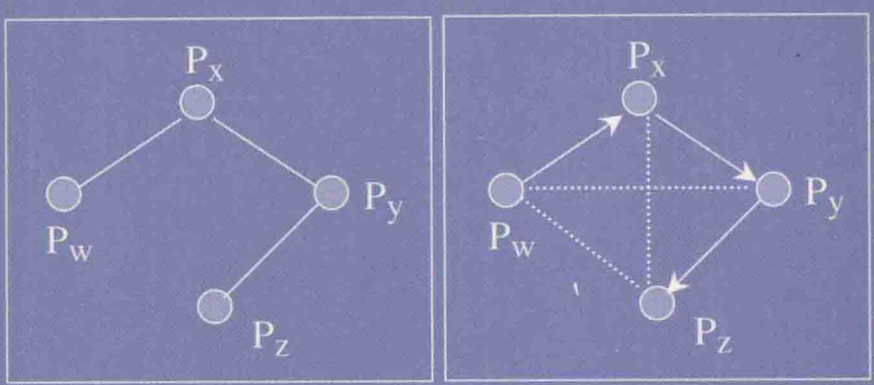
Hot Topics

LNAI 2872

Gianluca Moro
Claudio Sartori
Munindar P. Singh (Eds.)

Agents and Peer-to-Peer Computing

Second International Workshop, AP2PC 2003
Melbourne, Australia, July 2003
Revised and Invited Papers



Springer

Gianluca Moro
Claudio Sartori
Munindar P. Singh (Eds.)

Agents and Peer-to-Peer Computing

Second International Workshop, AP2PC 2003
Melbourne, Australia, July 14, 2003
Revised and Invited Papers

江苏工业学院图书馆
藏书章

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Gianluca Moro
Università di Bologna
Dipartimento di Informatica, Elettronica e Sistemistica (DEIS)
Via Venezia 52, 47023 Cesena, Italy
E-mail: gmoro@deis.unibo.it

Claudio Sartori
Università di Bologna
CSITE - CNR
Viale Risorgimento, 2, 40136 Bologna, Italy
E-mail: claudio.sartori@unibo.it

Munindar P. Singh
North Carolina State University
Department of Computer Science
Raleigh, NC 27695-7535, USA
E-mail: singh@ncsu.edu

Library of Congress Control Number: 2004116524

CR Subject Classification (1998): I.2.11, I.2, C.2.4, C.2

ISSN 0302-9743
ISBN 3-540-24053-5 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11365105 06/3142 5 4 3 2 1 0

Preface

Peer-to-peer (P2P) computing is attracting enormous media attention. Typical applications are file sharing, as in Gnutella, and exploiting distributed computing power, as in the SETI (Search for Extra Terrestrial Intelligence) project.

The most popular applications at present are limited in their scope, but they are highlighting some of the key challenges of P2P computing and exposing the limitations of traditional approaches to addressing such challenges. First, the peers are autonomous entities: they can cooperatively participate or not according to their own choice. Second, the peers are heterogeneous, meaning that in general we would not be justified in making strong assumptions about how they are designed or how their information structures are conceptually modeled. The applications of P2P computing go beyond file sharing or load balancing of computing resources. Understood more generally, P2P computing is a natural approach to the development of large systems from autonomous, heterogeneous components. The obvious idea would be for entities to function as peers that provide services or expose resources for sharing. Services or resources can then be composed dynamically to yield novel functionalities. Rigorous composition techniques are a major research direction.

First, let's consider heterogeneity. One aspect of the above-mentioned techniques for developing P2P systems is dealing with the information structures of the various peers. Another aspect is dealing with the underlying processes. How do we ensure that peers are able to share knowledge and able to act in unison? Addressing both aspects involves modeling the peers appropriately and reconciling their conceptual differences.

Next, let's consider autonomy. Since the participants are autonomous and not governed by any central agency, certain new challenges must be addressed. One, we need mechanisms for trust and reputation, and, related to these, for governance and regulation. Two, we need to develop economic environments or incentive mechanisms that foster knowledge sharing and collaboration, i.e., lead the peers to prefer cooperative over non-cooperative behaviors in sharing resources. Systems such as Gnutella already suffer from the problem of *free riding*, where some participants take advantage of the system but never contribute to it. What business models would properly support those who contribute or give an incentive to the peers to cooperate? What techniques would sustain such business models?

