

# Virtual and Dynamic Hierarchical Architecture for E-science Grid

Lican Huang, Zhaohui Wu, and Yunhe Pan

Department of Computer Science and Engineering, Zhejiang University  
Hangzhou 310027, PRC  
{lchuang,wzh}@cs.zju.edu.cn, yhpan@sun.eju.edu.cn

**Abstract.** This paper presents an e-Science Grid architecture called as Virtual and Dynamic Hierarchical Architecture (VDHA). VDHA is a decentralized architecture with some P2P properties. VDHA has scalable, autonomous properties, and full and effective service discovery. The synchronization can be easily achieved by using VDHA. Further more, VDHA has the security architecture fulfilling the requirements of Grid. VDHA was adopted by Chinese University e-Science Grid as its architecture. In this paper, the advantages and several protocols of VDHA are also discussed.

## 1 Introduction

“e-Science is about global collaboration in key areas of science, and the next generation of infrastructure that will enable it” [1]. e-Science enables scientists to generate, analyze, share and discuss their insights, experiments and results in a more effective manner. The main characteristics of e-Science are coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations called as VO [2], and dynamically involving a large number of nodes generally distributed globally in geography.

The computing architecture of e-Science is usually based on Grid [2]. The newly proposed Open Grid Services Architecture (OGSA) [3] of Grid integrates the so called computational/data Grid architecture [2] with Web services [4], and this architecture is a service-oriented architecture. From data exchange point of view, the architecture of e-Science can be classified from raw data to knowledge data as three-layered services architecture: computation/data service Grid, information service Grid and knowledge service Grid [5].

There are two major types of computing models. The prevalent client-server model is suitable for slim hosts as clients, especially mobile apparatus such as palm computers, but it may cause a performance bottleneck and an entire breakdown due to a single point of failure. Peer to Peer (P2P) model [6,7] can solve the scalable and fault tolerance problems, but it has some challenges such as security, network bandwidth, and architecture designs. We present Virtual and Dynamic Hierarchical Architecture (VDHA) (some ideas were formed in the paper [8]) to combine the advantages of the above two models and avoid their shortcomings.

VDHA is suitable for autonomous systems such as Internet, which are prerequisite for scalability; VDHA has easy authentication and authorization schema for the requirements of Grid; VDHA can fully and exactly discover services; VDHA makes synchronization easy. In this paper, we describe VDHA, its advantages, and several protocols related to this architecture.

The structure of this paper is as following: Sect. 2 describes VDHA, related protocols and several properties; Sect. 3 gives out an example about virtual cooperative research projects granted by China Educational Ministry, and finally we give out conclusions.

## 2 Overview of VDHA

We define the kind of Grids mainly for scientific research as e-Science Grid. Its nodes are usually located in the Universities or Institutes. The nodes are relative stable compared with other type Grids. The Universities are always formed into virtual organizers according to specific domains, and several virtual organizers share a more general common domain. Apart from general computers, there are many electronics equipment such as PDA, sensors and so on to access the e-Science Grid. According to these properties, we proposed the network architecture of e-Science Grid (see Fig. 1) and VDHA architecture to satisfy these requirements. In network architecture of e-Science Grid, there are a core circle formed by e-Science Grid nodes, and a surrounding circle, which is consisted of desktop computer, mobile computer, palm, PDA, sensors, other networks, etc. The core circle uses VDHA as its architecture. The users can use any apparatus such as PDA, palm, mobile phone or Grid node as host machines and can login into Grid system anywhere. More than one user can share a same computational apparatus. The other network such as telecommunication networks such as GSM and so on can access the system via an entrance grid node. The Grid node has sole IP address, which is used for its identification ID. The hierarchical virtual group is identified by its name. The virtual groups are generally hierarchically arranged according to the related domains.

