An Efficient Resource Discovery while Minimizing Maintenance Overhead in SDDS Based Hierarchical DHT Systems

Riad Mokadem, Abdelkader Hameurlain

IRIT, Paul Sabatier University, Toulouse, France
{mokadem,hameur}@irit.fr

Abstract

Using Distributed Hash Tables (DHT) for a resource discovery in large-scale systems generates considerable maintenance overhead which affects the routing efficiency. In this paper, we propose a hierarchical DHT solution based on scalable distributed data structures (SDDS) for an efficient resource discovery in data Grids. Our solution deals with a reduced number of gateway peers running a DHT protocol. Each of them serves also as a proxy for second level peers in a single Virtual Organization (VO), structured as an SDDS. We present a cost analysis for a resource discovery process and discuss its capabilities to reduce both lookup and maintenance costs while minimizing the overhead added to the system. The analysis results proved that an SDDS based hierarchical DHT solution offers good performances especially for intra-VO resource discovery queries. They also provide significant system maintenance saves especially when peers frequently join/ leave the system.

Keywords: *Resource discovery, Data Grid, Peer to peer system, Distributed hash table, Scalable distributed data structure, Super peer models, Hierarchical systems.*

1. Introduction

A resource discovery is a real challenge in unstable and large scale environments. It consists to discover resources (e.g., computers, data) that are needed to perform distributed applications in such systems [32]. Throughout this paper, we focus on the discovery of metadata describing data sources in data Grid systems.

Large amounts of research works have been adopt Peer-to-Peer solutions to deal with resource discovery in Grid systems [16, 42]. P2P routing algorithms have been classified as structured or unstructured [43]. The most popular service provided by P2P unstructured systems is the popular file sharing over the Internet (e.g., Gnutella [13], KaZaa [21]). Although the good fault tolerance properties in these systems, the flooding –used in each search- is not scalable since it generates large volume of unnecessary traffic in the network. Structured Peer-to-Peer systems as DHT are self-organizing distributed systems designed to support efficient and scalable lookups in spite of the dynamic properties in such systems. Classical flat DHT systems organize peers, having the same responsibility, into one overlay network with a lookup performance of O(log(N)), for a system with N peers. However, the using of a flat DHT do not consider neither the autonomy of virtual organizations and their conflicting interests [15] nor the locality principle, a crucial consideration in Grids [15]. Moreover, typical structured P2P systems as Chord [38] and Pastry [37] suffer not only from temporary unavailability of some of its components but also from churn. It occurs in the case of the continuous leaving and entering of nodes into the system. Recent research works as [32] proved that hierarchical overlays have the advantages of faster lookup times, less

messages exchanged between nodes, and scalability. They are valuable for small and medium sized Grids, while the super peer model [48] is more effective in very large Grids [3]. In this context, several research works [10, 11, 20, 26, 27, 30, 39, 49] proved that hierarchical DHT systems based on the super peer concept can be advantageous for complex systems. A hierarchical DHT employ a multi level overlay network where peers are grouped according to a common property such as resource type or locality for a lookup service used in discovery [10]. In this context, a Grid can be viewed as a network composed of several, proprietary Grids, virtual organizations (VO) [7, 19, 41] where every VO is dedicated to an application domain (e.g., biology, pathology). Within a group, one or more peers are selected as super peers to act as gateways to peers in the other groups.

Furthermore, most existing hierarchical DHT solutions neglect the churn effect and deal only with the improving performance of the overlay network routing. They mainly generate significant additional overhead to large scale systems. Several proposals [12, 14, 25, 35, 40, 50], for reducing maintenance costs, have also appeared in the literature. [25] proposed the SG-1 algorithm to find the optimal number of super peers in order to reduce maintenance costs. It is based on the information exchange between super peers through a gossip protocol [1]. Also, despite a good strategy to manage a churn in [40] through a lazy update of the network access points, inter-organizations lookups were expensive because of the complex addressing system. Hence, Most of these solutions add significant load at some peers which generates an additional overhead to large scale systems.

In this paper, we propose to improve both lookup and maintenance costs while minimizing the overhead added to the system. Our solution combines a scalable distributed data structure (SDDS) routing scheme [24] with DHT systems for an efficient resource discovery in data Grids. SDDS based Hierarchical DHT (SDDS- HDHT) solution consists of a two level hierarchical overlay network dealing with super peers (called also gateways) and second level peers. Gateway peers establish a structured DHT based overlay. Only one peer (node) per VO is considered as a gateway. Then, each of them serves as a proxy for second level peers in a single VO, structured as an SDDS. SDDS were among the first research works dealing with structured P2P systems [5]. [24] noted numerous similarities between Chord and the best known SDDS scheme: LH* (Linear Hashing) [23]. Both implement key search and have no centralized components. Resource discovery queries, in our system, are classified into intra-VO and inter-VO queries. The intra-VO discovery consists to apply the principle of locality by favoring the metadata discovery in a local VO through the efficient LH* routing system. Key based queries in LH*, in its $LH*_{RS}^{P2P}$ versus, need at most two hops to find the target when the key search in a DHT needs O(log N) hops, N is the number of peers in the system [47]. In fact, super peers are not concerned by intra-VO queries unlike previous solutions as [49] which put super peers more under stress. Regarding Inter-VO queries, they are first routed to the reduced DHT overlay which permits to locate the gateway peer affected to the VO containing the resource to discover. Then, another $LH*_{RS}^{P2P}$ lookup is done in order to discover metadata of this resource.

We mainly demonstrate that our solution can handle high churn rates. The proposed protocol deals with the reduction of maintenance costs especially when peers frequently join or leave the system (dynamicity properties of Grid environments). We explore the different factors that affect the behavior of hierarchical DHT under churn (super peer failure addressing, timeouts during lookups and proximity neighbor selection) [35]. Only the arrival of a new VO requires the DHT maintenance. The connection/ disconnection of gateways do not require excessive messages exchanged between peers in order to maintain the system. This is done through a lazy system update which avoids a high maintenance costs. SDDS-

HDHT solution also overcomes the single point of failure problem without putting super peers on stress neither using complex protocols.

A simulation analysis evaluates performances of the SDDS-HDHT solution through comparison with previous solution performances. It shows the reduction of lookups costs especially for intra-VO queries. It also provides a significantly maintenance costs reduction, especially when peers frequently join/leave the system. We also interest to the optimal ratio between super peers and second level peers for the compared solutions. The rest of the paper is structured as follows. Section 2 details related work. Section 3 recalls hierarchical DHT and SDDS principles. Section 4 presents our resource discovery solution through the proposed protocol. Section 5 contains an example of a resource discovery process. The next section present a cost based analysis for both lookups and system maintenance. The performance study section shows the benefit of our proposition. The final section contains concluding remarks and future works.

2. Related Work

Classical resource discovery approaches in Grids are either centralized or hierarchical and were proved inefficient as the scale of Grid systems rapidly increases [17]. Excessive access to a centralized peer generates bottleneck and its failure paralyzes all the system. The using of web services, inspired from hierarchical models, has been explored in several research works as [46]. Although the advantage of being Open Grid Service Architecture (OGSA) [7] compliant, i.e. each resource is represented as a web service, this strategy is not adapted for grid environments since the dynamicity peer properties in large scale Grids [17]. Resource discovery in Grid systems has been the topic of many researches. [31] proposed super peer based resource discovery solution. Some research works as [18, 43] classified the super peer model [48] as 'hybrid' compared to unstructured and structured classes. Many research works [10, 11, 20, 26, 27, 50] presented advantages of hierarchical DHT systems based on the super peer concept. However, most of them add a significant overhead to the system. [10] proposed a two-tier hierarchy using chord for the top level to reduce the lookup costs, but only with the goal of improving performance of the overlay network routing. [44] demonstrated the high maintenance state needed (memory, CPU and bandwidth) when all peers in the overlay are attached to different levels of the hierarchy. [27] explored the using of multiple Chord systems in order to reduce latency of lookups. Nevertheless, it neglects the churn effects. [50] gives a cost-based analysis of hierarchical P2P overlay network with super peers forming DHT and leaf nodes attached to them. However, super peers are put more under stress for both intra and inter-VO resource discovery queries especially if the leaf nodes number increase. Moreover, performances depend on the ratio between super peer's number and the total number of peers in the system. [20] presented a two-layer structure 'Chord2' to reduce maintenance costs in Chord. The lower layer is the regular Chord ring when the upper layer is a ring for maintenance constructed from super peers. On the other hand, several algorithms [12, 14, 25, 35, 40, 50] were proposed to resolve these problems. We cite the SG-1 algorithm to find the optimal super peer ratios generating a low traffic network [25] and the Bamboo protocol [35] designed to handle networks with high churn efficiently. We also cite the self organizing distributed algorithm developed [50] in which all decisions taken by the peers are based on their partial view in the sense that the algorithm became fully decentralized and probabilistic. However, there is trade-off between minimizing total network costs and minimizing the added overhead to the system. In the next section, we show how to combine DHT and SDDS structures in order to minimize these costs without excessive overheads.

3. Preliminaries

In this section, we present scalable distributed data structures (SDDS) and its principles. Then, we recall some existing hierarchical DHT solutions, their efficiency but also problems generated by their maintenance.

3.1. LH*$_{RS}^{P2P}$ Scalable Distributed Data Structure

Scalable Distributed Data Structures (SDDS), designed for P2P applications, are a class of data structures for distributed systems that allow data access by key in constant time [29]. Many variant of SDDS were proposed. In this paper, we deal with LH*$_{RS}^{P2P}$ scheme which improves latter LH* variants (LH*$_{RS}$, LH*$_{g}$…). We assume that the reader is familiar with a linear hashing algorithm LH* as presented in [23, 24]. Each peer stores records in a bucket which splits when the file grows. Every LH* peer is both client and, potentially, data or parity server which interacts with application using the key based record search, insert, update or delete query or a scan query performing non key operations. In the resource discovery process, we deal with the key search operations which can concern one relation in order to discover its metadata. In this paper, a search can concern some relation or its metadata.

