A Constellation Resource Discovery Model Based on Scalable Multi-tape Universal Turing Machine*

Yinfeng Wang, Xiaoshe Dong, Hua Guo,
Xiuqiang He, and GuoRong Liu

School of Electronics and Information Engineerin, Xi'an Jiaotong University,
Xi'an, 710049, China
xsdong@mail.xjtu.edu.cn
{wangyf, guohua}@mailst.xjtu.edu.cn

Abstract. Constellation resource discovery model is a novel model for discovering the dynamic resources in Grid. In constellation model, we propose a new Scalable Multi-tape Universal Turing machine (SMUTM) to present the processes of simultaneous discovery tasks on the constellation nodes, and formally describe the usability of the Grid resource based on the SMUTM. In this research, we design an algorithm used in the constellation resource discovery, which guarantees overlaying all the nodes in Grid and optimizes the network cost. Preliminary simulation shows that this algorithm complexity is linear; the average response time can be reduced by 12%. Constellation model ensures the QoS of the resource discovery in the scalable and dynamic Grid environment.

1 Introduction

A basic service in Grid is resource discovery: given a description of resources desired, a resource discovery mechanism returns a set of resources that matches the description [1]. MDS [2] is the typical centralized resource discovery mechanism. MDS organizes the resources into a layered tree structure. As Grid sizes increase, some of centralized or hierarchical functions should be decentralized to avoid the bottlenecks and guarantee scalability.

Peer-to-Peer (P2P) model can be adopted to overcome the deficiency of the centralized or hierarchical mechanisms. In order to solve the flooding-based search mechanism and topology mismatch problem, location-aware topology matching (LTM) was proposed [3]. But other problems still exist. In P2P, each peer acts as client and server to implement resource discovery. Although peers have different performance and different network connections, they have to do the same computation tasks and maintain the same size record of overlay connections, which unnecessarily consume Grid resources and increase the response time.

Super-Peer network [4] is growing popularity, which combines the efficiency of autonomy, load balancing and robustness of distributed search. However, in Grid,

---

[*] This research is supported by 863 project of China (Grant No.2002AA104550) and China Education and Research Grid (Grant No.CG2003-CG008).

H. Zhuge and G.C. Fox (Eds.): GCC 2005, LNCS 3795, pp. 633 – 644, 2005.

service node is organized by Virtual Organizations and future interconnection environment will be a large-scale human-machine environment [5]. Using P2P system, the management and organization of nodes become difficult with the enlarging of the scale of Grid, and security of Grid is also difficult to be guaranteed. Because the nodes join and leave Grid frequently, the overlay network built on them is unsteady, and need to rebuild at a regular intervals, so the number of hops required covering all the nodes and the quality of service (QoS) of services in Grid is uncertain.

In this paper, we adopt the constellation model [6] for the resource discovery to resolve the problems mentioned above. Just like the constellation in the sky can guide the voyage, the constellation in Grid can navigate resource discovery.

A Scalable Multi-tape Universal Turing Machine (SMUTM) is proposed in this paper. SMUTM is used to denote the behavior of the concurrent tasks and their relations in dynamic Grid environment, and to describe the Grid resources' properties for the constellation model. We design a new discovery algorithm to reduce the response time of resource discovery, and it provides benefits such as anonymity and using less bandwidth.

The rest of the paper is organized as follows. Section 2 introduces the constellation model. Section 3 gives a formal description of resource discovery and the properties of the Grid resources. Section 4 discusses a new discovery algorithm and its optimization solutions. Section 5 analyzes the results of the simulation on a prototype. Section 6 draws a conclusion.

2 Architecture of Constellation Model

Reference [6] proposes a Constellation model for Grid resource management, which addresses the criterions for node selection and redundancy. The nodes in Grid constitute multiple constellations according to geographically dispersed or application policy. Nodes in a constellation can be divided into fixed star, planet and meteor nodes. Each planet and meteor node must connect to at least one fixed star node, and the fixed star nodes constitute a steady formwork of the constellation. Resources are published on the constellation. Constellations are connected together to form a virtual galaxy overlay network.

