DELIS
Dynamically Evolving, Large-scale Information Systems

Integrated Project
Member of the FET Proactive Initiative Complex Systems

Deliverable D2.3.3

New algorithms for a distributed management platform
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 2006
Actual submission date: January 2007
Dissemination level: PU – public

Work Package 2.3: Global management of competing transport services over large net domains
Participants: RadioLabs (RAL), Universita di Roma “Tor Vergata”, Italy
University of Paderborn (UPB), Germany

Authors of deliverable: Giovanni Cortese (g.cortese@computer.org)
Fabrizio Davide (fabrizio.davide@telecomitalia.it)
Emiliano Casalicchio (emiliano.casalicchio@uniroma2.it)
Felix Heine (fh@upb.de)
## Contents

1 Introduction .................................................. 2

2 Algorithms .................................................. 2
   2.1 Data dissemination via publish-subscribe communication .................. 2
   2.2 Distributed subscription clustering in content-based publish/subscribe systems .......... 3
   2.3 Data dissemination and query processing algorithms for semi-structured data in p2p environments ........................................... 4
   2.4 Integration of BabelPeers and XDM ........................................ 4

3 Contributions to Deliverable ..................................... 5
1 Introduction

This report collects new findings and extensions to the work described in Deliverable 2.3.2. In workpackage 2.3 we research topics in distributed data management and dissemination, which can be applied to realize a management infrastructure for large collections of computing and communication resources (including transport networks, GRIDs, content distribution networks, etc). More specifically, the algorithm we present are relevant to:

- content-based publish-subscribe and complex queries
- peer-to-peer storage and query of RDF data
- information dissemination based on gossip

Research on Item 1 and 2 leverages structured P2P networks as overlay architecture, while the third research line assumes unstructured networks as underlying overlay.

Section 2 describes the individual research topics which have been investigated. Full technical reports are provided in the DELIS TR series and referenced from this deliverable.

2 Algorithms

In this section we present a summary of research activities on specific design approaches and algorithms. Full descriptions of the algorithms and designs are provided in the DELIS TR series.

2.1 Data dissemination via publish-subscribe communication

The advent of large-scale distributed applications based on the peer-to-peer communication model, posed to system designers new problems. These problems are strongly connected to the dynamic behaviour of the environment in which such applications are run. Overlay management protocols have been introduced to guarantee peer-to-peer systems connectivity in dynamic large-scale scenarios. Some of these protocols have been specifically designed to avoid the partitioning of the overlay network in large clusters (network breakage) despite massive node failures and the continuous arrivals/departures of nodes (churn). In a previous report [DELIS-TR-0317] we analyzed the ability to maintain connectivity of some Overlay Maintenance Protocols under heavy churn. Now [DELIS-TR-0469], we identify a second effect connected to churn, namely network erosion. We show how erosion affects overlay network connectivity and point out that even a strongly connected overlay network, when exposed to continuous churn, can be disaggregated in a relatively short time. We analyzed this phenomenon, through simulations based on the Peersim [Sit05] simulator. The simulation models were based on popular Overlay Maintenance Protocols such as Cyclon [VGvS05] and ADH [ADH05]. Therefore we propose a connection recovery mechanism to be endowed at each node which is able to collaboratively detect node isolation and the presence of small clusters.

The dynamic behavior shown by these systems, poses problems even at higher applicative layers, that inhibit the usage of many techniques developed for quasi-static distributed systems. In the second part of our work we analyze the behavior of a simple subscription-flooding based algorithm for publish/subscribe in a highly dynamic environment.

The main problem in a publish/subscribe system is how to correctly identify the set of subscribers target for each specific event. From an abstract point of view, this problem is solved building for each event a list containing pointers (e.g. the IP address or other form of identification) to all the target processes. In the following we will refer to this list as the event distribution list (EDL).

We study the problem of building event distribution lists in highly dynamic p2p networks; we consider an unstructured overlay where no dedicated broker exists and each peer (node) can act
both as publisher and subscriber. We show how EDLs built via the simple subscription flooding
approach [CS05] incur in a continuous degradation of accuracy in a dynamic scenario, and, for this
reason, we propose a simple variant of the same approach, where subscriptions are kept updated by
means of a periodic retransmission while pending ones are removed through a simple lease mechanism
[BCM+06]. Through a preliminary study based on simulations, we show how, in the same dynamic
scenario, our variant of the algorithm is able to continuously pursue the optimum in terms of both
accuracy and completeness ([DELIS-TR-0469]).

2.2 Distributed subscription clustering in content-based publish/subscribe systems

In our work we propose a subscription clustering mechanism that allow to group subscription and that
satisfy the main system requirement: expressiveness, efficiency, scalability and adaptability (dynamic
system reconfiguration).

Content-based systems are relatively new and give users the ability to express their interests by
specifying predicates. Publications are matched to submissions on the basis of their content. The
content-based publish/subscribe paradigm is more powerful than topic based and is able to support
rich subscription languages, by introducing a subscription scheme based on the actual content of the
considered events. In other terms, events are not classified according to some pre-defined external
criterion (e.g., topic name), but according to the properties of the events themselves.

The main issues to be considered in the design of content-based publish/subscribe systems are the
following:

- maximize the expressiveness [CRW01], that is the ability of the event notification service to
  provide powerful data model capable to capture information about the events, capable of ex-
  pressing filter and pattern on the notifications of interest.

- Guarantee the scalability with respect to the number of subscriber and the published events.
  The main limitation to the scalability is given by the multicast techniques. The group-based
  multicast techniques, the most suitable in this context, are not readily applicable to resolve the
  scalability problem, as stated in [OAA+00].

