Resource Discovery Service while Minimizing Maintenance Overhead in Hierarchical DHT Systems *

Riad Mokadem
IRIT Lab., Paul Sabatier University,
118, route de Narbonne, 31062,
Toulouse France.
Mokadem@iirit.fr

Abdelkader Hameurlain
IRIT Lab., Paul Sabatier University,
118, route de Narbonne, 31062,
Toulouse France.
Hameur@irit.fr

A Min Tjoa
Institute of Software Technology,
Vienna University of Technology
Favoritenstr 9-10 Wien, Austria
Amin@ifs.tuwien.ac.at

ABSTRACT

Using Distributed Hash Tables (DHT) for resource discovery in large-scale systems generates considerable maintenance overhead. This not only increases the bandwidth consumption but also affect the routing efficiency. In this paper, we deal with resource discovery while minimizing maintenance overhead in hierarchical DHT systems. The considered resources are metadata describing the data resources. In our solution, only one gateway in one overlay is attached to the superior level overlay. It aims to reduce both lookup and maintenance costs while minimizing the overhead added to the system. We present a cost analysis for a resource discovery process and discuss capabilities of the proposed protocol to reduce the overhead of maintaining the overlay network. The analysis result proved that our design decrease significantly the maintenance costs in such systems especially when nodes frequently join/leave the system.

Keywords

Grids, Peer to peer systems, Resource discovery, Distributed hash tables, Hierarchical model, Super peer model.

1. INTRODUCTION

A resource discovery is a real challenge in unstable and large scale environements. It consists to discover resources (e.g., computers, data) that are needed to perform distributed applications in such systems [15].

Large amounts of research works have been adopted Peer-to-Peer solutions to deal with resource discovery in Grid systems [8]. Structured Peer-to-Peer systems as DHT are self-organizing distributed systems designed to support efficient and scalable lookups in spite of the dynamic proprieties in such systems. Classical flat DHT systems organize nodes, 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 does not consider the autonomy of virtual organizations and their conflicting interests [9]. Moreover, typical structured P2P systems as Chord [21] and Pastry [18] are very sensitive to the dynamics of the network. They 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 [15] 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 [23] is more effective in very large Grids. Several research works [3, 4, 12, 13, 19, 24] 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 [3]. In this context, a Grid can be viewed as a network composed of several, proprietary Grids, virtual organizations (VO) [1] where every VO is dedicated to an application domain (e.g., biology, pathology). Within a group, one or more nodes are selected as super peers to act as gateways to nodes in the other groups. However, most of existing solutions neglect the churn effect and deal only with the improving performance of the overlay network routing. More, they generate significant additional overhead to large scale systems. Several proposals [7, 10, 11, 17, 20, 25] for reducing maintenance costs have also appeared in the literature. For example, [11] proposed an algorithm SG-1 to find the optimal number of super peers in order to reduce its maintenance costs. It is based on the information exchange between super peers on their capacities through a gossip protocol. Also, despite a good strategy to manage a churn in [20] with a lazy update of the network access points, inter-organization lookups were expensive because of the complex routing 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 a single-gateway based hierarchical DHT solution (SG-HDHT) for an efficient resource discovery in Grids. We focus on the discovery of metadata describing the data resources. Our solution reduces both lockup and maintenance costs while minimizing the maintenance overhead added to the

* *This work is supported in part by 'the Ministry of Foreign and European Affairs' and 'The Ministry of Higher Education and Research'. Amadeus program 2010 (French-Austrian Hubert Curien Partnership – PHC). Grant Number 23066XC.*

system. The system forms a tree of structured overlay (e.g., DHT) and consists of a two level hierarchical overlay network. Only one peer (called gateway or super peer) in one overlay is attached to the superior level of the hierarchy when previous solutions as [13] employ several gateways which increase the required maintenance overhead cost. Resource discovery queries, in our system, are classified into intra-VO and inter-VO queries. The intra-VO discovery is a classical discovery based on a DHT lookup. In fact, super peers are not concerned by intra-VO queries unlike previous hierarchical DHT solutions as [24] which put super peers more under stress by maintaining pointers between super peers and their leaf nodes. We also discuss inter-VO lookup costs. Queries are first routed to the reduced DHT overlay which permits to locate the super peer responsible of the VO containing the resource to discover. Then, another DHT lookup is done in this V0 to discover metadata of this resource.

We also demonstrate that ou solution can handle high churn rates. SG-HDHT solution mainly deals with the reduction of maintenance costs especially when nodes frequently join or leave the system. We explore the different factors that affect the behavior of hierarchical DHT under churn (super peer failure adressing, timeouts during lookups and proximity neighbor selection) [17]. The reduced number of top level DHT entries, in our solution, offers a significant maintenance global DHT saves and only the arrival of a new VO requires its maintenance. Other super peer model problems are also resolved in our system. A failure of a super peer node does not paralyze the system. To be able to react to this failure, we need to store and maintain less information than previous solutions as [24]. Instead each leaf nodes sends periodically a message to its super peer, this later sent list of its neighbours to only one second level node by VO except for the connection step. Other second level nodes update their super peer neighbours list during resource discovery queries. This protocol overcomes the single point of failure problem in super peer models without putting super peers on stress. A simulation analysis evaluates the efficiency of our resource discovery service. It shows that our solution reduces resource discovery lookups especially for intra-VO queries. They also provide good performances to reduce significantly maintenance costs when nodes join/leave the system.

