A Grid-enabled Web Map Server

Giovanni Aloisio†, Massimo Cafaro†, Dario Conte††, Sandro Fiore†, Italo Epicoco†, Gian Paolo Marra††, Gianvito Quarta†
†) Center for Advanced Computational Technologies/ISUFI, University of Lecce via per Monteroni, 73100 Lecce – Italy
††) Institute of Atmospheric Sciences and Climate of the Italian National Research Council via per Monteroni, 73100 Lecce – Italy
{giovanni.aloisio, massimo.cafaro, dario.conte, sandro.fiore, italo.epicoco, gianpaolo.marra, gianvito.quarta}@unile.it
{d.conte, gp.marra}@ le.isac.cnr.it

Abstract

Today Geographic Information Systems (GIS) provide several tools for studying and analyzing varied human and natural phenomena, therefore GIS and geospatial data has grown so much in both public and private organizations. A Challenge is the integration of these data to get innovative and exhaustive knowledge about topics of interest.

In this paper we describe the design of a Web Map Service (WMS) OGC-compliant, through the use of grid computing technology and demonstrate how this approach can improve, w.r.t. security, performance, efficiency and scalability, the integration of geospatial multi-source data. End users, with a single sign-on, securely and transparently, gets maps whose data are distributed on heterogeneous data sources belonging to one o more Virtual Organizations via distributed queries in a grid computing environment.

Keywords: Geographic Information Systems, Web Map Server, Grid Computing.

1. Introduction

Geographic Information Systems (GIS) include a wide range of different applications including automated mapping and facilities management as well as land information system. Since the number of applications has grown considerably, the term GIS is increasingly used as shorthand for a wide range of computer-based applications involving the capture, manipulation, analysis and display of geographic information. GIS are actually used in several fields (navigation, transport infrastructure planning, geodesy, urban development planning, environment monitoring, etc.) and huge volumes of geospatial data have been produced and collected. Moreover, GIS and geospatial data products produced by different data providers may differ in file format, resolution, coordinate system, datum, etc. Often, data coming from a specific source can contain complementary information with respect to other sources. Integrating all of these data could result in further, innovative and exhaustive knowledge about a topic of interest.

In order to facilitate this multi-source data integration and interoperability, government, private industry, and academia are working together to develop a set of standards and protocols to ensure these goals. In particular, the *Open Geospatial Consortium (OGC)* [1] is a non-profit international organization that is leading the development of standards for geospatial and location based services. The OGC has developed the OGC *OpenGIS Web Services Architecture* [2] to improve the interoperability among geospatial system. However, this technology does not provide any transparent access mechanism for integration of multi-source and geographically distributed data hiding the complexity, nature and location of data sources.

Today, grid computing [3,4] is an emerging technology to solve large scale problems in dynamic, multi-institutional Virtual Organizations (VOs) coordinated by sharing resources such as high-performance computers, observation devices, data and databases over high speed networks, etc. Grid computing provides several capabilities (like secure and distributed resource sharing and virtualization, etc.) that are very suitable to improve the development of OGC Web Services.

In this paper we describe the development of a Web Map Service (WMS) OGC-compliant, through the use of grid computing technology and demonstrate how this approach can improve, with respect to security, performance, efficiency and scalability, the integration and retieving of geospatial multi-source data.

The paper is organized as follow: the next section presents an overview on the Open Geospatial Consortium activity; the openGIS Web Service and related Web Map Server is then described. In section 3 and 4, requirements and the architecture of the proposed web map server are analyzed. Finally, relevant related works are presented before concluding the paper.

2. Geospatial Data and Service Interoperability: the Open Geospatial Consortium

With the spreading of GIS, several organizations began to work to improve the interoperability among geospatial services, data and applications. In particular, the OGC works with government, private industry, and academia to create open and extensible software application programming interfaces for geographic information systems and other mainstream technologies. One important document produced by the OGC is the *OpenGIS Web Services Architecture* [2] that is a specification and description of a common architectural framework for the design and implementation of Open Distributed Processing applications based on the Web Services specification.

2.1 The OpenGIS Web Services Architecture

In this specification, some fundamental components are identified:

- a **Web Coverage Service (WCS)** represents a web interface for accessing multi-dimensional and/or multi-temporal geospatial data (like remotely sensed imagery, ortho-photos, etc),
- a **Web Features Service (WFS)**, defines web interfaces for accessing feature-based geospatial data (vector data like administrative and political information, streets, cities, etc),
- a **Web Map Service (WMS)** defines interfaces for assembling maps from multi-sources data over the Web through conversion of available data to visualizable form,
- a **Web Registries Service (WRS)** specifies web interfaces for finding data or services from registries.

