SP^{2}A: Enabling Service-Oriented Grids using a Peer-to-Peer Approach

Michele Amoretti, Monica Reggiani, Francesco Zanichelli, Gianni Conte
Dipartimento di Ingegneria dell'Informazione
Parco Area delle Scienze, 181A, I-43100, Parma, Italy
{amoretti, reggiani, zanichelli, conte}@ce.unipr.it

Abstract

Service-oriented Grids are now aiming at new distributed applications beyond traditional massive computing for the scientific domain. However, mechanisms currently available for the key functions of service publishing and discovering are mostly based on centralized directories, thus raising robustness, scalability and performance concerns.

In this paper we first discuss the main issues regarding the distribution and search of service descriptions through a totally decentralized Peer-to-Peer approach. Then we recall the main features of our Service-oriented Peer-to-Peer Architecture (SP^{2}A), a flexible solution for an advertisement-based publication and discovery of Grid services. The current SP^{2}A prototype, built upon JXTA and OGSA frameworks, is presented along with functional and performance results.

1. Introduction

Service-Oriented Architectures (SOA) are rapidly becoming the key approach for achieving new levels of interoperability and scalability in the development of distributed applications. Grid computing is also evolving in this direction, with the recent adoption of the Web Service Resource Framework (WSRF) [11], which builds upon the experience gained with the previous Open Grid Service Infrastructure (OGSI) specification [21].

Within SOA approaches, current solutions for advertising service providers and for allowing prospective clients to discover them are nowadays mostly based on industry-backed centralized registries such as the Universal Description, Discovery, and Integration (UDDI) project [6], established in 2000 by Ariba, IBM, and Microsoft, with the intention of defining and providing master directories of publicly available e-commerce services. Clearly, such directories are endowed with rich features, and the technical and commercial weight of their maintainers is outstanding, yet their centralized nature exhibits obvious shortcomings in terms of fault-tolerance and performance, thus raising scalability concerns.

In order to tackle this problem, there has been recently considerable work on using Peer-to-Peer (P2P) networks to allow for decentralized publishing and discovery of Web services [7, 17]. In the related field of Grid Services, traditionally oriented to the Client/Server (C/S) approach, it appears there have been less results beyond the introductory work in [8, 13]. Unlike a centralized registry which will contain service descriptions of service providers which have been approved and thus endorsed by the registry maintainers (trusted themselves because of their commercial power), a P2P-based decentralized management of Grid service advertisements calls for new tradeoffs between the liberal but risky possibility for providers to expose self-promoted services and the strong albeit not so flexible security framework available in Grid systems. Once endowed with proper security mechanisms, in our view the P2P approach can become a viable solution for the support of Virtual Organizations, providing significant benefits over the C/S approach.

The result of our efforts in the theory and practice of Grid and P2P architectures is the *Service-oriented Peer-to-Peer Architecture (SP^{2}A)* [5]. The approach followed in SP^{2}A allows to cope with the requirements of applications with a large number of users dynamically connecting to the system, and provides high levels of scalability, decentralization and interoperability. This paper presents the P2P-based capabilities of SP^{2}A for service sharing and discovery as well as some performance results we have obtained with the current prototype.

The paper is organized as follows. Section 2 introduces SP^{2}A layered architectural model and the technical background for its capabilities, namely service description and discovery. The prototype under development is presented in Section 3 where network establishment, service sharing and discovery are described. Section 4 illustrates some experimental results we achieved with SP^{2}A on a mid-sized setup across our campus network. Section 5 discusses related work on service-oriented architectures. Finally, an outline of SP^2A open issues concludes the paper.

2. Overview of SP^2A

SP^2A [5] is based on the current Grid Computing and P2P state-of-the-art specifications. We are exploiting the Open Grid Service Architecture (OGSA) [9], whose key concept is the Grid Service, essentially a stateful (and potentially transient) Web Service with an associated lifetime and whose behavior must conform to a set of interfaces defining the service semantics. Moreover, Grid Services should be compliant with standard mechanisms to support state management, referenceable handles and event notification. The SP^2A P2P infrastructure is based on the JXTA project, a Sun Microsystems open P2P initiative [3]. JXTA project standardizes a common set of protocols which defines the minimum required network semantics allowing peers to form and join a virtual network.