The architecture of the resource discovery based on constellation model is shown in figure1. The constellation layer is the core layer. Galaxy overlay network can be built on constellation layer in various ways. Nodes in the service provider layer could choose any constellation to publish its services. Based on some policies such as the load balancing, the services will be finally published on a chosen fixed star or planet node. User can submit requests via some portal or other job-entry interface. The portal shown in the figure could then transfer the request to an appropriate local constellation according to the location of the user.

Various communication protocols and security policies can be used inside the constellation. A constellation could be a little laboratory or large-scale industry. It can extend to multiple organizations or multiple physical locations.

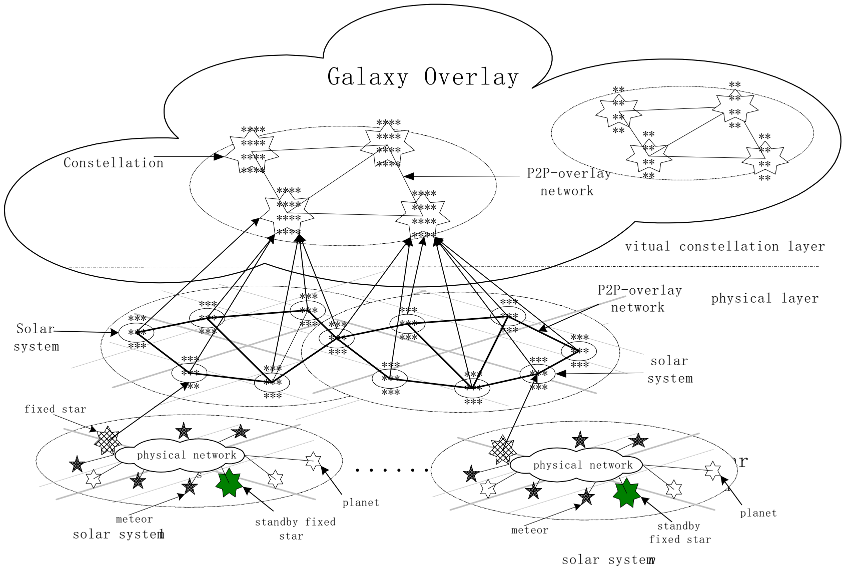

Fig. 1. The Architecture of Constellation Model

3 Formalization Descriptions

3.1 The States Transition of Discovery Process

Automata is a model of computation that can record the states at any given time and identify the input of system. It can be used to describe the process of resource discovery in Grid, discovery events and transitions of the states as a result of an action.

Definition 1: Automata of resource discovery is based on the constellation model $M=\{\{s_0,s_1,s_2,s_3,s_4\},\{0,1,2\},\delta,\{s_0\},\{s_0,s_4\}\}$ where $K=\{s_0,s_1,s_2,s_3,s_4\}$, K is a finite set of states; $\Sigma=\{0,1,2\}$,$\Sigma$is an input alphabet; $\delta$is a transition function from $K\times\Sigma$to K; $s_0\in K$ is the initial state. $F=\{s_0,s_4\}$, F is the set of final states, $F\subseteq K$.

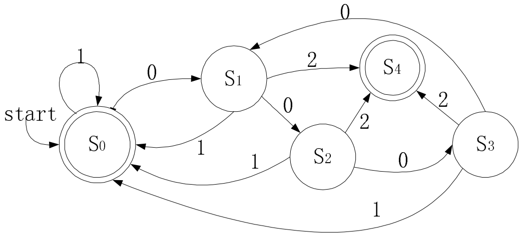

Fig. 2. States diagram represent the transition of states in M

If the value returned by the query function is true, it means that resource discovery is successful, and *M* is driven to s0. State s1 denotes choosing the neighbor constellation. State s2 denotes request's being transferred to other constellations. State s3 denotes querying on the nodes in the same constellation. *M* will be driven to state s4 if none of qualified resources are found or the query is abandoned for any reasons.

In *M*, the letter *0* shows doing querying, while *1* stands for query is successful and *2* denotes exiting for errors. The transitionδ: $S\times\sum\rightarrow S$ is shown in table 1. In practice *M* must be restarted when a new query request is submitted to the system, no matter at which state *M* is before.