### 2.1 Description of VDHA

VDHA is a virtual and dynamic hierarchical architecture (see Fig. 2) in which Grid nodes are grouped virtually. Nodes can join the group and leave the group dynamically. The groups are virtually hierarchical, with one root-layer, several middle-layers, and many leaf virtual groups (these groups are called VOs). Among these nodes of VOs, one (just one) node (called as gateway node) in each group is chosen to form upper-layer groups, from the nodes of these upper-layer groups to form upper-upper-layer groups in the same way, and this way is repeated until to form one root-layer group. In the same group all nodes are capable to be gateway node. Gateway node is the node which is not only in low-layer group, but also in up-layer group. Gateway nodes will forward the low-layer group's status information to all the nodes in the up-layer group, and distribute

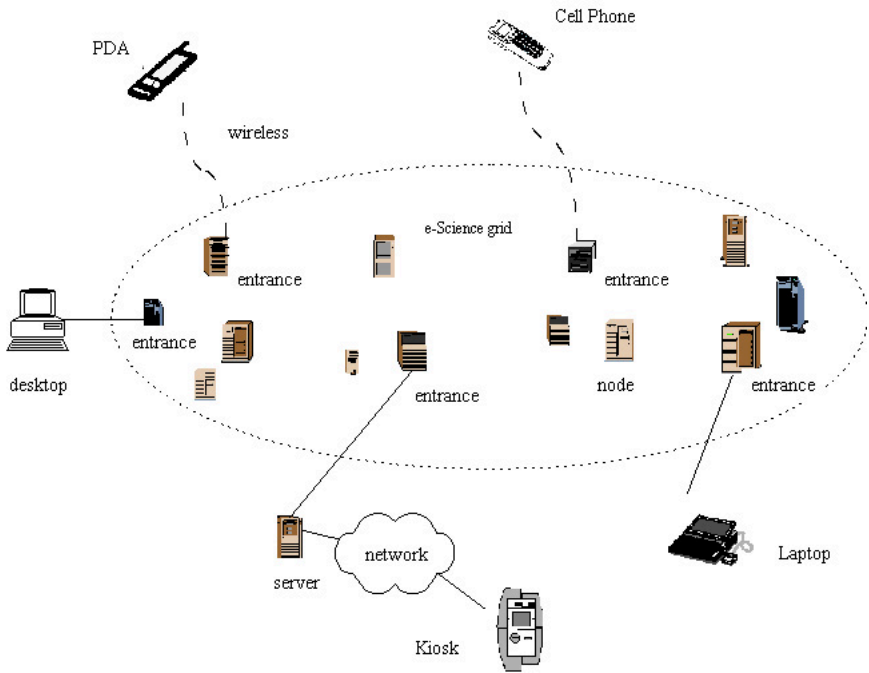


Fig. 1. Network architecture of e-Science grid

the upper-layer group's status information to all the nodes in the lower-layer group. The numbers of nodes in a VO can be dynamically changed by the way that the node can dynamically join and leave the VO. A VO may join and leave the Grid system as a whole, and this autonomous property makes the large scalable systems possible.

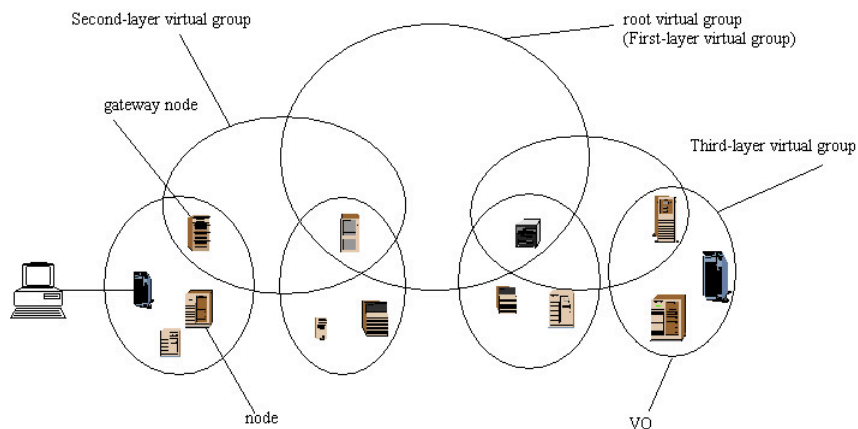
## 2.2 Formal Definition of VDHA

**Definition 1.** *Grid node (symbol as  $p$ ) is the node in the Grid system. All  $p$  form a set  $PS$ , that is,  $PS = \{p_i | i \in N\}$ ,  $N = \{1 \dots n\}$ , here,  $n$  is the number of the Grid nodes, each  $p_i$  has ID (usually Internet IP address).*

**Definition 2.** *Entrance node (symbol as  $ent$ ) is a Grid node, which is an entrance point for users to login into the Grid system.*

