MDS-BASED GEO-SPATIAL METADATA FRAMEWORK*

WANG Yuelong, WANG Wenjun, LUO Yingwei, WANG Xiaolin and XU Zhuoqun

Dept. of Computer Science and Technology, Peking University, Beijing, 100871

ABSTRACT

MDS (Monitoring and Discovery Service) is a key infrastructure in grid computing, it mainly undertake registry, updating and discovery of all sorts of resource information in the grid environment. The information organized and released by the MDS is the descriptive information of diversified resources (net resources, computing resources, data resources etc), including data, services and entities of the grid environment, MDS can provide a veracious and real-time vies of dynamic resources in the grid. This paper developed a MDS-based framework for coping with the problem of distributed geo-spatial metadata, the framework is called GridMeta. First, we introduced the organization of GridMeta Servers; then we put forward a information model based on DIT, according to this information model, geo-spatial metadata, service metadata, user metadata and all kinds of network (grid) resource information can be depicted and stored identically; and the last, we provide a set of accessing interfaces of GridMeta API, by employment of these interfaces, the registering, adding, deleting, updating and (automatic) discovery of distributed metadata can be implemented.

KEY WORDS: Grid Computing, Metadata, MDS, LDAP, PAI, GridMeta

1. INTRODUCTION

Metadata technology plays a key role in the integration and application of geo-spatial information. Metadata is the data about data, is the descriptive information of information

* Corresponding author: LUO Yingwei, lyw@pku.edu.cn.

resources. With the development and popularization of the Internet/Intranet technology, the metadata technology is no longer a method of data description and data index, and now it is becoming one indispensable and powerful method and tool in the domain of information sharing and interoperation all over the whole Internet, this kind of tool can help cope with a series of bottleneck problems such as data discovery, data transformation, data management and data employment.

Generally speaking, a perfect GIS application system, may always maintain a metadata management component, with which the GIS system could descript and issue the information about the data, services, users and other resources. But until now there is not a universal specification about the management and release of metadata, and the organization and management of metadata is mostly a kind of freewill behavior, this brings about some obstacles to the metadata sharing between different fields. Namely, the traditional metadata managing systems are commonly domain-faced, when there is a need to integrate the geo-spatial information of different GIS systems, the problem of heterogeneity of metadata will emerge. The heterogeneity of metadata usually shows itself as following:

Geographical distribution: Generally, the metadata of different application systems is distributed geographically.

Systematic autonomy: A metadata system operated by an application system is autonomous, and has its own managing and accessing authority.

A**rbitrary choices of accomplishment**: Metadata can be described with different templates according to system features and application requests, for example: table structure, tree structure and relation structure, etc. And different metadata systems can be implemented with different tools, such as Access, Oracle, DB2 and the others.

Variety of Information resources: As a kind of descriptive data of data, metadata can describe various information resources.

Semantic heterogeneity: Each metadata framework describes information based on its own semantic template, for it is impossible to design a universal semantic template for the metadata of different domains. When implementing system integration, we inevitably encounter the problem of semantic heterogeneity, for instance: one concept with different meanings, the same meaning expressed with different words, etc.

It should be pointed out that metadata can be divided into broad-sensed metadata and narrow-sensed metadata. Generally speaking, all descriptive data about data is called metadata, for example, in the framework of PAI (PKU spatial Application Integrating infrastructure) that is developed by GIS laboratory Peking University, metadata is divided into three parts, namely data metadata, application metadata and software metadata.

Narrow-sensed metadata always referred to geo-spatial metadata. In a geographical information application system, geographical information is usually a kind of leading resource. From this point of view, a geographical information system is primarily a solving scheme for geo-spatial data (base), and the next is a domain-oriented application case. Geo-spatial information always has such features as geographical distribution, heterogeneity and autonomy, it is these features that result in the key role of metadata technology in geo-spatial information sharing and interoperation. In this thesis, if there is not special specification, the concept of metadata refers to geo-spatial information metadata.

In the PAI system, we designed and implemented a metadata-integrating framework named GridMeta, which is based on MDS (Monitoring and Discovery Service) technology of grid computing. GridMeta focuses on the distributed, heterogeneous, and dynamic metadata in the Internet (or Grid) environment, effectively solves the problem of automatic discovery and integrated application of geo-spatial information and relative services. Our manipulating ideas are: Do not change or update the traditional metadata framework, but to construct a uniform mapping view for the metadata of all GIS systems, and to provide necessary services such as registering, deleting, updating and querying, etc. Based on this framework, the navigation of distributed geo-spatial information can be easily realized. Figure 1 shows the role of GridMeta in the PAI infrastructure.

