Deliverable D6.3.1

Intermediate report on the requirements and restrictions of the desired P2P infrastructure, the capabilities of existing P2P systems, and proposed architectures
Start date of the project: January 2004
Duration: 48 months
Project Coordinator: Prof. Dr. math. Friedhelm Meyer auf der Heide
Heinz Nixdorf Institute, University of Paderborn, Germany

Due date of deliverable: December 2004
Actual submission date: January 2005
Re-submission date: April 2005
Version: 2
Dissemination level: PU – public

Work Package 6.3: Strategies for Self-Organizing Information Dissemination and Load Sharing

Participants: Research Academic Computer Technology Institute (CTI), Patras, Greece
Department of Computer Science, Universita di Bologna (UniBO), Italy
Wroclaw University of Technology (TU Wroclaw), Poland
Telenor Communication AS (Telenor), Oslo, Norway

Authors of deliverable: Peter Triantafillou (peter@ceid.upatras.gr)
Nikos Ntarmos (ntarmos@ceid.upatras.gr)
Ioannis Aekaterinidis (aikater@ceid.upatras.gr)
Theoni Pitoura (pitoura@ceid.upatras.gr)
1 Introduction and Overview

Work Package D6.3.1 is primarily concerned with self-organizing information dissemination and load sharing. The work conducted within DELIS for D6.3.1 addresses both dimensions of information dissemination and of load sharing.

With respect to information dissemination, our efforts have focused on addressing open problems within structured peer-to-peer (p2p) data networks and especially with networks based on Distributed Hash Tables (DHTs). In particular, we provide a solution to the problems of:

- efficiently processing range queries over DHT-based networks, [40]. Range queries have been neglected by such network architectures that only cater to simple, exact-match lookups and this has spurred considerable research into supporting range queries.
- supporting queries over string attributes over DHTs-based networks, which leads to a new paradigm for constructing publish/subscribe information dissemination networks, based on DHTs, [4] [61].

In regard to our work on load sharing in structured peer-to-peer networks we have:

- developed a new architecture that can address on the one hand the load balancing problem using the replication of hot data and, on the other hand, provide an efficient support for range queries [47]. This combined problem is particularly challenging as the replication of data yields load balancing since it allows different replicas to be randomly accessed. However, the efficient support of range queries is accomplished using clever data placement of consecutive values on the network. Thus, the random accesses imposed by data replication to ensure load balancing interfere with any clever data placement techniques used to support efficiently range queries. Our efforts have resulted into HotRoD: an architecture and related algorithms for replication management and utilization for load balancing and algorithms for efficient processing of range queries.
- In addition, our research efforts, under the name LION, [46] have focused on the issues of (i) providing natural definitions and metrics for measuring the fairness of load distributions in a p2p network and (ii) developing a DHT-based architecture that can be used to record and measure the fairness of the load distribution at any time, and to use this information online in a load balancing strategy.

2 Main Results on Load Sharing in Internet-Scale Data Networks

Our results are centred around the LION and HotRoD architectures and related algorithms.

2.1 LION (Lorenz –Load Info Overlay Network)

2.1.1 The Problem

This work addresses the issue of fair load distribution in peer-to-peer (P2P) networks from a global perspective. Significant load imbalances arise from skewed data and query distributions, from a high degree of heterogeneity in data items’ load and node capacities, as well as from continuous nodes’ arrivals and departures, and continuous data items’ insertions and deletions. Although there are plenty and significant solutions to address the issue of fair load distribution, to our knowledge, none of them addresses it from a global perspective. Our work specifically focuses on two main problems found in related work: the metrics problem and the local-optima problem.
The metrics problem. What is very much lacking in the P2P community is the definition of appropriate metrics which naturally capture the notion of fair load distribution and provide rich, compact information about this distribution. Furthermore, and perhaps more importantly, such metrics should not be simply utilized offline, i.e. to simply test the efficacy of specific algorithms after the fact (i.e. after they ran) but, in addition to driving and guiding the functionality offered by a specific algorithm online, they should help the algorithm to achieve fair load distribution.

The local-optima problem. It is true that without an overall appreciation of the state of the load distribution, imbalances may cause severe inefficiency and scalability problems.

2.1.2 State of the Art Overview

Load balancing goals in all related work ([20], [23], [25], [32], [48], [62]) focus on minimizing maximum load and avoiding overloaded nodes in P2P networks. None of this work, however, deals with creating a scaleable infrastructure to measure the fairness of load distribution among nodes and to ensure it. This work states and addresses the issues of the local optima and the metrics problems by proposing a global perspective to load distribution and defining inexpensive and rich load distribution metrics. It is also a generalized approach that can be applied on top of any P2P network.