The rest of the paper is structured as follows. Section 2 details related works. Section 3 recalls hierarchical DHT principles. Section 4 presents our resouirce discovery solution and the associated protocol for an efficient resource discovery in Grids. The performance study section shows the benefit of our proposition. The final section contains concluding remarks and future works.

2. HIERARCHICAL DISTRIBUTED HASH TABLES

Structured systems such as DHT is 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 [21], Pastry [18] and Tapestry [26], 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 node is responsible to manage a small number of peers and maintains its location information. Pastry DHT system offers deterministic query search results within logarithmic bounds as sending message complexity ($O(log_B(N))$ 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 node arrivals, node failures and recoveries. Unlike Chord nodes, Pastry peer permits to easily locate both the right ad left neighbours 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 nodes into a multi level overlay network. Because a node 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, each node is assigned a group identifier (*gid*) and a unique node identifier (*nid*). Groups are organized in the top level as Pastry overlay network. Whithin each group, nodes are organized as a second level overlay using the *nid* identifier. In each group, one or more nodes are designated as super peers. They act as gateway to other nodes in the group as in [3, 13, 24]. Throughout this section, we interest to two previous hierarchical DHT solutions which we consider comparable to our solution. In Figure 1-left, super peers establish a structured DHT overlay network when leaf nodes maintain only connexion to their super peers, denoted by SP-HDHT solution (direct connection between a super peer and its other nodes) [24]. However, as shown in [12], this strategy can maintain super peers more under stress. Also, performances depend on the ratio between super peer's number and the total number of nodes in the system. [13] is another example of 2-levels hierarchy system, denoted by MG-HDHT solution and having multiple gateways by VO (Figure 1- right). The nodes connecting by lines are instances of the same node, 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 o*rganizational rings* at the lower level. This solution provides administrative control and autonomy of the participating organizations. However, unlike efficient intra-organization lookups, inter-organization lookups are expensive since the high maintenance cost of the several gateway nodes.

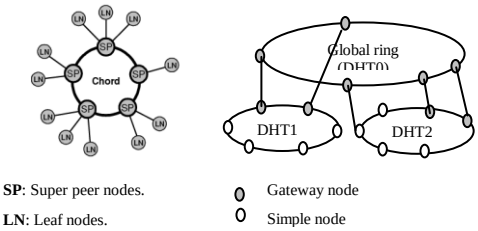

Figure 1: SP-HDHT (left) and MG-HDHT (righ) solutions.

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.

3. PROPOSED RESOURCE DISCOVERY METHOD

Today, the resource discovery is a very important topic in large scale data Grids and constitutes an important step in the evaluation of a query in Grid environment [16].

Grid environment is likely to scale to millions of resources shared by hundreds of thousand of participants. In consequence, a centralized placement scheme can be a bottleneck for the system and its distribution over nodes in Grid will be very desirable [8]. Also, the cost to discover resources participating in some query can be very important since we must discover all metadata of any resource. More, the fact that nodes frequently leaved and joined the grid system constitutes a serious problem for maintenance on the same basis of the unavailability of some nodes. In this case, managing a churn can add overhead to the system. In this section, we discuss capabilities of the proposed hierarchical DHT solution to reduce both lookup and maintenance costs while minimizing overhead added to the system. Recall that, in this paper, we have not interesting on the assignment of a joining second level node to an appropriate super peer, i.e., loads balancing. We defer this issue to a future work.

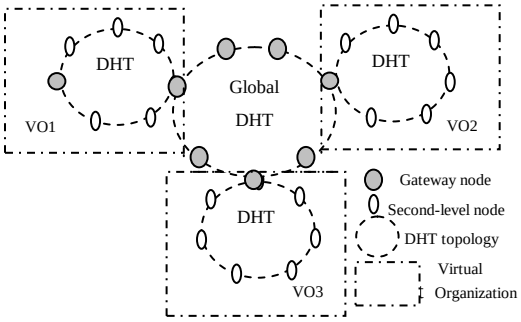

Figure 2: A single gateway hierarchical DHT (SG-HDHT) architecture.

Our solution consists in defining a DHT for each VO. Resource Discovery through our solution deals with two different classes of peers: super peers and simple nodes (called also second-level nodes). For each VO, one peer viewed as a super peer acts as a proxy for other nodes which can be viewed as simple nodes. It also establishes a DHT overlay with other second-level nodes of the same VO. This solution favors principle of locality when the query is submitted first in the local VO.

Super peers establish a structured DHT-based overlay. Each super peer constitutes an entry point for its corresponding VO. Figure 2 shows the two-level hierarchy architecture. The top level DHT routing system permits to lookup super peers. The second level DHT routing system, completely transparent to the top level DHT one, allows an efficient access to simple nodes in one VO. In order to facilitate location of nodes responsible for the discovered resource, the top level DHT lookup is done through a hash function H which is different to hash functions h_i used in second levels DHT routing system.