The SP^2A is designed as a network of peer service host environments. Each of them can expose local services, publishing their descriptions remotely, and search for remote services. The SP^2A architectural model embodies three layers of services, namely *System-level Services*, *Base Services* and *Application Services*. System-level Services are embedded in each peer and transparently perform infrastructural tasks (*e.g.* service interaction, security). Every peer must be configured with the Base Services (e.g. peer management, peer information retrieval, local service management, remote service management). Moreover, a peer may be specialized in providing particular Application Services (e.g. bandwidth brokering, certification authority, streaming, instant messaging, file sharing, screen sharing), and eventually it can be member of one or more subgroups of the SP^2A network.

Application Services are shared in the SP^2A network. Basically, each of them is a Web Service with a WSDL interface which defines the allowed operations and expresses the binding information (e.g. how to interact with the service). On the other side, the WSDL document is unable to express the semantics of the service, thus preventing important features such as automatic service composition, discovery and invocation as well as execution monitoring.

For this reason, we are investigating the use of OWL-S [16] descriptions to contextualize the service in an ontology shared among SP^2A peers. We plan to derive a single XML document named Grid Service Advertisement (GSA) from the OWL-S description and the WSDL interface of a service. The GSA extends the basic JXTA advertisement and could be obtained by (a possibly partial) mapping of the information contained in the OWL-S (ServiceProfile and ServiceModel) and the WSDL descriptors into a set of (key, value) pairs, e.g. (Name, "StreamingService").

Besides information about the service functionalities, the GSA also should describe how to interact with the service, and provide some informations about the service's status. This is OWL-S's concept of grounding, which is generally consistent with WSDL's concept of binding.

Such a complete description would allow the peer to elaborate high-level queries, relying on the shared ontology. Complex queries can be translated to sequences of atomic queries for (key,value) pairs, corresponding to the GSA entries, which are directly managed by JXTA discovery protocols. The low-level P2P search process, along with the publication process, is illustrated in next section. The architecture cannot guarantee that all requests will find a suitable service, even if available in the system, due to the lack of assurance that every atomic query reaches the proper set of peers. If a suitable service is found, its WSDL can be retrieved and the grounding can be performed.

3. SP^2A Prototype Implementation

The diagram in figure 1 illustrates the basic activities performed by the SP^2A peer prototype. The Peer object joins the SP^2A peergroup invoking the joinGroup() method on a singleton instance of the PeerManagementService, a base service class. This invocation creates two JXTA objects: the RendezVousService, managing rendezvous properties of the peer, and the DiscoveryService, handling the search and publication of advertisements. The LocalApplicationManagementService interacts with the ContainerRegistryService, an OGSA object exposing the services currently available in the local container.

To share the GSAs, the peer invokes the shareLocalService method on RemoteApplicationManagementService that publishes the advertisements on the DiscoveryService previously created.

RemoteApplicationManagementService provides also the findRemoteService() method, searching suitable remote services through the JXTA network.

3.1. Service Sharing

Current SP^2A prototype allows two kinds of peers:

- *Edge peers*, peers which are able to send query/reply messages but not to propagate queries.
- *Rendezvous super-peers*, peers that have agreed to cache advertisement indices, i.e. pointers to edge peers that cache the corresponding advertisement, thus they are able to propagate queries.

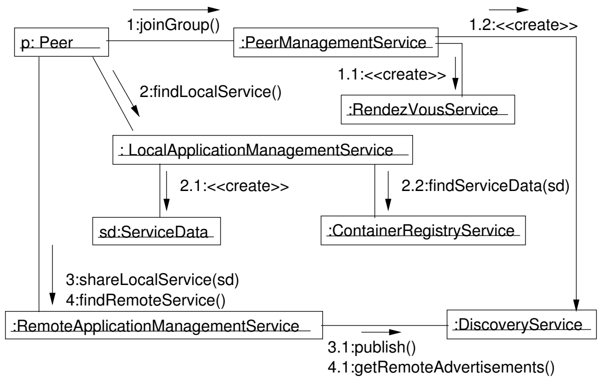

Figure 1. Collaboration diagram of the Peer-Manager subsystem.

