Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

P2P Protocols for Membership Management and Resource Discovery in Grids

C. Mastroianni, D. Talia, O. Verta

RT-ICAR-CS-04-02	**Marzo 2004**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

P2P Protocols for Membership Management and Resource Discovery in Grids

C. Mastroianni[1], D. Talia[2], O. Verta[1]

Rapporto Tecnico N.: **RT-ICAR-CS-04-02**	***Data:*** **Marzo 2004**

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Cosenza, Via P. Bucci 41C, 87036 Rende(CS).
[2] Università degli Studi della Calabria (o altro), DEIS, Via P. Bucci 41C, Rende (CS).

P2P Protocols for Membership Management and Resource Discovery in Grids

Carlo Mastroianni[1], Domenico Talia[2], Oreste Verta[1]

[1]ICAR-CNR
Via P. Bucci, cubo 41-c, 87036 Rende (CS), Italy

[2]DEIS
Università della Calabria, Via P. Bucci, cubo 41-c, 87036 Rende (CS), Italy

Abstract

P2P techniques and protocols can be used to implement scalable services and application in Grids. This paper proposes an approach based on different P2P protocols for handling two key aspects in Grid environments: membership management and resource discovery. The membership protocol defines how a Grid node can join a Grid network and determines the remote nodes that a node is allowed to contact directly, i.e. the “neighbour” nodes. Resource discovery protocols are used by a Grid node to search the network for hardware or software resources needed by that node. Our approach allows for a separated management of the two mentioned aspects that in P2P systems are generally handled with very similar protocols. In particular, the membership algorithm exploits the use of “contact nodes” elected within each Virtual Organization to play the role of intermediary nodes for interconnections and communication among Grid nodes. The resource discovery protocol uses these interconnections to give ordinary Grid nodes the opportunity to explore the Grid and discover a large variety of resources. Due to the large heterogeneity of Grid resources, the distribution of resources in a Grid does not usually adhere to distribution patterns experienced in classical file sharing P2P systems, and can very likely depend on particular application domains. In fact, within each application domain, categorization of heterogeneous resources is widely exploited to improve the effectiveness of resource discovery tasks. The paper analyzes the performance of the proposed protocols on Grid networks with different sizes, and in particular investigates the impact that different resource distributions can have on performance.

1. Introduction

Grid computing and P2P computing models share several features and have more in common than we generally recognize. The integration of the two computing models could bring benefits in both fields and could results in future integrations. In fact, the convergence of P2P and Grid systems is an emergent topic that will drive the future of distributed information systems. In particular, the use of P2P protocols is expected to improve the efficiency and scalability of large-scale Grid systems [15].

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 thus help to ensure Grid scalability. Designers can use the 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 Globus Toolokit 3 is built upon the OGSA architecture [4], based on Web Services) will surely favour a convergence between the two models, since Web Services can be used to implement peer-to-peer interactions between hosts belonging to different domains. In particular, an ongoing effort aims at studying how it is possible to drive this integration trend to efficiently handle two central issues in Grid information systems: *membership management* and *resource discovery*. The objective of the membership management protocol, according to the definition used in [6] is twofold: adding a new node to the network, and assigning this node a set of neighbour nodes. The latter operation, referred to as "node discovery", is invoked by a node when it joins the network, and periodically at successive times. The resource discovery protocol is exploited when a node needs to access and use a hardware or software resource having certain characteristics, and wants to discover that resource on the network.

In currently deployed Grid systems, resources are often owned by research and public institutions or large enterprises. As a consequence, the number of nodes is generally low, and both hosts and resources are quite stable. Dealing with the membership management issue is not a very hard task, since connections among Grid nodes, if compared to connections in P2P systems, are generally more persistent, and communication sessions require strong authorization mechanisms. In a typical configuration, users of a Grid node know in advance the hosts to which they want to connect, and follow the authorization procedures that allow for the mutual access and use of remote resources. Similarly, the resource discovery issue is managed by taking into account that usually current Grid systems do not have to deal with very dynamic environments. Resources are added, removed or modified with a relatively low frequency: hence resource discovery can be managed through centralized or hierarchical approaches.

Opposite to Grids, current P2P systems present a high degree of dynamicity for both nodes and shared resources: nodes are very often connected and disconnected from the network, and the set of resources shared by each node can be frequently updated. This leads to a very different way to tackle membership management and resource discovery issues. In a pure P2P system [8], a node, soon after connecting to the network, initiates a procedure to discover its neighbour peers. The set of neighbours should include a high number of nodes, since many of these nodes can be switched off or disconnected in a relative small amount of time. For the same reason, the peer discovery procedure has to be periodically repeated, in order to update the set of neighbours. Resource discovery protocols differ from system to system: they may use flooding strategies, index servers, or more efficient schemas. In all cases these protocols have to cope with a high level of dynamicity, and thus often use caching or metadata migration techniques.

As said above, the large diffusion of Grid systems and the availability of large-size Grids will favour the deployment of P2P Grid systems in which techniques and protocols currently used in P2P systems will be efficiently exploited. However, such techniques and protocols should be revisited and adapted in order to exploit the features of future Grid systems, in which:

- the number of nodes will rapidly increase, and many of them will be low and medium-performance computers;
- node interconnections will be more volatile, without reaching the degree of dynamicity of current P2P systems. The level of dynamicity will be limited by security considerations that will continue to be important in future Grids: two nodes have to know and authenticate each other, directly or indirectly, before sharing their resources.
- the number, variety and dynamicity of shared resources will also increase, and will depend on the typology of Grid systems (e.g., general-purpose Grids or domain-specific Grids).

When applying P2P techniques to a Grid system, these features should be taken into account. Accordingly, this paper proposes P2P protocols that manage membership management and resource discovery in a Grid environment. The paper analyzes the performance of the proposed protocols on Grid networks with different sizes, and in particular investigates the impact that different resource distributions can have on the performance.

