An Efficient Distributed Weather Data Sharing System Based on Agent

Tinghuai Ma[1], Hao Cao[1], Donghai Guan[1], and SungYoung Lee[2]
[1]Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, China
[2]Department of Computer Engineering, Kyung Hee University, Korea

Abstract: *This paper presents a multi-agent based framework for managing, sharing, and accessing weather data in a geographical distributed environment. There are two tiers in this framework including national central centre and local centre. In each node, the services for querying and accessing datasets based on agent environment are designed, which includes data resource agent, local management system, and metadata system. Information retrieval can be conduced either locally or distributed, by querying local weather data or exploiting global metadata respectively. With a variety of advantages, this proposed platform is designed to provide a useful platform for research on weather data sharing system in a national area.*

Keywords: *Weather data, sharing, distributed, agent-based, data management.*

Received July 27, 2009; accepted May 20, 2010

1. Introduction

With the development of information technologies, the research on climate and meteorology has also made a great progress with the ability to collect and use a huge amount of data. China uses 158 weather radars to sense climate information. Each of these radars generates about 6GB data per 6 minutes and about 1TB data per day. A huge amount of data thus will be used for meteorology prediction every data. Especially climate modelling becomes a challenging problem as extremely long-duration computations are needed to analyze the simulated climate [2, 13]. Therefore, it is crucial to develop a new infrastructure which can support management, transport, distributed access, and data analysis for thousands potential users.

To understand the complex climate processes, the geographically distributed research teams need the capability to effectively and rapidly retrieve new knowledge from these massive and distributed data holdings, and to share the results with a wider community. Achieving this capability requires new methodologies involving managing, accessing, recombining, analyzing, and distributed data inter-comparing.

'Data Grid' is a desirable solution [4, 5, 12] for distributed data management. To meet the requirement of both Chinese Meteorology Administration's operation and researchers, the system software should be national collaborative, metadata and service oriented. And the required capabilities for the framework include: containing detailed knowledge of its components and status; constructing system dynamically; seeking to optimize its behaviour to achieve its goals, and being aware of its environment.

Some of the requirements with this distributed data grid applications are potentially conflicted. Here agent technology is one of the solutions to meet these conflict requirements in the grid development [5, 9, 18]. First of all, deploying a big number of nodes requires that the node systems should be simple and with minimal requirement on local sites. For this reason, it is desirable that the system can deploy them automatically. Second, since these systems need to achieve many different complex and performance-sensitive applications, they have to provide a range of potentially sophisticated services. In addition, the workflow and dataflow of these systems cannot be decided as they are complicated. Also, the cooperation between system modules is not clear. Using agent's autonomous ability can develop a complex software system as a society. Third, the system should be scalable and evolvable for future systems and services. If the components are a system are not known in advance, the system's structure should be dynamically. Agent can be added into the system anytime. Furthermore, agent composed system could be highly heterogeneous.

Agent-based system can satisfy the functions of data grid, which in turn solves the requirements of distributed climate data managements. We can view the data grid as a number of agents interacting with each other based on knowledge. Developing intelligent agents to realize the grid vision have been made by academic researchers during the past several years [7]. The agent grid is a specific construct or mechanism within that layer for making services and resources available. Some agent-based grid computing (AGEGC) system has been built [17].

The rest of the paper is organized as follows. Section 2 introduces some works on meteorological data sharing. Section 3 introduces the background and data type of meteorology. Section 4 describes the conceptual model of agent-based sharing model. We describe the data accessing process locally and remotely in section 5. Finally, we point out the future work and conclude the paper in section 6.

2. Related Works

Accessing the national meteorological information is the key issue of weather prediction to decrease and avoid natural disaster. From 2001, Chinese Meteorology Administration (CMA) began to provide meteorological information publicly and openly. During 2001 and 2005, CMA has built a centralized sharing platform to provide history meteorological information to meteorological scientist over world.

It is noted that almost all the meteorological data is national geographically distributed. Therefore, the centralized storage method requires high bandwidth, high performance computing machine and huge disk space to transfer, manage and store data. Thus, distributed storage methods are adopted [15]. American established the Earth System Grid (ESG) project [2], which goal is creating a next-generation environment that harnesses the combined potential of massive distributed data resources, remote computation, and high bandwidth wide-area networks as an integrated resource available to the research scientist. It enhanced the data transfer security, metadata management and request management issues in the ESG-II [6].

The distributed data storage and management idea has also been used in other hot issues, such as bioinformatics. The data produced by mass spectrometry, which has been used in proteomics experiments, is geographically distributed and heterogeneous because it produced by different laboratory. Moreover, this data is enriched by observations and descriptions added by specialists through metadata. P. Veltri etc., present a software system for managing, sharing, and querying data in a distributed laboratory, which is a large and scalable storage solutions, and high performance computational platforms [19].