Table 1. The state transition δ

$\delta : S\times\Sigma\rightarrow S$	0	1	2
s_0	s_1	s_0	/
s_1	s_2	s_0	s_4
s_2	s_3	s_0	s_4
s_3	s_1	s_0	s_4
s_4	/	/	/

3.2 Scalable Multi-tape Universal Turing Machine

Universal Turing machine U is suitable to analyzing single algorithm or program. Intuitively, U takes two arguments, a description of a machine "*M*", and a description of an input string "*w*". We want U to have the following property: U halts on input "*M*", "*w*" if and only if *M* halts on input *w*. It is U("*M*","*w*")="*M*(*w*)" [7]. But resource discovery in Grid includes many concurrent tasks and actions, and various query policy could be used in the constellations.

The Turing machine is not durable. The history computation cannot be utilized, so we need to modify U in order to describe the behavior of the concurrent tasks and their relations. We added work-tapes to U for recording the previous output. Multiple input and output tapes are added so that there will be a durable work-tape for each output one. The output of the computation will be recorded as part of the input for the next computation.

Definition 2: SMUTM is a finite set of sextuples. A sextuple is an expression of one of the following forms consisting of six symbols:

(1) $q_i\ s_j\ w_j\ s_k\ w_i\ q_l$; (2) $q_i\ s_j\ w_j\ R\ w_i\ q_l$; (3) $q_i\ s_j\ w_j\ L\ w_i\ q_l$;

We intend a sextuple of type (1) to signify that in state q_i scanning symbol s_j and w_j, the device *M* will print s_k and go into state q_l, meanwhile record w_i on the work-tape. Similarly, a sextuple of type (2) signifies that in state q_i scanning s_j and w_j, the device *M* will move one square to the right and then go into state q_l, and record w_i on the work-tape. Finally, a sextuple of type (3) is like one of type (2) except that the motion is to the left.

We can use the SMUTM model to simulate the process of resource discovery. The leftmost end of the tape is always marked by a special symbol denoted by $\triangleright$. Computations of resource discovery take place in all k tapes of the SMUTM. Resource

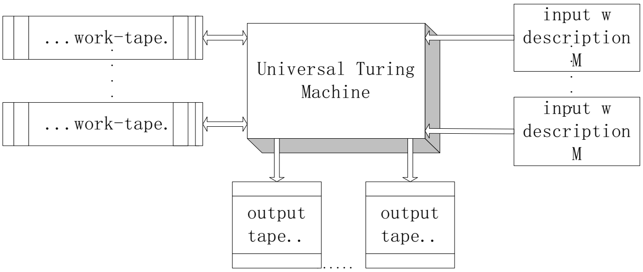

Fig. 3. SMUTM model

discovery automata *M* in definition 4 could simulate the working process of each tape head. A configuration of such a machine must identify the state, the tape contents, and the head position in each of the k tapes. A configuration is a member of $K \times (\triangleright \Sigma^* \times (\Sigma^*(\Sigma - \{B\}) \cup \{e\}))^k \times (\Sigma^* \times (\Sigma^*(\Sigma - \{B\}) \cup \{e\}))^k$. *B* denotes blank symbols. The symbol *e* denotes ending.

SMUTM reads string *Mw* as its input. If the last required resource could be found in the next computation, the information on the work-tape will be copied to the output tape and regarded as the output of this discovery task.

Computations can be performed concurrently when new requests come. The discovery process should return all the resources that match a requirement description

3.3 Formal Definition of Resource′s Property

In the OGSA framework each resource is represented as a Grid Service [8], thus the Grid can be regarded as a scalable set of Grid services. In order to research the characteristics of resources in Grid, we will give a formal definition of them and introduce some theorems and their proofs.

Definition 3: A pair *R* denotes available resource (*M, W*). *M* is a SMUTM; *W* is a nonempty finite services set. *Available* means resources can accept requests from users and return the acceptable results.