The paper is organized as follows. Section 2 discusses related work. In Section 3 we introduce a novel method that can be exploited to set up interconnections between Grid nodes. This method is based on the presence of *contact nodes*, i.e. nodes that are used as intermediate nodes to deal with the membership managment problem. A simple approach for resource discovery is also described. In Section 4 the proposed P2P protocols are evaluated by means of an event-driven simulation framework: in particular, simulation runs are executed to investigate the impact of some relevant protocol parameters and the influence of the network size. Furthermore, the role of different resource distribution patterns is analyzed and discussed. Section 5 concludes the paper.

2. Related Work

There are several existing studies on the performance of P2P systems and in particular on membership protocols and resource discovery protocols. P2P systems are organized according to different degrees of centralization. In pure systems such as Gnutella [8] and Freenet [5], all peers have equal roles and responsibilities, while in hybrid systems such as Napster, search operations are performed by using centralized indexes.

In Gnutella, both the membership protocol and the resource discovery protocol are based on a flooding approach: requests, aimed at discovering neighbour peers or resources, flood the network until the time to live expires, while responses come back by following the same path. Freenet adds a replica management mechanism that allows for the replication of “popular” files in peers located in the path from the source to the destination. An accurate study of these two systems can be found in [13]. There it is shown that performance is heavily influenced by possible degenerations to the client/server model, that can happen when a high number of peers request to access remote resources without offering an appropriate number of resources to other peers.

In Napster, prior to access the network, a node registers at a central index and communicates the resources it wants to share. Resource requests are directed to a number of central indexes: once found, a resource is retrieved with a peer-to-peer communication between the requesting peer and the peer that possesses the resource. Much research has been focused to improve search efficiency: CAN [11], Chord [14] and Pastry [12] build efficient indexing structures that provide good scalability and search performance.

In [17], an experimental setup uses log data, extracted from a real Gnutella network, to evaluate a number of enhanced strategies aimed at improving search performance, namely “iterative deepening” (queries are iteratively send to a progressive number of nodes until results are satisfactory), “directed BFS” (queries are forwarded to a selected subset of nodes) and “local indices” (each peer maintains metadata about resources held by neighbour peers).

In [1], the Gridella system uses a decentralized information structure refereed to as “P-Grid”. Resources are indexed by means of binary strings, and each peer is randomly assigned a portion of the overall search space, individuated by a bit pattern. If a peer receives a query string it cannot satisfy, it is always able to forward the query to another peer that possesses resources whose associated bit pattern is closer to the searched one. The main problem with this approach is that it needs a wide agreement among peers about the modalities that allow for the association of binary strings to resources.

Membership and resource discovery are key issues in Grid systems. In Grids, resource discovery is usually managed with centralized or hierarchical mechanisms. In the Globus Toolkit 2 [2], a node that wants to connect to the Grid registers at a centralized index server, the Globus Index Information Server (GIIS), and periodically communicates to that server information about the resources offered to other nodes. GIIS servers are organized according to a hierarchical approach: query messages are delivered to a high-level GIIS and then possibly forwarded to lower-level index servers.

The information model exploited in the Toolkit Globus 3, the newly version of Globus built upon OGSA, is based on Index Services [7], a specialized type of Grid Services. Index Services are used to aggregate and index “Service Data”, i.e. metadata that are associated to the resources provided by Grid hosts. There is typically one Index Service per Virtual Organization but, in large organizations, several Index Services can be hierarchically included in a higher-level Index Server.

However, the Grid community agrees that it is not easy to devise scalable Grid resource discovery based on centralized or hierarchical mechanism when a large number of Grid hosts, resources, and users has to be managed also because of the heterogeneity of such resources.

In [6], characteristics of Grid and P2P systems are discussed and compared, and it is argued that these two worlds are likely to converge in terms of their concern, as Grids scale and P2P systems address more sophisticated application requirements. Furthermore, an emulated large-scale environment is exploited to analyze the performance of four request propagation strategies, based on the hypothesis that each node is assigned a high number of neighbours, and each query message is delivered to a chosen subset of these neighbours. To perform such a choice, the four evaluated strategies use different combinations of random techniques and techniques based on past experience. The importance of determining the distribution of different resource types, instead of considering just the distribution of elementary resources, is discussed in this work, even if the impact of resource distributions on system performance is not deeply analyzed.

Recently, 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. While a super-peer node act as centralized resource for a number of regular nodes, super-peers connect to each other to form a P2P network. In [16], performance of super-peer networks is evaluated, and rules of thumb are given for an efficient design of such networks, where the objective is to achieve a good search performance and at the same time to limit bandwidth and processing load.

Super-peer networks can be efficiently exploited for supporting resource discovery in Grids. In fact, Grids can be naturally modelled as super-peer networks, where generic nodes in a Virtual Organization can be considered as peers while, within each Virtual Organization, one or more nodes, e.g. those that have the largest capabilities, can act as super-peers.

3. A P2P Protocol for Membership Management and Resource Discovery in a Grid Environment

We already mentioned that node interconnections in Grids, if compared with classic file sharing P2P systems are much more stable, and the frequency at which node go up and down is significantly lower. Therefore, it is not necessary for a Grid node to maintain a large set of neighbours. As shown in [17], forwarding a query to a small set of neighbours, or even to only one neighbour, is sufficient to achieve high probabilities of success and also allows for a considerable decrease of the network load. Furthermore, a frequent execution of the node discovery procedure is neither useful, since the Grid network dynamicity is not very high, nor efficient, since it would cause a relevant increase of network and host load.

As a consequence of these considerations, we propose to clearly separate the procedures used by membership management and resource discovery protocols.

The P2P membership management protocol is based on the introduction of a particular type of nodes, the *contact nodes*. A contact node is a Grid node that plays the role of an intermediary node during the Grid network building phase. One or more contact nodes should be made available by each Virtual Organization that wants to connect to the Grid. In this way, a set of contact nodes is published and can be accessed by all the other Grid nodes.

Whenever a Grid node wants to connect to the network, it contacts a small subset of contact nodes - for example the contact nodes from which it experiences the lowest communication delays - 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 newly interconnected Grid node.

A Grid node should turn to the contact nodes either periodically or whenever it detects the disconnection of a neighbour node in order to ask for its substitution. However, the updating frequency is foreseen to be much lower than the frequency at which node discovery procedures are launched in classical P2P systems.

