Published in IET Software
Received on 23rd March 2009
Revised on 13th April 2010
doi: 10.1049/iet-sen.2009.0027

ISSN 1751-8806

Weather data sharing system: an agent-based distributed data management

T.H. Ma[1] *W. Tian*[1] *B. Wang*[2] *D.H. Guan*[3] *S.Y. Lee*[3]
[1]*School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, People's Republic of China*
[2]*Computer Division, National Meteorological Information Center, Beijing, People's Republic of China*
[3]*College of Electronics & Information, Kyung Hee University, Suwon, Korea*
E-mail: thma@nuist.edu.cn

Abstract: Severe weather causes human disasters. The most useful way to decrease a national disaster is by building more atmospheric sensing equipments to monitor the climate change. The data produced by these sensing equipments are of a huge amount and play an important role for weather prediction. Moreover, new sensing equipments enrich weather data. Everyday terabyte and petabyte-scale data are collected. Retrieval of such information requires access to large volumes of data; thus an efficient organisation is necessary both to reduce access time and to allow for efficient knowledge extraction. A new class of 'data grid' infrastructure is efficient to support management, transportation, distributed access and analysis of these data sets by thousands of potential users. Intelligent agents can play an important role in helping achieve the 'data grid' vision. In this study, the authors present a multi-agent-based framework to implement manage, share and query weather data in a geographical distributed environment, named weather data sharing system (WDSS). In each node, some services are designed for querying and accessing data sets based on agent environment. Information retrieval can be conducted locally, by considering portions of weather data, or in a distributed scenario, by exploiting global metadata. The agents' local and remote search is evaluated. The transfer speeds for different file types are also evaluated. From the presented platform, the system extensibility is analysed. The authors believe that this will be a useful platform for research on WDSS in a national area.

1 Introduction

Nowadays, with the development of information technology, climate and meteorology, research can take a big step forward with the capability of collecting and using huge data. In China, there are 158 weather radars that collect climate information. Each radar can generate about 6 GB of data every 6 min and about 1 TB of data every day.

Hence, large volumes of data for meteorology prediction are available and will be used every day. Climate modelling is especially a challenging problem as extremely long-duration predictions are accompanied by frequent output of very large files that are needed to analyse the simulated climate [1, 2]. The simulations must be compared with what is known as an observed climate. New infrastructures are required to support the management, transportation, distributed access and analyse these data sets by thousands of potential users.

To understand complex climate processing, geographically distributed teams of researchers can effectively and rapidly retrieve new knowledge from these massive, distributed data holdings and share the results with a wider community. Fundamentally, new methodologies for managing, accessing, recombining, analysing and inter-comparing distributed data are required.

For the management of distributed data, 'Data Grid' is a perfect solution [3–5]. To satisfy the requirement of both China Meteorology Administrations operation and related researchers, the system software should have three characteristics: national collaboration, metadata and service-oriented approaches. The framework should have the following capabilities:

- containing detailed knowledge of its components and status;
- constructing system dynamically;
- seeking to optimise its behaviour to achieve its goals; and
- being aware of its environment.

However, there are some potential conflicts between requirements within this distributed data grid application. Agent technology is one of the ways to meet these requirements in the grid development [3, 6, 7].

First, there are huge amounts of nodes that are needed to manage their own data. A need for broad deployment implies that node systems must be simple and of minimal requirements on local sites. The ideal scenario is the system that can be deployed automatically.

Second, the need to achieve a wide variety of sensitive complex performance application implies that these systems must provide a range of potential sophisticated services. The system can be so complicated that its workflow and dataflow cannot be decided. Furthermore, the co-operations among modules in the system are not clear; using an agent's autonomous, a complex software system just like a society can be developed.

Third, the need for scalable and evoluable systems and services in the future. For a system, if its components are not known in advance, its structure is capable of dynamic changes. Agent can be added to a system at any time. Agent-composed systems may be highly heterogeneous.

Data grid can solve the requirement of distributed climate data managements. The agent-based system can satisfy the data grid functions. We can view the data grid as a number of agents interacting with each other based on knowledge. Developing intelligent agents to realise the grid vision has been made by academic researchers during the past few years [8]. The agent grid is a specific construct or mechanism within that layer for making services and resources available. Some agent-based grid computing (AGEGC) systems have been built [9].

The rest of the paper is organised 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 implementation and evaluation in Section 5, which focuses on data accessing process locally and remotely. The system extensibility is discussed in Section 6. Finally, we point out the future work and conclude the paper in Section 7.

2 Related works

Accurate weather prediction is important to decrease and avoid natural disaster. The national distributed meteorological information is the key issue for weather prediction. Recently, there has been an increasing interest for meteorological prediction by the cooperation between meteorological and computer scientists. Designing software systems to manage, utilise and distribute meteorological information is one of the most important works in climate prediction.

From 2001 onwards, Chinese Meteorology Administration (CMA) began to provide meteorological information publicly and openly. From 2001 and 2005 onwards, CMA has built a centralised sharing platform to provide historical meteorological information to meteorological scientists all over the world.

But meteorological data are geographically distributed nationwide. The centralised storage methods require high bandwidth, high-performance computing machines and huge disk space to transfer, manage and store data. Thus, distributed storage methods are adopted [10]. Agent software architecture can realise the distributed data management. Agent technologies have been welcomed in ecological and environmental applications mainly as a metaphor for decomposing complex systems and studying the emergence of collective behaviour [11]. We regard every distributed node as an agent. An agent provides service to access local data. System software maintains the service directories of all agents. These are the main concepts of agent-based distributed data management [5, 7]. Shi [12] developed a software system that provides specific mechanism to make services and resources available, remotely. Veltri *et al.* [13] utilised grid middleware, metadata publishing, resource advertisement and discovery to execute efficient, reliable and secured data transfer based on grid computing.

