P-Grid: A Self-organizing Structured P2P System

Karl Aberer, Philippe Cudré-Mauroux, Anwitaman Datta, Zoran Despotovic,
Manfred Hauswirth, Magdalena Punceva, Roman Schmidt

Distributed Information Systems Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
Contact: karl.aberer@epfl.ch

1 Self-organizing Structured P2P Systems[1]

In the P2P community a fundamental distinction is made among unstructured and structured P2P systems for resource location. In unstructured P2P systems in principle peers are unaware of the resources that neighboring peers in the overlay networks maintain. Typically they resolve search requests by flooding techniques. Gnutella [9] is the most prominent example of this class. In contrast, in structured P2P systems peers maintain information about what resources neighboring peers offer. Thus queries can be directed and in consequence substantially fewer messages are needed. This comes at the cost of increased maintenance efforts during changes in the overlay network as a result of peers joining or leaving. The most prominent class of approaches to structured P2P systems are distributed hash tables (DHT), for example Chord [17].

Unstructured P2P systems have generated substantial interest because of emergent global-scale phenomena. For example, the Gnutella overlay network exhibits the following characteristics [15]:

1. The network has a small diameter, which ensures that a message flooding approach for search works with a relatively low time-to-life (approximately 7).
2. The node degrees of the overlay network follow a power-law distribution. Thus few peers have a large number of incoming links whereas most peers have a very low number of such links.

These properties result from the way Gnutella performs network maintenance: each peer maintains a fixed number of active links. Using the network maintenance protocol a peer discovers new peers in the network by flooding discovery messages. From the responses it (randomly) selects certain peers to which direct network links are established.

The resulting power-law distribution of node degrees has been discovered for many other types of networks as well, for example, the World Wide Web, citation networks, and genetic networks. The property is accounted to the mechanism of how these networks are being constructed: New nodes preferentially attach to already well-connected ones exactly what is observed for Gnutella. Thus Gnutella is a completely decentralized but also self-organizing system: From randomized interactions of peers global structures emerge.

Despite the similarity of the network maintenance and search protocols in Gnutella, they serve fundamentally different purposes and are independent. The network maintenance protocol implements a self-organization process changing the system state, i.e., the overlay network's structure, whereas the search protocol implements a distributed algorithm in the overlay network. The properties of the emergent Gnutella overlay network are relevant for the search performance. The independence of the network maintenance and search protocols makes it possible to use alternative search protocols which may exploit the emergent overlay network structure more efficiently. Examples are the random walker model [14] and the percolation search model [16], which both exploit the specific overlay network structure.

In contrast, standard structured P2P systems follow a different approach with respect to network maintenance. They assign static identifiers to peers and the distributed data structures (e.g., DHTs) are constructed based on these identifiers by distributed algorithms. As a result the overlay network structure is mainly determined by the choice of identifiers and in turn any self-organization of the system is prevented.

However, there exists an example of a structured P2P system, Freenet [8] that exploits a self-organization process for optimizing resource allocation. Freenet maintains routing tables just the way as a structured P2P system does, but the overlay network is modified as a result of query

[1] The work presented in this paper was supported in part by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322 and by SNSF grant 2100-064994, "Peer-to-Peer Information Systems."

execution, such that resources with similar keys tend to cluster and in turn queries can be answered more efficiently. Thus Freenet attempts to implement a self-organization process, similarly as Gnutella, with the purpose of optimizing the system's performance. The Freenet data structures are constructed in a heuristic manner, so no probabilistic execution guarantees on search efficiency can be given. Experimental results are inconclusive whether the same degree of efficiency as in DHT-based systems is achieved in general [5].

This motivated us to ask the following question: Is it possible to use a self-organization process (such as in Gnutella or Freenet) to construct an overlay network that is a DHT-like routing infrastructure such that both probabilistic guarantees on search efficiency can be given and resource allocation is optimized? In particular, with respect to resource allocation we are interested in the problem of load balancing in the presence of non-uniform data key distributions.

