SIG-DDS: a Grid-based Remote Sensing Data Distribution System

Huang ZhenChun †, Li GuoQing ‡
†*Department of Computer Science and Engineering, Tsinghua University, Beijing 100084*
‡*China Remote Sensing Satellite Ground Station, Beijing 100086*
huangzc@tsinghua.edu.cn, ggli@ne.rsgs.ac.cn

Data distribution is one of the most important steps for the remote sensing applications; it depends on the Remote Sensing Data Distribution System (RS-DDS). The conventional RS-DDS such as GCMD Mapserver [1] or CEOP [2] often provides a web based user interface and supports little data types only. Although the WCS [3]/WFS [4]/WMS [5] purposed by OGC try to make it possible for applications to access heterogeneous data through a universal interface, it is still hard to be used because of the absent of data searching and difficulty of application programming. The distributed integrated applications are eager for a distributed, extensible, and co-operable RS-DDS which can distribute different types, different formats, and different goals RS-data in one system.

Spatial Information Grid (shortly SIG) is a kind of application grid; it is built on the general grid and provides a perfect infrastructure for distributed spatial information applications. Based on SIG, a 3-layered remote sensing data distribution system named SIG-DDS is designed and implemented in this paper. It can be deployed in many java-based dynamic web servers such as Apache Tomcat which passes the test.

1. Architecture and Design

The SIG-DDS is built up by the RS-data source, RS-data broker, RS-data distribution client library, user interface, resource registry meta-service [6] (or UDDI registry center [7]), etc. The components can be deployed on different nodes in grid. The SIG-DDS takes a 3-layered architecture shown as Figure 1.

The foundation of three layers is the data source layer, where the RS-data sources are. In the SIG-DDS, data source is defined as a logical service which can provide operations including RS-data query and RS-data access for the data distribution system no matter with the type, format, organization, and feature of the stored data. The registry meta-service or UDDI center is also a part of this layer. The RS-data sources must be registered in it.

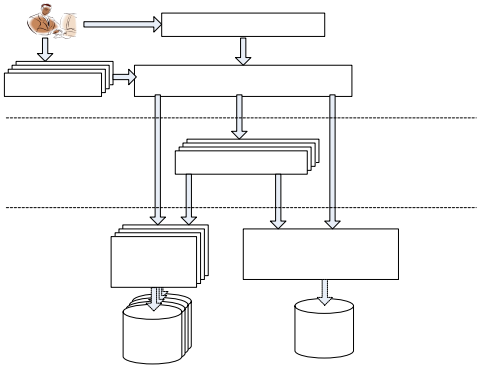

Fig.1. the 3-layered Architecture

The next layer is the data broker layer which includes a mass of RS-data brokers. A data broker collects information of stored data from registered data sources, re-organizes and optimizes it so that the data distribution client can find and download all types and formats of distributed RS-data easier and more efficiently. It adopts the same interface as RS-data sources to simplify the system design.

As the top layer, there are the data distribution client library, applications and user interfaces in the user/application layer. The client library provides a simple and powerful programming interface for remote sensing applications and makes RS-application development much simpler and faster. Because it is based on the data distribution client and the whole SIG based SIG-DDS, the web-based user interface can provide data in different types, different formats, and different goals in a universal user interface, which is impossible in the conventional RS-DDS.

2. RS-DAL: the XML based RS-data source accessing language

Because the data sources in the system are often implemented on different platforms, they must be

Proceedings of the Second International
Conference on Semantics, Knowledge, and Grid (SKG'06)
0-7695-2673-X/06 $20.00 © 2006 IEEE

invoked by a platform-independent protocol, and SOAP [8] is the choice. Furthermore, a well-defined extensible RS-data accessing language which suits any data source and any data type is needed for the RS-data source accessing based on SOAP. In order to make it platform-independent, XML should be adopted as its format. It is named RS-DAL.

