

An Effective Framework for Distributed Geospatial Query Processing in Grids

Zhou HUANG^{1,2}, Yu FANG¹, Bin CHEN¹, Mao PAN²

¹*Institute of Remote Sensing & GIS, Peking University*

Beijing, P. R. China 100871

²*Key Laboratory of Orogenic Belts and Crustal Evolution, Ministry of Education, Peking University*

Beijing, P. R. China 100871

huangzhou@pku.edu.cn

Abstract—The emergence of Internet has greatly revolutionized the way that geospatial information is collected, managed, processed and integrated. There are several important research issues to be addressed for distributed geospatial applications. First, the performance of geospatial applications is needed to be considered in the Internet environment. In this regard, the Grid as an effective distributed computing paradigm is a good choice. The Grid uses a series of middleware to interconnect and merge various distributed resources into a “super-computer” with capability of high performance computation. Secondly, it is necessary to ensure the secure use of independent geospatial applications in the Internet environment. The Grid just provides the utility of secure access to distributed geospatial resources. Additionally, it makes good sense to overcome the heterogeneity between individual geospatial information systems in Internet. The Open Geospatial Consortium (OGC) proposes a number of generalized geospatial standards e.g. OGC Web Services (OWS) to achieve interoperable access to geospatial applications. The OWS solution is feasible and widely adopted by both the academic community and the industry community. Therefore, we propose an integrated framework by incorporating OWS standards into Grids. Upon the framework distributed geospatial queries can be performed in an interoperable, high-performance and secure Grid environment.

Index Terms—Distributed Query, Grid Computing, Geospatial Information Service, Interoperability.

I. INTRODUCTION

Over recent years, the emergence of Internet has greatly revolutionized the way that geospatial information is collected, managed, processed and integrated [1]. There are several important research issues to be addressed for distributed geospatial applications. First, the performance of geospatial applications is needed to be considered in the Internet environment. In this regard, the Grid as an effective distributed computing paradigm is a good choice [2]. Grid computing aims to develop a number of middleware to interconnect and merge various geographically dispersed resources such as CPUs, storage systems, communication systems, data and software sources, and instruments into a “super-computer” with capability of high performance computation [3,4]. Thus the Grid can be used for effective geospatial data management as well as high-performance distributed geospatial applications [5]. Much related research has been conducted in the intersection field of Geographic Information System (GIS) and Grid computing. “Spatial Information Grid”, namely “Cyber-Infrastructure”,

has been discussed largely by both the academic community and the industry community [6-8].

Current Grid technologies are mature and provide a stable technical basis for geospatial applications. The Globus Toolkit 5 (GT5), which consists of a series of services and software libraries, is the key middleware that provides core Grid capabilities [9]. Based on GT5, various usable Grids are able to be set up feasibly. With the GT5-based Grids, researchers can submit their very complex and compute-intensive tasks which can be automatically assigned onto multiple high-performance machines, even though the machines might be located far apart from each other. Thus the Grid technology facilitates the usage of distributed applications and enables complex remote collaborations. It is suitable for distributed geospatial applications, which are usually involved with geographical dispersed data, transfers of large amounts of geospatial data, and large computational jobs.

On the other side, the heterogeneity problem is another important issue that needs to be addressed for the Internet-based applications. For historical reasons, there are many differences between individual geospatial resources such as data, software and instrumentations. In order to utilize the distributed resources in a generalized way, it makes good sense to developing generalized standards to overcome the heterogeneity. The Open Geospatial Consortium (OGC) was set up for this purpose and is an international consensus standards organization that is leading the development of open standards for accessing and manipulating geospatial information [10]. The representative geospatial information standard of OGC is the OWS (OGC Web Services) [11], which specifies a series of open interfaces and protocols to ensure geospatial information to be accessed in a standardized service-oriented way. The OWS-based specifications have been broadly adopted by the Geographical Information System (GIS) community.

Therefore, it is natural to make a combination of the OWS standards for overcoming the heterogeneity problem and the grid-computing technology for processing highly compute-intensive geospatial tasks. The combination enables an interoperable and high-performance distributed computing environment for various geospatial applications. There have been some related studies on methods for implementing Grid-enabled OGC-compliant services [12,13]. However, in most cases the non-Grid OGC Web services exist broadly and totally reconstructing the Grid-enabled OGC-compliant services is tough and time-consuming. Hence, it makes good

sense to study an effective way for integrating existing OGC Web services into Grids. The integration approach is just the emphasis of this paper.

The rest of this paper is organized as follows. After giving a short overview on the various related technologies, mainly including the Grid technology and the OGC Web services, the integration framework for OGC-compliant Grids as well as how geospatial queries are performed in the framework are proposed. Then, several important issues for the combination, including Grid-enabling the OWS services, Grid security, and service orchestration approaches, are discussed in detail. Finally, we conclude and discuss the future directions.

II. RELATED TECHNOLOGIES

Before enabling the combination of the Grid technology and OWS standards, the two technologies in diverse application scenarios should be introduced briefly. The integration approaches are designed from the observation of the differences.