**Definition 3.** *Owned node (symbol as  $ow$ ) is a Grid node, which manages the Users.*

**Definition 4.** *User (symbol as  $user$ ) is the role which uses the Grid system. User is managed only by its owned node, not by the entire Grid. And it may be the same user which belongs to owned node before the owned node joins the Grid system.*



**Fig. 2.** Structure of VDHA. Note: There are 13 nodes in the grid system. These nodes are grouped as 4 VOs. The number of nodes in each VO is 4,3,3,3 respectively. From each VO we choose one node as gateway node to form two up-layer groups with each having 2 nodes. Then from these two groups, one node each was chosen to form a root group

**Definition 5.** *Client host (symbol as cli) is an apparatus (such as desktop computer, palm, PDA, mobile computer, etc), which are used by users to login into the Grid system and to do the business.*

**Definition 6.** *Gateway node (symbol as gn) is a Grid node which takes coordinate functions in several different layer virtual groups.*

**Definition 7.** *Virtual group (symbol as VG) is formed virtually by the Grid nodes.  $VG_{\alpha}^i$  means the group is in the  $i$ th layer and the name of this virtual group is  $\alpha$ . The virtual group is identified by its group name and layer number.*

**Definition 8.** *Coordinator of virtual group (symbol as cvg) is a gateway node taking coordinate functions in the virtual group. The symbol  $cvg_{\alpha}^i$  ( $cvg_{\alpha}^i \in VG_{\alpha}^i$ ) means that it is a gateway node in the  $i$ th layer  $\alpha$  – named virtual group which functions as coordinator.*

**Definition 9.** *Virtual group tree (symbol as VGT) is hierarchical tree formed by virtual groups. In VGT there is a root virtual group (symbol as RVG), many leaf VGs called as virtual organizer (symbol as VO).  $VO_{\alpha_m}^m$  means that the virtual organizer is in the  $m$ -th layer and its name is  $\alpha_m$ . The order of layers is counted from RVG, which is defined as the first-layer VG. VG except VO is formed purely by gateway nodes. VO is formed by Grid nodes with one (and just one) gateway node. RVG can not be a VO, and VO can be within all the layers except the first layer.  $N_{\alpha}^i$  is the numbers of the nodes in  $VG_{\alpha}^i$ .  $N_g^i$  is the number of virtual groups in the  $i$ th-layer of VGT.*

**Definition 10.** *VDHA is a virtual group tree with depth of at least two layers. VDHA has dynamic properties in the number of Grid nodes, layers and virtual groups, virtual group compositions, and so on. In VDHA, we have following properties:*

1.  $VG_{\alpha}^i = \{gn \in VG_{\beta}^{i+1} | \beta \in A^i\}, i > 0, VG_{\alpha}^i$  is not a VO, here,  $A^i$  is the subset of the names of the  $i$ -th layer virtual groups. (This sentence means that the VG is formed from lower-layer groups.)
2. If  $gn_1 \in VG_{\alpha}^i \cap gn_1 \in VG_{\beta}^{i+1}$  and  $gn_2 \in VG_{\alpha}^i \cap gn_2 \in VG_{\beta}^{i+1}$ , then  $gn_1 = gn_2$ .
3. Each VG has one and only one node (cvg) which takes coordinate functions.
4. Grid node  $p$  can join more than one VO
5.  $PS = VO_1 \cup VO_2 \cup \dots \cup VO_{n1}$ , Here,  $n1$  is the number of virtual organizer.
6. If  $p$  satisfies the following condition:  $p \in VO_{\alpha_m}^m \cap p \in VO_{\alpha_{m-1}}^{m-1} \dots p \in VO_{\alpha_{m-k}}^{m-k}$ ,  $m \geq 2, k \geq 1$ , the  $p$  is gateway node. It is expressed with symbol  $gn(m, k, \alpha_{m-k} \dots \alpha_{m-1} \alpha_m)$ . The meanings of parameter values are:  $m$  is the layer order of VO in VGT ( $gn \in VO$ );  $k$  is the number of layers in which the gateway node functions;  $\alpha_{m-k} \dots \alpha_{m-1} \alpha_m$  are the names of the virtual groups from  $VO_{\alpha_m}^m$  to  $VO_{\alpha_{m-k}}^{m-k}$ . Symbol  $gn_{\alpha_i}^i \in VG_{\alpha_i}^i$  means that the gateway node is in the  $i$ th layer group with name  $\alpha_i$ .

