# Automatic Discovery of Data Resources in the E-Government Grid

XIANHU MENG[1*] YAN WANG[2], WENYU ZHANG[3] and JINQI MENG[4]

[1]School of Information
Zhejiang University of Finance & Economics
18 Xueyuan Street, Hangzhou 310018, CHINA
mengxianhu@163.com

[2]School of Information
Zhejiang University of Finance & Economics
18 Xueyuan Street, Hangzhou 310018, CHINA
wangyan@zufe.edu.cn

[3]School of Information
Zhejiang University of Finance & Economics
18 Xueyuan Street, Hangzhou 310018, CHINA
wyzhang@pmail.ntu.edu.sg

[4]Library
Zhejiang University of Finance & Economics
Hangzhou 310018, CHINA
mengjinqi163@163.com

***Abstract:* -**
Nowadays, there is a growing number of e-government portals and solutions that provide integrated governmental data resources to the customers (citizens, enterprises or other public sectors). However, the administration of distributed data resources are faced with increasing challenges caused by the discovery difficulties across the internet. To overcome this, this paper puts forward the concept of a model for automatic data resource discovery in the data grid environment for e-government applications. The paper elaborates the rule of global naming, brings forward metadata's registration and storage on the naming rule, and explores how to use it to find out the long-distance data resources rapidly. The paper also sets forth the method for restraining, sending and accessing data resources repeatedly by searching "comparison table" and using the time marker under the multiple data grid nodes, and points out that "comparison table" should adopt the strategy of prefering frequently used data and clearing rarely used data.

*Key-Words:* - Data grid, E-government; Metadata; Global naming; Data discovery

## 1 Introduction

Due to current trends in the public administrative fields towards highly specialized portal and solution providers cooperatively offering integrated governmental data resources to the customers (citizens, enterprises or other public sectors), there is an increased need for ubiquitous virtual governmental agencies to establish and maintain a Computer Supported Cooperative Work (CSCW) through effective communication, interoperation, integration and collaboration at the data level. Although the current internet and World Wide Web (WWW) have given birth to the emerging concepts of e-government towards CSCW, the automatic discovery of data resources is still a major obstacle to sharing and coordinated use of multiple governmental data provisions, which may be geographically dispersed across the internet due to their largely unplanned and unanticipated growth.

The development and maturing of grid technology [1] provides the opportunity for various heterogeneous systems based on various platforms

[*] Corresponding author

to cooperate with each other, and realizes the smooth communication and data sharing among them. As of today, on the top of grid research, the work on data grid has also been developed all over the world. In the United State, based on the research and development of Globus — a tool of grid system, National Laboratory and Southern California University probes into the system frame and key technology [2, 3] of the data grid, and emphasizes on the data storage, management of metadata, management of data copy, and so on. The research on grid in some developing countries, like China is still in the elementary stage, such as the project called Vega Grid [4] developed by the Institute of Computing Technology, Chinese Academy of Sciences. One of the key problems of data grid is to access the heterogeneous data distributed in the heterogeneous systems. In the grid environment all of the databases we are interested in can be represented as one integrated database, namely a virtual database in which data could be acquired through simple access statement.

For example (figure 1), a cross-city governmental agency stores its commercial or technical data across Newyork, Washington and Boston. These databases are physically seperated and distributed, but are logically standalone and integrated. Wherever the data is migrated, the user who wants to acess the data doesn't need to know where the data is located, because decentralized databases can be integrated by broadcating, migrating, routing and transferring data among them freely. The data grid realizes the tranparent access to distributed data.