Interestingly and significantly, research on multiagent systems and on large-scale information systems has at least partially addressed many of the challenges of P2P systems. The work on information systems has studied the consequences of heterogeneity of knowledge and process. The work on multiagent systems has studied the consequences of autonomy. In particular, the basic doctrine of multiagent systems—that the member agents are autonomous—agrees with what P2P systems require. Research on topics such as task decomposition, protocols, economic models involving game theory and decision theory, and coordination and teamwork all feed naturally into P2P systems.

For the above reasons, this workshop series aims at addressing the following nonexhaustive list of topics:

- Intelligent agent techniques for P2P computing
- P2P computing techniques for multi-agent systems
- The Semantic Web and semantic coordination mechanisms for P2P systems
- Scalability, coordination, robustness and adaptability in P2P systems
- Self-organization and emergent behavior in P2P systems
- E-commerce and P2P computing
- Participation and contract incentive mechanisms in P2P systems
- Computational models of trust and reputation
- Community of interest building and regulation, and behavioral norms
- Intellectual property rights in P2P systems
- P2P architectures
- Scalable data structures for P2P systems
- Services in P2P systems, including service definition, discovery, filtering, composition, and so on
- Knowledge discovery and P2P data mining
- P2P-oriented information systems
- Information ecosystems and P2P systems
- Security considerations in P2P networks
- Ad hoc networks and pervasive computing based on P2P architectures and wireless communication devices.

The workshop series emphasizes discussions about methodologies, models, algorithms and technologies, strengthening the connection between agents and P2P computing. These objectives are accomplished by bringing together researchers and contributions from these two disciplines but also from more traditional areas such as distributed systems, networks, and databases.

This volume is the postproceedings of AP2PC 2003, the 2nd International Workshop on Agents and P2P Computing,[1] which took place in Melbourne on July 14, 2003 in the context of the 2nd International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2003).

This volume is organized according to the sessions held at the workshop. Besides the invited papers related to the invited talk and to the panel, these were framed into the following topics:

- Paradigm integration and challenges
- Trust
- Self-organization
- Incentives
- Search and systems
- Adaptive applications
- Mobile agents

[1] http://p2p.ingce.unibo.it/

此为试读，需要完整PDF请访问：www.ertongbook.com

This proceedings brings together papers presented at the workshop, fully revised to incorporate reviewers' comments and discussions at the workshop, plus three invited papers related to the panel. After the call for papers we received 22 submissions. All submissions were reviewed for scope and quality; finally, 11 were accepted as full papers and 6 as short papers. AP2PC 2003 drew over 40 attendees. Given the dual threats of SARS and war this year and logistical challenges of getting to Melbourne, it is not surprising that this was one of the better attended workshops at AAMAS.

We express our deepest appreciation to the participants for their lively discussions. We would like to acknowledge the contributions of the invited speaker, Prof. Beng Chin Ooi (National University of Singapore), the authors for their excellent submissions, and the program committee members for their diligence in reviewing submissions on a tight schedule. We would also like to thank the panel chair, Aris M. Ouksel, and the invited panelists, Sonia Bergamaschi (University of Modena and Reggio-Emilia), Rajkumar Buyya (University of Melbourne), and Onn Shehory (IBM Haifa). We would like to acknowledge the steering committee for its guidance and encouragement.

This workshop followed the successful first edition, which was held in conjunction with AAMAS in Bologna in 2002. In recognition of the interdisciplinary nature of P2P computing, a sister event called the International Workshop on Databases, Information Systems, and P2P Computing was held in Berlin in September 2003 in conjunction with the International Conference on Very Large Data Bases (VLDB).

Autumn 2003

Gianluca Moro
Claudio Sartori
Munindar P. Singh

Executive Committee

Organizers

Program Co-chairs

Gianluca Moro
Dept. of Electronics, Computer Science and Systems,
University of Bologna, Italy

Claudio Sartori
IEIIT-BO-CNR, University of Bologna, Italy

Munindar P. Singh
Dept. of Computer Science,
North Carolina State University, USA

Panel Chair

Aris M. Ouksel
Dept. of Information and Decision Sciences,
University of Illinois at Chicago, USA

Steering Committee

The steering committee consists of the above plus the following people:

Karl Aberer, EPFL, Lausanne, Switzerland

Sonia Bergamaschi, Dept. of Science Engineering,
University of Modena and Reggio-Emilia, Italy

