

# Scalable DHT-based Information Service for Large-scale Grids

Hai Jin, Yongcai Tao, Song Wu, Xuanhua Shi

Cluster and Grid Computing Lab

Services Computing Technology and System Lab

Huazhong University of Science and Technology, Wuhan, 430074, China

hjin@hust.edu.cn

## ABSTRACT

Current grid information service is centralized or hierarchical and proves inefficient as grid scale rapidly increases. The introduction of P2P techniques into grids breaks an encouraging path. However, frequent join and departure of resource nodes require strong self-organization capacity of system to maintain their rigid structure. Moreover, arranging identifier space for P2P nodes is knotty and has great impact on system performance. If the identifier space is too large, some nodes will be overloaded. On the contrary, small identifier space will bring the same problem as millennium bug. To address the issues, this paper proposes a scalable *DHT-based (Distributed Hash Table) Information Service (DIS)* for grid system, which organizes grid resources into a DHT ring based on *VO (Virtual Organization)*. To save the identifier space while retaining the scalability and system performance, only stable VOs can join DIS via a new DHT node, whereas volatile VOs join DIS through being the sub-domain of other VO. Experimental results show that DIS provides rapid resource query, strong scalability and high throughput, meanwhile avoiding the key node failure as well as the bottleneck problem.

## Categories and Subject Descriptors

C2.4 [Distributed Systems]

## General Terms

Design, Experimentation, Standardization.

## Keywords

Grid computing, Information service, P2P, DHT.

## 1. INTRODUCTION

Due to the intrinsic features of grids, the integrated resources belong to different VOs in which resources comply with common sharing rules [1]. Resources join and depart at any time and the performance varies significantly over time and users have little or no knowledge of all resources. As a result, one great challenge in such a wide-area grid is to build a scalable and efficient

information service framework to support the initial discovery and ongoing monitoring of the existence and characteristics of resources contributed by different VOs. Effective and efficient information service can help user identify resources with desired attributes quickly and accurately and guarantee the QoS of grid. Obviously, information service is of first important in grid. Existing grid system mainly adopts centralized or hierarchical architecture to manage resource information, such as UDDI [10], Globus MDS [2, 11]. These modes are easy to implement, but while the scale of grid increases rapidly, the root node would become a potential bottleneck and a single point of failure, resulting in long service query latency and even breakdown of entire system.

In recent years, P2P and grid are slowly converging [3, 4], encouraging the increasing application of P2P techniques into grids. To maximize the efficiency, it is helpful to consider the differences between grid and P2P systems. First, In P2P, resources are completely independent and managed in peer mode, and can enter, leave, and rejoin the system unpredictably at the whim of individual users. In grid, however, resources are often owned by research centers, public institutions, or large enterprises, in such organizations hosts and resources are generally stable or slowly changing. So, grids organize resources in VO mode and the dynamic natures mainly result from the fact that resources performance varies significantly over time (e.g., available CPU cycle, memory, storage, and network bandwidth). Second, the services provided by P2P mainly focus on file sharing (e.g., Gnutella, Kazaa), real time data transfer (e.g., telephony such as Skype), and cycle stealing (e.g., SETI@Home). On the contrary, the service that grid is concerned with is not primarily file exchange but rather direct access to various kinds of resources. Finally, P2P usually indexes and manages <key, values> pairs of resources, and naturally supports exact queries for a value (e.g., file location) given a search key, whereas resource queries in grid are mostly attribute-dependent, e.g., range query, arbitrary query, multi-attribute query, and so on.

Recently, there are growing interests in studying the use of P2P DHT technique for large scale grids [5, 6, 7, 8]. DHT-based P2P is a structured distributed system that forms a structured overlay network allowing more efficient routing than the underlying network. These researches integrate grid resources as DHT nodes in DHT ring and adopt multiple DHTs to manage attributes of resources and services, not considering the management mode of VO. In grid, VO is an integrated grid system, comprising management nodes and resource nodes. The participant nodes in a VO can not work well if any management node breaks down. In addition, the frequent join and departure of grid nodes require the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CF'08, May 5-7, 2008, Ischia, Italy.

Copyright 2008 ACM 978-1-60558-077-7/08/05...\$5.00.

real-time updating of DHT ring, consuming many system resources. Most of all, in DHT-based P2P, the identifier space assigned to nodes is specified at the beginning so that it is troublesome to arrange a moderate identifier value for inconstant system scale. Large identifier space can make some nodes be overloaded. Contrarily, small identifier space will bring forth the same problem as millennium bug.

To address the above issues, this paper proposes a scalable *DHT-based Information Service* (DIS) for grid system. DIS exploits P2P DHT technique to link grid resources into a DHT ring based on VO. To save the identifier space while retaining the scalability and system performance, the stable VO can join DHT ring via a new DHT node, the volatile VO only through being the sub-domain of other VOs. Furthermore, XML-based schema is exploited to describe various kinds of resources and services in grid environment, with which, DIS can support efficient attribute-dependent query.