Algorithm 1: LH* Addressing algorithm rule

$$a \leftarrow h_i(C)$$

$$\text{if } a < n \text{ then } a \leftarrow h_{i+1}(C)$$

Basic linear hash functions are $h_i(C) = C \bmod 2^i$. Each record in LH * is identified by its key. The key *C* determines the record location (the bucket number a= 0, 1, 2…) according to the linear hashing Algorithm (Algorithm1). (*i, n*) are the file state. Here i= 0, 1… is the file level which determines the linear hash function h_i to be applied. Likewise, n is the split pointer, indicating the next bucket to split. The file starts with one data bucket and one parity bucket. It scales up through data bucket splits, as the data buckets get overloaded. It can be occurred when a peer splits its data bucket. In old SDDS scheme, one peer acted as a coordinator peer. It was viewed as the single peer knowing the correct state of the file or relation. However, [47] improves this scheme. Split coordinator does not constitute a centralized peer for the SDDS scheme. It intervenes only to find a new data server when a split occurs and never in the query evaluation process. Any other peer uses its local view 'image', which may be not adjusted, to find the location of a record given in the key based query. Suppose that it sent its key based query by using its image (*i', n'*) which can lead to an incorrect bucket *a'*. An outdated image could result in $\grave{a} < a$. The peer server applies another algorithm LH*$_{RS}^{P2P}$ (Algorithm 2). It first verifies whether its own address is the correct one by checking its guessed bucket level *j'* in the received query against its actual level *j* (the level of LH function last used to split or create the bucket). It calculate *j'* as *i'* for $\grave{a} > n'$ and as *i'+1* otherwise. If needed, the server forwards this query. The query always reaches the correct bucket *a* in this step i.e., if forwarding occurs, the new address has to be the correct one. It sends an Image Adjustment Message (IAM) informing the initial sender that the address was incorrect and the sender adjusts its image reusing the LH* image adjustment algorithm described in [24]. It was not the case with LH* which may need an additional hop. Proof of this property is in [47]. Thus, the most important property here is that the maximal number of forwarding messages for key-based addressing is one.

Furthermore, [24] proved the efficient recovery of more than one peer without any duplication. LH*$_{RS}^{P2P}$ scheme allows the correction possibility through parity calculus. It is

done using Reed Solomon (RS) codes [24]. This process is accelerating since RS codes are based in Galois Field (GF) calculus. High availability is a desirable feature of the data in Grid systems. The complete process of a peer recovery and parity peers updating are well detailed in [24]. Another advantage of using SDDS is the less vulnerability in the presence of high churn [47].

Algorithm 2: LH $*_{RS}$ ^{P2P} Key Based Addressing

```
If a'> n' then j' = i' else j'=i'+1;
          If j = j' then Process Query; Exit;
    Else if j'-j=1 then a← h_j(C);
If a > a' then forward Query to bucket a; Exit;
    Else Send 'Error image' to the sender.
```

3.2. Hierarchical Distributed Hash Tables

Structured systems such as DHT are a decentralized lookup scheme designed to provide scalable lookups. It offers deterministic query search results within logarithmic bounds as sending message complexity. In systems based on DHT as Chord [38], Pastry [37], CAN [34] and Tapestry [51], the DHT protocol provides an interface to retrieve a key-value pair. Each resource is identified by its key using cryptographic hash functions such SHA-1. Each peer is responsible to manage a small number of peers and maintains its location information. In this paper, we have focused on a Pastry DHT system [37], adapted for several different applications including a global, persistent storage utility as PAST [6] and a scalable publish/subscribe system called SCRIBE [36]. But, our method can be applied to other DHT systems. Pastry DHT system offers deterministic query search results within logarithmic bounds. It requires Log_B (N) hops, where N is the total number of peers in the system and B typically equal to 4 (which results in hexadecimal digits). Pastry system also notifies applications of new peers arrivals, peer failures and recoveries. Unlike Chord peers, Pastry peer permits to easily locate both the right ad left neighbors in the DHT (through the neighborhood set parameter which is useful in maintaining locality properties). These reasons motivate us to choose the Pastry routing system. Hierarchical DHT systems partition its peers into a multi level overlay network. Because a peer joins a smaller overlay network than in flat overlay, it maintains and corrects a smaller number of routing states than in flat structure. In such systems, one or more peers are often designated as super peers. They act as gateways to other peers organized in groups in second level overlay networks. Throughout this section, we interest to two previous hierarchical DHT solutions which we consider comparable to our solution.

In Figure 1-left, gateway peers (called super peers) establish a structured DHT overlay network when second level peers (called leaf nodes) maintain only connection to their super peers. This corresponds to the Super Peer HDHT (SP-HDHT) solution [49] as shown in Figure 1- left. [26] proved that this strategy can maintain super peers more under stress by maintaining pointers between super peers and their leaf nodes. Furthermore, a super peer stores information's of all leaf nodes which it is responsible and acts as a centralized resource for them. Then, performances depend on the ratio between super peer's number and the total number of peers in the system.

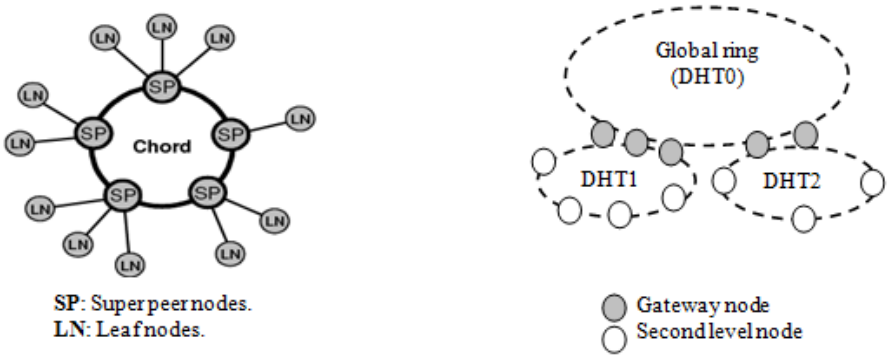

Figure 1: SP-HDHT (left) and MG-HDHT (right) Solutions.

Multi-Gateway Hierarchical DHT (MG-HDHT) solution [27] is another example of 2-levels hierarchy system having multiple gateways by VO (Figure 1- right). The peers connecting by lines are instances of the same peer, running in different DHTs. The system forms a tree of rings (DHTs in this example). Typically, the tree consists of two layers, namely a global ring as the root and organizational rings at the lower level. A group identifier (*gid*) and a unique peer identifier (*pid*) are assigned to each peer. Groups are organized in the top level as DHT overlay network. Within each group, nodes are organized as a second level overlay using the *pid* identifier. This solution provides administrative control and autonomy of the participating organizations. Unlike efficient intra-organization lookups, inter-organization lookups are expensive since the high maintenance cost of the several gateway peers.

In spite of the several algorithms -cited above- proposed to reduce the overhead added to the system, there is a trade-off between minimizing total network costs and minimizing the added overhead to the system. In the next section, we propose a resource discovery protocol which reduces these costs without excessive overheads.

4. Resource Discovery through SDDS Based Hierachical DHT

A resource discovery constitutes an important step in the evaluation of a query in data Grid environments [22, 33]. The fact that users have little or no knowledge of the resources contributed by other participants in the grid poses a significant obstacle to their use. Hence, a data placement scheme must be defined. It is done since we have not a centralized scheme which forms naturally a bottleneck for the system [17]. The duplicated approach forces the update in every peer containing a copy of the dictionary which will result in flooding the network. Hence, the distributed approach is more appropriate in large scale systems [16, 28]. In this context, distributed Peer to Peer techniques are used to discover metadata describing data sources in data Grids. Furthermore, Grid environment is likely to scale to millions of resources shared by hundreds of thousand of participants. In consequence, resource discovery generates high costs. More, the fact that peers frequently leaved and joined the grid system (dynamicity of peers) generates high maintenance costs on the same basis of the unavailability of some peers. In this case, managing a churn can add overhead to the system.

In this section, we present an SDDS based hierarchical DHT solution for resource discovery in data Grids. It aims to reduce both lookup and maintenance costs while minimizing overhead added to the system. Resource Discovery through our solution deals with two different classes of peers: gateways (called also super peers) and second-level peers. A Grid can be viewed as a network composed of several, proprietary Grids, virtual

organizations (VO) [19] as shown in Figure 2. Every VO is dedicated to an application domain (e.g., biology, pathology). It permits to take into account the locality principle of each VO [15]. Within a VO, one peer is selected as a super peer. It acts as a gateway (or a proxy) for other peers, called second level peers, in the other VOs. Gateways communicate with each other through a DHT overlay network. Each of them knows, through the $LH*_{RS}^{P2P}$ routing system, how to interact with all second level peers belonging to the same VO. We may assume that gateway peers are relatively more stable than second level peers. In contrast, gateways establish a structured DHT based overlay when each VO -regrouping second level peers- is structured as an SDDS. This type of architecture could also be advantageous for highly heterogeneous environments but it is not the aim of this paper. Then, we consider here the peers as homogenous. Recall that we have not interesting on the assignment of a joining second level peer to an appropriate gateway, i.e., loads balancing. We defer these issues to future works.