2.1.3 Our Contributions

First, we define “inexpensive, accurate, natural and rich” load distribution metrics, in the sense that they are capturing naturally the notion of fair load distributions, and, thus, providing an accurate and rich, global picture of them. These metrics are the Gini and the Lorenz asymmetry coefficients, both used in economics and ecology to quantify and qualify inequality. The former expresses the total amount of load imbalance, and the latter indicates which load classes contribute most to the load’s total inequality.

Second, we develop an infrastructure, which we call LION [46] (Lorenz –or, Load – Info Overlay Network) with which the computation/evaluation of the above metrics and their maintenance is efficient and scalable and, hence, capable of being utilized online, guiding the functionality of any load balancing approach. LION comprises: (a) a K-node overlay network storing load information of the underlying P2P network in the form of load tuples, ordered by load, (b) algorithms to maintain load information on LION, and (c) algorithms to calculate load distribution metrics and ensure fair load distribution by load reassignment.

Our early analysis shows that LION appears to be promising in that it assesses the state of load distribution at any time and ensures measurably fair load distributions whereas it incurs sublinear costs (in number of messages/hops) for its frequent operations.

Ongoing work includes extensive experimentation to evaluate the proposed load balancing infrastructure and its optimizations, in both homogeneous and heterogeneous networks, and using multiple resources to count load (i.e. storage, access, and computation). Other important issues in the core of our ongoing efforts are the use of intelligent techniques for selecting the LION nodes from the nodes of the underlying network, and the exploitation of the Gini and the Lorenz asymmetry coefficients in order to predict the number of reassignments that are needed to bring the load distribution within a certain distance from the optimal.

2.2 HotRoD (Hot Rings/Ranges of Data)

2.2.1 The Problem

HotRoD is an architecture and related algorithms for studying the two problems of ensuring access load balancing and efficiently processing range queries in a structured peer-to-peer (P2P) data network. Placing consecutive data values in neighboring peers is frequently used in structured P2P
networks since it accelerates range query processing. However, such a placement is highly susceptible to load imbalances, which are typically handled by either transferring or replicating data.

One of the biggest shortcomings of DHTs that has spurred considerable research is that they support only exact-match queries. In case of range queries, each data value in a range should be queried individually, which is greatly inefficient and thus infeasible in most cases. Much related research deals with developing P2P systems that can support efficient range queries over DHT networks, but their solutions suffer from data access load imbalances in the presence of skewed access (query) distributions. There are a few solutions that handle the difficult problem of balancing data access loads in the presence of range queries. These solutions are based on data transferring which is inadequate in highly skewed access distributions, since transferring hot data items, simply transfers the associated hot spots. Hence, in such cases, access load balancing is best addressed using the replication of popular data values to equally distribute the access load among the peers storing these replicas.

2.2.2 State of the Art Overview

There is significant work dealing with either range query processing, or load balancing in P2P networks. However, the great majority of related research deals only with one of these problems, avoiding the most difficult problem of balancing data access loads in the presence of range queries, while speeding up their processing. This combined problem is addressed in [20], [23] and [32], where load balancing is based on transferring load from peer to peer. We expect that this will prove inadequate in highly-skewed access distributions where some values may be so popular that single-handedly make the peer that stores them heavy. Simply transferring such hot values from peer to peer only transfers the problem. Related research in web proxies has testified to the need of replication [64]. Replication can also offer a number of important advantages, such as fault tolerance and high availability.

2.2.3 Our Contributions

We draw upon earlier work, [62] where we defined a novel order-preserving hash function and data placement algorithms to facilitate fast range query processing. The key contribution [47] proposes algorithms for range query processing, load distribution, and replication management. The above can be applied to any DHT architecture, with only minor slight modifications. The proposed solution is tunable: by setting a parameter, i.e. the replication degree, we can tune between replication overhead costs and load imbalance. This is useful in cases we know, or can predict, the query workload. Our extensive experimental study of the proposed solution shows that with relatively small replication degrees, significant load balancing can be achieved, while continuing to provide efficient range query processing.

To our knowledge, this is the first work to concurrently address the issues of replication-based access load balance and efficient range query processing in structured P2P networks. Our main results show a significant hop count saving in range query processing from 5% to 80%, compared to an enhanced version of Chord with different range query spans. Furthermore, access load is more smoothly distributed among peers: for example, with a data replication overhead of about 100% we have found that the top 3% of the most-hit nodes receive 10% of the total load, compared to 60% of the total load they would receive without HotRoD. At the same time, the load is transferred to less-hit nodes some of which receive only a small load increase compared to a perfectly uniform load distribution. Also the further the range query spans, or the access skewness increases, the more the benefits of our solution increase.