Agent-based techniques have been used for modelling several environmental fields, including socio-ecological systems [14], Earth System Grid [1, 15], integrated water resources management [16], residential power consumption [17], traffic data analysis [18], water trading [19], land-use change [20], environmental health [21] etc. After studying 23 environmental systems that utilise agent technology, Athanasiadis pointed out that Abacus radar data management [22, 23] is the most complete agent-based environmental system from conception to deployment.

But, for agent-based computing in meteorology applications, there are not many success stories. Mathieson *et al.* [24] have developed an agent-based forecast system for Australian Bureau of Meteorology. Hughes and Lewis

[25] developed a layered agent architecture to manage radar data. This was the first time a software agent approach was employed. A three-layered architecture-based agent radar data management, similar to [25], was developed in Switzerland [22]. To the authors knowledge, Dance and Potts [26] are the first to employ a software agent approach for managing the weather radar data.

From the above, it can be concluded that agent technology is widely used in environmental fields, but is limited in use in the meteorology system. From the existing project, the software agent has just been used for radar data managing. In China, for national widely distributed weather data sharing, there is no software agent technique. We are yet for the first time design and implement the weather data sharing system (WDSS) based on agent technology, adopting the data agent to represent each data set, considering the data transferring and network weather prediction service.

3 Weather data

To manage its weather information, China has been divided into eight region areas. There are four aspects of information in each centre:

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 a variety of numerical models 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 a special theme.

We need to design a framework working in an agent-based environment, which is composed of several nodes. Each node associated to a region centre database server (called 'node' from now on) run its local database. The framework should efficiently support the 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 [27].

There are three types of data as follows:

- *Raw files:* Original data produced by the observation equipments, remote sensing equipments, lattice information, satellite images and radar images.

- *Structured data in a database:* Structured data are stored in a relational database. An Oracle database instance is used to store the data from automated observation station, whose origin data are stored as a raw file; some statistical data of the raw file; some production data from raw file; and characteristic production from raw radar images.

- *XML instances:* 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 personalised access to such data. Each node may define a personalised view allowing remote clients to query local data.

4 System architecture

- WDSS requires the collection and analysis of data produced in national distributed observation stations. The collection, storage and analysis of huge weather data can leverage the computational power of a distributed environment, using these data to make weather prediction. We choose a grid-based architecture that offers efficient data transfer, effective management of large data storage, the computational power required by huge data manipulation, and the security and privacy required for data belonging to different organisations. The management of data is the principal operation of the WDSS [1, 5]. The basic function of resource management is to accept requests for resources within the grid, and assign specific node resources from the overall pool of grid for which the user has 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, and hence service management is also important. WDSS is now mainly concerned with resource management and service management. The services on each node should include data exploration, data querying, data transforming and data delivering service. Nevertheless, users never have direct access to remote data sources. They just retrieve data through a query service issued to the data sources of each node. The architecture of the proposed system is composed of a set of agent-based grid nodes, and each node runs:

- An agent management service (AMS), providing an interface on weather data (stored in relational, XML and flat file formats) for data retrieval.

- A metadata catalogue service [28], providing yellow pages service to other agents. Agents may register their services with the metadata catalogue service or query the metadata catalogue 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.

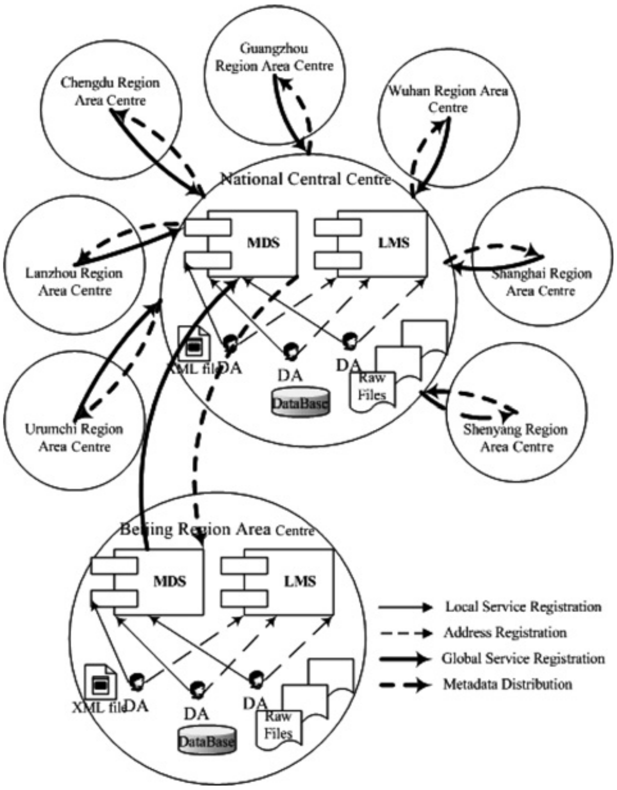


Figure 1 *Data management architecture in WDSS, which has one central and 8 region centre points*

In China Meteorology Administration, the National Meteorological Information Centre acts as a central node which coordinates with eight regional centre nodes. Fig. 1 demonstrates the resource management architecture of WDSS.