The rest of the paper is organized as follows. Section 2 introduces related work and motivation. Section 3 proposes DIS. Section 4 details the DIS protocols. Section 5 introduces the scalable resource query mechanism in DIS. Section 6 conducts the performance evaluation and section 7 concludes the paper.

## 2. BACKGROUND

### 2.1 Related Work

A large amount of work has been done on P2P information service, including both unstructured and structured systems. Early unstructured P2P systems, such as Gnutella [12], use the flooding technique to broadcast resource requests in the network. In structured P2P systems, DHT is widely used. DHT-based systems [9, 13, 14] arrange  $\langle \text{attribute}; \text{attribute-value} \rangle$ , that is,  $\langle \text{key}; \text{value} \rangle$  pairs in multiple locations across the network. The *key* is the hash of a keyword from a service name or description by exploiting a hash function, such as MD5 or SHA-1, and the *value* stored in the DHT can be any object or a copy or reference to it. The DHT keys are obtained from a large identifier space, which is distributed to the nodes in a distributed and deterministic fashion. The nodes of a DHT maintain links to some of the other nodes in the DHT in terms of specific geometry mode, e.g., Bamboo DHT [15], Pastry DHT [14] and Chord DHT [9], etc.

Recently, a number of efforts have studied the using of DHT-based P2P approaches into large scale grids [5, 6, 7, 8]. In these studies, grid resources are lined into a ring via DHT. Standard DHT-based P2P system only supports exact query [9, 13, 14]. To meet the requirements in grid environment, the range query, multi-attribute query and dynamic query should be also supported. The literatures [6, 16] adopt locality preserving hashing functions that retain the order of numerical values in DHTs to support range queries over DHTs. To support multi-attribute resource query, some systems focus on weaving all attributes into one DHT [7] or one tree [8]. Others adopt multiple DHTs for attributes [6, 16, 17]. Existing approaches mainly exploit flooding technology for dynamic resource discovery, leading to high traffic overhead. In conclusion, current applications of DHT-based P2P in grids do not consider the resource management mode of VO, and proves poor when grid comprises more VOs.

### 2.2 Motivation

The use of P2P into grids greatly enhances the performance of grid. However, existing DHT-based grid information services are facing some new challenges. First, resource nodes join DHT ring via new DHT node, instead of grid VO. This leads to large amount of DHT nodes in DHT ring and makes it difficult to maintain the rigid structure. For instance, consider a grid system with  $n$  VO, and there are  $m$  resource nodes in each VO. Thus, in DHT ring, there would be  $n \times m$  DHT nodes. The join and departure of any resource node would result in the updating of DHT ring, e.g., refreshing the finger table of each node, deleting and transferring the correlative  $\langle \text{key}; \text{value} \rangle$  pairs. Moreover, it is troublesome to arrange a moderate identifier value. Large identifier space can make some nodes be overloaded. Contrarily, small identifier space will bring forth the same problem as millennium bug. Second, grid organizes resources in VO unit, and a VO is an integrated grid system. The participant nodes in a VO can not work well if any management node breaks down. Therefore, in current DHT-based grid information service, the failure of management node in a VO can result in the malfunction of several DHT nodes. Finally, to support attribute-dependent and dynamic query in grid environment, existing approaches mainly adopt multiple DHTs for each attribute and flooding technology. Take an  $n$ -attribute resource query as example: assume that the  $i$ th attribute query has  $m_i$  candidate resources, there are  $x$  candidate matches as shown in equation 1.

$$x = \prod_{i=1}^n m_i \quad (1)$$

Note that:

$$(\text{Min}(m_1, \dots, m_i, \dots, m_n))^n < x < (\text{Max}(m_1, \dots, m_i, \dots, m_n))^n.$$

Therefore, the time complexity rises exponentially as the number of attributes increases.

## 3. DESIGN OF DIS

### 3.1 DIS Architecture

DIS is built on top of Chord DHT infrastructure [9], which acts as a rendezvous network connecting multiple grid VOs. Figure 1(a) highlights the architecture of DIS. DIS links multiple VOs into a ring. In grid, VO is an integrated grid system as shown in Figure 1(b). Each VO exploits hierarchical mode and only one node can become DHT node, which acts as a bridge to link local VO into the DHT ring and consists of three modules as shown in Figure 1(c). Info Manager is responsible for managing resource information, e.g., information registering, information querying, etc. Repository maintains the data information about grid resources and services. DHT Proxy mediates between local VO and remote VOs, publishing and deleting information from the dispersed repository and performing queries across VOs.

DIS assigns DHT nodes and *key* an  $m$ -bit identifier and exploits SHA-1 as a base hash function [18]. A DHT node's identifier is obtained by hashing the node's IP address, while the *key* is the hash of keyword from service name or description. The value stored in DIS is the service information document described by XML-based schema which would be detailed in sub-Section 3.3. Similar to Chord DHT, service's key  $k$  would be assigned to the

first DHT node whose identifier is equal to or follows  $k$  in the identifier space. This node is called the successor node of key  $k$ , denoted by  $successor(k)$ . Similarly, each DHT node maintains a finger table, containing up to  $m$  entries of DHT nodes as shown in Table 1. Here  $m$  is the number of bits in the resource key/DHT node identifiers. The  $i$ th finger of nodes  $n$  is  $n.finger[i]=successor(n + 2^{i-1})$ , where  $1 \leq i \leq m$ .

