A P2P Approach for Membership Management and Resource Discovery in Grids

Carlo Mastroianni[1], Domenico Talia[2] and Oreste Verta[2]
[1]ICAR-CNR, Via P. Bucci 41 c, 87036 Rende, Italy
mastroianni@icar.cnr.it
[2]DEIS, University of Calabria, Via P. Bucci 41 c, 87036 Rende, Italy
{talia, verta}@deis.unical.it

Abstract

As deployed Grids increase from tens to thousands of nodes, peer-to-peer (P2P) techniques and protocols can be used to implement scalable services and application in Grids. This paper proposes a P2P approach for handling two key services in Grid environments: membership management and resource discovery. The membership management service exploits the use of "contact nodes", i.e. of nodes elected within Virtual Organizations to facilitate the assignment of neighbours and the construction of interconnections between Grid nodes. The resource discovery service uses these interconnections to give Grid nodes the opportunity to explore the Grid and discover available resources. The paper analyzes the performance of the resource discovery service on Grid networks with different sizes.

1. Introduction

As Grids used for complex applications increase from tens to thousands of nodes, we should decentralize their functionalities to avoid bottlenecks. The P2P model could help to ensure Grid scalability: designers can use P2P philosophy and techniques to implement non-hierarchical decentralized Grid systems. The adoption of the service oriented model in novel Grid systems (for example the Open Grid Services Architecture (OGSA [1]), or the Web Services Resource Framework (WSRF) [4]) will favor the convergence between Grid and P2P models, since Web Services can be used to implement P2P interactions between hosts belonging to different Grid organizations.

P2P techniques can be particularly useful to manage two key services in Grid information systems: *membership management* (or simply *membership*) and *resource discovery*. The objective of a membership management service is twofold: adding a new node to the network, and assigning this node a set of neighbour nodes. The resource discovery service is invoked by a node when it needs to discover hardware or software resources having specified properties.

In currently deployed Grid systems, resources are often owned by research centres, public institutions, or large enterprises: in such organizations hosts and resources are generally stable. Hence, membership management and resource discovery services are efficiently handled through centralized or hierarchical approaches, as in the OGSA and WSRF frameworks. As opposed to Grids, in P2P systems nodes and resources provided to the community are generally dynamic: peers can be frequently switched off or disconnected. In such an environment a distributed approach is more effective and fault-tolerant than a centralized or hierarchical one.

This paper proposes and discusses a P2P approach for the design of membership and resource discovery services in a Grid environment. The paper describes the protocols exploited by such services, and analyzes the performance of the resource discovery protocol on Grid networks with different sizes.

The remainder of the paper is organized as follows. Section 2 discusses related work. The membership and resource discovery protocols are described in Section 3. Section 4 analyzes the performance of the discovery protocol by means of an event-driven simulation framework. The influence of network and protocol parameters is evaluated. Section 5 concludes the paper.

This work was partially supported by the Italian MIUR FIRB Grid.it project RBNE01KNFP on High Performance Grid Platforms and Tools and by the project KMS-Plus funded by the Italian Ministry of Productive Activities.

2. Related Work

P2P membership and discovery services can be classified as using unstructured or structured approaches to search resources. Gnutella [5] and Freenet [2] are examples of unstructured P2P networks: hosts and resources are made available on the network without a global overlay planning. Structured P2P networks, such as CAN [9] and Chord [10], use highly structured overlays and exploit a Distributed Hash Table (DHT) to route queries over the network. A DHT is a data structure for distributed storing of pairs (key, data) which allows for fast locating of data when a key is given.

Membership and resource discovery services are also key issues in Grid systems. A centralized or hierarchical approach is usually adopted. For example, the information model exploited in the Globus Toolkit 3 (GT3), the version of Globus built upon OGSA, is based on Index Services [3], a specialized type of Grid Services. Index Services are used to aggregate and index Service Data, i.e. metadata associated to the resources provided by Grid hosts. There is typically one Index Service per Virtual Organization (VO) but, in large organizations, several Index Services can be organized in a hierarchy. A similar approach is used in the WSRF-based Globus Toolkit 4: ServiceGroup services are used to form a wide variety of collections of WS-Resources, a WS-Resource being a Web service associated with a stateful resource.