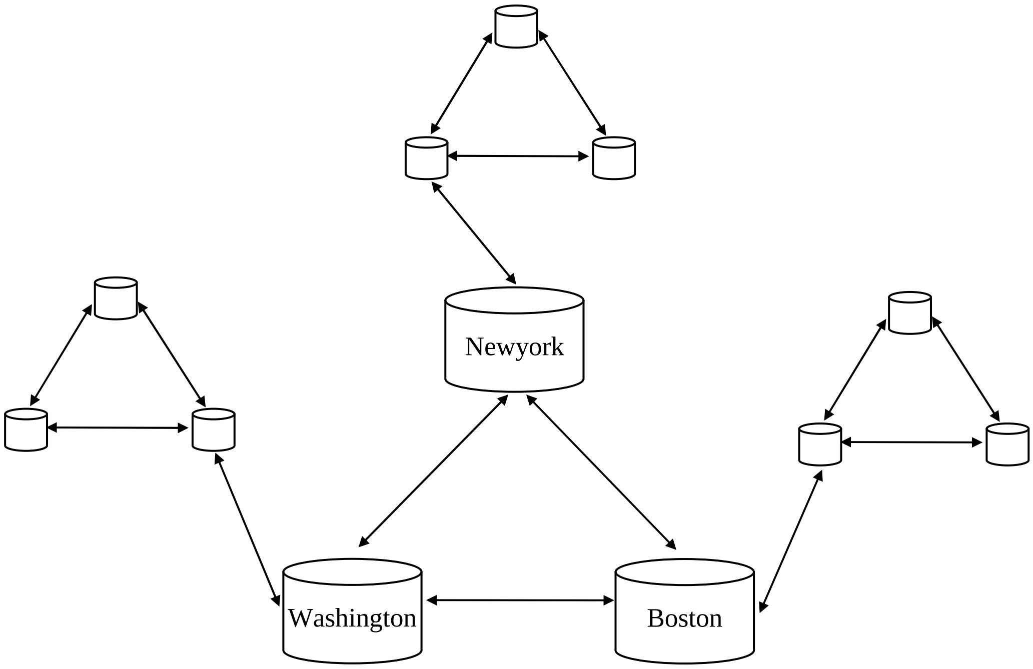

Figure 1 Distributed but integrated databases in the data grid

Regarding e-government applications on the grid, Silva & Senger [5] discussed the utilization of grid computing platforms as an enabling infrastructure for e-government, e-commerce and e-business application. A set of grid services that are fundamental for e-government applications, such as database access and integration and knowledge discovery services are identified. Fu et al. [6] implemented a grid-based e-government information portal as a ubiquitous communication channel between the government and its citizens, by taking grid services as its basic units and applying publishing/subscribing mechanism to transport messages between servers. Maad & Coghlan [7] explored the potential of grid infrastructure for e-government applications, by assessing the potential benefits of Opern Grid Service Architecture (OGSA) [8] in meeting the critical success factors for e-government, which are identified as integration, knowledge management, personalization, and customer engagement.

Li et al. [9] adopted the workflow middleware, transaction middleware and the real-name citizen mailbox in an e-government grid, enabling to provide coordinated, seamless and secure access to massive amount of data held across various agencies in a government in heterogeneous environments. Yang et al. [10] proposed a service grid based framework for the interoperability, which facilitates “horizontal” resource sharing and interoperability among “vertical” e-government subsystems. “Horizontal” means cross-organizational application, and “vertical” means information system within one organization. Terregov [11] is a European Union e-government project that adopts the principles of SOA based on interoperable components with dynamic support for finding e-government services. Terregov makes it possible for local, intermediate and regional administrations to deliver online a large variety of services in a straightforward and transparent manner regardless of the administrations actually involved in providing those services.

Zhang and Zhu [12] proposed the grid administration system and a case of the street civil service application system in Wuliqiao Street in Luwan District - a major district in Shanghai, China. Since urban government can provide to civil citizens high level services and good environment for living and working through grid E-Government platform. this study provides a model of community E-Government application utilizing design research approach and combining grid administration and service. Maad et al. [13] proposed to establish an appropriate match between the grid and e-government, which is achieved by comparing the grid concept, standards, programming paradigms and implementation with e-government as seen from the five viewpoints suggested by the standards and architectures of e-government applications. These different viewpoints are: the enterprise, the information, the engineering, the technology, and the computational. Carlos et al. [14] proposed the grid technology as an integration method of information, existing procedures and resources in the Public Administration in a gCitizen project, which is a grid middleware based on the GT4 components and WSRF [15] implementation (which are the state-of-the-art in middleware for Grid computing), incorporating new protocols and services which cover the requirements for the integration purposes in the eGovernment frameworks. The gCitizen middleware also defines a data model to provide interoperability in the exchange of the information among the different gCitizen services.

In our earlier work [16], the authors have also developed a grid infrastructure for distributed management of e-government resources across ubiquitous virtual governmental agencies. An ontology-based service-oriented approach to problem-solving in e-government is proposed, enabling to provide, in an open, dynamic, loosely coupled and scalable manner, the service publication, discovery and reuse for connecting the customers and agencies of e-government services based on their semantic similarities in terms of problem-solving capabilities. Such an infrastructure is depicted in figure 2, which consists of five layers: resource layer, ontology layer, middleware layer, application service layer and portal layer.

