have label? ☐ Edges have label? ☐ Edges have edgeId?

☑ Header? Token delimiter: ,

Preview

Shown rows 10

Click column header to change column properties (* = name/type user settings)

sourceId	targetId	date	name	kind	country	category	sequence	patent_se...	citation_s...	date_pars...	patent_se...	citation_s...	Year	Year_bi
5354551	4875247	1989-10-01	Berg	NULL	US	NULL	11	A	A	1989-10-01			1989	80s
7264974	6420230	2002-07-01	Derderian...	B1	US	cited by e...	6	H	H	2002-07-01			2002	00s
8385995	5919134	1999-07-01	Diab	A	US	cited by a...	47	A	A	1999-07-01			1999	90s
8480058	5160154	1992-11-01	Seydel et ...	A	US	cited by e...	12	B	B	1992-11-01			1992	90s
5431689	4726380	1988-02-01	Vollmann ...	NULL	US	NULL	12	A	A	1988-02-01			1988	80s
6524488	5069404	1991-12-01	Bouchard	A	US	cited by o...	44	B	B	1991-12-01			1991	90s
9061320	6262946	2001-07-01	Khuri-Yak...	B1	US	cited by a...	3	H	H	2001-07-01			2001	00s
8896720	7773124	2010-08-01	Silverbroo...	B2	US	cited by a...	1542	H	H	2010-08-01			2010	10s
9573201	5904453	1999-05-01	Gavia	A	US	cited by a...	34	B	B	1999-05-01			1999	90s
5931062	4357844	1982-11-01	Welbon	NULL	US	NULL	15	B	B	1982-11-01			1982	80s

Fig. 1: CSV import: a graph of patent citations is read from a file containing the vertices and a file which contains the source and target id of a citation.

To overcome this, data transformation nodes between Gradoop and Knime are necessary. As a first working solution, we provide a Knime node for importing CSV data and transforming into a property graph to start an analysis workflow. Fig. 1 shows the CSV import functionality: multiple files can be chosen and configured to be transformed into sets of vertices or edges. Further efforts will deal with supporting recently proposed operators for graph transformation [KPR19] and to exchange data between the local machine running KNIME and a cluster which stores graph data in a distributed manner.

3 Demonstration

In our demonstration, we show workflows of large graph analytics using KNIME and Gradoop. We consider two datasets: the medium-sized citation networks of research papers and authors as well as a large patent citation graph consisting of 6.6 million US patents and 94.7 million citations between them.[2]

We build several workflows for the analysis of these large graphs. First, we see how a summary of graphs can be generated as a starting point for further operators. The patent graph needs several steps of preprocessing to obtain a handy set of data. For example, we can first determine connected components and apply further operators on selected components. The resulting graphs from different processes are visualized either completely if possible or in a clustered way. We see how the visualizations give us a first insight into data and help to decide on further analysis steps.

3.1 Research Papers

In this workflow, we first read a publication graph from a JSON data source. After a reduction, the connected components are computed and one of them is visualized. There are two visualization nodes, called **Big Graph View**. For an interactive visualization, large graphs are written out to some format (like CSV) and the path will be given to the view nodes.

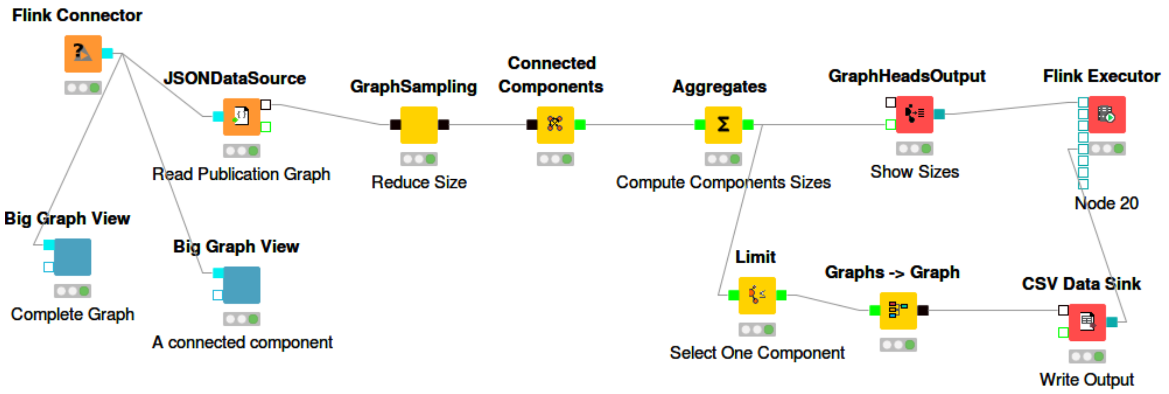


Fig. 2: A Gradoop workflow in KNIME.

3.2 US Patent - Citation Network

Goal of the analysis is to find and collapse so-called patent families. This structure reflects the patent holders not to claim a single patent for a new invention, but to submit several

[2] http://www.patentsview.org

slightly modified patents in order to prevent rivals from adopting the invention easily. These patent families clutter the space of patents, hence clustering them makes further analysis easier because it reduces the size and complexity of the graph.

The workflow uses a Gradoop matching operator which uses the Cypher[Ju17] query (listing 1) language to detect pairs of patents from the same assignee that have been claimed in the same year and are connected with a citation.

```
MATCH (p1:PATENT)<-[e1:citation]-(p2:PATENT)
      (p1)-[:assignedBy]->(a1:ASSIGNEE)<-[:assignedBy]-(p2)
WHERE e1.years_difference = 0
```

List. 1: Cypher-query to identify citations between patents from the same assignee in the same year.

Afterwards the connected components algorithm is used to combine the pairs into stars that are assumed to be a patent family. Finally the families can be fused in the original graph resulting in only a single vertex representing what was a star before. both graphs (patent families and reduced graph) are stored for further analysis and processing afterwards.

4 Acknowledgements

This work was funded by the German Federal Ministry of Education and Research within the projects BIGGR (BMBF 01IS16030B) and ScaDS Dresden/Leipzig (BMBF 01IS14014B).

References

[Ba07] Berthold, M. R.; et al.: KNIME: The Konstanz Information Miner. In: Proc. 31st Annual Conference of the Gesellschaft für Klassifikation e.V. Pp. 319–326, 2007.

[Ju15] Junghanns, M.; Petermann, A.; Gómez, K.; Rahm, E.: Gradoop: scalable graph data management and analytics with Hadoop, 2015, arXiv: 1506.00548.

[Ju17] Junghanns, M.; Kießling, M.; Averbuch, A.; Petermann, A.; Rahm, E.: Cypher-based graph pattern matching in Gradoop. In: Proc. Fifth International Workshop on Graph Data-management Experiences & Systems. ACM, p. 3, 2017.

[Ju18] Junghanns, M.; Kiessling, M.; Teichmann, N.; Gómez, K.; Petermann, A.; Rahm, E.: Declarative and distributed graph analytics with GRADOOP. Proceedings of the VLDB Endowment 11/12, pp. 2006–2009, 2018.

[KPR19] Kricke, M.; Peukert, E.; Rahm, E.: Graph data transformations in GRADOOP. In: Proc. BTW Conference. 2019.

[Ro19] Rostami, M. A.; Kricke, M.; Peukert, E.; Kuehne, S.; Wilke, M.; Dienst, S.; Rahm, E.: BIGGR: Bringing Gradoop to Applications. Datenbank-Spektrum 19/1, to appear, 2019.