The following are possible operations on a RS-data source or RS-data broker: searching an image, downloading an image (or a part of an image), and querying its capability description. They are all implemented by RS-DAL in *REQUEST-RESPONSE* mode. In the RS-DAL, a *REQUEST* should has a root element named <query>, <access>, <getCapability>, <getStatus>, or <getResult>, for functions of searching an image, downloading an image, getting capability description of data source, and for commands of getting operation status, getting operation result. A *RESPONSE* should has a root element named <response>, <status>, or <result>, for responses of starting an operation, getting operation status, and getting operation result. An access to the data source (for searching an image, downloading an image, or getting its capability description) should accomplish in three steps: starting the operation by sending an operation request to the data source, monitoring the operation status for its finish, and getting its result.

Considering the extensibility of the system, the RS-DAL is designed as a XML based extensible language. Both the request of <query>, <access>, <getCapability>, and the response of <result> can be extended a lot. By extending the request and response, the RS-DAL will be suit for almost all kinds of data sources and remote sensing applications.

3. Implementation of SIG-DDS

The SIG-DDS is implemented by Java language and tested in Apache Tomcat with Axis 1.2. Because RS-data is different in type, origin, and organization, the implementation of data source must be flexible. The framework of "plug-in with configuration" is a good choice. Now, several data sources are implemented in this framework, they are different in type, origin, and organization. As soon as the data source is implemented, deployed, and registered into the resource registry service of grid, it can serve for the clients. The continuing joined data sources provide powerful data supports for the SIG-DDS. Because they are far different in goals and functions, the data sources should be implemented, deployed, and registered to raise the capability of system step by step.

Data brokers will enlarge the capability of system, raise the productivity of system, and make the system more powerful and easy-to-use. A data broker named Remote Sensing Data Catalogue Service which can collect, organize and index the information of RS-data is designed and implemented now. It provides function of data searching for users by collecting information from a series of data sources and re-organizing, storing and indexing the information. Users can search image data from the Data Catalogue Service to get its information to get the image data from data source.

When the data sources and data brokers are implemented, the clients for accessing them are implemented at the same time. By appending some additional function for users, the client of SIG-DDS is built up. Furthermore, a Web based user interface is implemented based on the clients and JSP technology. Users can search an image by conditions such as time, satellite ID, sensor type, and region covered; and operation on the results such as download all data, view the quick view, get the meta-data, and download data of some wave bands.

A test of the SIG-DDS deployed on Apache Tomcat 5.0.30 and Axis 1.2 shows that user can search image data, download data, and get meta-data, etc. on a mass of data sources through a uniform user interface. It also shows that the SIG-DDS can distribute Remote Sensing Data with different types, different sources and different goals through a uniform programming interface for the remote sensing applications.

Acknowledgments. This work is partially funded by the 863 project with ID 2003AA135113.

References

[1] GCMD Mapserver of NASA, http://mapserver.gsfc.nasa.gov/website/CEOS/viewer.htm
[2] The Coordinated Enhanced Observing Period, http://www.ceop.net/
[3] Open Geospatial Consortium Inc., *Web Coverage Service(WCS) Version 1.0.0*, https://portal.opengeospatial.org/files/?artifact_id=3837
[4] Open Geospatial Consortium Inc., *Web Feature Service Implementation Specification*, http://www.opengeospatial.org/docs/02-058.pdf
[5] Open Geospatial Consortium Inc., *Web Map Service Implementation Specification*, http://www.opengeospatial.org/docs/01-068r2.pdf
[6] ZhenChun Huang, LeiGu, Bin Du and Chuan He: *Grid Resource Specification Language based on XML and its usage in Resource Registry Meta-Service*, 2004 IEEE International Conference on Services Computing, Sep.15-18, ShangHai, China, pp. 467-470
[7] OASIS: *UDDI Specification TC*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec
[8] SOAP: The fundamental message enveloping mechanism in Web services. http://www.w3.org/TR/SOAP.