Notwithstanding the promising research results and implemented prototypes reported from existing research work for grid enabled e-government applications, there is still a major gap between these e-government systems and the vision of grid computing to realize more integrated solutions that accelerate the convergence of grid computing and database technologies. In the recent study of data grid, most of the researchers consider that the management of the data directory is equivalent to the management of metadata directory. But we argue that the function of data grid information service is to map the data access request into a particular data carrier. Hence, the data grid information service should contain two parts: metadata directory and resource information directory. Metadata directory takes charge of mapping the data access request into the request of data access carrier by creating information dynamically; while, the resource information directory takes charge of mapping the request of data access carrier into the particular data carrier, and its storage state is more stable.

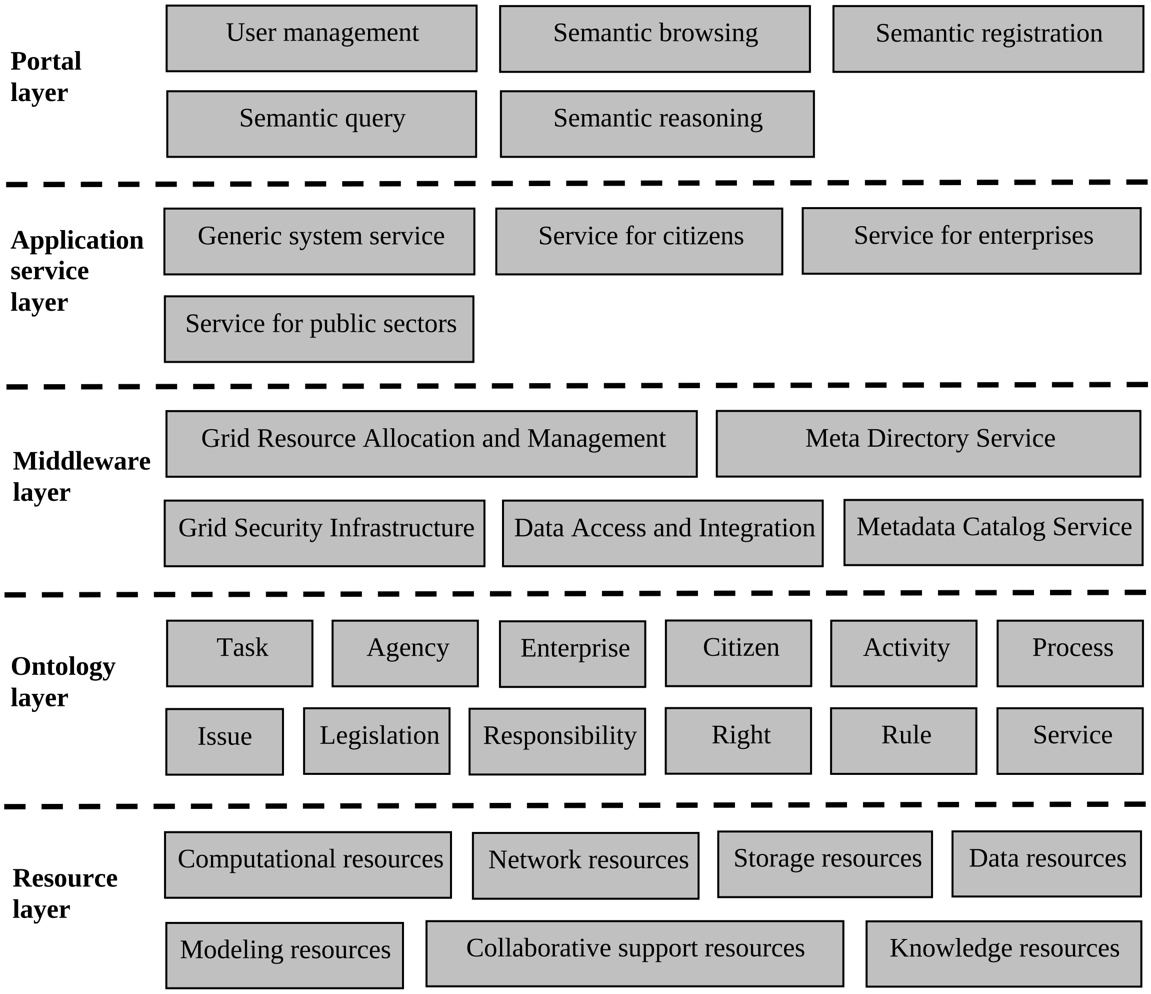

Figure 2. The Semantic Grid Infrastructure for e-government applications [16]