3.1 Resource Discovery Protocol

Suppose that a peer (second level node) $p_i \in VO_i$ wants to discover the peer that is responsible for a resource *Res* over a resource discovery query *Q*. Let sp_i the super peer responsible for VO_i, sp_i_list the list of its neighbors in the top level DHT and *Response* the metadata of *Res.*

In our SG-HDHT solution, gateway nodes communicate with each other through a top level DHT overlay network. Each of them knows, through another second level DHT routing system, how to interact with all other nodes in the same VO. 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*. An other hash join h is also applied to this resource in order to generate a node identifier *nid*. Thus, a lookup request for resource *Res* implies locating the group responsible for *Res*. It is routed to the super peer of the initiating group. Then, the lookup request is routed, via the top level DHT routing, to the super peer of group, whose group identifier gid is *gid=Successor(Res)*. Finally, it is forwarded to one of the second level nodes through the second level DHT routing system. This permits to associate each resource to its VO [12]. We can also define a time to live parameter (TTL), as in [14]. It is useful to maximize time to discover resources. It is mainly useful when a failure occurred in a super peer node. In this case, p_i do not expect indefinitely. When TTL is exceeded, it consults its super peer neighbours list and sends its query to one of the peers founded in this list.

```
Discover (pJ, Res, VOi)
{ if (DHT_hi_route(Res, VOi)==pJ) then Send Response to pi;
  Else forward Q to spi
      If spi do not respond then
                          Consult spi_list;
                  Forward Q to neighbors of spi ;
      If  DHT_H_route(Res)==spJ  then
              if (DHT_hJ_route(Res, OVJ)== pJ)
                        then send Response to pi ;
                  else response= { Result not found }
    else response= { Result not found }
}
```

Figure 3: Metadata discovery pseudo code.

An example is to discover metadata of a database relation *R* referenced in some SQL query *Q: Select * From R*. Metadata of R contains: (i) attributes of *R* (ii) placement of *R* (address IP of each fragment, its fragmentation, duplication information's and the construction rule of *R)* and (iii) size of *R*. Before detailling the resource discovery process, let's recall that in the connection step of sp_i, this later sent its list neighbors sp_i_list (the left and right neighbor) to the nearest second level node p_{SL0} in its VO. These neighbors are located by using *neighborhood set* parameter we described above. Then, p_{SL0} forwards sp_i_list to all other second level nodes. It is done just on the connection step. Our resource discovery service is defined as five steps: (i) The metadata resources discovery query *Q* is first performed in the local VO_i. It uses the local DHT routing system, completely transparent to the top level DHT. In this case, it constitutes an intra-VO query. (ii) When *Res* is not found in VO_i, *Q* constitutes an inter-VO query. The peer p_i routed the query to its super peer sp_i. Otherwise, if sp_i failure is detected (TTL is elapsed), it requests one neighbor of sp_i (received during the connection step). (iii) Once the query reaches a super peer sp_i, a hash function H is applied to *Res* in order to discover the super peer responsible for the VO that containing *Res*. The query arrives at some super peer sp_J. This is valid whenever a resource, matching the criteria specified in the query, is found in some VO_J. (iv) Using the DHT routing system in the founded VO_J, sp_J routes the query to the peer $p_J \in VO_J$ that is responsible for Res. (v) Finally, Metadata of *Res* are sent to sp_j which forward it to p_i via the reversing path. We resume these steps by the pseudo code showed in Figure 3.

3.2 Lookup cost Analysis

Let's analyze the metadata lookup costs in the resource discovery process. Suppose 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_{SP} represents the number of super peers which is also the number of VOs in the system. Let N_T represents the total number of peers. Let N_{SL} the total number of second level nodes. An important parameter for the architecture proposed in last section is the ratio between super peers and the total number of peers. We denote it by α and call it the super peer ratio. Then, $N_{SP} = \alpha . N_T$ and the total number of second level nodes is $N_{SL} = (1-\alpha).N_T$ when N_{SL}/ N_{SP} constitutes a number of second level nodes by VO. During our analysis, we interest to the number of messages generated in the system regardless on the message size. We further assume here that a system is in a steady state, i.e., no churn occurs and that super peers' DHT overlay is nonempty. We discuss the resource discovery costs in the two cases:

(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.

Throughout this section, we analyze lookup costs generated by our solution. Then, we compare them to those of the SP-HDHT solution already discussed in section2. We do not compare our costs to those of MG-HDHT [13] in this case since costs are very close without simultaneous messages which analyze in the performance analysis section. Then, we discuss the impact of α on each solution. It is clear that a total lookup traffic cost (*Lc*) consists of the lookup costs through second level nodes Lc_L and lookup cost through super peers Lc_S. Thus, the total lookup cost here is $Lc = Lc_L + Lc_S$.