The distinct requirements of the distributed data management software can be addressed by the agent software architecture. We regard every distributed node as an agent. Agent provides service to access local data. System software maintains the service directories of all agents. These are main concepts of agent-based distributed data management [4, 18]. Shi Z. developed a software system, which provides specific construct or mechanism makes services and resources available remotely [16]. They developed a multi-agent environment replacing any grid middleware to maintain the services' metadata and data transfer protocol. This is not the only realization for any group to develop a distributed data, service management. Based on grid computing, utilizing grid middleware, metadata publishing, resource advertisement and discovery, efficient, reliable, secure data transfer etc. can be automatically excused, as P. Veltri mentioned [19].

3. Weather Data

In China, the country consists of eight region areas to manage its weather information. In the centre of each region area, there are four types of information:

1. Observation information obtained from ground, upper-level air sounding, solar radiation and agro-meteorological stations, and some statistical products from this information.
2. Lattice information from variety of numerical model of assimilation, and various numerical inversion products of remote sensing data.
3. Graphics and video information of all types of satellite images and radar images.
4. Comprehensive integrated meteorological data in a particular region area with specially theme.

We need to design a framework that worked in an agent-based environment, which composed of several nodes, each one (node) associated to region centre database server (called node from now on) run its local database. The framework should efficiently support distributed storage and manipulation of weather data. Each node contains an application program interface mounted on a data storage system hosting weather data produced within its region. Weather data are stored using three different formats: raw file, structured data and XML file [11].

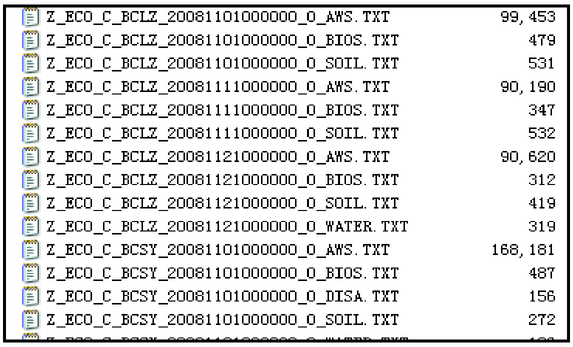
3.1. Raw File

Raw files: original data produced by the observation equipments, remote sensing equipments, lattice information, satellite images and radar images. Figure 1 shows the data stored in file system. Figure 2 shows the content of raw file which records the ecology factors.

3.2. Structure Data in Database

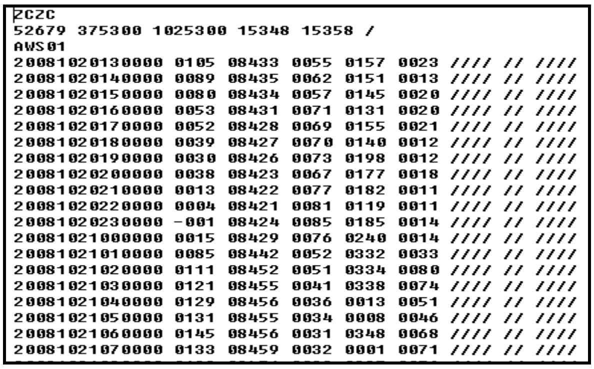
Structure data is stored a relational database. An Oracle database instance is used to store the data from automated observation station, which origin data is stored as raw file.

Especially, we construct a mete database to describe the raw data. Searching raw data file is transfer to searching metadata in database. The record in Meta database links to raw data file through *BFILE* attribute.



```
Z_ECO_C_BCLZ_20081101000000_O_AWS.TXT       99,453
Z_ECO_C_BCLZ_20081101000000_O_BIOS.TXT         479
Z_ECO_C_BCLZ_20081101000000_O_SOIL.TXT         531
Z_ECO_C_BCLZ_20081111000000_O_AWS.TXT       90,190
Z_ECO_C_BCLZ_20081111000000_O_BIOS.TXT         347
Z_ECO_C_BCLZ_20081111000000_O_SOIL.TXT         532
Z_ECO_C_BCLZ_20081121000000_O_AWS.TXT       90,620
Z_ECO_C_BCLZ_20081121000000_O_BIOS.TXT         312
Z_ECO_C_BCLZ_20081121000000_O_SOIL.TXT         419
Z_ECO_C_BCLZ_20081121000000_O_WATER.TXT        319
Z_ECO_C_BCSY_20081101000000_O_AWS.TXT      168,181
Z_ECO_C_BCSY_20081101000000_O_BIOS.TXT         487
Z_ECO_C_BCSY_20081101000000_O_DISA.TXT         156
Z_ECO_C_BCSY_20081101000000_O_SOIL.TXT         272
```