The proposed framework is the best solution according to the requirements of China Meteorology Administration's operation. First, the metadata that describe the meteorological data detail can be shared among each node, including the national central centre (NCC). Therefore, the data detailed knowledge and status are stored nationally. Second, every node is an autonomic component. The node implements data accessing and locating itself. It joins the data share platform just by sending its metadata to the NCC. The added or deleted node is dynamical and does not affect any other nodes' operation. Third, the distributed data storage is secure. The meteorological data are stored in the NCC in the former platform. The shared platform will not work when the NCC node goes down. The platform we provide can tolerate such flaws should any node breakdown. Few node crash downs will affect the platform less. Fourth, the distributed data storage will decrease the bandwidth requirement. In the former solution, every node will transfer various amounts of meteorological data to the NCC. Then, the NCC transfers the data to some nodes according to requirements. Now, the data are transferred directly between nodes. There is no need to transfer data to the NCC. Because it can reduce half of data transfer compared with the former.

4.1 Overview of WDSS

WDSS is composed of eight regional local nodes and one national central node. Every node has one sharing graphic user interface (GUI) and provides sharing data for different users. The components in one region area centre consist of a local multi-agent system (MAS). As shown in Fig. 1, the components of each local MAS mainly include DA (data resource agent), LMS (local management system) and MDS (metadata system).

'DA' is the representative of a hybrid data resource. The current data types are flat file, relational database and XML files. These data resources can be modelled and provided as service. Each DA should register itself as service in the metadata system. Furthermore, the high-performance computing power can also be encapsulated as service in times to come. At that time, DA will be renamed as resource agent (RA).

'LMS' is the manager of the local MAS and offers white page services (mapping from agent name to agent address) to other agents. It provides the interface for users to retrieve information from WDSS. The interface must be web-based because the user may be in or out of CMA. Through the unified interface, users can retrieve locally or across the whole grid.

'MDS' provides yellow page services to other agents, especially for MAS users [28]. DAs may register their services linked to data source with the MDS or query the MDS to find out what services are offered by other agents. Local MDSs register managed service to the NCC MDS. MDS in the NCC distributes the integrated service's registration over the whole grid system. The service registration information we call metadata includes data type, generated time, generated place, store place, services, invoking methods, etc. All these information is transferred in XML format.

In our data management architecture, LMSs are organised in the hierarchical manner. An LMS can register 'its service' to another LMS. Here, the service of an LMS means that all the services provided by the DAs are registered in it. Its registration with other LMSs is achieved through interaction between their respective metadata files. For example, if LMS A will register its service with LMS B, then MDS in LMS A will send registration information (XML file) to MDS in the NCC. MDS in the NCC will broadcast the whole integrated registration information to every regional centre, which includes LMS B. Registration information contains the main services that LMS A can provide.

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

4.2 Local Management System

An LMS is a collection of tools supporting the entire function of data retrieval. Within it, AMS is a mandatory component to exert supervisory control over access to agent services. Through GUI, users query and download data, the administrator adds new data resource to the LMS. While a new data resource is added, the agent linked with these data registers its service to AMS, and AMS will assign its agent identifiers (AIDs). Each agent must register with an AMS in order to obtain a valid AID. The AMS maintains a directory of AIDs. There are some assistant tools in an LMS, such as network weather prediction service; data organise service; data FTP service; and message transport service. Fig. 2 is the logical architecture of an LMS.

'GUI service' allows the user to retrieve weather data from remote nodes and to perform services. It provides an interface for querying data and supports retrieval functions based on meta information such as generated date, equipment type, generated place and so on. Of course, an LMS offers an application programming interface (API) to query and retrieve 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 Fig. 3, the main GUI for users describe their requirements.

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

'AMS' is a mandatory component in MAS. It acts as a control role in the system. Only one AMS exists in the MAS. The AMS maintains a directory of AIDs, which contains transport addresses (among other things) for agents registered in the MAS. The AMS offers white page services to other agents. Each agent must register with an AMS in order to obtain a valid AID. AMS manages the agent's life cycle, responsible for creating and destroying of agents. AMS is also responsible for executing the application and returning the results.

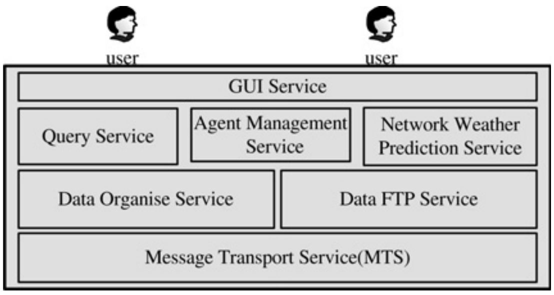


Figure 2 *Logical architecture of local management system*

Input the Location based station ID

Z9010;Z9210;Z9220;Z9451;Z9452;Z9453;Z9454;Z9456;
Z9457;Z9454;Z9431;Z9433;Z9436;Z9439;Z9240;Z9411;Z9415;Z9417;Z
Z9471;Z9476;Z9477

Account

Password

Data Type Selection

☐ AutoStation data ☐ Radar data ☐ SolarRadiation
☐ AgriData ☐ Sate Pic ☐ OceanWeather
☐ HighAltitudeData ☐ Disaster ☐ AtmosphereComposition

Figure 3 *Data source selection GUI*

From here, users can decide the data type, time period and location

'Network weather service (NWS)' [29] 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 have replica in other places. Now, the region area centre and national information centre store the same data simultaneously. A request for data has two choices from different places (origin region area centre and national information centre). The important aspect is the network performance.