SMUTM reads the request $\omega$and writeω' on the tape, If$\omega' \in \boldsymbol{W}$, which indicates that a result can be returned to the user after the service is invoked. If ω is a request for a certain resource, when SMUTM reads it, the result will appear on output tape. If $\omega' \notin \boldsymbol{W}$, then SMUTM will reject the request and the service become available.

Theorem 1: An available resource has the following 4 characteristics: service is stateful within a lifecycle; service can halt even when exceptions happen; service is self-reproducing; and service has standard interfaces.

Proof: We say SMUTM computes lifetime function *f*. $f:(\Sigma - \{\triangleright, B\})^* \mapsto (\Sigma - \{\triangleright, B\})^*$, if the following two conditions hold for all $w \in (\Sigma - \{\triangleright, B\})^*$

a) $\exists N$, depending on SMUTM andω,such that there is no configuration C satisfying $(s, \triangleright \underline{B}\, \omega) \vdash_M^N C$

b) $(s, \triangleright \underline{B}\, \omega) \vdash_M^* (h, u\underline{a}v)$ if only if $ua = \triangleright B$, and $v = f(w)$

We say SMUTM compute f; N is an "upper bound" depending on the machine and the input (condition (a) above). There is no computation continuing after N steps. It demonstrates that SMUTM for all resource discover and f halt. Use Step-Counter Theorem [9]. Adding a step counter, which limits the maximum resources that requestor occupies, such as CPU time, disk space etc. One simply "runs" program number y for up to t steps.

Use the Recursion Theorem corollary $\exists e, for\, \forall x, \phi_e(x) = e$ [9]

This result means the program with number e consumes its "environment" and outputs a copy of it. When users release the resource, the service will be self-reproducing.

Use the Fixed Point Theorem [6].

Let $f(z)$ is a computable function. $\exists e, \forall x, \phi_{f(e)}(x) = \phi_e(x)$.

This means all the resources must provide a standard interface (GridService [8]).

Corollary 1: A P2P network can be represented by an undirected graph [10]. Resources discovery issue is how to cover all the nodes in the Grid. Services can be published on some nodes; if we can find all the nodes in Grid, all exist services will be discovered. Moreover we can discover the resources using minimum nodes set and reduce the consuming of Grid resources. But the *VERTEX-COVER* [9] is NP-complete problem. APPROX-VERTEX-COVER algorithm returns a vertex-cover that is almost twice the size of an optimal cover.

3.4 Formal Describe the Constellation Model Discovery

Resources in constellation model can be denoted by the formal definition in Section 3.3. A node in constellation can be treated as a SMUTM according to definition 3. The services will be available during the using of the requestor, which conform to Theorem 1.The fix-star and planet nodes in a constellation can be organized into various models, such as C/S or P2P. If only any fixed star node accepts a request, the whole constellation could deal with it ultimately.

- The discovery execution time T (n) is presented in the following expression

$$T(n) = \begin{cases} \Theta(1), & if\ n=1 \\ Max(T(n/m)) + \Theta(n), & if\ n>1 \end{cases} \tag{1}$$

Where n denotes that a request has n items; $\Theta(1)$ is the execution time of an undivided request. If a request can be divided into several sub-requests, each sub-request will have $\lfloor n/m \rfloor$ items. $\Theta(n)$ stands for the merging result time that is only relative to n. The execution time of a resource discovery is determined by the last task, which can avoid one task blocking the following.

- The architecture of constellation model can be expressed as

$$\Phi = \{SMUTM, Network, QoS\text{-}Radius\}. \tag{2}$$

Φ means the set of all performance features of constellation model, for each element $\phi_i \in \Phi$ quantitatively describe the tolerable variation in ϕ_i. From the point of view of Operating System, Grid Service typically consists of one or more processes; so let *SMTUM* represent the service and data of a constellation. *Network* denotes the overlay network connecting the constellations together according to different optimizations. The values of all of the system and environment parameters may impact the QoS performance features called perturbation parameters [11]. *QoS-Radius* is a two-element perturbation vector $<a_i^{min}, a_i^{max}>$. gives the bounds of the tolerable variation in ϕ_i. Take response time as an example, the acceptable variation is up to 50 percent of the estimated discovery time. ϕ_i is the time required to finish the assigned discovery task by the *i*th constellation, and its corresponding$<a_i^{min}, a_i^{max}>$could be <0, 1.5×*(estimated response time)*>.