Figure 1. The ecology factor files, which is stored in OS as txt file, the number means the files capacity.



```
ZCZC
52679 375300 1025300 15348 15358 /
AWS01
20081020130000 0105 08433 0055 0157 0023 //// // /////
20081020140000 0089 08435 0062 0151 0013 //// // ////
20081020150000 0080 08434 0057 0145 0020 //// // ////
20081020160000 0053 08431 0071 0131 0020 //// // ////
20081020170000 0052 08428 0069 0155 0021 //// // ////
20081020180000 0039 08427 0070 0140 0012 //// // ////
20081020190000 0030 08426 0073 0198 0012 //// // ////
20081020200000 0038 08423 0067 0177 0018 //// // ////
20081020210000 0013 08422 0077 0182 0011 //// // ////
20081020220000 0004 08421 0081 0119 0011 //// // ////
20081020230000 -001 08424 0085 0185 0014 //// // ////
20081021000000 0015 08429 0076 0240 0014 //// // ////
20081021010000 0085 08442 0052 0332 0033 //// // ////
20081021020000 0111 08452 0051 0334 0080 //// // ////
20081021030000 0121 08455 0041 0338 0074 //// // ////
20081021040000 0129 08456 0036 0013 0051 //// // ////
20081021050000 0131 08455 0034 0008 0046 //// // ////
20081021060000 0145 08456 0031 0348 0068 //// // ////
20081021070000 0133 08459 0032 0001 0071 //// // /////
```

Figure 2. The content of ecology data in raw txt files, the character "/" means missing data.

3.3. XML File

Meta information is stored as XML element instances, using simple coding compression formalism. The XML representation is very useful to share whole data among nodes. Moreover, using an XML representation, it is possible to define simple views on XML data thus achieving a fine-tuned personalized access to such data. Each node may define a personalized view thus allowing remote clients to query local data.

4. System Architecture

Weather Data Sharing System (WDSS) requires the collection and analysis of data produced in national distributed observation stations. The collection, storage and analysis of huge weather data in a right way can leverage the computational power of distributed environment. We choose a grid-based architecture that offers efficient data transfer, effective management of large data stores, the computational power which required by huge data manipulation, and the security, privacy requirement while data belonging to different organizations. The data management is the principal operation of the weather data sharing system [2, 4]. The basic function of resource management is to accept requests for resources from users within the grid and assign specific node resources from the overall pool of grid resources, for which, the user has the access permission. A resource management system matches requests to resources, schedules the matched resources, and executes the requests using the scheduled resources. Meanwhile, data resources are provided by agents as services, so service management is important. Now, WDSS is mainly concerned with resource management and service management. The service on each node should include data exploration, data querying, data transforming, and data delivering service. Nevertheless, user never accesses directly to remote data sources but just retrieves data through a query service issued to the data sources of each node. The architecture of the proposed system is composed by a set of agent based grid nodes. Each node runs:

- An agent management service, providing an interface on weather data(stored in relational database, XML and raw file formats) for data retrieval;
- A metadata catalog service [3], providing yellow pages services to other agents. Agents may register their services with the metadata catalog service or query the metadata catalog service to find out what services are offered by other agents;
- Data agent, representing local data resource in a grid environment. Local resource means the data stored in each node, which can be database or file.

In Chinese Meteorology Administration, the National Meteorological Information Centre acts as a central node, around with eight regional centre nodes. Figure 3 demonstrates the resource management architecture of WDSS.

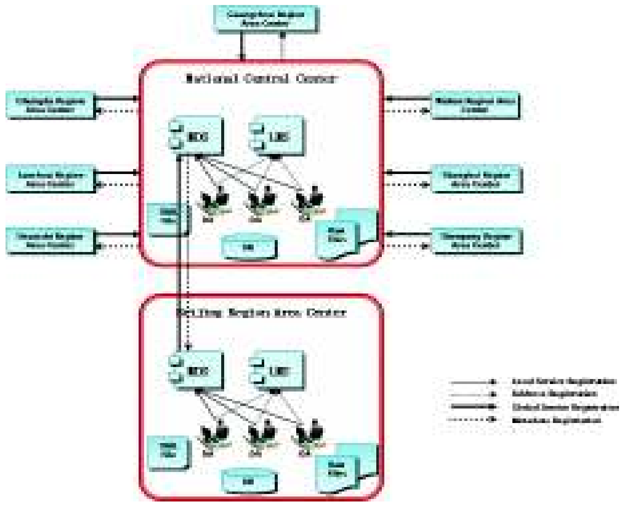


Figure 3. Data management architecture in WDSS, which has one central and 8 region centre points.