Today, the Grid community agrees that in large scale Grids it is not feasible to adopt a centralized or hierarchical approach for providing a scalable resource discovery service. In [12] the main features of Grid and P2P systems are compared, and it is argued that these two worlds will converge in terms of their concern, as Grids scale and P2P systems address more sophisticated application requirements.

Super-peer networks have been proposed to achieve a balance between the inherent efficiency of centralized search, and the autonomy, load balancing and fault-tolerant features offered by distributed search. In fact, Grids can be naturally modelled as super-peer networks, in which a super-peer has the capacity of managing metadata associated to the resources provided by the hosts of a VO, and super-peers connect to each other to form a peer network at a higher level. In [13], performance of super-peer networks is evaluated, and rules of thumb are given for an efficient design of such networks. In [8] a general mechanism for the construction and the maintenance of a superpeer network is proposed and evaluated. In this work, a gossip paradigm is used to exchange information among peers and dynamically decide how many and which peers can efficiently act as superpeers.

3. Membership Management and Resource Discovery Protocols

As mentioned in Section 1, the peer-to-peer model can be advantageously exploited in Grid systems for the deployment of information services, and in particular of discovery services. To maximize the efficiency of such a model in Grids, it is useful to compare the characteristics of Grids and P2P systems.

(i) Grid systems are less dynamic than P2P networks, since Grid nodes and resources often belong to large enterprises or public institutions and security reasons generally require that Grid nodes authenticate each other before accessing respective resources.

(ii) Whereas in a P2P network users usually search for well defined resources (e.g. audio/video files), in Grid systems they often need to discover software or hardware resources that match an extensible set of resource descriptions. Accordingly, while structured protocols, e.g. based on distributed indices, are usually very efficient in P2P file sharing networks, unstructured or hybrid protocols seem to be preferable in largely heterogeneous Grids. Another consequence is that the performance of a discovery service is influenced by the distribution of classes of resources, a class of resource being a set of resources that satisfy some given constraints on resource properties, as discussed in Section 4.

(iii) In a Grid, it is feasible to individuate, for each VO, a subset of powerful nodes having high availability properties.

We designed our peer-to-peer membership and discovery protocols basing on the above mentioned considerations. In the following of this section we describe such protocols and show how they can be exploited in the OGSA-based Globus Toolkit 3.

The membership protocol exploits the characteristics of *contact nodes*. A contact node is a Grid node that plays the role of an intermediary node during the building process of the Grid network. One or more contact nodes are made available by each Virtual Organization. Whenever a Grid node wants to connect to the network, it contacts a subset of contact nodes and registers at those nodes. In turn the selected contact nodes randomly choose a number of previously registered Grid nodes and communicate their addresses to the requesting node: these nodes will constitute the neighbour set of the new Grid node.

A Grid node communicates with the contact nodes either periodically or whenever it detects the disconnection of a neighbour node, in order to ask for its substitution. Figure 1 shows a graphical description of the membership management protocol. A number of contact nodes are depicted, and for each of them the corresponding set of registered nodes is reported. In Figure 1(a) node `N` wants to connect to the network, and selects two contact nodes. In Figure 1(b), the selected contact nodes add node `N` to the list of registered nodes and respond to it by communicating the addresses of a number of neighbour nodes, which will constitute the neighbour set of node `N`.

The membership management protocol requires a proper setting of two parameters: (i) the `contact parameter K`, i.e. the number of contact nodes at which a new node registers (`K`=2 in Figure 1); (ii) the `number of neighbours V` (`V`=4 in Figure 1).