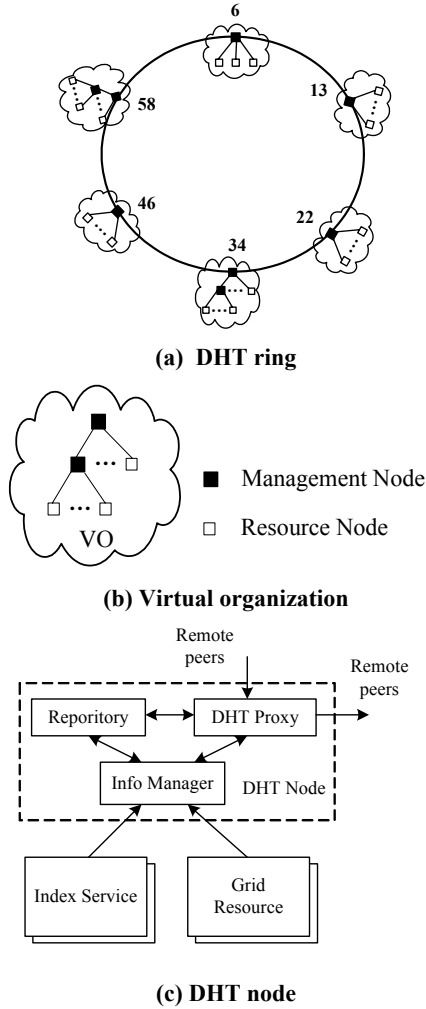


Figure 1. Architecture of DIS

Table 1. Finger table of DHT node 6

N6+1	N13
N6+2	N13
N6+4	N13
N6+8	N22
N6+16	N22
N6+32	N46

### 3.2 Scalability

Generally, hierarchical mode is subject to potential performance bottleneck and a single point of failure. For instance, in a complete  $n$ -level binary-tree, root server would maintain  $(2^n-1)$

nodes (or VO) service information. As the scale increases, the root node would be overloaded and its failure would result in system breakdown. In DHT ring mode, data items of resource information are distributed among peer DHT nodes. This provides a degree of natural load balancing. However, frequent join and departure of resource nodes require strong self-organization capacity of system to maintain their rigid structures. Moreover, the identifier space assigned to DHT nodes is specified at the beginning so that it is troublesome to arrange a moderate identifier value for inconstant system scale. To save the identifier space while retaining the scalability and system performance, in DIS, only stable VOs can join DHT ring via a new DHT node, whereas the volatile VOs join DHT through being the sub-domain of other VOs. Through the above mechanisms, DIS can scale easily, and have good load balancing and improve the resiliency to a single point of failure.

### 3.3 Uniform Service Description Schema

Based on the idea of OGSA (*Open Grid Service Architecture*), grid resources are represented by services. So, it is pivotal to provide users and applications with efficient service information description. In addition, as mentioned above, resource query in grid differs greatly from that in P2P. P2P only provides exact query for file sharing service, whereas grid needs to support range query, dynamic query and multi-attribute query, etc. To support these requirements, DIS exploits uniform service description schema as shown in Figure 2. In this schema, the elements *Cluster* and *Node* are used to represent the grid node resources. The element *Application* is utilized to describe the application services deployed in grid nodes. The element *DHTNodeName* denotes which DHT node, namely, which VO, the grid node resources and application services belong to. This facilitates the information updating while some resources and services join or depart.

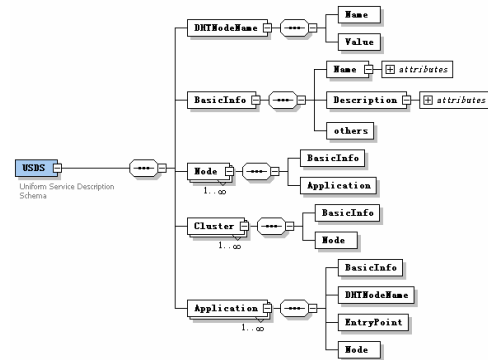


Figure 2. Uniform service description schema

## 4. DIS PROTOCOLS

In this section, we describe the main DIS protocols. The protocols specify how new resources join system, and how to recover from the failure (or planned departure) of existing DHT nodes. We assume that  $m$  is the number of bits in the resource key/DHT node identifiers.

### 4.1 Resource Joins

DIS organizes various kinds of grid resources in DHT ring in terms of VO unit. Management node in each VO serves as a DHT node in DIS. However, due to the dynamic characteristic of grid,

resources can join and depart at any time, even the entire VO. In order to guarantee the efficient resource query, DIS must ensure that the DIS ring and the information of each DHT node's finger table are up to date. In DIS, the joined resources include entire grid VO and single grid node resource and application service. Algorithm 1 shows the joining process of VO  $v$  via a new DHT node. The joining process of single grid node resource and application service is similar and for the sake of brevity is omitted here.