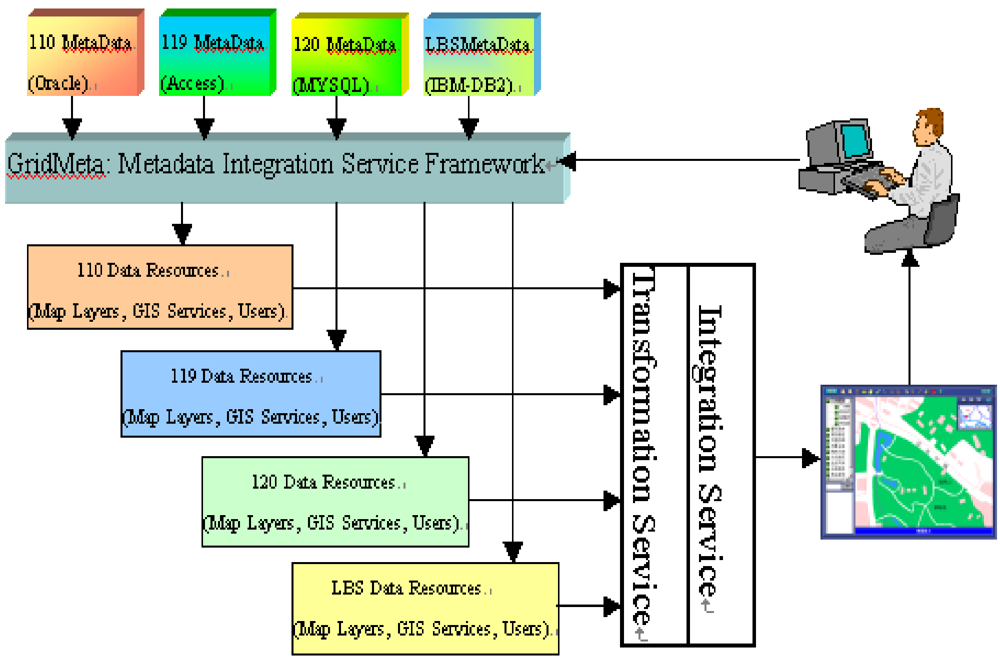

Figure 1. The Role of GridMeta in the PAI Infrastructure

GridMeta is primarily designed for PAI, but it has a universal value for the same kind of problems. Firstly, the paper summarily introduces the basic principle of MDS (Monitoring and Discovery Service), and addresses the framework and implementing method of GridMeta; secondly puts forward a set of universal descriptive specifications for metadata based on the MDS information model; at last gives a querying and searching strategy based on MDS API.

2. IMPLEMENT OF GridMeta

2.1 Introduction of MDS

We adopt the MDS (Monitoring and Discovery Service) technology to implement our Metadata Integration Service in the PAI System. MDS is an information management framework serving the Grid Computing, and it is brought forward by the Globus project. MDS concentrates on solving the problem of managing, issuing and utilizing all sorts of massive, distributed and dynamic resource information in the Grid Computing environment. The content of MDS mainly includes resource description, resource detection, resource monitoring, resource updating and so on.

Acting as the infrastructure for information service, MDS manages the dynamic as well as static information of all sorts of resources in the Grid Computing environment by means of dynamical and expansible framework. The advantages of MDS are as following:

- Common descriptive view above the different resources;
- Uniform and flexible accessing interfaces;
- Capacity to access multiple resource roots;
- Automatic self-configuration according to different and dynamic environment;
- Distributed and autonomous management.

In order to achieve data producing, data storage, data publishing, data searching and data visualization, MDS provides an configurable information provider component, which is called Grid Resource Information Service (GRIS), and a configurable index collection component, which is called Grid Index Information Service (GIIS). Based on the GRIS/GIIS architecture, a stratified and distributed information service structure can be constructed.

Each GRIS component takes the responsibility of registry, maintenance and searching of distributed objects within its domain. GIIS supports the ability to perform efficient simultaneous queries to several GRIS components. GIIS could provide whole-ranged query service by combining distributed GRIS services. And it also provides a uniform resource view to facilitate the application programs to perform searching and query operation.