4 Discovery Algorithms and Optimization

In the traditional P2P system such as Gnutella, the query response is forwarded back along the reverse path that carried the incoming query message, which can guarantees the anonymity of query source and avoids the bombard temporary direct connection requests. While this method has the following shortages:

- Firstly, if the status of nodes is dynamic or the network is unsteady, the discovery response could not return for the failure of the nodes or network.
- Secondly, it will take a long time for the responder's return when the path is very long. Because the response results should be stored in each node in the path and then sent to the next node, a lot of unwanted operations are performed, which probably make the response time unacceptable;
- Thirdly, the path that incoming query message follows is the same as query response message, which will likely to cause network congestion. While the other paths are free.

Apply constellation model to resource discovery, the query source such as from planet can take the fixed star as proxy, therefore keep its anonymity. The fixed star can support more direct connection that reduces the aggregate bandwidth. Hence, using equation 1 and 2 we bring forward new algorithms in that the response messages are sent back to the user directly.

4.1 Non-Original-Trace Return (NOTR) Algorithm

The resource discovery is represented as a service. Users create an instance of the resource query service with a randomly created name *RandomGSH*. The instance has a service data named *ServicesFound* recording the query result, and a method named *RecordServices* to record the query result in ServicesFound.

A request is firstly submitted to a constellation called the **beginning constellation**, and then transferred to other constellations called the **midst constellations**. This algorithm includes operations performed on the two separate parts.

1. Procedures on the beginning constellation:

```
Beginning constellation: ( Do local querying, and then judge the result of the local query)
If (no qualified services are found )
{
    Add RandomGSH into the query request message;
    Update the request message;
    Send the message to filtered neighbors of this constellation;
    While (the time to stop querying is not reached )
    {
        If (ServicesFound is updated )
        { Notify the query result to the user application; }
    }
    Destroy the RandomGSH instance;
    Stop listening ;
    Stop resource querying;
}//end if;
Else
{
    return the query result to requestor；
    stop resource querying；
}
```

2. Procedures on the midst constellations:

```
Midst constellations: ( Do local querying, and then judge the result of the local query)
If (no qualified services are found)
{
   Update the query request message;
   Send the message to the filtered neighbors of this midst constellation;
   Exit;
}
Else
{
   Invoke the method RecordServices of the instance RandomGSH to update the ServicesFound
   Exit;
}//end
```

RandomGSH is sent to neighbor constellations as part of the request. When some qualified services are found in any constellation, the method *RecordServices* of the *RandomGSH* instance will be invoked to modify *ServicesFound* directly, thus the response is returned to the beginning constellation without passing along other constellations. Then the Notification-mechanism in GT3 will notify the result to requestor immediately. Therefore, the response time of the resource discovery could be decreased.

4.2 Constellation Response Time Optimizations

In [12], messages that will be delivered to the same node are buffered and sent in a single packet at regular time intervals, while messages with the same header are merged into a single message with a cumulative body. This protocol efficiently reduces the number of Grid services operations. In order to further reduce the response time, some optimization solutions are proposed as follows:

- All the constellations in the path that incoming request followed must be recorded, and this information will be sent to neighbors within the request. If some neighbors have been present in the experienced path, the request will not be sent to them.
- A request will carry the name of its beginning node and the serial number of itself in the beginning node as its mark. If a constellation had handled with the same request, it will simply discard the request without doing anything. Otherwise it will cache the mark of the request before processing in local.

5 Prototype Simulations

The prototype system of constellation model is implemented in the National High-performance Computing Environment of Xi'an Jiaotong University. Overlay network simulator on the constellation layer adopts the 3-CCC (3-node cycle cube-connected cycles) [13] connected-network model.