In the following, we briefly describe the membership management protocol, i.e. the protocol used by a node to join the network and build its neighbour sets, and then describe the resource discovery protocol.

Figure 1 shows a graphical description of the membership management protocol. A number of contact nodes are depicted, and for each one of them the corresponding set of registered nodes is reported. In Figure 1(a) a new node wants to connect to the network, and selects two contact nodes, for example on the basis of a minimum distance criterion. In Figure 1(b), the selected contact nodes add the requesting node to the list of registered nodes and respond by communicating the addresses of a number of neighbour nodes, which will constitute the neighbour set of the newly interconnected node. The neighbours are selected randomly, but their number is fixed and predetermined.

The membership management protocol requires a proper setting of three main parameters:

1. the `number of contact nodes` N_{cn}, given as a percentage of the overall number of nodes;
2. the number of contact nodes, or `contact parameter K` ($K<=N_{cn}$), to which a new node has to register (in this example `K` is set to 2);
3. the `number of neighbour nodes V`, set to 4 in this example.

In section 3, we will show how these parameters can have a significant impact on system performance.

Periodically, each node verifies the presence of its neighbours and, if one or more of them result to be disconnected, asks the contact nodes to substitute the missing neighbours. For the sake of simplicity we will not consider this aspect in our analysis.

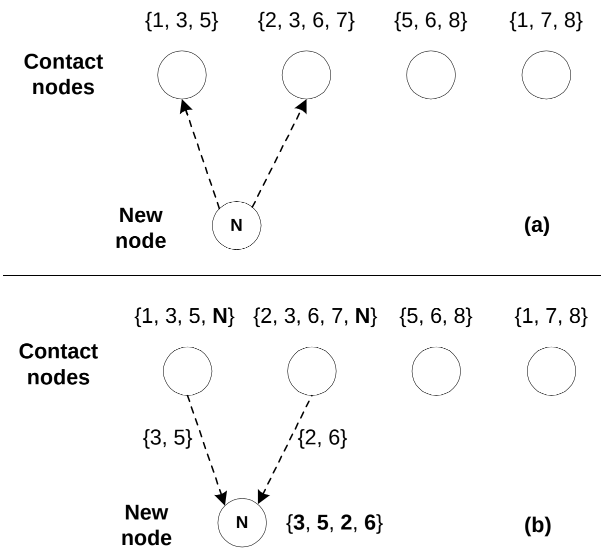


Figure 1. The membership management protocol: a new node joins the P2P network.

The resource discovery protocol does not request the intervention of contact nodes: a Grid node sends its queries to the nodes belonging to its neighbour set, and these queries are in turn forwarded according to the chosen P2P policy. The resource discovery policy is similar to that used in Gnutella, and is based on a flooding approach: each node sends its `query` messages to all its neighbours, which in turn forward them 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: at this point, the requesting node can directly contact the remote node to access the discovered resource. Note that, as said above, contact nodes are not involved in the resource discovery protocol. Also note that, with respect to the Gnutella algorithm, the limitation in the size of the neighbour set contributes to limit the number of messages that circulate in the network.

A number of further techniques are adopted to decrease the network load:

- the number of hops is limited by a Time-To-Live (`TTL`) parameter;

- within each query message, a number of bits is used to annotate all the nodes that the query traverses along its path. By exploiting this information, a node can avoid to forward a query to a neighbour node that has already received that query;
- each node maintains a cache memory where it annotates the IDs of the last received query messages. A node is therefore able to discard those queries that already reached that node by following a different path.

In current P2P systems, node discovery and resource discovery mechanisms adopt very similar policies, both based on the forwarding of request/response messages (often referred to as "ping/pong" messages in node discovery and "query/queryHit" in resource discovery). The differentiation we propose, based on the role of contact nodes, can be useful for the following reasons:

- The processing load at normal Grid nodes can be lowered, since they only have to process query messages, but do not receive node discovery requests, which are instead managed by contact nodes.
- The network load can decrease since node discovery messages do not travel around the network.
- The number of neighbours is the same for all Grid nodes. This avoids the load unbalancing that can occur in typical P2P systems, where different peers are linked to different number of neighbour peers, and thus can receive and send very different numbers of messages.
- The performance of the protocol is similar for different Grid nodes; for example, the success probability of a query does not strongly depend neither on the particular node that forwards the query, nor on which are the neighbours of the requesting node. This allows for analyzing the protocol, and properly tuning its parameters, without the need of analyzing the performance of single Grid nodes.

4. Simulation Analysis

The performance of the proposed P2P protocols were evaluated in order to assess their effectiveness when they are used in a Grid environment, and to estimate the impact of protocol parameters on performance measures.

Analysis was performed by using an event-based object-oriented simulator that has been already used and validated in previous works [3]. In this context, the simulator was used both for modelling the construction of the P2P network, driven by contact nodes, and for simulating the behaviour of the resource discovery protocol in very large Grid networks.

Emphasis is given to the role of different resource distribution patterns. Whereas in a file sharing P2P networks users usually search for well definite resources (e.g. MP3 or MPEG files), in Grid systems it is often required to discover software or hardware resources that rather belong to classes of resources (e.g. software that provides a given function, hosts that offer specific features). These classes can be more or less wide, depending on their semantic definition within a particular application domain.

In the following, we will define a set of resource distribution patterns that will be used and compared in our simulations. Then, parameters and performance indices will be defined and discussed. Finally, we will show a number of interesting results obtained first by varying the values of protocol parameters on a fixed P2P network, then by varying the network size in order to investigate the scalability features of the protocols.

4.1 Resource distribution patterns

In both Grid and P2P environments, resource distribution among nodes is far from being uniform: it usually happens that some nodes share a much larger number of resources with respect to other nodes, even if several approaches have been proposed to increase the level of fairness and to encourage nodes to share their resources instead of solely using other nodes' resources. In our model (see also [6]), we assume that the average number of resources shared by each node is constant as the network size increases. In our experiments, we chose this constant equal to 5, and assumed that the actual number of resources of a given node is randomly extracted from a negative exponential distribution with mean 5.