Load balancing as a resource allocation problem is critical to support high scalability, availability, accessibility, and throughput. Poor load balancing may in fact gradually transform a P2P system into a backbone-based system as it was observed for Gnutella [7]. For systems supporting equality-based lookup of data only, the problem of non-uniform workloads may be circumvented by applying hash functions to the data keys, thus uniformly distributing workload, both for storage and query answering. In combination with using balanced search structures, i.e., balanced distributed search trees, this approach leads to uniform load distribution among the participating peers. However, it is limited if further semantics of the data keys is exploited, for example, in the simplest case when the ordering of data keys is used to support prefix or range queries. This is critical for DB-oriented applications.

2 P-Grid in a Nutshell

As a result of our research we can provide a solution to the problem we have posed above. P-Grid [3] is a peer-to-peer lookup system based on a virtual distributed search tree, similarly structured as standard distributed hash tables: Figure 1 shows a simple P-Grid.

Each peer holds part of the overall tree. Every participating peer's position is determined by its path, that is, the binary bit string representing the subset of the tree's overall information that the peer is responsible for. For example, the path of Peer 4 in Figure 1 is 10, so it stores all data items whose keys begin with 10. For fault-tolerance multiple peers can be responsible for the same path, for example, Peer 1 and Peer 6. P-Grid's query routing approach is as follows: For each bit in its path, a peer stores a reference to at least one other peer that is responsible for the other side of the binary tree at that level. Thus, if a peer receives a binary query string it cannot satisfy, it must forward the query to a peer that is "closer" to the result. In Figure 1, Peer 1 forwards queries starting with 1 to Peer 3, which is in Peer 1's routing table and whose path starts with 1. Peer 3 can either satisfy the query or forward it to another peer, depending on the next bits of the query. If Peer 1 gets a query starting with 0, and the next bit of the query is also 0, it is responsible for the query. If the next bit is 1, however, Peer 1 will check its routing table and forward the query to Peer 2, whose path starts with 01.

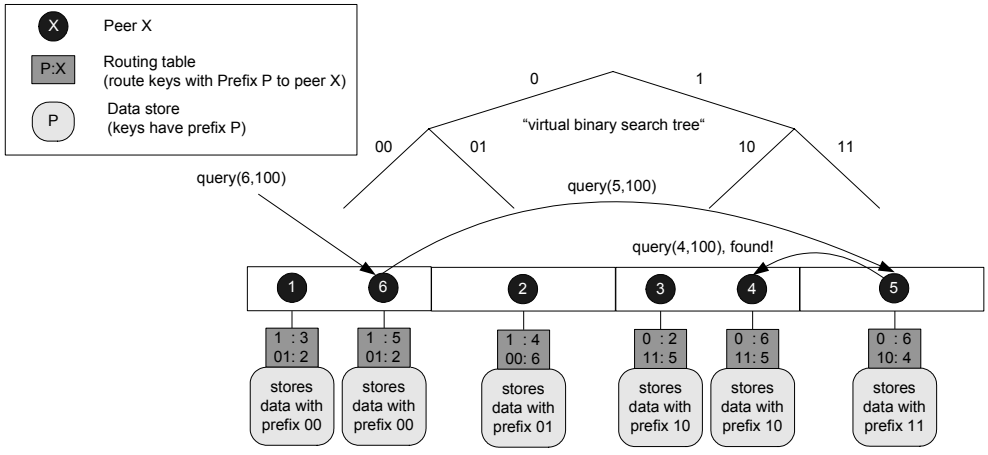

Figure 1: Example P-Grid

The salient feature of P-Grid, in contrast to other DHT-based P2P systems, is the separation of concern between peer identifier and peer's path. In P-Grid peer paths are not determined a priori but are acquired and changed dynamically through negotiation with other peers as part of the network maintenance protocol. Thus P-Grid's prefix-routing infrastructure is constructed by means of a decentralized, self-organizing process in which it adapts to a given distribution of data keys stored by the peers.

