One-stop Service for Multi-Source Heterogeneous Remote Sensing Data

ZhenChun Huang[1, a], AnRun Zhong[1, b], GuoQing Li[2, c]

[1]Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China

[2] Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, 471023, China

[a]email: huangzc@tsinghua.edu.cn, [b]email: zar16@mails.tsinghua.edu.cn, [c]email: ligq@radi.ac.cn

Keywords: Remote Sensing Data Service; One-stop Service; Data Integration; Data Management

Abstract. The increasing growth of remote sensing data and geoscience research poses great challenges to remote sensing data services. One of them is integrate heterogeneous data provided by multi space agencies and provide a one-stop data service. In this paper, a 3-layerd architecture is proposed and two different ways for service implementation logic are discussed first. Based on the discussion about some technologies for implementation of one-stop data service such as data collection, data management, and service interfaces, a one-stop remote sensing data service system is built, which connects 9 data sources from different agencies, and collects more than 28 million remote sensing data items. It shows that one-stop data service is able to integrate heterogeneous remote sensing data from multi data sources and provide one-stop service with universal programming and user interfaces, and to enable the evolution from remote sensing data services to remote sensing data processing platforms which provide PaaS (Platform-as-a-Service) services for developing more remote sensing data analysis applications.

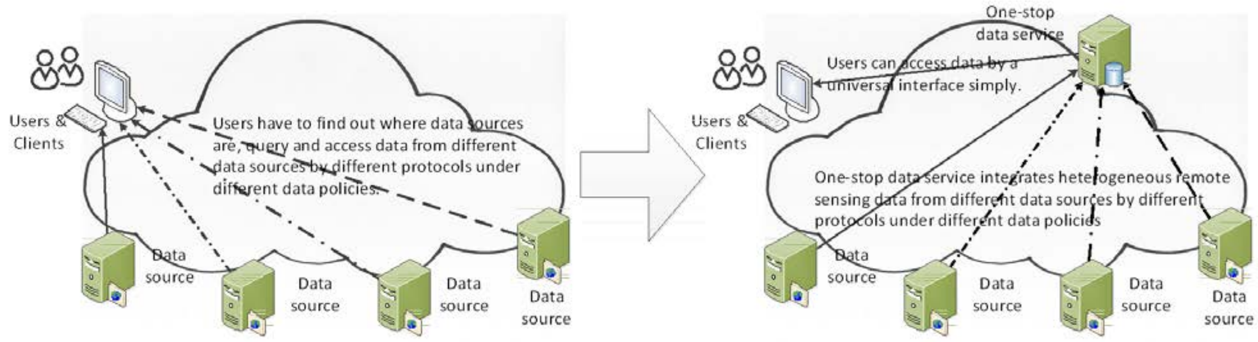
Introduction

Remote Sensing data are growing in size and variety very fast. New high spatial, temporal and radiometric resolution remote sensing systems based on satellite, airborne and ground are available continually. The volume of RS data continues to multiply exponentially due to the launch of new flying platforms, hosting more powerful, multispectral, and accurate sensors; for example, European Space Agency (ESA) archive exceeds 1.5 PB in 2015 and it is foreseen that this volume will exceed 2 PB in few years. At the same time, the increase of computing power enables simulations at a global scale with unprecedented accuracy. For many scientists, the ever-growing observation capability and computing power enable a completely new approach for data intensive scientific discovery known as Fourth Paradigm [1]. It is widely accepted that earth sciences have been some of the disciplinary domains most strongly pushing, and potentially benefiting from, the e-Science approach which are facing big data challenges [2]. It is a great opportunity to enhance our knowledge of the Earth System, but it also poses great challenges to the integration and service of remote sensing big data, such as data collection, data management, and one-stop data service.

To respond to the challenges, some projects are launched for data organization, management and service. For example, Open Geospatial Consortium (OGC) develops interface standards, such as Web Coverage Service, Web Feature Service, Web Map Service, etc., to enable the sharing of remote sensing image data [3]. GEO, the most important partnership of governments and organizations in earth observations, creates a Global Earth Observation System of Systems (GEOSS) to link Earth observation resources world-wide, and make those resources available for better informed decision-making [4].