The resource discovery protocol is based on a unstructured approach: whenever a user initiates a search procedure, the corresponding node sends a query message to its neighbours, which in turn forward it to their own neighbours. If a node possesses the requested resource, it sends a queryHit message that will follow the same path back to the requesting node. Note that, as assumed above, contact nodes are not involved in the resource discovery protocol.

A number of techniques are adopted to decrease the network load. (i) The number of hops is limited by a Time-To-Live (`TTL`) parameter. (ii) Each query message contains a field used to annotate the nodes that the query traverses along its path. A peer does not forward a query to a neighbour peer that has already received it. (iii) Each peer maintains a cache memory where it annotates the IDs of the last received query messages. A peer discards the queries that it has already received. Techniques (ii) and (iii) are used to avoid the formation of cycles in the query path, and are both useful, since technique (ii) can *prevent* cycles only in particular cases (i.e. when a query, forwarded by a peer, is subsequently delivered to the same peer), whereas technique (iii) can *remove* cycles in all the other cases (e.g., when *two* copies of a query, sent by a peer `A` to two distinct neighbour peers `B` and `C`, are subsequently both delivered to the remote peer `D`).

The membership and resource discovery protocols we propose can provide a significant performance enhancement with respect to classical unstructured P2P systems, in which such protocols adopt very similar policies, based on the forwarding of request/response messages (referred to, respectively, as “ping/pong” messages and “query/queryHit” messages). In particular, the use of contact nodes is advantageous for the following reasons:

- the processing load carried by Grid nodes is lowered, since they only have to process query messages, but do not receive node discovery requests.
- the network load is decreased since node discovery messages do not travel around the network.

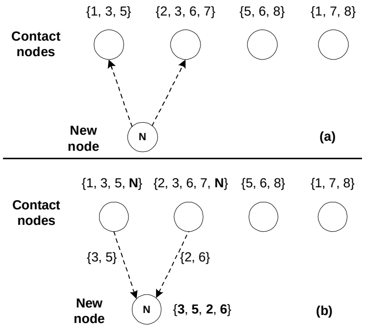


Figure 1. The membership management protocol: (a) a new node registers at a set of contact nodes and (b) receives the identities of its neighbours.

Figure 2 shows how our P2P approach can be exploited in the Globus Toolkit 3. In a Grid VO, a GT3 Index Service subscribes to the Service Data stored in the Grid Services hosted on the nodes of that VO. GT3 aggregators collect Service Data, which typically contain metadata information about Grid Services, and send it to the Index Service. The Peer Service is a static Grid Service that processes query messages coming from remote VOs; when receiving a query, it asks the Index Service to find resources matching the query constraint. A Peer Service forwards query and queryHit messages through the Network module. This architecture extends the one presented in [11], and implicitly exploits the super-peer model; indeed a Peer Service uses the Index Service to manage information stored in the Grid nodes of a VO, whereas Peer Services provided by different VOs communicate to each other to form a P2P network at a higher level.

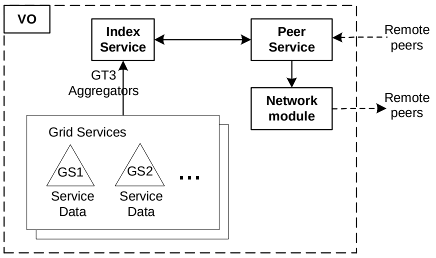


Figure 2. Implementation of the peer-to-peer model using the GT3 framework.

4. Simulation Analysis

We analyzed the performance of the P2P discovery protocol described in Section 3, in order to assess its effectiveness in a Grid environment and estimate the influence of network and protocol parameters on performance measures. An event-based object-oriented simulator was used for modelling the construction of the P2P peer network, driven by the membership protocol, and for simulating the behaviour of the resource discovery protocol in Grid networks.

In Section 4.1 we define a set of resource distribution patterns that are used and compared in our simulations. Parameters and performance indices are defined in Section 4.2. Finally, in Section 4.3 we discuss the performance of the resource discovery protocol obtained for different network sizes.