4.1. Overview of WDSS

WDSS is composed of eight region local nodes and one national central node. Every node acts one sharing GUI and provides sharing data for different users. The components in one region area centre consists a local Multi-Agent System (MAS). As shown in Figure 3, every local MAS has mainly components as: Data resource Agent (DA), Local Management System (LMS) and Meta Data System (MDS):

- *DA* is the representative of a Hybrid data resource. The data types now are flat file, relational database and XML files. These data resource can be modelled as a service. Each DA should register itself as service in metadata system. Further more, the high performance computing power can be encapsulated as service also, in the later time. In that time, Data resource Agent (DA) will be renamed as Resource Agent (RA).
- *LMS* is the manager of the local multi-agent system and offers white pages services (mapping from agent name to agent address) to other agents. It provides the interface for user to retrieval information from WDSS. The interface must be based on web because the user maybe in CMA or out. Trough the unified interface, users cans retrieval locally and over whole grid.
- *MDS* provides yellow pages services to other agents, especially for MAS user [3]. DAs may register their services linked data source with the MDS or query the MDS to find out what services are offered by other agents. Local MDSs register their managed service to the national central centre MDS. MDS in national centre distributes the integrated services' registration over whole grid system. The service registration information we called metadata, includes data type, generated time, the generated place, store place, services, invoke methods etc. All these information transferred in XML format.

The architecture has only two tiers, central centre and local centre. For real geographically distribution, every region area centre dominates several provinces, and every province holds several cites. So, the data resource management should be a hierarchical organization. The low organization can register its services to upper organization's MDS; upper organization's MDS distributed metadata to its dominated domain in XML format.

4.2. Local Management System (LMS)

LMS is a collection of tools supporting the entire function of data retrieval. Within it, Agent Management Service (AMS) is a mandatory component to exert supervisory control over access to agent service. Through GUI, users query and download data; administrator adds new data resource to LMS. While new data resource is adding, an agent linked with this data registers its service to AMS, and AMS will assign this Agent an Identifier (AID). Each agent must register with an AMS in order to get a valid AID. The AMS maintains a directory of AIDs. There are some assistant tools in LMS such as network weather prediction service; data organize service, data FTP service and message transport service. Figure 4 is the logical architecture of LMS's function.

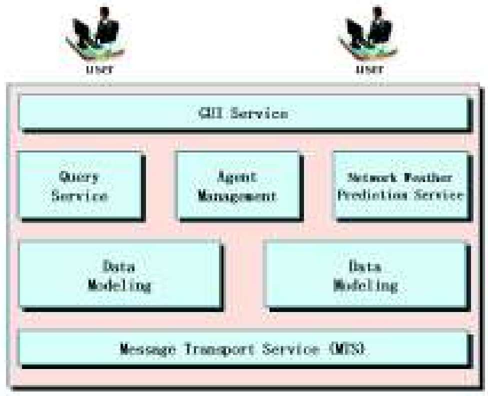


Figure 4. Local management system architecture.

- *GUI Service*: Allows the user to retrieve weather data from remote nodes and to perform services. It provides an interface for querying data supporting retrieval functions based on Meta information such as generate date, equipment type, generate place and so on. Of course, LMS offers API to query and retrieval data too. The other applications can invoke the API for data retrieval for weather production generation. The interface also allows performing data pre-processing and storing pre-processed production in the data storage. The functions include selecting input and output data sources, loading raw files, saving metadata information and so on. As shown in Figure 5, the main GUI for users to retrieval data.

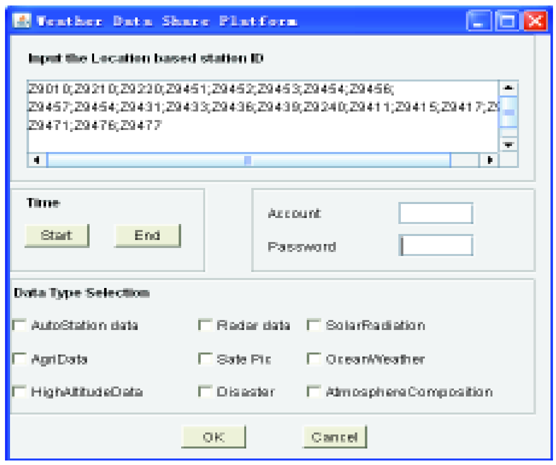


Figure 5. The data source selection GUI. From here, user can decide the data type, time period, and location.

- *Query Service*: Is a set of specialized APIs to enhance performance of data pre-processing and analysis. When a user wants to perform a query against the grid, as a first step he/she contacts the local Query Service. Such module parses the query, accesses to the MDS and generates a query plan with list of nodes to be queried. The MDS Service is organized as a set of data domains encoded in a set of XML documents, one for each node of the grid.