Furthermore, new research productions such as grid computing and cloud computing are continuously used in remote sensing data storage and sharing. For example, Roberto et al. built remote sensing data service G-POD for high speed remote sensing data sharing and processing [5], GEO tried to share remote sensing data based on grid computing technology [6], Li et al. proposed new data service system for inter-agency remote sensing data sharing and distribution based on spatial information grid [7], Huang et al. built new remote sensing data infrastructure based on on-demand data service and meta-data adapter [8], etc. Although it is a bit short at performance and security, cloud computing is still widely used in remote sensing data management and processing because of its outstanding features such as parallel processing and manageability. One of the most famous examples is that NASA reconstructed a part of remote sensing data service to AWS platform of Amazon, and built NASA Earth Exchange (NEX), a platform for scientific collaboration, knowledge sharing and research for the Earth science community [9].

But, most research projects on remote sensing data storage and management are still short at data sharing, one-stop service and collaborative research supporting. For example, when remote sensing data in multi space agencies are needed in a scenario, the remote sensing application has to search data in all data service systems of multi space agencies, download data found to an application-side workstation and process them. The application must search and download data via different programming interfaces, and parse meta-data from different data providers. It makes applications complex and difficult to develop (Figure 1a).


Users & Clients
Users have to find out where data sources are, query and access data from different data sources by different protocols under different data policies.
Data source
Data source
Data source
Data source
One-stop data service
Users can access data by a universal interface simply.
Users & Clients
One-stop data service integrates heterogeneous remote sensing data from different data sources by different protocols under different data policies
Data source
Data source
Data source
Data source

(a) Classical data service (b) One-stop data service

Fig.1. Classical data service vs. one-stop data service

In this paper, a solution to integrate heterogeneous remote sensing data services provided by multi space agencies is proposed. It tries to create a one-stop data service for remote sensing applications based on heterogeneous data from multi data providers, and simplify the development of applications (figure 1b). In this paper, after the architecture design of data service system in section 2, section 3, 4 and 5 proposes implementation details in meta-data collection, data storage, and service interface. In section 5, a one-stop remote sensing data service instance is also built up. At last, section 6 concludes the paper and discusses about future work in remote sensing data service.

Architecture Design

In order to integrate heterogeneous data from multi agencies, the one-stop remote sensing data service adopts a layered architecture shown in the dashed box of figure 2. The one-stop remote sensing data service collects data from multi data sources by data source connectors, serves users and clients by HTTP based data service interface, and connects data service interface and data source connectors by service implementation logic.

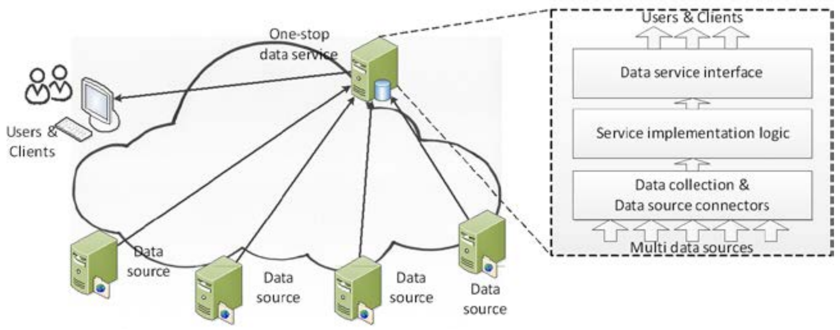


Fig.2. Architecture

“Service implementation logic” is the key layer of service system. It collects data from data providers by data source connectors, organizes data got from data providers, and serves queries from customers via service interface. There are two different ways for the service implementation logic to manage data from data providers and serve customers. One of them is called “offline data collection”, which collects data incrementally from distributed data providers periodically, and serves query from users by data collected before. This way is better in service performance, but the collected data may be asynchronous with their sources when data in providers are changed. For example, image data which are captured later than the last data collection will not visible to users.