4.1 Resource Distribution

The performance of a discovery protocol depends on the distribution of resources among the hosts of the network. As mentioned in Section 3, in Grid systems users often need to discover resources that belong to classes of resources, rather than well defined resources. For example, when building a distributed data mining application [7], a user may need to discover a software that performs a clustering task on a given type of source data. A query, containing the appropriate constraints, is generated to find such resources on the Grid; at a later time, one of the discovered resources will be chosen by the Grid scheduler for execution. Therefore the performance of a resource discovery protocol in a Grid is related to the classification of heterogeneous resources in a given application domain.

We assumed, as in [6], that the average number of elementary resources offered by a single node remains constant as the network size increases. In the simulation, this average value was set to 5, and a gamma stochastic function was used to determine the number of resources owned by each node. However, as the network size increases, it becomes more and more unlikely that a new node connecting to the network provides resources belonging to a new resource class. Therefore, we assumed that the overall number of distinct resource classes offered by a network does not increase linearly with the network size.

To take into account the impact of the distribution of resource classes on performance results, we analyzed three kinds of distributions:

- a logarithmic distribution (distribution *A*): the number of resource classes offered by a Grid network with `N` nodes (with `N` comprised between 10 and 10000) is equal to `5*(log`$_2$`N)^2`.
- a square root distribution (distribution *B*): the number of resource classes is `40+5*sqrt(N)`;
- a constant distribution (distribution *C*): the number of resource classes is constant and equal to 100.

As an example, if distribution *A* is assumed, a Grid having 1024 nodes provides 5120 resources belonging to 500 different classes. The three distribution patterns are plotted in Figure 3. Note that in distributions *A* and *B*, the respective numbers of resource classes are comparable when `N=10`, but increase with different trends for larger networks. Distribution *C* is a non realistic one, since any new node joining the network can only provide resources belonging to a predetermined set of resource classes: however it can be used for comparison purposes.

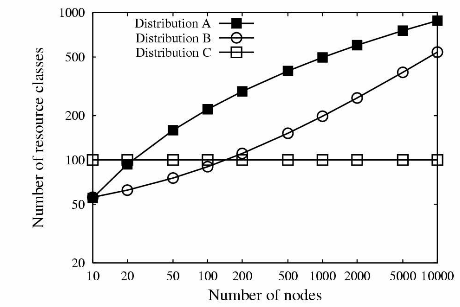


Figure 3. Distributions of resource classes: overall number of resource classes w.r.t. the network size.

4.2 Simulation Parameters and Performance Indices

Table 1 reports the simulation parameters used in our analysis. During a simulation run, a node randomly selects, with a frequency determined by the mean query generation time `MQGT`, a resource class, and forwards a query for resources belonging to that class.

In Table 2, the performance indices calculated at the end of each simulation run are defined. The `Nres` index is deemed to be more important than the probability of success `Psucc`, since the *satisfaction of the query* depends on the number of results (i.e. the number of discovered resources) returned to the user that issued the query: a resource discovery operation could be considered *satisfactory* only if the number of results exceeds a given threshold. The message load `L` should be kept as low as possible. This performance index often counterbalances the success indices, in the sense that high success probabilities sometimes are only achievable at the cost of having high elaboration loads. A trade-off can be obtained by evaluating the ratio R, which is an index of efficiency: if we succeed in increasing the value of R, it means that a higher relative number of queryHit messages, containing useful results, is generated or forwarded with respect to the overall number of messages.

