Peanuts: A Top-Down Peer-to-Peer Network

Peter Mahlmann* Christian Schindelhauer

We introduce Peanuts – a top-down Peer-to-Peer network combining the benefits of reliable random graphs and semantic search trees. The main goal of Peanuts is to overcome the restricted query languages of almost all current peer-to-peer networks. These restrictions are a consequence of the use of distributed hash tables where any semantic relationship of the data is lost. In a top-down approach such semantic relationship is preserved since the peers are assigned to the data and not vice versa.

Peanuts allows nontrivial lookups, like range queries, neighborhood search, and estimation of the popularity of prefixes. Furthermore, the maintenance of the network structure is kept local so that periodic handshake operations suffice to maintain the network structure. Simplifying, Peanuts supports three kinds of locality what results in the possibility to support non-trivial queries:

Network Locality: Lookup operations can be performed with small latency.

Information Locality: Closely related data elements are stored on network-wise close peers.

Interest Locality: Peers can choose on providing lookup service and data storage for certain data. If peers choose to provide certain data, the network structure will improve their lookup of related data.

Random graphs are Peanuts main component to build arbitrary search trees supporting these kinds of locality. There is a *Reliable Backbone* which is a random graph consisting of all participating peers. This Backbone represents the root of the semantic search tree. The backbone network should be simple and reliable under churn. Therefore, the backbone is realized by a random network. A random network has several advantages compared to deterministic structured networks. For example the maintenance costs in highly dynamic networks are minimal, since there is no predetermined neighborhood. In [1] we introduce a simple scheme to maintain such random networks. The main component of this scheme is a simple periodically performed local link exchange, guaranteeing connectivity and establishing expander graphs with high probability.

The peers of the backbone are recursively assigned into sub-sets, each representing one child node of the current tree node. Instead of the usual approach of hashing data onto peers, we hash peers onto data. The assignment of peers to tree nodes is done using weighted consistent hashing [2], an extension of consistent hashing to support non-uniform load distributions. In Peanuts, the number of peers assigned to a tree node depends on its load, i.e. the amount respectively popularity of data stored on this nodes subtree. This way unbalanced trees can be handled efficiently.

Peanuts is currently implemented in cooperation within the EU-project DELIS to provide an efficient peer-to-peer based data structure for the implementation of a distributed Web search engine.

References

[1] Peter Mahlmann, Christian Schindelhauer, Peer-to-Peer Networks based on Random Transformations of Connected Regular Undirected Graphs, 17th ACM Symposium on Parallelism in Algorithms and Architectures, (SPAA 2005), Las Vegas, NV, USA, July 17-20, 2005

[2] Christian Schindelhauer, Gunnar Schomaker, Weighted Consistent Hashing, 17th ACM Symposium on Parallelism in Algorithms and Architectures 2005, (SPAA 2005), Las Vegas, NV, USA, July 17-20, 2005

*{mahlmann,schindel}@upb.de, Department of Computer Science and Heinz Nixdorf Institute, University of Paderborn, Germany. Partially supported by the DFG-Sonderforschungsbereich 376 and by the EU within 6th Framework Programme under contract 001907 “Dynamically Evolving, Large Scale Information Systems” (DELIS).