For an intra-VO query, a lookup is done through a classical DHT routing system. It is equal to $Lc_L = O(log_B(N_{SL}/N_{SP}))$ when this lookup requires $O(log_B(N_T))$ in flat DHT solutions. When we compare our solution to the SP-HDHT one, it is clear that leaf nodes in this solution must first contact its super peer which forwards the query to the leaf node responsible for *Res*. However, this super peer constitues a centralized resource. This is a serious disadvantage especially when we have several simultaneous messages. This is not the case when our solution is adopted.

Let examine the case of an inter-VO resource discovery query. When *Res* is not found in a VO_i, p_i sends *Q* to its super peer sp_i. Localization of the super peer sp_J responsible for the VO containing *Res* requires $Lc_S = O(log_B(N_{sp}))$ hops. Another DHT lookup is required to search metadata of a resource in the founding VO_J. The total lookup cost for an inter-VO resource discovery query here is $Lc = 2*O(log_B(N_{SL}/N_{SP})) + O(log_B(N_{sp}))$ messages. Certainly, compared to the flat solution complexity, both are logarithmic. But, the difference can be real if $N_{SL} << N_T$ *i.e.*, in a system with many VOs, having a reduced number of nodes, this cost became interesting.

In SP-HDHT solution, the fact that all second leaf nodes forward their queries to its super peer constitutes a disadvantage. It is also the case with our solution. Hence, it is clear that the impact of α is real in inter-V0 queries especially with simultaneous messages. The using of several gateway nodes as in [13] put them less under stress but, in other hand, should have a serious inconvenient when the system is maintained. In the performance evaluation section, we evaluate capabilities of each solution in the presence of simultaneous resource discovery queries.

4. SYSTEM MAINTENANCE

The continuous leaving and entering of nodes into the system is very common in Grid systems (dynamicity proprieties). In general, node departures can be divided into *friendly leaves* and *node failures*. Friendly leaves enable a peer to notify its overlay neighbors to restructure the topology accordingly. Node failures possibility is more complex and seriously damages the structure of the overlay with data loss consequences. Throughout this section, we discuss the maintenance cost in each possibility. Dealing with our two-level hierarchy architecture, maintenance costs are: (i) DHT maintenance in its two levels (local DHT defined for each VO_i and the global DHT overlay formed by super peers) required when a node join/leave the system and (ii) the maintenance of the access point establishments which consists of defining how access point are established and updated when a super peer or a second level node is connected/ disconnected. We deal with these costs and compare them to costs generated by the SP-HDHT and MG-HDHT solutions. We also explain how to interact with a super peer failure. This failure paralyzes access to all second level nodes which is responsible for them. Remedying this failure generates additional maintenance cost.

4.1 Node Connection/ Disconnection

We analyze the connection step for both super peers and second level nodes. When a new second-level node p_{SLi} connects/ disconnects to/from the system, the local DHT maintenance is updated as in classical DHT maintenance. When p_{SLi} joins some VO, it asks its neighbor in the local DHT about sp_i_list. In consequence, only 2 messages are required since only one gateway exists in our solution. This process avoid that several new arrival nodes asked simultaneously the same super peer which can constitute a bootleneck as in SP-HDHT solution. Our node connection process is also less complex than in [13]. When a new second level node arrive, it should not retrieve all gateway nodes as in MG-HDHT solution. Let's now analyze the connection of a super peer. We propose a protocol in order to reduce the overhead added to the system. The connection step of any sp_i consists that:

(i) Super peer sp_i sent its list neighbors sp_i_list (the left and right neighbor) to the nearest second level node p_{SL0} in its VO.

(ii) Peer p_{SL0} contacts one super peer in sp_i_list to inform him about its existence (in order to have an entry to this VO_i in the case of sp_i failure).

(iii) Peer p_{SL0} sent this list to all second level nodes in its VO via a multicast message (Recall that other second level nodes do not report their existence to neighbors of sp_i).

Recall that this process is done just once at the initial connection of a super peer and only p_{SL0} periodically executes a *Ping/Pong* algorithm with its super peer sp_i. It sends a *Ping* message to sp_i and this one answers with a *Pong* message in order to detect any failure in sp_i. Let us discuss the case of super peer failure/ update. When sp_i is replaced by another, the process of maintenance (after the global DHT maintenance) is:

(i) The new super peer sp_{New} contacts only one second level node (p_{SL0}) and gives him its neighbor's list sp_{New}_list.

(ii) Peer p_{SL0} inform peers in sp_{New}_list about its existence.

Remark that the peer p_{SL0} do not sent description of the new super peer sp_{New} and its updated sp_{New}_list to other second-level nodes at this moment. It is done during resource discovery queries. Recall that a failure of p_{SL0} does not paralyze the system since the new super peer always contacts its nearest second level node. Note also that a super peer can inform p_{SL0} before disconnecting. Otherwise, if a super peer does not respond after TTL period, a second-level node visits its sp_i_list to contact one of neighbors of the failed super peer and sends its request. Thus, it rejoins the overlay network (any VO) in spite of super peer failure. The entry to the VO can also be done through node p_{LN0} since this one reported its existence in the connection step.

