A Simulation framework for schema-based query routing in P2P-networks

Wolf Siberski and Uwe Thaden

Learning Lab Lower Saxony, 30539 Hannover
`{siberski,thaden}@learninglab.de`

Abstract. Current simulations of P2P-networks don't take any kind of schemas into account. We present a simulation-framework and first results for query routing based on extensible schema information to describe peer content, providing more value than simple categorizations like the filename as abstraction for an MP3-song. Using different parameterization, we compare the impact of introducing the HyperCuP-topology in a P2P-network for routing and possible clustering in super-peers and discuss first simulation results. We take into account the importance of the Zipf-distribution which is known for being the typical content distribution in internet networks.

1 Introduction

Metadata and schemas are important for both peer-to-peer (P2P) networks and databases. Our open source project Edutella [1, 2] offers an infrastructure combining semantic web and peer-to-peer technologies in order to make distributed learning repositories possible and useful. It is based on the exchange of RDF metadata and allows to query different data sources. As a schema-based peer-to-peer network, Edutella extends conventional peer-to-peer networks by allowing different and extensible schemas to describe peer content, a necessary feature for information rich peer-to-peer networks. In Edutella everything is based on the Resource Description Framework (RDF) and RDF Schema, which allows to represent schemas based on classes, properties and property constraints. In our educational context schemas used in Edutella are e. g. Dublin Core (DC) or Learning Object Metadata (LOM). These are standardized categorizations one can use to store metadata going further than simply storing the filename of a music-file.

For traditional databases, schemas are nothing new, for peer-to-peer networks such approaches are just beginning to emerge. There are some aspects in that field which bring knowledge from the database community and knowledge from the P2P-community together. In their nature peers in a P2P-network are inhomogeneous regarding their technical aspects (storage-power, up- and downtime, etc.) and their usage from the topic-point of view. These facets bring the focus to the questions how to efficiently connect peers (topology) and how to extend representation and querying over P2P-networks (schemas).

Recent research has focused either concentrated on gaining knowledge using crawls of systems like Gnutella or using simulations which show message-behavior etc. on a very low (i. e. transport) level. Ideas like schema-based peer-to-peer networks cannot be simulated like that since the overlay-topology is the point of interest. On the other hand it is not possible to setup a large P2P-network to test how efficient a topology with different parameters is regarding search- and query-algorithms.

Using a schema-based approach, we had to find a good way how the schemas and their attributes are distributed over the network. Current research has shown that consumers in a P2P-network are interested in subsets of all available content and that they are often only interested in some content categories only [3]. E.g. for our eLearning-context we can say that students are mainly searching for resources related to their current courses. It was observed that in the domain of information retrieval the documents are distributed following Zipf's law. This means that many consumers are interested in some resources which are held by few providers. Recent (empirical) studies [4–6] have shown that despite the randomness of the internet [7], it also exhibits the Zipf distribution.

The remainder of this paper is as follows: In section 2 we analyze which requirements are needed to simulate a schema-based P2P-network. Section 3 presents our simulation framework for schema-based P2P-networks. Some first results/hypotheses of our simulations are discussed in section 4.

2 Simulation Context

Ehrig et al. [8] present a theoretical model of evaluation. They discuss several aspects of a P2P-simulation and give some recommendations, but no final set of parameters. Our following list of requirements has some of their ideas and new ones combined to form a set of requirements that can be implemented and analyzed after the simulation runs.

2.1 Schema-based resource description

We assume that there won't be one fixed schema to describe resources in a P2P network. Instead, peers will choose one of (more or less) standardized schemas for resource description(s). This is a trend in recent P2P systems [9–12]. In Edutella we use RDFS for as schema definition language. For our simulation we only assume that a schema is identifiable and consists of arbitrary many named properties. We don't take into account any relation between properties.

2.2 Super Peer based Topology

While the simulator framework is not tied to a specific topology, we assume a super-peer topology, where the super-peers form a backbone of the network and take care of request routing. Only a small percentage of nodes are super-peers, but these are assumed to be highly available nodes with high computing capacity. For example in our learning repository network, each university would run one such super-peer.