The process is based on pair-wise interactions of peers in which they locally decide whether to modify the routing infrastructure (by path extension or retraction) in a given data key subspace, if the present data justifies such a modification. As a result the shape of the (virtual) trie underlying the construction of routing tables will adapt to the data key distribution. Thus we achieve a uniform load distribution for peers with respect to storage (and querying assuming uniform query distribution).

This leads to an interesting problem with respect to search. In the worst case, for degenerated data key distributions, the tree shape no longer provides an upper bound for search cost as it might be up to linear depth in network size. However, it can be shown by theoretical analysis that for a (sufficiently) randomized selection of links to other peers in the routing tables, probabilistically the search cost in terms of messages remains logarithmic, independently of the length of the paths occurring in the virtual tree [2].

Another aspect of load balancing is uniform replication of data to support uniform availability. In current structured P2P systems this problem is typically tackled by controlled replication, where a globally constant replication factor is assumed. Besides introducing global knowledge into the systems, which is undesirable from the viewpoint of decentralization and peer autonomy, this approach also lacks the ability to adaptively exploit existing storage resources in an optimal manner.

In contrast, we use an adaptive, self-organizing mechanism to globally balance data replication. Different to storage load, peers cannot locally detect non-uniform replication of data in the entire network. We employ a sampling-based method to detect imbalance and to dynamically adapt replication. Thus data will be dynamically replicated while peers aim at using their storage capacity optimally. An important aspect is the mutual dependency among storage load balancing and uniform replication. When peers attempt to locally balance their storage load they may compromise globally uniform replication. By simulation we show for our approach that the system converges to a state where both load balancing goals are achieved in combination. This reactive load-balancing of replication factor in a self-organized manner is possible in P-Grid without affecting the structural properties of the system because of the independence of peer identifier and data (keys) associated with the peer.

With P-Grid we have shown that self-organization principles can also apply to structured P2P systems. However, different to the situation in unstructured systems, where search algorithms are designed in order to take advantage of the emergent overlay network structures, we design the self-organization process to converge to an overlay network such that provable efficient search algorithms can be applied and at the same time load balancing goals are achieved.

3 Updates in P-Grid

Until recently P2P systems were primarily used for sharing static, read-only files. Thus most P2P systems did not provide update mechanisms that would work in the presence of replication. For example, centralized (or hierarchical) P2P systems, such as Napster or FastTrack, maintain a centralized index of data items available at online peers. If an update of a data item occurs this means that the peer that holds the item changes it. Subsequent requests would get the new version. However, updates are not propagated to other peers which replicate the item. As a result multiple versions under the same identifier may co-exist. The same holds true for most decentralized systems such as Gnutella.

Some systems partially address updates. For example, in Freenet an update is routed "downstream" based on a key-closeness relation. Since the network may change during this and no precautions are taken to notify peers that come online after an update has occurred, consistency guarantees are limited.

To address updates in a decentralized way we have designed an update algorithm [10] based on rumor spreading that provides probabilistic guarantees for consistency and is compatible with the self-organizing nature of P-Grid. It was inspired by the fundamental work on randomized rumor spreading presented in [13]. The update algorithm is efficient (analytically proven) and based on a generic push/pull gossiping scheme for highly unreliable, replicated environments, dealing with the realistic situation that peers are mostly offline. [10] provides an analytical model to demonstrate the significant reduction of message overhead using optimizations techniques (partial lists) and proper tuning of the gossiping

(push) phase which in consequence improves the scalability of the algorithm. The efficiency of the pull phase depends solely on the efficiency of searches in the P2P system. The analytical model for the gossiping algorithm is a significant contribution in contrast to most of the literature in this area which relies solely on simulation results. Since our algorithm is generic the analytical model is valid for many of the other variants of flooding algorithms and so are the results of our analysis. The algorithm is totally decentralized and uses no global knowledge but exploits local knowledge instead which makes it suitable for the P2P, mobility, and ad-hoc networking domains.