'Data organise 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 can control the data flow; he/she can choose where to obtain data from and where to save results. On the platform, administrator can edit metadata for data resources, load data into storage and save the metadata.

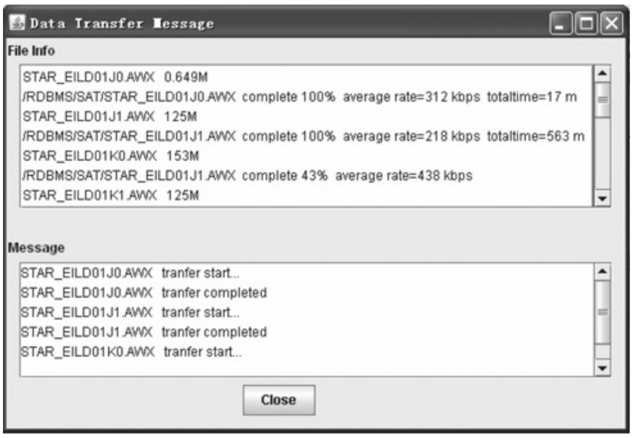
'Data FTP service' provides efficient, reliable and secure data transfer in grid computing environments. It can be a simple prototype of Grid FTP [30]. For any data grid solution, data accessing and data transferring must be critical components. An LMS must move the file securely over the wide-area network to its destination. FTP is commonly used for data transferred over the Internet and the most likely candidate for meeting the Data Grid's needs. The data transfer monitoring information is shown in Fig. 4.

'Message transport service' is the default communication method between agents. It is based on FIPA Agent Management Specification [31], and there are some products to choose.

Among the services in LMS, the data organisation service is timed to start or triggered by the change of data resources. It is run first to prepare the data resources and register its information to agent management service, as shown in Fig. 5. This is the pre-process of data sharing.

The LMS supported data exploration is described as follows:

1. User goes through the GUI for data searching, and sends data requirement.



```
STAR_EILD01J0.AWX  0.649M
/RDBMS/SAT/STAR_EILD01J0.AWX  complete 100%  average rate=312 kbps  totaltime=17 m
STAR_EILD01J1.AWX  125M
/RDBMS/SAT/STAR_EILD01J1.AWX  complete 100%  average rate=218 kbps  totaltime=563 m
STAR_EILD01K0.AWX  153M
/RDBMS/SAT/STAR_EILD01J1.AWX  complete 43%  average rate=438 kbps
STAR_EILD01K1.AWX  125M
```

```
STAR_EILD01J0.AWX  tranfer start...
STAR_EILD01J0.AWX  tranfer completed
STAR_EILD01J1.AWX  tranfer start...
STAR_EILD01J1.AWX  tranfer completed
STAR_EILD01K0.AWX  tranfer start...
```

Figure 4 *Data transfer monitoring information*

The file in the waited queue can be found in the 'file info box' and the file transferring indicated in 'Message box'

2. Data requirements are sent to query service. Query service lists the requirement in queue and schedules the queue.

3. The query queue is sent to an AMS. Using the white pages, AMS locates the data AID and address. AMS creates an agent to obtain the data resource.

4. AMS inquires NWS, finds out the best way of performance and optimises the transfer performance.

5. AMS notifies the data FTP service to transfer the data resource to user's destination.

AMS plays the role as the core, it is surrounded by query service, NWS, data FTP service and data organise service. These messages among the services are transferred by message transportation service. Message transportation service acts as a message transfer bus.

4.3 Data agent

A DA is a representative of hybrid data resources. These data resources can be accessed through services provided by its agent. These data have their mapped agents and services. One data agent can have several services to access the same data resource. Each agent holds its AID created by AMS, and registers its metadata information into MDS. As an agent, the data agent has common characters [9]; its structure is shown in Fig. 6.

Agent architecture should satisfy the flexibility and capability of software. Capability means that the software module is powerful enough to handle things such as co-operation, negotiation, mobility and so on. We hope that the agent can have better functions. But the complexity of the agent's structure will embarrass the agent's deployment.