# 2 A Two-Level Data Resource Discovery Method

## 2.1 Definition of metadata ID

In order to store and manage metadata directory reasonably in the data grid, we establish a uniform data object name, and provide the same file name, orientation and access mechanism for all of the users. Different data views can also be provided according to different users' interests. This approach can reduce the metadata information for particular users in the system, thereby providing the efficiency of data orientation.

First of all, the difference between the external reference name and the global system identification name of a data object should be distinguished. The external reference name is the name of a data object when it is cited and searched normally by users, while the global system identification name is the global unique internal identification in the data grid [17].

**Definition 1**: The global system identification name in the system is constituted of 5-tuples (IDcreater, IDrunode, IDstorager, IDdb, Ta), and their definitions are as follows:

IDcreater: the ID of user who creates this data object;

IDrunode: the ID of grid node's address to which the data object registers;

IDstorager: the node ID of grid node field where the data objects are stored;

IDdb: the identification name of data object;

Ta = {table1|file1, table2|file2,…table*n*|file*n*}: the list of tables or files in IDdb data object for users to operate or inquire.

**Definition 2**: The whole constitution format of the global system identification name is as follows:

GlobalName =
IDdb@IDcreater.Idrunode.IDstorager

For example, an integrated global system identification name could be:

hardware@peter.washington.newyork

It is such a data object whose local name is hardware (such as hardware database), and which is registered in Washington grid nodes by Peter, but its data object is stored in Newyork node's field. This name could be ensured never changed in metadata directory, even when the data object is transferred to another grid node.

**Definition 3**: The constitution format of the integrated external reference name — CiteDataName is as follows:

CiteDataName = <IDdb->Ta>

User usually cites the data object through the external reference name. To cite an external reference name means to use the system identification names' identification ID and tables or files of data object. For example, when we inquire hardware's (the hardware database) car table, we can use hardware->car directly.

**Definition 4**: The metadata items describing the data object are constituted of GlobalName, data structure, context, and other descriptions.

## 2.2 Definition of resource information

**Definition 5**: The description information of data resource in its storage node contains IDdb in metadata — the identification name of data object and particular mapping information, such as the particular storage place of this data resource, the type of data resource, and so on.

## 2.3 Discovering the resource information through double-data directory

### 2.3.1 Metadata directory and resource information directory

**Definition 6**: The data information must include the metadata information when it is registered. Then these metadata is constituted of the metadata directory table which could be named as RegisteredTB.

**Definition 7**: After establishing resource information directory table in all of nodes which store data resource, this table can record all data resource of this node field. The directory table could be named as StoragedTB.

**Definition 8**: The corresponding table of the external reference name ("CiteDataName") and the address ID of grid nodes are registered by data object matched by semantics. The table can be stored in the grid node registered by data resource when it is created for the first time. We can name it as CorrespondingTB. It is used for the above metadata directory and resource information directory.

Based on the definitions above, if users want to search a particular data resource, they can get the external reference name through corresponding semantics analysis, and get the integrated global system identification name by definition 2, then get registration node through StoragedTB of definition 8, and finally find out the data resource by using metadata directory and resource information directory. Thus it can be seen, every grid node should maintain a table of CorrespondingTB for all of the users on this node, and they should allow the users know the external reference name and then locate the registration node.

### 2.3.2 Registered storage of metadata

Refer to figure 3 for the process of registered storage of metadada.