MDS provides a set of methods for searching grid resources, these methods has rich functions and are convenient to use, for instance, MDS can perform the query of certain grid resources (data, software, hardware and so on) of special properties or services; MDS can list the up-to-date status of a certain kind of resources or whole resources belonged to a virtual organization; MDS could make a sum of all the distributed grid resources of a virtual organization by GIIS's registry mechanism, and thus creates a general resource mapping of the grid.

It should be mentioned that, the information service framework based on GRIS/GIIS is of much scalability. MDS's resource information servers (GRIS and GIIS) can be flexibly configured in accordance with specific conditions. It may has only one level (here there only exists GRIS), and it also could owns two levels, what's more, coalesce of information servers based on multiple levels. To further point out, GRIS and GIIS are only relative concepts, especially in the multi-leveled MDS architecture, a middle-leveled information server acts as a GIIS to the low level server, on the other hand, it is regarded as GRIS by the high-leveled server.

2.2 Organization of GridMeta Servers

Based on MDS technology, we employ GRIS/GIIS architecture to config our GridMeta servers. First, a GRIS server is built for each GIS system (or a grid node). These GRIS servers are responsible for registry, maintenance and publishing of metadata, and metadata here describes domain-related spatial information, service information and user information. In common cases, a GIS application system (or a grid node) maintains a local GRIS component. Nevertheless, if necessary, several GIS systems can share a same GRIS server. In default cases, GRIS service may be self-configurable, and perform listening on web port 2135.

Above GRIS, GridMeta designs a configurable component for index collection, which is called GIIS. GIIS is responsible for maintaining and publishing each GRIS's registered information, and providing a cache service which is analogous to a Web Searching Engine. GRIS and GIIS are linked by the pointers of referral.

Metadata of GRIS can be registered to GIIS through HeartBeating mechanism, or when GIIS receives a user's request, and its cached information expires, it may obtain relevant refreshing information from GRIS. The organization of GridMeta metadata servers is illustrated as Figure 2.

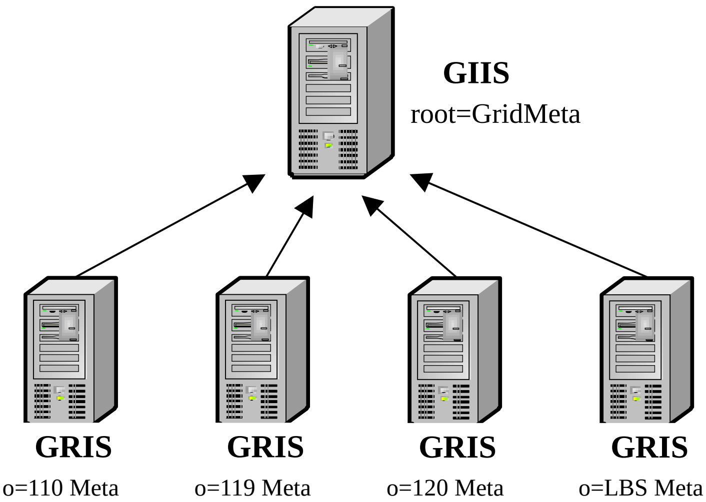

Figure 2. Organization of GridMeta servers

GRIS and GIIS are implemented based on LDAP server. To solve the problem of load balance, LDAP server's information index is maintained with redundance technology. Each grid node's LDAP server (GRIS or GIIS) stores local metadata. Meanwhile, the metadata of other grid nodes or of the whole virtual organization is also stored locally.

On each grid node, local metadata composes the main database, while other grids' metadata composes the stoke database (or slave database). The data consistency among different nodes is guaranteed by the stoke mechanism of LDAP server. When the data of the main database is being modified, the data in the slave database is updated automatically (timed update or in-time update, while the refreshing speed may not be very quick).

The schematic instance of users' access to GRIS and GIIS is illustrated as Figure 3. The metadata that a user requires may be satisfied through direct access to GRIS, or through GIIS. GridMeta provides the a set of uniform accessing interfaces which only demand users to present their access qualifications, without concerning the lower details (from GRIS or from GIIS).

