TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science, Chair of Privacy and Data Security

Graph-based Analysis of Dynamic Systems

Dissertation

Submitted to the Faculty of Computer Science, TU Dresden,
in Partial Fulfillment of the Requirements for the Degree of Dr.-Ing.

by
Dipl.-Inform., Dipl.-Math. **Benjamin Schiller**
born on 23 February 1983 in Frankfurt Höchst, Germany

Committee members:

Prof. Dr.-Ing. Thorsten Strufe	TU Dresden, Dresden, Germany
Prof. Dr.-Ing. Wolfgang Lehner	TU Dresden, Dresden, Germany
Prof. Dr. George Fletcher	Eindhoven University of Technology, Eindhoven, Netherlands

Dresden, October 2016

Abstract

The analysis of dynamic systems provides insights into their time-dependent characteristics. This enables us to monitor, evaluate, and improve systems from various areas. They are often represented as graphs that model the system's components and their relations. The analysis of the resulting dynamic graphs yields great insights into the system's underlying structure, its characteristics, as well as properties of single components. The interpretation of these results can help us understand how a system works and how parameters influence its performance. This knowledge supports the design of new systems and the improvement of existing ones.

The main issue in this scenario is the performance of analyzing the dynamic graph to obtain relevant properties. While various approaches have been developed to analyze dynamic graphs, it is not always clear which one performs best for the analysis of a specific graph. The runtime also depends on many other factors, including the size and topology of the graph, the frequency of changes, and the data structures used to represent the graph in memory. While the benefits and drawbacks of many data structures are well-known, their runtime is hard to predict when used for the representation of dynamic graphs. Hence, tools are required to benchmark and compare different algorithms for the computation of graph properties and data structures for the representation of dynamic graphs in memory. Based on deeper insights into their performance, new algorithms can be developed and efficient data structures can be selected.

In this thesis, we present four contributions to tackle these problems: A benchmarking framework for dynamic graph analysis, novel algorithms for the efficient analysis of dynamic graphs, an approach for the parallelization of dynamic graph analysis, and a novel paradigm to select and adapt graph data structures. In addition, we present three use cases from the areas of social, computer, and biological networks to illustrate the great insights provided by their graph-based analysis.

We present a new benchmarking framework for the analysis of dynamic graphs, the *Dynamic Network Analyzer* (DNA). It provides tools to benchmark and compare different algorithms for the analysis of dynamic graphs as well as the data structures used to represent them in memory. DNA supports the development of new algorithms and the automatic verification of their results. Its visualization component provides different ways to represent dynamic graphs and the results of their analysis.

We introduce three new stream-based algorithms for the analysis of dynamic graphs. We evaluate their performance on synthetic as well as real-world dynamic graphs and compare their runtimes to snapshot-based algorithms. Our results show great performance gains for all three algorithms. The new stream-based algorithm StreaM$_k$, which counts the frequencies of k-vertex motifs, achieves speedups up to 19,043× for synthetic and 2882× for real-world datasets.

We present a novel approach for the distributed processing of dynamic graphs, called *parallel Dynamic Graph Analysis* (pDNA). To analyze a dynamic graph, the work is distributed by a partitioner that creates subgraphs and assigns them to workers. They compute the properties of their respective subgraph using standard algorithms. Their results are used by the collator component to merge them to the properties of the original graph. We evaluate the performance of pDNA for the computation of five graph properties on two real-world dynamic graphs with up to 32 workers. Our approach achieves great speedups, especially for the analysis of complex graph measures.

We introduce two novel approaches for the selection of efficient graph data structures. The compile-time approach estimates the workload of an analysis after an initial profiling phase and recommends efficient data structures based on benchmarking results. It achieves speedups of up to 5.4× over baseline data structure configurations for the analysis of real-word dynamic graphs. The run-time approach monitors the workload during analysis and exchanges the graph representation if it finds a configuration that promises to be more efficient for the current workload. Compared to baseline configurations, it achieves speedups up to 7.3× for the analysis of a synthetic workload.

Our contributions provide novel approaches for the efficient analysis of dynamic graphs and tools to further investigate the trade-offs between different factors that influence the performance.

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. This thesis was not previously presented to another examination board and has not been published in this form. I am aware that a false statement entails legal consequences.

Benjamin Schiller	City, Date

Acknowledgements

Over the past years, I was blessed with the continuous support and encouragement of a number of people. Without all of their help and support, this thesis would not have been possible.

First and foremost, I like to thank my adviser Thorsten Strufe. I learned a lot from him about doing research, managing projects, and transforming ideas into papers. He allowed me to pursue interesting new ideas and pushed me to do things the right way. He always made time to discuss new ideas and answered the many questions I had about everything. I am very thankful to George Fletcher who agreed to review this thesis as the external referee. I also like to thank Wolfgang Lehner for being my Fachreferent.

The contributions in this thesis would not have been possible without the help and input from collaborators and colleagues. For their input, the lively discussions, and the collaborations in various projects, I thank Jeronimo Castrillon, Clemens Deusser, Kay Hamacher, Irina Heimbach, Oliver Hinz, Sven Jager, Dirk Kohlweyer, Giang Nguyen, Stefanie Roos, Hani Salah, and Jan Seedorf.

I am grateful to the following colleagues and friends for taking the time to read parts of this thesis and helping me to improve it based on their comments: Clemens Deusser, Jan Reubold, Stefanie Roos, Sven Jager, Cornelius Stöhr, and Réné Wilmes.

Over the past years, I had the pleasure of working closely with a large number of students. Many of them extended the DNA framework with their own implementations. For their work, I thank Nico Haase, Benedict Jahn, Christoph Schott, and Marcel Wunderlich. A special thanks goes out to Réné Wilmes for implementing large parts of the framework's visualization and data processing components.

This thesis would not have been possible without $C_8H_{10}N_4O_2$ and $C_{25}H_{45}N_5O_{13}$ as well as the joy I got from the work of people like Tom Araya, Jack Black, Geezer Butler, Eric Clapton, Kyle Gass, David Gilmour, Greg Graffin, Toni Iommi, Jamey Jasta, Lemmy Kilmister, Ivan L. Moody, Ozzy Osbourne, John Petrucci, Mike Portnoy, Joey Ramone, Patti Smith, Michael Starr, Corey Taylor, Bill Ward, Steve Winwood, and Frank Zappa (just to name a few).

Last but not least, I thank my family and friends for being there for me, all the time. Thanks to my parents for always believing in me, enabling me to pursue my own path, and always encouraging me to do the things I love. Thanks to Thomäs for supporting me and standing by my side in countless situations. Thanks to Laura for being my Schnecksche.

Contents

List of Figures

List of Tables

List of Listings

List of Algorithms