SDDS based Hierarchical DHT Systems for an Efficient Resource Discovery in Data Grid Systems

Riad Mokadem, Abdelkader Hameurlain, Franck Morvan

Institut de Recherche en Informatique de Toulouse (IRIT)
118, route de Narbonne, Toulouse, France
{mokadem, hameur, morvan}@irit.fr

Abstract. Despite hierarchical Distributed Hash Table (DHT) systems have addressed flat overlay system problems, most of the existing solutions add a significant overhead to large scale systems. In this paper, we propose a hierarchical DHT solution based on scalable distributed data structures (SDDS) for an efficient data sources 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. The proposed solution offers good performances especially for intra-VO resource discovery queries since they are completely transparent to the top level DHT lookups. The analysis results proved significant system maintenance save especially when nodes join/ leave the system.

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

1 Introduction

A resource discovery consists to discover resources (e.g., computers, data) that are needed to perform distributed applications in large scale environments [21]. It constitutes an important step in a query evaluation in such environments. Throughout this paper, we focus on the discovery of metadata describing data sources in data Grid systems.

Several research works have adopted the Peer-to-Peer solutions to deal with resource discovery in Grid systems [19] and [26]. P2P routing algorithms have been classified as structured or unstructured [27]. Although the good fault tolerance properties in P2P unstructured systems (e.g., KaZaa [13]), 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 nor the locality principle, a crucial consideration in Grids [10]. Moreover, typical structured P2P systems as Chord [25] and Pastry [24] 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 [21] 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 is more effective in very large Grids [30]. In this context, several research works [5], [6], [12], [17], [18], [20] and [31] 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 [5]. In this context, a Grid can be viewed as a network composed of several, proprietary Grids, virtual organizations (VO) [18] 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 for reducing maintenance costs, have also appeared in the literature [7], [9], [14], [16], [23] and [32]. Despite a good strategy to manage a churn in [14] through a lazy update of the network access points, inter-organizations lookups were expensive because of the complex addressing system. [16] proposed the SG-1 algorithm, based on the information exchange between super peers through a gossip protocol [1], to find the optimal number of super peers in order to reduce maintenance costs. However, most of these solutions add significant load at some peers which generates an additional overhead to large scale systems.

In this paper, we propose a scalable distributed data structure (SDDS) based Hierarchical DHT solution (SDDS- HDHT) for an efficient resource discovery in data Grids. It combines SDDS routing scheme [15] with DHT systems and aims to improve both lookup and maintenance costs while minimizing the overhead added to the system. Our 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 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. [29] noted numerous similarities between Chord and the best known SDDS scheme: LH* (Linear Hashing) [15]. 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 [29]. In fact, super peers are not concerned by intra-VO queries unlike previous solutions as [31] 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. The proposed solution takes also into account the continuous leaving and joining of nodes into the system (dynamicity properties of Grid environments). 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 high maintenance costs [14].

A simulation analysis evaluates performances of the proposed 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. The rest of the paper is structured as follows. Section 2 recalls hierarchical DHT and SDDS principles. Section 3 presents our resource discovery solution through the proposed protocol. It also describes the maintenance process. The simulation analysis study section shows the benefit of our proposition. Section 5 details related work. The final section contains concluding remarks and future works.

2. Preliminaries

2.1 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 later LH* variants (LH^*_{RS}, LH^*_{g}…). We assume that the reader is familiar with a linear hashing algorithm LH* as presented in [15]. Each node stores records in a bucket which splits when the file grows. Every LH* peer node 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.

Each record in LH * is identified by its key whivh determines the record location according to the linear hashing Algorithm described in [29]. 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 node knowing the correct state of the file or relation. However, [29] ameliorates this scheme. Split coordinator does not constitute a centralized node 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. The peer server applies another algorithm $LH^*_{RS}{}^{P2P}$ described in [29]. It first verifies whether its own address is the correct one. If needed, the server forwards this query. The query always reaches the correct bucket in this step. Then, 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 [29]. Hence, the most important property here is that the maximal number of forwarding messages for key-based addressing is one. Another advantage of using SDDS is the possibility to support range queries very well and the less vulnerability in the presence of high churn [29].

2.2 Principles of Hierarchical Distributed Hash Tables

Structured systems such as DHT offer deterministic query search results within logarithmic bounds as sending message complexity. In systems based on DHT as Chord [25], Pastry [24] and Tapestry [33], 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 [24]. 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. 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.