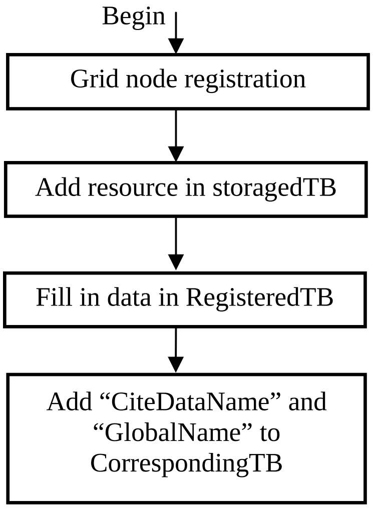
Begin
Grid node registration
Add resource in storagedTB
Fill in data in RegisteredTB
Add "CiteDataName" and "GlobalName" to CorrespondingTB

Figure 3 registration of metadata

1) When the data resource in a grid node field is registered in this grid node, they should add their resource data information in the local node StoragedTB;

2) then fill in the table RegisteredTB of registration node with registered metadata information, including global system identification name — GlobalName; and

3) fill in the table CorrespondingTB of registration node with the external reference name ("CiteDataName") and the corresponding items of

system identification name (“GlobalName”) at the same time.

Because the data resource object may migrate, the authors separate the storage data resource from the metadata information registered on the registration node. This way could bring three advantages. Firstly, it can ensure that the metadata information be stored only once in all of the grid nodes, and there exists no copy version or problem of consistence of metadata. Secondly, because metadata can be registered on any grid node, there is no salient centralized node in the system. Thirdly, every node can control the metadata directory of their own absolutely.

### 2.3.3 Discovering the data resource through double-data directory

Refer to figure 4 for the process of data resource discovery through double-data directory.

1) User submits a request that has semantic relations with the external reference name (“CiteDataName”). (This request is the single accessing statement illustrated above, such as a “SELECT” statement).

2) The system searches for the corresponding external reference name in CorrespondingTB of local grid node according to the semantics.

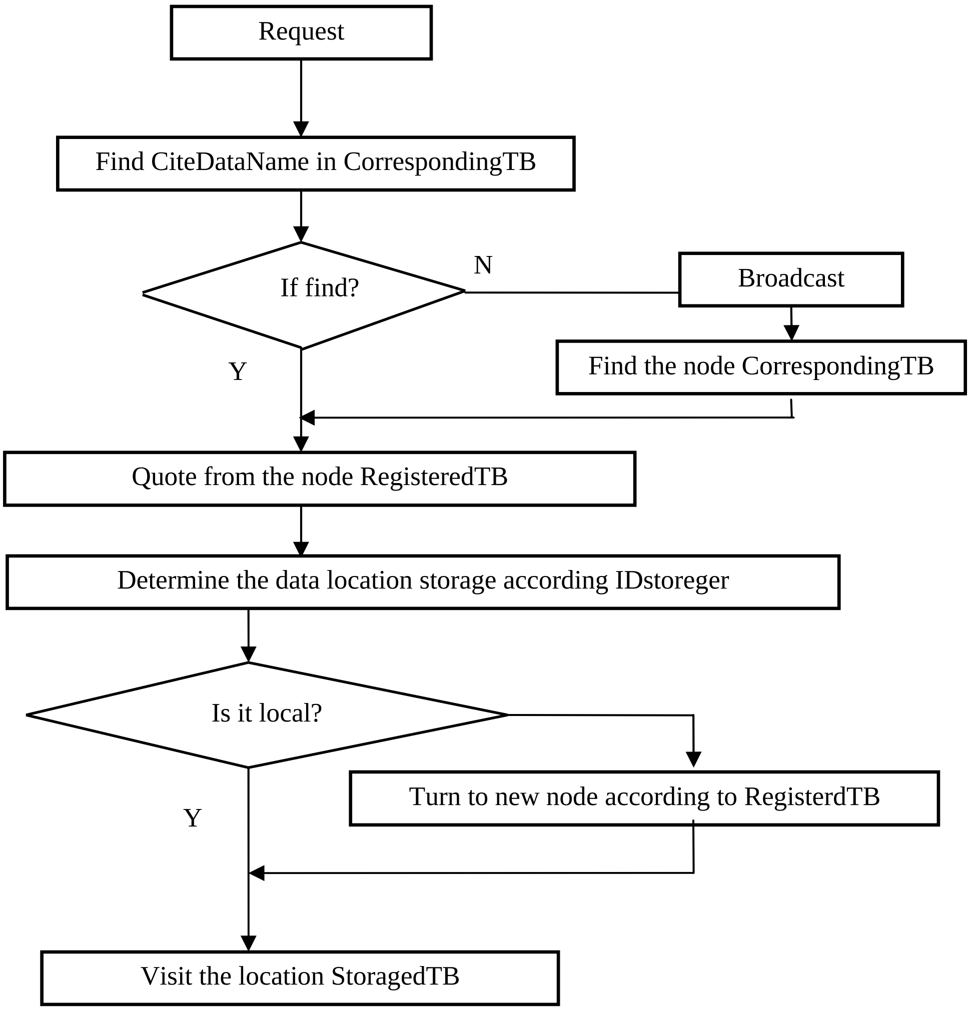
Request
Find CiteDataName in CorrespondingTB
If find?
N
Broadcast
Find the node CorrespondingTB
Y
Quote from the node RegisteredTB
Determine the data location storage according IDstoreger
Is it local?
Turn to new node according to RegisterdTB
Y
Visit the location StoragedTB