- *Agent Management Service*: Is a mandatory component in multi-agent system (MAS). It acts as a control role in system. Only one AMS exists in MAS. The AMS maintains a directory of AIDs (agent identifiers), which contain transport addresses (amongst other things) for agents registered in MAS. The AMS offers white pages services to other agents. Each agent must register with an AMS in order to get a valid AID. AMS manages the agent's life cycle, responsible for the creating and destroying of agent. AMS is also responsible for running the application and returning the results.
- *Network Weather Service* (*NWS*) [20]: Provides network performance measurements and predictions. The query service uses NWS information to select the replica of the desired data that is likely to provide the best transfer performance. This is an additional function for best choosing while the data has replica in other place. Now, the region area centre and national information centre storage the same data simultaneity. A request for data has choice from two places (origin region area centre and national information centre). The important critical is the network performance.
- *Data Organize Service*: Provides a platform to add new data source to the WDSS. It allows selecting input and output data sources. With this function the user has the control of the data flow; he/she can choose where to get data from and where to save results. On the platform, administrator can edit metadata for data resource, load data into storage, save the metadata.
- *Data FTP Service*: Provides efficient, reliable, secure data transferring in grid computing environments. It can be a simple prototype of GridFTP [1]. For any data grid solution, data accessing and data transferring must be critical components. LMS must move the file securely over the wide-area network to its destination. FTP is commonly used for data transfer on the Internet and the most likely candidate for meeting the Data Grid's needs. The data transfer monitoring information is shown in Figure 6.

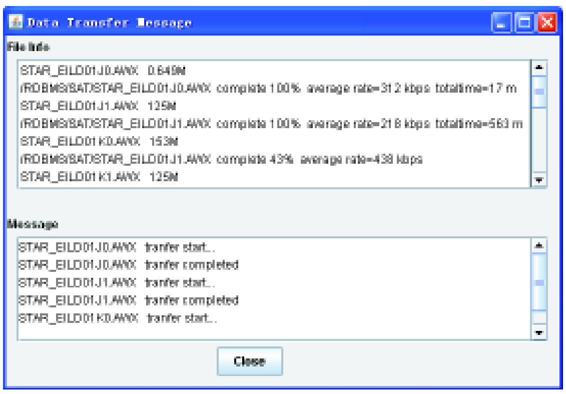


Figure 6. Data transfer monitoring information.

In which, the file in waited queue can be find in above and the file transferring indicated in below.

- *Message Transport Service*: Is the default communication method between agents. It is based on FIPA Agent Management Specification [8], and there are some products to choose.

4.3. Data Agent

DA is a representative of a Hybrid data resource. These data resource can be accessed by services provided by its agent. Every data has its mapped agent and services. One data agent can have several services to access the same data resource. Every agent holds its AID created by AMS, and registers its metadata information into MDS. As an agent, data agent has common characters [17], its structure shown in Figure 7.

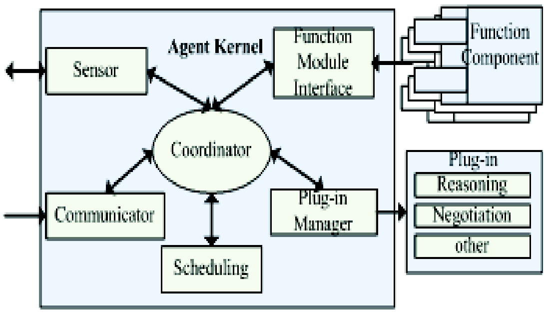


Figure 7. Architecture of the generic agent.

Agent architecture should satisfy the software's flexibility and capability. Capability means software module is powerful enough to handle things such as co-operation, negotiation, mobility and so on. We hope the agent can have more functions better. But complexity of agent's structure will embarrass the agent's deployment. This will decrease the agent's expansibility. For system's flexibility, agent is simple enough to deploy, and could be customized by users with their own defined components. This supports the system structure's dynamic changing. So, agent structure definition is a trade-off between flexibility and capability.

Generally, agent has a kernel and some plug-in like components [16, 17]. Kernel components are solution of flexibility and plug-in like component are solution of capability. Agent kernel consists of the following parts. Sensor perceives the outside world. Function Module Interface makes an effect to the outside world. Communicator handles communications between the agent and other agents. Coordinator makes decisions concerning the agent's goals, and it is also responsible for co-coordinating the agent interactions with other agents using given co-ordination protocols and strategies, e.g. the contract net protocol or different kinds of auction protocol. Scheduler plans the agent's tasks based on decisions taken by the Co-ordination Engine and the resources and task specifications available to the agent. Resource Ontology maintains a list of ontology describing the semantic content of resources that are owned by and available to the agent. Task ontology provides logical descriptions of tasks known to the agent. Plug-In Manager manages the components provided by users that can be plugged into agent kernel. Plug-In Manager manages the components provided by users that can be plugged into agent kernel.