### 2.3 Grid Group Management Protocol (GGMP)

GGMP is a protocol used to manage membership of virtual group and virtual group tree. Before giving out the protocol, we first define several functions and primitives as Table 1 shows and  $gn.Reselect\_GatewayNode\_Coordinator()$  – a function for selecting a new gateway node.

```

gn.Reselect_GatewayNode_Coordinator () {
  if (gn  $\neq$   $\phi$ ) {
    if (Pnode(gn) == cvg  $\in$  VGroup (gn))
      choose_new_cvg(cvg, VGroup(gn), (cvg  $\in$  VGroup(gn)  $\cap$ 
Pnode(cvg)  $\neq$ 
Pnode(gn)  $\cap$  cvg_w =Maxium of  $p_{iw}, p_i \in$  VGroup(gn) and
online));
    Remove(gn, VGroup (cvg));
    If (Layer(cvg)  $\neq$  Layer(TOP_gn(gn))) add(cvg, VGroup(UP_gn(gn)));
    Down_gn(gn).Reselect_GatewayNode_Coordinator (); } }

```

Now, we give out the algorithm of GGMP. (Due to the length of paper, the synchronization is not described, and the algorithm is outline with a lot of details omitted and may not be an optimum one)

/\*  $p_w$  means that Grid node has weight value  $w$ . Weight values of nodes are assigned as several classes according to nodes' resources etc. Suppose  $gn$  at the layer  $gn_{\alpha}^i$ , For all  $p_i \in VGroup(gn_{\alpha}^i), p_i$  contains three tables. (If  $p_i$  is in the root group or leaf group (VO), then two tables). 1. State table which includes

**Table 1.** Primitives and functions

Description	Meanings
<b>sender.send (message, receiver)</b>	sender sends message to receiver
<b>sender.send (message,receiver <math>\in</math> Set)</b>	sender sends message to all the receiver belong to Set
<b>remove(p,VO)</b>	p is removed from VO
<b>choose_new_cvg (p, VG, condition)</b>	p is chosen from VG according to the condition
<b>add (p, VG)</b>	p is added into VG
<b>Pnode(cvg)</b>	returns node ID of cvg
<b>group_name(VGroup(p))</b>	returns this group's name
<b>Layer (<math>var_{\alpha_m}^m</math>)</b>	returns the layer order of cvg or gn
<b>UPcvg(<math>cvg_{\alpha_m}^m</math>)</b>	returns cvg in the up-layer
<b>LOWcvg(<math>cvg_{\alpha_m}^m</math>)</b>	returns cvg in the low-layer
<b>VGroup (<math>cvg_{\alpha_m}^m</math>)</b>	returns virtual group in which there is $cvg_{\alpha_m}^m$
<b>Type(VG)</b>	returns VG's type (VG or VO)
<b>BOTTOM_gn(gn)</b>	returns gn down to the bottom layer
<b>TOP_gn(gn)</b>	returns gn up to the top layer
<b>Down_gn(<math>gn_{\alpha_i}^i</math>)</b>	downs gn into the low layer
<b>UP_gn(<math>gn_{\alpha_i}^i</math>)</b>	ups gn into the up layer
<b>gn.Down_Update ()</b>	update status of all nodes in all the lower layer groups the gn involved
<b>gn.Up_Update ()</b>	update status of nodes in all the upper layer groups the gn involved

group member list,group\_name, and cvg, etc . 2. Up state table which includes upper layer group member list,upper layer group\_name, and upper layer cvg, etc. 3. Down state table which includes lower layer group member list, lower layer group\_name, and lower layer cvg, etc . These tables are needed to be synchronized and consistent, that is, every node keeps a copy of the status\*/

```

while(true) {
  switch(event) {
    case: a  $VO_{\alpha}$  joins VDHA Grid system
      choose_new_cvg (gn,  $VO_{\alpha}$ , ( $gn \in VO_{\alpha} \cap gn_w = \text{Maximum of } p_{iw}$ ,
         $p_i \in VO_{\alpha}$  and online)); /* If  $p_{iw}, p_{jw}$ , etc are with the same value,
        random node is chosen. */
      set  $cvg \in VO_{\alpha} = gn$ ;
      cvg uses QDP protocol (defined as Sect. 2.4) to find the
        interested parts of the structure of virtual group tree such as
         $VG_{\beta}^k$ ;
      cvg.send(JOIN_MESSAGE,  $cvg_{\beta}^k$ );
      if ( $cvg_{\beta}^k$  accepts the requisition) add (Pnode(cvg),  $VG_{\beta}^k$ );
      cvg.send(state_ table of VGroup(cvg)_message, UPcvg(cvg));
      UPcvg(cvg).send (state_ table of VGroup(cvg)_message,