A. Grid Computing

The Grid technology uses a number of standard protocols to interconnect the geographically dispersed computational resources and data storages together to form a generalized infrastructure. According to Ian Foster, the Grid infrastructure is able to “deliver nontrivial qualities of service” [14]. In the early days the Grid was more or less like a coordinated job system, upon which various remote tasks can be submit in the form of job scripts and performed automatically. And over recent years the Service-Oriented Architecture (SOA) has become a popular computing paradigm. The Web service standards e.g. SOAP (Simple Object Access Protocol) and WSDL (Web Service Definition Language) is broadly accepted and used by the industry community. Hence, the Grid has evolved into the age of service-oriented architecture. The official organization for common Grid standards, namely the Open Grid Forum (OGF), specifies the Open Grid Services Architecture (OGSA) as a standardized open Grid architecture. This means all interactions in OGSA are in the form of service invocations.

In OGSA, the Web Services Resource Framework (WSRF) is defined as the implementation framework of the OGSA model. WSRF is a set of specifications that define the modeling and management of stateful resources using existing Web services technologies [15]. WSRF defines the interfaces of Grid services which address issues like dynamic service creation, lifetime management, notification, and manageability. These features enable applications to interact with Grid services in a generalized service-oriented way (see Figure 1). Hence, the WSRF-based stateful service has more features than the non-Grid Web service and can be deemed as an extension of the common Web service.

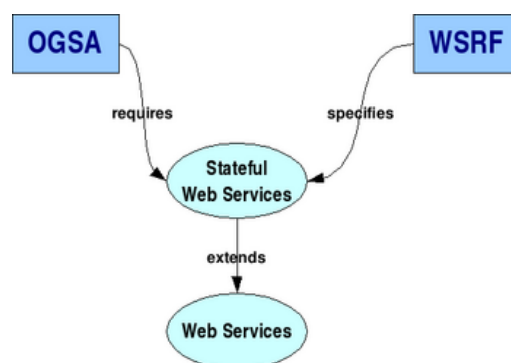


Figure 1. The Grid architecture and its implementation framework [16]

To facilitate the creation of OGSA-based Grids, GT5 is developed as a software system consisting of a series of service libraries, which can provide Grid capabilities (see Figure 2). Based on GT5 users can build OGSA-based Grids conveniently.

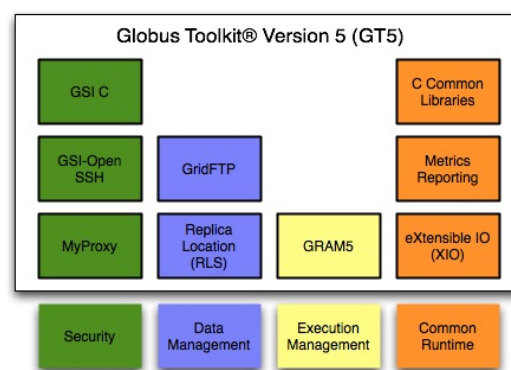


Figure 2. Overview of the GT5 components [17]

GT5 is composed of four types of Grid libraries: security, data management, execution management and runtime libraries. Firstly, security is one of the major requirements for a Grid. GT5 has its Grid security models and implementations like SimpleCA and MyProxy. Secondly, Data management tools like GridFTP for data transfer and RLS for replica management are provided by GT5. Thirdly, GT5 enables execution management capabilities for Grids. For example, GT5 contains a Web service-based Grid resource allocation and management component GRAM5. GRAM5 is a WSRF-based Web service used by computational resources to remotely submit, monitor, and cancel jobs. Lastly, GT5 contains several common runtime libraries for building Grids as well.

B. OGC Web Services

In order to access the distributed geospatial information in a standardized way, OGC has specified several types of geospatial Web services which form the OWS family together. Over recent years, OGC has produced a set of OWS specifications as shown in Figure 3. Currently the representative OWS standards include the Web Map Service (WMS) that allows interactively generating map images by specifying multiple geospatial datasets, the Web Feature Service (WFS) that is used to access the geospatial features on the remote server, and the Web Coverage Service (WCS) that provides capabilities for obtaining remote raster datasets.

These OWS specifications describe the standardized formats for service operation requests, parameters and response contents.

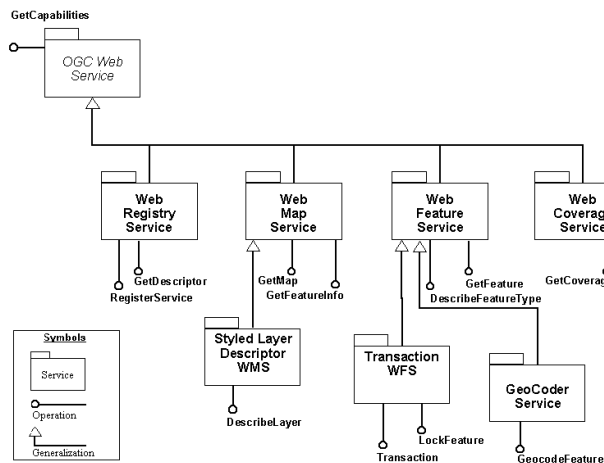


Figure 3. Overview of OGC Web Services [18]

The WMS specifies a number of standardized operations: the GetCapabilities operation returns parameters about the WMS service and the available layers, the optional WMS getFeatureInfo operation is able to provide a more detailed description of objects included in the maps, and the GetMap operation returns a map rendered in a format such as PNG, GIF or JPEG.