As said above, a peculiar property of Grid systems is that very often users, in order to build and execute their applications, need to discover not exactly a particular resource, but rather a resource that belongs to a given class. For example, when building a distributed data mining application [9], a user might need to use software having given features (e.g. software that executes a cluster algorithm on semi-structured data or running on particular machines).

The performance of a resource discovery protocol is therefore strictly related to the categorization of the heterogeneous resources in a given application domain. In this sense, the number and distribution of resource types, more than the number and distribution of elementary resources, can contribute to determine the result of a discovery operation.

Opposite to the number of resources, the number of resource types does not increase linearly with the network size, since it often happens that a new node connecting to the network shares resources belonging to resource types already provided by other nodes. To take into account the impact of the distribution of resource types on performance results, we analyzed three kinds of distributions:

- a logarithmic distribution (from now on also refereed to as *distribution A*): the number of resource types increases as `5*log₂(N)^2`, where `N` is the number of nodes in the network;
- a square root distribution (or distribution B): the number of resource types increases as `40+5*sqrt` (N);
- a constant distribution (or distribution C): the number of resource types does not vary with the size of the network, and is set to 100.

These three distributions are plotted in Figure 2. In distributions A and B, the respective numbers of resource types are almost equal for `N` = 10, but increase with different trends. Distribution C is an extreme case, since it is assumed that any new node can only provide resources belonging to a predetermined set of resource types, i.e. it does not have the opportunity to introduce new types of resources: distribution C is mostly used for comparison purposes.

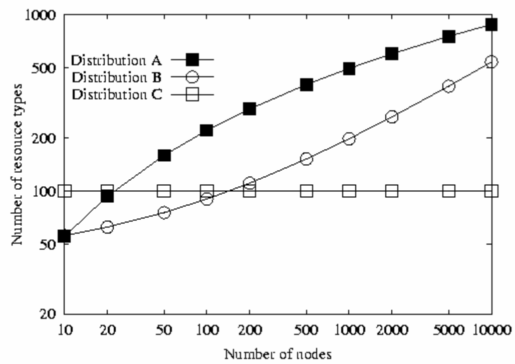


Figure 2. Distributions of resource types used in the protocol analysis.

4.2 Simulation Parameters and Performance Measures

In Table 1, we summarize the main simulation parameters and their respective values.

The mean query interarrival time `MQAT` is the mean time that elapses between two successive queries issued by a single node, and is set to *5 min* in all simulation runs; the actual interarrival time is extracted from a gamma statistical distribution with a shape parameter equal to 2. When issuing a query, a node indicates the class of resources it is searching for: in the simulation, such class is randomly selected among a number of classes that depends on the network size and on the chosen distribution of resource types.

Parameter	**Value**
Number of nodes N	10 to 10000
Number of contact nodes Ncn	10% of N
Mean number of resources per node Nr	5
Distribution of resources	Distributions A, B, C
Contact parameter K	2 to 5
Number of neighbours of a node V	2 to 6
Time to live TTL	3 to 7
Mean query interarrival time MQAT	300 sec

Table 1. Simulation parameters

In Table 2, we report and define the performance measures that were calculated at the end of each set of simulation runs. The performance index N_{qh} is deemed to be more relevant than the probability of success P_{succ}, since it is often argued that the "satisfaction of the query" depends on the number of results returned to the user: for example, in [17] a resource discovery operation is "satisfactory" only if the number of queryHits exceeds a given threshold. The L_p index gives a measure of the "time to satisfaction": in our simulation we suppose that each hop causes a delay equal to 0.1 seconds, including both the communication delay and the processing delays at the transmitting and at the receiving node. The message load L should obviously 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 network loads. Finally, the ratio R is an indication of the "utility" of messages that circulate through the network: if this index increases, it means that a higher relative number of queryHits messages are generated or forwarded with respect to the overall number of messages (queries + queryHits).

Performance index	**Definition**
Probability of success 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 queryHits N_{qh}	Mean number of queryHits that a node receives from remote (distinct) nodes, for a *successful* query (i.e. a query with at least one queryHit)
Mean length of the success path L_p	Mean number of hops that a query traverses before reaching a node that possesses the requested resource
Message load L	Frequency of the messages (queries and queryHits) received by a node
QueryHits/Messages ratio R	Number of queryHit messages received by a node divided by the overall number of messages received by that node

Table 2. Performance indices

4.3 Results for a network with fixed size

The proposed P2P protocols were first analyzed for a Grid network composed of 1000 nodes with the hypothesis that the distribution of resource types follows the pattern A.

A first set of simulations were run to evaluate protocol performance with respect to the values of the `TTL` and the contact parameter `K`. The value of `K` determines the topology of the network, since this parameter, used by the membership management protocol, defines the number of contact peers at which a peer attempts to register. We generated different network topologies corresponding to different values of `K`; then, for a fixed network topology, different values of `TTL` were experimented.

In Figure 3 the mean number of queryHits, the mean path length, the queryHits/messages ratio, and the message load are reported, for `K` values ranging from 2 to 5, and `TTL` values ranging from 2 to 7. The number of neighbours `V` is set to 4. It is evident that an increase of `TTL` causes a remarkable increment of the query satisfaction but also a corresponding strong increase of the network load.

The impact of the contact parameter can be also analyzed. An increase of `K` allows for a faster exploration of the network, since the neighbours of a given node are registered at a larger number of different contact nodes. However, increasing `K` beyond a certain threshold (i.e. beyond 3 for this network size) appears to be ineffective. From results not shown in this paper, it can be seen that the effect of increasing `K` is more relevant for larger networks.

In Figure 4 we report the results obtained with a fixed value of `K` (i.e. `K`=2), and by varying the `TTL` value from 2 to 7 and the number of neighbours `V` from 2 to 5. It is shown that an increase of either `TTL` or `V` leads to a strong increase in the indices `Nqh` and `L`. However, the effect on `Nqh` diminishes when `TTL` is greater than 6 and `V` is greater than 4.

In this Section, only results obtained with the distribution pattern A are shown. Simulations were also executed with the other two distribution patterns, resulting in performance curves having similar qualitative trends. However, a comparison between different distribution patterns is more interesting and significant if variable network sizes are considered, as we do in the next Section.