Super-peer routing is usually based on different kinds of indexing and routing tables, as discussed in [13] and [14]. The Edutella super-peers employ routing indexes which explicitly take schema information into account.

Super-peers in the Edutella network are arranged in the HyperCuP topology [15]. The HyperCuP algorithm is capable of organizing peers in a P2P network into a recursive graph structure from the family of Cayley graphs, out of which the hypercube is the most well-known topology. The hypercube topology allows for $\log_2 N$ path length and $\log_2 N$ number of neighbors, where N is the total number of nodes in the network (i.e. the number of super-peers in our case). In the simulation runs described in section 4 we discuss characteristics of the hypercube topology and the influence of sorting/clustering peers and super-peers.

2.3 Possible distributions

Our simulation framework is open to any kind of distribution of the schemas, schema properties, etc. Looking at the current research [4–6] we know that a typical distribution of information (i.e. content in a P2P-network) follow Zipf's law.

Zipf's law, named after the Harvard linguistic professor G. K. Zipf, comes from research in the 1930s. It is one of the most empirical validated laws in the domain of linguistic quantities. If we count the number of times each word appears in a text (called frequency) and assign each word a rank based on its frequency (i. e. rank=1 is the word that appears the most), we can see that the product frequency x rank for each word is roughly equal to a constant. In general, it is the observation that the frequency of occurrence of some event, as a function of the rank is a power-law function [16]. Zipf showed this by other examples, e. g. the population of cities. The population of cities plotted as a function of the rank is a power-law function. It was shown that Zipf is quite accurate except for very high and low rank. Plotted both axes as log, power laws give a straight line.

Today it is taken as the most validated law on distribution in many empirical studies. Research in the last months showed that also the internet follows a Zipfian distribution in various aspects (e.g. content distribution, number of links). Initially the internet was impressing by its variety in the size of its features. Then soon it was discovered a widespread pattern in measurements: Most objects in the internet are small, but only few large ones. Most of the sites contain a very limited number of sites, a few sites consist of millions of pages. The in- and out-degree of sites are mostly low while some site have many links. This leads to the assumption that many users are interested in only a few selected sites, giving little attention to millions of others. This implies that a small number of users is responsible for most of the requests.

In our paper, we assume that Zipf is not only valid for content in networks, but also for schema representing the categories content is described by. While there are no other results (maybe because the topic is too new), we have no reason to believe that there will be another distribution than Zipf.

2.4 Existing Simulators

This section gives an overview of other P2P simulation frameworks and compares them to our work.

The **SimP2 simulator** [17] is designed to provide support and additional depth to an analysis of ad-hoc P2P-networks. The analysis is based on a non-uniform random graph model similar to Gnutella, and is limited to studying basic properties such as reachability and nodal degree. They leave out complex queries which are very important for our approach, since we want to broadcast such queries efficiently. On the other hand, SimP2 is very good for more detailed performance characteristics such as queuing delays and message loss.

3LS [18] is a discrete simulator using a central step-clock. It provides three levels: Network model, protocol model and user model. The network model uses a two dimensional matrix to define distance values between the nodes. The protocol model represents the P2P-protocol which should be investigated. Input can be simulated using the user model (which could be a interesting addition to our simulation-framework). Since 3LS is not efficient regarding memory usage, it is limited to rather small networks.

The authors of the **Packet-level Peer-to-Peer (PLP2P) Simulator** [19] state that one of the most important things in a simulation is the correct and mostly complete underlying network-structure. They assert that failure to consider low-level details can lead the simulation to inaccuracies. PLP2P provides a framework that can be used together with other simulators to achieve more accuracy in the simulations.

Narses Simulator [20] is a flow-based network simulator and thus does not concentrate on the packet-level to avoid the overhead of packet level simulators. To do this Narses offers a range of models that trade between fast runtimes and accuracy. Narses is therefore somewhere between packet level simulators and analytical models. Nevertheless the assumptions made by Narses are targeted towards reducing the complexity if simulations by approximations of physical aspects.