2.2 The OpenGIS Web Map Server

In this section we describe in depth the *OpenGIS Web Map Server* component and its implementation specifications, derived from the OpenGIS Web Map Server Cookbook [5].

The aim of these specifications is to remove the lack of interoperability among data and services provided by vendor's solutions like specific vendor's proprietary map server implementations. To address this problem, the OGC has developed a non-proprietary Web mapping approach based on open interfaces, encoding and schemas. The WMS specification enables the creation of a network of interoperable map servers from which WMS clients can overlay and build customized maps starting from the client request (geographic bounding box, layers, styles, coordinate reference system, etc).

In Figure 1 a communication pattern among the WMS clients and the WMS Server proposed by the OGC is depicted.

The WMS specifications is an API that allows programmers to add an interoperability layer to several geo-processing systems from different vendors and of different types; the API includes three main components:

- a vocabulary for the request of information,
- a vocabulary for the response to requests,
- a protocol for the exchange of requests and responses.

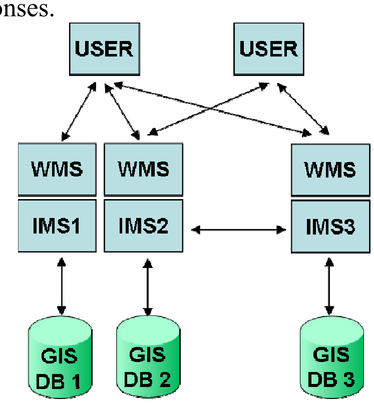

Figure 1 – The WMS Communication (from [2])

2.3 Requirement of Web Map Server OGC-compliant

A WMS is OGC-compliant if it provides three WMS basic operations [6]:

- **GetCapabilities**, returns service-level metadata, which is a description of the service's information content and acceptable request parameters;
- **GetMap**, returns a map image whose geospatial and dimensional parameters are well defined;
- **GetFeatureInfo** (optional), returns information about particular features shown on a map.

We can invoke these operations by utilizing a WMS client in which we must specify a WWW URL whose syntax was defined by OGC. Also, an eXtensible Markup Language (XML) dialect is defined for service-level metadata.

When requesting a map, a client may specify the information to be shown on the map (one or more layers), possibly the styles of those Layers, what portion of the Earth has to be mapped (a bounding box), the projected or geographic coordinate reference system to be used, the desired output format, the output size, background transparency and color and so on.

In order to produce a map, a client invokes *GetCapabilities* to obtain general information about WMS and specific information about the available map layers. The response generated by *GetCapabilities* consists of an XML document. The form of the URL-encoded request is as follows:

http://mapgrid.le.isac.cnr.it/cgi-bin/mapserv?map=/var/www/virtual/mapgrid/puglia/layer-control-query.map&REQUEST=GetCapabilities

Then the client invokes *GetMap* to obtain the desired map. The form of the URL-encoded request is as follows:

http://mapgrid.le.isac.cnr.it/cgi-bin/mapserv?map=/var/www/virtual/mapgrid/puglia/layer-control-query.map&VERSION=1.1.1&REQUEST=GetMap&LAYERS=strade,limiti

When two or more maps are produced with the same bounding box, spatial reference system, and output size, the results can be accurately layered to produce a composite map.

Furthermore, individual map layers can be requested from different servers (or data sources) thus enabling the creation of a network of distributed Map Servers that clients can utilize to build customized maps.

A compliant WMS client does not require any special software viewer other than a Web browser. Finally, through the above mentioned operations a WMS client can:

- list the contents of a map-based catalogue;
- select map layers, viewing regions and scales;
- compose and display maps constructed from data coming from one or more remote servers;
- query the Web for attribute information of a map feature selected from a map displayed in a client and support applications, based on visualization of map data obtained in real time from disparate data sources.

3. Grid-enabled WMS

Starting from a WMS OGC-compliant, in this section we move towards a Grid-enabled WMS highlighting requirements and architecture.

3.1 Requirements

In order to meet important requirements such as compliance, efficiency, security, robustness and flexibility, a Grid-enabled WMS has to face several challenges such as for instance:

Compliance: in order to grant the compliance with the OGC specification, the system must expose at least the three interfaces above mentioned (*GetCapabilities, GetMap, GetFeatureInfo*).