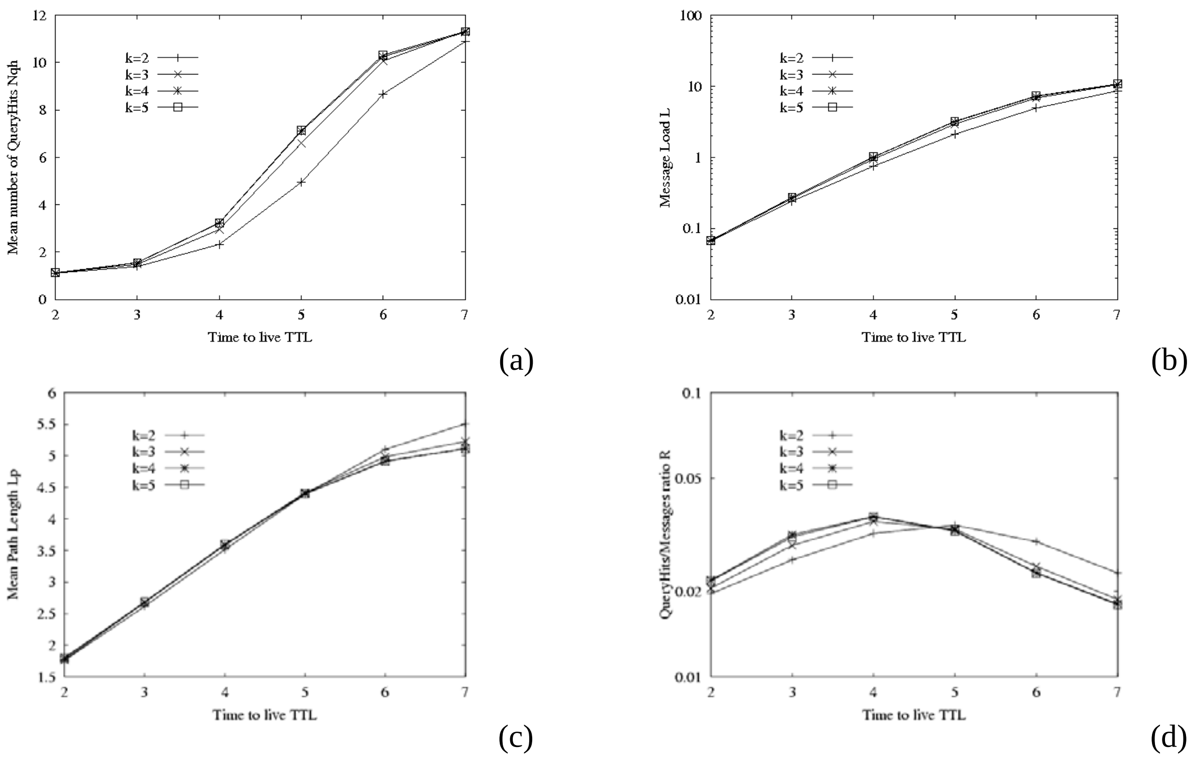


Figure 3. Distribution A. Mean Number of QueryHits (a), Message Load (b), Mean Path Length (c), and QueryHits/Messages ratio (d) w.r.t. the `TTL` value, for different values of the contact parameter `K`

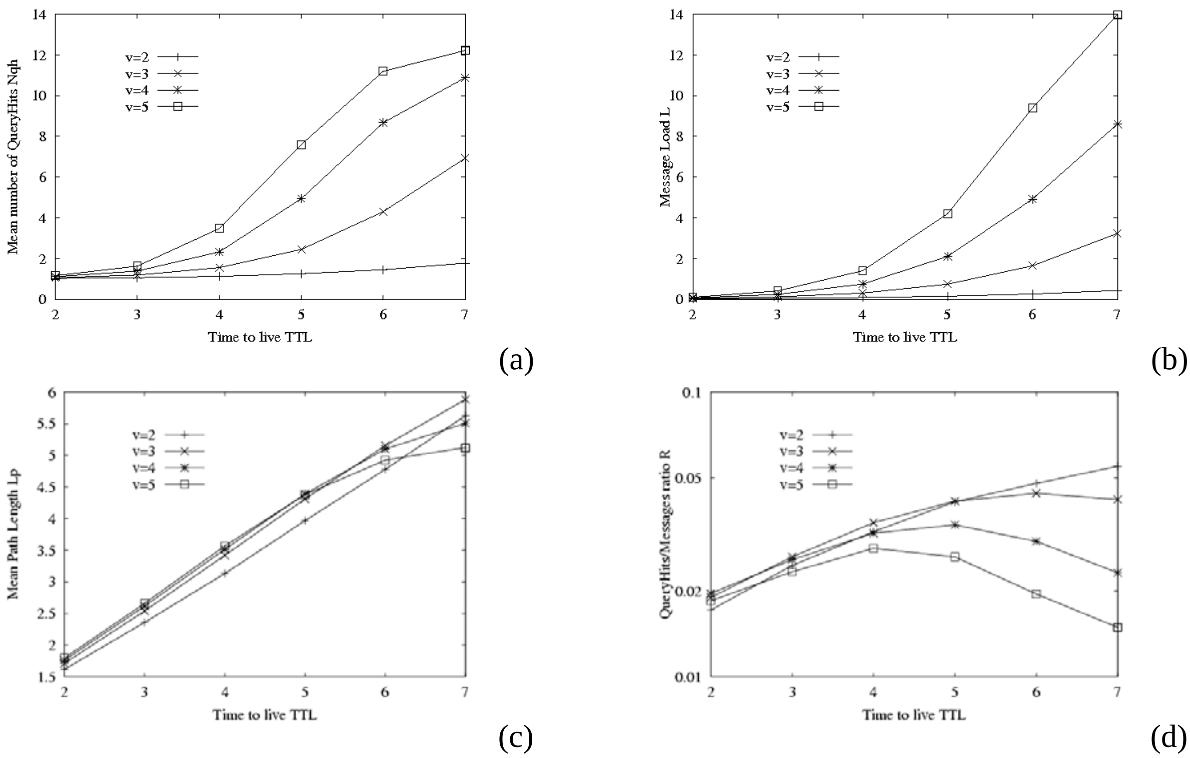


Figure 4. Distribution A. Mean Number of QueryHits (a), Message Load (b) Mean Path Length (c), and QueryHits/Messages ratio (d) w.r.t. the TTL value, for different values of the number of neighbours V.