5.1 NOTR Algorithm

The query request messages are produced randomly. We will examine the response time of the prototype, which adopts the (NOTR) algorithm and the follows the route of original trace return algorithm respectively. Moreover the average time of all the query results is also calculated. The result is shown in figure 4.

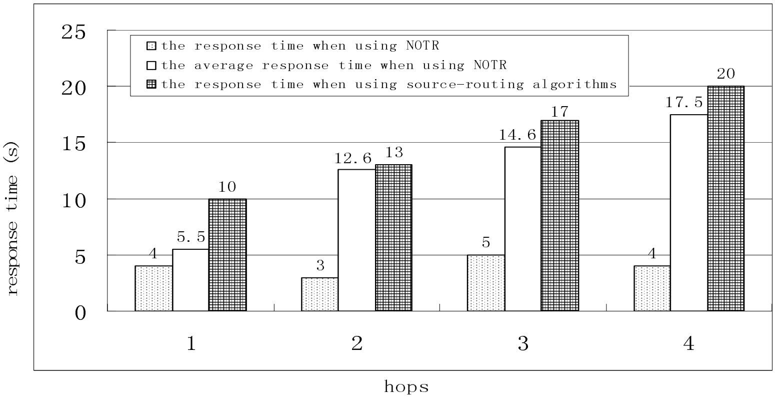

Fig. 4. The comparison response time while adopting the NOTR algorithm and others

The response time only depends on the location of the qualified service firstly found in NOTR. But the response time is linear with the biggest hops when adopting the traditional algorithm. The NOTR algorithm reduced the response time greatly. The average time of query results is also reduced nearly by 12%.

5.2 Constellation Network Load Optimization Solution

The scale of the constellation layer is as much as 80 constellations. We send to the prototype with an unfruitful query request message, to which no services will be matched. This unfruitful request is transmitted among the nodes in the constellation layer. We measure the network load of the constellation layer using the times of the request's being transmitted. Figure 5 shows the times of transmitting when using simple flooding algorithm and when using our optimization solutions.

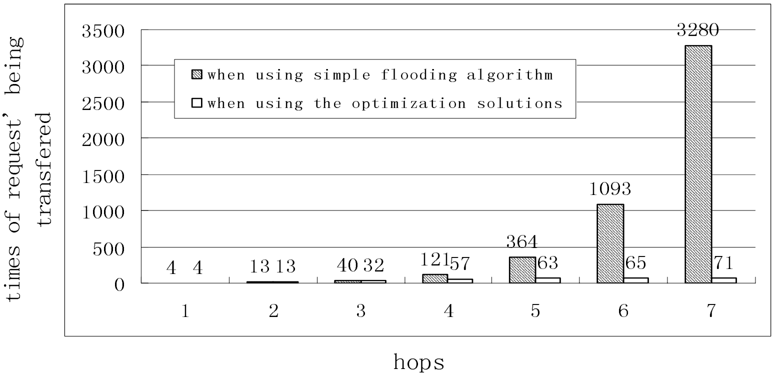

Fig. 5. Transmitting times comparison

The times increase exponentially with the increasing of constellation scale when only simple flooding algorithm is used. But optimization solutions can limit the amount of times to a limitary range, which will not exceed the number of constellations. So the optimization solution can efficiently reduce the network load.

5.3 Prototype Evaluation

With the advent such as Dense Wavelength Division Multiplexing (DWDM) in fiber-optic transmission and so forth, the overlay network based on it is also reliable. Resource discovery mechanism is based on the fixed-star nodes. The dynamic nodes joining and leaving do not impact the entirety of resource discovery. We choose the CCC model for the following reasons:

- The node degree of CCC is always 3. "From crawls of the Gnutella network performed in June 2001, we found the average outdegree of the network to be 3.1" [4]. So the adoption of the CCC model can represent the characteristics of network, and overlay network will be easy to construct.
- The CCC model is characterized by its good scalability.

Table 2. The CCC model attributes[13]

Model	Degree	Diameter	Scale
CCC	3	2k–1+[k/2]	N = k×2k

The letter k denotes the dimension of the CCC model. N is the number of nodes in the constellation layer. Given N=100; 1000; 10,000; 100,000; then k=5; 8; 10; 13. The diameters are 11; 19; 24; 31 correspondingly. The number of hops required to cover all the nodes in the constellation layer equals the diameter adding 1.