4.2 Maintenance Cost Analysis

In a classical DHT as Chord, the main overhead is due to the periodical refreshing of routing tables, which takes $O(\log^2 N)$ messages per node per refreshing. It is clear that churn generates important routing table maintenance cost when a flat DHT based solution is adopted especially in the case of a great number of nodes. In this section, we demonstrate that ou solution can handle high churn rates and reduce maintenance costs. Dealing with our scheme, the maintenance cost is the sum of: (i) the DHT update cost in the super peers' DHT overlay when a super peer join/leave the system (We notes it Mc_{S1}), (ii) communication costs between a super peer and its second level nodes (we notes it Mc_{S2}) (iii) the DHT update cost in second level nodes' DHT (denoted by Mc_{S3}) and (iiii) cost of messages exchanged between second-level nodes (denoted by Mc_L). Hence, $Mc= Mc_{S1}+Mc_{S2}+ Mc_{S3}+Mc_L$.

In our solution, a global DHT maintenance is required only when super peers join/leave the system. Each connection/disconnection of one super peer to/from its pastry DHT generates $Mc_{S1}= 2B.log_B(N_{sp})$ messages [18] when $Mc_{S2}= 1$ (sending sp_i_list to p_{SL0}). In other hand, a connection of a new second-level node requires $Mc_{S3}=2B.log_B(N_{LN})$ messages to update the local DHT_i. and $Mc_L= 1$ (keeping sp_i_list).

Let's now examine consequences of super peer connection in the SP-HDHT solution in which each leaf node periodically runs a *Ping/Pong* algorithm in order to detect any super peer failure [24]. It also stores the sp_i_list. In this solution, each super peer periodically runs a '*Republish*' algorithm consisting to update the sp_i_list in a real time. Thus, [12] proved that maintenance costs in such system depend strongly on the ratio between super peers and leaf nodes since each leaf node maintains a permanent connection with its super peer. Let's analyze its connection step cost: when a new leaf node joins its VO, (i) it contacts its super peer to have neighbours of this super peer sp_i_list through $2*(N_{LN} / N_{sp})$ messages and then, (ii) it report its existence to (at least one) neighbour of sp_i_list (N_{LN} / N_{sp} messages). Hence, the cost –For one VO- consists of $Mc_{S2}=3*(N_{LN} / N_{sp})$ messages directly with the super peer. Then, any update concerning a super peer generates communication with all leaf nodes. Then, except the connection step, the cost Mc_{S2} is equal to $2*(N_{SL} / N_{sp})$ for each super peer update process.

Let us examine this cost for the MG-HDHT solution. Let *n* the number of gateways between levels. It corresponds to the number of gateways by VO. Let's analyze the connection cost of a second-level node. First, it should retrieve all gateway nodes by asking its neighbors. This later asked also its neighbors. This process is repeated successively until having all gateway nodes ($Mc_L= N_{SL}$ -1 messages). In other hand, costs Mc_{S1} and Mc_{S2} are as in SP-HDHT solution when a connection of a second level node require a minimum of $Mc_{S2}= n$ messages in order to contact all the gateway nodes.

We proceed otherwise in our resource discovery solution: In the connection of a super peer, we have only one message between this one and it's nearest second level node p_{SL0} ($Mc_{S2}= 1$) plus one message between p_{SL0} and one peer belonging to sp_i_list. We have also a multicast message between p_{SL0} and other second level nodes in the VO. This process is done just once at the initial connection of a super peer. For one OV, the total number of messages of initial connection step is $Mc_{S2}=2+ ((N_{SL}/N_{sp})-1)$ messages but the messages interchanged with super peers is only two (during the connection step) for one VO. Then if a new super peer arrives, it contacts only one peer (p_{SL0}) since other second-level nodes used their sp_i_list to access the global DHT. Thus, in this case and except the other costs which are unchanged, the cost Mc_{S2} is equal to only one message with our solution. A Second level node updates its sp_i_list when it receives a result of resource discovery query. Finally, notes that one of the limitations that our solution suffers from is remedying the failure of both a gateway node and its left and righ neighbours at the same time. A solution consists to enrich the neighbours list of the gateway node.

5. EVALUATION PERFORMANCE

To evaluate performances of the proposed resource discovery solution, we based on a virtual simulated network as ten thousand nodes to prove the efficiency of our method in large grid networks. We deal with a simulated environment since it is difficult to experiment these nodes organized as virtual organizations in a real existing platform as Grid'5000 [6]. We also used FreePastry [2], one implementation of the Pastry DHT. We simulate performances of (a) a flat DHT solution in which all nodes run a same DHT protocol in order to measure the benefits the hierarchical system, (b) SP-HDHT solution in which super peers establish a structured DHT overlay network when leaf nodes maintain only connexion to their super peers, (c) MG-HDHT solution in which several gateways are maintained betwenn hierchical levels and (d) our solution (SG-HDHT). Then, we compare theirs performances. Throughout this section, we deal with two classes of experiments:

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

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

5.1 Simulation Environment

Several nodes run in a single Java Virtual Machine (JVM). Each virtual node 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*. 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. We also deal with a same distribution for resources in all experiments. 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} second level nodes for each super peer. But, the total number of peers stays always constant. Key of the discovered resource corresponds to a relation name in our experiments. An example is to discover metadata of a database relation *R* referenced in some SQL query *Q: Select * From R*. Recall also that we have used B=2 as the base used for the *nodeId*. We have also fixed TTL to 1 sec.