Heterogeneity: in a distributed environment heterogeneity is the rule (different platforms, networks, DBMSs, operating systems, data format, etc.). Moreover, systems which try to bring together programs, services and protocols implemented by different developers cannot communicate unless using common standards. To address this issue we design a cross-DBMS and cross-platform system, paying special attention to portability. We exploit the Service Oriented Approach which in turn leads to higher level of interoperability among the system components.

Security: in a distributed environment, security (confidentiality, integrity etc.) is a critical and complex issue. We address it, performing mutual authentication, and data encryption by means of the Grid Security Infrastructure (GSI) [7]. System-defined authorization callbacks can implement robust authorization policies defining specific roles and privileges.

However, the data encryption is a feature that can be turned on/off depending on the kind of data transfered (some time their are sensible, whereas other ones publicly available).

Efficiency/Robustness: the system has to provide efficient and stable access to distributed geospatial data sources. All of the modules (protocols, libraries, etc.) are written using the C language (to address performance). Furthermore the system offers advanced delivery/storing mechanisms leveraging compression and gridFTP [8] protocol.

3.2 Architecture

In order to satisfy all of the requirements, the proposed architecture (shown in Fig. 2) has been designed using the University of Minnesota mapserver as Map Server [9] to provide an OGC-compliant WMS front-end, the GRelC libraries [10] and Globus Toolkit [11] to achieve secure and efficient access to distributed and heterogeneous information systems representing the various data sources. Let us now go into more details about the most important components depicted in Fig.2.

The Grid-WMS, consists of the following components:

1) the GIS-DBs, i.e., the legacy databases managed by DBMSs (i.e. Oracle [12], PostgreSQL [13], etc.) which contain GIS data; these data sources are the back-end of our system. The DBMSs heterogeneity represents an important issue for higher layers in the architecture.
2) the GRelC Service (GS) [14], which acts as a standard front-end for each GIS data source. This service provides the basic primitives to

transparently get access and interact with different data sources, concealing the back-end heterogeneity and other low level details (it is built on top of the Globus Toolkit, libxml2 and DBMSs proprietary libraries).

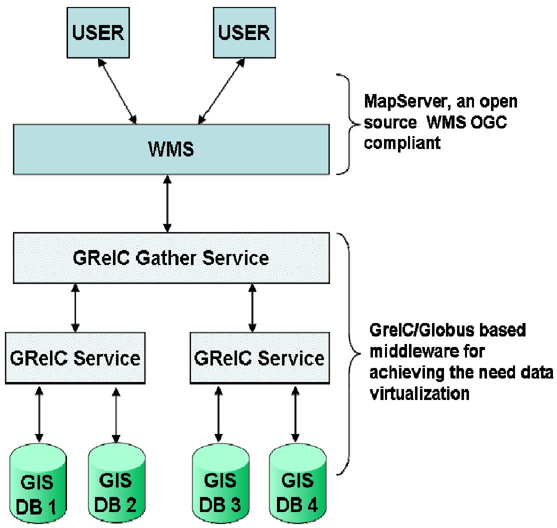

Figure 2 – Grid-enable WMS Architecture overview

To date, the GS module provides (by means of wrappers written by using proprietary libraries) dynamic binding to several physical RDBMSs such as Oracle or PostgreSQL. The security of the connection between DBMSs and GS relies on Socket Secure Layer (SSL). A wrapper for Unix-ODBC [15] data sources allows the GS to also access textual databases (for instance, the GRelC Service provides – for grid applications in the bioinformatics field - an ad hoc wrapper for the Swiss-Prot database). Furthermore, due to the system modularity new database-specific wrappers can be easily written and added to our system. To date, two versions of this service are available: the former leverages a client/server architecture whereas the latter (used in this context) a GSI enabled Web Service (the last one exploits the gSOAP Toolkit [16]; moreover, to guarantee a secure data channel between client and GRelC Service, it uses the Grid Security Infrastructure (GSI) support, available as a gSOAP plug-in [17,18]. From the developer point of view, the GRelC Service provides a rich set of methods/APIs supporting:

- connection management (set properties/open/close connection);
- data manipulation operations;
- advanced delivery mechanisms leveraging an efficient data transport protocol (GridFTP protocol) and compression libraries (Zlib [19]) (Additional information can be found in [20]);
- additional functionalities.

3) The GRelC Gather Service (GGS) [21] which must:
- submit queries to the connected GSs (submission phase);
- gather partial result sets coming from several GSs (leaves node) directly connected with it (retrieving, merging and filtering phases).

Currently, a GSI enabled Web Service version of the GGS is available, exploiting the GSI plug-in for gSOAP. A set of methods/APIs provides the user with a lot of functionalities in order to easily develop applications on top of the GGS.

