University of Glasgow | School of Computing Science

Implementation of Novel Subgraph Query Processing methods within GraphX

Iva Babukova

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March 20, 2016

Abstract

The GraphX system has recently been developed at Berkeley, over the Spark massively-parallel data processing system, as a system for high performance analytics over graph data. It is currently an important tool for graph-analytic tasks, which are core to many data science endeavours. At the same time, graph datasets have become increasingly popular, used to model applications from numerous domains from social networks to biology and bioinformatics. The goal of this project is to design, implement, and test an algorithm for subgraph queries, on top of GraphX.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

Chapter 1

Introduction

The first page, abstract and table of contents are numbered using Roman numerals. From now on pages are numbered using Arabic numerals. Therefore, immediately after the first call to \chapter we need the call \pagenumbering{arabic} and this should be called once only in the document.

The first Chapter should then be on page 1. You are allowed 50 pages for a 30 credit project and 35 pages for a 20 credit report. This includes everything up to but excluding the appendices and bibliograph, i.e. this is a limit on the body of the report.

You are not allowed to alter text size (it is currently 11pt) neither are you allowed to alter the margins.

Note that in this example, and some of the others, you need to execute the following commands the first time you process the files. Multiple calls to pdflatex are required to resolve references to labels and citations. The file bib.bib is the bibliography file.

```
> pdflatex example0
> bibtex example0
> pdflatex example0
> pdflatex example0
```

1.1 First Section in Chapter

Owner/Supervisor: Peter Triantafillou *Suitable as a Software Engineering project.

Description: The GraphX system has recently been developed at Berkeley, over the Spark massively-parallel data processing system, as a system for high performance analytics over graph data. It is currently an important tool for graph-analytic tasks, which are core to many data science endeavours.

At the same time, graph datasets have become increasingly popular, used to model applications from numerous domains from social networks to biology and bioinformatics.

This is a project best suited for one or a team of two L4 students. The goal of this project is to design, implement, and test a (set of) algorithms for subgraph queries, on top of GraphX. As subgraph querying currently

escapes GraphX, we hope with this project to contribute, test, and evaluate such a solution, based on recent research results achieved by our group's PhD students.

The L4 students will be closely collaborating with PhD students to implement subgraph processing methods into GraphX.

[1] https://amplab.cs.berkeley.edu/tag/spark/ (see the GraphX related papers in this site).

Special Requirements : Strong coding skills are required.

Chapter 2

The Fox and Dog

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

2.1 The Fox Jumps Over

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over Uroborus (Figure 2.1).

The quick brown fox jumped over the [2] lazy dog. [5] The quick brown fox jumped over the lazy dog. [4] The quick brown fox jumped over [1] the lazy dog. The quick brown fox jumped over [3] the lazy dog.

2.2 Assessment criteria

Has the student surveyed relevant research literature? Has he/she analysed the research problem, and devised a suitable approach for solving it?

Has the research been conducted well? Does it show evidence of original thinking? Are there significant errors? Might the research be worth of publication, perhaps after revision?

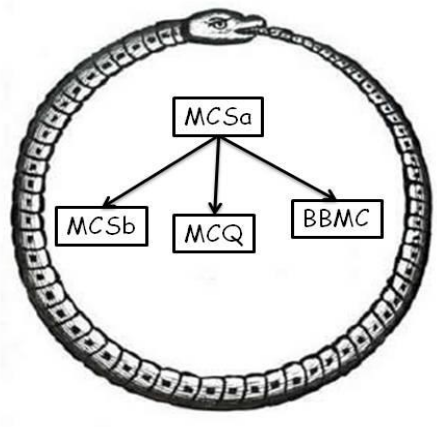

Figure 2.1: An alternative hierarchy of the algorithms.

Has the student critically evaluated and analysed the research results? Does he/she understand their significance? Does he/she have good suggestions for further work?

Is the dissertation complete, well organised, and literate? Does it clearly explain the problem, and how the software was designed, implemented, tested, and evaluated? Does it contain a bibliography and proper citations?

Did the student complete a one-page summary of the project satisfactorily? Did the student attend meetings, and engage effectively with the supervisor?

Did the content reflect a knowledge and understanding of the work done? Were questions handled well? Were visual aids used effectively? Was the delivery fluent and confident, with good eye contact?

Chapter 3

Introduction

3.1 Graph Databases

3.2 Subgraph Isomorphism

3.3 Graph Indexing

3.4 Spark and GraphX

Chapter 4

Graph Indexing Algorithms

4.1 Structure-based approach

4.2 some other technique

Chapter 5

Tools

5.1 Spark

5.1.1 Resilient Distributed Datasets

5.2 GraphX

Chapter 6

Implementation

6.1 Choice of algorithm

6.2 Choice of programming language

6.3 Graph Data Structures

6.3.1 Structures for the implementation in Java

6.3.2 Structures for the implementation in Spark

Chapter 7

Experimental Evaluation

7.1 Exprimental Data and Methodology

7.2 Comparison of Spark implementation to Java implementation

Chapter 8

Conclusion

Appendices

Appendix A

Running the Programs

An example of running from the command line is as follows:

```
> java MaxClique BBMC1 brock200_1.clq 14400
```

This will apply $BBMC$ with $style = 1$ to the first brock200 DIMACS instance allowing 14400 seconds of cpu time.

Appendix B

Generating Random Graphs

We generate Erdós-Rënyi random graphs $G(n, p)$ where n is the number of vertices and each edge is included in the graph with probability p independent from every other edge. It produces a random graph in DIMACS format with vertices numbered 1 to n inclusive. It can be run from the command line as follows to produce a clq file

```
> java RandomGraph 100 0.9 > 100-90-00.clq
```

Bibliography

[1] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the really hard problems are. In *Proceedings IJCAI'91*, pages 331–337, 1991.

[2] P. Mutzel K. Klein, N. Kriege. Ct-index: Fingerprint-based graph indexing combining cycles and trees. *Data Engineering (ICDE), 2011 IEEE 27th International Conference, Hannover*, pages 1115 – 1126, 11-16 April 2011.

[3] Reynold Xin, Joseph Gonzalez, Daniel Crankshaw, Michael Franklin, Ankur Dave, and Ion Stoica. Graphx: Unifying data-parallel and graph-parallel analytics. February 2014.

[4] Xifeng Yan, Philip S. Yu, and Jiawei Han. Graph indexing: A frequent structure-based approach, 2004.

[5] Matei Zaharia, Mosharaf Chowdhury, Ankur Dave, Michael Franklin, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. *NSDI 2012*, 2012.