- Adapt the network topology and the groups of common interest subscriptions to the dynamics
  of the system thus to maintain the desired level of performances. New subscriptions, changes
  in users preferences, node failures or unsubscriptions must be considered in the management
  of the system.

A trade off between expressiveness and scalability can be obtained providing an algorithm to
properly groups subscription. Our idea is to create a set of multicast groups that better fit the
subscription preferences, that allows fast and scalable event matching. In our solution we try also
to address the dynamic system reconfiguration problem, allowing the use of a scalable and self-
adaptable overlay network. Inspired to the solution proposed in [RLW+03] we decide also to cluster
subscription in groups of similar interests, but the novelty of our approach is that we propose a fully
distributed clustering algorithm that use only partial information on the system state (the event
space and subscription space). This property drastically reduces the information stored on peers,
reduces the cost of information exchange among peer and avoid the cost to maintain a global state.
Our solution differs from those presented in [RLW+03] because the capability to adapt dynamically
the groups of subscriber, as function of the system dynamic.

Our solution is based on the concept of application domain data model [CMCD06], a model for the
event space that have three main characteristics: flexibility, because any application domain can be
represented; expressiveness, because complex preferences can be expressed through predicates over
the set of attributes; representability, because the application domain data model can be represented
by an N-dimensional cartesian space. The use of an N-dimensional cartesian space allow to use clustering algorithm to group subscriptions with similar preferences.

We propose:

- an algorithm to dynamically assign a subscription to the right cluster, disambiguating subscriptions that indicate too wide ranges or subscriptions that contain wildcard for some attributes;
- an algorithm to reconfigure the actual clusters if needed;
- an event matching algorithm based on the concept of distance in the euclidean space.

What we show is that our solution is designed to be independent on the clustering algorithm used, indeed we show the results for both hierarchical and non-hierarchical algorithm.

The main results is that it's possible to tune the parameters of the disambiguation algorithm thus to have a number of false positive and false negative less the the 5% and without duplicated messages.

From our proposal are emerged two problems: (i) the clustering table consistency maintenance; (ii) the evaluation of the consumed bandwidth to run our distributed clustering algorithm with limited knowledge sharing. These two open problems are strictly correlated topic and will be investigate in a future work.

2.3 Data dissemination and query processing algorithms for semi-structured data in p2p environments

In this work, we continue the research line about disseminating, reasoning, and querying semi-structured RDF [MM04] and RDF Schema [BG04] data in p2p environments. The prototype we are developing is based on the global data system model, i.e. individual query answers combine information originating from different peers. This feature is an advantage over systems which route queries from peer to peer, generating individual results only on a single peer. It helps schema integration because mediators can be contributed by any peer and will be used automatically. See [DELIS-TR-0414] for details.

In deliverable 2.3.2 we have sketched the basics of RDF query processing and RDF Schema reasoning. We have now elaborated on these issues, going into details in several issues.

First of all, individual queries might generate huge numbers of results. Thus the approach to exhaustively generate every result immediately hinders scalability. Consider Google would try to generate and deliver every hit for a typical web query having hundreds of millions of results. In analogy to Google, we improved our query engine to render only the top k results for a given query. The basic idea is to fetch intermediate candidate for the query on the fly only as needed for the query evaluation. We combined this approach with a look-ahead and caching mechanism to prevent a multitude of very small messages. Details of this query algorithm can be found in [DELIS-TR-0344] and [DELIS-TR-0419].

Further work includes the very important issue of load balancing the DHT both with respect to storage load and to query processing load. We investigated these problems and presented and compared different strategies. These strategies all base on the idea to build an overlay tree originating from an overloaded node that spans multiple nodes which have little load. These overlay trees are built dynamically based on load statistics collected during the operation of the network. Related to this issue is the question how to maintain the reasoning procedures because the changed dissemination destroys the locality properties that were necessary for our reasoning algorithm. The algorithms and methods can be found in [DELIS-TR-0415] and [DELIS-TR-0416].

2.4 Integration of BabelPeers and XDM

We continued our work on integrating the prototypes XDM and BabelPeers. While XDM bases on XML, BabelPeers is based on RDF. Both systems have their own strengths and focus. XDM
operates on a document level, storing the XML documents in place where they originate. XDM supports sophisticated publish/subscribe mechanisms. BabelPeers distributes the RDF data in small pieces, leading to a global data system model, and includes reasoning features.

The main idea behind the integration is to use the XDM system to store the data as XML files, and to use BabelPeers to store meta-data about this data. This allows to store large XML documents in place, and to use the publish/subscribe mechanisms from XDM, in combination with the querying and reasoning features of BabelPeers.

The integrated prototype uses a single Pastry ring where both XDM and BabelPeers run on. The client first issues an RDF query to learn more about the used schemas and data formats in the network, and to perform reasoning about the meta data. Using this information, an XPath query for XDM is formulated and issued. This allows e.g. to query XML document collections that use different XML Schemas in one integrated query. This work is described in [DELIS-TR-0418].

3 Contributions to Deliverable

The main contribution to section 2.2 is described in [DELIS-TR-0462].

The main contribution to section 2.1 is described in the technical report [DELIS-TR-0469]. This work is fully granted by DELIS project.

Details for section 2.3 can be found in multiple documents. First of all, [DELIS-TR-0414] gives an overview of related work concerning p2p schema management and p2p data integration, and summarizes the basics of our work. The research line about top k query evaluation is documented in [DELIS-TR-0344]. Work about load-balancing and triple dissemination, together with reasoning about RDF Schema, can be found in [DELIS-TR-0415] and [DELIS-TR-0416].

The main contribution to section 2.4 is described in the technical report [DELIS-TR-0418].

Finally, a doctoral dissertation has been finished concerning this topic which was written with support from the DELIS project: [DELIS-TR-0419].

References