**Algorithm 1:** Resource joining algorithm

**Input:** joining VO  $v$

**Output:** Updated DIS ring

// generate DHT node identifier for VO  $v$  and join DIS ring.

$v.ID = \text{SHA-1}(v.IP);$

$q = \text{find\_successor}(v.ID);$

$p = q.predecessor;$

$v.successor = q;$

$v.predecessor = p;$

$\text{updateFingerTable}(v.ID);$

// transfer correlative  $\langle key; value \rangle$  pairs to DHT node  $v$ .

For each  $\langle key; value \rangle$  in  $q$

  If  $key \leq v.ID$  then

    transfer  $\langle key; value \rangle$  from  $q$  to  $v$ ;

  Endif

Endfor

// distribute resource key to correlative DHT nodes.

For each resource in VO  $v$

  If resource's type  $\in$  node resource

$key = \text{SHA-1}(\text{node.IP});$

  Elseif resource's type  $\in$  application resource

$key = \text{SHA-1}(\text{application.keyword});$

  Endif

$\text{successor}(key) = \text{find\_successor}(key);$

    maintaining key in node  $\text{successor}(key);$

Endfor

// updating the finger table of each DHT node

$\text{updateFingerTable}(n)$

  While  $n.\text{updateFingerFlag} = \text{false}$  do

    For  $i = 1$  to  $m$

$n.\text{finger}[i] = \text{find\_successor}(v.ID + 2i - 1);$

$n.\text{updateFingerFlag} = \text{true};$

    Endfor

$n = n.\text{finger}[1];$

  Endwhile

As described in Algorithm 1, when a new VO joins DIS, a new DHT node joins DIS ring and the identifier is obtained by hashing its IP address. Meanwhile, relevant DHT nodes' finger table is updated and some keys in the newly joined DHT node's successor are transferred to the new node. The resources' key in newly joined VO is dispersed onto the DIS ring. Note that the key of node resource is calculated by hashing node's IP address to ensure that the key is exclusive, and the key of application is obtained by hashing its keyword. If a new resource joins the VO, the management node will generate corresponding key and place it into corresponding DHT node. In addition, in Algorithm 1, the function of  $\text{find\_successor}(key)$  is to find the first node whose identifier is equal to or follows  $key$ , denoted by  $\text{successor}(key)$ ,

and then the  $\langle key; value \rangle$  pair will be kept in  $\text{successor}(key)$ . The function of  $\text{find\_successor}(key)$  is detailed in [9].

## 4.2 Resource Departs

In grid, resource departure can be classified into two categories: voluntary leaving and unexpected leaving. In voluntary mode, resources will inform system while leaving. Thus, system can proactively update relative information. For example, while a resource or an application service leaves their VO, the  $\langle key; value \rangle$  pair should be deleted from its DHT node. While a VO leave DIS, firstly, the  $\langle key; value \rangle$  pairs maintained in the DHT node should be transferred to its successor. Then, this DHT node is removed from DIS ring and the  $\langle key; value \rangle$  pairs belonging to the VO are deleted from corresponding DHT nodes. Finally, the pointer information of relative DHT nodes is updated. The pseudo code of voluntary resources departure is shown in Algorithm 2.

**Algorithm 2:** Voluntary VO departure algorithm

**Input:** leaving VO  $v$

**Output:** Updated DIS ring

transferring all  $\langle key; value \rangle$  pairs in  $v$  to  $v.\text{successor}$ ;

$p = v.\text{predecessor};$

$q = v.\text{successor};$

$p.\text{successor} = v.\text{successor};$  //deleting DHT node  $v$ ;

$q.\text{predecessor} = v.\text{predecessor};$

$\text{updateFingerTable}(v);$

$\text{deletingKeys}(v, q);$

// deleting the  $\langle key, value \rangle$  pairs of VO  $v$  through DHT node  $n$ ;

$\text{deletingKeys}(v, n)$

  deleting  $\langle key, value \rangle$  pairs with  $\text{resource}(key) \in v$  in  $n$ ;

$\text{broadcast}(n, \text{deleting}(v));$

To delete the  $\langle key; value \rangle$  pairs of the leaving VO, all nodes have to be traversed. In Algorithm 2, the function of  $\text{broadcast}(n, \text{deleting}(v))$  is to reach all nodes from node  $n$  without redundant messages to delete the  $\langle key; value \rangle$  pairs of VO  $v$ , which is based on the broadcast strategies proposed in [19] and will be detailed in Section 4.3.