The WFS provides several standardized operations for accessing the geospatial features stored in the remote WFS server. In addition to the GetCapabilities operation that returns a XML-based brief description document about the WFS service, the DescribeFeatureType operation can return more detailed metadata such as the XML schema of a specified feature type. Moreover, the GetFeature operation actually performs the query of the feature data and returns a Geography Markup Language (GML) document.

Similar to WMS/WFS, the WCS defines a number of standardized discovery, query or data transformation operations as well: the mandatory GetCapabilities operation that returns the brief metadata of the WCS service, the DescribeCoverage operation that provides more detailed coverage metadata, and the GetCoverage operation that returns a coverage object in a raster format like DTED, GeoTIFF, or NITF.

Once the OGC Web service (WMS/WFS/WCS) is set up, users can submit their requests to it using the HTTP protocol. Currently the mainstream GIS desktop clients such as QGIS and UDig all provide this functionality. In the process of service interaction, the encoding for the requests and the responses should be XML (HTTP POST/SOAP). Thus, interpretable access to geospatial information is achieved and the response data are in a standardized format.

III. INTEGRATED FRAMEWORK OF OWS-ENHANCED GRIDS

Based on the discussions above, integrating the OWS standards into Grids would enable an interoperable and high-performance environment for complex and compute-intensive geospatial applications. We propose an integrated framework for implementing an OWS-enhanced Grid (see Figure 4).

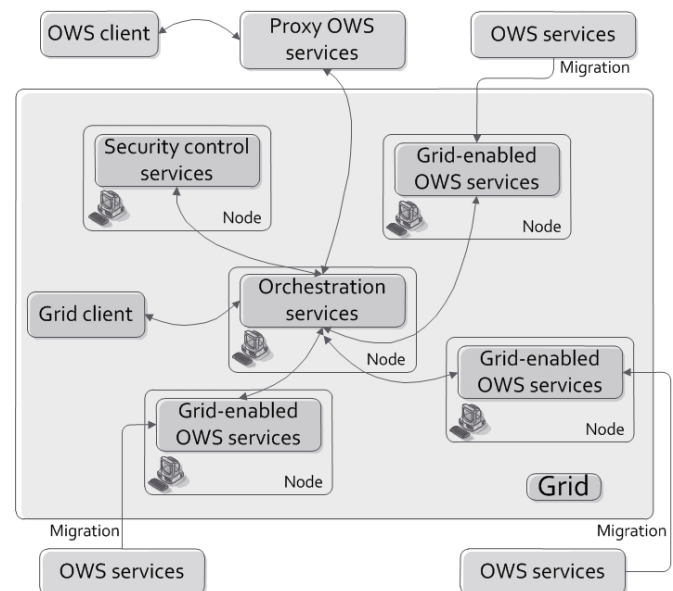


Figure 4. Integrated framework for incorporating OGC Web services into Grids

As shown in Figure 4, in the framework the non-Grid OWS services are migrated into Grid-enabled OWS services. Besides Grid-enabled OWS services, security control services are set up to form a security environment for service usage in Grids and orchestration services are built for scheduling multiple Grid-enabled OWS services. The Grid client is able to submit cooperative geospatial queries by invoking the orchestration service. Moreover, to facilitate non-Grid users the proxy OWS service is built for wrapping the Grid-enabled orchestration service. Thus non-Grid users can connect the proxy OWS service to obtain the Grid capabilities even through non-Grid OWS clients such as QGIS and UDig. In order to realize this vision, the following research issues should be addressed:

- 1) Grid-enabling OWS services. In order to build the Grid-enabled OWS service, the common OWS service should be adapted to suffice the WSRF specifications, which are not supported by OGC at present. Because of the broad existence of implemented OGC Web services, it would be better to wrap them into Grid-enabled OWS services rather than to implement the Grid-enabled OWS services from backend. An effective migration approach is needed to construct Grid-enabled OWS services which enable non-Grid OGC services to be accessed via standard Grid protocols.

- 2) Security. Security is a basic requirement for distributed geospatial applications [19]. A robust security mechanism should be designed for the OWS-enhanced Grid to guarantee secure usage of Grid resources such as services, data and instruments. GT5 implements a certificate-based security framework for Grids and OGC sets up the Security Working Group (SWG) to establish an interoperable security framework for OWS services to enable protected geospatial information processing [20]. The work can be referred in our security mechanism design. Moreover, authentication (proof of identification) and authorization (access control) are two fundamental requirements for establishing the licensing of geospatial information in OWS-enhanced Grids.

- 3) Distributed query service orchestration. Although the

Grid-enabled OWS service can provide interoperable access to geospatial information, one of the most important characteristics of the Grid is the distributed query processing capability, which enables users to perform cooperative geospatial tasks. Hence an appropriate service orchestration approach should be designed to schedule individual Grid-enabled OWS services for distributed geospatial queries. GRAM5 in GT5 allows users to define a distributed query process, but it cannot support automatic query service orchestration. In this study, we aim to develop an automatic service orchestration method with the support of the Grid catalogue that provides the distribution information of services and geospatial data. In this scenario, the orchestration service is able to accept global queries from users, automatically schedule the individual Grid-enabled OWS services hosting on Grid nodes, and return final responses (see Figure 5).