Manolis Koubarakis, Dept. of Electronic and Computer Engineering,
Technical University of Crete

Paul Marrow, Intelligent Systems Laboratory,
BTexact Technologies, UK

Program Committee

Karl Aberer, EPFL, Lausanne, Switzerland
Sonia Bergamaschi, University of Modena and Reggio-Emilia, Italy
M. Brian Blake, Georgetown University, USA
Rajkumar Buyya, University of Melbourne, Australia
Ooi Beng Chin, National University of Singapore, Singapore

Paolo Ciancarini, University of Bologna, Italy
Costas Courcoubetis, Athens University of Economics and Business, Greece
Yogesh Deshpande, University of Western Sydney, Australia
Asuman Dogac, Middle East Technical University, Turkey
Boi V. Faltings, EPFL, Lausanne, Switzerland
Maria Gini, University of Minnesota, USA
Chihab Hanachi, University of Toulouse, France
Mark Klein, Massachusetts Institute of Technology, USA
Matthias Klusch, DFKI, Saarbruecken, Germany
Yannis Labrou, PowerMarket Inc., USA
Tan Kian Lee, National University of Singapore, Singapore
Dejan Milojicic, Hewlett-Packard Labs, USA
Alberto Montresor, University of Bologna, Italy
Luc Moreau, University of Southampton, UK
Jean-Henry Morin, University of Geneva, Switzerland
John Mylopoulos, University of Toronto, Canada
Andrea Omicini, University of Bologna, Italy
Maria Orlowska, University of Queensland, Australia
Aris M. Ouksel, University of Illinois at Chicago, USA
Mike Papazoglou, Tilburg University, Netherlands
Terry R. Payne, University of Southampton, UK
Paolo Petta, Austrian Research Institute for AI, Austria
Jeremy Pitt, Imperial College London, UK
Dimitris Plexousakis, Institute of Computer Science, FORTH, Greece
Martin Purvis, University of Otago, New Zealand
Omer F. Rana, Cardiff University, UK
Katia Sycara, Robotics Institute, Carnegie Mellon University, USA
Douglas S. Reeves, North Carolina State University, USA
Thomas Risse, Fraunhofer IPSI, Darmstadt, Germany
Pierangela Samarati, University of Milan, Italy
Christophe Silbertin-Blanc, University of Toulouse, France
Maarten van Steen, Vrije Universiteit, Netherlands
Markus Stumptner, University of South Australia, Australia
Peter Triantafillou, Technical University of Crete, Greece
Anand Tripathi, University of Minnesota, USA
Vijay K. Vaishnavi, Georgia State University, USA
Francisco Valverde-Albacete, Universidad Carlos III de Madrid, Spain
Maurizio Vincini, University of Modena and Reggio-Emilia, Italy
Fang Wang, BTexact Technologies, UK
Gerhard Weiss, Technische Universitaet Muenchen, Germany
Bin Yu, North Carolina State University, USA

Sponsors

This workshop would not have been possible without the generous support from our sponsors, UNI.TU.RIM S.p.A., Fondazione Cassa di Risparmio di Rimini, and Microsoft. To these we express our gratitude.

Table of Contents

Paradigm Integration and Challenges

Trust

Self-Organization

Incentives

Search and Systems

Adaptive Applications

Mobile Agents

此为试读,需要完整PDF请访问: www.ertongbook.com

Information Acquisition Through an Integrated Paradigm: Agent + Peer-to-Peer

Beng Chin Ooi[1], Wee Siong Ng[2], Kian-Lee Tan[1], and AoYing Zhou[3]

[1] Department of Computer Science,
National University of Singapore, Singapore
{ooibc, tankl}@comp.nus.edu.sg
[2] Singapore-MIT Alliance,
National University of Singapore, Singapore
smangws@nus.edu.sg
[3] Department of Computer Science and Engineering,
Fudan University, Shanghai, P.R. China
aoying@fudan.edu.cn

Abstract. Agent computing provides developers with a way to define problem-solving computation at an abstract level, whereas, the key strength of current P2P development centers on resource gathering and defining efficient resource discovery strategies. Integration of the two paradigms is required for the development of self-evolving, open and scalable systems. In this paper, we first investigate varieties of P2P facilities that could benefit agent development and discuss broadly different ways of integration of the two paradigms. Second, we present a prototype system, BestPeer, that exploits both agent and P2P computing. In P2P environments, the schema is typically not given in advance or it might be implicit in the data. Consequently, it is notably challenging to acquire, manage and analyze data in order to produce meaningful information for decision-making. We next present PeerDB that is built on top of BestPeer to facilitate data sharing without a global schema.