5. Prototype System Implementation

5.1 Data Retrieval Process

Data retrieval can be executed anywhere through Internet. Every region area centre provides user the web page for data querying and retrieving, and provides APIs for application programs too. These function realized by LMS, which has twofold goal:

1. Retrieving efficiently portions of weather dataset for local weather product producing (e.g., making local weather prediction).
2. Retrieving weather datasets shared on the grid through XML querying.

In the following, some examples of both applications are briefly described. The user may use Query Service performing queries on the XML database through GUI. The current architecture considers a Query Service that supports both queries on XML file, performed by using JDOM [10] on XML Schema, and SQL like queries on relational Oracle data. There is an appropriate query translation module.

Retrieval data from WDSS is divided into three big steps:

1. Invoke the query service to input the query condition.
2. Query service search the metadata information, match the right DA whose data can satisfy the query.
3. AMS invoke the data matched DA to load data and transfer it to user.

The detailed processes are shown in Figures 8 and 9 according to the data resides on locally or on grid separately. For local enquiry, all operations are executed in one LMS, which is physically mapped to one node. As shown in Figure 8, step 1 is invoking the query service, steps 2 to 5 is searching Metadata information, steps 6 to 7 is data loading and transferring. The Metadata information query is in the local LMS.

As shown in Figure 9, step 1 is invoking the query service, steps 2 to 6 is searching Metadata information, steps 7 to 8 is data loading and transferring. The difference between Figures 7 and 9 is that there are two query processes in Figure 9. One is step 2, query in local LMS. The other is step 3, query metadata information in MDS of NCC, because it can not find right Metadata information in local MDS. This means query data is stored in remotely.

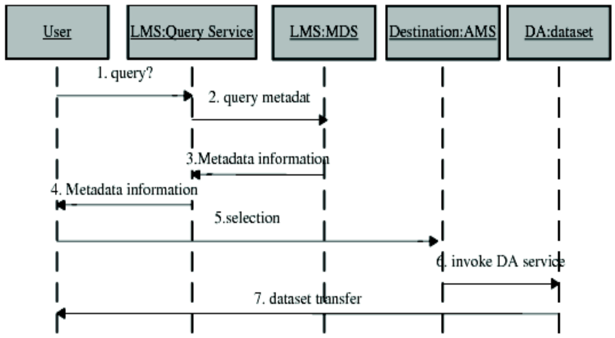


Figure 8. Querying process in WDSS locally.

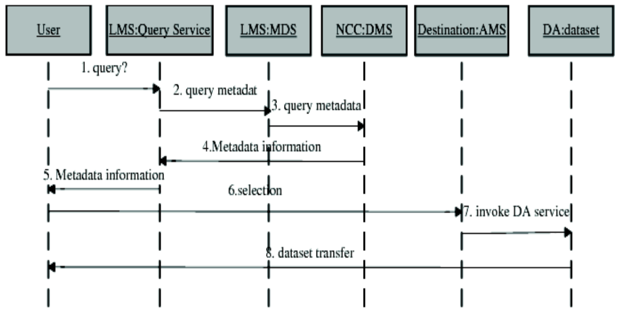


Figure 9. Querying process in WDSS globally.

5.2. Prototype System Development and Evaluation

We have already developed the system prototype based on our proposed architecture. Two centres are involved in this prototype, including national central centre and Beijing regional centre. The system configurations of these two centres are as follows:

1. National central centre: including 2 Intel 2.10 GHz *4 nodes with 16 GB of RAM memory which are connected by a Gigabit network; output bandwidth is 155 Mbps with SDH; the hard disks in each node provide approximately 50 MB/s.
2. Beijing regional centre: including 2 Intel 1.65 GHz *4 nodes with 16 GB of RAM memory which are connected by a Gigabit network; output bandwidth is 8 Mbps; the hard disks in each node provide approximately 30 MB/s.

In the client side, we used an Intel Dual-Core 2.10 GHz system featuring two hard disks set in a RAID 0 array. This high-performance client is used to prevent the bottleneck.

In terms of the efficiency of the system, the most information aspect is the search time. In the evaluation, we consider two kinds of search time: local search time and global search time. The two variables which are important for the evaluation include the type of agents and the number of agents.

There are several observations which we can get from Figure 10. Firstly, both local search time and global search time to locate a data agent are slowly increasing with the number of registered agents. The increased time to locate data agents is because of two factors: increased processing time of query service, and the increased network delay when returning responses from remote notes.