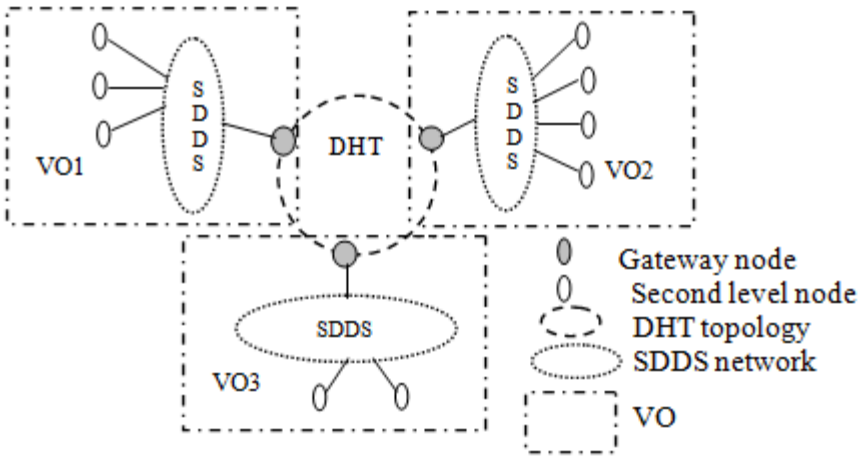

Figure 2: SDDS Based Hierarchical DHT Architecture.

4.1. Resource Discovery Protocol

After describing how VOs are connected, we present the resource discovery process used in the proposed SDDS- HDHT solution. Suppose that a second level peer $p_i \in VO_i$ wants to discover a resource *Res* through a resource discovery query *Q*. Let the peer p_J the peer responsible for *Res*. Let Gp_i the gateway peer responsible for VO_i, Gp_i_list the list of its neighbors in the top level DHT (e.g., the left and right neighbor) and *Response* the metadata of *Res*. Thus, a lookup request for *Res* implies locating the peer responsible for *Res*. Hence, we distinguish two scenarios classifying resource discovery queries:

(i) Peers p_i and p_j belong to the same VO. Then, the query Q corresponds to an intra-VO resource discovery query.

(ii) Peers p_i and p_j are in different VOs. Then, the query Q corresponds to an inter-VO resource discovery query.

4.1.1. Intra-VO resource discovery query: Generally, users often access data in their application domain, i.e. in their VO. In consequence, it is important to search metadata source first in the local VO (VO_i) before searching in all the system (other VOs). This solution favors principle of locality [15]. The intra-domain ontology queries are evaluated according to the routing system of a classic $LH*_{RS}^{P2P}$ routing system which is completely transparent to

the top level DHT. Furthermore, it has proved its efficiency with respect to the scalability and search process. Recall that finding a peer responsible of metadata of the searched resource requires two messages. In this context, the peer p_J sends metadata of Res to p_i.

4.1.2. Inter-VO resource discovery query: [10] proved that a DHT lookup algorithm required only minor adaptations to deal with groups instead of individual peers. Super peer *Ids* (*SPIds*) are viewed as a sequence of digits in a configurable base *b*. In order to make a resource in VO_i visible through the top level DHT, hash join *H* is applied to this resource, when it joins the system, to generate a group identifier *gid* (same prefix with the corresponding gateway). Then, an other hash function *h* is applied to this resource in order to generate a peer identifier *pid*. This permits to associate each resource to its VO [26].

```
Discover (p_i, Res, VO_i)
{     if     (LH*_RS^P2P_Lookup(Res,     VO_i)==1)
         then Send Response to p_i;
    Else forward Q to Gp_i
    if Gp_i do not respond
       then{ consult Gp_i_list;
           Forward Q to neighbours of Gp_i;}
    If  DHT_route(Res)==Gp_J  then
      if (LH*_RS^P2P_Lookup(Res, OV_J)==1)
        then send Response to p_i.
       else Response = {} is sent to  p_i.
    else Response = {}is sent to p_i.
}
```

Figure 3: Metadata Discovery Pseudocode.

Before introducing the resource discovery process, described by the pseudo code in Figure 3, let's recall that we have defined a certain period of time (e.g. Round- Trip Time RTT) as in [31]. The manner in which the RTT values are chosen during lookups can greatly affects performances under churn. [35] has demonstrates that a RTT is a significant component of lookup latency under churn. In fact, requests in peer to peer systems under a churn are frequently sent to a peer that has left the system. At the same time, A DHT rooting has several alternate paths to complete a lookup. This is not the case when a failure concerns the gateway peer. In our solution, a RTT is mainly useful to maximize time to discover resources when a failure occurred in a gateway peer. In this case, p_i do not expect indefinitely. When RTT is exceeded, it considers that Gp_i is failed and consults the gateway neighbours list Gp_i_list received in the connection step. Then, p_i sends its query to one of the peers founded in Gp_i_list. Let's recall that in the connection step of any gateway peer Gp_i, this latter sent its list neighbors Gp_i_list to the nearest second level peer p_0 in its VO. These neighbors are located by using *neighborhood set* parameter we described above. Then, p_0 forwards Gp_i_list to all other second level peers. It is done just on the connection step. When *Res* is not found in VO_i, *Q* constitutes an inter-VO query. The resource discovery process is defined in four steps:

(i) The peer p_i routed the query to the gateway Gp_i. If a Gp_i failure is detected (RTT is elapsed), it requests one neighbor of Gp_i (received during the connection step).

(ii) Once the query reaches a gateway peer Gp_i, a hash function *H* is applied to *Res* in order to discover the gateway peer responsible for the VO that containing *Res*. The query arrives at some gateway peer Gp_J. This is valid whenever a resource, matching the criteria specified in the query, is found in some VO_J.

(iii) Using the $LH^*_{RS}{}^{P2P}$ routing system in the founded VO_J, Gp_J routes the query to the peer $p_J \in VO_J$ that is responsible for Res.

(iv) Finally, Metadata of *Res* are sent to Gp_j which forward it to p_i via the reversing path.

4.2. System Maintenance

When a peer joins/leaves system, the affected data structure on some existing peers must be updated accordingly to reflect the change. The continuous leaving and entering of peers into the system is very common in Grid systems (dynamicity properties). In consequence, updating the system is required. Peer departures can be divided into friendly leaves and peer failures. Friendly leaves enable a peer to notify its overlay neighbors to restructure the topology accordingly. Peer failures possibility is more complex and seriously damages the structure of the overlay with data loss consequences. In structured peer-to-peer systems, such as Pastry [37] used in our system, the connection / disconnection of one peer generates $2B*Log_B(N)$ messages [37] where N is the number of peers in a Pastry system. Furthermore, the maintenance can concern the connection/ disconnection of one or more peers. Throughout this section, we discuss the connection/ disconnection of a gateway peer and second level peers.

4.2.1. Second Level Peer Connection/ Disconnection: The connection/ disconnection of a second level peer p_i do not affect lookups in other peers except the possible split of a bucket if this latter gets overloaded. Let's discuss the only one required maintenance. When p_i joins some VO_i, it asks its neighbor about Gp_i_list. In consequence, only two messages are required. This process avoid that several new arrival peers asked simultaneously the same gateway which can constitute a bottleneck as in SP-HDHT solution. In other terms, when a new second level peer arrives, it searches its gateway (only one) and neighbors of this one. This process permits also to reduce messages comparing to the complex process in the MG-HDHT solution in which the new second level peer should retrieve all gateways.

4.2.2. Gateway Peer Connection/ Disconnection: Throughout this section, we present the connection/ Disconnection steps for gateway peers. We will not discuss the classical maintenance of a DHT [38]. This occurs at the connection / disconnection of a gateway peer. Without any maintenance protocol, a disconnection or a failure of a gateway peer paralyzes access to all second level peers which is responsible for them. Adressing this failure generates additional maintenance cost.

Before describing the maintenance process, let's analyze the connection of a gateway peer Gp_i to VO_i.

(i) Gateway peer Gp_i sent its list neighbors Gp_i_list (the left and right neighbor) to the nearest second level peer p_0 in VO_i.

(ii) Peer p_0 contacts peers in Gp_i_list to inform them about its existence (in order to have an entry to VO_i in the case of Gp_i failure).

(i) Peer p_0 sent this list to all second level peers in VO_i via a multicast message. Recall that other second level peers do not report their existence to neighbors of Gp_i.

Recall also that this process is done just once at the initial connection of Gp_i and only p_0 periodically executes a *Ping/Pong* algorithm with iGp_i. It sends a *Ping* message to Gp_i and

this one answers with a *Pong* message in order to detect any failure in Gp_i. Let us discuss the case of a gateway failure/ update. When Gp_i is replaced by another, the process of maintenance (after the DHT maintenance) is:

(ii) The new gateway GpNew contacts the nearest (only one) second level peer p0 and gives him its neighbor's list GpNew_list.

(iii) Peer p0 inform peers in GpNew_list about its existence. But, it does not inform other second level peers about GpNew_list (lazy update).

A lazy update consists that second level peers, except $p_{0,}$ have not the description of the new gateway peer Gp_{New} and its updated Gp_{New}_list at this moment. When Gp_i does not respond after RTT period, a second level peer consults its old Gp_i_list to reach other VOs. Thus, it rejoins the overlay network (any VO) in spite of a gateway failure. The update of this list is done during the reception of the resource discovery result. Also, a failure of p_0 does not paralyze the system since the new gateway peer always contacts its nearest second level peer. The entry to the VO can also be done through peer p_0 since this one reported its existence in the connection step. This process allow a robust resource discovery process although the presence of dynamicity of peers. This is not the case in MG-HDHT solution when failures of all gateways in some VO paralyze the input/ output to/ from this VO.

Recall also that one of the limitations that our solution suffers from: the failure of both a gateway peer and its neighbors in Gp_i_list. A solution consists on enrich the neighbors list of the any gateway node.