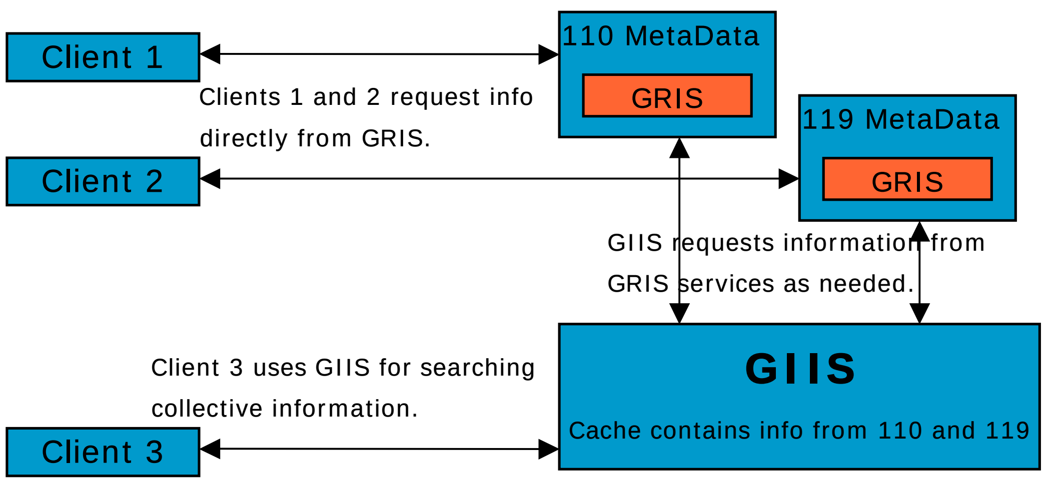

Figure 3. Users accessing GRIS and GIIS

3. INFORMATION MODEL OF GridMeta

Based on LDAP protocol, GridMeta introduced a stratified tree structure that is called DIT (Directory Information Tree), to organize metadata information. Starting from the root node, DIT maintained a hierarchical view on all the resource information of the whole tree, and provided a searching mechanism based on the tree structure. The DIT of GridMeta is illustrated as Figure 4.