In the proposed architecture we just considered the GGS, whereas in the future we plan to integrate an Enhanced GGS described in [22] which supports a more scalable infrastructure.

In particular, the WMS exploits the GGS methods for access to the distributed data catalogues. Indeed, the GGS module provides a transparent layer and allows the WMS to work in standard way, such as thorough ODBC connection.

Mapserver connect itself with GGS, and hence can access to vector data sets in their native format, through OGR library [23].

The OGR Simple Features Library is a C++ open source library providing read (and sometimes write) access to a variety of vector file formats. OGR allows mapserver to access spatial and non-spatial tables via ODBC. By default the ODBC searches for GEOMETRY_COLUMNS table. If found it is used to identify the set of spatial tables that should be treated as layers by OGR.

Currently the ODBC OGR driver is read-only, so new features, tables and datasources cannot normally be created by OGR applications. This limitation may be removed in the future.

It is worth noting here that in our system, the query language is the Standard Query Language (SQL) containing GIS extensions, as specified in the *Open GIS Simple Features Specification For SQL* [24] of the OGC. Indeed, as specified in [24], geospatial features collection can be stored as a record in a table in which the non-spatial attributes will be mapped onto columns whose types are drawn from the set of standard ODBC/SQL92 data types. The spatial attributes will be mapped onto columns whose data types are based on features shall be referred to as a feature table. The SQL92 environment with a set of Geometry Types is referred as SQL92 with Geometry Types and it allows a standard SQL schema that supports storage, retrieval, query and updated of simple feature collection via the ODBC API and through geographic operations.

4. Related Work

Several works deal with the use of Grid technologies to develop geospatial data management software architectures and some of which are strictly related to GIS data [25-26].

In [25] the authors describe a prototype in which Grid technologies and OGC Web Services are integrated for evaluating the usability in the geospatial disciplines. The purpose of this project is to test OGC interfaces in the NASA's data environment. Currently, NWGISS consists of five components: a Map Server, a Coverage Server, a Catalog Server, a multi-protocol geoinformation client (MPGC), and a Toolbox. It is a preliminary work that demonstrates the great potential of Grid technologies in this field.

In [26] the authors discuss a research project that integrate the OGC and technologies for making NASA EOSDIS data accessible to Earth science modelling and application communities. In this work they demonstrate how the integration of both OGC and Grid technologies are very promising for application in Earth science in order to provide interoperable sharing of geospatial data, information, knowledge and computational resources.

In [22], a distributed Earth observation system information service is presented. It aims at managing and accessing Earth observation and geospatial heterogeneous data sources, in a grid environment. This work is slightly different because the target is connected with the remotely sensed data (such as radar imagery, ortho-photos, etc.) and the proposed architecture is based on a common metadata schema describing heterogeneous data sources.

In our work, we have been exploiting the Grid technologies to develop Grid-enabled WMS OGC-compliant and we have been focusing on the data access level. Our target is to achieve the needed level of data sources virtualization and scalability, exploiting specific libraries layered on top of the Globus Toolkit.

5. Conclusions and future directions

In this paper we presented the development of a OGC-compliant WMS using Grid technologies. We discussed a secure and controlled access (leveraging the GRelC libraries and Globus Toolkit) on huge quantity of GIS data wide spread on a grid environment . So, end-users, with a single sign-up, securely and transparently get maps whose data are distributed on heterogeneous data source belonging to one o more VOs via distributed queries on the grid information system. This work belongs to a research project, named *MapGrid* [27], which aims to develop a Grid infrastructure for GIS data management.

In the near future, we plan to add new features such as an OGC-compliant Web Coverage Service to support representations of space-varying phenomena that relate a spatio-temporal domain to a (possibly multidimensional) range of properties. It is very important in meteorological and scientific study environments.

Moreover, we plan to migrate our MapGrid architecture towards a WSRF [28]/OGSA [29] specification traducing our services in grid services.

Acknowledgments

This work has been partly supported by the Italian Ministry of Scientific Research and Technology (MIUR) Fund through project no. CNR-ISAC-245-2002.

References

[1] The Open Geospatial Consortium, Inc. URL:[http://www.opengeospatial.org/].

[2] OpenGIS® Web Services Architecture of the Open Geospatial Consortium, Inc. URL: [http://www.opengeospatial.org/docs/03-025.pdf].

[3] I. Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of High Performance Computing Applications, 2001, 15(3): pp.200-222.

[4] I. Foster and C. Kesselman,. The Grid: "Blueprint for a New Computing Infrastructure, Morgan Kaufmann , 1999.