Figure 4 Data discovery through double-data directory

3) If found, according to the items in CorrespondingTB, the registration node of data resource need to be visited. For example, if we know that the registration node is Washington, we can inquire RegisteredTB's directory table of Washington. RegisteredTB is a metadata directory table and the global system identification name (GlobalName) in metadata contains data object's storage place Idstorager. According to definition 4 and definition 2, the directory table of Washington would contain a whole metadata item of this data object. If the required data object is stored in Washington, the system will search StoragedTB of Washington, then the data resource searched by user would be found finally.

However, if the data object has been migrated to New York, the items of RegisteredTB's directory table will show this information. Then the system can inquire StoragedTB of node in Washington (the second distant access). According to definition 7, StoragedTB's directory table in New York will contain an item of this data object, finally this data object can be found by two remote accesses at most. If this data object is to be migrated to Boston again, the system will insert an item into StoragedTB of Boston node, delete the directory item in New York node, and modify the item of RegisteredTB in Washington as pointing to Boston, not New York. By this way, this data object can be found through two remote accesses.

Figure 5 shows the necessary changes made to migrate the data object.

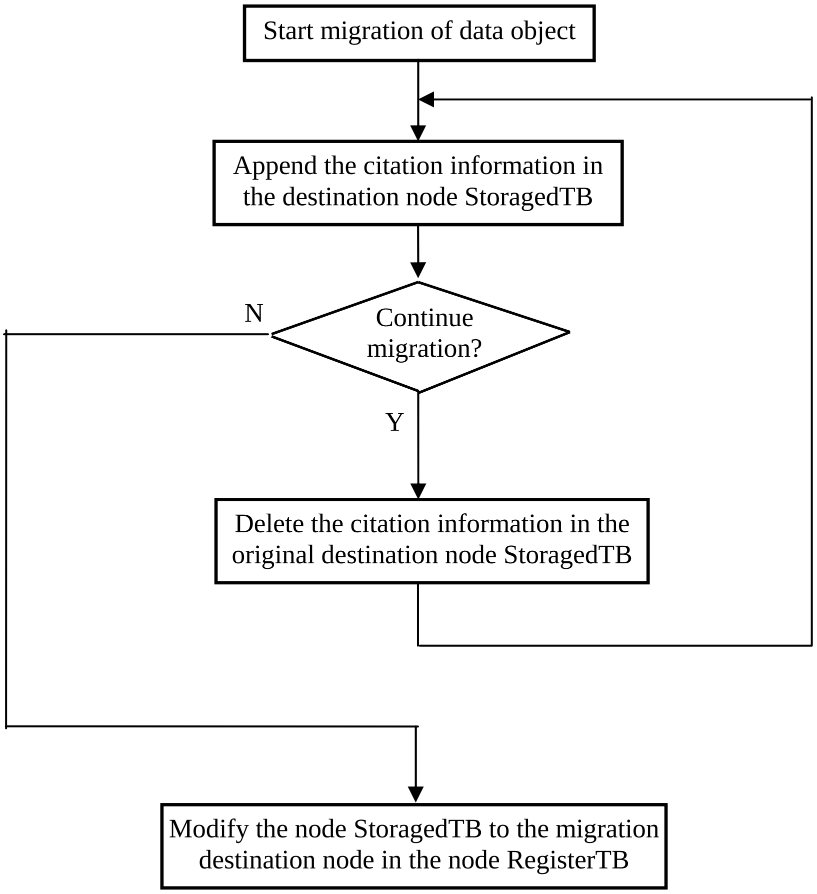
Start migration of data object
Append the citation information in the destination node StoragedTB
Continue migration?
N
Y
Delete the citation information in the original destination node StoragedTB
Modify the node StoragedTB to the migration destination node in the node RegisterTB