5. Resource Discovery Query Example

Let us give an example which explains an inter-VO resource discovery process. Therefore, let us have four virtual organisations: Protein, Genome, BioMedical and Pathology. Let *R, S, ALZ, U* relations in *VO_Pathologie*. We consider the following SQL query: ***Q : { Select * from ALZ***;}. Supposed that Peer7, considered as a second level peer (SLP) belonging to *VO_BioMedical*, propagates the resource discovery query in order to discover metadata of the searched resource (ALZ relation). Let *Peer1* (resp. *Peer6*) the gateway peer (GP) of Pathology VO (resp. Bio-Medical VO). Suppose also that SLP0 constitute the nearest second level peer to the gateway in each VO. Recall that each VO is structured as an SDDS.

Results of the resource discovery query are metadata of the ALZ relation including the profile of ALZ and its placement including the peer responsible for ALZ (Peer4 on the VO_Pathology in Figure 4). Let's analyze the containing of these metadata. It includes: (i) attributes of ALZ (ii) placement of ALZ and (iii) size of ALZ. Metadata (i) include the name of each attribute, its size, its type, minimal and maximal value and the number of different values when the metadata (ii) include the address IP of each fragment of ALZ, fragmentation of ALZ (horizontally or vertically), duplication of ALZ (eventually) and the construction rule of ALZ.

Since the resource *ALZ* is not present in VO_Bio-Medical, the $LH*_{RS}^{P2P}$ routing system does not return any results. Then, it is clear that Q constitutes an inter-VO query which is redirected to other VOs. Let's now resume the inter-VO metadata *ALZ* discovery steps: *Peer7* sends *Q* to the gateway of its VO. It corresponds to *Peer6*. Through the DHT lookup, the application of hash function *H* to *ALZ* returns the address IP of *Peer1*, the gateway peer in VO_Pathology. It's done through the Routing Data Table (RDT). Since ALZ is not present in the gateway peer, the $LH*_{RS}^{P2P}$ lookup permits to find *Peer4*, the peer responsible for the resource *ALZ*. It kept metadata of *ALZ* and sent it to *Peer 7* via the reversing path.

Let discuss the case of a failure in *Peer7*. Suppose also that a *RTT* is exceeded without any response. We conclude that it was a failure in the gateway *Peer6*. Then, *Peer7* sent Q to one peer in the gateway neighbors list {*Peer42, Peer9*}, received in the connection step. When a result is returned, it is done through *Peer42* or *Peer9* which sent the result to *Peer 7*. There are 2 possibilities: (i) the reversing path if *Peer6* is still down or (ii) through *Peer8* which constitutes the only second level peer having a contact with the new gateway. In this latter case, *Peer8* takes the opportunity to send an update message (new gateway and its neighbors list) to *Peer7*. Hence, the system update is done at the receiving results of metadata discovery. Proceeding in this manner, a failure of Peer6 does not paralyze the system. Thus, a gateway peer communicates with only its nearest second level peer. Dealing with this scenario is different in [49] solution in which complex protocols are used: each leaf peers must initially ask its super peer about the other super peers without any network connection between these leaf peers. When a new super peer joins a VO, all leaf peers must update their Gp_i_list and a network connection is established between each leaf peer and its super peer. Our solution offers another advantage: in the arrival of a new peer (e.g., *Peer10*) in VO_Pathology, it asked only the nearest second level peer without any communication with its gateway as in previous hierarchical DHT solutions. This process is completely transparent to the DHT level.

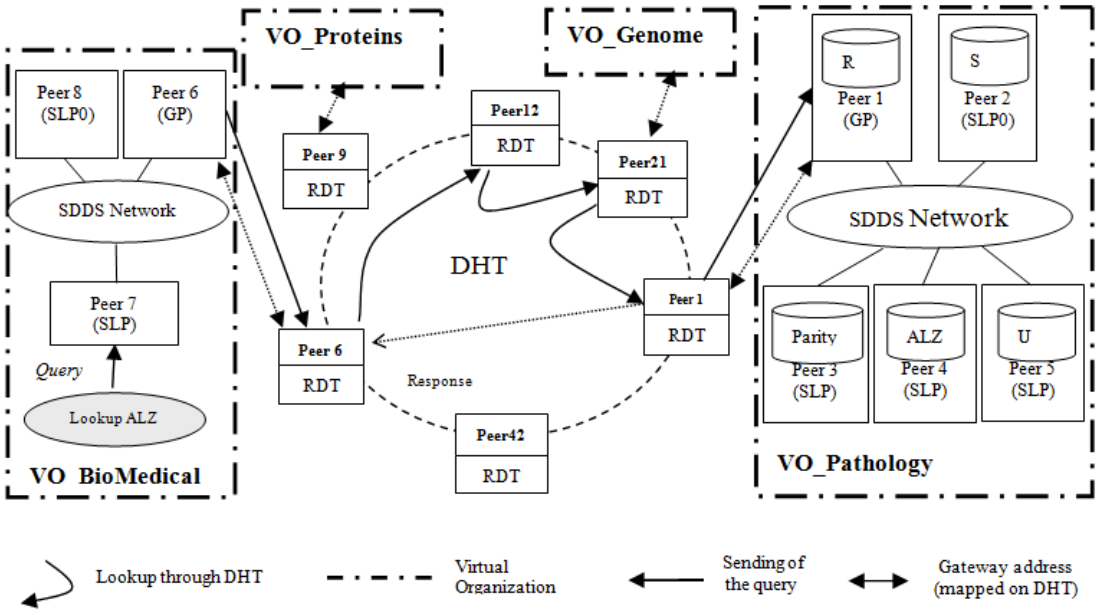

Figure 4: Example of a Resource Discovery Process through the SDDS-HDHT Solution.

6. Cost Based Analysis

Throughout this section, we analyze lookup and maintenance costs generated by our SSDS-HDHT solution. Then, we compare them to cost generated by flat DHT, SP-HDHT and MG-HDHT solutions already discussed in the preliminaries section.

Suppose that a peer $p_i \in VO_i$ wants to discover the peer that is responsible for a resource *Res* through a resource discovery query *Q*. Suppose that this peer is p_J. Let N_G represents the number of VOs in our system. It represents also the number of gateways. Let N_T represents

the total number of peers in the system. In the SDDS-HDHT scheme, the ratio between gateways (super peers) and the total number of peers constitutes an important parameter. We denote it by α and call it the gateway ratio: $N_G = \alpha . N_T$. The total number of second level peers is $N_{SL} = (1-\alpha).N_T$, when N_{SL}/ N_G constitutes a number of second level peers by VO. We discuss the impact of α on each solution. In this analysis, we have interest to the number of messages generated in the system. We assume that every message sent/received by a peer p_i create same cost regardless on the message size. It is clear that a good total cost analysis must deal with a lookup cost Lc and maintenance costs Mc. The total cost is given by: Tc= Lc + Mc.

6.1. Lookup Traffic Costs

We discuss the resource discovery costs in both intra-VO and an inter-VO resource discovery queries. Then, we discuss the impact of α on each solution. Here, we further assume that a system is in a steady state, i.e., no churn occurs and that gateways' DHT overlay is nonempty.

The total lookup traffic cost (Lc) consists of the lookup costs through second level peers Lc_{SL} and lookup cost through gateways Lc_G. Thus, the total lookup cost here is

$$Lc=Lc_{SL}+Lc_G.$$

When Q constitutes an intra VO query, only a lookup through the $LH*_{RS}^{P2P}$ routing system is required. Then, $Lc=Lc_{SL}$. It requires a maximum of two hops when this lookup requires $O(log_B(N_T))$ messages in flat DHT solutions. In consequence, the lookup cost does not depend on the super peer ratio with our solution for this class of queries. It is also the case in the MG-HDHT solutuion but not in the SP-HDHT solution in which leaf nodes must first contact its super peer which forwards the query to the leaf node responsible for the searched resource. This accelerates the lookup but the initialization and maintenance steps are costly. Furthermore, a super peer acts as a centralized resource for other leaf nodes. This is a serious disadvantage especially when we have several simultaneous messages.

Let examine now an inter-VO lookup cost in SDDS-HDHT solution. When the resource *Res* is not found in VO_i, the query is propagated to the gateway Gp_i. Then, the localisation of the gateway responsible for the VO_J containing *Res* requires $Lc_G=O(log_B(N_G))$ hops. After that, another lookup through the $LH*_{RS}^{P2P}$ routing system is required to search metadata of *Res* in VO_J. It requires two additional hops at most. Then, the total lookup cost for an inter-VO resource discovery query is $Lc=O(log_B(N_G))+4$ messages. Comparing to the cost generated in the flat DHT solution, both complexities are logarithmic. But, the difference can be real when $N_G << N_T$. In a system with a reduced gateway ratio, this cost became interesting comparing to flat DHT lookup cost. In SP-HDHT solution, the fact that all leaf nodes in one VO forward their queries to one gateway constitutes a disadvantage as in SDDS-HDHT solution when the using of several gateway peers in MG– HDHT solution put gateways less under stress. Hence, it is clear that the impact of α is real in inter-V0 queries especially with simultaneous messages. In the performance evaluation section, we evaluate capabilities of each solution in the presence of simultaneous resource discovery queries.

6.2. Maintenance Traffic Costs

Due to the dynamicity of peers, manage the departure/ arrival of peers to the system often constitute the major maintenance cost. [50] proved that there is real trade-off between minimizing total network costs and minimizing the added overhead to the system. The

maintenance costs are measured by the number of messages generated to maintain the system. Throughout this section, we discuss maintenance costs generated by the SDDS-HDHT solution. We discuss different scenarios and compare the generated costs in the compared solutions. We do not take into account the recovering process. It is done through a parity calculus in the case of SDDS peers failure. We defer this issue to another future research work since the recovery process is not the aim here. Dealing with our two-level hierarchy architecture, the maintenance cost is the sum of three costs:

(i) The DHT maintenance cost generated when a gateway joins/leaves the system, called Mc_{G1} cost.