Evaluation Regarding our plan to simulate a schema-based peer-to-peer-network, none of the current available simulators is capable of that, since they all concentrate on the traffic or information-’flow’ on a much deeper level of the OSI-model. The observations are made directly from the transport-level or by making abstraction or assumptions on the (physical) aspects which are in contrast to our needs. For our purpose we need a way to describe a specific topology in combination with schema-information, so that we can get results for search and routing in schema-based peer-to-peer networks. Furthermore most simulators cannot be used to simulate different topologies with several parameters as we need it for our task to compare different shapes of the HyperCuP-topology. To overcome these problems, we developed a simulator-framework which is described in the next section.

3 The Simulation Framework

The following sections describe our design and implementation of our simulation framework. We assume some basic knowledge on simulation. A good introduction can be found in [21].

3.1 Design

Schema based P2P-networks are a subset of P2P-networks as they are used for e. g. sharing music-files. So what we needed was a tool set for creating the ”normal” requirements for a P2P-simulation like message-exchange and a simulation-stepper. For that we used a framework called SSF. Furthermore we needed to model and implement the behavior of a P2P-network that uses schemas.

Schema-based resource description The main goal of our simulation is to experiment with query routing based on schema information. To represent this information, we use schema elements which can be either complete schemas or single properties (the term property stems from semantic web terminology; in a relational database a schema property would correspond to a table column).

Query messages don't contain concrete requests, but only a list of properties used to formulate the request. Provider peers 'answer' to these requests on a probabilistic basis, depending on the schema information used by the provider and the information used in the query. For example, our model of a query which asks for *(dc:title*[1] $=$ *"The Power of Metadata"' and dc:date* $>$ *"1.1.2000")*, is just a list of the used properties *(dc:title, dc:date)*.

For the generation of such queries a configurable distribution is taken into account. We can set the following parameters:

- the number of available schemas and their frequency distribution
- the average number of properties per schema (and deviation)
- the average number of properties used in a query (and deviation)

The same applies to peer content. When a peer is created, we do not assign content to them but only schemas and/or schema properties which this peer is presumed to use for its content. For this assignment, the same schema and property distributions are taken into account as for the query generation. Additionally, the average number of schemas and properties (and deviation) used by a peer can be configured. When a query is received by a peer and matches its assigned schema elements, an abstract response is generated with a configurable probability.

For our network it makes no difference whether the queries originate at peers or directly at super-peers. The generated queries are distributed evenly to the super-peers input queues.

This approach allows to simulate the routing behavior without needing to generate huge amounts of test data.

Super Peer based Topology The simulation framework assumes a super-peer topology. All simple peers have exactly one connection to a super-peer. The super-peers form their own peer-to-peer network (it would be possible to simulate a conventional P2P network by instantiating the super-peer backbone only). The super-peer network topology and protocol is pluggable. For our first experiments we used only the HyperCuP topology.

In contrast to other simulations our approach doesn't rely on a TCP/IP network simulation, but models connections between peers on a higher level. Any connection has a bandwidth (specified by messages per second) and a delay (in msec). Both properties are modeled as normal distributions with configurable deviation. As we assume that SP/SP connections typically have a higher capacity than SP/P connections, these parameters can be set separately for these connection categories.

---

[1] dc is used here as abbreviation for the Dublin Core metadata schema

Because super-peers are assumed to be highly available, we don't model their up- and downtime, but simulate using a static backbone. This makes it very simple to create different super-peer topologies because it isn't necessary to implement a full connection/disconnection protocol. Instead, a topology class creates all super-peers and the connections between them on simulation startup. Of course, the implementation for the real network has to consider joining and leaving super-peers. But, as super-peer joins or failures will be rare, their influence on the network performance won't be significant.

In contrast, peers will join and leave the network frequently. We model this by a giving each peer a designated lifetime, which is assigned according to a configurable distribution.

Connections (network characteristics) All connections are bi-directional. Each peer (including super-peers) has an incoming message queue per connection, one processing queue and an outgoing message queue per connection. We can configure the time necessary to process a message and the number of processors available at a peer. Messages between the peers are interpreted as discrete events.