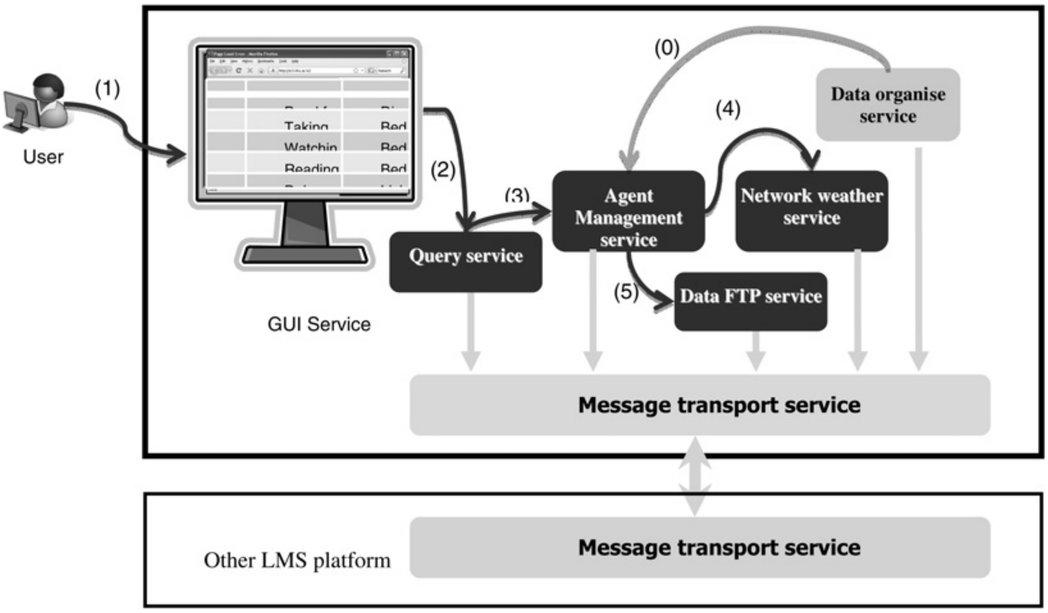


Figure 5 *Interaction of services in LMS*

doi: 10.1049/iet-sen.2009.0027

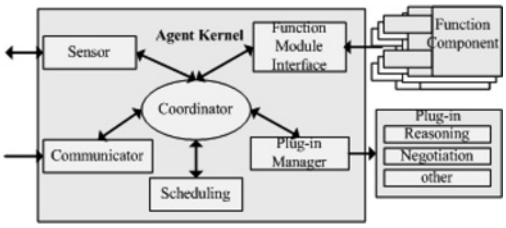


Figure 6 *Architecture of the generic agent*

This will decrease the agent's expansibility. For the system's flexibility, an agent is simple enough to deploy, and can be customised by users with their own defined components. This support the systems structure of dynamic change. So, agent structure definition must make a trade-off between flexibility and capability.

Generally, an agent has a kernel and some plug-in like components [9, 12]. Kernel components are the solution of flexibility and plug-in like components are the solution of capability. An agent kernel consists of the following parts. The sensor perceives through the outside world. The function module interface makes an effect to the outside world. Communicators handle communication between their agents. The coordinator makes decision concerning the agent's goals, and it is also responsible for co-coordinating the agents interaction with other agents using the given co-ordination protocols and strategies like the contract net protocol or different kinds of auction protocol. Schedulers plan the agent tasks based on the decision taken by the co-ordination engine, the resources and task specifications available to the agent. Resource ontology maintains the lists of ontology describing the semantic content of resources that are owned by the agent. Task ontology provides logical description of tasks known to the agent. Plug-in manager manages the components provided by users that can be plugged into the agent kernel.

5 Implementation and evaluation

5.1 Data retrieval process

Data retrieval can be executed anywhere through Internet. Each region area centre provides the user the web page for data querying and retrieving, and also the APIs for application programs. These functions are realised by an LMS, which has two-fold goals: (I) retrieve efficiently portions of weather data set for local weather product producing (e.g. making local weather prediction) and (II) retrieve weather data sets shared on the grid through XML querying. Some examples concerning these two types of application are briefly described in the following. The users can use query service to perform 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 [32] on XML Schema, and SQL like queries on relational Oracle data. This is an appropriate query translation module.

Retrieving data from WDSS is divided into three steps: (i) invoke the query service to input the query condition; (ii) query service searches the metadata information, matching the right DA whose data can satisfy the query; (iii) AMS invokes the data-matched DA to load data and transfer it to the users. The detailed processes are shown in Figs. 7 and 8 according to the data residing locally or on the grid separately. For local enquiry, all operations are executed in one LMS, which is physically mapped to one node. As shown in Fig. 7, step 1 invokes the query service, steps 2–5 search metadata information, and steps 6–7 represent data loading and transferring. The metadata information query is in the local LMS.

As shown in Fig. 8, step 1 invokes the query service, steps 2–6 search metadata information, and steps 7–8 represent data loading and transferring. The difference between Figs. 7 and 8 is that there are two query processes in Fig. 8. One is step 2, query in local LMS. The other is step 3, query metadata information in MDS of the NCC, because it cannot find right metadata information in local MDS. This means that query data are stored in remotely.

Retrieving data from the grid, query service in LMS asks the MDS whether or not it is satisfied with the operators' requirement. If not, it continues to ask the NCC's MDS. The NCC's MDS return the matched metadata to the operators for confirmation. After the users have finished