Table 1. Simulation parameters

Parameter	Value
Network size (number of nodes) **N**	10 to 10000
Overall number of resources offered by the network vs. the network size	5 **N**
Overall number of resource classes offered by the network vs. the network size	Obtained with distributions *A*, *B*, *C*
V: number of neighbours of a node	2 to 6
Time to live **TTL**	3 to 7
Mean query generation time **MQGT**	300 sec

Table 2. Performance indices

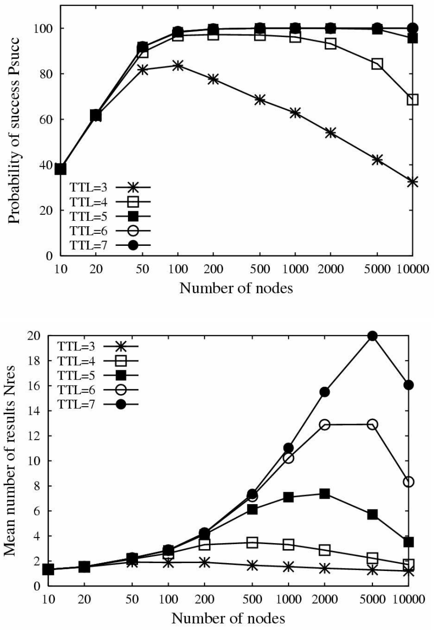
Performance index	Definition
Probability of success $\mathbf{P_{succ}}$	Probability that a query issued by a generic node will succeed. i.e. will be followed by at least one queryHit.
Mean number of results **Nres**	Mean number of resources that a node discovers after its query.
Message load **L**	Frequency of all messages (queries, queryHits) received by a node.
QueryHits/messages ratio **R**	Number of queryHits received by a node divided by the overall number of messages received by that node.

4.3 Performance Results with Variable Network Size

In this section we examine how performance results vary with respect to the network size; we considered networks with numbers of nodes ranging from 10 to 10000. This permitted to investigate the scalability of the resource discovery protocol. Furthermore, we tested different values of TTL, in order to know how that parameter can be tuned to improve performance for a given network size.

A first set of results was obtained by adopting the distribution pattern *A*. We set the number of neighbours V to 4, whereas the TTL value was varied from 3 to 7. Results are reported in Figure 4. For small networks, performance is poor because a few available nodes are not able to offer the entire set of resource classes (e.g. with N=10, there are more than 50 classes of resources and about elementary 50 resources shared by the whole network: for simple statistical considerations, it is very unlikely that all the resource classes can be actually provided by the network). When the number of nodes increases, success probabilities increase as well.

An increase of the TTL value causes a benefit (in terms of probability of success and mean number of results) that is negligible for small networks, but more and more significant for larger networks. Note that, with a fixed value of TTL, the number of results decreases when the network size exceeds a threshold value: beyond the threshold that TTL value does not permit a full exploration of the network. In particular, in a network with more than 5000 nodes, a TTL value higher than 7 is needed to obtain higher performance.



(a)

(b)

Figure 4. Probability of success (a) and mean number of results (b) w.r.t. the network size, for different values of TTL, with resource distribution *A*.

From Figure 5(a) we see that a high processing load is a toll to pay if a high number of results is desired. Indeed curves of message load show a trend comparable to the trend of the number of results. A trade-off should be reached between maximizing the number of results and minimizing processing load; to this aim, we can use the `R` index. Figure 5(b) helps to identify, for a given value of the network size, the `TTL` value that maximizes the efficiency of the network. For networks having less than 400 nodes, the highest values of `R` are obtained with a `TTL` value equal to 3. For larger networks, the "optimum" `TTL` value gradually increases. For example, a `TTL` equal to 4 is preferable for network sizes ranging from about 400 nodes to about 2000 nodes. With 10000 nodes, a `TTL` equal to 6 outperforms other `TTL` values, and curve trends seem to indicate that, for even larger networks, a `TTL` equal to 7 would be the best choice.

In Figure 6, the number of results and the ratio `R`, obtained with the distribution pattern *B*, are reported. The overall number of elementary resources is the same with the two distributions, but the number of resource classes, for a given network size, is lower with distribution *B* than with distribution *A* (see Figure 3). As a consequence, with distribution *B* a higher number of results can be obtained when a query is issued for resources belonging to a specific resource class. Figure 6(b) can be used to tune the `TTL` value, as described for distribution *A*.