3.2 Implementation

Our implementation is based on the discrete simulation framework SSF (Scalable Simulation Framework [22]). The Scalable Simulation Framework is an open standard of discrete event-simulations.

The general layer is responsible for establishing the super-peer topology and the connection between peers and super-peers. Instead of using an IP network simulation as foundation, connections are specified by only two parameters, bandwidth (in number of messages per second) [2] and latency (in milliseconds). For both parameters average and deviation can be specified.

The SSF provides an interface for discrete-event simulations supporting object-oriented models to utilize and extend the framework. Extended the framework by this the potential for direct reuse of model code is maximized, while the dependencies on a particular simulator kernel implementation are minimized. The framework's primary design goal was to support high performance simulations and to make models efficient.

The SSF provides several classes that we used to map the P2P-behavior to the mode. The *Entity* is the central class in SSF. Entities can have processes for event-processing. In our simulator the peers are implemented using entities. An *Event* changes the status of the system or is used for communication between entities. Regarding our simulator when use the events as messages between the peers. *Processes* are used to handle events during the simulation. An entity can have one or more processes. *In- and out-channels* are the communication channel between the entities. An entity can have several in- and out-channels, which are always connected 1:1.

The configuration of the simulator is very simple using three XML-files which define the topology, duration of the simulation, time to live (TTL) for messages, number of peer, etc.

---

[2] As our network is not concerned with transport of the content, only with content description, messages don't vary much in size

4 First Simulation Results

The most interesting question for us was how clustering of the peers according to their schema influences the routing efficiency in the super-peer network. Therefore, for our first experiments we focused on this issue. We had the following hypotheses:

4.1 Hypotheses

1. Clustering peers at super-peers according to their schema will reduce query distribution effort significantly.
2. Clustering super-peers according to their schema (the schema of their peers) will furthermore reduce query distribution effort.

We didn't include a hypothesis about the influence of increasing the number of peers, because in our approach this can already be predicted. If the peers are clustered, then adding new peers will not change the query distribution within the super-peer network. As we currently distribute any query to any peer which uses the corresponding schema, the number of messages between super-peers and peers will grow linearly with the number of peers. See section 5 for proposals to improve this ratio.

4.2 Experiments

We compared three different approaches:

Peers and super-peers randomly distributed. In this case peers connect to super-peers in a random fashion, independently of the schema they use (see 1 for an example). This scenario is abbreviated with U (unclustered).

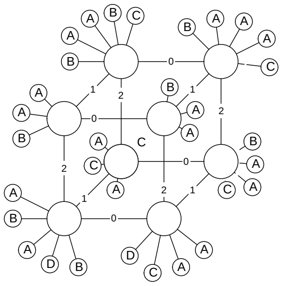

Fig. 1. Example of a network with arbitrary peer distribution

Peers clustered, super-peers randomly distributed. Here the super-peers collect peers using the same schema. The super-peers are still placed in the hypercube at a random position, regardless of their peers schema information (2). We try to distribute the load evenly by assigning approximately the same number of peers to each super-peer. Therefore, for rare schemas super-peers will take the responsibility for several schemas. We use the short-hand P (peers clustered) for this scenario.

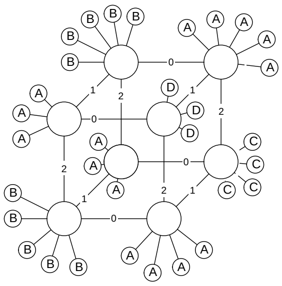

Fig. 2. Example of a network where super-peers collect peers using the same schema, but are placed at arbitrary hypercube positions

Peers and super-peers clustered. In this variant we try to find optimal positions for super-peers in the hypercube as well, depending on the schema information. We have to optimize the hypercube for the most frequent schemas; a promising approach is sorting the schemas in hypercube dimensions according to their frequency. Dimension 0 is assigned to the most frequent schema, and therefore a query regarding this schema will be in the right partition of the hypercube after one hop. Queries regarding the second most frequent schema are in the right partition after two hops, etc. (see 3). In the following we refer to this scenario as SP/P (super-peers and peers clustered).