(ii) Communication costs between a gateway and its second level peers, called Mc_{G2} cost.

(iii) Messages exchanged between second-level peers, called Mc_{SL} cost.

Dealing with these three costs, the total maintenance cost is: $Mc = Mc_{G1} + Mc_{G2} + Mc_{SL}$.

6.2.1. Gateway Connection/ Disconnection: The continuous leaving and entering of peers into the system, which is very common in Grid systems, generates important routing table maintenance in flat DHT based solutions especially in the case of a great number of peers. The main overhead is due to the periodical refreshing of routing tables, which takes O ($\log^2 N_T$) messages per peer per refreshing in DHT systems. Then, a reduced number of peers running the DHT protocole constitutes a good solution. In SDDS-HDHT solution, only one gateway per VO runs the DHT prorocol. Hence, the DHT maintenance is required only when gateways join/leave the system. This generates $Mc_{G1} = 2B*log_B(N_G)$ messages [37] when this cost is $Mc_{G1} = log^2_B(N_G)$ messages in the Chord based SP-HDHT solution.

Let's examine a new gateway (Gp_{New}) connection cost. It consists that Gp_{New} sent its Gp_i_list to p_0. Then, p_0 reports its existence to gateways in Gp_i_list. Inconsequence, Mc_{G2} is equal to 3 if Gp_i_list is constituted by the left and right neighbours of Gp_{New}. Also, p_0 sent Gp_i_list to all second level peers in VO_i via a multicast message. But, it is done just once at the initial gateway connection (second-level peers used their last Gp_i_list to access the gateways DHT).

In the SP-HDHT solution, any super peer update generates communication with all its leaf peers. Hence, except the connection step, $Mc_{G2} = 2*(N_{SL} / N_G)$ for each super peer update process. Thus, although the good maintenance cost in this solution, costs depend strongly on the ratio between gateways and second leaf peers since each each leaf peer maintains a permanent connection with its super peer. In our solution, just a second level peer p_0 - the nearest peer to Gp_i - periodically executes a *Ping/Pong* algorithm. It sends a *Ping* message to its gateway and this one answers with a *Pong* message in order to detect any failure in the gateway. Then, we can easily remark that Mc_{G2} is significantly more important than the cost generated by the gateway arrival in our solution. Recall that Mc_{G1} is the same for both SP-HDHT and SDDS-HDHT solution.

Let analyze the gateway connection in MG-HDHT solution. Let *n* the average number of gateways between levels. It corresponds to the average number of gateways by VO. Let also N_{SLi} the second level peers number in VO_i. Then, a connection of a new gateway generates $Mc_{G2} = N_{SL} - n - 1$ messages in order to contact all second level peers in VO_i.

6.2.2. Second level Peer Connection/ Disconnection: When a new second level peer joins a VO, it receives the Gp_i_list from the nearest second level peer in the VO without contacting the gateway peer as in [49]. Then, Mc_{SL} corresponds to 2 messages. Recall that the connection step consist that only one peer (p0) contacts the gateway peer.

Let's analyze this cost in SP-HDHT solution: when a new leaf peer joins some VO_i, it contacts Gp_i to have Gp_i_list. In consequence, $Mc_{SL}= 2$ messages. But, The connection step is more complex and generates $2*(N_{SL} / N_G)$ messages between leaf peers and a same super peer. Then, each leaf node reports its existence to (at least one) neighbour of Gpi. This generates (N_{SL} / N_G) messages. Furthermore, each leaf peer periodically runs (every T_{PING} second) a simple Ping/Pong algorithm in order to detect any super peer failure. It also stores a list containing other available super peers in Chord. Each super peer also periodically runs a 'Stabilize', 'Fix Finger' and 'Republish' algorithms consisting to update the super peers' Chord overlay and store leaf nodes resource information [49].

Let's analyze the connection cost of a second-level peer to VO_i in the MG-HDHT solution: First, it should retrieve all gateways by asking its neighbor which also asked its neighbors. This process is repeated successively until having all gateways in VO_i ($N_{SLi} -1$ messages). Then, it contacts all gateways in its VO. In consequence, $Mc_{SL}= N_{SLi} -1+n$.

In summary, the SDDS-HDHT solution presents several advantages. The lookup complexity to discover any resource is improved especially for intra-VO lookups. This process is completely transparent to the top level DHT lookup. Mainly, our solution handles high churn rates and reduces maintenance costs. Except the connection step, a gateway communicates with only one second level peer in its VO. Other second level peers update their gateway neighbor's list during the reception of the resource discovery result. The protocol permits a good management of peers in spite of gateway failures. It can also be applied to other DHTs like Skip graphs [2] and eCAN (conceptually a hierarchical version of CAN) [45]. In the next section, we evaluate the efficiency of the proposed resource discovery solution.

7. Performance Evaluation

We based on a virtual network as a thousand peers to prove the efficiency of our solution in large Grid networks. We deal with a simulated environment since it is difficult to experiment thousands of peers organized as virtual organization in a real existing platform as Grid'5000 [GRI]. To ease this evaluation, we based our experiments on a platform having four features: (i) emulation of peers, ii) emulation of network, (iii) using FreePastry [9], one implementation of the Pastry DHT and (iiii) $LH*_{RS}^{P2P}$ SDDS prototype implemented by Litwin's team in Dauphine University [4. We simulate performances of (a) a flat DHT solution in which all peers run a same DHT protocol in order to measure the benefits the hierarchical system, (b) SP-HDHT solution in which gateways establish a structured DHT overlay network when each leaf peers maintains a connection to its gateways, (c) MG-HDHT solution in which several gateways are maintained between hierarchical levels and (d) the proposed SDDS-HDHT solution. Then, we compare theirs performances. Throughout this section, we deal with three classes of experiments:

(i) Lookup performances experiments in which we interest to the hops number and elapsed times.

(ii) Maintenance overhead experiments in which we simulate a join/leave peers scenario and interest to the required update messages. We deal with connection/disconnection of both gateway and second level peers.

(iv) We also interest on the optimal ratio between peers at the top level and peers at the lower level in order to evaluate the impact of the gateway ratio.

7.1. Simulation Environment

Several homogenous peers run in a single Java Virtual Machine (JVM) on top of FreePastry. Each virtual peer run as process and uses an assigned virtual IP address. The network topology of internet is emulated in LAN by using 'Traffic Control', again provided in Linux. We have simulated homogenous bandwidth networks and local network *100 Mb/s*. We also implemented an interface API (Application and Programming Interface) to allow the request of SDDS components from FreePastry prototype. Our environment is constituted of a Windows computer (processor speed 2.8 GHZ, Memory: 3GB, cache 1024 KB). Programs are implemented in Java 5.0. We also fixed the number of resources in VOs. It can be equal to $5*(log_2 N_T)\hat{}2$ which corresponds to a logarithmic distribution. Variables used bellows are defined as follows: N_T is the number of peers in the system, N_G the number of super peers, α the super peer ratio and B=2 the base used for the *gid* in the Pastry system.

We simulate overlay network with 10000 homogenous virtual peers $\{p_{0,}...,p_{9999}\}$ in which {10, 100, 1000, 2500, 10000} peers are super peers with respectively {1000, 100, 10, 4, 0} leaf peers for each super peer. But, the total number of peers stays appreciatively constant (N_T=10000). An example is to discover metadata of a database relation *R* referenced in some SQL query *Q: Select * From R*. Recall also that the key of the discovered data source corresponds to the relation name. Recall also that we have used same principles of PAST [6] to have persistent storage of resources metadata in each peer. For the detection of failed peers, we set a RTT time to 1 sec.

7.2. Simulation Analysis and Comparisons

In this section, we experiment with flat DHT, SP-HDHT and MG-HDHT solutions. Then, we compare their performances to the SDDS-HDHT performances. We evaluate performances of both intra-VO and inter-VO resource discovery queries. We interest to the impact of simultaneous messages and analyze the impact of α □ □in each solution. In following experiments, our response time includes the query processing (matching of metadata describing the resources) and communication costs. It corresponds to the amount of time that elapses between the generation of query and the reception of a response.

7.2.1. Lookup Resource Queries: First experiments simulate a flat DHT solution in which all peers run a DHT protocol. Thus, specify the equivalence between such systems and SDDS-DHT systems when N_T/N_G=1.

When we nalyze the hops number required to discover one resource in both solutions, our results are always better when it concerns an intra-VO resource discovery query. In fact, $LH*_{RS}^{P2P}$ lookup requires a maximum of two (2) messages when this number is always $log_B(N_T)$ in flat DHT solutions. For inter-Vo queries, we have showed in last section that the theoretically worse case corresponds to $O(log_B(N_G))+4$ hops with SDDS-HDHT scheme. By a simple calculation, we deduce that flat DHT performances are better when our DHT overlay is composed by more than 1000 gateways. In other terms, from 10 leaf peers/VO (α<1%), our results are better. This is due to the fact that adding new second level peers do not influences $LH*_{RS}^{P2P}$ lookup performances.

However, these results correspond to theoretical numbers of hops for only one resource discovery query. In the case of simultaneous resource discovery messages, the results should take into account that all messages are forward to the same gateway (in one VO). This generates some congestion in this peer. To confirm this, we have experiment systems with (i) 2000 gateways (5 leaf peers/ VO, α=20%) and (ii) 500 gateways (20 leaf peers/VO, α=5%).