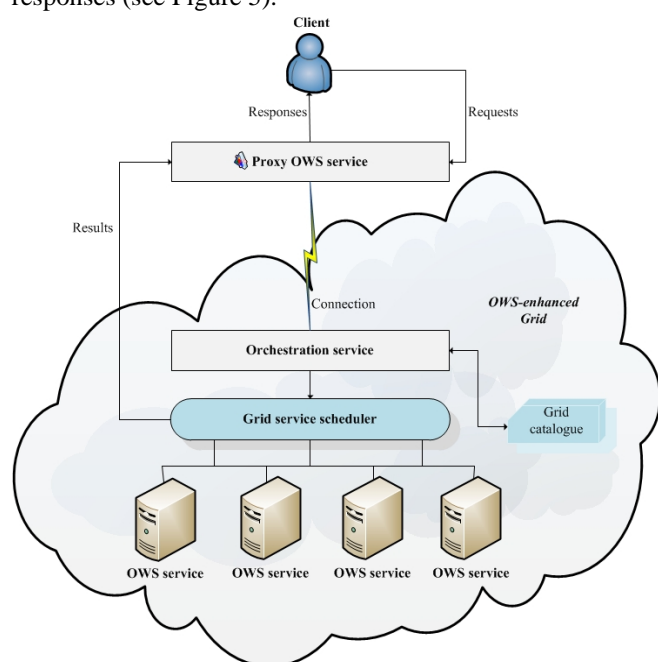


Figure 5. Geospatial Query processing in OWS-enhanced Grids

IV. GRID-ENABLING THE OGC WEB SERVICES

As described in Section II, the OWS services currently comply with the specifications of standardized Web services. For each OGC service type (WMS/WFS/WCS), several mandatory and optional operations like GetCapabilities are defined. These operations can be requested by an OWS client via HTTP and performed by an OWS server. The requests and responses are encoded as SOAP messages, and the embedded geospatial data are in several standardized exchange formats such as GML (WFS), binary (WMS and WCS) and ASCII text (WCS).

In contrast to the standardized Web services, which are stateless, the WSRF-based Grid services are stateful. Under OGSA, an instantiated WSRF-based Grid service is called a WS-Resource, which has lifetime and can expose resource state to clients [21]. This feature facilitates building of consolidated service workflows and enables complex distributed geospatial queries in Grids. The Grid service client uses the WS-Addressing protocol, which specifies a construct called an endpoint reference (EPR), to address a

WS-Resource. Besides the Uniform Resource Identifier (URI) of the Grid service, the endpoint reference contains more information such as the unique resource key that is created when the WS-Resource is instantiated. Hence, in order to invoke the stateful WS-Resource, the Grid service client has to know the URI of the associated Grid service and the resource key to reach the desired resource.

Therefore, in order to integrate the OWS services into Grids, it is necessary to implement WSRF features for the OWS services. GT5 includes a toolkit called Crux (a replacement of the Java WS Core in GT4) for building WSRF-based Grid services. So we can use the Crux toolkit to implement Grid-enabled OWS services which are WSRF-based and have standardized OWS operations. The developing steps for Grid-enabling the OGC Web services are concluded as follows (see Figure 6):

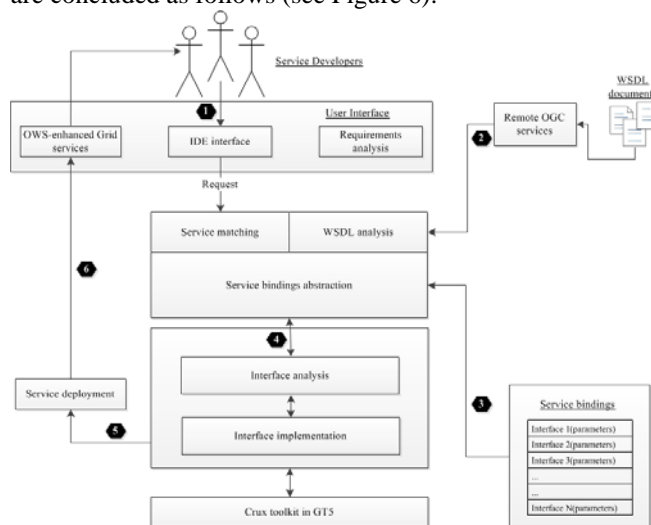


Figure 6. Developing flow for Grid-enabling the OGC Web services

- 1) Service developers carry out requirements analysis and perform the service matching operation to find the OGC Web service that needs to be integrated into Grids;
- 2) The WSDL document of the desired OWS service returns;
- 3) Service developers analyze the service bindings (operations) from the WSDL document;
- 4) Service developers use the Crux toolkit of GT5 to build a Grid service which contains the same operations with the OWS service, and each Grid service operation can simply be implemented by wrapping the associated OWS service operation;
- 5) Service developers can deploy the implemented service code under a GT5-based Grid environment;
- 6) The Grid-enabled OWS service is built up finally.

V. SECURITY

Security is another import issue that needs to be addressed in OWS-enhanced Grids. GT5 implements the Grid Security Infrastructure (GSI) to provide security functionalities [22]. GSI uses public key cryptography (also known as asymmetric cryptography) as the security basis. Standardized X.509 certificates are used to identify Grid entities such as users, hosts, services and the Certificate Authority (CA).