5.2 Simulation Analysis

We experiment with flat DHT, SP-HDHT and MG-HDHT solutions. Then, we compare their performances to our SG-HDHT solution performances. We interest to the impact of simultaneous messages and analyze the impact of α in each solution. Recall that the average response time includes the query processing (matching of resource metadata) and communication costs.

5.2.1 Lookup Resource Queries

First experiments simulate a flat DHT solution. In this scheme, the number of required hops to lookup any resource corresponds always to $log_B(N_T)$ which corresponds to 14 hops in ours configuration(N_{sp} =10000). In this case, it is clear that as the number of peers increase as the lookup cost increase for both intra-VO and inter-VO queries. This is not the case with our solution. Although the lookup is logarithmic, our solution is always better for intra-VO queries ($log_B(N_{SL}/N_{sp})$ hops in one VO) especially when we have VOs with reduced number of second-level nodes. Hence, when $N_T/N_{sp}=1$, our solution is equivalent to a flat DHT one.

For inter-Vo queries, we have showed in section 4.2.1 that the maximum number of hops is $2*O(log_B(N_{SL}/N_{SP}))+O(log_B(N_{sp}))$ with our scheme (except when we have 2 second level nodes by VO). By a simple calculation, we remark that the hops number required to lookup a resource is always less with a flat DHT solution since lookups through both global and local DHT required more hops.

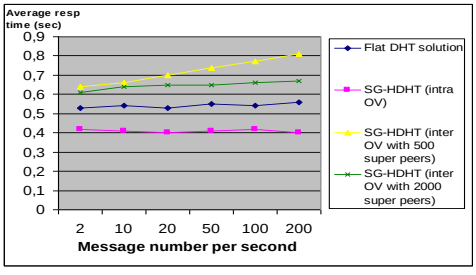

Figure 4: Flat overlay DHT vs. SG-HDHT performances.

However, these results correspond to theoretical number of hops for only one resource discovery query. In the case of simultaneous resource discovery messages, our results should take into account that all messages are forward to the same gateway (in some VO). This generates some congestion in this peer. To confirm this, we have experiment systems with (i) 2000 VOs (5 second-level nodes/ VO, α=20%) and (ii) 500 VOs (20 second-level nodes/VO, α=5%). Figure 4 shows elapsed response times for resource discovery queries (intra and inter-VO queries). It shows that our performances are always better when the resource discover query is an intra VO request. The SG-HDHT method is 50% better than flat DHT solution (esperiment i). This is due to the fact that there are fewer nodes in each VO than in a flat DHT solution which regroup all system nodes. Also, we have not a centralized node in each VO. Thus, intra-VO queries performances does not depend on the number of simultaneous messages. Let analyze performances of inter-VO queries. Figure 4 shows that flat DHT solution is 15% better than SG_HDHT solution when we have a reduced number of second level nodes in each VO (α=20%). This rate is 25% better when α=5%. Indeed, the two DHT lookups provokes an additional cost. We remark that the average response time increases with a great number of second-level nodes. Thus, inter-Vo queries with a reduced α generates an increased cost when a simultaneous messages increases (more than 20 messages per second). It is due to the fact that all messages transit by the same gateway node.

We have also compared our results to those of SP-HDHT and MG-HDHT solutions, described in section 2. We interest to the number of simultaneous resource discovery queries. It is useful since it shows if our method is also scalable in the presence of high number of messages. [24] proved that best performances are obtained with small number of super peers. We simulate a network with 100 VOs (super peers) and 100 second level (called leaf nodes in [24]) connected to each super peer (α=1%). For the MG-HDHT solution, we experiment with 10 gateway nodes in each VO. Figure 5 shows the SP-HDHT solution is slighly better for intra-VO queries when less simultaneous messages are used. We remark that the average response time is almost constant when we have several simultaneous messages in our solution which is not the case in SP-HDHT experiments. From 70 messages/ second, our solution is 10% better than SP-HDHT solution. Same results are obtained with MG-HDHT solution. We explain this by the fact that lookups are done without any gateway node intervention in our solution where a bottleneck is generated in each super peer in the compared SP-HDHT solution. Indeed, a super peer node constitutes a centralized resource for all leaf nodes connected to them in SP-HDHT solution when intra VO-queries in our solution are transparent to the gateway node. These is the reasons why the simultaneous messages influenced significantly the SP-HDHT results since the leaf nodes must first send a query to its super peer which forwards it to other leaf nodes (the reverse path is traversed to send the result). We conclude that the save can be better if we experiment with great simultaneous resource discovery queries.. Note also that these experiments do not include the more costly connection step.

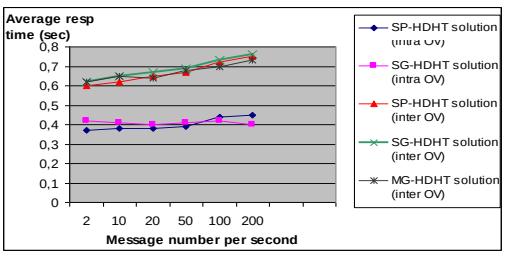

Figure 5: Comparison between SP-HDHT, MG-HDHT and SG-HDHT performances.