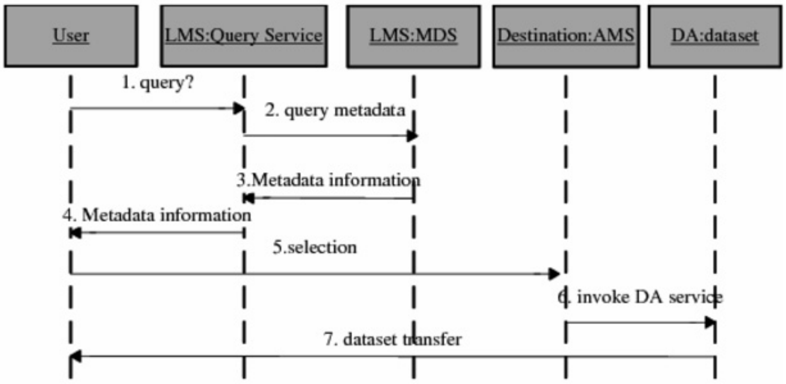


Figure 7 *Querying process in WDSS locally*

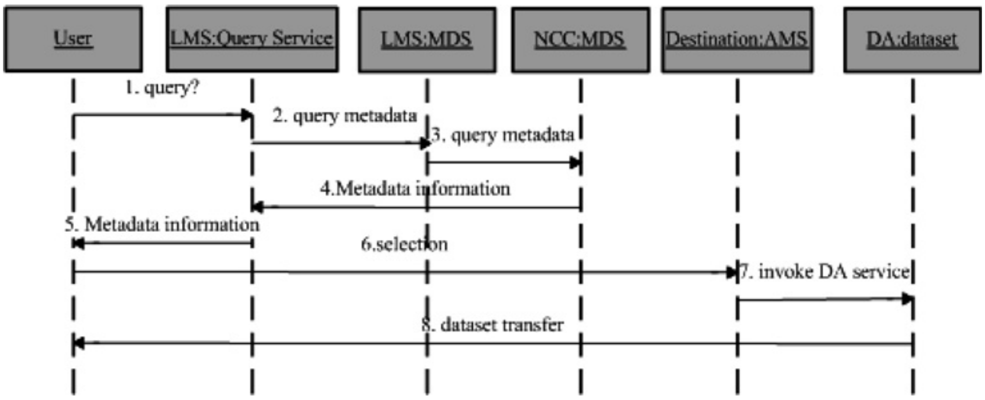


Figure 8 *Querying process in WDSS globally*

choosing DAs metadata, the destination nodes AMS get the DAs AID, which is obtained from the metadata. Then the AMS invokes the DA, and the DA returns the result to the operators.

5.2 Evaluation

The WDSS system has been implemented as a prototype, which includes two centres; these are NCC and Beijing regional centre. The system in the NCC consists of 2 Intel 2.10 GHz *4 nodes with 16 GB of RAM memory, connected by a Gigabit network. Its output bandwidth is 155 Mbps with synchronous digital hierarchy (SDH). The hard disks in each node provide approximately 50 MB/s. In the regional centre, it consists of 2 Intel 1.65 GHz *4 nodes with 16 GB of RAM memory connected by a Gigabit network. Its output bandwidth is 8 Mbps. The hard disks in each node provide approximately 30 MB/s. To prevent the client from being the throughput bottleneck of this system, a high-performance client is required. An Intel Dual-Core 2.10 GHz system featuring two hard disks set in a RAID 0 array is selected as the client.

There are two main aspects that describe the system's efficiency, one is the search time and the other is the data transfer speed.

For search time, we devise tests with local search and global search. The experiment is designed to evaluate two factors, which affect the search times: the type of agents and the number of agents.

Table 1 shows that the time required to locate a data agent is slowly increasing with the number of agents registered becoming larger in local search. Time is also increasing proportionally in remote search. There are two factors that are accountable for the increase in time needed to locate data agents. One is the increased processing time of query service, and the other is the increased network delay used to return responses from remote nodes. Table 1 shows the data agent number below 10 the search time increases with the number growing. While the data agent number increases to 10, the search time does not increase sharply. There are just some little vibrations of search time while the data agent number is between 10 and 70. Searching data agent information means finding right data agent description in metadata. Metadata increase with increasing agent numbers. So, while agent numbers are between 1 and 10, the searching time increases with increasing metadata file. While agent numbers are greater than 10, finding the right agent description which randomly located in metadata, the searching time may be randomly. It displays as vibration.

For evaluation of data transfer, we devise the data resource in the regional centre. So, the maximum speed is 8 Mbps. We devise the test according to the raw file, image file and radar volume scanned data file, the size of which are about 128 k, 256 k and 15 M, respectively.

Fig. 9 shows the results obtained with these three configurations varying parameters that have a clear influence in the performance of I/O operations:

For the same data size, such as 100 MB, the file number is about 800 (100 MB/128 k) for raw files, 400 (100 MB/256 k) for image files and 7 (100 MB/15 M) for radar volume scanned files. Although the data size is the same 100 M, at the same transfer rate, the file number is different, and the transfer time are different. Besides, as it

Table 1 Search times against type of agents and number of agents registered

Number of agents registered		1	5	10	20	30	40	50	60	70
search time (ms)	local	8.9	12.2	19.2	18.2	15.8	21.2	14.8	19.9	21.7
	remote	26.8	30.1	37.1	36.1	33.7	39.1	32.7	37.8	39.6

doi: 10.1049/iet-sen.2009.0027

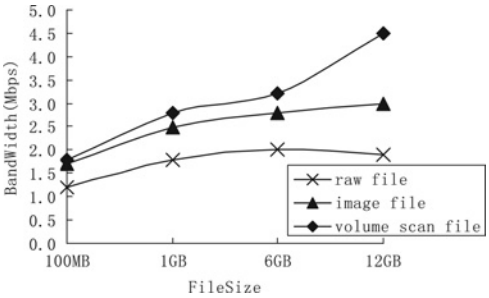


Figure 9 *Evaluation of the agents with different file type*

costs more time to access files for the large file number, the performance of less file number is better. As Fig. 9 shows, the transfer time increases sharply under the large file number (while there are 12 G data in raw file type).

Furthermore, if a big size file which is a multiple of the 64 kB, may lead to an efficient transfer. The Grid FTP is adopted in our system to transfer massive amounts of data. The maximum data size that DMA drivers can manage is 64 kB for one thread. Since the thread number is an integer, a file size that is a multiple of 64 kB can be transferred by integer parallel threads. A file size that is a multiple of 64 kB can induce the better performance.

