Guest Editor's Introduction

P2P computing has become an extremely popular topic in computer science. It affects diverse areas such as networking, distributed systems, information systems, algorithms and databases. The P2P paradigm introduces an architectural principle "replacing" the paradigm of client-server computing. It is based on the concepts of decentralization and resource sharing. By avoiding central bottlenecks and distributing workload it facilitates the deployment of applications at a global scale.

The use of the P2P paradigm as a practical approach for developing scalable applications has not emerged from academic research, but resulted from a number of grass-root efforts on the Web. The most popoular ones were systems for the purpose of (music) file sharing, such as Napster, Gnutella and Kaaza. The enormous success of these systems subsequently had a major impact on the various research communities.

The database community is particularly concerned by this development since key applications of P2P computing concern information sharing. The impact of P2P computing on the various research communities is reflected by in a rapidly increasing number of publications (for example, at the time of this writing we have roughly 7000 documents and 4000 citations in CiteSeer referring to P2P or peer-to-peer most of them dating from the period 2000-2003) and a number of new events specifically dedicated to various aspects of P2P computing (e.g., O'Reilly's P2P conference, IPTPS, AP2PC, DBISP2P, IEEE P2P conference).

In this special issue I wanted to provide a snapshot of the current state of research in P2P data management. I invited some of the currently leading groups in the field to report on their research and to highlight some of their technical results. In particular, this special issue is intended to show that the P2P paradigm applies to various functional dimensions of data management.

The probably most prominent problem in P2P computing is resource location. Essentially it concerns the issue of locating a resource given its identifier. Two important directions exist: unstructured P2P networks using flooding techniques and structured P2P networks using distributed hash table (DHT) techniques for directed routing of queries. Bawa et al. (*Peer-to-peer Research at Stanford*) summarize in their article several techniques they developed to improve existing solutions for resource location, both in unstructured and structured P2P networks.

In their paper they address also another important aspect of P2P computing, i.e., how to cooperate with potentially malicious peers while sharing resources with them. They present a method for deriving trust values from reputation information within a P2P architecture. Based on these trust values peers may decide whether to cooperate.

Aberer et al. (*P-Grid: A Self-Organizing Structured P2P System*) emphasize in their article the importance of the principle of self-organization in decentralized P2P systems. They demonstrate how it can be applied to solve resource allocation problems and to implement basic services such identity and trust management.

Whereas the problems of resource location and peer cooperation are central, they are of general nature reaching beyond the specific scope of databases into communication systems and agent technology. Hellerstein (*Toward Network Data Independence*) indicates what the specific contributions of the database community to P2P computing are. In particular, he considers the principle of data independence that is a pillar of database technology but much less applied in other disciplines as a central idea for the P2P arena. From that he develops multiple specific questions for which the database community can provide solutions, specifically for addressing the problem of resource location.

One of the major contributions the database community is making at the moment is clearly introducing data schemas into P2P systems. This is a problem at the heart of data management.

Nejdl et al. (*Design Issues and Challenges for RDF- and Schema-Based Peer-to-Peer Systems*) introduce their Edutella architecture, which uses RDF-based schemas to enable

schema-based querying in P2P information sharing systems. They support distributed querying of RDF databases and provide techniques for efficient query processing based on a super-peer topology and using various indexing and clustering techniques.

Tatrinov et al. (*The Piazza Peer Data Management Project*) take the use of schemas one step further. They introduce the concept of schema mappings in order to allow querying across heterogeneous (relational) schemas. They adapt existing schema integration techniques (GAV, LAV) for the P2P context. Thus they replace the notion of an integrated schema by the notion of a semantic network of schema mappings in which each peer has an individual view on the rest of the network.

Arenas et al. (*The Hyperion Project: From Data Integration to Data Coordination*) extend the concept of schema mappings: as part of their architecture they support the specification of instance mappings and the use of triggers to maintain data consistency across peers during updates.

Having schema-based mappings allows then to query among semantically heterogeneous domains. However, the important problem of how to establish these mappings remains to be solved.

Ooi et al. (*Relational Data Sharing in Peer-based Data Management Systems*) propose to generate such mappings on the fly applying IR based techniques to schemas within their PeerDB architecture. Peers propogate their schemas to other peers and by applying keyword matching to schema constituents they can route their queries to other peers having potentially useful information.

Ouksel (*In-Context Peer-to-Peer Information Filtering on the Web*) proposes the use of inferencing techniques to establish mappings semi-automatically. In his SCOPES approach schema mappings are established by peers through negotiation. In this way the peers create mutually accepted beliefs on how to resolve schema conflicts. As a result of this process the peers establish communities of interest that can share information.

Originally the most popular P2P systems were intended to support ad-hoc querying, for example, for the purpose of information and content sharing. The last two papers show that more general services can be envisaged.

Koubarakis et al. (*Selective Information Dissemination in P2P Networks: Problems and Solutions*) propose the use of a P2P architecure for the active dissemination of data, a problem that has been studied for client-server settings already intensively, e.g. for information filtering or continuous querying. Publications and subscriptions for information resources are managed by a super-peer architecture. Since information sharing applications require both structural and text-based querying capabilities they introduce a theoretically well-founded data model integrating those two aspects.

Pitoura et al. (*DBGlobe: A Service-Oriented P2P System for Global Computing*) consider the support of general services in a P2P architecture. They describe an approach to managing metadata for service descriptions and to use this metadata in order to support the matching of service providers and service requestors. Specifically they propose the use of Bloom filters in order to support the efficient processing of service requests.

In summary, the contributions to this special issue nicely illustrate that P2P computing has initiated a rich set of activities in database research, covering almost every aspect of data management, but particularly focussing on the use of data schemas for P2P data management at the moment. For the future we may expect a number of developments building on the foundations that are currently laid out: more work on query optimization, the use of distributed decision mechanisms for resource allocation, approaches to the update problem, a novel approach to semantic interoperability, based on consensus building among peers, and the application of the P2P architecture to other data management problems such as data mining and Web retrieval.

Karl Aberer, EPFL, July 2003