Peer-to-Peer Computing

Quang Hieu Vu • Mihai Lupu • Beng Chin Ooi

Peer-to-Peer Computing

Principles and Applications

Springer

Quang Hieu Vu
Institute for Infocomm Research (I²R)
1 Fusionopolis Way
#21-01 Connexis, South Tower
Singapore 138632
Singapore
qhvu@i2r.a-star.edu.sg

Mihai Lupu
Information Retrieval Facility (IRF)
Operngasse 20b
1040 Wien
Austria
mihailupu@gmail.com

Beng Chin Ooi
Department Computer Science
School of Computing
National University of Singapore
Computing Drive
Computing 1
Singapore 117590
Singapore
ooibc@comp.nus.edu.sg

ISBN 978-3-642-03513-5 e-ISBN 978-3-642-03514-2
DOI 10.1007/978-3-642-03514-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2009938717

ACM Computing Classification (1998) C.2, D.4, H.3, K.6

Cover design: KuenkelLopka, GmbH

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Peer-to-peer (P2P) technology, or peer computing, is an emerging paradigm that is now viewed as a potential technology to redesign distributed architectures (e.g., the Internet) and, consequently, distributed processing. In a classical P2P network, all participating computers (or nodes) have equivalent capabilities and responsibilities. The nodes can directly exchange resources and services between each other without the need for centralized servers. They can collaborate to perform tasks by aggregating the pool of resources (e.g., storage, CPU cycles) available in the P2P network. The distributed nature of such a design provides exciting opportunities for new killer applications to be developed.

P2P computing distinguishes itself from traditional distributed computing in three main aspects. First, the scalability of P2P systems goes far beyond that of traditional distributed systems. In particular, since P2P systems are able to scale to thousands of nodes, they can harness the power of computers over the Internet. Second, P2P, in its most uncompromising definition, requires everything to be completely decentralized. Ideally, no centralized structures should exist in P2P systems. Finally, and also the most important one, P2P applications often work in highly dynamic environments. Specifically, in terms of network topology, since P2P nodes can join and leave the system anytime, P2P systems do not have a fixed topology. Instead, their topology changes according to nodes in the system. Furthermore, the system's content and load are distributed in real time according to the actual demand and resource capability of nodes. For example, if a sharing file becomes "hot", i.e., it is repeatedly requested by several users, the file can be duplicated and deployed in many parts of the system.

The scale and dynamism that characterize P2P systems require traditional distributed technologies to be reexamined. A paradigm shift that includes self-reorganization, adaptation and resilience is called for. In recent years, there has been a proliferation of research efforts to design P2P systems and applications. This book attempts to present the *technical* challenges offered by P2P systems, and the efforts that have been proposed to address them. The purpose of this book is to provide a thorough and comprehensive review of recent advances on routing and discovery methods; load balancing and replication techniques; security, accountability and

anonymity, as well as trust and reputation schemes; programming models and P2P systems and projects. Besides surveying existing methods and systems, the book also compares and evaluates some of the more promising schemes.

The need for such a book is evident. It provides a single source for practitioners, researchers and newcomers on the state-of-the-art in the field. For practitioners, this book explains best practice, guiding selection of appropriate techniques for each application. For researchers, this book provides a foundation for development of new and more effective methods. For newcomers, this book is an overview of the wide range of advanced techniques for realizing effective P2P systems. This book can also be used as a text for an advanced course on Peer-to-Peer Computing and Technologies, or as a companion text for a variety of courses, including courses on distributed systems, grid, and cluster computing.

Organization of the Book

This book consists of ten chapters. Besides the first chapter that sets up the context and the last chapter that concludes with directions on the future of P2P, each of the other eight chapters is essentially self-contained and focuses on one aspect of P2P computing. These eight chapters can thus be read and used on their own, independently of the others.

- In Chap. 1, we provide background on P2P computing in general. We discuss the characteristics of P2P systems that distinguish them from other distributed systems. This chapter also looks at the benefits and promises of P2P, and some of the applications that will benefit from P2P computing. It examines the issues in designing P2P systems and sets the stage for subsequent chapters.
- Chapter 2 presents the various architectures of P2P systems. At one extreme, we have P2P systems that are supported by centralized servers. At the other extreme, pure P2P systems are completely decentralized. Between these two extremes are hybrid systems where nodes are organized into two layers: the upper tier "super" nodes act as servers for lower tier nodes. We compare these different architectures. In parallel to the static architectural considerations, we also look at how peers are defined—statically or dynamically. Support for dynamic reorganization of peers allows communities to be formed based on some common interests among nodes. For hybrid systems, we examine how nodes that are more powerful can be exploited to shoulder more responsibilities. Issues on incentives and fairness are also addressed.
- In Chap. 3, we focus on the issue of searching. There are several modes in which searching can be performed. First, a query node can broadcast queries to all nodes. Second, the query can be directed to nodes that are more likely to contain useful information first. This requires nodes to organize their peers based on some optimization criterion. Third, hashing techniques can be applied. We also look at how load-balancing can be realized in the hash-based category. Each of these techniques call for different metadata to be maintained.

- Chapter 4 presents techniques to perform complex queries. Besides simple keyword search, there is an increasing need to support more semantic-based queries for database and multimedia applications. These include partial match queries, range and join queries, and queries involving high-dimension vectors. We also look at how distributed queries are optimized and processed in the P2P context.
- Replication and caching are very effective mechanisms that can bring the data/results closer to the users to improve performance. However, in the P2P environment, it becomes much harder to control the optimal degree of replication, as well as to maintain the consistency between replicas. Chapter 5 presents the issues that need to be addressed and examines some of the existing solutions. In particular, we look at techniques that manage replicas/cache dynamically.
- Before P2P can be widely accepted by users, there are several other issues that need to be addressed: trust, privacy, anonymity, accountability, reliability, and security. These issues are discussed in Chaps. 6 and 7. In Chap. 6, we focus on security, privacy, and anonymity issues. We begin by discussing techniques designed to secure data as well as the overall P2P environment from different types of attacks. Then we present methods that prevent users from taking advantage of the system by freeloading off the resources contributed by a few. Finally, we look at techniques that are designed to support anonymity and privacy, to protect both the users that disseminate the data, as well as nodes that store the data. Techniques that authenticate third-party data publication are also examined in this chapter.
- Chapter 7 focuses on accountability, trust, and reputation. Here, we look at techniques that automate the collection and processing of information from previous queries to help users assess whether they can trust a server with a new query.
- In Chap. 8, we look at programming tools that are suitable for P2P environments. After having presented in the previous chapters the theoretical aspects of P2P systems, in this chapter we will identify tools to develop P2P systems, ranging from low level network programming tools, like sockets, to specific programming languages designed to be used for P2P applications.
- Chapter 9 describes some representative P2P systems and applications that have been deployed. We look at how different application environments and requirements drive the design and architecture of the systems. We discuss popular techniques employed in each type of applications. In particular, we present systems that support file sharing, data backup, structured data management, and data caching. Additionally, we also introduce mobile systems employing P2P technologies.

Finally, in Chap. 10, we make a conclusion of the book and suggest promising research topics that deserve further attention. Additionally, we also discuss a potential use of P2P in industry by analyzing a case of supply chain management system.

Acknowledgements

This book cannot be finished without the help of several people. The authors of this book would like to thank Dr. Kian-Lee Tan, Dr. Panos Kalnis, and Dr. Bei Yu at National University of Singapore, Dr. Aoying Zhou at East China Normal University, and Dr. Linhao Xu at the IBM China Research Laboratory for their contributions in the initial phase of the book.

Contents