In unexpected mode, resources departures without any notice are mainly due to the diverse failures and error conditions. To keep the system robust in the face of failures, DIS exploits heartbeat mechanism to detect unexpected resources departure and keep system up to date. Each node in DHT ring periodically sends heartbeat signal to its immediate successor, if its next node does not response, the updating program will be triggered. When the successor of node  $n$  fails, node  $n$  will find its nearest successor through finger table. Then, the failed nodes and their  $\langle key; value \rangle$  pairs will be removed from DIS ring. Finally, the relative finger tables of DHT nodes are refreshed. The algorithm is shown in Algorithm 3.

**Algorithm 3:** Unexpected VO departure algorithm

**Input:** leaving VO  $v$

**Output:** Updated DIS ring

If  $\text{heartbeat}(n.\text{successor}) = \text{null}$  then

  For  $i = 2$  to  $m$  do

    If  $\text{heartbeat}(n.\text{finger}[i]) \neq \text{null}$  then

      break;

    Endif

  Endfor

```

If i > m then
    report fault;
break;
Endif
q = n.finger[i];
While q.precessor != null do
    q = q.precessor;
Endwhile
n.successor = q;
q.precessor = n;
For k = 1 to i-1
deletingKeys(q, n.finger[i]);
Endfor
Endif

```

### 4.3 Broadcast Protocol

The broadcast protocol in DIS is based on the broadcast strategies proposed in [19]. Taking a DIS ring with  $N=2m$  nodes and  $m$ -bit identifier space as an example, each DHT node  $k$  has a finger table, with fingers pointing to nodes  $k+2^{i-1}$  ( $1 \leq i \leq m$ ). Each of these  $m$  nodes, in turn, has its fingers pointing to another  $m$  nodes. Each node transfers the information to all nodes in its finger table and these nodes also forward the information to nodes in their finger tables. In this way, all nodes are traversed in  $m$  steps. As multiple fingers may point to the same node, redundant messages should be avoided to reach the same node. To address the issue, each message contains a *Limit* argument, which is used to restrict the forwarding space of a receiving node. The *Limit* argument for the node pointed by finger  $i$  is finger  $i+1$  [19]. This approach avoids the repeated accessing of one node with same messages and reduces the response time, which ensures scalability in large-scale grids.

## 5. SCALABLE RESOURCE QUERY

### 5.1 Resource Query Types

The goal of resource query is to locate resources that satisfy a given set of requirements on their attribute values. Due to the differences between grid and P2P, grid must support various types of resource queries as follows:

- Exact query performs a lookup using a unique hash key rather than a set of search parameters.
- Range query looks for resources specified by a range of attribute values (e.g., a CPU with speed from 1.2GHz to 3.2GHz).
- Arbitrary query mainly refers to the partial phrase match or semantic search.
- Multi-attribute query refers to the problem of finding resources that are described by a set of attributes or characteristics (e.g., OS version, CPU speed) and is composed of a set of sub-queries of the above mentioned query types. In addition, the attributes of grid resources can be static or dynamic.

### 5.2 Resource Query Techniques

Grid resources are mostly described in XML-based schema. To meet the requirement of various types of resource queries in grid

environment, DIS integrates DHT query, flooding query and XML-based query techniques as follows:

- DHT query directly locates the DHT node which is the successor of the key value obtained by hashing the keyword of resource or service (see `find_successor(key)` in sub-Section 4.1).
- XML-based query retrieves the desired information by parsing the resource description document.
- Flooding query is utilized to reach some or all nodes to find services and resources. In DIS, the broadcast protocol proposed in section 4.3 is adopted to implement flooding query.

### 5.3 Scenario

In this section, we illustrate the operation of querying for one typical scenario. Considering the following example which is a typical case of multi-attribute dynamic range query in order to execute a job:

$Q = \{\text{Service}='image' \mid R \in \{R_1, \dots, R_N\} \mid \text{CpuSpeed}[R] \geq 2.0\text{GHz}$  and  $\text{RamSize}[R] \geq 1\text{GB}$  and  $\text{Utilization}10[R] \leq 0.3$  and  $\text{Free\_Space}[R] \geq 2\text{GB}\}$

This is a query that looks for image service, and the resource nodes must satisfy the following requirements: CPU speed at least 2.0GHz, at least 1GB of RAM, utilization over the last 10 min of at most 0.3, and at least 2GB of available disk space. Note that the utilization parameter is a typical dynamic attribute, which varies over time. The flowchart of this scenario is shown in Figure 3 and can be described as follows:

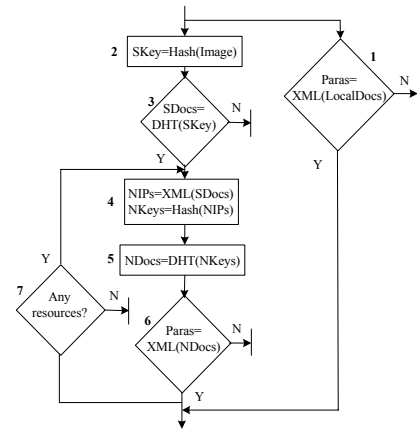


Figure 3. The flowchart of query

- (1) The query request issued by a client is firstly tackled in local VO. The service documents are parsed to find the satisfied resources. If exists, return them to user.
- (2) Meanwhile, DIS performs query across VOs. The key value is obtained by hashing the query keyword (*image*).
- (3) DIS locates the DHT nodes keeping *image* service information by means of DHT query and finds the service description documents.
- (4) The service documents containing *image* service are parsed, and the hash value of nodes IP is calculated.