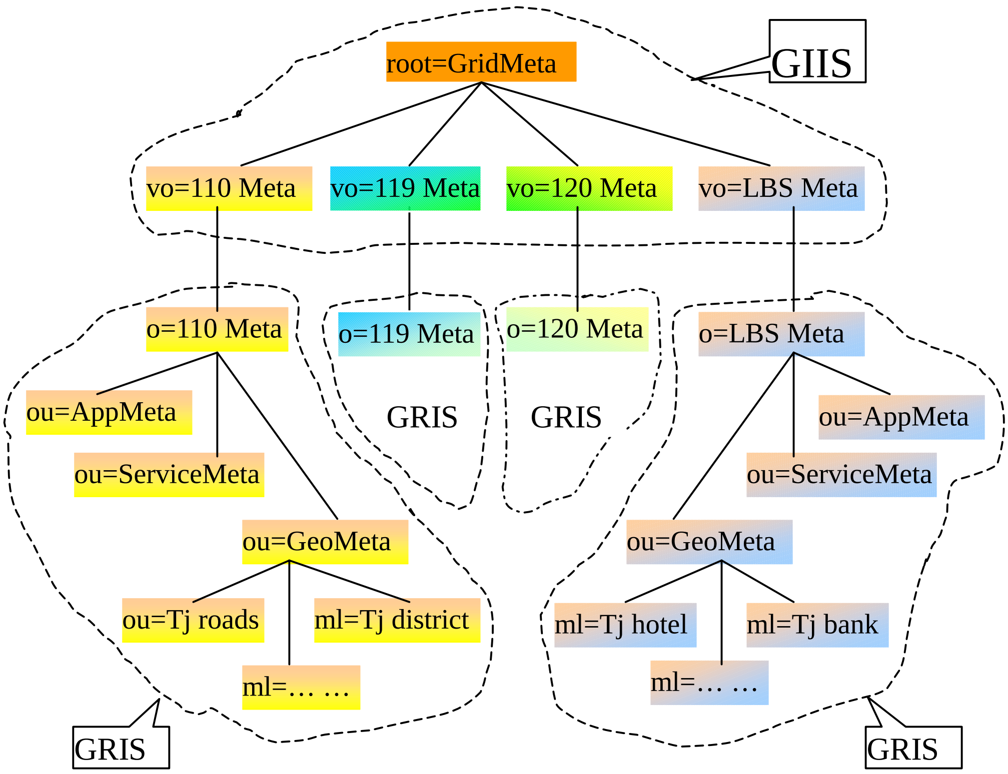

Figure 4. DIT of GridMeta

As shown in Figure 4, information of a DIT in GridMeta is stored in more than one subordinate tree, and all these sub trees compose a distributed directory. These sub trees are implemented with several LDAP servers (A subordinate LDAP server forms a GRIS component, while an upper LDAP server forms a GIIS), or maintained by one LDAP server if necessary. LDAP servers are linked by referral mechanism. If a subordinate tree does not contain the metadata that a user needs, it can redirect to other locations of the DIT by the referral mechanism, and this is transparent to the user.

DIT consists of many objects that are called entries. Each object is either a container or a leaf. A container can comprise other objects, while a leaf node does not contain any other objects. An entry contains one or more records depicting the real or abstract objects in grid computing environment such as hosts, users, services, maps and maplayers. Records are stored as <attribute/value> pairs, and are represented as “name=value”.

Each DIT node can be identified by one or more attributes, that is, the Relative Distinguished Name, in short RDN. RDN can only be used to distinguish an entry within one container, in order to identify a specific entry in an entire tree scope, GridMeta specifies the DN (Distinguished Name). DN is formed by concatenating the serial RDN strings from the current entry to the root entry.

For instance, there is an entry of maplayer, its RDN is: lyrid=00104, and its DN is lyrid=00104, ou=GeoMeta, o=LBS Meta, root=GridMeta.

A DN can identify an entry in a whole tree scope. With DN, users can access any an entry in an information tree at will.

GridMeta information model has a scalable data structure and an extensible architecture, this can be reflected in following aspects: (1)An attribute of an object can be defined as MUST or MAY; (2)There can exist multiple-valued attributes; (3)Users can introduce new object and attribute type based on their own schema. All above properties have guaranteed that a user can add his own metadata in the GridMeta framework conveniently.

In GridMeta, metadata of geo-spatial information is organized on the unit of maplayer. A maplayer’s metadata forms a leaf node of DIT, namely an entry. Figure 5 shows an entry that describes a maplayer. The maplayer’s RDN is lyrid=00104.

Name	Value	Type	Size
lyrid	00104	text attribute	5
objectClass	MapLayer	text attribute	8
mapid	003	text attribute	3
mapname	lbs	text attribute	3
lyrname	bj-bank	text attribute	7
lyrkind	lounge	text attribute	6
place	beijing	text attribute	7
lyrscale	1:2000	text attribute	6
lyrdate	2003/05/05	text attribute	10
lyrsrs	longitude/latitude	text attribute	18
rng-left	117.119082	text attribute	10
rng-right	114.967047	text attribute	10
rng-top	40.470121	text attribute	9
rng-bottom	36.694378	text attribute	9
rng-zmax	0	text attribute	1
rng-zmin	0	text attribute	1
dt-fmrt	mapinfo	text attribute	7
dt-path	vectdbj	text attribute	7
dt-owner	jwzh	text attribute	4
acs-userid	vector	text attribute	6
acs-port	1521	text attribute	4
acs-passwd	vector	text attribute	6
acs-mode	002	text attribute	3
src-ip	162.105.17.111	text attribute	14
lbl-zmax	0	text attribute	1
lbl-zmin	0	text attribute	1
validfrom	20030707091038Z	text attribute	15
validto	20030708091038Z	text attribute	15
keepto	20030709091038Z	text attribute	15
subschemaSubentry	cn=Subschema	operational attribute	12

Figure 5. An entry depicting the metadata of a maplayer

Besides geo-spatial metadata, GridMeta can also integrate other kinds of metadata such as service metadata, user metadata, and software/hardware/net information of the network (or Grid). To make the upper-leveled applications more flexible and convenient, GridMeta introduced "relation" mechanism between all kinds of metadata. For example, adding a "lyrid" attribute to the service metadata or user metadata, then you could position a maplayer's metadata based on the service information or user information automatically. An instance, there is a service in LBS system, the service's RDN is lbsid=12, one of its attributes is svcname=getAroundServ. The function of this service is to find nearby banks on a city map, so it needs to integrate a maplayer (e.g. RDN is lyrid=00104). Therefore, when designing corresponding service metadata, we can add one attribute lyrid=00104 to the service entry (RDN is lbsid=12), the entry is shown in figure 6. So, when a user calls the service (RDN is lbsid=12), the system will automatically get the attribute: lyrid=00104, then query the maplayer entry whose RDN is lyrid=00104, and get the maplayer's metadata. All preceding operations are transparent to the users.