Figure 5 Data migration

4) If the system hasn't found the corresponding external reference name in CorrespondingTB's directory table of local grid node, it will broadcast to all registration node servers in the grid, and search the corresponding external reference name in the CorrespondingTB. Then system will go to

]

registration grid node according to Idrunode in CorrespondingTB, and go to RegisteredTB of this node next. The following steps are the same as the above step (3). At the same time, the system appends this information in local CorrespondingTB, and registration's Idrunode should be appended too, so that the request from local node can be searched locally the next time and the registration node can be got through Idrunode. The global system identification name (GlobalName = <IDdb@IDcreater.Idrunode.IDstorager>) stored in RegisteredTB plays a very important role in searching the data object.

# 3 Discussions

## 3.1 Restraining repeated answers when searching CorrespondingTB of each node by broadcasting

As described in the last section, because the feedback from broadcasting will append information to the local CorrespondingTB, the information of one external reference name will be stored repeatedly in many grid nodes' CorrespondingTB. The system won't stop broadcasting to all registration nodes' servers in the grid until the user finds the corresponding external reference name in the index of local grid nodes' CorrespondingTB. This will cause two obvious abuses:

1) As soon as each middle node receives the broadcast, it will copy the information and send to other neighbor nodes except the source nodes. Then, the network will be full of these copies rapidly, and one node will receive and send the information repeatedly, causing the net crowded. 2) It will answer repeatedly because of storage redundancy existing in CorrespondingTBs of different grid nodes. The repeated response will result in access to the corresponding data source repeatedly which is forbidden in data grid system.

The resolution to the above is to identify time labels to the request sent out by users. The request has semantic relationships with external reference name. In this way, for the first problem of the related nodes receiving semantic requests which have the same time labels in succession, the result is that system will only transmit the first request received, and filter the others off automatically. And for the second problem, when the system finds the external reference name (CiteDataName) corresponding to semantics in CorrespondingTB of one node, it will attach this time label in the same way, so when the information searched returns to the storage node of data resource, StorageTB will answer the first one, and filter the feedback with the same time label off automatically, thereby ensuring single answering of data resource under one searching semantics.

## 3.2 The search efficiency analysis

1) If the corresponding external reference name could be found in metadata directory of local nodes, the searching time can be ignored.

2) If the corresponding external reference name could not be found in metadata directory of local nodes, it will be searched by broadcasting. Because it only transport external reference name which has short length, we can ignore the consumption on network bandwidth by the worst route during broadcast and network transportation. So we can find out the external reference name quickly. As soon as the external reference name is found, the remote data resource could be found through two acesses at most.

## 3.3 Discussion about storage in CorrespondingTB

Because each node of CorrespondingTB has characteristic of changing frequently, in order to ensure its search efficiency and internal robustness, we can establish mechanism of giving priority to frequently used data and clearing rarely used data. According to this mechanism, for the tuples constituted by corresponding external reference names used frequently in local system, the system can rank them in the front of CorrespondingTB, and for the element group rarely used, the system can clear them. By this mechanism, we can improve the efficiency of reading and writing. Because the hot data accessed frequently increases in local nodes, the probability of hitting the target of external reference name in local system increases greatly, then the load of network communication could be alleviated.

Choosing the hot metadata in remote field which is accessed frequently by local system will decrease the updating frequency to local CorrespondingTB. In this way, it can not only increase users' access efficiency, but also cost not too much overhead of synchronizing the local system and remote system because of low updating frequency.

# 4. Summary

This paper has presented an approach of automatic discovery of data resources in the data grid. The approach can support distributed e-government applications across ubiquitous virtual governmental agencies. Because the proposed system adopts double-directory table, the global name (GlobalName = <IDdb@IDcreater.Idrunode.IDstorager>) can be stored only once in the entire data grid, and there are no copy and no consistency problem of metadata copies. There is no salient centralized node in the system, and each node can self-govern its metadata directory completely. The approach can also be extended for other grid applications, such as e-science, e-biology, e-commerce and e-business.