- (5) DIS locates the DHT nodes maintaining the corresponding node resource information by means of DHT query and finds the resource description documents.
- (6) The node resource documents are parsed and the resource performance parameters are compared to find satisfied resource. If exists, return them to user.
- (7) If there are XML documents about ‘image’ service not tackled, go to step (4).

## 6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of DIS over a real grid environment: ChinaGrid [20]. ChinaGrid, funded by Ministry of Education of China, aims to provide the national-wide grid computing platform and services for research and education purpose among 100 key universities in China eventually. Its underlying infrastructure is the *China Education and Research Network* (CERNET), covering 1500 more universities, colleges and institutes in China. Currently, CERNET is the second largest national-wide network in China. The bandwidth of CERNET backbone is 10Gpbs. Based on CGSP (*ChinaGrid Support Platform*) [20], currently, ChinaGrid has finished integration of 200TB storage capability and 20TFlops computing capability distributed in 20 key universities, and derives some representative applications, such as bioinformatics, computational fluid dynamics, image processing, massive data processing, and remote education.

**Table 2. Number of grid nodes in each university\***

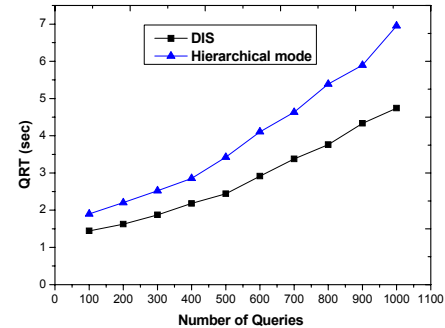
Univ.	No. of nodes	Univ.	No. of nodes
HUST	10	NUDT	8
THU	10	NEU	7
PKU	8	SDU	9
BUAA	8	ZSU	6
SCUT	8	SEU	7
SJTU	6	XJTU	8

### 6.1 Effectiveness and Efficiency

We evaluate the effectiveness and efficiency of DIS over ChinaGrid, and the testbed includes 12 sites distributed in different universities. Each site, regarded as a VO, is an integrated grid system and integrates different resource nodes, listed in Table 2, and is deployed different applications, covering bioinformatics, computational fluid dynamics, image processing, etc. These sites are respectively organized in hierarchical and DIS modes, and we perform multi-attribute dynamic query at different sites simultaneously.

\* *Huazhong University of Science and Technology (HUST), Tsinghua University (THU), Peking University (PKU), Beihang University (BUAA), South China University of Technology (SCUT), Shanghai Jiao Tong University (SJTU), Southeast University (SEU), Xi’an Jiaotong University (XJTU), National University of Defense Technology (NUDT), Northeastern University (NEU), Shandong University (SDU), Sun Yat-Sen University (ZSU).*

Figure 4 shows the QRT (*Query Response Time*) comparison between hierarchical mode and DIS mode. It can be seen that DIS outperforms hierarchical mode. This is mainly because that in hierarchical mode, each site will be accessed and all service description documents are traversed to find the satisfied resources or services. On the contrary, DIS locates the site keeping the service description documents via exact query with the hashing value of service’s keyword. Then, the correlative service description documents are parsed to find the qualified resources or services. With exact query provided by DHT technique, DIS avoids traversing all nodes and parsing each service description document, speeding up the query process.



**Figure 4. QRT vs No. of queries**

To evaluate the robustness of DIS in the presence of failure, we adopt failure injection mechanism to simulate the failure or departing of resources. The failure arrival ratio follows Poisson distribution. Table 3 illustrates the success ratio of queries at different failure arrival ratio. It shows that DIS performs better than hierarchical mode. This is mainly because that DIS is free from key nodes failure and performance bottleneck by exploiting P2P DHT technology, which improves the success ratio of queries.

**Table 3. Success ratio of queries**

Failure arrival ratio	Success ratio	
	Hierarchical	DIS
$p=0.05$	0.948	0.967
$p=0.1$	0.823	0.871
$p=0.5$	0.423	0.752

### 6.2 Scalability

In traditional DHT-based grid information service, grid resource nodes join DHT ring as a new DHT node and multiple DHTs are adopted for each attribute. Whereas, DIS organizes grid resources into DHT ring based on VO. This makes DIS easy to self-refreshing and suitable to manage large scale grid resources. We evaluate the scalability between DIS and traditional DHT-based grid information service in terms of QRT and throughput.