Name	Value	Type	Size
lbsid	12	text attribute	2
objectClass	LBS-Application	text attribute	15
svcname	getAroundServ	text attribute	13
lyrid	00104	text attribute	5
servtype	bank	text attribute	4
comment	03	text attribute	2
validfrom	20030707091038Z	text attribute	15
validto	20030708091038Z	text attribute	15
keepto	20030709091038Z	text attribute	15
subschemaSubentry	cn=Subschema	operational attribute	12

Figure 6. An entry depicting the metadata of a service

Sometimes, a service (or user) needs to integrate more than one maplayers. To achieve this, one can add more than one attributes such as lyrid=* to the corresponding service entry or user entry, because GridMeta permits object of multi-valued attributes.

4. ACCESS TO GridMeta

Because GridMeta servers run on the TCP/IP protocol, all requests on the Internet can access the metadata of GridMeta through the TCP/IP protocol, regardless of the operation system (Linux, Windows or other platforms).

Based on the MDS API, GridMeta developed a set of access interfaces that are not related with the substrate layer, so when accessing GridMeta, the client programs should not know the substrate details. When the substrate structures have been changed, the only thing to do is to adjust the programming interfaces, and the upper application would not be influenced. GridMeta provided four kinds of principal interfaces: query, add, modify and delete. Application programs can use these APIs to implement all kinds accessing operations, and users also can extend the metadata management framework as need with the interfaces.

Because GridMeta is constructed on the base of MDS and LDAP protocol, users also can use MDS API or JNDI API to access GridMeta servers directly.

GridMeta provides a mechanism of positioning geo-spatial metadata according to service metadata, user metadata or other kind of metadata, this mechanism realizes the function of metadata's automatic discovery primarily, and offers great convenience to the design of upper applications. Using the accessing interfaces and methods provided by GridMeta, users and upper-leveled applications could access GridMeta servers conveniently. Listing 1 is a section of Java codes that utilized GridMeta API to query a

GridMeta server.

Listing 1: Searching the GridMeta

```
    GridMeta    gridmeta;
    gridmeta = new GridMeta("gis.pku.edu.cn", "2135",
                                                "ou=GeoMeta, o=LBS Meta, root=GridMeta");
    try
    {
        gridmeta.connect();
        String       filter    = "(&(svcname=getArounServ)(servtype=bank))";
        String       scope    = " gridmeta.SUBTREE_SCOPE";
        String       layerMeta;
        Hashtable    maplayer;
        maplayer = gridmeta.searchLayer(filter, scope);
        layerMeta = toXML(maplayer);
        gridmeta.disconnect();
    }
    catch(MDSException e)
    {
        System.err.println( "Error:"+ e.getLdapMessage() );
  }
```

First step, to construct an object instance (named gridmeta) of the class GridMeta with the following piece of code (namely, to employ GridMeta's constructing method).

```
gridmeta = new GridMeta("gis.pku.edu.cn", "2135",
```

```
"ou=GeoMeta, o=LBS Meta, root=GridMeta");
```

The parameters in class GridMeta are GridMeta server's name, port (default 2135), DN (Distinguished Name) or Base Directory. Default DN is "root=GridMeta", the root node.

The next step, after running the constructing function, is to use the function of **connect()** to construct a connection. After having done this, a searching operation can be undertaken with the sub row of code:

```
maplayer = gridmeta.layerSearch(filter, attribs, gridmeta.SUBTREE_SCOPE);
```

In the preceding row of code, the parameter of "filter" defines what a searching filter should be used. When the value of "filter" is "objectclass=* ", this denotes to search all the objects of a DIT, and when the value of "filter" is "lyrname=bj-mainroad", that means to search all the maplayers which name is "bj-mainbroad" in a DIT. One of the principal advantages of GridMeta is that it supports locating the geo-spatial metadata based on service metadata, user metadata and others. For an instance, in Listing 1, when the parameter of "filter" is given a value as following:

```
filter="(&(svcname=getAroundServ)(servtype=bank))"
```

Above value of "filter" means to search a maplayer (or several maplayers), this(these) maplayer(s) could support such a service: "**to query the nearby banks**".