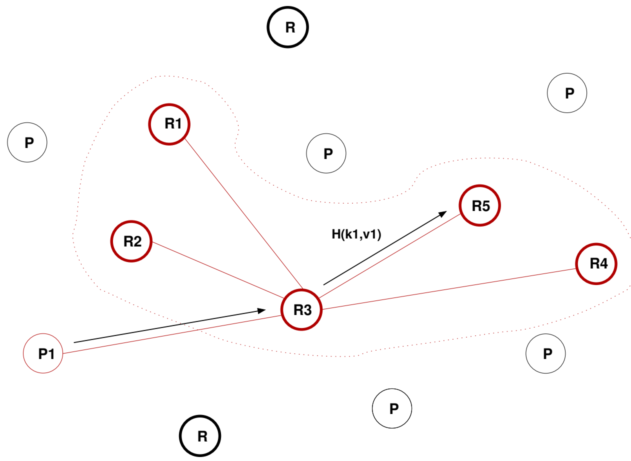

Figure 2. Example of GSA publication and (key, value) replication. The dashed line indicates R3's RPV.

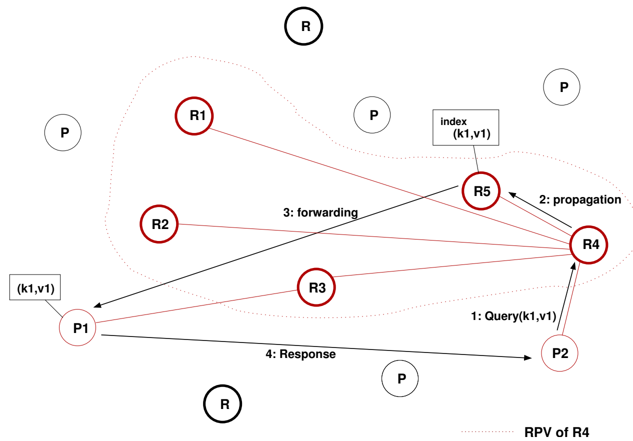

Figure 3. Example of atomic query. The dashed line indicates R4's RPV.

Each edge peer is connected to one rendezvous super-peer. Each rendezvous has a Rendezvous Peer View (RPV), which is an ordered list of other known rendezvous.

To publish a service, the peer creates a Shared Resource Distributed Index (SRDI) message which is composed by a set of entries corresponding to the (key, value) pairs of the GSA. If the peer is an edge, it sends the SRDI message to its rendezvous (propagation with TTL=1). Each rendezvous stores SRDI message entries in a dynamic index including also, for each entry, the ID of the peer which originated the SRDI message and an expiration time. If the RPV is > 3, each index entry is replicated to a known rendezvous, chosen by hashing the entry itself, with the SHA1 address space divided into the size of the peerview. To reduce message exchange, all the entries addressed to the same rendezvous are sorted into replica "bins" and transformed into a single SRDI message which is pushed to the addressee. The rendezvous indexes constitute a Distributed Hash Table (DHT), where service description are stored.

For example figure 2 describes the situation in which edge peer P1 publishes a GSA to its rendezvous super-peer R3. The RPV of R3 contains R1, R2, R4 and R5. Suppose that the hash function applied to the GSA entry (k_1,v_1) returns R5. Thus R3 pushes the index entry to R5.

3.2. Service Discovery

Figure 3 shows an example of the discovery of a shared service, describing how an atomic query is managed. The query for (k_1, v_1) is issued by edge peer P2 to its rendezvous R4, which looks for one or more suitable entries in its index. If local search fails, R4 applies the hash function to (k_1,v_1) which returns R5 because R4 has the same RPV of R3. Thus R4 propagates the request to R5, which has an index entry for (k_1,v_1) pointing to P1. Finally, R5 forwards the query to P1 which then responds to P2.

In the previous example, we assumed that the RPV is the same on R4 and R3. This is usually not true for every rendezvous, even if JXTA implements a distributed algorithm aiming at the convergence of all RPVs to a common RPV. Suppose that R1 and R3 leave the network, P1 connects to R2, and R4's RPV changes (two new rendezvous join). Suppose that when R4 applies the hash function to (k_1,v_1) it finds R6, which has no index entries for (k_1,v_1). In this case, an alternative mechanism is used to continue searching: a limited-range walker explores the RPV from the initial DHT rendezvous target [20].

This solution differs from the Document Routing Model approaches of Chord [19] or Pastry [18], which are based on a relatively stable peer infrastructure where a distributed consistent view can be maintained with minimum overhead. JXTA assumptions are closer to the limited-range walker approach for unstructured networks proposed in [14]. It is important to point out that the loosely-coupled unstructured policy proposed by JXTA is just the default policy. In our first SP^{2}A prototype we adopted this solution, but in the future we plan to study new policies.

4. Experimental Results

The prototype presented in the previous section has been evaluated in a network of SP^{2}A peers.