We also interest to the number of simultaneous resource discovery queries. It is useful since it shows if the SDDS-HDHT solution is also scalable in the presence of high number of messages.

Figure 5 shows elapsed response times for resource discovery queries (intra and inter-VO queries). It confirms that our performances are always better when queries constitute intra-VO resource discovery queries. Elapsed response times are 50% better than flat DHT solution. This is due to the reason mentioned above. Let analyze performances of inter-VO queries. When we experiment with α=20%, performances are almost similar for a reduced simultaneous discovery queries. But, elapsed responses time increase from 20 queries/sec. It is due to the fact that all queries transit by the same gateway in each VO. However, a great leaf peers number (α=5%) improves significantly our performances which are better. The save is close 10% compared to the flat DHT solution in spite of the simultaneous messages. It provides from the gain in the DHT lookup. In fact, the probability to find the searched resource in a local VO is greater.

We have also compared our results to both SP-HDHT and MG-HDHT results. [49] proved that best performances are obtained with small number of gateways. We simulate a network with 100 VOs (100 level peers/ VO, α=1%). We also deal with 10 gateways/ VO in the MG-HDHT solution. Figure 6 shows that the SP-HDHT solution is slightly better for intra-VO queries when less simultaneous messages are used. From 70 messages/ second, our solution is 10% better than SP-HDHT solution. We explain this by the fact that intra-VO lookups are done without any gateway peer intervention when a bottleneck is generated in each gateway in the compared SP-HDHT solution. This is the reasons why the simultaneous messages influenced significantly the SP-HDHT results. We remark that the average response time is almost constant when we have several simultaneous messages in both SDDS- HDHT and MG-HDHT solution. We conclude that the save can be better if we experiment with great number of simultaneous resource discovery queries. Note also that these experiments do not include the more costly connection step.

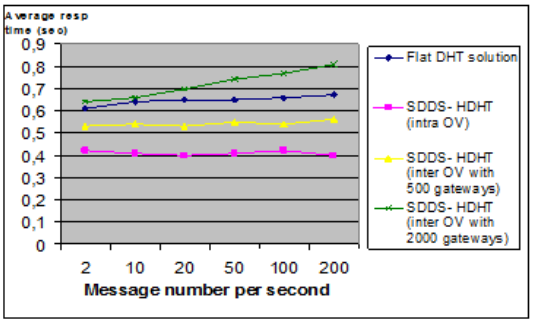

Figure 5: SDDS - HDHT performances vs. Flat DHT performances

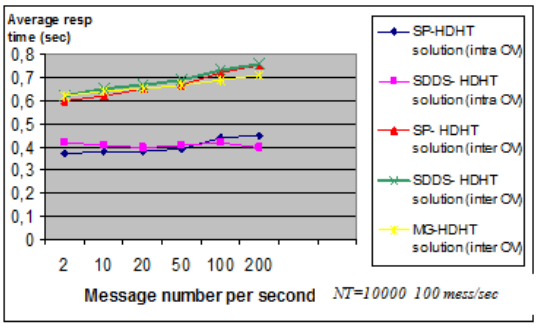

Figure 6: SDDS- HDHT performances vs. SP-HDHT performances.

For inter-V0 queries, simultaneous resource discovery queries influences performances of both solutions. Bottleneck is generated since all queries transit by the same gateway peer which increases response times in SP-HDHT and SDDS-HDHT solutions. Then, SP-HDHT results are slightly better when we have less than 70 messages per second. From this value, results are almost close for the two solutions with slight advantage to SDDS-HDHT solution since intra-VO queries always precede inter-VO queries. We conclude that in inter-VO queries, we have dependence between performances and simultaneous queries for these two

solutions. The same impact is observed with a reduced gateway ratio α. In the other hand, performances of MG-HDHT solution are better (rate of 5%) especially for high simultaneous messages since queries are propagated through the several gateways in the same VO.

7.2.2. Maintenance Overhead Evaluation: We measure the impact of the join/ leave peers in the system. We interest to the total messages number required when a peer joins/leaves the network. We tabulate churn in an event-based simulator which processes transitions in state (*down*, *available*, and *in use*) for each peer as in [12]. We simulate a churn phase in which several peers join and leave the system but the total number of peers N_T stays appreciatively constant. The maintenance costs are measured by the number of messages generated to maintain the system when peers join/leave the system.

Impact of peers Connection/ Disconnection in the System Maintenance. Lets a system with a peers distribution as {N_G=100, N_{SL}/N_G=100 peers/ VO}. This configuration corresponds to average results in inter-VO discovery queries performances. In these experiments, when a number of new connections/ disconnections exceed 20 peers, 10% of them concern gateway peers. Figure 7 shows the impact of peers connection/ disconnection in the total messages number in the system. Flat DHT solution generates the greater number of messages in the connection /disconnection of peers. Compared to our solution, the messages number ratio is 1.1 (resp 4.5) for the connection of one leaf peer (resp 100 peers). It is clear that maintaining a flat DHT generates greatest costs especially when several peers join/leave the system. When a gateway join/leave the system in our solution, it generates $2BLog_B(N_G)$ messages. It corresponds to only two messagse for a connection of a second level peer and three messages for a connection of a new gateway without any update in the gateway's DHT.

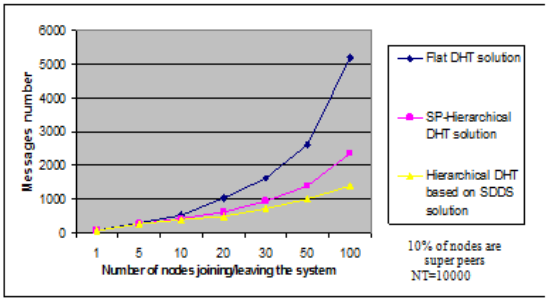

Figure 7: Impact of the connection/ disconnection of nodes in the messages number exchanged in the system.

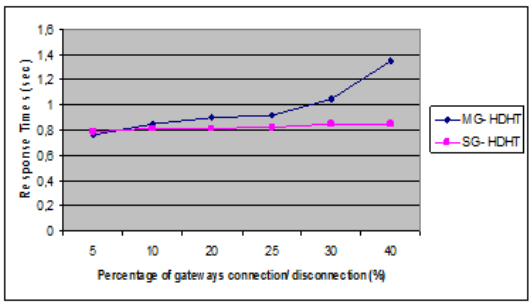

Figure 8: Impact of the percentage of the gateways connection/ disconnection in the total response

We compare these results to the SP-HDHT performances. The numbers of update messages are closes when we have only second level peers connections/disconnections. It corresponds to the case when less than 10 peers joining the system (Figure7). In fact, all new peers must contact their super peer in SP-HDHT solution. Increasing the number of connection/ disconection of second level peers can generates a bottleneck. Our solution offers a significant maintenance cost gain when the update occurs in gateways. As the number of gateways connection increase as the gain is important since the required update messages is less with our solution. The save is 59% for the connection of 90 leaf peers and 10 gateways. Certainly, update DHT messages concern both solutions. But, in the SP-HDHT solution experiments, the new gateway establishes connections with all its leaf nodes. It is also the case in the MG- HDHT solution. The fact that new second level peers in MG-HDHT must contact several gateways generates additional messages. It is not the case in our solution. A

new second level peer contacts only its neighbour and the connection of a new gateway generates only two additional messages.

We also experiment the impact of the percentage of the gateways arrival/ departure in the total response time as shown in Figure 8. It corresponds to resource discovery process under a high churn. When only 5% of gateways are replaced by other gateways (gateways disconnection followed by connections of other gateways), MG- HDHT solution have slightly better results than SDDS- HDHT performances. However, when this percentage increases, SDDS-HDHT performances remain stable since second level peers used the gateway neighbor's list to reach other gateways in the DHT when second level nodes in MG-HDHT solution used other not failed gateways in the same VO pending the update of the new gateways. From 25% gateways connection/ disconnection in the system, MG-HDHT curve increase significantly. Recall that we have deliberately ensured that not all gateways in the same VO are failed in MG-HDHT solution. Otherwise, a second level peer in some VO_i will be not able to contact any gateway of other VO_j ($i \neq j$) until. It is not the case in our solution in which second level peers can use neighbors of its gateways. But, recognize that if all peers in the Gp_i_list failed, consequences are also the same as above.

Impact of the Gateway Session's Length in the System Maintenance. We also experiment with the mean session duration of peers. We define a session length in our system as follow: we define an observation interval as T (equal to 5 sec). Each session length is represented as a percentage of T and corresponds to the duration in which the peer stays in the network, and then leaves the network. It corresponds to values {5%, 10%,…, 100%} in Figure 8. Connections/ disconnection concerns only gateway peers. Simultaneous inter-VO queries (100 messages/ sec) are sent at the beginning of T. Then, disconnections of $10\%*N_G$ gateways occur. The concerned gateways are soon replaced by other gateways. The result collection is done when the duration T is finished. Figures 9 shows response times corresponding to the resource discovery queries when we vary the session length.

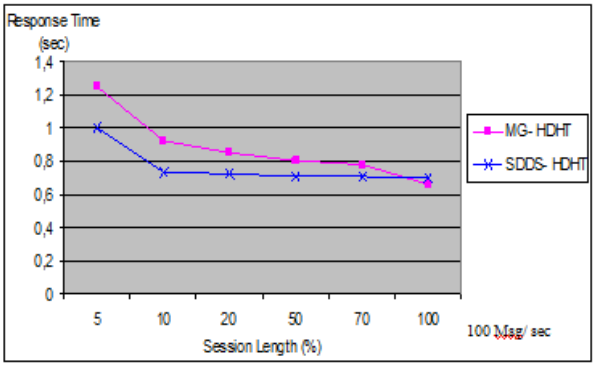

Figure 9: Impact of the gateway session's length in the total response time.