4.4 Performance results versus the network size

After examining the impact of main protocol parameters on a fixed Grid network, we will now examine how performance results vary with respect to the size of the network, in order to investigate the scalability of the protocol.

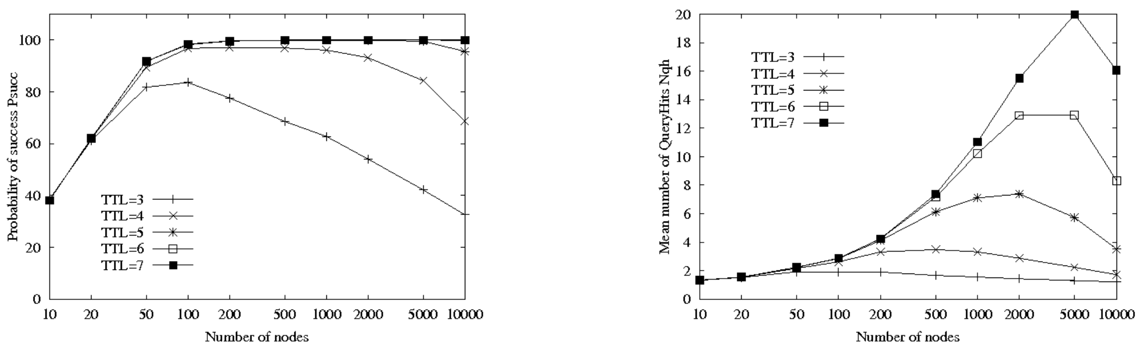
We considered networks with numbers of nodes ranging from 10 to 10000, to take into account network sizes belonging to a very wide range.

In our first experiments, we continued to adopt the distribution pattern A. Furthermore, we set the number of neighbours V and the contact parameter K to 4 and 5, respectively; finally, the TTL value was varied from 3 to 7.

By observing the success performance values reported in Figure 5, it can be noted that an increase of the TTL value causes a benefit (in terms of success probability and number of queryHits) that is negligible for small networks, but more and more significant for larger networks. Furthermore, performance results are strictly correlated both to the number of nodes and to the characteristics of the resource distribution chosen. For small networks, performance is poor because a few available nodes are not able to offer the entire set of resource classes (e.g. when N=10, we have more than 50 types of resources and about 50 resources shared by the whole network: from simple statistical considerations, it follows that it is very unlikely that all the resource types are actually provided). When the number of nodes increases, success probabilities increase. However, for very large networks, performance is limited by the TTL value, which does not allow for an exploration of the whole network. In a network with more than 2000 nodes, a TTL value higher than 7 would be needed to obtain higher performance.

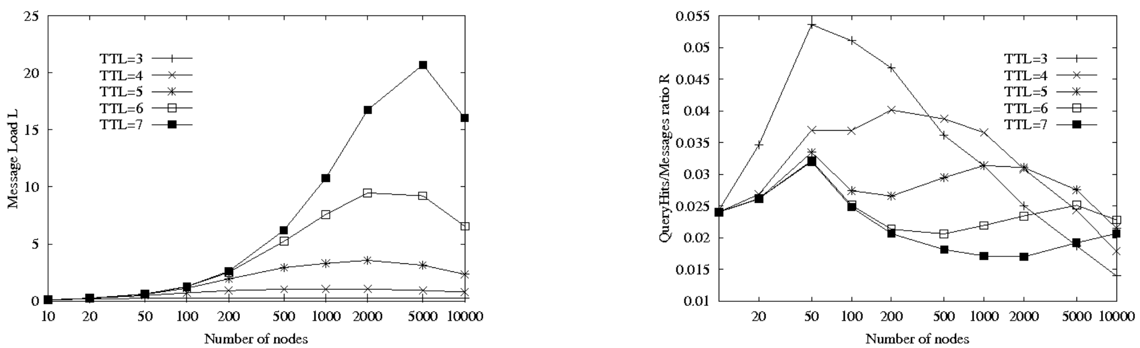
Figure 6(a) shows that the message load experienced by a node is strictly related to the number of successful queryHits that it expects to receive. It can be deduced that higher probabilities of success can only be obtained by accepting higher network and processing loads. However, results depicted in Figure 6(b) show that the percentage of queryHit messages, i.e. the relative amount of “useful” traffic, is an index that can be used to effectively tune the P2P network. Indeed, for networks having less than 400 nodes, the highest QueryHits/messages ratio R is 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 a proper tuning value for network sizes ranging from about 400 nodes to about 2000 nodes. With 10000 nodes, a TTL equal to 6 slightly outperforms other TTL values, while curve trends seem to indicate that for even larger networks a TTL equal to 7 would be the best choice.

Finally, from Figure 7, we can note that the mean path of successful queries increases with the TTL value and with the network size.



(a) (b)

Figure 5. Distribution A. Probability of success (a) and Mean Number of QueryHits (b) w.r.t. the number of nodes, for different values of TTL



(a) (b)

Figure 6. Distribution A. Mean path length (a) and QueryHits/Messages ratio (b) w.r.t. the number of nodes, for different values of TTL