The peer code[1] is built upon Globus Toolkit 3 (GT3) [1], an open source implementation of the Open Grid Services Architecture (OGSA), and JXTA Java Binding [3], a portable implementation of JXTA protocols supporting connectivity and service sharing and discovery. To verify the correctness of the implementation, we set up a set of SP^{2}A peers and distributed them on several 100Mbps university LANs. The host machines are quite heterogenous, ranging from Intel Pentium III, Pentium IV and Xeon to AMD Athlon, with 512MB RAM or more, and running either Windows XP or Linux (kernel 2.4.x to 2.6.x).

The functional evaluation demonstrated the effectiveness of JXTA in the implementation of a distributed discovery service. Each edge peer is able to connect to a rendezvous peer and perform a search and invocation of required services. Moreover, the edge peers are able to autonomously change their rendezvous peer when the current one becomes overloaded or unreacheable.

To evaluate prototype performance, we distributed N SP^{2}A peers, with

$$N = R + E$$

where R is the number of rendezvous super-peers and E is the total number of edge peers. Each peer knows the R rendezvous and chooses its own randomly. M services are randomly published in the R rendezvous, before the E edge peers start their activity, *i.e.* discovery of services in the NetPeerGroup. Each edge peer searches for MS_i services. The execution of the i-th peer ($i = 1, .., E$) returns, for the request of j-th service, the $responseTime[i, j]$, the time between the query and the first response, and MD_i, the number of services which have been found at least once ($MD_i \leq MS_i$). In our experiments, we have evaluated the total number of query hits on query ratio:

$$QH/Q\% = \frac{\sum_{i=1}^{E} MD_i}{\sum_{i=1}^{E} MS_i} \times 100$$

1 current version is 0.0.4, downloadable from the project's homepage [5]

and the average response time per service:

$$\mu_{RT} = \frac{\sum_{i=1}^{E} \sum_{j=1}^{MD_i} responseTime[i, j]}{QH}$$

and its standard deviation.

We performed a series of experiments modifying the value of E and R, while keeping the other parameters at a fixed value ($M = 10$, $MS_i = 1$). The edge peers execute the Q queries at the same time and then wait until the required service is found.

Figure 4 illustrates the $QH/Q\%$ computed 5 seconds after the queries were sent. The best performance is achieved

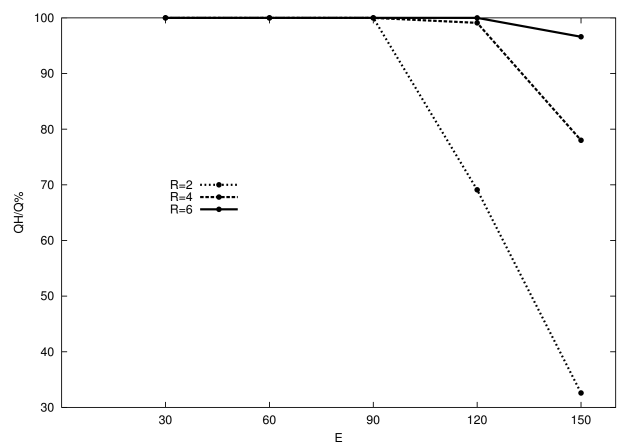

Figure 4. $QH/Q\%$ versus E in the time interval $\Delta T = 5s$.

with 6 rendezvous peers, when more than 95% of queries are replied within 5 seconds even when the largest number of edge peers are involved.

The distributions of the replies over time, received by 150 edge peers, are illustrated in figure 5. The worst performance are observable when $R = 2$, more than 50% of the replies are received after $10s$. Increasing the number of rendezvous peers sensibly improves the performance with the whole set of query answered within 10 seconds when $R = 6$.

Figure 6 reports the average and standard deviation of response time per service. When the number of edges is low ($E \leq 90$), performance is not influenced by the number of rendezvous peers. For large number of edges, μ_{RT} increases rapidly and only the configuration with $R = 6$ shows a good reactiveness even in presence of 150 edge peers performing queries at the same time.