Another parameter of "scope" defines the bound of searching operation, ONELEVEL_SCOPE denotes to search only one level that contains current entry (or node), while SUBTREE_SCOPE means to search the subordinate tree of the current node.

As for the method of searchLayer() owned by the class of GridMeta, its searching result is a hash table, the template of the hash table is org.blobus.common.MVHashtable, this data model is encapsulated in the package of MDSResult. GridMeta can transform the searching result from hash table to XML-based text with the method **toXML()**, this step of operation can make great convenience for the upper applications or methods to comprehend and process the searching result.

Listing 2 shows a XML-based searching result which contains a maplayer's metadata, this searching result is generated by the above-mentioned section of codes.

Listing 2: XML-based searching result: a maplayer's metadata

```
<GridMeta svcname=Locationv, servtype=bank>
```

```
&lt;Layers&gt;
  &lt;Layer dn="lyrid=00104, ou=GeoMeta, o=LBS Meta, root=GridMeta"&gt;
    &lt;lyrid&gt;00104&lt;/lyrid&gt;
    &lt;objectClass&gt;MapLayer&lt;/objectClass&gt;
    &lt;mapid&gt;003&lt;/mapid&gt;
    &lt;mapname&gt;lbs&lt;/mapname&gt;
    &lt;lyrname&gt;bj-bank&lt;/lyrname&gt;
    &lt;lyrkind&gt;lounge&lt;/lyrkind&gt;
    &lt;place&gt;beijing&lt;/place&gt;
    &lt;lyrscale&gt;1:2000&lt;/lyrscale&gt;
    &lt;lyrdate&gt;2003/05/05&lt;/lyrdate&gt;
    &lt;lyrsrs&gt;longitude/latitude&lt;/lyrsrs&gt;
    &lt;rng-left&gt;117.119082&lt;/rng-left&gt;
    &lt;rng-right&gt;114.967047&lt;/rng-right&gt;
    &lt;rng-top&gt;40.470121&lt;/rng-top&gt;
    &lt;rng-bottom&gt;36.694378&lt;/rng-bottom&gt;
    &lt;rng-zmax&gt;0&lt;/rng-zmax&gt;
    &lt;rng-zmin&gt;0&lt;/rng-zmin&gt;
    &lt;dt-fmrt&gt;mapinfo&lt;/dt-fmrt&gt;
    &lt;dt-path&gt;vectdbj&lt;/dt-path&gt;
    &lt;dt-owner&gt;jwzh&lt;/dt-owner&gt;
    &lt;acs-userid&gt;vector&lt;/acs-userid&gt;
    &lt;acs-port&gt;1521&lt;/acs-port&gt;
    &lt;acs-passwd&gt;vector&lt;/acs-passwd&gt;
```

```
      <acs-mode>002</acs-mode>
      <src-ip>162.105.17.111</src-ip>
      <lbl-zmax>0</lbl-zmax>
      <lbl-zmin>0</lbl-zmin>
      <validfrom>20030707091038Z</validfrom>
      <validto>20030708091038Z</validto>
      <keepto>20030709091038Z</keepto>
    </Layer>
  </Layers>
</GridMeta>
```


5. CONCLUSION AND FUTURE WORK

This paper introduces an MDS-based management framework named GridMeta for distributed geo-spatial metadata, including MDS-based architecture, information model, information access and programming interfaces etc. This framework can undertake discovery, registry, query and updating of all kinds of metadata in network or grid environment, it also can describe and store the information of diversified objects (permanent and temporary) in network or grid, so it is suitable for solving the problem of distributed and dynamic geo-spatial metadata.

The principal advantages of GridMeta is as following:

- Suitable for solving the problem of distributed, autonomous and heterogeneous metadata of WebGIS;
- Based on the MDS technology, regarding the metadata as a kind of grid resource that can be monitored momently by the HeartBeating mechanism, this measure solved the problem of unstable metadata in distributed environment.
- Locating geo-spatial metadata on the base of server and use information, this archived automatic discovery of geo-spatial metadata primarily;
- Distributed LDAP servers based on GRIS/GIIS, solved the problem of

load balance.

- A GridMeta server is constructed on the base of OpenLDAP, so it has a faster respond speed than relation database; especially when more than one user access a GridMeta server simultaneously, the superiority is more outstanding.