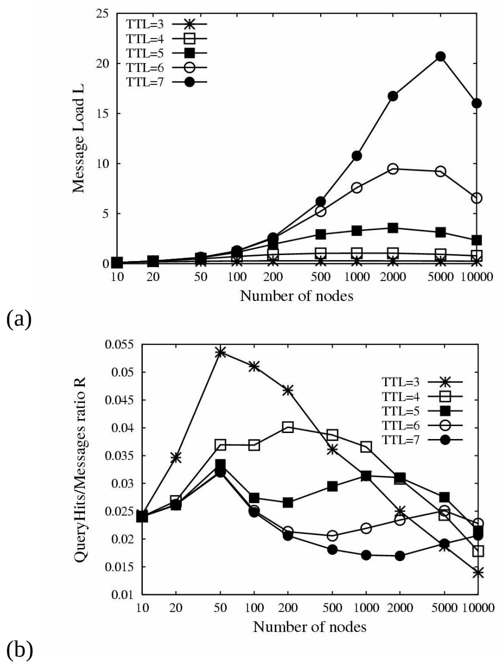


Figure 5. Message load (a) and queryHits/messages ratio (b) w.r.t. the network size, for different values of TTL, with resource distribution *A*.

In Figure 7 we compare the number of results and the values of the ratio `R` obtained with distributions *A*, *B*, and *C*. It can be observed that the highest values are obtained with the constant distribution, especially for large networks. As mentioned before, the constant distribution *C* is a non realistic one, but its performance can be considered as an upper limit for the performance that can be achieved in a large network. The performance gap between distribution *A* and *B* can be explained by comparing the distribution patterns reported in Figure 3: performances are similar if the network is very small (up to 20 nodes) or large (more than 500 nodes). However, in a medium-sized network, distribution *B* outperforms distribution *A*, because the number of resource classes provided with the distribution *B* is lower.

From Figure 7, it appears that the classification of resources within an application domain has an important impact on the performance of the discovery protocol. A fine-grained classification is useful to discover resources having well defining properties. However, a coarse-grained classification permits to discover a higher number of resources belonging to a specific class; among such resources, a Grid scheduler can choose the best one (according to some given criteria) at a later time.

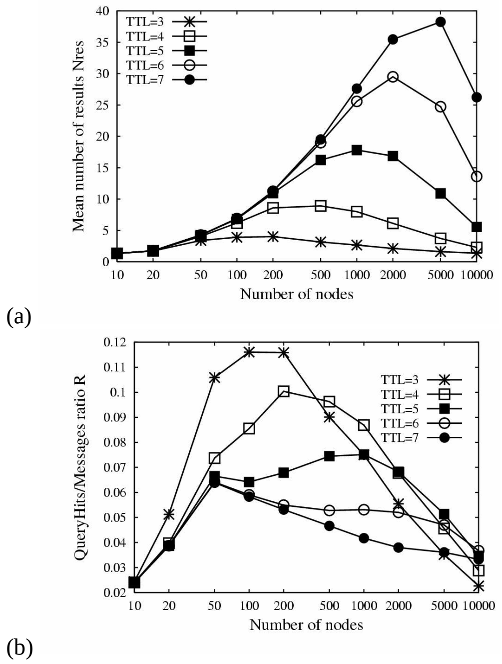


Figure 6. Mean number of results (a) and queryHits/messages ratio (b) w.r.t. the network size, for different values of TTL, with resource distribution *B*.