The other way is called “online data integration”, which translates and forwards operations on data (such as query and access) from users to distributed data providers “on line”, and keep the operation result always on time. It is more synchronous, but often slower due to its higher implementation complexity. According to the heterogeneity among accessing interfaces for data from data providers, metadata adapters are required for translating request and response of queries and accesses, for both offline data collection and online data integration.

To adopt strong points of both ways, a hybrid of “online data integration” and “offline data collection” can be used by service implementation logic in one-stop data service. In the hybrid implementation logic, data service collects data periodically and serves customers by data in local database like it does in “offline data collection”. But, for the new data which are not collected yet, data service forwards the operation requests to data sources like it does in “online data integration”, and make the result synchronous with data sources.

Data organization and storage

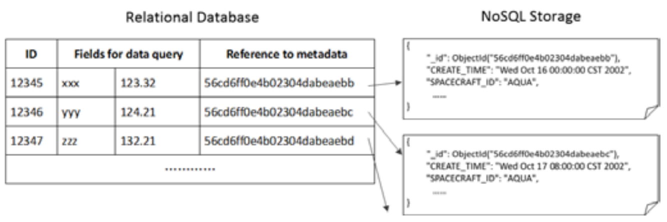
By its very nature, metadata is a property set of remote sensing data, which describes features of image captured such as its coverage, capture time, and parameters about satellite/sensor. For its flexibility and extensibility, metadata can be organized as a document, a data structure composed of key and value pairs. Different metadata document may be made up by fields with different keys, and the value of fields may be a string, a number, other document, array, and array of documents. For the extensibility, complexity and variety, JSON (JavaScript Object Notation) is used to describe and exchange metadata. For example, metadata of a MODIS captured image may be partially described by JSON as figure 3.

```
{
    "id": "1456304112338__796__956875114",
     "CREATE_TIME": "Wed Oct 16 00:00:00 CST 2002",
     "UL_LAT": 30.05501, "UL_LON": 115.0986,
     "UR_LAT": 29.999105, "UR_LON": 127.035,
     "LL_LAT": 19.940222, "LL_LON": 106.0635,
     "LR_LAT": 19.886341, "LR_LON": 117.0474,
     "CENTER_LAT": 24.997616, "CENTER_LON": 116.54925,
    "SENSOR_ID": "MODIS", "SPACECRAFT_ID": "AQUA",
    "RESOLUTION": 500, "LEVEL": "l2",
    "FORMAT": "HDF", "DATASIZE": 72675949,
}
```

Fig.3. Samples of metadata

Considering the complexity and flexibility of metadata storage, relational database is too strict in its data model to store remote sensing metadata. MongoDB [10], a document based NoSQL database is selected to store the metadata due to its flexibility, extensibility and horizontal scalability. Metadata with different contents and formats can be stored in MongoDB as documents together because of the flexible data model. And the storage can be horizontally extended for more metadata due to its distributed architecture.

But, MongoDB is not good enough at data query, especially for complex queries. In order to search the meta-data more quickly, relational database is also used as an “index database”. Because only a few specific fields in metadata are usually used in data query, these selected “query-able” fields, such as band, sensor, coverage and captured time of image data, are stored in the relational database which is used as an “index database”. And, the whole metadata is stored in MongoDB which is more flexible in its data model (figure 4). In the solution, data id, “query-able” fields in metadata, and a reference to storage item in MongoDB are saved and indexed in relational database for data query. When data query is processed, the relational database engine will start a search for “references to metadata” by given query conditions first. Then, the found references are used to read the entire metadata from NoSQL databases.