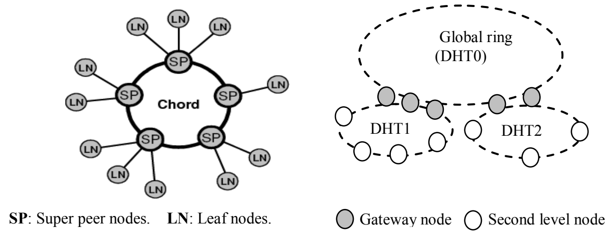

Fig. 1. SP-HDHT (left) and MG-HDHT (right) solutions.

In Fig. 1-left, 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 [31]. However, [17] 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. Multi-Gateway Hierarchical DHT (MG-HDHT) solution [18] is another example of 2-levels hierarchy system having multiple gateways by VO (Fig. 1- right). 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. 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. Hence, there is a trade-off between minimizing total network costs and minimizing the added overhead to the system.

3. Resource Discovery through SDDS based Hierarchical DHT Systems

A resource discovery is a real challenge in unstable and large scale environments. It constitutes an important step in the evaluation of a query in Grid environment [22]. The fact that users have no knowledge of the resources contributed by other participants in the grid poses a significant obstacle to their use. Hence, a centralized scheme forms naturally a bottleneck for the system [20]. The duplicated approach forces the update in every peer which will result in flooding the network. The distributed approach is more appropriate in such systems [19]. In this context, distributed Peer to Peer techniques are used to discover resources in data Grids. Furthermore, Grid environment is likely to scale to millions of resources shared by hundreds of thousand of participants. In consequence, the fact that peers frequently leave/join the system generates high maintenance costs especially on the presence of a churn effect. We have first study a flat DHT resource discovery solution. When one searches a peer responsible for some resource, the typical number of hops in DHT is $O\ (log_B(N_T))$ when N_T is the total number of nodes in the system. However, value of N_T can be a greater number and the maintenance of the DHT will be more complex. More, this solution does not take into account the autonomy of organizations. One solution to this problem is to deal with a super peer model. However, a super peer acts as a centralized resource for a number of peers which depend on the availability of the super peer. Also, a single point of failure of this peer constitutes a serious problem. We have study some previous hierarchical DHT solutions. Existing solution as [31] improves significantly the routing performance. But, complex algorithms are suitable to manage connection between nodes and performances depend on the ratio between super peers and total number of peers.

3.1 Architecture

Instead to adopt one of these solutions, we propose 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) [11] as shown in Fig. 2. Every VO is dedicated to an application domain (e.g., biology, pathology) [14]. It permits to take into account the locality principle of each VO [10]. 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. In this context, [5] proved that a DHT lookup algorithm required only minor adaptations to deal with groups instead of individual peers. 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*. 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 [17]. 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. We consider here the peers as homogenous. Recall also 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 work.

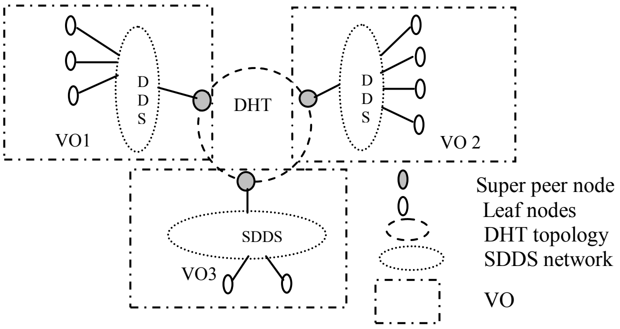

Fig. 2. SDDS based hierarchical DHT architecture.

3.2 Resource Discovery Protocol

In this section, we describe the resource discovery protocol used in the proposed SDDS-HDHT solution. Suppose that a second level peer p_i Є 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*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.

Intra-VO resource discovery queries are evaluated through a classical $LH*_{RS}^{P2P}$ routing system which is completely transparent to the top level DHT. 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_i before searching in other VOs. This solution favors principle of locality [10]. Recall that finding a peer responsible of metadata of the searched resource requires only two messages. Finally, the peer p_J sends metadata describing Res (if founded) to p_i, the peer initiator of Q.

When the researched resource *Res* is not available in the local VO_i, resource discovery is required in other VOs. This corresponds to an inter-VO resource discovery process. Before introducing the resource discovery process, let's recall that we have defined a certain period of time (e.g. Round- Trip Time RTT) as in [21]. The manner in which the RTT values are chosen during lookups can greatly affects performances under churn. [23] 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 p_0 in its VO. Then, p_0 forwards *Gp_i_list* to all other second level peers. It i the nearest second level peer s done just on the connection step.

Let now examine an inter-VO lookup cost in SDDS-HDHT solution. When *Res* is not found in VO_i, the query is propagated to the gateway Gp_i. 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. In summary, 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, already received.