The results achieved for these small to medium networks proved the effectiveness of JXTA for decentralizing publishing and discovery of services. Nevertheless, the number

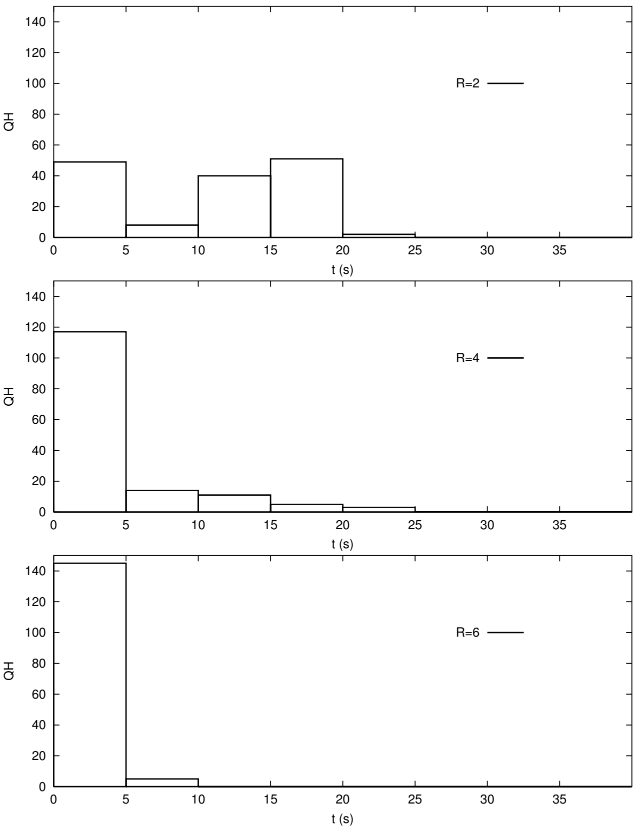

Figure 5. QH temporal distribution with $E = 150$ and different values of R (from the top: $R = 2, 4, 6$).

of rendezvous peer should be carefully defined to avoid a performance loss when the number of edge peers increases.

To evaluate the correctness of the search policy available in SP^{2}A for larger networks, including more than 150 edge peers and 6 rendezvous super-peers, we are currently implementing a ns2-based [2] JXTA simulator. The final goal is to to compare the performance of competing discovering protocols on simulated networks of several thousand nodes.

5. Related Work

The convergence of Grid and Peer-to-Peer environments is clearly depicted in [13], envisioning systems with a larger number of stable nodes than in today's P2P networks and resources with highly diverse types and characteristics. Although, the authors focus on resource sharing, rather than service sharing. The evaluated request propagation strategies are enhancements of the resource intensive Flooded Requests Model (FRM) while no Document Routing Model-based algorithms are evaluated.

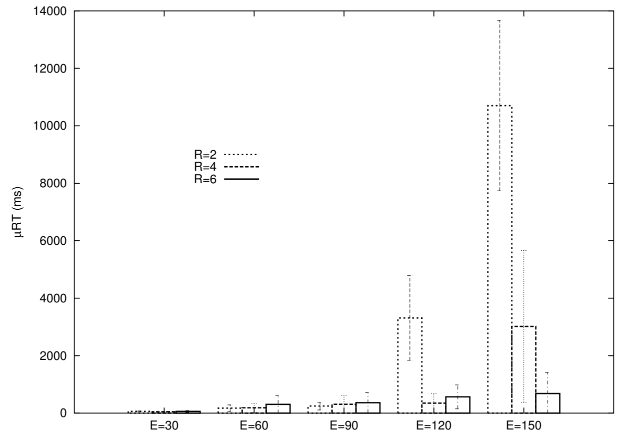

Figure 6. Average response time per service μ_{RT} versus E, for different values of R (from the top: $R = 2, 4, 6$).

The objective of the project illustrated in [12] is to provide a model for Web Service discovery and invocation in the JXTA P2P framework. The idea is that, given a reachable Web Service, it should be possible to invoke it from inside a JXTA Group. More precisely, the WSDL document is used to dinamically generate a client proxy which will invoke the remote Web Service using SOAP to communicate with it. The user's client does not need to understand SOAP, because it interacts with a JXTA application which encapsulates the proxy. Our approach is different, in that we separate the P2P issues (network establishment, service sharing, service discovery) from the service interaction aspects. This design choice implies also technological decoupling in the implementation phase, allowing radical changes such as replacement of the JXTA Java binding or GT3 with equivalent (or maybe more efficient) solutions.

The ICENI Grid Middleware [10] aims to build an implementation independent service-oriented architecture, although the proposed Jini, JXTA e OGSI implementations are alternative and do not attempt at obtaining a close P2P/Grid integration.