- The entry-based security and authentication mechanism, and the encrypt transmission of LDAP can guarantee the information security in the open network and grid environment.

The future work is: First, to set up a cache mechanism for the client program, thus a searching request will not be sent to a LDAP server every time, and this can increase the respond speed further more; Second, to explore the security and authentication mechanism on the base of users' authorities; And the third, to introduce "WSDL+UDDI" technology to our framework, this can facilitate the automatic broadcasting and detecting of metadata greatly, so to implement a GridGIS infrastructure based on Grid Services at last.

ACKNOWLEDGEMENTS

Supported by the National Natural Science Foundation of China under Grant Nos. 60030005, 60073016, 60203002; the National High Technology Development 863 Program of China under Grant Nos. 2001AA113151, 2002AA135330, 2002AA134030; the National Grand Fundamental Research 973 Program of China under Grant No. 2002CB312000; the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20020001015; the Natural Science Foundation of Beijing of China under Grant No. 4012007.

REFERENCES

[1] Steven Fitzgerald, Ian Foster, Carl Kesselman, Gregor von Laszewski, Warren Smith, Steven Tuecke (1997). "A Directory Service for Configuring High-Performance Distributed Computations", *6th International Symposium on High Performance Distributed Computing (HPDC '97)*, August 05-08, 1997, Portland, OR.

[2] Gregor von Laszewski and Ian Foster (1998). "Usage of LDAP in Globus", http://www.globus.org/mds/globus_in_ldap.html, April 1998.

[3] Han Hua, Dai Yafei, Li Xiaoming, Guo Chaoyang (2002). "A Distributed Object Naming Service Based on Distributed LDAP", *High Technology Letters*. 2002, Vol.12, No.10, P.33-38.

[4] Zha Li, Xu Zhiwei, Lin Guozhang, Liu Yushu, Liu Donghua, Li Wei (2002). "A LDAP Based Monitoring System for Grid", *Journal of Computer Research and Development*, Vol.39, No.8, Aug. 2002.

[5] Wang Zhiwen, Xia Qin, Li Zengzhi, Li Pingjun (2001). "Study of Lightweight Directory Access Protocol in Network Management", *Journal of Computer*, Vol. 35, No. 12, Dec. 2001.

[6] Xiao Nong, Ren Hao, Xu Zhiwei, Tang Zhimin, Xie Xianghui, Li Wei (2002). "Design and Implementation of Resource Directory in National High Performance Computing Environment", *Journal of Computer Research and Development*, Vol. 39, No. 8, Aug. 2002.

[7] Wang Yijie, Xiao Nong, Ren Hao, Lu Xicheng (2002). "Research on Key Technology in Data Grid", *Journal of Computer Research and Development*, Vol. 39, No. 8, Aug. 2002.

[8] Huang Yang, Li Wei, Xu Zhiwei (2003). "Resource Directory Management in National High Performance Computing Environment", http://www.gridhome.com/grid/talksanddocs.htm.

[9] Li Wei, Xu Zhiwei, Bu Guanying, Zha Li (2003). "An Effective Resource Locating Algorithm in Grid Environment", http://www.gridhome.com/grid/talksanddocs.htm.

[10] Vladimir Silva (2003). "Querying the Grid with the Globus Toolkit Monitoring and Discovery Service", http://www-106.ibm.com/developerworks/grid/library/gr-mds.html.

[11] Vladimir Silva (2003). "Implementing a Grid information provider with the Monitoring and Discovery Service", http://www-106.ibm.com/developerworks/library/gr-ipmds.html.

[12] Wang Xiaolin, Luo Yingwei, Cong Shengri, Zhang Ning, Xu Zhuoqun, Lu Zhonghui (2001). "A Study of Spatial Metadata and Its Applications", *Journal of Computer Research and Development*, Vol. 38, No. 3, Mar. 2001.

[13] Li Qi, Chen Aijun, Xia Shudong (2000). "Research and Design of Geospatial Metadata Management Systems on WebGIS", *Chinese Journal of Image and Graphics*, Vol. 5(A), No. 10, Oct. 2000.

[14] Chen Aijun, Li Qi, Xu Guangyou (2002). "A Geospatial Information Layer Sharing Model for Digital Earth", *Journal of Software*, Vol.13, No.08, 2002.