ID	Fields for data query		Reference to metadata
12345	xxx	123.32	56cd6ff0e4b02304dabeaebb
12346	yyy	124.21	56cd6ff0e4b02304dabeaebc
12347	zzz	132.21	56cd6ff0e4b02304dabeaebd
…………			

```
{
    "_id": ObjectId("56cd6ff0e4b02304dabeaebb"),
    "CREATE_TIME": "Wed Oct 16 00:00:00 CST 2002",
    "SPACECRAFT_ID": "AQUA",
    ......
}
```

```
{
    "_id": ObjectId("56cd6ff0e4b02304dabeaebc"),
    "CREATE_TIME": "Wed Oct 17 08:00:00 CST 2002",
    "SPACECRAFT_ID": "AQUA",
    ......
}
```

Fig.4. Hybrid storage

On the other side, there are two favorite modes for remote sensing image data bodies: vector which comprised of vertices and paths, or raster which made up by a grid of pixels. Because the geographic location of each cell is implied by its position in the cell matrix, raster data often has a mass amount and is stored as a big file in file system. Vector data, which is much smaller, can be stored not only as a file in file system independently, but also be stored as a BLOB (Binary Large OBject) with its metadata in databases.

In most instances, both meta-data and image data are duplicated into multi copies and stored in geographically distributed storage nodes. The redundant data storage benefits safety and robustness for remote sensing data, and favors performance of data queries and downloading by dispatching query and download tasks parallel. Furthermore, for its extensibility, the implementation of remote sensing data distributed storage depends on storage systems which can be scaled out by adding new storage devices, such as distributed file systems, cloud storage systems, and so on.

Data collection

There are several ways to collect remote sensing data, one of them is to define a network protocol for collection. SOAP or RESTful web services are very popular for data collection. These services often provide functions such as metadata queries and image data accesses for data collection. The metadata query functions search metadata databases by given conditions in the requests received, and send the results in pre-defined formats. The data exchange formats often defined based on XML or JSON. An XML based asynchronous metadata query protocol is defined as a SOAP, which receives a request with the condition of query, and returns search results which match condition in XML format. A possible request for data from Landsat7 satellite, covers Beijing (115.7°E-117.4°E, 39.4°N-41.6°N), and captured in the last ten days of 2010 is shown in Figure 5.

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
   <condition relation="AND">
      <condition>
         <operator>EQ</operator>
         <attr>SPACECRAFT_ID</attr>
         <value>Landsat7</value>
      </condition>
      <condition>
         <bbox>115.7,39.4,117.4,41.6</bbox>
      </condition>
      <condition>
         <operator>GT</operator>
         <attr>CREATE_TIME</attr>
         <value>2010-12-22 00:00:00</value>
      </condition>
      <condition>
         <operator>LT</operator>
         <attr>CREATE_TIME</attr>
         <value>2010-12-31 23:59:59</value>
      </condition>
   </condition>
</request>
```

Fig.5. Sample of condition strings