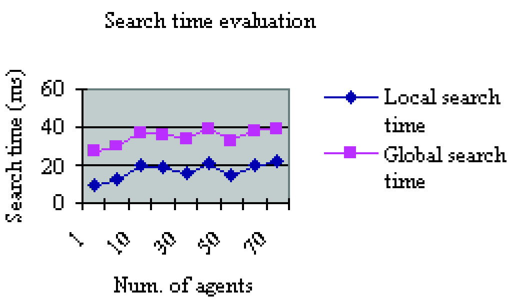


Figure 10. Search times vs. type of agents and number of agents registered.

Secondly, although the big trend in Figure 10 is that more agents lead to more search time. But this is not absolute. When the number of data agent is less than 10, the search time increases with the number of agent. However, when the number of data agent reaches to 10, this relationship shows some vibrations: the search time will not absolutely increase with the number of agent.

The two observations are not supervised. Searching data agent information means finding appropriate data agent description in Metadata. The size of metadata will increase with the number of agent. When the agent number increases in the small scale (1 to 10 in our evaluation), more search time is required since the size of Metadata file become larger. However, when the agent number is above some value (10 in our evaluation), the searching time shows the randomness behaviour as the right agent description is randomly located in Metadata. It is noted that small number of agent does not have the enough randomness.

5.3. Discussion

As a new data sharing platform, WDSS has mangy advantages compared with CMA's centralized sharing platform. Future more, in some areas, WDSS is better than ESG in America because it adopts the agent technology. But, as each system runs in an especially environment, including servers, network bandwidth, the performance of these system are difficult to evaluate in quantitative. Though, the prominent character of WDSS is the extensibility.

As we know, the LMS is charge with data retrieval, and the metadata represented the data. So, adding new centre into the whole system only need modifying the metadata, this is maintained by MDS. The new LMS in the new centre may be developed on a different agent platform. Thus, the autonomic new centre can be integrated into the whole system easily. For the data agent level, adding and removing data agents is trivial because our platform is a multi-agent system. Changing the data agent will not affect other agents' operation. Once the new data is added or changed, only the Metadata XML file needs to be changed.

Thus, what ever centre integration or data agent changing, the agent based technology provides a convenient way.

6. Conclusions and Further Work

It is a challenge to store and share the sensed weather data from radar and remote sensor as the collected data is usually from the distributed environment and with huge amount. This work proposed an agent-based distributed environment where weather data can be queried and loaded by using agent-grid services. Future works will emphasis on follows:

1. Extend current framework; it will go down to province level; more nodes will be added.
2. Some grid middleware will be adopted to provide such as metadata synchronizing, dataFTP, transfer security and so on.
3. As data is valuable, economic model for utilizing weather data should be built, and competitive rules for off/ demand data should be inducted [14].
4. Consider the weather data's replica mechanism, the huge data can be transferred from nearest place, there need a mechanism for data source choosing.
5. Improve database architecture to store different data formats, especially data prepared for data mining.

Acknowledgement

Authors thank the anonymous reviewers for their very useful comments that helped us a lot to enrich the quality of the paper.

This work was supported in part by the NSFC (61173143, 60804047), and also supported by the Special Public Sector Research Program of China (GYHY201206030), and also supported by A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

[1] Allcock B., Bester J., Bresnahan J., Chervenak A., Foster I., Kesseman C., Meder S., Nefedova V., Quesnel D., and Tuecke S., "Data Management and Transfer in High-Performance Computational Grid Environments," *Journal of Parallel Computing Intensive Algorithms and Applications*, vol. 28, no. 5, pp. 749-771, 2002.

[2] Allcock B., Foster I., Chervenak A., Deelman E., Kesselman C., Nefedova V., Lee J., Sim A., Shoshani A., Williams B., and Middleton

D., "High-Performance Remote Access to Climate Simulation Data: a Challenge Problem for Data Grid Technologies," *Journal of Parallel Computing,* vol. 29, no.10, pp.1355-1356, 2003.

[3] Chervenak A., Deelman E., Kesselman C., Singh G., Bharathi S., Manohar M., Patil S., and Pearlman L., "A Metadata Catalog Service for Data Intensive Applications," *Technical Report,* Information Sciences Institute, 2003.

[4] 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 Data Sets," *Journal of Network and Computer Applications*, vol. 23, no. 1, pp. 187-200, 1999.

[5] Sokolova M. and Fernandez A., "Modeling and Implementing an Agent-Based Environmental Health Impact Decision Support System," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2603-2614, 2009.