1 Introduction

Agent and Peer-to-Peer (P2P) are two paradigms that realize the real power of computing through autonomous, distributed and dynamic systems. These systems are becoming increasingly popular as they enable users to exchange digital information and share in problem-solving by participating in complex networks. In particular, many researchers consider the agent system as an autonomous problem-solving entity while P2P provides support for pooling resources together. Merging these two disciplines by adopting the best of each approach could potentially provide an ultimate solution that is inexpensive, easy to use, self-learning and modifying, highly scalable and needing no central administration.

In order to deal with the autonomous, scale and dynamism that characterize P2P and agent systems, a merged paradigm is required that includes

G. Moro, C. Sartori, and M.P. Singh (Eds.): AP2PC 2003, LNAI 2872, pp. 1–12, 2004.
© Springer-Verlag Berlin Heidelberg 2004

self-organization, adaptation and automated information matching, and support discovering as intrinsic properties. In this paper, we first define different approaches on merging infrastructures from these two disciplines. Second, we present BestPeer [1,11], a system that integrates both paradigms to support fast and easy P2P application development. Our solution incorporates a self-configurable mechanism whereby a node in the BestPeer network can dynamically reconfigure itself to have direct (logical) connections with peers that benefit it most.

Finally, we elaborate on an interesting issue based on the integrated paradigm: how can an agent perform information acquisition in the P2P system without relying on global knowledge? We present our experience in addressing this problem in the context of PeerDB, a full-fledged data management system that supports fine-grain content-based searching with the help of agent technologies. Our solution incorporates Information Retrieval (IR) techniques which enable peers to share data without shared schema. PeerDB employs a name-based matching technique that matches schema elements by relying on the user to supply additional information (metadata) in order to reduce mismatch. PeerDB primarily concerns itself with the online information exploration. Online information exploration is different from traditional data translation and schema integration strategies. In the former, results are transient and users are more tolerant of mismatch candidates. Schema integration, on the other hand, needs to ensure certain degree of consistency and accuracy, which in turn, requires more complicated approaches. PeerDB provides a simple and yet effective approach for information acquisition in environments with heterogeneous data sources.

2 The Infrastructures

In this section, we shall discuss the strategies for merging infrastructures from P2P and agent computing.

2.1 Facilities Provided by P2P

The P2P community has contributed much to the development of efficient resource discovery and routing strategies. Clearly, an efficient resource discovery strategy together with query routing strategy forms fundamental problems of resource sharing. Earlier efforts such as Napster adopt a centralized model of resource sharing. Here, the central server maintains a master list of all the metadata of peers in the network. This metadata is being used for describing data housed in peers and it might include file name, IP address, line speed and so on. However, the data is located in the peers. In this case, the servers are simply playing the role of answering queries and indexing the meta-information submitted by connecting peers. Perhaps this centralized architecture is most similar to existing development of multi-agent systems [7,6,22,14]. Agents are required to contact a centralized resource manager for locating the services. However, such an approach has several limitations. First, there is a single point of failure. In additional, maintaining a unified view is computationally expensive and scaling up can be a serious problem. More recently, several routing mechanisms in pure decentralized environment have been proposed. For example the Breadth-First-Traversal [4] (BFT) and distributed hash table (e.g., Chord [21] and CAN [16]). These facilities may potentially to be reused in agent development for developing a truly autonomous and decentralized system.

2.2 Merging of Infrastructures: P2P and Agent

There are three broad approaches for merging the two technologies. One is based on integrating P2P technology to underlying agent systems (the left image of Figure 1). For instance, a DHT-based [16,21,2] routing strategy could be integrated into an existing agent system for efficient agent routing. This approach is agent-oriented since it defines P2P as a subset of tools to facilitate efficient routing of agents. The second approach is a P2P-oriented merging strategy, where the main idea is to build a proprietary software agent on top of an existing P2P system (the right image of Figure 1). The third approach operates on three tiers, with a middleware in between the agent and P2P layers (the centre image of Figure 1).