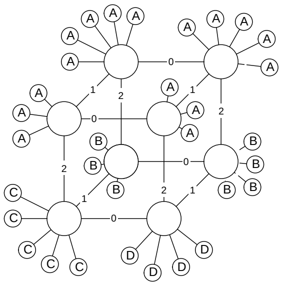

Fig. 3. Example of a network where super-peers are clustered by schema

We simulated a network with 64 super-peers and 10000 peers. We estimate that such a network would suffice to connect a large percentage of all German university learning material providers. The number of schemas in use was set to 32 and a response probability of 5% was assumed. While our framework allows the usage of more fine grained schema elements, we chose to start with a simplified scenario, and to refine it step by step, guided by the results of the completed experiments.

For each scenario 1000 queries were distributed in the network. As the distribution algorithm doesn't depend on previously evaluated queries, the comparatively low number of queries is sufficient to avoid arbitrary results.

The usage probability of the schemas follows a Zipf distribution (skew factor 0.1). This distribution is used to calculate the number of peers which use a specific schema to describe their content as well as the number of queries formulated using this schema.

4.3 Results

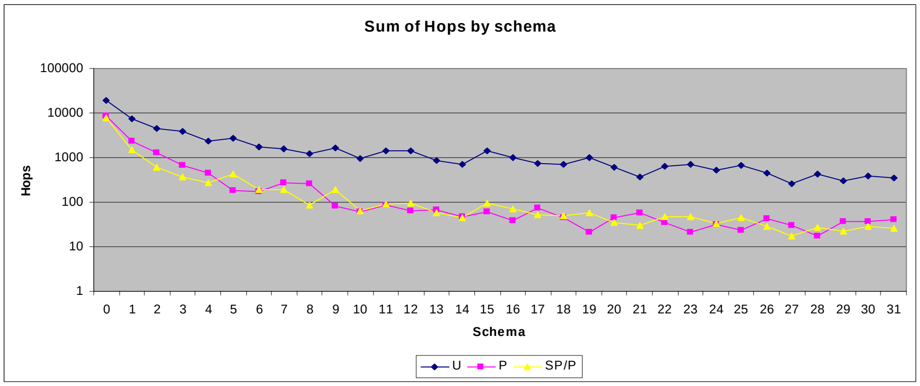

Fig. 4. Sum of super-peer hops needed to distribute queries

Figure 4 shows the sum of hops which were necessary to distribute the queries sorted by schema. For example, to distribute all queries regarding schema 0, we needed nearly 20.000 hops in scenario U, but only about 8500 in P. These results show that clustering peers at the super-peers has a substantial effect on query routing performance. The number of queries a super-peer has to handle on the average is reduced significantly. We can say that hypothesis 1 has been confirmed.

Arranging the super-peers in the hypercube according to their schemas has only a very small effect; hypothesis 2 has not been confirmed in this experiment. Assigning optimal positions to super-peers in a decentralized and efficient manner seems to be a very complex self-organizing task (especially if the hypercube has to perform a dimension increment or decrement). The first results at least indicate that clustering peers alone is a sufficient optimization.

As we saw, clustering reduces the load of the network. However, this comes at a price regarding the load distribution. Fig. 5 shows the minimum, average and maximum number queries a super-peer had to handle. For scenarios P and SP this load is becomes distributed much more unevenly. The reason is that the clustered case super-peers responsible for the more frequent schemas bear a higher load, because they get more queries.

As scenario P turned out to be the most interesting, we varied the number of super-peers between 1 and 1024 to evaluate the influence of the backbone size.

Fig. 6 shows the average load per super-peer for these different sizes. For example, in the case of the 4-node network, on average each super-query has to process a

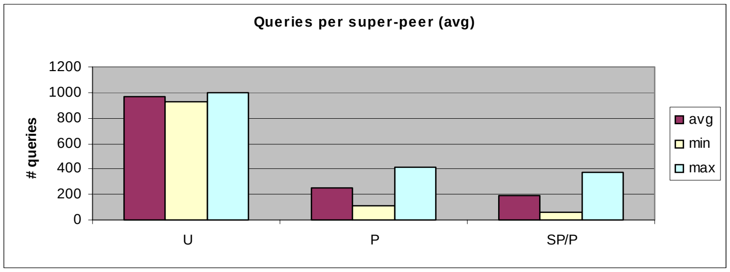

Fig. 5. Number of queries

little more than 200 queries related to schema 0. In the (extreme) case of a 1-node 'network', the super-peer has to process all (273) queries related to schema 0. The average super-peer load is reduced when increasing the backbone size but the gain becomes insignificant for larger networks.