```
<?xml version="1.0" encoding="UTF-8"?>
<result_list>
  <result>
    <field name="id">
          LSAT_LE71220322010365EDC00</field>
    <field name="CREATE_TIME">
          Fri Dec 31 00:00:00 CST 2010</field>
    <field name="SPACECRAFT_ID">LANDSAT7</field>
    <field name="SENSOR_ID">ETM+</field>
    <field name="UL_LON">117.40405</field>
    <field name="UL_LAT">41.28446</field>
    <field name="UR_LON">119.64354</field>
    <field name="UR_LAT">40.95797</field>
    <field name="LL_LON">116.91972</field>
    <field name="LL_LAT">39.682</field>
    <field name="LR_LON">119.10786</field>
    <field name="LR_LAT">39.36316</field>
    <field name="RESOLUTION">30</field>
    <field name="CLOULD_COVER">0.0221</field>
    <field name="LEVEL">l2</field>
    <field name="FORMAT">Geotiff</field>
    <field name="DATASIZE">228085541</field>
  </result>
  <result>
     <!-- ………… -->
  </result>
</result_list>
```

Fig.6. Sample of query results

There are three kinds of conditions in this definition. One of them is “simple compare” defined with elements named <operator>, <attr> and <value>, which is often used to query by satellite ID, sensor ID, observation time, and so on. Another one is “bounding box” defined with a <bbox> element filled by upper and lower bounds of longitude and latitude, which is often used to query by coverage area as a bounding box. The last one is logical operation on conditions, such as AND, OR and NOT, for complex queries.

In the result string, metadata found are organized as a list of <result> elements with <field> children elements for features such as sensor ID, observation time, product level, data format, longitudes and latitudes of four coordinates, as their attributes with name “name”. The <result> elements may have children <field> elements with different “named” attributes, so that data from different spacecrafts and different data agencies can be collected by the same extensible XML-based protocol.

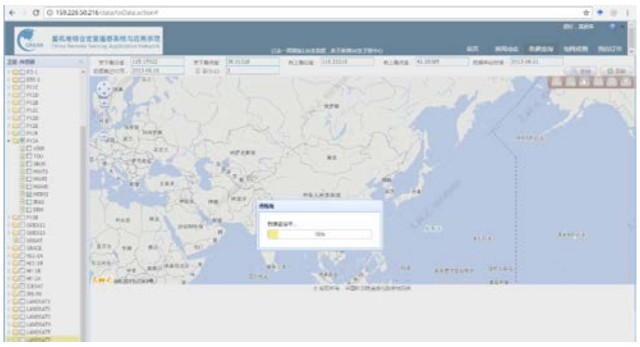
Furthermore, a similar access protocol is also defined as a SOAP service for preparing data to be accessed and get prepared data, so that raw image data can be downloaded.

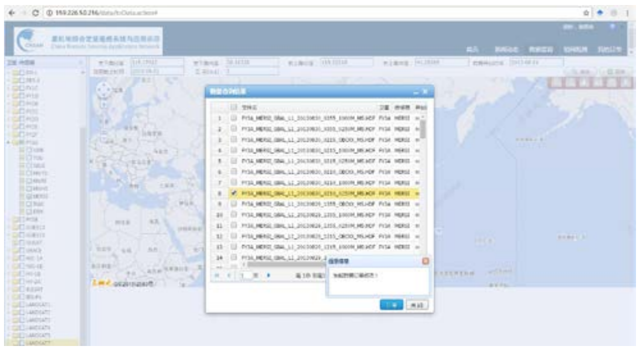
Interface and example of data service

The basic operations of one-stop remote sensing data service should be query() and access(). Operation query() searches data files by given conditions, and access() downloads data files for further processing. The remote sensing data service provides programming interfaces by SOAP based web services and RESTful web service for extensibility and platform-independent.

Furthermore, based on Robinia [11], a data-intensive distributed processing framework, An one-stop remote sensing data service system is built for public users, with a web based user interface for query and access remote sensing data in the data service. Through the web based user interface, users can search remote sensing data by conditions such as cover area, capture time, and name of spacecraft and sensor, and purchase an order to download data in the search result.

In this data service system, 9 data sources from different agencies are connected, and more than 28 million remote sensing data items captured by 30 spacecrafts and 51 sensors are collected. Users can search in the 28 million data items and find data interested to download by a universal interface simply. Thanks to the extensible architecture and implementation technologies, it spends only several months to build and initialize the service system. (Figure 7)


(a) Searching remote sensing data


(b) An order to access remote sensing data selected in search result

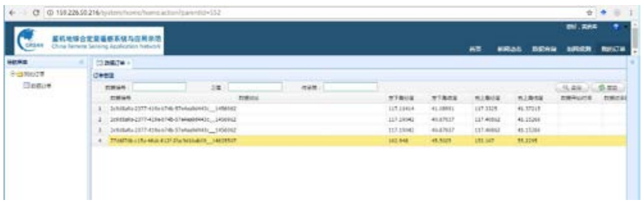

(c) Remote sensing data to be downloaded

Fig. 7. User interface of example service system

Conclusion

One-stop data services integrate heterogeneous remote sensing data from geographically distributed data providers and offer universal programming and user interfaces for their users. It adopts a layered architecture with three layers: service interface layer, service implementation logic layer, and data source connector layer. As the bottom layer, the duty of data source connector layer is to communicate with data sources provided by space agencies and work as a client for data query and access on data sources.