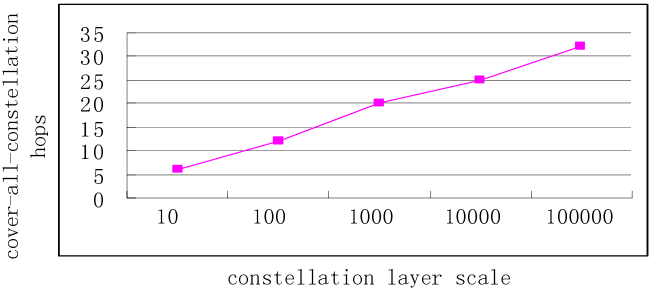

Fig. 6. The least number of hops with the scale of the constellation layer

In terms of the sharp increasing of the scale of the constellation layer, the minimal number of hops can be acceptable. The complexity of the resource discovery can be limited in an acceptable range for a relatively reliable environment as shown in figure 6. Only the QoS of resource discovery service can be guaranteed if there are qualified services in Grid could be found.

6 Conclusions

We have presented the constellation model for the Grid resource discovery. This model has the following advantages: it could avoid the network congestion resulted by flooding, cancel the limit of discovery hops; it has good scalability; resources can join dynamically; the optimized path could be found after the galaxy overlay network is built. We propose a scalable multi-tape universal Turing machine model, and use the SMUTM model to formally describe the process of resource discovery and properties of resources. To investigate issues in QoS in the constellation model is among our further research goals.

References

1. Adriana Iamnitchi, I.Foster: On Fully Decentralized Resource Discovery in Grid Environments. Second International Workshop on Grid Computing, 2001
2. The Globus Toolkit 3 Programmer's Tutorial3.
3. Yunhao Liu,Li Xiao,Xiaomei Liu,Lionel M. Ni ,Xiaodong Zhang: Location Awareness in Unstructured Peer-to-Peer Systems. Parallel and Distributed Systems, IEEE Transactions on Feb. 2005, Volume: 16, Issue: 2, pp. 163- 174
4. Beverly Yang, B. Garcia-Molina, H: Designing a super-peer network. Proceedings. 19th International Conference on Data Engineering, March 2003, pp 49 - 60

5. H.Zhuge, The Future Interconnection Environment, IEEE Computer, 38 (4), 2005, pp. 27-33
6. Yinfeng Wang, Xiaoshe Dong, Xiuqiang He, Hua Guo,Fang Zheng and Zhongsheng Qin: A Constellation Model for Grid Resource Management. The Sixth International Workshop on Advanced Parallel Processing Technologies (APPT 2005), LNCS 3756, pp 263-272, 2005
7. Harry R.Lewis, Christor H.Papadimitrion：Elements of the theory of computation 2nd Ed. Prentice-Hall, Inc, 1998, pp. 247-250
8. I.Foster, H. Kishimoto: The Open Grid Services Architecture, Version 1.0 June 7, 2004
9. M.D Davis, E.J Weyuker: Computability, Complexity and Languages. Academic Press, 1986, pp.59-60,66-67
10. Bin Xiao, Jiannong Cao, and Edwin H.-M. Sha: Maintaining Comprehensive Resource Availability in P2P Networks, GCC 2004, LNCS 3251, pp. 543-550
11. Ali,S. Maciejewski, A.A. Siegel, H.J. Jong-Kook Kim: Measuring the robustness of a resource allocation. Parallel and Distributed Systems, IEEE Transactions on July 2004, Volume: 15, Issue: 7, pp. 630- 641
12. Domenico Talia and Paolo Trunfio：Web Services for Peer-to-Peer Resource Discovery on the Grid. DELOS Workshop: Digital Library Architectures 2004: pp.73-84
13. F.P.Preparata and J.Vuillemin: the cube-connected cycles: A versatile network for parallel computation, Comm, ACM, Vol.24, No.5, 1981. pp.300-309