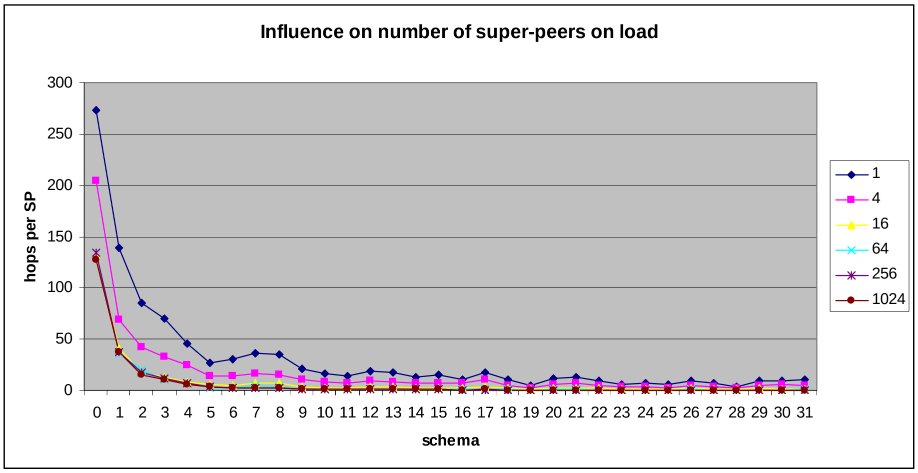

Fig. 6. Super-peer load in various network sizes

4.4 Consequences

Currently, a complete answer is retrieved for each query. This results in a linear increase of messages proportional to the increase of the number of peers. We are not able to compensate for this by enlarging the super-peer network.

Therefore, to reduce the amount of processing, we need to restrict the number of responding peers and/or super-peers. We see following options to achieve that goal:

- **Introduction of a Top-k query evaluation approach.** The most promising technique to reduce the network load seems to restrict the number of responses. One approach would be to let each super-peer wait for responses from its own peers until it forwards the query. If sufficiently good responses can be retrieved, the query isn't further distributed. Otherwise, a result counter within the query is incremented by the number of matches found, and then the query is forwarded.
- **Result Caching** Caching frequent matches and answering from the cache first could also result in a significant improvement. For example, [23] shows that load balancing can be achieved by replicating content within a cluster of peers based on a fairness metric based on content popularity. Other approaches are described in [24] and [25].
- **Peer preselection** Super-peers could store statistics about the response rate for their peers and forward queries to the most promising peers first. The other peers would get the query only if the first step didn't produce sufficient results.

5 Conclusion and Further work

Current approaches don't support the simulation of schema-based P2P networks. We have collected a minimal set of requirements for the simulation of such networks and implemented a corresponding simulation framework.

We could confirm our hypothesis that schema-based clustering of peers at the super-peers improves the network performance significantly.

To get more detailed results, we will analyze the time-based measurements also, after having conducted some real-world experiments to calibrate the simulation parameters. We also plan to extend the simulator to support different response probability distributions, to model different amounts of content at peers. Our next scenarios will include peer and super-peer dynamics (up- and downtime, lifetime) as well. Additionally, we will use more fine-grained peer content and query descriptions.

This will result in an improved prediction of the behavior of our P2P networks.