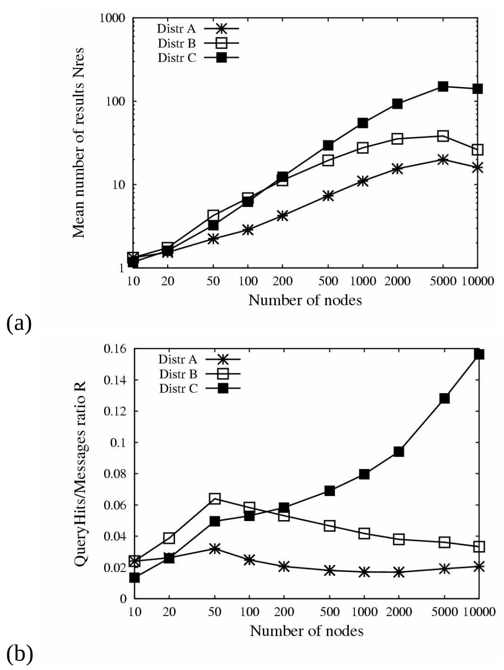


Figure 7. Comparison of resource distributions *A*, *B* and *C*. Mean number of results (a) and queryHits/messages ratio (b) w.r.t. the network size, with TTL=7.

5. Conclusions

The P2P model is emerging as a new distributed computing paradigm because of its potential to harness the computing, storage, and communication power of hosts in the network to make their underutilized resources available to others. P2P shares this goal with the Grid, which was designed to provide access to remote computing resources for high-performance and data-intensive applications.

Resource discovery in Grid environments is based mainly on centralized or hierarchical models, like in the Globus Toolkit. Because such information systems are built to address the requirements of organizational-based grids, they do not deal with more dynamic, large-scale distributed environments, in which useful information servers are not known a priori. The number of queries in such environments quickly makes a client-server approach ineffective. To overcome these limitations, future Grid systems should implement a P2P-style decentralized resource discovery model that can support Grids as open resource communities.

This paper proposed P2P protocols for providing membership management and resource discovery services in a Grid environment. The membership management protocol exploits the use of “contact nodes” for an efficient construction of interconnections among Grid nodes. The resource discovery protocol uses those interconnections to give Grid nodes the opportunity to explore the Grid and discover a large variety of resources.

The paper analyzed and discussed the performance of the resource discovery protocol, and investigated the impact that network and protocol parameters can have on the performance.

6. References

[1] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, “Grid services for distributed system integration”, IEEE Computer, 35(6), 2002, pp. 37-46.

[2] Freenet, http://freenet.sourceforge.net.

[3] The Globus Alliance, “Information Services in the Globus Toolkit 3.0 Release”, http://www-unix.globus.org/toolkit/mds.

[4] The Globus Alliance, the Web Services Resource Framework, http://www.globus.org/wsrf.

[5] The Gnutella Protocol Specification v.0.4, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[6] A. Iamnitchi, I. Foster, J. Weglarz, J. Nabrzyski, J. Schopf, and M. Stroinski, “A Peer-to-Peer Approach to Resource Location in Grid Environments”, eds. Grid Resource Management, Kluwer Publishing, 2003.

[7] C. Mastroianni, D. Talia, P. Trunfio, “Managing Heterogeneous Resources in Data Mining Applications on Grids Using XML-Based Metadata”, International Parallel and Distributed Processing Symposium (IPDPS), Nice, France, April 2003.

[8] A. Montresor, “A Robust Protocol for Building Superpeer Overlay Topologies”, Proc. of the International Conference on Peer-to-Peer Computing (P2P 2004), Zurich, Switzerland, August 2004.

[9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A scalable content-addressable network”, Proceedings of ACM, SIGCOMM, August 2001.

[10] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H.Balakrishnan, “Chord: a scalable peer-to-peer lookup service for internet applications”, Proceedings of ACM SIGCOMM, August 2001.

[11] D. Talia, P. Trunfio, “A P2P Grid Services-Based Protocol: Design and Evaluation”, Proc. of the European Conference on Parallel Computing (EuroPar 2004), LNCS 3149.

[12] D. Talia, P. Trunfio, “Towards a Synergy between P2P and Grids”, IEEE Internet Computing 7(4): 94-96 (2003).

[13] B. Yang, H. Garcia-Molina, “Designing a Super-Peer Network”, 19th Int'l Conf. Data Engineering, IEEE Computer Society Press, Los Alamitos, CA, March 2003.