For inter-V0 queries, the fact that all leaf nodes sent their queries to a same gateway node increases response times in SP-HDHT and our solution. Simultaneous resource discovery queries influences performances of both solutions. Bottleneck is generated since all queries transit by the same gateway node. Then, SP-HDHT results are slighly better when we have less than 70 messages per second. From this value, results are almost close for the two solutions with sligh advantage to SG-HDHT since intra-Vo queries always precede inter-VO queries (the discovery should propagated towards the global DHT only if the intra-VO query fails). We conclude that, in inter-VO discover queries, we have dependence between performances and simultaneous queries for these two solutions. In the other hand, performances of MG-HDHT solution are better (rate of 5%) especially for high simultaneous messages. The same empact is observed with a reduced super peer ratio α. Total network cost decreases with a high number of gateways and few number of VO (decreasing of N_{sp}) while a total centralization (N_{SP}=1) may overload and therefore endanger system stability in all solutions.

5.2.2 Maintenance Overhead Evaluation

In following experiments, we measure the impact of the join/ leave nodes in the system. We tabulate churn in an event-based simulator which processes transitions in state (*down*, *available*, and *in use*) for each node 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 nodes join/leave the system. [18] shows that the number of required messages when a node joins/leaves a flat Pastry DHT is $2B\ Log_B(N)$ when N is the total number of nodes. It is clear that maintaining this DHT generates greatest costs especially when several nodes join/leave the system. In our solution, when a gateway node joins/leaves the system generates $2BLog_B(N_{SP})$ messages plus a connection between the new super peer and one second-level node. When the connection concerns a second-level node, the cost corresponds to the number of messages sent by a new node to having neighbors of its super peer. It is done without any update in the gateway's DHT.

Lets a system with a nodes distribution as {N_{SP}=100 gateway nodes with N_{SL}/N_{SP}=100 second-level nodes for each of them}. This configuration corresponds to average results in inter-VO discovery queries performances. In these experiments, when a number of new connections/ disconnections exceeds 20 nodes, 10% of them concern gateway nodes. Figure 6 shows the impact of nodes which join/leave the system in the total messages number in the system. Flat DHT solution generates the greater number of messages in the connection or disconnection of nodes. In the case of connection/ disconnection of one second-level node, it generates 10% messages more than our solution. The speedup of the SG-HDHT solution with regrad to the flat DHT solution is 4.5 for the connection/ disconnection of 100 second-level nodes.

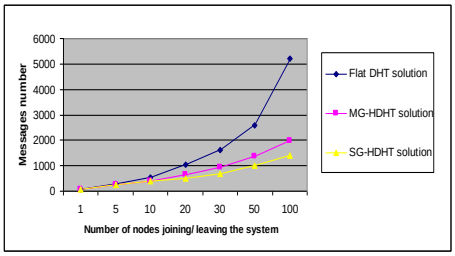

Figure 6: Impact of the connection / disconnection of nodes in the messages number exchanged in the system.

We also measured the maintenance ovehead reduced by our solution as opposed to the ordinary MG-HDHT solution. We experiment MG-HDHT solution with a same node distribution described above with ten (10) gateway nodes in each VO. We interest to the total number of generated messages required to maintain the system in both solutions. The numbers of update messages are very closes when we have only second-level nodes connections/disconnections. It corresponds to the case when less than 20 second-level nodes joins the system in the figure 7. In fact, when a second-level node joins some VO, it generates a local DHT update and contacts only one node (its neighbor) in our solution to keep the gateway node with its neighbors. However, the fact that new nodes in MG-HDHT must contact several gateways in their VO generates additional messages. This explain that our performances are slightly better (5%) for connection of 20 second-level nodes. Our solution offers a significant maintenance cost gain when the update occurs in gateway nodes. As the number of gateway nodes increase as the gain is important since the required update messages is less with our solution. The save is 30% for connection of 90 second-level nodes and 10 gateway nodes. Certainly, update DHT messages concern both solutions but, in the MG-HDHT solution experiments the new gateway node establishes connections with all its second-level nodes. It is not the case in our solution. A new gateway generates only two additional messages (send neighbours list of this one sp_list to the nearest second-level node and report existence of this node to one node in sp_list). In a case of gateway node failure, other second-level nodes used their old gateway neighbours list (received in the connection step) to join the global DHT overlay. Then, each second-level node updates its gateway neighbours list during the result reception of resource discovery queries. Recall that in both MG-HDHT and SG-HDHT solutions, the connection of a second level node generates more additional update messages that the disconnection of this node. In fact, its connection requires an additional number of messages to establish connection between the new arrival node and the gatway nodes. An interesting future work consists to evaluate the gain in term of maintenance times.