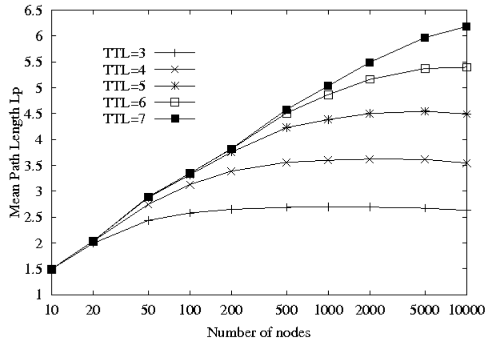
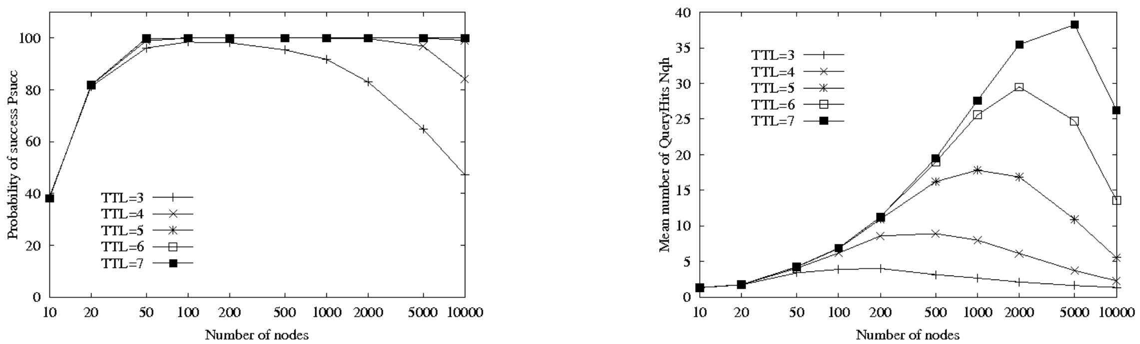


Figure 7. Distribution A. Message load w.r.t. the number of nodes, for different values of TTL

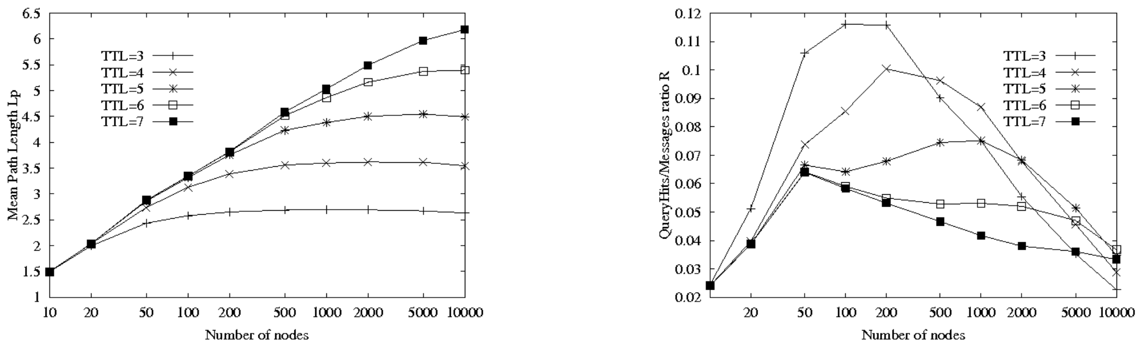
In Figures 8 and 9, performance measures are reported for a different distribution of resource types, i.e. distribution pattern B. Qualitative trends of performance indices are similar for distribution A and B, so we can deduce that these trends are not strictly dependent on the type of distribution. However, with distribution B, the number of resource types, for all network sizes, is lower than the corresponding number obtained with distribution A, resulting in higher success probabilities.

In Figures 10 and 11, we show the performance obtained with distribution C.



(a) (b)

Figure 8. Distribution B. Probability of success (a) and Mean Number of QueryHits (b) w.r.t. the number of nodes, for different values of TTL



(a) (b)

Figure 9. Distribution B. Mean path length (a) and QueryHits/Messages ratio (b) w.r.t. the number of nodes, for different values of TTL

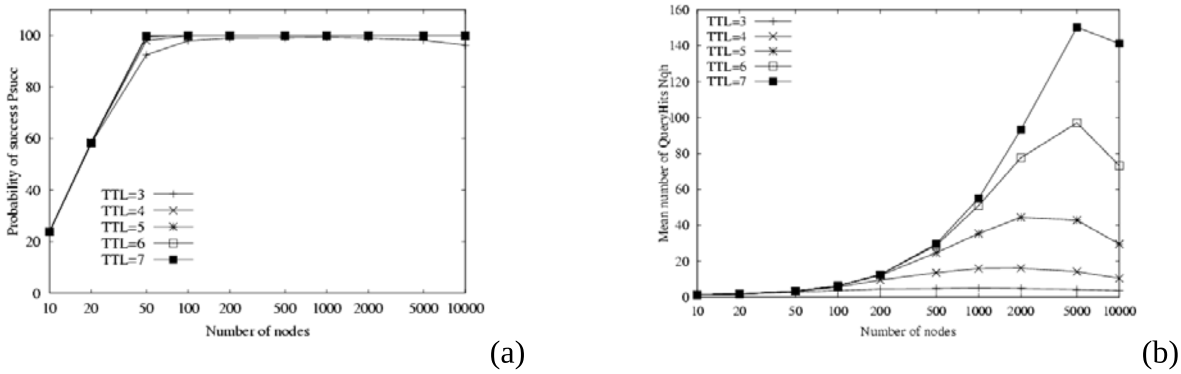


Figure 10. Distribution C. Probability of success (a) and Mean Number of QueryHits (b) w.r.t. the number of nodes, for different values of TTL

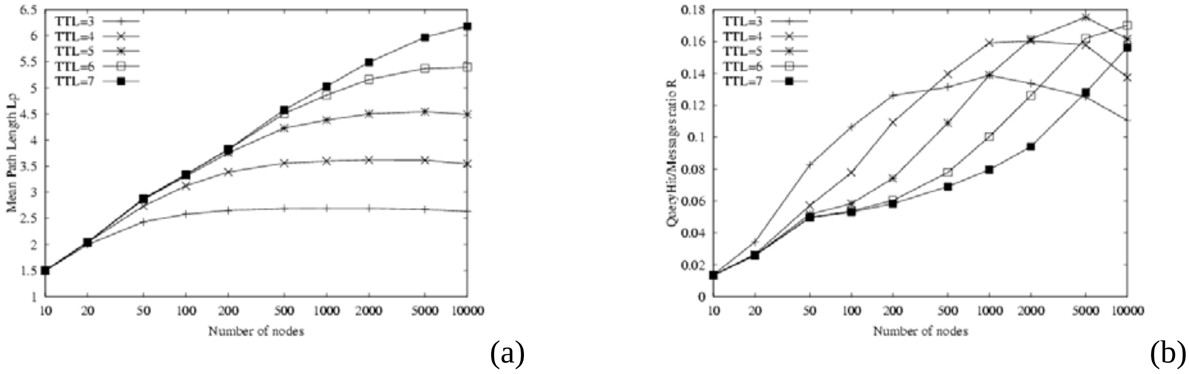


Figure 11. Distribution C. Mean path length (a) and QueryHits/Messages ratio (b) w.r.t. the number of nodes, for different values of TTL

To emphasize quantitative differences among performances achievable with different resource distributions, in Figure 12 we depict the values of indices `Nqh` and `R` obtained with all the three distributions discussed (A, B and C).