```

```

    p ∈ VGroup(UPcvg(cvg));
    UPcvg(cvg).send (state_ table of VGroup(UPcvg(cvg))
    _message, cvg);
    cvg.send(state_ table of VGroup(UPcvg(cvg))_message,
    p ∈ VGroup(cvg));
case: a  $VO_\alpha$  leaves from VDHA Grid system
    gn = Pnode(TOP_gn(gn ∈  $VO_\alpha$ );
    gn. Reselect_GatewayNode_Coordinator();
    gn = Pnode(BOTTOM_gn(gn ∈  $VO_\alpha$ ), gn.Up_Update ());
    gn = Pnode(TOP_gn(gn ∈  $VO_\alpha$ ), gn.Down_Update ());
    Delete  $VO_\alpha$ ;
case: gn leaves VDHA Grid system
    VG = VGroup(BOTTOM_gn(gn));
    gn. Reselect_GatewayNode_Coordinator ();
    set new gn = cvg in VG;
    gn = Pnode(BOTTOM_gn(gn ∈ VG)), gn.Up_Update ());
    gn = Pnode(TOP_gn(gn ∈ VG), gn.Down_Update ());
case: cvg fails to receive messages from  $p \in VGroup(cvg)$ ,
     $p \in VGroup(UPcvg(cvg))$  and  $p \in VGroup(Lowcvg(cvg))$ 
    exceeding a given times
    set gn = Pnode(cvg);
    VG = VGroup(BOTTOM_gn(gn));
    gn. Reselect_GatewayNode_Coordinator ();
    set new gn =  $cvg \in VG$ , add (Pnode(gn),VG);
    /*change previous gn to an ordinary node.*/
    gn = Pnode(BOTTOM_gn(gn ∈ VG)), gn.Up_Update ());
    gn = Pnode(TOP_gn(gn ∈ VG), gn.Down_Update ());
case: a node p joins a VO
    p.send(REQUISITION_MESSAGE,  $p_{neighbor}$ );
    /* $p_{neighbor}$  is the neighboring node in the VO, which is known to p*/
     $p_{neighbor}$ .send(REQUISITION_MESSAGE,  $cvg \in VO$ );
    if (cvg accepts the requisition) add(p, VO);
    cvg.send(p_joins_message,  $p_i \in VGroup(cvg)$ );
    cvg.send(copy_all_table_message, p);
    cvg.send(p_joins_message, UPcvg(cvg));
    UPcvg(cvg).send ( $p\_joins\_message, p_i \in VGroup(UPcvg(cvg))$ );
case: a node p leaves from a VO
    p.send (REQUISITION_MESSAGE,  $cvg \in VO$ );
    if (cvg accepts the requisition) remove(p, VO);
    cvg.send(p_leaves_message,  $p_i \in VGroup(cvg)$ );
    cvg.send(p_leaves_message, UPcvg(cvg));
    UPcvg(cvg).send ( $p\_leaves\_message, p_i \in VGroup(UPcvg(cvg))$ ); } }

```

### 2.4 Query and Discovery Protocols

In VDHA, query and discovery protocols are used for querying and discovering some entities such as resources and services, virtual group name, node status, etc. Every node has resources and services which are described by WSDL [9] or ontology languages, etc. Matching the request message is done by the agent of node which has the services. There are two kinds of QDP: Full Search Query and Discovery Protocol (FSQDP), which searches all nodes to find nodes that match the request message, and Domain-Specific Query and Discovery Protocol (DSQDP), which searches nodes in only specific domains. FSQDP first finds root virtual group, and then the coordinator of root virtual group forwards the query message to its all members. All of these members execute parallel forwards of the message down to the members of their low-layer groups until leaf virtual groups. FSQDP has time complex  $O(\log N)$  ( $N$  is the numbers of nodes), space complex  $O(N_{vvg})$  ( $N_{vvg}$  is node numbers of each virtual group), and message-cost  $O(N)$ . FSQDP is effective, but may cause much traffic. Domain-Specific Query and Discovery Protocol (DSQDP) (see Fig. 3) is quite similar to FSQDP but it only searches the nodes whose catalogue matches the requested group keywords. To use this protocol, the object of virtual group must maintain the catalogue with classifying services from general to detail. It may be done by the nodes' joining the proper virtual group of Grid system. The protocol DSQDP has time complex  $O(N_{vvg})$ , space complex  $O(N_{vvg})$ , and message-cost  $O(N_{vvg})$ . This protocol is effective and message cost is low. The details can be found in the paper [10].