Of course, there are still some problems to be resolved in this resolution, for example, when users send out the information of searching data resource, in order to realize the resolution above, the authors need to do more research in our future work on the solution of matching the searching terms in users' digital signature and the semantics of external reference name.

## Acknowledgement

The work has been supported by the Zhejiang Natural Science Foundation of China (No. Y1080170) and Zhejiang Significant Science & Technology Project with Special and Priority Topic (No. 2007C13050).

*References:*

[1] Foster, I. and Kesselman, C., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999.

[2] Hoschek, W., Jaen-Martinez, J., Samar, A., Stockinger, H. and Stockinger, K., Data Management in an International Data Grid Project. In: *Proceedings of the 1st IEEE/ACM International Workshop on Grid Computing*, Bangalore, India, pp. 333-361, Dec. 17th 2000.

[3] Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. and Tuecke, S., The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Joumal of Network and Computer Applications*, 23, pp. 187-200, 2000.

[4] Wang, H., Xu, Z. W., Gong, Y. L. and Li, W., Agora: Grid Community in Vega Grid, *Lecture Notes in Computer Science*, 3032, pp. 685-691, 2004.

[5] Silva, F. and Senger, H., The Grid: An Enabling Infrastructure for Future E-Business, E-Commerce and E-Government Applications, In: Mendes, M. J., Suomi, R. and Passos, C. (Eds.) *Digital Communities in a Networked Society: e-Commerce, e-Government and e-Business*, Dordrecht - Holanda: Kluwer Academic Publishers, pp. 253-266, 2004.

[6] Fu, X. F., Peng, D., Xu, H. S., Lu, Y. S. and Zhan, Y. W., Research and Implementation of E-Government Information Portal Based on Grid Technology, In: *Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design*, Coventry, UK, May 24-26, 2005, Lecture Notes in Computer Science, 3865, pp. 141-150, 2006.

[7] Maad, S. and Coghlan, B., Assessment of the Potential Use of Grid Portal Features in E-Government, *Transforming Government: People, Process and Policy*, 2 (2), pp. 128-138, 2008.

[8] Foster, I., Kesselman, C., Nick, J. and Tuecke, S., Grid Services for Distributed System Integration, *Computer*, 35, pp. 37-46, 2002.

[9] Li, Y., Li, M. L. and Chen, Y., Towards Building E-government on the Grid. In: *Proceedings of the International Conference on E-government: Towards Electronic Democracy, TCGOV-2005*, Bolzano, Italy. *Lecture Notes in Artificial Intelligence*, 3416, pp. 205-212, 2005.

[10] Yang, D. J., Han, Y. B. and Xiong, J. H., eGovernment: A Service-Grid-Based Framework for E-Government Interoperability. In: P*roceedings of IFIP International Federation for Information Processing*, Volume 252, *Integration and Innovation Orient to E-Society*, Volumn 2, pp. 364-372.

[11] Busson, A. and Keravel, A., *Prospective Study Report on Interoperability in Local e-Government Projects*, Technical Report TERREGOV Project, 2005. Available: www.egovinterop.net.

[12] Zhang, P. Z. and Zhu L., A Case Study on Urban Community E-government: From Grid Administration to Grid Service. In: *Proceedings of the 1st International Conference International Conference on Theory and Practice of Electronic Governance*, Macao, China, pp. 449-450, 2007.

[13] Maad, S., Coghlan, B., Ryan, J., Kenny, E., Watson, R. and Pierantoni, G., The Horizon of the Grid for E-government. In: *Proceedings of the eGovernment Workshop*, Brunel University, United Kingdom, 2005.

[14] Carlos, de A., Miguel, C., Jose, V. C. and Vicente, H., gCitizen: A Grid Middleware for a Transparent Management of the Information

about Citizens in the Public Administration. *Journal of Theoretical and Applied Electronic Commerce Research*, 2, pp. 18-32, 2007.

[15] Snelling, D., Robinson, I. and Banks, T., OASIS Web Services Resource Framework (WSRF) TC Website. [Online]. Available: http://www.oasis-open.org/committees/wsrf, 2004.

[16] Zhang, W. Y. and Wang, Y., Towards Building a Semantic Grid for E-Government Applications, WSEAS Transactions on Computer Research, 3 (4), pp. 273-282, 2008.

[17] Date, C. J., *An Introduction to Database Systems*, Addison-Wesley, Boston, 2000.