Some of the services discussed in the following such as dynamic address management (Section 4.1) and trust (Section 4.2) depend heavily on the provision of update functionality.

4 Self-Organizing Services

This section presents identity and trust management as two sample, self-organizing services implemented on top of P-Grid.

4.1 Handling Dynamic Addresses and Identity of Peers

As IP addresses have become a scarce resource most computers on the Internet no longer have permanent addresses. For client computers this is usually not a big problem but with the advent of P2P systems, where every computer acts both as a client and as a server, this has become increasingly problematic. In advanced P2P systems ad-hoc connections to peers have to be established, which can only be done if the receiving peer has a permanent IP address. To handle this we have designed a completely decentralized, self-maintaining, lightweight, and sufficiently secure peer identification service based on P-Grid. It allows us to consistently map unique peer identifiers, in particular the logical identity of peers used for routing in P-Grid, onto dynamic IP addresses. It is designed to operate in environments with low availability of the peers [12].

The basic idea is to store the mappings in P-Grid itself: Peers store their current id/IP mapping in P-Grid and update it if the IP address changes (for example, if they come online again). For routing search requests while searching id/IP mappings using P-Grid's routing infrastructure peers use cached id/IP mappings. If cached entries are stale they are updated by recursively querying the P-Grid again. Although at first sight this may look as an unsolvable, recursive "hen-egg problem," we demonstrate that not only most of the original queries will be answered successfully, but also, that the recursions triggered by failures will lead to a partial "self-healing" (a different form of self-organization) of the whole system by updating the caches.

For security we apply a combination of PGP-like public key distribution and a quorum-based query scheme. The public keys themselves are stored in P-Grid, and replication can provide guarantees that are probabilistically analogous to PGP's web of trust. The approach can easily be adapted to other application domains, i.e., be used for other name services, because we do not impose any constraints on the type of mappings. Motivated by the problem of handling peer identity in a setting where peers' physical addresses change because of network dynamics we thus achieved a self-contained and self-maintaining directory service for P-Grid.

4.2 Trust as the Basis for E-commerce

The vast majority of interactions among peers in a P2P system are between complete strangers who do not have any prior knowledge about each other. Since peers are fully autonomous this leaves much room for exercising opportunistic behavior of various forms, ranging from "free riding" in file sharing P2P networks to fraud and deception in e-commerce related interactions. Researchers have recognized the importance of this problem [7] and trust and reputation management, as a social control mechanism, has been accepted as an appropriate solution.

In [1] we present our decentralized trust management model that analyzes past interactions among peers to make a probabilistic assessment of whether any given peer cheated in its past interactions. The emphasis is put not only on assessing trust but also on providing a scalable data management solution particularly suitable for decentralized networks. To this end, we apply P-Grid in such a way that for any particular peer we designate a set of replicas to store the ratings of trust-related behavior of that peer (complaints filed by it about others and complaints filed by others about it) so that the reputation data can be accessed and collected in logarithmic time. As replicas may provide false data, an appropriate replication factor along with a proper voting scheme to identify the most likely correct reputation data set are applied to achieve accurate predictions. Trust assessments themselves are made based on an analysis of peer interactions modeled as stochastic processes. As it was shown by simulations, cheating behavior of the peers can be identified with a very high probability. The model is simplistic in the sense that, for any peer,

the sense that, for any peer, it decides whether it cheated in the past or not. Extensions that would give probabilistic estimates of the peers' future behavior are currently underway.

Since we use P-Grid's directory service to report and store the reputation related information, we implicitly employ the peer identification service presented above, thus preventing distributed denial of service attacks originating from impersonation or trust data manipulation. Such resilience for higher level services derived from lower levels of the P-Grid system highlight P-Grid's self-organizing features that span beyond a communication network buildup.