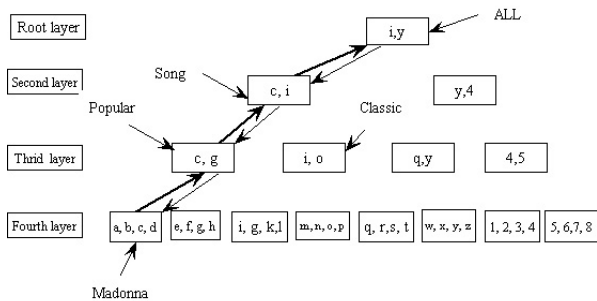


Fig. 3. DSQDP searching process

### 2.5 Authentication Protocol (AP)

Authentication Protocol in VDHA are based on public key infrastructure. In VDHA, the owned node takes as CA of the users and generates the user's public key and private key. The owned node keeps its owned users' public key, and also some information of the owned users such as password, etc., which are used to identify user in ordinary ways. So this authentication policy is compatible with



the common authentication policy used before joining Grid system. Because the numbers of nodes are smaller than the numbers of users, and for security and easy implementation reason, all the nodes' public keys are authenticated by CA centers. This AP protocol is somewhat different with Globus GSI [11]. We use authentication ticket to solve the problems such as single-sign-on, etc. Meanwhile, because the client host's IP address is generally LAN IP address, not the Internet IP address, we use the entrance nodes as proxy stations to help the client to connect to the Grid system. There are four modes about user's login (remote/local ow via client/node cli). Details of Authentication Protocol can be found in paper [8].

## 2.6 Message-Based Implementation of VDHA

The implementation of VDHA is based on the broker of message/event, as Fig. 4 shows. One of the working scenarios is as following: The application layer requests a service A in the Grid by sending query message which indicates the service name, searching method, and so on. Then QDP locates the nodes which have the service A. Then Service Lifetime Management Service (SLMS) creates service A instance, and this service instance provides the services to the application layer. After finishing providing the service, SLMS will destroy service A instance. Monitor and Control Service (MCS) is an optional service which is used to monitor and control the status of the node and service instances.

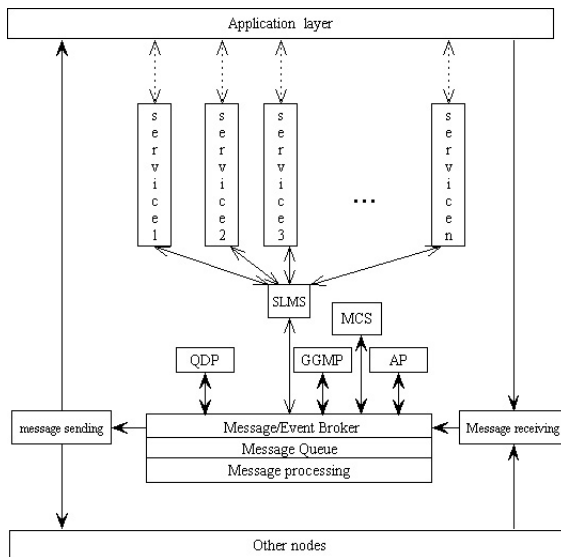


Fig. 4. The message-based implementation of VDHA

## 2.7 Grid Service Description Framework (GSDF)

As the above section, the service is the key in the VDHA Grid. The services can be dynamically appended into or deleted from the nodes. As infrastructures are different, MDS in Globus [12], and WSDL in Web Service are not enough to satisfy the needs. In VDHA, the services are used with three kinds of ways. (1) the service is simple, and the client end application can directly use it; (2) the client end application software uses the service's client-end API; (3) the client end program must be modified by programmer. So, the service description language must include service definitions which are understood by computer. This can be solved by the Ontology method. The language must include entities that are needed for implementing the service by SLMS. It also includes authorizations, accounting, protocol binding, and message format, etc. Therefore, GSDF must answer the following questions:

- (1) How does QDP use GSDF to find the services?
- (2) What protocol does the service bind and how does the protocol marshals?
- (3) How does SLMS use GSDF to implement the services?
- (4) How does the service account?
- (5) How does the service grant access right (authorization)?
- (6) What QoS does the service support?