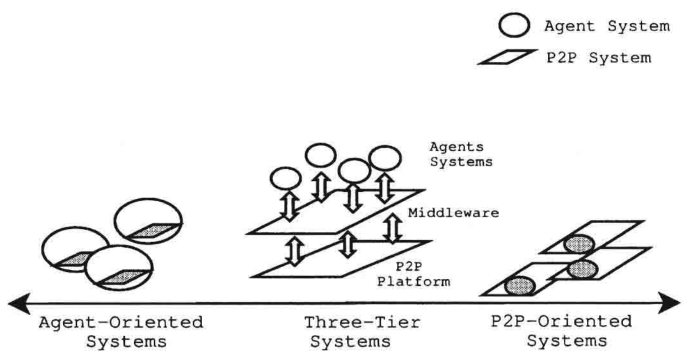


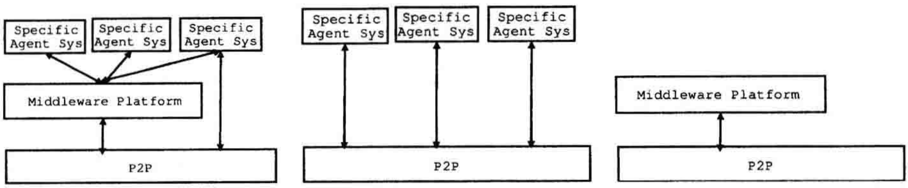
Fig. 1. Infrastructure of P2P and Agents

Most of the existing agent systems provide support for agent collaboration and communication but are not native to P2P technology. The development of P2P applications based on these platforms would require a longer and more costly effort. There are several reasons that suggest the limitation of applying a traditional agent system in a P2P model. First, traditionally, mobile search agents perform search operations by moving themselves to the site containing the target information and executing a given task. The agent's path is either predefined or the agent has knowledge of where to find the services. For example, in order to find the cheapest airfare, a travel agent is given a set of sites that provide airfare query services. The agent's programmers have to know where the agent needs to go and where the next destination is after the task at a site is completed. However, this may require a pre-defined knowledge of the environment – which is not always feasible, e.g., there may not exist any pre-defined knowledge of who is offering a particular service and where. The problem may be solved by integrating P2P query routing strategies into agent systems to form agent-oriented systems. Obviously, the main drawback concerns the extensibility of the system to upgrade the services, e.g., incorporating new routing strategies or new P2P services into the system will cause a major disruption of the system. Moreover, the whole architecture may possibly become "fatter", which may in return result in unpredictable behavior. Also, there may exist several agent systems with P2P supports but which are unable to communicate with each other. This may be due to the fact that they employ either different agent communication languages or different P2P protocols. In apparent recognition of this problem, the agent community has started to standardize agent communication languages such as KQML [3] and FIPA ACL [20]; meanwhile, P2P is still evolving. The details of the agent computing roadmap can be found in [17].

P2P-oriented system have inherited similar issues that are faced by the agent-oriented approach. This paradigm may be useful in a specific corporate environment where the predefined protocol and languages have been set up as in the agent-oriented approach. The two approaches that have just been discussed tend to be closed systems rather than sustainable ones that could adapt to future publicly-advertised standards.

The alternative solution – which is the third approach to the merger of agent and P2P technologies – operates at the following three tiers: 1) an agent system running on the peer to provide application-related services, 2) a P2P platform to handle communication and the necessary message routing strategy, and 3) a middle tier that handles the communication between the agent and P2P layers. Each tier focuses exclusively on its assigned tasks. For example, when a new P2P routing strategy is invented, only the P2P layer needs to be updated. Similarly, to accommodate large numbers of participants, only the middle tier needs to be scaled by employing industry agreed protocols and languages. Such an approach would help to develop a fully open and truly scalable distributed data sharing system that supports dynamic networking and heterogeneity in the data environment.

Figure 2(a) depicts the three-tier system. Middleware platform provides general-purpose agent's behavioral functions, such as sending and receiving messages, repository for data storing and retrieving. It also offers negotiation and coordination management among peers. These functions are commonly needed for any kind of agent systems and regardless of the applications domains. Domain specific behaviors, on the other hand, are provided by the specific agent systems. In general, middle layer is a generic agent platform which provides common skeletons and basic agents functionalities. The purpose is to allow agents from different systems to cooperate. Agents from different systems can be transformed to a common agent that operates in middleware layer and vice versa.