[6] Foster I., Alpert E., Chervenak A., Drach B., Kesselman C., Nefedova V., Middleton D., Shoshani A., Sim A., and Williams D., "Earth System Grid II (ESG): Turning Climate Model Datasets Into Community Resources," *in Proceedings of the American Meterologcal Society Conference*, USA, pp. 1335-1356, 2003.

[7] Foster I., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 200-222, 2001.

[8] FIPA, "FIPA Agent Management Specification, Document Number: XC00023H, Foundation for Intelligent Physical Agents," available at: http://www.fipa.org/, Geneva, Switzerland, 2000.

[9] Jennings N. and Wooldridge M., *Agent Technology: Foundations, Applications, and Markets*, Springer-Verlag, 1998.

[10] JDOM, JDOMv1.1 API Specification, available at: http://www.jdom.org/docs/apidocs/, last visited 2007.

[11] Kruger A. and Krajewski W., "Efficient Storage of Weather Radar Data," *Journal of Software Practice and Experience*, vol. 27, no. 6, pp. 623-635, 1997.

[12] Li C. and Li L., "Agent Framework to Support the Computational Grid," *The Journal of Systems and Software*, vol. 70, no. 1-2, pp. 177-187, 2004.

[13] Miller M. and Smolarkiewicz K., "Predicting Weather, Climate and Extreme Events," *Journal of Computational Physic*, vol. 227, no. 7, pp. 3429-3430, 2008.

[14] Stefano A. and Santoro C., "An Economic Model for Resource Management in a Grid-Based Content Distribution Network," *Future Generation Computer Systems*, vol. 24, no. 3, pp. 202-212, 2008.

[15] Shoshani A., Sim A., and Gu J., "Storage Resource Managers: Essential Components for the Grid," *in Proceedings of Resource Management for Grid Computing*, USA, pp. 329-348, 2003.

[16] Shi Z., *Intelligent Agent and Applications*, Science press, 2000.

[17] Huang H., Luo J., Lin F., and Zharg H., "Agent-Based Grid Computing," *Journal of Applied Mathematical Modelling*, vol. 30, no. 7, pp. 629-640, 2006.

[18] Thain D. and Livny M., *Building Reliable Clients and Services, The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 2004.

[19] Veltri P., Cannataro M., and Tradigo G., "Sharing Mass Spectrometry Data in a Grid-Based Distributed Proteomics Laboratory," *Journal of Information Processing and Management*, vol. 43, no. 3, pp. 577-591, 2007.

[20] Wolski R., Spring N., and Hayes J., "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Meta Computing," *Future Generation Computer Systems*, vol. 15, no. 5-6, pp. 757-768, 1999.

Tinghuai Ma is currently working as Professor in Nanjing University of information Science & Technology, P.R. China. In 2009, he worked as visiting professor in Kyung Hee University, Korea. In 2005, he was appointed as Associate Professor at the Nanjing University of Information & Science Technology, China. He received his B.Sc. and M.Sc. in 1997 and 2000 at Huazhong University of Science & Technology, Wuhan, China. He obtained Ph.D. from Chinese Academy of Sciences, China in 2003 in Computer Software & Theory. His current research interests are Software Engineering, Data Mining, Ubiquitous Computing, Cloud Computing, and Privacy. He is a member of the IEEE.

Hao Cao received his Bachelor degree in computer science and engineering from Nanjing University of Information Science & Technology, China in 2010. Currently, he is a candidate for the degree of Master of Meteorological Information Technology & Security in Nanjing University of Information Science & Technology. His research interests include grid computing, cloudy computing.

Donghai Guan received his B.S. from Harbin Engineering University, Harbin, China. He got his M.S. degree in computer science from Kumoh National Institute of Technology (KIT), Gumi, South Korea in 2004. He got his Ph.D. degree in computer science from Kyung Hee University, South Korea in 2009. From 2009, he was a Post Doctoral Fellow at Computer Science Department, Kyung Hee University. His research interests are machine learning, data mining, activity recognition, and trust management.

SungYoung Lee received his B.S. from Korea University, Seoul, South Korea. He got his M.S. and Ph.D. degrees in computer science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA in 1987 and 1991, respectively. He has been a professor in the Department of Computer Engineering, Kyung Hee University, South Korea since 1993. He is a founding director of the Ubiquitous Computing Laboratory, and has been affiliated with a director of Neo Medical ubiquitous- Life Care Information Technology Research Center, Kyung Hee University since 2006. Before joining Kyung Hee University, he was an assistant professor in the Department of Computer Science, Governors State University, Illinois, USA from 1992 to 1993. His current research focuses on ubiquitous computing and applications, context-aware middleware, sensor operating systems, real-time systems, and embedded systems. He is a member of the ACM and IEEE.