(ii) Once the query reaches a gateway peer Gp_i, a hash function H is applied to *Res* in order to discover the gateway responsible for the VO that containing *Res*. The query arrives at some 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) Metadata of *Res* are sent to Gp_j which forward it to p_i via the reversing path.

3.3 System Maintenance

The continuous leaving and entering of nodes into the system is very common in Grid systems (dynamicity proprieties). 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 seriously damages the structure of the overlay with data loss consequences. Remedying this failure generates additional maintenance cost. In structured peer-to-peer systems, such as Pastry [24] used in our system, the connection / disconnection of one peer generates $2B*Log_B(N_T)$ messages [24]. Furthermore, the maintenance can concern the connection/ disconnection of one or more peers. Throughout this section, 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) [23]. Then, we discuss the connection/ disconnection of both gateways and second level peers.

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.

Gateway Peer Connection/ Disconnection. For this aim, we propose a protocol in order to reduce the overhead added to the system. When a gateway peer connection/ disconnection occur, we distinguish two types of maintenance: (i) maintenance of the DHT and (ii) maintenance of the neighbour's lists. We will not discuss the first maintenance since it corresponds to a classical DHT maintenance [25]. In the other hand, 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. Addressing 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).

(iii) 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 i*Gp_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:

(i) The new gateway *Gp_{New}* contacts the nearest (only one) second level peer p_0 and gives him its neighbor's list *Gp_{New}_list*.

(ii) Peer p_0 inform peers in *Gp_{New}_list* about its existence. But, it does not inform other second level peers about *Gp_{New}_list* (lazy update).

Remark that the peer p_0 do not sent description of the new gateway peer *Gp_{New}* and its updated *Gp_{New}_list* to other second-level nodes at this moment. A lazy update is adopted. When Gp_i does not respond after a RTT period, a second level peer consults its old *Gp_i_list* to reach other VOs. Thus, it rejoins the overlay network in spite of a gateway failure. The update of this list is done during the reception of the resource discovery result as in [14]. 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.

4. Performance Analysis

Experimental results based on a simulation of the suggested resource discovery solution are presented in this section. We based on a virtual network as 10000 nodes 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 nodes organized as virtual organization in a real existing platform as Grid'5000 [8]. We based our experiments on a platform having four features: (i) emulation of nodes, ii) emulation of network, (iii) using FreePastry [4], one implementation of the Pastry DHT and (iiii) $LH^*_{RS}{}^{P2P}$ SDDS prototype implemented by Litwin's team in Dauphine University [2]. Variables used bellows are defined as follows: N_T is the number of nodes in the system, N_G the number of super peers, NSL the number of second level nodes and α the super peer ratio. It is the ratio between gateways and the total number ($N_G= \alpha. N_T$). Key of the discovered resource corresponds to a relation name in our experiments. For the detection of failed peers, we set a TTL to 1 sec. We simulate performances of (a) a flat DHT solution in order to measure the benefits hierarchical systems and previous hierarchical DHT solution b) SP-HDHT solution [31] in which gateways establish a DHT overlay network when each leaf peers maintains a connection to its gateway, (c) MG-HDHT solution in which several gateways are maintained between hierarchical levels. Then, we compare theirs performances.

Throughout this section, we deal with three classes of experiments: (i) Lookup performances experiments in which we interest to elapsed times which includes the query processing and communication costs. (ii) maintenance overhead experiments in which we simulate a join/leave peers scenario and interest to the required update messages and (iii) experiments to find the optimal ratio between gateway and second level peers in order to evaluate the impact of the gateway ratio in performances. For this aim, we have varied N_G but the total number of peers always stay constant.

4.1 Lookup Performances Analysis

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 sections 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. Fig. 3-left 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.

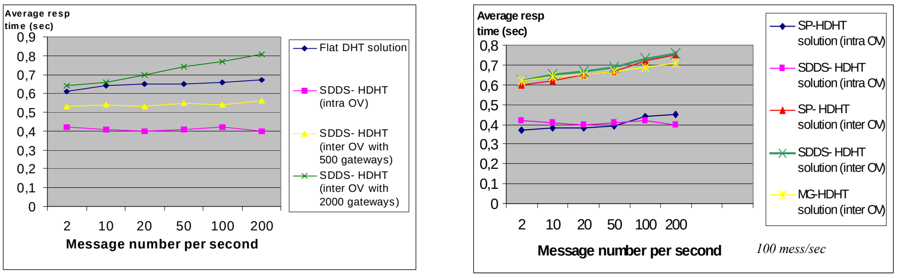

Fig. 3. SDDS-HDHT performances vs. Flat DHT (left) and SP-HDHT (right) performances

We have also compared our results to both SP-HDHT and MG-HDHT results. [31] proved that best performances are obtained with small number of gateways. We simulate a network with 100 VOs (with 100 level peers/ VO). Fig. 3-right 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 discovery queries. Note that these experiments do not include the more costly connection step.