Our future HotRoD-inspired research at the micro-level is planned to involve a mathematical study and performance fine-tuning of load tolerance thresholds and values for the maximum data replication factor and to ensure that certain performance properties hold (e.g. cost of replication,
number of replicas, hop counts, node hit count distributions, etc). In addition, at the macro-level, we plan to investigate range query optimization and join query optimization, exploiting both the ordering properties of our hash function (which ensure that the same joined attribute values are stored at the same nodes) and the available replicas in the system, and to derive methods for partitioning and parallelizing the processing tasks.

3 Main Results on Information Dissemination in Internet-Scale Data Networks

3.1 The RangeGuard

3.1.1 The Problem

In this work [40] we address the issue of efficient processing of range queries in DHT-based peer-to-peer data networks. The novelty of the proposed approach lies on architectures, algorithms, and mechanisms for identifying and appropriately exploiting powerful nodes in peer-to-peer (P2P) data networks, whose existence has been well documented in the literature.

DHTs provide us with strong probabilistic guarantees on the worst-case number of hops required to route a message from one node to any other node in the system. Unfortunately, all currently available DHT overlays are designed to support only single identity, exact-match queries. This has led researchers to investigate how they could enhance P2P systems to reply to more complex queries; however, up to now support for range query optimization remains an open problem. Nevertheless, one of the main characteristics of widely deployed P2P networks (e.g. Gnutella, Mojoration, etc.) is that participating peers are largely heterogeneous, with regard to their processing power, available main memory and disk storage, network bandwidth, and internet connection uptime. Relevant studies of P2P networks have shown that this large heterogeneity is also depicted in the distribution of the query processing chores across the node population. Therefore, we believe that harnessing the power of such powerful and altruistic nodes is the key to providing efficient range query processing in a P2P setting.

3.1.2 State of the Art Overview

Related work in range query processing mainly proposes architectures based on DHTs, such as Chord ([27], [62], etc), and CAN ([5], [17], etc). However, their efficiency is inferior compared to our proposed solution, either because they only provide approximate solutions ([27]), or because they provoke significant load imbalances [62], or because the underlying CAN routing protocol is expensive (in terms of hop counts) and less fault-tolerant, compared to Chord ([5], [17]). Karger and Ruhl [31] have presented a load balancing solution based on relaxing some of the constraints of Chord’s consistent hashing scheme: first, by allowing every node to take one of O(log(N)) positions on the Chord ring (N being the number of nodes), and then by completely decoupling the placement of nodes on the ring from any verifiable information. This latter decision is core to their approach to range query processing. On the contrary, our architecture keeps the secure-hash-based placement of nodes on the ring, while providing for efficient range query processing and storage/access load balancing. Finally, as pointed out in [51], appropriately using powerful nodes in the location/routing primitives of DHTs, can lead to both a more scalable and more efficient system.

3.1.3 Our Contributions

Our approach leverages existing DHT-based P2P research. It is based on:
1. extending the basic Chord architecture, using an order-preserving hash function and related
algorithms for tuple addition/deletion and routing requests (query processing for range queries),
and
2. a new architecture that facilitates the exploitation of powerful nodes in the network, assigning
specific tasks to them for further significant speedups during range query processing.

This architecture is based on adding a second order-preserving Chord ring, composed of powerful
nodes, the RangeGuards, burdened with extra routing state and functionality above an initial order-
preserving P2P data ring containing all nodes of the network. RangeGuards take responsibility
(using consistent hashing) for arcs (ranges) of nodes on the lowlevel Chord ring. In addition, this
work defines a way to identify and collect RangeGuards, and presents mechanisms to utilize them
during range query processing.

Our analysis and experiments show that the proposed RangeGuard architecture can significantly
improve the efficiency of range query processing in terms of average hop counts. Furthermore, our
architectures can achieve good storage and access load distribution. Finally, and perhaps most impor-
tantly, a key advantage of the proposed architectures is that they avoid the dangers and inefficiencies
of relying on weak nodes for range query processing, with respect to their processing, storage, and
communication capacities, and their intermittent connectivity.

