Implementing Graph Transformations in the Bulk Synchronous Parallel Model

Christian Krause
SAP SE
Potsdam,
Germany
christian.krause01@sap.com

Matthias Tichy
Chalmers | University of
Gothenburg,
Sweden
matthias.tichy@cse.gu.se

Holger Giese
Hasso Plattner Institute,
University of Potsdam,
Germany
holger.giese@hpi.uni-potsdam.de

Abstract: Big data becomes a challenge in more and more domains. In many areas, such as in social networks, the entities of interest have relational references to each other and thereby form large-scale graphs (in the order of billions of vertices). At the same time, querying and updating these data structures is a key requirement. Complex queries and updates demand expressive high-level languages which can still be efficiently executed on these large-scale graphs.We use graph transformation rules and units as a high-level modeling language with declarative and operational features for transforming graph structures. To apply them to large-scale graphs, we introduce a method to distribute and parallelize graph transformations by mapping them to the Bulk Synchronous Parallel model. Our tool support builds on Henshin as modeling tool and consists of a code generator for Apache Giraph. We evaluated our approach with the IMDb movie database on a cluster with 24 servers with 8 cores each.

1 Introduction

Graph-based modeling and analysis becomes relevant in an increasing number of domains and in many of these areas the big-data dimension of the graph processing problem is a limiting factor for existing modeling and analysis approaches. There is a demand for high-level, declarative modeling languages which, at the same time, must be executed efficiently also on large-scale graphs in the order of millions or even billions of vertices.

In this paper, we map the high-level modeling concepts of algebraic graph transformations [EEPT06] to the bridging model *Bulk Synchronous Parallel* (BSP) [Val90] which provides an abstraction layer for implementing parallel algorithms on distributed data. Thereby, we enable the use of the expressive language concepts of graph transformations for analytical processing of large-scale graph data. In our prototypical tool suppport we use the Henshin [ABJ$^+$10] graph transformation language and tool to specify transformation rules and units. We have implemented a code generator that takes Henshin models as input and generates code for the BSP-based framework Apache Giraph[1] which builds on the distributed file system and map-reduce infrastructure of Apache Hadoop[1]. Our code generator is available as part of Henshin[2].

[1] Apache Giraph/Hadoop are available at *http://giraph.apache.org/* and *http://hadoop.apache.org/*

[2] Henshin is available at *https://www.eclipse.org/henshin/*

2 Evaluation

We solved parts of the TTC 2014 movie database challenge [HKT14] with our prototype. This challenge uses the data of the IMDb database comprising 0.9 mio movies and 2.8 mio actors and actresses. The goal of this part of the challenge is to find all pairs of actors/actresses that played together in at least three movies. We use the Henshin transformation units and rules shown in Fig. 1. First, we create couple vertices for every actor pair. Second, we create edges to the common movies and calculate their average rating.

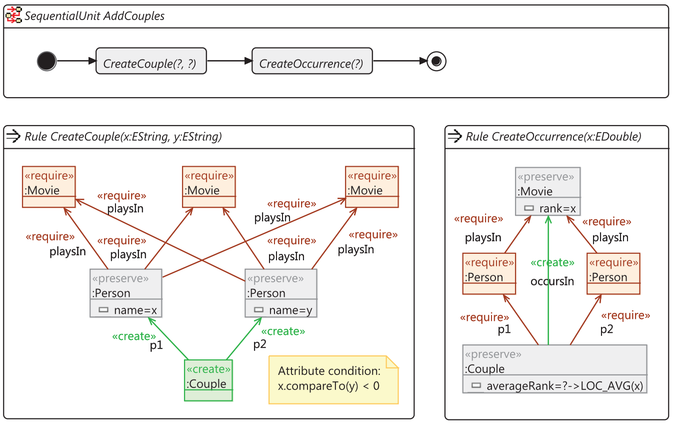

Figure 1: Graph transformation units/rules for movie database example

Workers	Run-time
5	1,220
6	866
8	666
10	587
12	518
14	453
16	430
18	401
20	440
22	387
24	373

Figure 2: Run-times in seconds

We generated Giraph code for a simplified version of this example and executed it on a cluster of 24 compute nodes with 2 AMD Opteron® 6220 processors (8 cores each at 3GHz) with 32GB of main memory (restricted to 10GB). Fig. 2 shows the resulting execution times averaged over 3 runs. The numbers indicate a good horizontal scalability until 14 nodes are used. More nodes still benefit the execution, but due to increasing fixed costs and communication overhead, the performance does not increase at the same rate.

References

[ABJ+10] T. Arendt, E. Bierman, S. Jurack, C. Krause, and G. Taentzer. Henshin: Advanced Concepts and Tools for In-Place EMF Model Transformations. In *MoDELS 2010*, LNCS 6394, pages 121–135. Springer, 2010. DOI: 10.1007/978-3-642-16145-2_9.

[EEPT06] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation (Monographs in Theor. Comp. Sci. - An EATCS Series)*. Springer, 2006.

[HKT14] T. Horn, C. Krause, and M. Tichy. The TTC 2014 Movie Database Case. In *TTC 2014*. CEUR-WS, 2014.

[KTG14] C. Krause, M. Tichy, and H. Giese. Implementing Graph Transformations in the Bulk Synchronous Parallel Model. In *FASE'14*, LNCS 8511, pages 325–339. Springer, 2014. DOI: 10.1007/978-3-642-54804-8_23.

[Val90] L. G. Valiant. A bridging model for parallel computation. *Commun. ACM*, 33(8):103–111, 1990. DOI: 10.1145/79173.79181.