Distributed Graph Analytics

Unnikrishnan Cheramangalath • Rupesh Nasre •
Y. N. Srikant

Distributed Graph Analytics

Programming, Languages, and Their Compilation

Springer

Unnikrishnan Cheramangalath
Department of Computer Science
and Engineering
Indian Institute of Technology Palakkad
Palakkad, Kerala, India

Rupesh Nasre
Department of Computer Science
and Engineering
Indian Institute of Technology Madras
Chennai, India

Y. N. Srikant
Department of Computer Science
and Automation
Indian Institute of Science
Bangalore, India

ISBN 978-3-030-41885-4 ISBN 978-3-030-41886-1 (eBook)
https://doi.org/10.1007/978-3-030-41886-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Graphs model several real-world phenomena. Graph algorithms have been an interesting area of study and research due to rich structural information embedded into graphs. However, it is only in the last couple of decades that systemic aspects of graph algorithms have been pursued more prominently. Especially with the advent of large networks, and storage of huge amount of unstructured data, efficient graph processing has become a crucial piece in the performance puzzle of the underlying applications. For example, efficient graph processing is central to detection of online frauds, improvements in advertising and business, study of how diseases spread, computational geometry, and natural language processing, to name a few.

In parallel, we witnessed prominent innovations in computer architecture. Shared memory systems with a large number of cores, NUMA architecture, accelerators such as GPUs and FPGAs, and large decentralized systems are so common that high-performance computing is no longer a niche area. Almost every field, be it physics, biology, or material science, need to depend on such architectures for efficient simulation of a natural phenomenon. On the industry front, data warehouses are becoming bulky with pressures to store, maintain, and analyze tremendous amount of structured and unstructured data. Graph databases, textgraphs, and their modeling as hypergraphs pose new challenges. Processing such multi-modal data requires different algorithms. In addition, an algorithm needs to be tuned for a certain backend hardware for efficient implementation, because an optimized implementation for one hardware may not be efficient on the other.

Our goal in this book is to marry these two important trends: graph algorithms and high performance computing. Efficient and scalable execution of graph processing applications demands innovations at multiple levels: algorithm, its associated data structures, their implementation, and their implementation tuned to a certain hardware. Managing complexity at so many levels of abstraction is clearly a challenge for users. The issue gets exacerbated as domain experts may not be HPC experts, and vice versa. Therefore, the role of programming languages and the associated compilers is crucial to automate efficient code generation for various architectures. This book pens down essentials on these aspects.

We divide this book in three parts: programming, languages, and their compilation. The first part deals with manual parallelization of graph algorithms. It uncovers various parallelization patterns we encounter while dealing especially with graphs. The second part exploits these patterns to provide language constructs using which a graph algorithm can be specified. A programmer can work only with those language constructs, without worrying about their implementation. The implementation, which is the heart of the third part, is handled by a compiler, which can specialize code generation for a backend device. We present a few suggestive results on different platforms, which justify the theory and practice covered in the book. Together, the three parts provide the essential ingredients in creating a high-performing graph application.

While this book is pitched at a graduate or advanced undergraduate level as a specialized elective in universities, to make the matter accessible, we have also included a brief background on elementary graph algorithms, parallel computing, and GPUs. It is possible to read most of the chapters independently, if the readers are familiar with the basics of parallel programming and graph algorithms. To highlight recent advances, we also discuss dynamic graph algorithms, which pose new challenges at the algorithmic and language levels, as well as for code generation. To make the discussion more concrete, we use a case study using Falcon, a domain-specific language for graph algorithms. The book ends with a section on future directions which contains several pointers to topics that seem promising for future research.

We believe this book provides you wings to explore the exciting area of distributed graph analytics with zeal, and that it would help practitioners scale their graph algorithms to newer heights.

We would like to especially thank Ralf Gerstner, Executive Editor of Computer Science at Springer-Verlag for his encouragement and patient advice.

Palakkad, India — Unnikrishnan Cheramangalath
Chennai, India — Rupesh Nasre
Bangalore, India — Y. N. Srikant
January 2020

Contents