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.

4.2 Maintenance Analysis

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 [7]. 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.

Lets a system with a peers distribution as {N_G=100 and 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. Fig. 4-left 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. 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 join the system. 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.

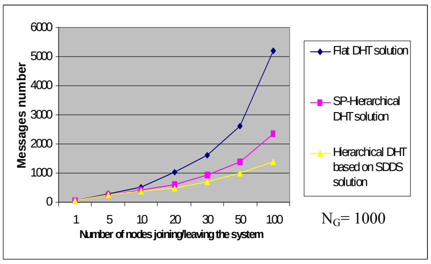

Fig. 4. Impact of the connection/ disconnection nodes in the messages number exchanged in the system.

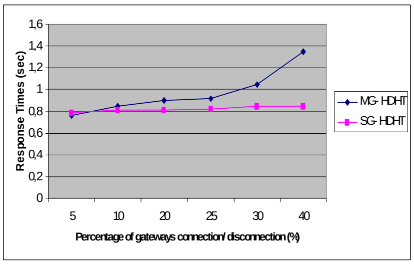

Fig. 5. Impact of the percentage of the gateways connection/ disconnection in the total response time.

We also experiment the impact of the percentage of the gateways arrival/ departure in the total response time as shown in Fig. 5. It corresponds to resource discovery process under a high churn. When only 5% of gateways are replaced by other gateways, MG-HDHT solution has 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 they used, in MG-HDHT solution, the 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 the Gp_i*_list*. But, recognize that if all peers in the Gp_i*_list* failed, consequences are also the same as above.

4.3 Impact of the Gateway Ratio in Performances

Through these experiments, our goal is to determine optimal configurations on the three compared solutions. In first experiments, without any peer arrival/departure to the system, a centralized overlay network with only one super peer in SP-HDHT solution generates the lowest traffic costs. The reason is that only lookup and *Ping/ Pong* messages are exchanged between the super peer and its second level nodes. Also, same performances are obtained with the configuration (α=100%) in the three experimented solution since all peers participate in a flat DHT overlay. If 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 from α=10% with our solution. We also deal with experiments taking into account the arrival/ departure of peers to the system. We deal with the connection/ disconnection of 10% of the gateways in the system and 10% of second level nodes in each VO. From α=1%, the maintenance cost of the MG-HDHT solution is always the most important since each gateway inform all its second level nodes in each arrival/ departure. It is also the case with the SP- HDHT solution with better results. This is not the case in SDDS- HDHT which has the best results with □between 1 and 50%. It is due to the fact that second level nodes used a lazy update to update their neighbor's gateway list. For each 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 nodes but also for the case when the departure/ arrival of gateways constitutes the major maintenance cost. We conclude that the best results are of SDDS-HDHT solution are obtained with $\alpha \in [1\%, 20\%]$ which is close to real grid systems with several VOs.

5. Related Work

Many research works [5], [6], [12], [17], [18] and [31] presented advantages of hierarchical DHT systems based on the super peer concept. However, most of them add a significant overhead to the system. [5] 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. [28] demonstrated the high maintenance state needed (memory, CPU and bandwidth) when all peers in the overlay are attached to different levels of the hierarchy. [18] explored the using of multiple Chord systems in order to reduce latency of lookups. Nevertheless, it neglects the churn effects. [31] 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. [12] 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 [7], [9], [16], [23] and [32] were proposed to resolve these problems. We cite the Bamboo protocol [23] designed to handle networks with high churn efficiently and the self organizing distributed algorithm [32] 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. Hence, there is trade-off between minimizing total network costs and minimizing the added overhead to the system. For these reasons, we have proposed to combine DHT and SDDS structures in order to minimize these costs without excessive overheads.

6. Conclusion and Future Works

We have proposed a hierarchical DHT solution for data sources 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. Only fewer nodes are mapped on a DHT. Each of them acts as a super peer for leaf-nodes and can serves a Virtual Organization (VO), structured as an SDDS, in a Grid. The first contribution is the improvement of lookup query complexity to discover metadata of any data source especially for intra-VO queries since these queries are transparent to the top level DHT lookup. Also, only the arrival of a new VO requires the DHT maintenance. Our solution addresses also other super peer problems as a single point of failure by using a minimum of messages. In fact, leaf nodes update theirs super peer neighbours during resource discovery queries. The performance analysis shows the benefit of our proposition through comparisons of our performances to those of previous solutions. It shows the improvement of lookup query performances especially when we have an important number of simultaneously resource discovery messages. It also shows a significantly maintenance saves especially in presence of dynamicity of nodes.