When the session length represent only 5% of T, SDDS- HDHT and SP-HDHT performances are better. It is due to the fact that second level peers in SDDS- HDHT (as in SP-HDHT solution) solution consults its Gp_i_list to find another gateway without any communication with its new gateway when each second level peer in MG-HDHT used another available gateway in its VO more to update its gateway list through a communication with the new gateway. As the session length is important as the gap between MG-HDHT and SDDS-HDHT is less important. It is due the fact that most gateways in MG-HDHT solution are available when the queries occur. Finally, when the session length is equal to the observed

time, no connection/ disconnection of gateways occur during the process of discovery queries. Then, performances of MG-HDHT are better. It is due to the fact that we have several gateways which can achieve the several messages with gateways less under stress.

Impact of the Neighbors Gateway List in the System Maintenance. We have also interested to the size of the list Gp_i_list containing neighbors of the gateway Gp_i in a VO_i. This list is propagated to all second-level peers in VO_i during the connection step. When a gateway fails or changes, this list can be updated when a peer received the resource discovery result. Hence, the enrichment of each second level peer by maintaining this increases the storage size in each second level peer. The storage space needed to store these lists is relatively small. Only 1KB is required to store one Gp_i_list. Also, if we suppose that the size of resource discovery query answer according to Pastry protocol is 2KB and the network bandwidth is 80 KB/s, the time needed to send this answer from a gateway peer to a second-level peer is 25ms. By adding the Gp_list (1KB) to the resource discovery response, the time needed to send this answer is about 38ms. Hence, the maintaining message number gaining by the using of our method is of some hundreds of messages.

7.3. Optimal Hierarchical DHT Configuration

Through these experiments, our goal is to determine optimal configurations on the three considered hierarchical DHT solutions. It is easy to see that the configurations are fully determined by specifying the fraction of peers in the top level of the hierarchy. Following experiments show lookup performances of simultaneous inter-VO resource discovery queries (100 messages/ sec) with different configurations corresponding to different gateway ratios {0.1%, 1%, 5%, 10%,…, 100%}. But, the total number of peers stays always constant and equal to 10000 peers. Since the maintenance costs constitute an important part in the total resource discovery costs, we have deal with two classes of experiments:

(i) Experiments without any peer arrival / departure to the system (Figure 10).

(ii) Experiments including the arrival/ departure of peers to the system (Figure 11).

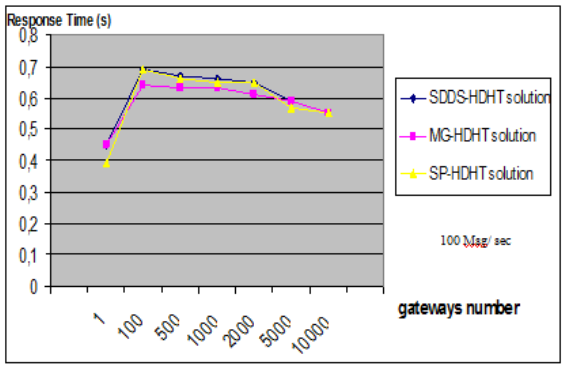

Figure 10: Impact of α in SP-HDHT, MG-HDHT and SDDS- HDHT performances (without maintenance costs).

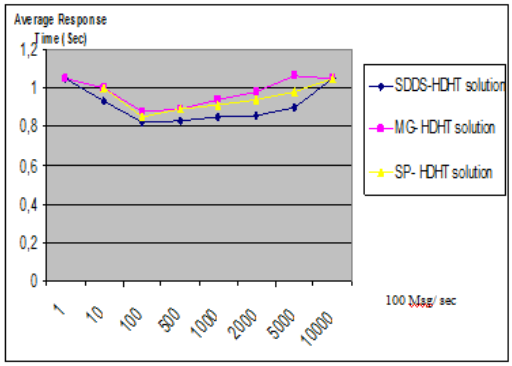

Figure 11: Impact of α in SP-HDHT, MG-HDHT and SDDS-HDHT performances (with maintenance costs).

In Figure 10, experiment with only one gateway in SDDS-HDHT (resp. MG-HDHT) solution corresponds to a flat SDDS (resp. flat DHT) overlay. In this case, the using of only $LH*_{RS}^{P2P}$ routing system generates the lowest traffic costs. Also, a centralized overlay network with only one super peer in the SP-HDHT solution generates even less messages

than other solutions. The reason is that only lookup and *Ping/ Pong* messages are exchanged between the super peer and its leaf nodes. In fact, the only one super peer receives the resource discovery query from its leaf nodes. Then, it transmits the result via the reveres path. Same performances are obtained with the configuration (α=100% i.e. N_G=10000) which correspond to a flat DHT. When the number of gateways increases (N_G>1), we notice increased lookup costs for the three compared solution. This cost is most important in SDDS-HDHT and SP- HDHT solution, mostly caused by the bottleneck in the only one gateway. Indeed, it is due to the fact that all queries transit by the same gateway when the several gateways are less in stress on the MG- HDHT solution. This cost decrease from α=20% in the SP-HDHT and SDDS- HDHT solutions. It is from α=10% in the MG-HDHT solution. We conclude that MG-HDHT solution constitutes the better solution when we have not or very little departures/ arrivals of peers in the system. Good performances, obtained with α=10%, constitute also an advantage since it is close to real grid systems with several VOs, each having many second level peers. However, we do not claim that these values are optimal (valid for this configuration).

Figure 11 shows same experiments taking into account maintenance costs. We simulate departures/ arrivals of peers in the system. It concern both gateways and second level peers (10% of gateways and 10% of second level peers in each VO). We have not traced the failure of the only one gateway peer in the SP-HDHT solution since it constitutes a single point of failure when the arrival/ departure of 1000 peers in the MG-HDHT solution generates the most important maintenance. When N_G is equal to 10, performances of SP-HDHT and SDDS-HDHT are equal since the departure/ arrival of 100 second level peers in each VO generates the same maintenance cost. A Second level peer in these solutions used their Gp_i_list when it searchs another gateway in MG- HDHT solution. A configuration with α=1% generates the best performances. The departure/ arrival of 10 gateways and 10 second level peers generates less average response times. Hence, maintenance cost of the SDDS- HDHT is less since it concerns only the DHT update without any connection between gateways and their second level peers. Until this value of α=1%, the major maintenance cost is caused by the departure/ arrival of second level peers. From this value of α, as the gateway number increases as this cost increases but it is less with the SDDS- HDHT solution for the same reason we have cited above. The major maintenance cost is caused by the departure/ arrival of gateways. We remark that the maintenance cost of the MG-HDHT solution is the most important since each gateway inform all its second level peers in each arrival/ departure. It is also the case with the SP-HDHT solution. It is not the case of SDDS-HDHT solution. It is due to the fact that second level peers used a lazy update to update their neighbor's gateway list (when returning the resource discovery result). Finally, all solutions have the same maintenance cost when all peers run a DHT overlay network (N_G= 10000).

We conclude that for each particular value of α (between 1 and 50%), the SDDS-HDHT solution generates the lowest total cost. It is valuable for the case when the major maintenance cost is generated by the departure/ arrival of second level peers but also for the case when departures/ arrivals of gateways constitute the major maintenance cost. The best results are obtained with $\alpha \in$ [1%, 20%] which is close to real grid systems with several VOs.

8. Conclusion and Future Work

We have proposed a hierarchical DHT solution for a resource discovery in data Grid systems. It deals with both the reduction of lookup costs and the managing of churn while minimizing additional overhead to the system. It also takes into account the content/path

locality of organizations in Grids. Our solution combines DHT systems to scalable distributed data structures (SDDS) in its $LH*_{RS}^{P2P}$ variant. The first contribution of our solution is the improvement of the lookup query complexity to discover metadata of any data source. This is valid especially for intra-VO queries since these queries are transparent to the top level DHT lookup. In consequence, this process neither depends on the gateway ratio nor puts gateways more under stress. Furthermore, only the arrival of a new VO requires the DHT maintenance. The high maintenance cost generated by the continuous joining/ leaving of peers is avoided by adopting a lazy update. Our solution addresses also super peer problems as a single point of failure by using a minimum of messages. The performance evaluation of SDDS-HDHT solution proved the reduction of resource discovery costs for both intra and inter-VOs queries. Also, it provides a significant maintenance cost reduction of such system especially when peers frequently join/ leave the system.

Our method can be useful in large scale grid environment since our solution generates less traffic network. Further work includes more performance studies in more realistic large grid environments with a high number of peers. Also, we would like to study the effects of alternate routing table neighbours as in [51]. Also, we would like include more realistic models of churn in our future work as to scale traces of sessions times [8] collected from deployed networks to produce a range of churn rates with a more realistic distribution.

References

[1] M.S. Artigas, P. García and A. F. Skarmeta. “Deca: A Hierarchical Framework for Decentralized Aggregation in DHTs”. Lecture Notes in Computer Science, 2006, Volume 4269/2006, 246-257.

[2] J. Aspnes, G. Shah. « Skip Graphs ». Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2003), Society for Industrial and Applied Mathematics, 2003, pp. 384–393.

[3] M. Castro M, M. Costa M & A. Rowston. “Peer to Peer Overlays: structured, unstructured or both”. Technical report of Microsoft Research MSR-TR-2004-73. Microsoft Research, Cambridge, UK.

[4] http://ceria.dauphine.fr.

[5] R. Devine. “Design and Implementation of DDH: A Distributed Dynamic Hashing Algorithm”. Proceeding of 4th International Conference on Foundation of Data Organization and Algorithms. 1993.