To simulate large-scale grid environment, the simulated testbed includes 25 nodes: 8 at *Cluster and Grid Computing Center (CGCL)* at HUST at Wuhan, and 17 at *National Hydro Electric Energy Simulation Laboratory (NHEESL)* at HUST at Wuhan. Each node in CGCL is equipped with Pentium III processor at 1GHz and 512MB memory. The nodes in NHEESL are composed of IA64 processor at 1.3GHz and 2GB memory. The nodes in

both CGCL and NHEESL are linked by 100Mbps switched Ethernet and the operating system is Red Hat Linux 9.0. In experiment, each node represents one autonomous VO, in which 40 resource nodes and 40 kinds of application services are simulated. Thus, the maximum identifier space is  $25 \times 80$  (2,000). So, in traditional DHT-based grid information service and DIS, 12-bit identifier is assigned to each DHT node and key. SHA-1 is exploited as a base hash function and its identifier space is  $2^{12}$ . We simulate up to 1000 multi-attribute dynamic queries with a waiting period of one second between receiving a request response and issuing the next query request at different VOs simultaneously. We adopt failure injection mechanism to simulate the failure or departing of resources, and the failure arrival ratio ( $p$ ) follows Poisson distribution.

### 6.2.1 Query Response Time (QRT)

Figures 5(a)-(c) show the QRT comparison between DIS and traditional DHT-based grid information service with  $p=0.05, 0.1,$  and  $0.5,$  respectively. It shows that when the number of queries and the failure arrival ratio are low, there are little difference in QRT between traditional mode and DIS. While the number of queries increases, DIS makes faster query response, especially while  $p$  increases. In addition, we find that  $p$  has an important influence on traditional mode and less on DIS. This is because

that in traditional mode, due to frequent join and departure of DHT nodes, system is busy updating correlative information (e.g., finger table, service information, etc.), which consumes more system resources and occupies much network bandwidth, accordingly increasing the query latency. DIS is based on VO, so the failure and departure of nodes in VO has less impact on the DHT ring, and only the service information of the failed nodes need to be deleted from correlative DHT nodes. DIS saves system resources and has little influence on the query response.

### 6.2.2 Throughput

Figures 6(a)-(c) illustrate the throughput comparison between traditional mode and DIS mode with  $p=0.05, 0.1,$  and  $0.5,$  respectively. It shows that DIS outperforms the traditional mode. As the number of queries increases, the system throughput is decreased. As we know, traditional mode exploits multiple DHTs for each attribute. While performing multi-attribute query, the time complexity rises exponentially with the increasing of number of attributes. In addition, the increasing of number of queries, the query latency would be enlarged and system throughput is lowered. Based on VO and uniform service description schema, DIS can efficiently deal with the join and departure of resource nodes and conduct multi-attribute query, accordingly enhancing system utilization.