6 System extensibility

One of the motivations for the use of agents is to create an open and extensible data sharing system, minimising the impacts of the addition of new data sources, analysis tools and changes to the employed legacy systems.

Now, there are eight area centres and one national centre in our framework. With the system developing, the centre will be cited on province or city level, and at that time there will be many centres that need to be added into the presented framework. The new added centre may have newer software, while the existing centre holds older software. The result is the two versions of the centre's software running in parallel for several months. The weather data stored in two separate archive databases, which run on separate servers.

For centre integration, in each centre, the LMS is responsible for data retrieval. The newer centre may develop the LMS on a different and newer agent platform. While the new centre is integrated into the whole system, only the MDS should coordinate the metadata file while it being changed, transferred, and communicated with other centres. The new centre is strongly autonomic, and it is easy to integrate a new centre to the whole system.

For the data agent level, as our platform is a MAS, adding and removing agents for a MAS is trivial and can be achieved while other agents continue to operate. Adding new agents to the MAS is relatively simple, but if data types change or new types of data are made available, then these changes must be reflected in the metadata file. The corresponding LMS software is responsible for retrieving changed or new added data. The changed metadata file will be passed from the local centre to the NCC centre, and broadcasted to every regional centre.

As the metadata file represents the data, only the LMS is required for modification to cope with the new terms. The best way for the query service of the LMS is to provide a uniform way to interpret the metadata file. Once the new data are added or changed, only the metadata XML file needs to be changed. The uniform parser is responsible for such metadata XML file modification.

7 Conclusions and further work

We considered the problem of treating huge amounts of weather data in a distributed environment. With the radar and remote sensing data collection, storage and sharing represent an extrude problem. We proposed an agent-based distributed environment where weather data can be queried, loaded by using agent-grid services according to the requirements.

The novelty of our proposal is the use of a multi-agent subsystem in a meteorological data sharing system, which is used by a grid framework for providing access to data resources. The details of the proposed architecture can be described as follows: (1) Scalability, the new centre can be easily added or deleted; (2) Efficiency, the weather data do not have to be transferred to the NCC. The data transfer can be made between regional centres; (3) Security, the distributed data storage can avoid data disaster. A remote backup can be formed between regional centres.

The limits of the approach are authorisation and data resource's change. For authorisation, the presented distributed grid-based platform users' identity will be authenticated at local nodes. Although the users have rights to access the local nodes, this does not mean that they have rights to access the remote data resource. That is to say, the users' identity may be authenticated twice or more times. A union authentication is needed in platform. For data resource change, as we know, the data amount increases with time. The agent that represents the data resource is also changed. So, when we modify the agent, changing the metadata is difficult. In the present work, new data are added, and a new agent will be generated to connect with data and update the metadata. To induce the data locating will be a hard work because there are many agents with several years.

When we deployed our platform national around, we encounter a dilemma. There exists a conflict between data sharing and data protection. Some local weather bureaus cost a lot of money to be equipped with advanced facilities. The weather data they hold is more abundant than that of others, because the data have different values as they come from different local bureaus. The enriched data bureau will not share data with others because their data value is higher. So, authoritative and forceful command is the guarantee for data sharing around national area.

In future works we emphasise on the following: (1) Extend the current framework; it will go down to the province level; more nodes will be added. (2) Some grid middleware will be adopted to provide data such as metadata synchronising, data FTP, transfer security and so on. (3) As data are valuable, the economics model for utilising weather data should be built, and competitive rules for off/demand data should be inducted [33]. (4) Consider the weather data's replica mechanism; the huge data can be transferred from nearest place; this need a mechanism for data source choosing. (5) Improve database architecture to store different data formats, especially related to prepared data for data mining.

8 Acknowledgment

The authors are grateful to Wenhai Shen for discussions on weather data grid. The authors thank the anonymous reviewers for their very useful comments that helped them a lot to enrich the quality of the paper. This research was supported by Natural Science Fund for colleges and universities in Jiangsu Province (08KJD520018), Natural Science Foundation from Nanjing University of Information & Science Technology (20080302), Jiangsu overseas study scholarship and Jiangsu Youth Project.

9 References

[1] ALLCOCK W., FOSTER I., NEFEDOVA V., *ET AL.*: 'High-performance remote access to climate simulation data: a challenge problem for data grid technologies'. SC'2001, 2001

[2] MILLER M.J., SMOLARKIEWICZ P.K.: 'Predicting weather, climate and extreme events', *J. Comput. Phys.*, 2008, **227**, pp. 3429–3430

[3] CHERVENAK A., FOSTER I., KESSELMAN C., *ET AL.*: 'The data grid: towards an architecture for the distributed management and analysis of large scientific datasets', *J. Netw. Comput. Appl.*, 2000, **23**, pp. 187–200

[4] CHUNLIN L., LAYUAN L.: 'Agent framework to support the computational grid', *J. Syst. Softw.*, 2004, **70**, pp. 177–187

[5] CHERVENAK A., FOSTER I., KESSELMAN C., SALISBURY C., TUECKE S.: 'The data grid: towards an architecture for the distributed management and analysis of large scientific data sets', *J. Netw. Comput. Appl.*, 2001, **23**, (3), pp. 187–200

[6] JENNINGS N.R., WOOLDRIDGE M.J. (EDS.): 'Agent technology: foundations, applications, and markets' (Springer-Verlag, 1998)

[7] THAIN D., LIVNY M.: 'Building reliable clients and services', in FOSTER I., KESSELMAN C. (EDS.): 'The grid: blueprint for a new computing infrastructure' (Morgan Kaufmann, 2004, 2nd edn.)