Building on the basic trust model we have also made some further steps towards fully-blown P2P markets. [11] presents our solution to the problem of self-enforcing exchanges of digital goods, while in other work we propose a fully decentralized double auctioning mechanism based on the continuous double auction scheme.

5 Conclusions

P2P systems are commonly classified into two categories: unstructured systems (e.g., Gnutella) exposing emergent phenomena driven from purely local interactions, and structured (DHT-based) systems with probabilistic execution guarantees. P-Grid combines the best of both worlds, using self-organization principles for constructing and maintaining a DHT-like routing infrastructure. It takes advantage of the resulting emergent properties for improving various services including routing, updates and identity management. One may also benefit from self-organizing principles when dealing with higher-level abstractions such as trust or global semantic interoperability [4], [6].

What started as a purely decentralized index structure is gradually evolving into a general-purpose distributed infrastructure. We have implemented P-Grid in Java and are currently in the final test phase. More information about P-Grid may be found on the project's web page at http://www.p-grid.org.

References

[1] Karl Aberer and Zoran Despotovic. Managing Trust in a Peer-2-Peer Information System. *10th International Conference on Information and Knowledge Management (ACM CIKM)*, 2001.

[2] Karl Aberer. Scalable Data Access in P2P Systems Using Unbalanced Search Trees. *Proceedings of Workshop on Distributed Data and Structures (WDAS-2002),* 2002.

[3] Karl Aberer, Manfred Hauswirth, Magdalena Punceva, and Roman Schmidt. Improving Data Access in P2P Systems. *IEEE Internet Computing*, 6(1), Jan./Feb. 2002.

[4] Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth: A Framework for Semantic Gossiping. *SIGMOD Record*, 31(4), Dec. 2002.

[5] Karl Aberer, Manfred Hauswirth, Magdalena Punceva. Self-organized construction of distributed access structures: A comparative evaluation of P-Grid and FreeNet. *5th Workshop on Distributed Data and Structures (WDAS'2003)*, 2003.

[6] Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth: The Chatty Web: Emergent Semantics Through Gossiping. *Twelfth International World Wide Web Conference (*WWW2003), 2003.

[7] Eytan Adar, Bernardo A. Huberman. Free Riding on Gnutella. *First Monday* 5(10) 2000. http://firstmonday.org/issues/issue5_10/adar/index.html

[8] Ian Clarke, Scott G. Miller, Theodore W. Hong, Oskar Sandberg, and Brandon Wiley. Protecting Free Expression Online with Freenet. *IEEE Internet Computing*, 6(1), Jan./Feb. 2002.

[9] Clip2. The Gnutella Protocol Specification v0.4 (Document Revision 1.2), Jun. 2001. http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[10] Anwitaman Datta, Manfred Hauswirth, and Karl Aberer. Updates in Highly Unreliable, Replicated Peer-to-Peer Systems. *23rd International Conference on Distributed Computing Systems*, 2003.

[11] Zoran Despotovic and Karl Aberer. Trust-Aware Delivery of Composite Goods. *International Workshop on Agents and Peer-To-Peer Computing*, 2002.

[12] Manfred Hauswirth, Anwitaman Datta, and Karl Aberer. Handling Identity in Peer-to-Peer Systems. *6th International Workshop on Mobility in Databases and Distributed Systems, in conjunction with DEXA'2003*, September 1-5, 2003 (to be published).

[13] Richard M. Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vöcking. Randomized rumor spreading. *IEEE Symposium on Foundations of Computer Science*, 2000.

[14] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. Search and replication in unstructured peer-to-peer networks. *International Conference on Supercomputing*, 2002.

[15] M. Ripeanu and I. Foster. Mapping the Gnutella Network: Macroscopic Properties of Large-Scale Peer-to-Peer Systems. IPTPS 2002.

[16] N. Sarshar, V. Roychowdury, P. Oscar Boykin. Percolation-based Search on unstructured Peer-To-Peer Networks, IPTPS 2003.

[17] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. *ACM SIGCOMM*, 2001.