[6] P. Druschel and A. Rowstron, "PAST: A large-scale, persistent peer-to-peer storage utility", HotOS VIII, Germany, May 2001.

[7] I. Foster (editor), Berry D, Djaoui A, Grimshaw A, Horn B, Kishimoto H (editor), Maciel F, Savva A, Siebenlist F, Subramania R, Treadwell J, Von Reich J: The Open Grid Services Architecture, V 1.0. July 2004. Global Grid Forum.

[8] T. Fei, S. Tao, L. Gao, and R. Guerin. How to select a good alternate path in large peer-to-peer systems? In Proceeding of the int. conf. IEEE INFOCOM 2006.

[9] http://Freepastry.org/FreePastry/.

[10] L. Garces-Erice, E. W. Biersack, K. W. Ross, P. A. Felber, and G. Urvoy-Keller. Hierarchical Peer to Peer Systems. In Proceedings of ACM/IFIP Intern. Conf. on Parallel and Distributed Computing (Euro-Par), 2003.

[11] P. Ganesan, K. Gummadi, and H. Garcia-Molina. Canon in g major: designing DHTs with hierarchical structure. In Proc of 24th Intern. Conf. on Distributed Computing Systems, pp 263–272, 2004.

[12] P. B. Godfrey, S. Shenker, and I. Stoica. “Minimizing Churn in Distributed Systems. Proc. of the Int. Conf. on Applications”, Technologies, architectures, and protocols for computer communications pp 147–158, SIGCOMM. Italy 2006.

[13] Gnutella Ptotocol Devlopement. http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html 11. GRID’5000. www.grid5000.org

[14] Gupta I & al. “Kelips: Building an Efficient and Stable P2P DHT through Increased Memory and Background Overhead”. Lecture notes in computer science, 2003. Springer.

[15] N Harvey & al. “Skipnet: A Scalable Overlay Network with Practical Locality Properties”. In Proc of USITIS 2003, Seattle, WA.

[16] A Hameurlain. “Data Management in Grid & P2P Systems”. Intern .Jour. of Computer Systems Science and Engineering, CRL Publishing, Leicester - UK, Vol. 23 N. 2, 2008

[17] A Hameurlain. "Evolution of Query Optimization Methods: From Centralized Database Systems to Data Grid Systems". Proceeding of Intern. Conf. on Database and Expert Systems Applications (DEXA'09), Springer-Verlag, p. 460-470 Linz - Austria, September 2009.

[18] A Hameurlain, F Morvan, M El Samad. "Large Scale Data Management in Grid Systems: a Survey". In IEEE Intern. Conf. on Information and Communication Technologies: from Theory to Applications (ICTTA 2008), IEEE, April 08, pp 1–6.

[19] A. Iamnitchi, I. Foster, "A peer-to-peer approach to resource location in grid environments", Proceedings of the 11th Symposium on High Performance Distributed Computing, Edinburgh, UK, August 02.

[20] Y Joung, J-C Wang. "Chord^2: A two-layer Chord for reducing Maintenance Overhead via Heterogeneity". Computer Networks, vol. 51, no. 3, pp. 712–731, 2007.

[21] Kazaa. http://www.kazaa.com/.

[22] I. Ketata, R. Mokadem, F. Morvan, "Biomedical Resource Discovery considering Semantic Heterogeneity in Data Grid Environments". INTECH Sao Carlos, Brazil (2011)

[23] W. Litwin. "Linear hashing: A new tool for file and table addressing". VLDB 1980. Reprinted in Readings in Database Systems, Stonebreaker ed, 2nd Ed, Morgan Kaufmann, 1995.

[24]. W. Litwin, R. Moussa and T. Schwarz. "LH*rs A Highly Available Scalable Distributed Data Structure". ACM-TODS, Sept 2005.

[25] A. Montresor, "A Robust Protocol for Building Superpeer Overlay Topologies," in IEEE International Conference on Peer-to-Peer Computing (P2P 2004).

[26] I. Martinez-Yelmo, R. Cuevas Rumín, C. Guerrero, A. Mauthe. "Routing Performance in a Hierarchical DHT-based Overlay Network". Proceeding of the 6th IEEE Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2008), 508-515. February 2008, Toulouse, France.

[27] A. Mislove and P. Druschel. "Providing Administrative Control and Autonomy in Structured Overlays". In Proceedings of IPTPS'04, pp 162- 172. San Diego, CA, February 2004.

[28] E. Meshkova & all. "A survey on Resource Discovery Mechanisms, Peer to Peer and Service Discovery Frameworks Computer Networks", in Science Direct. Elsevier 2008 (2097- 2128).

[29] R Mokadem, W Litwin. "String-Matching and Update through Algebraic Signatures in Scalable Distributed Data Structures". Proceeding of the IEEE Inter. Conf. on P2P Data Management, Security and Trust, pp 708-711 (PDMST 2006), Krakow.

[30] R. Mokadem , A. Hameurlain, A. Min Tjoa, "Resource Discovery Service while Minimizing Maintenance Overhead in Hierarchical DHT Systems". In International Conference on Information Integration and Web-based Applications & Services (iiWAS'10), France, Paris (2010)

[31]. Mastroianni C., Talia D. and Verta O. "A Super Peer Model for Building Resource Discovery Services in Grids: Design and Simulation Analysis". Future Generation Computer Systems. 2005. Elsevier.

[32]. Mastroianni C., Talia D. and Verta O. "Evaluating Resource Discovery Protocols for Hierarchical and Super-Peer Grid Information Systems". 19th Euromicro Intern. Conf. on Parallel, Distributed and Network-Based Processing (PDP'07).

[33].E. Pacitti, P Valduriez & M Mattosso. "Grid data management: Open Problems and News Issues"; In Intl. Journal Grid Computing. Springer, 2007, Vol. 5, pp. 273-281.

[34] Ratnasamy & al. "A Scalable Content-adressable Network". Proc. of the ACM SIGCOMM'01 conference on applications, technologies, architectures and protocols for computer communication.2001, pp161-172.

[35]. S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling churn in a DHT". in Proceedings of the General Track : 2004 Usenix Annual Technical Conference, Boston, USA.

[36] Ranjan Rajiv & all. "Peer to Peer Based Resource Discovery in Global Grids: a Tutorial". IEEE communication surveys. 2 nd Quarter 2008, Volume 10, No 2.

[37] A. Rowston & P. Druschel. "Pastry: Scalable Distributed object location and routing for large-scale peer-to-peer systems". Proceeding of the 18th IFIP/ACM international conference on Distributed Systems Platforms. Vol 2218, 2001, pp 329-350.

[38] I. Stoica, Morris, Karger, Kaashoek, Balakrishma. CHORD : A scalable Peer to Peer Lookup Service for Internet Application. SIGCOMM'O, August 27-31, 2001, San Diego, California USA

[39] M. S´anchez-Artigas, P. Garc´ya, J. Pujol, and A. G. Skarmeta, "Cyclone: A Novel Design Schema for Hierarchical DHTs," in IEEE International Conference on Peer-to-Peer Computing (P2P 2005).

[40] M. El Samad, F. Morvan, A. Hameurlain. "Resource Discovery for Query Processing in Data Grids". Proceeding of the Inter. Conf. ISCA PDCCS. pp 59-66, 2009.

[41] D. Talia, P. Trunfio, "Peer-to-Peer protocols and Grid services for resource discovery on Grids", in: L. Grandinetti (Ed.), Grid Computing: The New Frontier of High Performance Computing, in: Advances in Parallel Computing, vol. 14, Elsevier Science, 2005.

[42] P. Trunfio, D Talia, H Papadakid, P Fragoupoulou, M mordachini, M Penanen, P Popov, V Valssov and S Haridi. "Peer-to-Peer resource discovery in Grids: Models and systems", Future Generation Computer Systems (2007).

[43] P. Valduriez P & E. Pacitti. "Data Management in Large-Scale P2P Systems". VECPAR 2004. M Daydé & al. (eds). LNCS 3402. pp 104-118. Springer-Verlag. Heidelberg 2005.

[44]. Z. Xu, R. Min, and Y. Hu. "HIERAS: a DHT Based Hierarchical P2P Routing Algorithm". Proceedings of Intern. Conf on Parallel Processing (ICPP'03), pp 187– 194, 2003.

[45] Z. Xu, Z. Zhang. « Building low-maintenance expressways for P2P systems ». Technical Report HPL-2002-41, Hewlett- Packard Laboratories, Palo Alto, March 2002.

[46] The Web Services Resource Framework, http:\\www.globus.org/wsrf.

[47] H. Yakouben, W. Litwin, T. Schwarz. "LH*$_{RS}^{P2P}$: A Scalable Distributed Data Structure For the P2P Environment". 8th Int conference on new technologies of Distributed Systems. Lyon'08. France.

[48] B. Yang and H. Garcia-Molina, "Designing a Super-Peer Network". Proc. of intern. conf. on Data Engineering (ICDE), Bangalore, India, 2003.

[49] S. Zöls, Z Despotovic, W Kellerer. "Cost-Based Analysis of Hierarchical DHT Design". Sixth IEEE International Conference on Peer-to-Peer Computing (P2P 2006). Cambridge, United Kingdom. IEEE Computer Society 2006 pp 233-239.

[50] S. Zöls, Q. Hofstatter, Z. Despotovic, W. Kellerer. "Achieving and maintaining Cost-Optimal Operation of a Hierarchical DHT System". Proceeding of the IEEE Inter. Conf. on communication ICC 2009, Germany.

[51] B. Zhao, Kobiatowicz and AD. Joseph. "Tapestry: A resilient global –scale overlay for service deployment". IEEE journal on selected Areas in communications 22 vol 1, p 41-53. 2004.