References

1. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., Risch, T.: EDUTELLA: a P2P Networking Infrastructure based on RDF. In: Proceedings of the Eleventh International World Wide Web Conference (WWW2002), Hawaii, USA (2002)
2. Nejdl, W., Wolpers, M., Siberski, W., Löser, A., Bruckhorst, I., Schlosser, M., Schmitz, C.: Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. In: Proceedings of the Twelfth International World Wide Web Conference (WWW2003), Budapest, Hungary (2003)
3. Crespo, A., Molina, H.G.: Semantic overlay networks for P2P systems. Technical report, Stanford University (2003)
4. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. ACM SIGCOMM Computer Communication Review **29(4)** (1999)
5. Chen, Q.: The origin of power laws in internet topologies revisited. In: 21st Annual Joint Conference of the IEEE Computer and Communications Societies. (2002)

6. Medina, A., Matta, I., Byers, J.: On the origin of power laws in internet topologies. ACM SIGCOMM Computer Communication Review **30(2)** (2000)
7. Gummadi, K.P.: Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In: 19th ACM Symposium on Operating Systems Principles (SOSP-19). (2003)
8. Ehrig, M., Schmitz, C., Staab, S., Tane, J., Tempich, C.: Towards evaluation of peer-to-peer-based distributed information management systems. In van Elst, L., et al., eds.: Agent-mediated Knowledge Management - AMKM-2003, AAAI Spring Symposium 2003, Stanford University, March 24-26. (2003)
9. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: The chatty web: Emergent semantics through gossiping. In: Proceedings of the Twelfth International World Wide Web Conference, New York, USA, ACM Press (2003) 197–206
10. Halevy, A.Y., Ives, Z.G., Mork, P., Tatarinov, I.: Piazza: Data management infrastructure for semantic web applications. In: Proceedings of the Twelfth International World Wide Web Conference (WWW2003), Budapest, Hungary (2003)
11. Bernstein, P.A., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihrayeu, I.: Data management for peer-to-peer computing: A vision. In: Proceedings of the Fifth International Workshop on the Web and Databases, Madison, Wisconsin (2002)
12. Nejdl, W., Siberski, W., Sintek, M.: Design issues and challenges for rdf- and schema-based peer-to-peer systems. SIGMOD Records (2003)
13. Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. In: Proceedings International Conference on Distributed Computing Systems. (2002)
14. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: Proccedings of the ICDE. (2003)
15. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: HyperCuP—Hypercubes, Ontologies and Efficient Search on P2P Networks. In: International Workshop on Agents and Peer-to-Peer Computing, Bologna, Italy (2002)
16. Adamic, L.A., Huberman, B.A.: Zipf's law and the internet. Glottometrics **3** (2002)
17. Kant, K., Iyer, R.: Modeling and simulation of ad-hoc/p2p file-sharing networks. In: Tools 2003. (2003)
18. Ting, N., Deters, R.: 3ls - a peer-to-peer network simulator. In: Third International Conference on Peer-to-Peer Computing. (2003)
19. he, Q., Ammar, M., Riley, G., Raj, H., Fujimoto, R.: Mapping peer behavior to packet-level details: A framework for packet-level simulation of peer-to-peer systems. In: MASCOTS. (2003)
20. Baker, M., Giuli, T.: Narses: A scalable flow-based network simulator. Stanford University Technical Report (2002)
21. Banks, J., II, J.S.C., Nelson, B.L., Nicol, D.M.: Discrete-event system simulation, third edition (2001)
22. Cowie, J., Liu, H., Liu, J., Nicol, D., Ogielski, A.: Towards realistic million-node internet simulations. In: International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA (1999)
23. Triantafillou, P., Xiruhaki, C., Koubarakis, M., Ntarmos, N.: Towards high-performance peer-to-peer content and resource sharing systems. In: Proceedings of the First Biennial Conference on Innovative Data Systems Research (CIDR 2003), Asilomar, CA, USA (2003)
24. Mondal, A., Goda, K., Kitsuregawa, M.: Effective load-balancing via migration and replication in spatial grids. In: Proceedings of the 14th International Conference on Database and Expert Systems Applications (DEXA 2003), Prague, Czech Republic (2003)
25. Stading, T., Maniatis, P., Baker, M.: Peer-to-peer caching schemes to address flash crowds. In: Revised Papers from the First International Workshop on Peer-to-Peer Systems, Springer-Verlag (2002)