GSI uses a third party namely CA to issue the certificates

which contain a public key. The certificate owner (entity) must keep the other key private (a private key) and never reveal them to the public. Then an entity can encrypt a message using its private key and send both the message and its certificate to a destination entity. If the recipient can decrypt the message using the sender's public key, the identity of the sender is certified because the sender must have used the private key to encrypt the message.

Hence, GSI uses X.509 certificates to prove the identity of each Grid entity, and can thus reach the objective of authentication in Grids. Furthermore, GSI implements an authorization framework via the combination of a certificate and several authorization constructs, e.g. the grid map file that can define an access control list for the Subject Name (SN) of a specified certificate, an authorization service that uses the Security Assertion Markup Language (SAML) to describe and exchange security data. On the other side, OGC SWG has proposed a new authorization standard Geospatial eXtensible Access Control Markup Language (GeoXACML), which provides standardized specifications for geo-specific access restrictions for geographic data [20].

The aforementioned work is a good basis for our security framework design. Incorporating the GeoXACML standard into GSI can build a feasible and elastic security framework in the OWS-enhanced Grid (see Figure 7). In the authentication level, X.509 certificates issued by the CA are used to certify the identity of Grid entities. As for the authorization, we make a combination of the certificate and a specified role that comprises of several privileges. The role privileges are in four levels: service, operation, dataset, and data restrictions. After the combination, the Grid entity such as a user that holds a certificate must have rights to access the resources defined in its role model, including the services, the operations, and the datasets with specified data restrictions. The role data can be managed by an authorization Grid service (namely the security control service in Figure 4), which uses SAML for certificate storage and GeoXACML for geo-specific access restrictions. The service-oriented approach is better than the grid map file because it provides interoperable and standardized access to the role data and thus suits the distributed Grid environment.

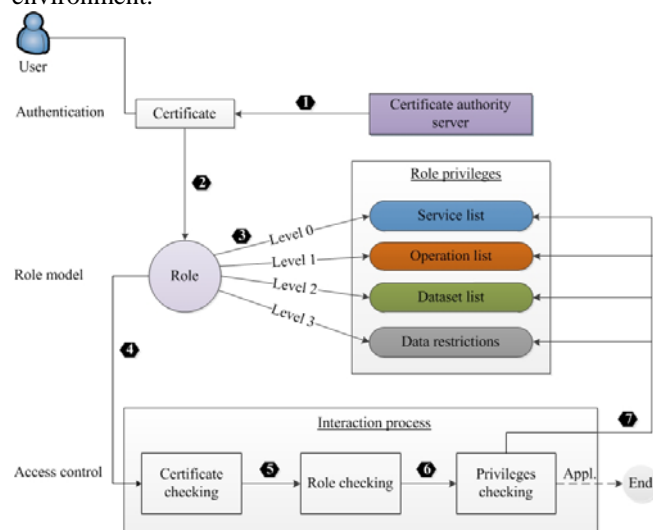


Figure 7. Security framework in OWS-enhanced Grids

After the authorization service is built up, the access

control process in the OWS-enhanced Grid can be concluded as follows:

1) Certificate checking. The mutual authentication is performed firstly. When a client request, that contains a certificate and an encrypted message, reaches a destination, the destination should check the certificate's validation and certify the identity of the client by using the client's public key to decrypt the message.

2) Role checking. After the authentication process, the destination must submit a request to the authorization service to get the role information that is combined with the client's certificate and comprises of several privileges denoted as service list, operation list, dataset list and data restrictions.

3) Privileges checking. When the privileges information returns, it is necessary to use the information to verify whether the client can access the destination resource or not. For example, when the client requests a Grid-enabled WMS service with the following parameters "REQUEST=GetMap &LAYERS=Oceans,Countries,Cities&STYLES=&SRS=EPSG:4326&BBOX=-124,21,-66,49&WIDTH=600&HEIGHT=400&FORMAT=image/png", the destination needs to check the privileges information to judge whether the client can access the WMS service, the GetMap operation, the datasets Oceans, Countries and Cities, and the spatial extent "BBOX=-124,21,-66,49&WIDTH=600&HEIGHT=400". If one of the resources or restrictions is not matched with the client's role information, the request would be rejected automatically, and otherwise the request would be processed by the Grid-enabled WMS service.

VI. GRID-BASED GEOSPATIAL SERVICE ORCHESTRATION

With the introduction of the Grid service technologies, e.g., WSDL for describing the interfaces (operations) of the Grid-enabled OWS services, SOAP that is used to exchange messages between the services, and MDS (Monitoring & Discovery System) for publishing and discovering the Grid services, distributed geospatial applications based on OWS-enhanced Grids can be quickly composed.

GT5 provides the GRAM5 toolkit as a WSRF-based Grid service that allows submitting and monitoring jobs. By using the GRAM5 service, some of the distributed geospatial applications can then be composed as executables in a Grid job. The grid job executed by GRAM5 is described in Resource Specification Language (RSL) [23]. The simple example as shown in Table I illustrates a staging Grid job. The job will first copy the "/bin/echo" command from a remote server to a local file named "/tmp/echo". Then it runs the /bin/lis command by specifying "/tmp/echo" as an input argument. Finally, it will clean up the echo file when the execution process is accomplished.