And so on.

For satisfying those above requirements, the service description at least has the following items:

### Service Definition:

Natural language description: WHAT it is?

Ontology definition: WHAT it is?

Natural language description: WHAT it does?

Ontology definition: WHAT it does?

Natural language description: HOW does it do?

Ontology definition: HOW does it do?

### Protocol

Natural language description: protocol document

Ontology definition: protocol

Port

Transport protocol

Port

Security Message Format

Message encoding

Method

Method name

Parameter, . . . Parameter

Authorization

Account

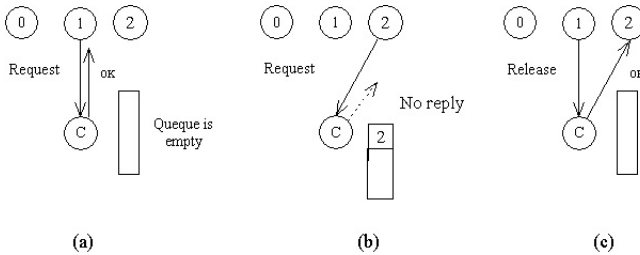
Qos

etc

We are engaging now on the draft of specification of GSDF and ontology-based Grid service description language (GSDL).

## 2.8 Synchronizing Schema

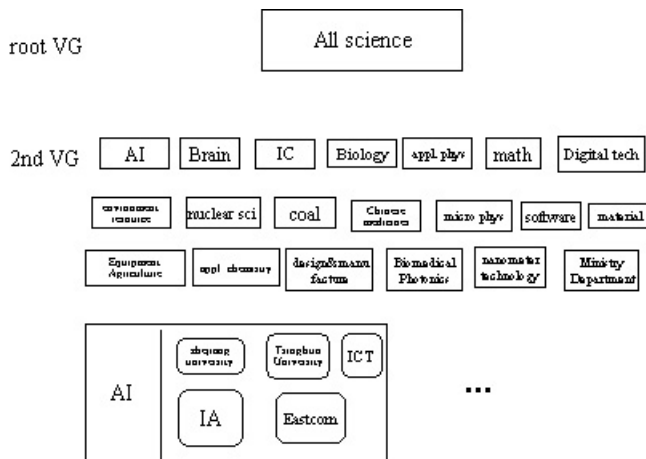
Because of the difficulty of implementing a distributed locking mechanism [13], synchronizing is still a problem hard to solve. But, VDHA has an easy schema to implement synchronization, because it has coordinator node in every virtual group. The coordinator of VG which is in the most top layer among the VGs with which a task involves will be taken as a center node to achieve mutual exclusion. The schema is similar to paper [14]. Whenever a node wants to enter a critical region it wants to enter and ask for permission. If no other node is currently in that critical region, the coordinator sends back a reply granting permission, as shown in Fig. 5. The schema may be explained more if coordinator is crashed down. When coordinator is crashed down, the GGMP will choose a new coordinator. Then coordinator by GGMP protocol will send message to notify all the nodes, and all the nodes which have requested or occupied the resource cancel their actions, and ask for permission of new coordinator again.



**Fig. 5.** synchronizing schema of VDHA. (a) Node 1 asks the coordinator for permission to enter a critical region. Permission is granted. (b) Node 2 then asks permission to enter the same critical region. The coordinate does not reply. (c) When node 1 exits the critical region. It tells the coordinator, which then replies to node 2

## 3 A Case Study

The virtual research projects granted by Chinese Educational Ministry are aimed to enhance the science and technology research by virtual cooperation via Internet. There are now 19 virtual organizers, each has a special domain. Each virtual organizer has average 6 nodes which are located in Universities or research institutes. In order to combine these 19 organizers into an e-Science Grid system, we use VDHA to model this e-Science Grid system prototype (called as Chinese University e-Science Grid CUEG). Nineteen nodes chosen from every 19 VOs each plus one Chinese Educational Ministry node form an up-layer



**Fig. 6.** VDHA architecture of CUEG (only nodes of AI are showed in the figure)

virtual group. Initially, the nodes located in the primary institutes of 19 VOs are chosen as gateway nodes (see Fig. 6). We have developed the prototypes of heterogenous information sharing service, literature resource service portal and knowledge service provider [8,15]

## 4 Conclusion

VDHA can solve the scale and autonomy problems. Some nodes can form a VO, and this VO can join the e-Science Grid without centralized administrator. In VDHA the messages are generally only concerned with the nodes of the three neighboring layer virtual groups, not with entire grid network. So, the e-Science Grid with VDHA has the possibility to become a huge net.