We have also interested to the size of the list containing the neighbours of the gateway node in each VO. A gateway node sent it neighbours list, in the connection step, to all its second-level nodes. When a gateway node peer fails or changes, this list can be updated during the resource disciovery query. The enrichment of each local DHT by maintaining list of gateway neghbours increases the storage size of the DHT distributed on all peers. However, the storage space needed to store these lists is relaticely small (1KB is required to store one 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 node to a second-level

node is 25ms. By adding the sp_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.

6. 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 [8]. The first version of the Monitoring and Discovery System (MDS) in Globus [5] employed a centralized index containing the metadata describing resources. The using of web services, inspired from hierarchical models, has been explored in several research works as [5]. Although the advantage of being Open Grid Service Architecture (OGSA) [1] compliant, *i.e*., each resource is represented as a web service, this strategy is not adapted here since the dynamicity node properties in large scale Grids [8]. Last years, [14] proposed resource discovery solution based on super peer model [23]. [8, 16] classified it as 'hybrid' compared to unstructured and structured classes. Many research works [3, 4, 12, 13, 19, 24] presented advantages of hierarchical DHT systems based on super peer concept. [3] 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. [13] also explored the using of multiple Chord systems in order to reduce both latency of lookups and maintenance costs. Nevertheless, most of these works neglects the churn effects especially that [22] have demonstrated the high maintenance state needed (memory, CPU and bandwidth) when all peers in the overlay are attached to different levels of the hierarchy. [11] proposed the SG1 algorithm to find the optimal number of super peers to achieve the desired reliability while minimizing maintenance costs while [17] presented Bamboo, a DHT protocol designed to handle networks with high churn efficiently. We also cited the self organizing distributed algorithm developped [25] 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.

7. CONCLUSION & FUTURE WORK

We have proposed a hierarchical DHT solution for 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 consists of a single-gateway based hierarchical DHT solution (SG-HDHT) to discover metadata of any resource in Grids. Intra-VO queries are very efficient since they are transparent to the top level DHT lookup. More, only the arrival of a new VO requires the global DHT maintenance. Our solution addresses other super peer problems as a single point of failure by using a minimum of messages. In fact, second level 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 good lookup query performances especially when we have an important number of simultaneously resource discovery messages. It concerns both intra and inter-VOs queries. It also shows a significantly reduction of the network traffic and offers important DHT maintenance saves especially when nodes 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 especially in a large grid environment with a high number of nodes in a real platform. We would like include more realistic models of churn in our future work as to scale traces of sessions times [7] collected from deployed networks to produce a range of churn rates with a more realistic distributtion.

8. REFERENCES

1. Foster, I. (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, Version 1.0. July'04. Global Grid Forum.

2. http://Freepastry.org/FreePastry/.

3. L. Garces-Erice, E. W. Biersack, K. W. Ross, P. A. Felber, G. Urvoy-Keller. Hierarchical Peer to Peer Systems. In Proc. of ACM/IFIP Int. Conf. on Parallel and Distributed Computing '03.

4. 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.

5. The Web Services Resource Framework, http:\\www.globus.org/wsrf.

6. GRID'5000. www.grid5000.org

7. 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.

8. 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

9. N Harvey & al. Skipnet: A Scalable Overlay Network with Practical Locality Proprieties. In Proc of USITIS 2003, Seattle.

10. Gupta I & al. Kelips: building an efficient and stable P2P DHT through increased memory and background overhead. Lecture notes in computer science, 2003. Springer.

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

12. I.Martinez-Yelmo, R.C Rumín, C. Guerrero, A. Mauthe: Routing Performance in a Hierarchical DHT-based Overlay Network. Proc. of the 6th IEEE Euromicro Intern. Conf. on Parallel, Distributed and Network-Based Processing (PDP 2008), 508-515., Toulouse, France.

13. 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.

14. Mastroianni C, Talia D and Verta O. A Super Peer Model for Building Resource Discovery Services in Grids: Design & Simulation Analysis. Future Generation Computer Systems. Elsevier 20005.

15. 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).

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

17. S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling churn in a dht," in Proceedings of the Usenix Annual Technical Conference, Boston, USA. 2004.

18. Rowston A & Druschel P. 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.

19. M. S´anchez-Artigas, P. Garc´ya, J. Pujol, and A. G. Skarmeta, "Cyclone: A Novel Design Schema for Hierarchical DHTs," in IEEE Intern. Conf. on P2P Computing *(P2P 2005)*.

20. M E. Samad, F Morvan, A Hameurlain: Resource Discovery for Query Processing in Data Grids. pp 59-66, ISCA PDCCS'09.

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

22. Z. Xu, R. Min, and Y. Hu. Hieras: a Dht Based Hierarchical P2P Routing Algorithm. In Proceedings of Intern. Conf on Parallel Processing, pp 187– 194, 2003.

23. B. Yang and H. Garcia-Molina, "Designing a Super-Peer Network," in Intern. Conf. on Data Engineering *(ICDE 2003)*.

24. S Zöls, Z Despotovic, W Kellerer. Cost-Based Analysis of Hierarchical DHT Design. 6 th Intern. Conf. on Peer-to-Peer Computing (P2P 2006). United Kingdom. IEEE Computer Society, pp 233-239.

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

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