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 nodes. Also, we would like include more realistic models of churn as to scale traces of sessions times [3] collected from deployed networks to produce a range of churn rates with a more realistic distribution. Also, we would like to study the effects of alternate routing table neighbours as in [33].

7. References

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

2. http://lamsade.dauphine.fr/~litwin/default.html

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

4. http://Freepastry.org/FreePastry/.

5. L. Garces-Erice, E. W. Biersack, K. W. Ross, P. A. Felber, and G. Urvoy-Keller. Hierarchical Peer to Peer Systems. In Proc. of ACM/IFIP Intern. Conf. Euro-Par’03.

6. P. Ganesan, K. Gummadi, and H. Garcia-Molina. Canon in g major: designing DHTs with hierarchical structure. Intern. Conf. on Distributed Computing Systems’04, pp 263–272.

7. P. B. Godfrey, S. Shenker, and I. Stoica. Minimizing Churn in Distributed Systems. Int. Conf. SIGCOMM. pp 147–158, Italy 2006.

8. GRID'5000. www.grid5000.org

9. I. Gupta, Ken Birman, P. Linga, A. Demers & R.V Renesse. Kelips: Building an Efficient and Stable P2P DHT through Increased Memory and Background Overhead. Lecture notes in computer science, 2003. Springer.

10. N Harvey, M Jones, S Saoiu, M. Theimer & A. Wolman. Skipnet: A Scalable Overlay Network with Practical Locality Properties. In Proc of USITIS 2003, Seattle, USA.

11. A. Iamnitchi, I. Foster, "A peer-to-peer approach to resource location in grid environments", Proc. of HPDC'02, Edinburgh, UK, August 02.

12. 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.

13. Kazaa. http://www.kazaa.com/.

14. I. Ketata, R. Mokadem, F. Morvan. Resource Discovery Considering Semantic Properties in Data Grid Environments. Proc. of Inter. Conf. Globe 2011, Toulouse, Springer, LNCS 6864.

15. 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'95.

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

17. I. Martinez, R. Cuevas, C. Guerrero, A. Mauthe. Routing Performance in a Hierarchical DHT-based Overlay Network. Euromicro Intern. Conf. PDP'08, 508-515, Toulouse.

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

19. E. Meshkova & al. A survey on Resource Discovery Mechanisms, Peer to Peer and Service Discovery Frameworks Computer Networks. Science Direct. Elsevier'08 (2097- 2128).

20. R. Mokadem , A. Hameurlain, A. Min Tjoa. Resource Discovery Service while Minimizing Maintenance Overhead in Hierarchical DHT Systems. In Intern. Conf. on Information Integration and Web-based Applications & Services (iiWAS'10), Paris, France.

21. Mastroianni C., Talia D. and Verta O. "Evaluating Resource Discovery Protocols for Hierarchical and Super-Peer Grid Information Systems". 19th Euromicro Intern. Conf. PDP'07.

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

23. 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.

24. 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.

25. I. Stoica, Morris, Karger, Kaashoek, Balakrishma. CHORD : A scalable Peer to Peer Lookup Service for Internet Application. SIGCOMM'O, August'01, San Diego, USA

26. 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).

27. 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. 2005.

28. 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.

29. H. Yakouben, W. Litwin, T. Schwarz. "LH*$_{RS}^{P2P}$: a Scalable Distributed Data Structure For the P2P Environment". Int conf. on new technologies of Distributed Systems. France, 2008.

30. B. Yang and H. Garcia-Molina. Designing a Super-Peer Network. Proc. of intern. conf. on Data Engineering ICDE'03, Bangalore, India.

31. S. Zöls, Z Despotovic, W Kellerer. "Cost-Based Analysis of Hierarchical DHT Design". Intern. Conf. P2P'06. Cambridge, IEEE Computer Society 2006 pp 233-239.

32. S. Zöls, Q. Hofstatter, Z. Despotovic, W. Kellerer. "Achieving and maintaining Cost-Optimal Operation of a Hierarchical DHT System". Proc. of Inter. Conf. ICC 2009, Germany.

33. B. Zhao, Kobiatowicz & A. Joseph. Tapestry: A resilient global scale overlay for service deployment. IEEE journ. on selected Areas in communications, 22 vol 1,2004.