TABLE I. A SIMPLE EXAMPLE OF RSL-BASED GRID JOB

```
&(rsl_substitution = (FTP_SERVER "gsiftp://gis4g.pku.cn:2811"))
(file_stage_in = $(FTP_SERVER)/bin/echo /tmp/echo))
(executable=/bin/lis) (arguments=-alt /tmp/echo)
(file_clean_up=/tmp/echo)
```

It is observed that the GRAM5 is suitable for the traditional jobs which comprise of a number of local command operations, and cannot thus support the geospatial

service orchestration.

On the other side, there has been a promising development on the orchestration technology for the common Web services. A number of Web service orchestration languages or specifications have been proposed, e.g., Business Process Execution Language for Web Services (BPEL or BPEL4WS), Web Service Choreography Interface (WSCI), and Business Process Management Language (BPML) [24]. Among these specifications, BPEL is a broadly accepted industry standard approved by the OASIS (Organization for the Advancement of Structured Information Standards) consortium and is supported by several open source service orchestration engines such as ActiveBPEL [25].

BPEL is able to construct complex SOA-based workflows. The language is based on XML and supports flow control, asynchronous communication and business process transactions [26]. The BPEL-based document is composed of activities that are the basic BPEL element and describe the control flow, such as invoking a Web service, assigning a value to a variable, or executing several sub-activities in sequence or parallel. However, BPEL is not designed to invoke stateful Grid services and consequently the BPEL-based engines cannot handle the BPEL process consisting of WSRF-based service invocations.

Therefore, it is necessary to design a new framework for the query service orchestration in OWS-enhanced Grids. We mainly address three issues for this purpose: the resource integration mechanism in OWS-enhanced Grids, the Grid-enabled OWS Service Orchestration Language (GOSOL), and the orchestration algorithm.

A. Resource Integration Mechanism

In most cases, geospatial data are geographically distributed on different nodes in the Grid. Some dispersed datasets might have the same schema and attributes, thus they are deemed as fragments of a resource and can be integrated into a virtual "global" dataset. Hence, the orchestration service is able to handle the requests for querying the integrated global dataset. Any OWS operation to the orchestration service can be split into several parallel operations to the Grid-enabled OWS services. This feature enables cooperative query processing and thus facilitates the query performance.

The resource integration process can be performed conveniently in OWS-enhanced Grids. The OWS services provide some operations which are able to get the description information of the original geospatial datasets. The GetCapabilities operation of WMS/WFS/WCS returns an XML-based brief service metadata document that describes the resources of the specific server implementation. The DescribeFeatureType operation of WFS generates a schema description for feature types serviced by a WFS implementation. The DescribeCoverage operation of WCS retrieves an XML document that fully describes the request coverage data. These OWS operations can be used as the initial source of metadata for the resource integration. The integration approaches can be concluded as follows (see Figure 8):

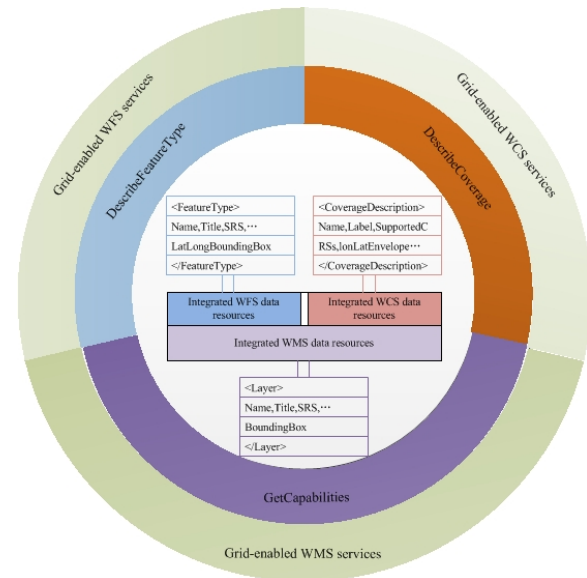


Figure 8. Resource integration in OWS-enhanced Grids

1) Integrating the Grid-enabled WMS data resources. Through the GetCapabilities interface of WMS, the detailed description information on WMS data resources (layers) can be obtained. The information contains the layer name, title, SRS (Spatial Reference System), BoundingBox (namely the spatial extent) and so on. If the description information of a layer, except the BoundingBox, matches that of the other layer, then the two layers can be integrated into a virtual global layer. Hence, through performing the comparison iteratively the integrated virtual WMS data resources are formed and can thus be accessed by the orchestration service that is able to split the query involved with an integrated (global) resource into several parallel sub-queries against local WMS services.

2) Integrating the Grid-enabled WFS data resources. Besides the GetCapabilities operation, WFS uses the DescribeFeatureType operation to return the WFS data resource's detailed schema information, including the feature type name, title, SRS, LatLongBoundingBox (namely the spatial extent), etc. Similar to the step above, the WFS data resources (feature types) with the same schema can be integrated into a global resource.

3) Integrating the Grid-enabled WCS data resources. Besides the GetCapabilities operation, the DescribeCoverage operation of WCS returns the WCS data resource's detailed metadata. The information includes the coverage name, label, supported SRS, LonLatBoundingBox (the spatial extent), and so on. Then, the WCS data resources (raster coverages) with the same schema can be integrated into a global WCS data resource.