VDHA has high performance and exact discovery of resources and services. From the virtual group tree, we can know the detail information of every virtual group, so we can exactly and fully search the resources and services.

VDHA may easily manage privileges and roles of users. The users can be grouped, and the groups may be a member of a super group, and so on. The group can inherit privileges from super group. So, a user, who is the member of the group which inherits from a super group, can access the resource, if the privilege of accessing this resource is assigned to the super group by the authorization policy. This strategy has advantage for simplifying resource authorization policy if the Grid net is huge.

We have proposed the security architecture and authentication for VDHA. The security architecture fulfills the requirements of Grid such as single sign on, protection of credentials, compatible with local security solutions, and scalability.

It is easy to implement synchronization in VDHA structure Grid, because every virtual group of VDHA has a coordinator, which functions as a referee of locking/unlocking.

Our further work will focus on completing and enriching services of CUEG prototype, and on increasing nodes of CUEG.

**Acknowledgments.** This paper is supported by Virtual cooperative research project granted by the Ministry of Education of PRC. The participation of conference is supported by NOKIA Co., Ltd. Thanks specially to our colleagues and graduate students in our Lab for their discussions, cooperation and contribution.

## References

1. John Taylor, e-Science definition, <http://www.e-science.clrc.ac.uk>
2. I. Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International Journal of High Performance Computing Applications*, 15(3), 200–222, 2001, <http://www.globus.org/research/papers/anatomy.pdf>
3. I. Foster, Kesselman, J.M. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", 2002.2.17 <http://www.globus.org/research/papers/ogsa.pdf>
4. Grid Web Services Workshop.2001, <http://gridport.npaci.edu/workshop/webserv01/agenda.html>
5. David De Roure, Nicholas Jennings and Nigel Shadbolt, "Research Agenda for the Semantic Grid: A Future e-Science Infrastructure", 2001, 9 <http://www.semanticgrid.org/v1.9/semgrid.pdf>
6. D. Clark, "Face-to-Face with Peer-to-Peer Networking", *Computer*, Vol. 34, No. 1, January 2001, pp. 18–21
7. Krishna Kant, Ravi Iyer and Vijay Tewari, "A Framework for Classifying Peer-to-Peer Technologies", *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02)*
8. Huang Lican, Wu Zhaohui and Pan Yunhe, "Virtual and Dynamic Hierarchical Architecture for Chinese University e-Science Grid", *In the proceedings of 2002 International workshop on Grid and Cooperative Computing (GCC2002)*, Publishing House of Electronics Industry, pp. 297–311
9. Christensen, E., Curbera, F., Meredith, G. and Weerawarana, "Web Services Description Language (WSDL) 1.1". W3C, Note 15, 2001, <http://www.w3.org/TR/wSDL>.
10. Huang Lican, Wu Zhaohui and Pan Yunhe, "Virtual and Dynamic Hierarchical Architecture: an Overlay Network Topology for Discovering Grid Services with High Performance", *submitted to Twelfth IEEE Symposium on High Performance Distributed Computing (HPDC12)*
11. Foster, I., Kesselman, C., Tsudik, G. and Tuecke, S., "A Security Architecture for Computational Grids", *In ACM Conference on Computers and Security*, 1998, 83–91
12. I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", *International Journal of Supercomputer Applications*, 11(2): 115–128, 1997

13. B. Allcock, J. Bester, J. Bresnahan, A.L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke. "Data Management and Transfer in High Performance Computational Grid Environments". *Parallel Computing Journal*, Vol. 28 (5), May 2002, pp. 749–771.
14. Andrew S. Tanenbaum, "Distributed Operating Systems", *Prentice-Hall International, Inc.*, 1995, pp. 134–135
15. Huang Lican, Wu Zhaohui and Pan Yunhe, Zhou Xuezhong, "Knowledge Services Provider Model based on Virtual and Dynamic Hierarchical Architecture", *In the proceedings of 2002 International workshop on Grid and Cooperative Computing (GCC2002)*, Publishing House of Electronics Industry, pp. 245–253