[5] OpenGIS® Web Map Server Cookbook of the Open Geospatial Consortium, Inc. [URL: http://www.ogcnetwork.org/docs/03-050r1.pdf].

[6] OpenGIS® Web Map Service Implementation Specification [URL: http://www.opengeospatial.org/docs/01-068r3.pdf].

[7] I. Foster, C. Kesselmann, G. Tsudik, S. Tuecke, "A security Architecture for Computational Grids", Proceedings of 5th ACM Conference on Computer and Communications Security Conference, pp. 83-92, 1998.

[8] GridFTP Protocol. URL: [http://www-fp.mcs.anl.gov/dsl/GridFTP-Protocol-RFC-Draft.pdf].

[9] The MapServer project. URL:[http://mapserver.gis.umn.edu/].

[10] G. Aloisio, M. Cafaro, S. Fiore, M. Mirto, "The GRelC Library: A Basic Pillar in the Grid Relational Catalog Architecture", Proceedings of Information Technology Coding and Computing (ITCC), April 5 to 7, 2004, Las Vegas, Nevada.

[11] The Globus Project, URL: [http://www.globus.org/].

[12] Oracle Grid Computing Technologies URL: [http://otn.oracle.com/products/oracle9i/grid_computing/index.html].

[13] Postgresql, URL: [www.postgresql.org]

[14] G. Aloisio, M. Cafaro, S. Fiore, M. Mirto, "The GRelC Project: Towards GRID-DBMS", Proceedings of Parallel and Distributed Computing and Networks (PDCN) IASTED, Innsbruck (Austria) February 17-19, 2004.

[15] UnixODBC, URL : [www.unixodbc.org]

[16] R.A. Van Engelen, K.A. Gallivan, "The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks.", Proceedings of IEEE CCGrid Conference, May 2002, Berlin, pp. 128-135.

[17] G. Aloisio, M. Cafaro, D. Lezzi and R.A. Van Engelen, "Secure Web Services with Globus GSI and gSOAP", Proceedings of Euro-Par 2003, Lecture Notes in Computer Science, Springer-Verlag, N. 2790, pp. 421--426, 2003

[18] G. Aloisio, M. Cafaro, I. Epicoco, D Lezzi and R.A. Van Engelen, "The GSI plug-in for gSOAP: Enhanced Security, Performance, and Reliability", submitted to ITCC 2005.

[19] Zlib, URL: [http://www.gzip.org/zlib]

[20] G. Aloisio, M. Cafaro, S. Fiore, M. Mirto, "Advanced Delivery Mechanisms in the GRelC Project" ACM Proceedings of 2nd International Workshop on Middleware for Grid Computing (MGC 2004), October 18 2004, Toronto, Ontario Canada.

[21] M. Mirto, G. Aloisio, M. Cafaro, S. Fiore, "A Gather Service in a Health Grid Environment", Proceedings of Medicon 2004, July 31 - August 05, Island of Ischia, Naples, Italy.

[22] G. Aloisio, M. Cafaro, S. Fiore, G. Quarta, "A Grid-Based Architecture for Earth Observation Data Access", to appear in Proceeding of the 20th Annual ACM Symposium on Applied Computing, March 13 -17, 2005, Santa Fe, New Mexico.

[23] OGR Simple Feature Library [URL:http://www.remotesensing.org/ogr/].

[24] OpenGIS® Simple Features Specification for Sql URL:[http://www.geo.unizh.ch/oai/spatialdb/ogcSFforSQL_99-049.pdf].

[25] Liping Di, Aijun Chen, Wenli Yang and Peisheng Zhao, "The Integration of Grid Technology with OGC Web Services (OWS) in NWGISS for NASA EOS Data". HPDC12 & GGF8 on June 24 - 27, 2003, Seattle, USA. URL:[**http://www.cs.vu.nl/ggf/apps-rg/meetings/ggf8/di.pdf**].

[26] Liping Di, "The Development of Geospatially-enabled Grid Technology for Earth Science Applications". Proceeding of the fourth annual Earth Science Technology Conference, June 22 to 24, 2004 Palo Alto, California, USA. URL: [http://www.esto.nasa.gov/conferences/estc2004/papers/a3p3.pdf]

[27] The MapGrid Project [URL:http://mapgrid.le.isac.cnr.it].

[28] The Globus Alliance - The WS-Resource Framework [URL: http://www.globus.org/wsrf/].

[29] The Globus Alliance - Open Grid Services Architecture [URL:http://www.globus.org/ogsa/]