It can be observed that both the mean number of queryHits and the percentage of queryHit messages are significantly higher with the constant distribution, especially for large networks. As said before, distribution C is a non realistic one. However, is worthy analyzing, since its performance index values can be considered as a superior limit for index values that are achievable in a large network.

The performance gap between distribution A and B reflects the distribution patterns reported in Figure 1: performances are similar if the network is very small (up to 20 nodes) or very large (more than 500 nodes), because the numbers of resource types corresponding to the abovementioned distributions are close to each other. In a medium-sized network, distribution B performs better than distribution A, because the number of resource types is lower with distribution B.

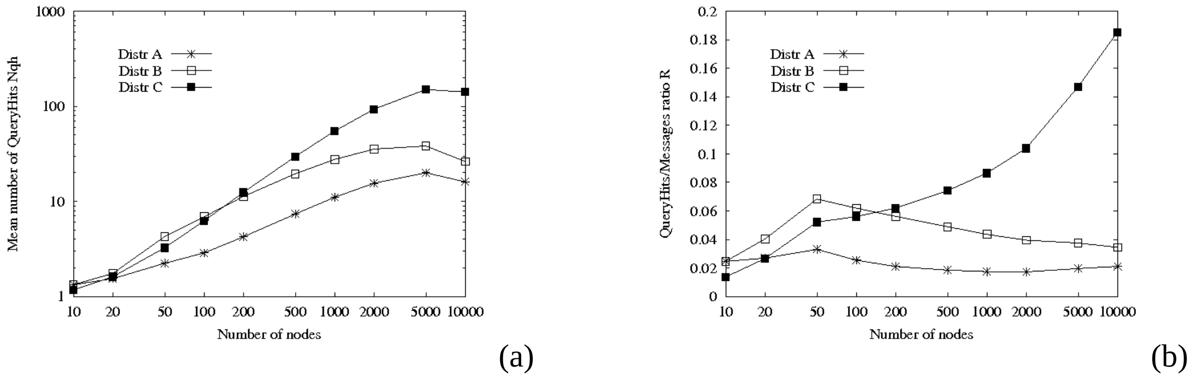


Figure 12. Comparison of Distributions A, B, C. Mean Number of QueryHits (a) and QueryHits/Messages ratio (b) w.r.t. the number of nodes, with TTL=7

5. Conclusions

The P2P model is emerging as a new distributed 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 applications and data-intensive applications.

Resource discovery in Grid environments is based mainly on centralized or hierarchical models. In the Globus Toolkit, for instance, a user or an application can directly gain information about a given node's resources by querying a server application running on it or running on a node that retrieves and publishes information about a given organization's node set. 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.

Resource discovery includes, in part, the issue of presence management—discovery of the nodes that are currently available in a grid—because global mechanisms are not yet defined for it. On the other hand, the presence-management protocol is a key element in P2P systems: each node periodically notifies the network of its presence, discovering its neighbours at the same time. 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 membership management and resource discovery in a Grid environment. Our approach allows for a separated management of the two mentioned aspects that in P2P systems are generally handled with very similar approaches. In particular, the membership algorithm exploits the use of “contact nodes” for an efficient construction of interconnections among Grid nodes. The resource discovery protocol uses these interconnections to give ordinary Grid nodes the opportunity to explore the Grid and discover a large variety of resources.

The paper analyzed and discussed the performance of the proposed protocols on Grid networks with different sizes, and in particular investigates the impact that different resource distributions can have on the performance. We presented simulation results based on several parameters such as resource distribution, contact nodes, neighbour nodes, time to live, and mean query interarrival time. By using such parameters we presented and discussed performance figures of Probability that a query issued by a generic node will succeed, mean number of queryHits, mean length of the success path, Message load, and QueryHits/Messages ratio. The evaluation of those performance indexes allowed for evaluating the proposed protocols under different resource distributions and network conditions.

References

[1] K. Aberer, M. Hauswirth, M. Punceva, R. Schmidt, “Improving Data Access in P2P Systems”, IEEE Internet Computing, 6(1), January/February 2002.

[2] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, “Grid Information Services for Distributed Resource Sharing”, Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[3] G. Fortino, C. Mastroianni, W. Russo, “Performance Analysis of an Application-Level Cooperative Control Protocol”, IEEE Symposium on Network Computing and Applications (NCA 2003), Cambridge, USA, April 2003.

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

[5] Freenet web site, http://freenet.sourceforge.net.

[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] The Globus Project, “Information Services in the Globus Toolkit 3.0 Release”, http://www.globus.org/mds.

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

[9] C. Mastroianni, D. Talia, P. Trunfio, “Managing Heterogeneous Resources in Data Mining Applications on Grids Using XML-Based Metadata”, Heterogeneous Computing Workshop (HCW 2003), Nice, France, April 2003.

[10] Morpheus website, http://www.morpheus.com.

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

[12] A. Rowstron, P. Druschel, “Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems”, Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms, November 2001.

[13] S. Saroiu, P. K. Gummadi, S. D. Gribble, “A Measurement Study of Peer-to-Peer File Sharing Systems”, Proceedings of Multimedia Computing and Networking 2002 (MMCN '02).

[14] 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.

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

[16] 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.

[17] B. Yang, H. Garcia-Molina, “Efficient search in peer-to-peer networks”, Proc. of ICDCS'02, Vienna, Austria, July 2002.