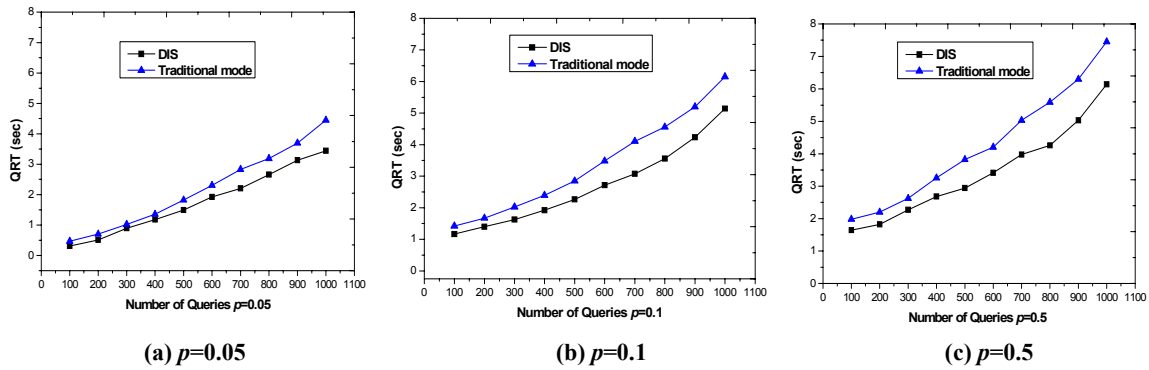


Figure 5. QRT vs No. of queries

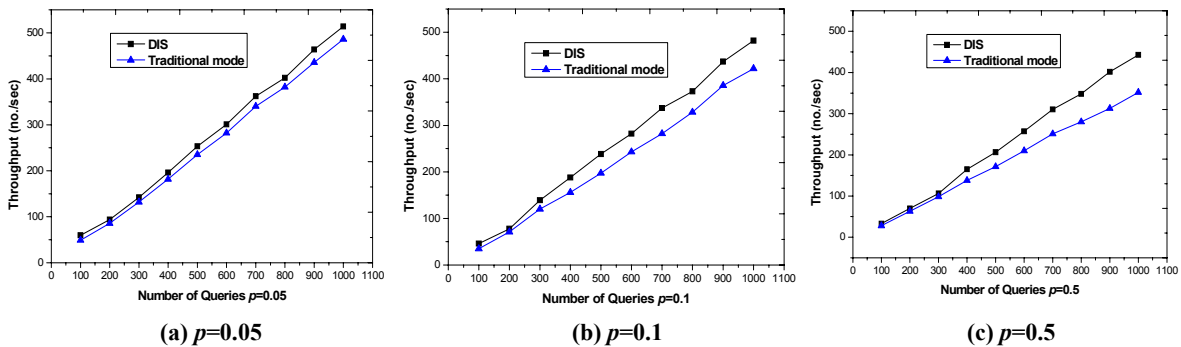


Figure 6. Throughput vs No. of queries

## 7. CONCLUSION AND FUTURE WORK

P2P and grid are two most common types of resource sharing systems widely used and have more in common. The integration of two systems could benefit both fields. In this paper, DIS is presented, which exploits P2P DHT technique to improve the efficiency and scalability of grid information service. DIS integrates the advantages of centralized/hierarchical management mode in grid VO and distributed management mode in P2P. Experimental results demonstrate that DIS has high scalability, efficiency and robustness. Ongoing work includes the deployment and further performance evaluation of DIS in larger grid environment. In P2P, both unstructured and structure modes have advantages and disadvantages. Our future research direction plans to employ the hybrid P2P mode to grid information service and further study the convergence of P2P and grid.

## 8. ACKNOWLEDGMENTS

This paper is supported by National Science Foundation of China under grant No.90412010 and ChinaGrid project from Ministry of Education of China.

## 9. REFERENCES

- [1] Foster, I. and Kesselman, C. *The Grid: Blueprint for a new computing infrastructure*. 2nd edition, Morgan Kaufmann Nov. 2003.
- [2] Czajkowski, K., Fitzgerald, S., Foster, I. and Kesselman, C. Grid information services for distributed resource sharing. In *Proceedings of HPDC'01*, 2001, pp.181-194.
- [3] Foster, I. and Iamnitchi, A. On death, taxes, and the convergence of peer-to-peer and grid computing. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003, pp.118-128.
- [4] Talia, D. and Trunfio, P. Toward a synergy between P2P and grids. *IEEE Internet Computing*, vol.7, no.4, July/Aug. 2003, pp.94-96.
- [5] Iamnitchi, A., Foster, I. and Nurmi, D. A peer-to-peer approach to resource discovery in grid environments. *Technical Report TR-2002-06*, University of Chicago, 2002.
- [6] Cai, M., Frank, M., Chen, J. and Szekely, P. MAAN: A multi-attribute addressable network for grid information services. *Journal of Grid Computing*, 2(1):3-14, 2004.
- [7] Oppenheimer, D., Albrecht, J., Patterson, D. and Vahdat, A. Scalable wide-area resource discovery. UC Berkeley *Technical Report, UCB/CSD-04-1334*, July 2004.
- [8] Basu, S., Banerjee, S., Sharma, P. and Lee, S. NodeWiz: Peer-to-peer resource discovery for grids. In *Proceedings of IEEE/ACM GP2PC'05*, Cardiff, UK, May 2005, pp.213-220.
- [9] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M. F., Dabek, F. and Balakrishnan, H. Chord: A scalable peer-to-peer lookup protocol for Internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17-32, Feb. 2003.
- [10] UDDI V3.0.2 specification. [http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm)
- [11] Globus MDS. <http://www.globus.org/toolkit/docs/4.0/info/>
- [12] Gnutella Protocol Development. [http://rfc-gnutella.sourceforge.net/src/rfc-0\\_6-draft.html](http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html)
- [13] Ratnasany, S., Francis, P., Handley, M., Karp, R. M. and Shenker, S. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM'01*, San Diego, USA, 2001, pp.161-172.
- [14] Rowstron, A. and Druschel, P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, Heidelberg, Germany, Nov. 2001, pp. 329-350.
- [15] Rhea, S., Geels, D., Roscoe, T. and Kubiawicz, J. Handling churn in a DHT. In *Proceedings of the USENIX Annual Technical Conference*, June 2004, pp.127-140.
- [16] Andrzejak, A. and Xu, Z. Scalable, efficient range queries for grid information services. In *Proceedings of 2nd IEEE Int. Conf. on Peer-to-peer Computing (P2P'02)*, Sweden, Sep. 2002, pp.30-40.
- [17] Spence, D. and Harris, T. XenoSearch: distributed resource discovery in the XenoServer open platform. In *Proceedings of HPDC'03*, Washington, USA, June 2003, pp.33-40.
- [18] FIPS 180-1. Secure Hash Standard. U.S. Departure of Commerce/NIST, National Technical Information Service, Springfield, VA, Apr.1995.
- [19] El-Ansary, S., Alima, L. O., Brand, P. and Haridi, S. Efficient broadcast in structured P2P networks. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, 2003, pp.304-314.
- [20] Jin, H. ChinaGrid: making grid computing a reality. *Digital Libraries: International Collaboration and Cross-Fertilization - Lecture Notes in Computer Science*, Vol.3334, Springer-Verlag, Dec. 2004, pp.13-24.