Service implementation logic, the key layer of service system, collects data from data providers by data source connectors, organizes data got from data providers, and serves queries from customers via service interface. There are two different ways for the service implementation logic to serve customers, which are called “offline data collection” and “online data integration”. The one-stop data service adopts a hybrids of online data integration and offline data collection for the balance of service performance and synchronize between data service and data sources.

Another key function of service implementation logic layer is to store and manage data collected from data providers. Because of the complexity and flexibility of metadata storage, MongoDB is used to store the metadata due to its flexibility, extensibility and horizontal scalability. For better query performance, some selected “query-able” fields are stored in the relational database while the whole metadata is stored in MongoDB.

Based on this architecture, a one-stop remote sensing data service system is built, which connects 9 data sources from different agencies, and collects more than 28 million remote sensing data items captured by 30 spacecrafts and 51 sensors. It shows that one-stop data service is able to integrate heterogeneous remote sensing data from multi data sources and provide one-stop service with universal programming and user interfaces.

One-stop data service makes it possible to search and access multi-source heterogeneous remote sensing data with a universal programming interface. Furthermore, if the remote sensing data can be on-demand processed by user-uploaded analysis models “near” where they are stored, the time cost for data transfer will be reduced because only analysis results which is much smaller are downloaded through low-bandwidth WAN. It is often called “nearby processing” or “on-demand processing”. On-demand processing will enable remote sensing data services to process data stored guided by uploaded analysis models. It will evolve the remote sensing data services into remote sensing data processing platforms or remote sensing data infrastructures, which provide PaaS (Platform-as-a-Service) services for developing more remote sensing data analysis applications.

Acknowledgement

In this paper, the research was sponsored in part by National Key Research and Development Plan of China under Grant No. 2016YFB0501504 and National High-Tech Research and Development Program of China (863 Program) under Grant No. 2011AA120306.

References

[1] Hey, T., Tansley, S., Tolle, K. (Eds.), 2009. The Fourth Paradigm: Data-intensive Scientific Discovery, p. 252. Microsoft Corporation edition.

[2] Ma, Yan, et al. "Remote sensing big data computing: challenges and opportunities." Future Generation Computer Systems 51 (2015): 47-60.

[3] Open Geospatial Consortium (OGC), OGC® Standards and Supporting Documents, http://www.opengeospatial.org/standards

[4] GEOSS Core Architecture Implementation Report , http://portal.opengeospatial.org/files/?artifact_id=24315

[5] Cossu, Roberto, et al. ESA Grid Processing on Demand for fast access to Earth Observation data and rapid mapping of flood events. European Geosciences Union General Assembly (2008).

[6] Sekiguchi, Satoshi, et al. "Design principles and IT overviews of the GEO grid." IEEE Systems Journal 2.3 (2008): 374-389.

[7] Li, Guoqing, Dingsheng Liu, Zhenchun Huang, Yi Zeng, and Yong Xue. Spatial data service models in grid environment. In Frontiers of High Performance Computing and Networking–ISPA 2006 Workshops, pp. 598-602. Springer Berlin Heidelberg, 2006.

[8] Huang Z C., On-demand data service for the next generation spatial data infrastructure, Fifth International Conference on Semantics, Knowledge and Grid, SKG 2009. IEEE, 2009: 286-289.

[9] NASA NEX, http://aws.amazon.com/cn/nasa/nex/, 2015-02-28

[10] Chodorow, K. (2013). MongoDB: the definitive guide. O'Reilly Media, Inc.

[11] Gu, Y., Li, G., Zou, Q., & Huang, Z. Robinia: scalable framework for data-intensive scientific computing on wide area network. International Journal of Numerical Analysis and Modeling, Series B, 5(1-2), 97-112.