[8] Workshop on Agent based Cluster and Grid Computing, 2001. Available from: http://www.cs.cf.ac.uk/User/O.F.Rana/agent-grid/

[9] SHI Z., HE H., LUO J., *ET AL.*: 'Agent-based grid computing', *Appl. Math. Model.*, 2006, **30**, pp. 629–640

[10] SHOSHANI A., SIM A., GU J.: 'Storage resource managers: essential components for the grid', in NABRZYSKI J., SCHOPF J., WEGLARZ J. (EDS.): 'Grid resource management: state of the art and future trends' (Kluwer Academic Publishers, 2003)

[11] OLSON R.L., SEQUEIRA R.A.: 'An emergent computational approach to the study of ecosystem dynamics', *Ecol. Model.*, 1995, **79**, pp. 95–120

[12] SHI Z.: 'Intelligent agent and applications' (Science Press, China, 2000)

[13] VELTRI P., CANNATARO M., TRADIGO G.: 'Sharing mass spectrometry data in a grid-based distributed proteomics laboratory', *Inf. Process. Manage.*, 2007, **43**, pp. 577–591

[14] BATTEN D.: 'Are some human ecosystems self-defeating?', *Environ. Model. Softw.*, 2007, **22**, pp. 649–655

[15] CHERVENAK A., FOSTER I., DEELMAN E., *ET AL.*: 'High-performance remote access to climate simulation data: a challenge problem for data grid technologies', *Parallel Comput.*, 2003, **29**, pp. 1335–1356

[16] BARTHEL R., JANISCH S., SCHWARZ N., *ET AL.*: 'An integrated modelling framework for simulating regional-scale actor responses to global change in the water domain', *Environ. Model. Softw.*, 2008, **23**, (9), pp. 1095–1121

[17] GU Y., BOUKERCHE A., ARAUJO R.B.: 'Performance analysis of an adaptive dynamic grid-based approach to data distribution management', *J. Parallel Distrib. Comput.*, 2008, **68**, pp. 536–547

[18] XU M., HU Z., WU J., ZHOU Y.: 'A hybrid society model for simulating residential electricity consumption', *Int. J. Electr. Power Energy Syst.*, 2008, **30**, (10), pp. 569–574

[19] SMAJGL A., HECKBERT S., WARD J., STRATON A.: 'Simulating impacts of water trading in an institutional perspective', *Environ. Model. Softw.*, 2009, **24**, (2), pp. 191–201

[20] BITHELL M., BRASINGTON J.: 'Coupling agent-based models of subsistence farming with individual-based forest

doi: 10.1049/iet-sen.2009.0027

models and dynamic models of water distribution', *Environ. Model. Softw.*, 2009, **24**, (2), pp. 173–190

[21] SOKOLOVA M.V., FERNA'NDEZ-CABALLERO A.: 'Modeling and implementing an agent-based environmental health impact decision support system', *Expert Syst. Appl.*, 2009, **36**, (2), pp. 2603–2614

[22] ATHANASIADISA I.N., MILISB M., MITKASC P.A., MICHAELIDESD S.C.: 'A multi-agent system for meteorological radar data management and decision support', *Environ. Model. Softw.*, 2009, **24**, (11), pp. 1264–1273

[23] ATHANASIADIS I.N., MITKAS P.A.: 'A methodology for developing environmental information systems with software agents, in advanced agent-based environmental management systems' (Springer Press, New York, 2009)

[24] MATHIESON I., DANCE S., PADGHAM L., GORMAN M., WINIKOFF M.: 'An open meteorological alerting system: issues and solutions'. Proc. 27th Australasian Conf. on Computer Science, ACSC'04, Australian Computer Society, Inc., Darlinghurst, Australia, 2004, pp. 351–358

[25] HUGHES E., LEWIS M.: 'Intelligent agents for radar systems', *Electron. Syst. Softw.*, 2005, **3**, (1), pp. 39–43

[26] DANCE S., POTTS R.: 'Microburst detection using agent networks', *J. Atmos. Oceanic Technol.*, 2002, **19**, (5), pp. 646–653

[27] KRUGER A., KRAJEWSKI W.F.: 'Efficient storage of weather radar data', *Softw. Pract. Exper.*, 1997, **27**, (6), pp. 623–635

[28] CHERVENAK A., DEELMAN E., KESSELMAN C., PEARLMAN L., SINGH G.: 'A metadata catalog service for data intensive applications'. Technical report, Information Sciences Institute, 2003

[29] WOLSKI R., SPRING N., HAYES J.: 'The network weather service: a distributed resource performance forecasting service for meta computing', *Future Gener. Comput. Syst.*, 1999, **15**, pp. 757–768

[30] ALLCOCK W., BESTER J., *ET AL.*: 'Data management and transfer in high-performance computational grid environments', *Parallel Comput.*, 2002, **28**, pp. 749–771

[31] FIPA: 'FIPA agent management specification'. Document number: XC00023H, Foundation for Intelligent Physical Agents, Geneva, Switzerland, 2000. Available from: http://www.fipa.org/

[32] JDOM v1.1 API Specification, 2007, available from: http://www.jdom.org/docs/apidocs/

[33] DI STEFANO A., SANTORO C.: 'An economic model for resource management in a Grid-based content distribution network', *Future Gener. Comput. Syst.*, 2008, **24**, pp. 202–212