3.2 The Publish/Subscribe Paradigm over DHTs

3.2.1 The Problem

The peer-to-peer (P2P) paradigm is appropriate for building large-scale distributed systems/applica-
tions. P2P systems are completely decentralized, scalable, and self-organizing. A large body
of research is currently targeting the extension and employment of DHTs for efficient data query
processing. The nature and functionality of the DHT-based P2P can guarantee the efficient man-
agement of queries with exact-match equality predicates. However, it is difficult to perform queries
with range predicates over numeric attributes and/or prefix, suffix, and containment predicates over
string attributes.

Our work on this area extends the functionality of traditional DHT-based P2P networks. We first
provide our solution that efficiently supports range predicates (=, <, >, etc.) on numeric values in
DHT-based systems [61]. The next step is to support a rich set of predicates on string attributes such
as equality, suffix, prefix, and containment, [4]. Specifically, supporting queries over string attributes
over a DHT infrastructure is an open problem.

We primarily focus on the pub/sub technology and formulate our solution in terms of a pub/sub
infrastructure built on top of a DHT-based P2P network. Our solution is independent of the type
of DHT and can be easily applicable to every DHT that can efficiently locate an object based on its
key identifier. The importance of our work lies in the fact that we are able to leverage DHTs as a
dominating technology for constructing efficient scalable overlay networks in creating a content-based
pub/sub infrastructure with extended functionality.

3.2.2 State of the Art Overview

The richest and most powerful publish/subscribe systems are the so-called content-based pub/sub
systems. Content-based systems give users the ability to express their interest by issuing continuous
queries, termed subscriptions, specifying predicates over the values of a number of well defined at-
tributes. Events are published into the system signifying the occurrence of a real-world event (such
as the addition of a data item, the execution of a stock-exchange transaction, etc.) The matching of
publications (events) to subscriptions (interests) is done based on the content (values of attributes).
Recently, some attempts on distributed content-based publish/subscribe systems use routing trees to disseminate the events to interested users based on multicast techniques [6], [9], [60], [44]. Some other attempts use the notion of rendezvous nodes, which ensure that events and subscriptions meet in the system [45].

3.2.3 Our Contributions

The main challenge in building a pub/sub system in a large-scale distributed environment is the development of an efficient distributed matching algorithm. In a pub/sub system an event is said to match a subscription if and only if all the subscription’s attribute constraints are satisfied. Each subscription has a k-bit identifier, called SubID. The main idea behind our solution is to store SubIDs for each subscription in appropriate nodes over a DHT network so as to force incoming events to check those nodes for potential matching and leverage the logarithmic performance guarantees of DHTs so as to ensure scalable event matching performance.

When dealing with string attributes our solution supports prefix, suffix, containment and equality constraints. The main idea of our approach is to store the subscription ids (SubIDs) at those nodes of the DHT network that were selected by appropriately hashing the values of the attributes in the subscriptions. In each node there are three different lists of SubIDs based on the kind of constraint (prefix, suffix, and equality. The containment can be easily transformed to prefix and suffix operation).

For each string attribute of the incoming event we ask all possible nodes for their stored SubIDs whose ids are derived by hashing all possible prefix or suffix sub-strings of the string value. Next, we check its SubID in the collected lists and based on the information we have about the number of attributes each subscription has (from the SubID identifier) we conclude on the matched subscriptions.

During the procedure for processing incoming subscriptions, our performance analysis has shown that in a DHT of N nodes we need to contact $O(\log(N))$ nodes in the worst case for each attribute of the subscription. The event-matching process requires to contact $O(c \times \log(N))$ nodes for each event, where $c$ is a constant which depends on the number of the event’s attributes and on the string-length of each attribute’s value.

We contributed several event-matching algorithms offering trade-offs mainly between the processing cost of collecting and performing the matching at a single node and the overall network traffic to transfer data. Our motivation is to distribute when possible/profitable the matching phase to a number of involved DHT nodes, without incurring high bandwidth overheads.

In conclusion, in this thread of work we have shown how to leverage DHT-based P2P systems, towards building scalable, self-organizing, well-performing systems that support queries with a rich set of constraints on string and numeric attributes. We specifically focused on and presented how our algorithms can be applied to a publish/subscribe environment with a broker network implemented using a DHT. The proposed solution is DHT-independent and can be applied in every DHT infrastructure that provides the basic functionality of finding and reaching the node that stores an object with a specific key value. To our knowledge, this is the first work that shows how string attribute queries (with equality, prefix, suffix, and containment predicates) can be processed over a DHT infrastructure.

4 Result Dissemination

The research conducted in the context of this deliverable has led to one international workshop publication [61] and another 4 papers, all of which are available from the DELIS site, are currently under preparation for submission for publication [47], [46], [40], [4].
References