An interesting metadata-based P2P infrastructure for educational purposes is provided by the Edutella Project [15]. Edutella Peers use JXTA P2P primitives to form the Edutella Network, in which they can share and retrieve services. The Edutella Query Exchange Language and the Edutella common data model provide the syntax and semantics for an overall standard query interface across the heterogenous peer repositories for any kind of RDF [4] metadata. Edutella wrappers are used to translate queries and results from the Edutella common format to the local format of the peer (*e.g.* RQL, TRIPLE, SQL, dbXML, AmosQL) and vice versa.

6. Conclusions and future work

In this paper we have described the current status of a the SP^2A project, a service-oriented architecture which takes advantage of the strengths of two recent technologies at different levels. SP^2A relies on a P2P network as a decentralized carrier of service advertisements and on Grid service query and invocation mechanisms which exploit robust and secure OGSA technologies. The experimental evaluation demonstrated the reliability and good performance of the SP^2A prototype implementation built upon JXTA and Globus frameworks.

Several open issues will be investigated in our future research. First of all, we are analyzing the *WS-Resource Framework (WSRF)* [11] which is a straightforward refactoring of the concepts and interfaces developed in the OGSI [21] specification in a manner that exploits recent developments in Web Services architecture (*e.g.*, WS-Addressing). We plan to adopt the forthcoming 4th release of the Globus Toolkit, supporting WSRF.

Another primary issue we will be concerned with is the introduction of security policies and mechanisms in the SP^2A system, aiming to integrate the somewhat contrasting security models of Grid and JXTA.

Finally, we plan to improve the prototype implementation by introducing a mapping from OWL-S descriptions of the available services to JXTA (key,value)-based query facilities. This extension would allow peers to perform semantically-enriched searches for published services, *i.e.* on the basis of ontologies defining relationships between service-related concepts.

7. Acknowledgments

This work has been partially supported by a IBM Faculty Award and by the ”WEB-MINDS” FIRB project of the National Research Ministry.

References

[1] The Glubus Alliance. http://www.globus.org.

[2] The network simulator (ns2). http://www.isi.edu/nsnam/ns/.

[3] Project JXTA. http://www.jxta.org.

[4] Resource Description Framework (RDF). http://www.w3.org/RDF/.

[5] Service-oriented Peer-to-Peer Architecture (SP^2A) homepage. http://dsg.ce.unipr.it/research/SP2A.

[6] Universal Description Discovery and Integration (UDDI). http://www.uddi.org, 204.

[7] F. Banaei-Kashani, C.-C. Chen, and C. Shahabi. WSPDS Web Services Peer-to-peer Discovery Service. In *The 2004 International Symposium on Web Services and Applications*, June 2004.

[8] I. Foster and A. Iamnitchi. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In *2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, February 2003.

[9] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. *Open Grid Service Infrastructure WG, Global Grid Forum*, June 2002.

[10] N. Furmento, J. Hau, W. Lee, S. Newhouse, and J. Darlington. Implementation of a Service-Oriented Architecture on Top of Jini, JXTA and OGSI. In *UK e-Science All Hands Meeting Conference*, September 2003.

[11] Globus Alliance, IBM, and Hewlett Packard. WS-Resource Framework. http://www.globus.org/wsrf/.

[12] S. H. Hajamohidden. A Model for Web Service Discovery and Invocation in JXTA. Master's thesis, Technical University Hamburg-Harburg, 2003.

[13] A. Imanitchi and I. Foster. A Peer-to-Peer Approach to Resource Location in Grid Environments. In *Grid Resource Management*. Kluwer Publishing, 2003.

[14] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and Replication in Unstructured Peer-to-Peer Networks. In *ACM Sigmetrics '02*, June 2002.

[15] W. Nejdi, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. Edutella: a P2P Networking Infrastructure Based on RDF. In *WWW2002*, May 2002.

[16] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara. Semantic Matching of Web Services Capabilities. In *Proceedings of the First International Semantic Web Conference on The Semantic Web*, pages 333–347, 2002.

[17] M. Paolucci, K. Sycara, T. Nishimura, and N. Srinivasan. Using DAML-S for P2P Discovery. In *First International Conference on Web Services (ICWS'03)*, June 2003.

[18] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, November 2001.

[19] I. Stoica, R. Morris, D. L. Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. *IEEE/ACM Transactions on Networking*, 11(1), 2003.

[20] B. Traversat, M. Abdelaziz, and E. Pouyoul. A Loosely-Consistent DHT Rendezvous Walker. *Project JXTA*, March 2003.

[21] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, P. Vanderbilt, and D. Snelling. *Open Grid Services Infrastructure (OGSI) Version 1.0*, June 2003.