(a) Three-tier System. (b) Without Middleware. (c) With Middleware.

Fig. 2. Variant of Three-tier System.

Since the middleware platform itself is an agent platform (with limited agent capabilities), there are two possible variances of the three-tier system: without middleware platform (Figure 2(b)) and with middleware platform (but without any specific agent systems) (Figure 2(c)). A three-tier system without the middleware platform is more functionalities-rich but it is a platform dependency approach, since each of the agents may be created based on APIs provided by different platform vendors. The different interaction protocol of each vendor makes coordination among peers from different agent systems difficult. In contrast, a system with only middleware platform has limited functionalities, but facilitates easy interaction.

3 The BestPeer Approach

As mentioned earlier, agent systems designers could have benefited from connections with P2P disciplines. A good evaluation of work on combining P2P and agent paradigms can be found in [10]. In this section, we shall discuss a working prototype of integrated agent-P2P system developed for serving as a platform on which P2P applications can be developed easily and efficiently based on agent technologies.

The BestPeer [1,11] project was initiated in the year 2000 at the National University of Singapore to study how P2P technologies can be employed for distributed applications, such as collaborative caching, information retrieval, distributed data management, etc. It is a three-tier architecture with an agent layer at the top of the hierarchy, middleware layer that resides in between the underlying P2P layer on the one hand and the agent layer on the other. The P2P layer is the lowest layer of the hierarchy for supporting low-level communication, resource sharing capabilities amongst nodes and self-network reconfiguration.

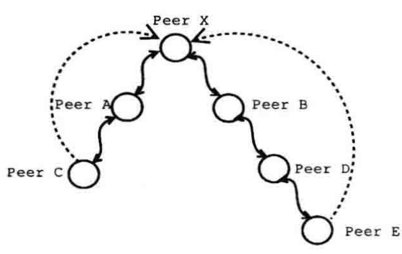
In BestPeer, the P2P technology provides resource sharing capabilities amongst nodes, while mobile agents technology further extends the functionalities. In particular, since agents can carry both code and data, they can effectively perform any kind of functions. With mobile agents, BestPeer not only provides files and raw data, but processed and meaningful information. For example, in

BestPeer, an agent can be sent to a peer with the data file to "digest" its content and to generate reports for the requester. In another word, in contrast to existing P2P systems, i.e., Gnutella, Napster, that provide only file level sharing (i.e., sharing of the entirety of a file), BestPeer supports for content-based search with the help of agent technologies.

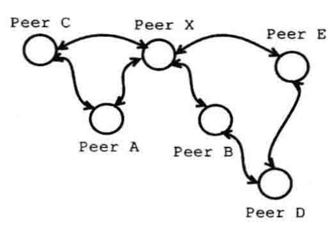
In BestPeer, we have implemented our own Java-based agent system instead of using existing systems (e.g., [9]). Like existing systems, both the agent and its class have to be present for the agent to resume execution at the destination engine. Thus, if the class is not already at the destination node, the class has to be transmitted also. For the moment, we have adopted a purely "code-shipping" strategy where a node will always perform its operation at the destination node (where the data reside). This is a reasonable approach as it exploits parallelism by enabling all peers to operate on their data simultaneously; otherwise, the node will become a bottleneck.

More importantly, the use of agents allows BestPeer nodes to collect information (e.g., what files/content are sharable, statistics, etc.) on the BestPeer network, and this can be done offline. This allows a node to be better equipped to determine who should be its directly connected peers or who can provide it better service.

BestPeer is self-configurable (P2P layer), i.e., a node can dynamically optimize the set of peers that it can communicate directly with based on some optimization criterion. By keeping peers that provide most information or services in close proximity (i.e, direct communication), the network bandwidth can be better utilized and system performance can be optimized.



(a) Before reconfiguration.



(b) After reconfiguration.

Fig. 3. Example on BestPeer's Reconfigurable Feature.

Figure 3 illustrates an example of BestPeer's reconfigurable feature. In Figure 3(a), Peer X is the base node that initiates a request. Here, Peer X initially has two directly connected peers - Peers A and B. However, only Peer C and Peer E contain objects that match Peer X's current query. Peer X can then obtain the results from Peer E and Peer C directly. At the same time, Peer X determines that Peer C and Peer E are not its direct peers and they benefit it

此为试读，需要完整PDF请访问： www.ertongbook.com