B. Orchestration Language: GOSOL

In order to enable service orchestration in OWS-enhanced Grids, it is necessary to design an appropriate service orchestration language. We proposed the Equivalence Distributed Program (EDP), which comprises of local GSQL (Geographic Structured Query Language) query statements and control flow statements, to express the complex distributed GSQL query process [27]. Thus EDP might be useful for the service orchestration language design. However, in contrast to distributed GSQL query

processing, service orchestration in OWS-enhanced Grids is much more straightforward because each OWS query operation to the integrated (global) resource can simply be split into several parallel local OWS operations. So we define a brief service orchestration language GOSOL. Then, the global OWS query operation would be translated into a distributed program described with GOSOL, which can be executed to get the final query result.

In GOSOL, there are two basic sentence types: the local service invocation sentence and the parallel service invocation sentence. The local service invocation sentence consists of local Grid-enabled OWS service address (URI), the OWS operation like the GetMap operation of WMS, and related parameters such as the data resource name, SRS, the spatial extent, output format, and so on. The parallel service invocation sentence illustrates that the sub-sentences (i.e. the local service invocation sentences) should be invoked simultaneously. Moreover, the eXtensible Markup Language (XML) is used as the representation format of GOSOL. Table II demonstrates the XML-based examples of typical GOSOL sentences.

TABLE II. EXAMPLES OF TYPICAL GOSOL SENTENCES

Type	Example
Local service invocation sentence	<pre><LSIS URL="http://192.168.1.101/wms" Operation="GetMap" Parameters="LAYERS= Countries&STYLES=&SRS=EPSG:4326&BBOX=- 126,26,-86,46&FORMAT=image/png" /></pre>
Parallel service invocation sentence	<pre><PSIS> <LSIS URL="http://192.168.1.101/wms Operation="GetMap" Parameters="LAYERS= Countries&STYLES=&SRS=EPSG:4326&BBOX=-126, 26,-86,46&FORMAT=image/png" /> <LSIS URL="http://192.168.1.102/wms Operation="GetMap" Parameters="LAYERS= Countries&STYLES=&SRS=EPSG:4326&BBOX=-86, 26,-56,46&FORMAT=image/png" /> </PSIS></pre>

C. Orchestration Algorithm

After the resource integration process, the virtual global data resources can be formed and available to the orchestration service. The Grid catalogue is used to record the mapping information between a global resource and corresponding local resources. Thus the service orchestration can be performed with the support of the Grid catalogue.

Then, as for an individual global data resource S_i , its distribution information can be denoted as a service set $\{SV_{ij}\}$ ($i=1, \dots, N; j=1, \dots, M_i$), where SV_{ij} represents a Grid-enabled OWS service that holds a local resource of S_i , i is a natural number that illustrates the resource index, N represents the number of the global resources, j is a natural number that indicates the service index, and M_i represents the number of services which own the local resource of S_i . And, S_{ij} denotes the local resource of S_i and is available to the service SV_{ij} . So we can then derive the following equations:

$$S_i = \bigcup_{j=1}^{M_i} S_{ij} \quad (1)$$

$$Operation(S_i) = \bigcup_{j=1}^{M_i} Operation(S_{ij}) \quad (2)$$

Equation 2 illustrates that the global query operation to S_i can firstly be split into several parallel local operations to S_{ij} ($1 \leq j \leq M_i$), and then the local Grid-enabled OWS services return some results which can be united as the final result.

Take a global GetMap/WMS operation with the parameters "LAYERS=Countries &STYLES=&SRS=EPSG:4326&BBOX=-126,26,-46,46&FORMAT=image/png" as an example, it is supposed the layer "Countries" is held by the service set {"http://192.168.1.101/wms", "http://192.168.1.102/wms", "http://192.168.1.103/wms"}, then a parallel query workflow can be generated as shown in Table III.

TABLE III. GOSOL PROGRAM OF AN EXAMPLE QUERY

<GOSOL>
<PSIS>
<LSIS URL="http://192.168.1.101/wms Operation="GetMap"
Parameters="LAYERS=
Countries&STYLES=&SRS=EPSG:4326&BBOX=-126,26,-
86,46&FORMAT=image/png" />
<LSIS URL="http://192.168.1.102/wms Operation="GetMap"
Parameters="LAYERS=
Countries&STYLES=&SRS=EPSG:4326&BBOX=-86,26,-
56,46&FORMAT=image/png" />
<LSIS URL="http://192.168.1.103/wms Operation="GetMap"
Parameters="LAYERS=
Countries&STYLES=&SRS=EPSG:4326&BBOX=-56,26,-
46,46&FORMAT=image/png" />
</PSIS>
</GOSOL>

After the service orchestration steps above, a distributed geospatial service workflow described with GOSOL is generated. Then, after executing and controlling the GOSOL program the result of the global OWS query operation returns. The distributed query process, from the resource integration to the service orchestration, is performed automatically. This feature facilitates the usage of OWS-enhanced Grids because users only need to submit global OWS queries without knowing resource distribution information in advance.

VII. CONCLUSION

Integration of Grid computing and OGC Web services facilitates managing, processing and analyzing geospatial information in a more efficient and convenient way. Performance, security and interoperability are key initiatives which need to be resolved under the new computing diagram. This paper proposes an integrated framework of the OWS-enhanced Grid to enable a secure, high-performance and interoperable distributed geospatial computing environment. To make a combination of OGC Web services and Grid computing, three issues including Grid-enabling OWS services, security and service orchestration should be addressed. Solutions of the three issues are proposed and discussed in detail. It is observed that upon the proposed framework distributed geospatial queries are able to be issued and performed. Currently the geospatial service workflow generated by our service orchestration algorithm is pre-compiled. For the next step, we will design a compensation mechanism of the OWS-enhanced Grid to enable handling of the run-time errors that

occur in the execution process of the geospatial service workflow.

ACKNOWLEDGMENT

This research was supported by the following grants: 1) National High Technology Development 863 Program of China (2007AA120502, 2009AA12Z231); 2) National Natural Science Foundation of China (41001218).

REFERENCES

- [1] Z. Huang, Y. Fang and M. Pan, "Toward an integrated framework for geosensor grid", *International Journal of Digital Earth*, vol.3, no.2, pp.207- 216,2010.
- [2] Y. Fang, Z. Huang, B. Chen, L. Wu and D.F. Yin, "Architecture and Key Technologies of Grid Geographic Information System", *Science in China, Series E: Technological Sciences*, vol.51,no.51,pp.102-113,2008.
- [3] I. Foster, C. Kesselman and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations", *International Journal of Supercomputer Applications*, vol.15,no.3,pp.200–222,2001.
- [4] F. Berman, G. Fox and A.J.G. Hey. *Grid Computing: Making the Global Infrastructure a Reality*. Wiley Press,2003.
- [5] Z. Huang,Y. Fang, X.T. Xie and M. Pan, "GEOBARN: A Practical Grid Geospatial Database System", *Advances in Electrical and Computer Engineering*, vol.9,no.3,pp.7-11,2009.
- [6] Z. Shen, J. Luo and C. Zhou, "Architecture design of grid GIS and its applications on image processing based on LAN", *Information Sciences*, vol.166,pp.1-17, 2004.
- [7] G. Aydin, A. Sayar and H. Gadgil, "Building and applying geographical information system Grids", *Concurrency and Computation: Practice and Experience*, vol.20, pp.1653–1695,2008.
- [8] C. W. Yang, W. W. Li, J. B. Xie and B. Zhou, "Distributed geospatial information processing: sharing distributed geospatial resources to support Digital Earth", *International Journal of Digital Earth*, no.3, pp.259-278, 2008.
- [9] Globus Toolkit Documents, Available: <http://www.globus.org/toolkit/> (accessed date 05/2010).
- [10] OGC Specifications, Available: www.openeospatial.org (accessed date 05/2010).
- [11] The OGC Web Services Initiative Document, Available: <http://www.openeospatial.org/projects/initiatives/ows-4> (accessed date 05/2010).
- [12] L. Di, A. Chen and W. Yang, "The development of a geospatial data Grid by integrating OGC Web services with Globus-based Grid technology", *Concurrency and Computation: Practice and Experience*, vol.20,pp.1617–1635, 2008.
- [13] The Grid/OGC Collision Programme, Available: <http://www.jisc.ac.uk/whatwedo/programmes/ereseachgridogccollison.aspx> (accessed date 03/2010).
- [14] I. Foster, "What is the Grid? A three point checklist", Available: <http://www.mcs.anl.gov/~itf/Articles/WhatIsTheGrid.pdf> (accessed date 03/2010).
- [15] WSRF Technical Documents, Available: www.globus.org/wsrf (accessed date 03/2010).
- [16] The Grid Architecture, Available: <http://gdp.globus.org/gt4-tutorial/multiplehtml/ch01s01.html> (accessed date 03/2010).
- [17] The Globus Toolkit 5 Overview Document, Available: <http://www.globus.org/toolkit/about.html> (accessed date 03/2010).
- [18] The OGC/WMS Specification Document, Available: <http://cite.openeospatial.org/OGCTestData/wms/1.1.1/spec/wms1.1.1.html> (accessed date 03/2010).
- [19] J. Lin, Y. Fang, W. Zhang and Z. Huang, "Fundamental aspects of access control for geospatial data", *International Journal of Digital Earth*, vol.2, no.3, pp. 275-289,2009.
- [20] OGC Security Working Group Documents, Available: <http://www.openeospatial.org/projects/groups/securitywg> (accessed date 05/2010).
- [21] The WS-Resource Framework Specification, Available: www.globus.org/wsrf/specs/ws-wsrf.pdf (accessed date 05/2010).
- [22] The Globus Grid Security Infrastructure (GSI) Specification, Available: www.globus.org/security (accessed date 05/2010).
- [23] The Globus Resource Specification Language (RSL) Specification, Available: http://www.globus.org/toolkit/docs/2.4/gram/rsl_spec1.html (accessed date 03/2010).
- [24] C. Visions,"A Comparison of XPDL, BPML and BPEL4WS", Available: http://www.ebpm.org/A_Comparison_of_XPDL_and_BPML_BPEL.doc (accessed date 03/2010).
- [25] The ActiveBPEL Project Introduction Document, Available: <http://www.activevos.com/community-open-source.php> (accessed date 05/2010).
- [26] The OASIS Business Process Execution Language (BPEL) Specification, Available: <http://www.oasisopen.org/committees/wsbpel> (accessed date 05/2010).
- [27] Z. Huang, X. Peng and K. Zhang, "An Approach to Geographic SQL Global Parsing under Distributed Computation Circumstance", *Geography and Geographic Information Science*, vol.21,no.3,pp.18-21, 2006.