

RESEARCH ARTICLE

# Testing OGC Web Feature and Coverage Service performance: Towards efficient delivery of geospatial data

Gregory Giuliani<sup>1,2,3</sup>, Alan Dubois<sup>4,5</sup>, and Pierre Lacroix<sup>1,3</sup>

<sup>1</sup>Institute for Environmental Sciences, University of Geneva, Geneva, Switzerland

<sup>2</sup>Division of Early Warning and Assessment, United Nations Environment Programme, Geneva, Switzerland

<sup>3</sup>Forel Institute, University of Geneva, Geneva, Switzerland

<sup>4</sup>Institute of Environmental Sciences, University of Geneva, Geneva, Switzerland

<sup>5</sup>HEPIA, University of Applied Sciences Western Switzerland, Geneva, Switzerland

*Received: October 4, 2012; returned: November 28, 2012; revised: January 18, 2013; accepted: March 4, 2013.*

---

**Abstract:** OGC Web Feature Service (WFS) and Web Coverage Service (WCS) specifications allow interoperable access to distributed geospatial data made available through spatial data infrastructures (SDIs). To ensure that a service is sufficiently responsive to fulfill users' expectations and requirements, performance of services must be measured and monitored to track latencies, bottlenecks, and errors that may negatively influence its overall quality. Despite the importance of data retrieval and access, little research has been published on this topic and mostly concentrates on the usability of services when integrating distributed data sources. Considering these issues, this paper extends and validates the FOSS4G approach to measure the server-side performance of different WFS and WCS services provided by various software implementations; and provides guidance to data providers looking to improve the quality of their services. Our results show that performance of tested implementations is generally satisfactory and memory tuning/data and storage optimization are essential to handle increased efficiency and reliability of services.

**Keywords:** quality of service, performance, web feature service, web coverage service, benchmarking, SDI, interoperability

---

## 1 Introduction

Facilitating access and making geospatial data interoperable are recognized as important factors in determining the future success of spatial data infrastructures (SDI) [5, 7, 33, 48]. Indeed, geospatial data is extensively used in various domains such as environmental monitoring, disaster management, and decision-making processes, requiring seamless integration in geographic information systems (GIS) and sharing of data across organizations and providers [50]. Geospatial data can be a shared resource maintained continuously and made accessible for different purposes in both the public and private sectors [49]. SDIs aim to support sharing and exchange of geospatial data across institutional, regional, and/or national borders, mostly through service-oriented architecture (SOA) principles [24, 53]. SOAs are defined as an architectural approach where standardized interfaces give access to functionalities as a set of independent, interoperable, loosely-coupled, distributed services that can be reused. The rapid evolution of web-based communication technology allows for easier access to distributed geospatial data sources and related services, and results in an increasing use of geospatial data in many areas [46].

To improve interoperability within the GIS community, the Open Geospatial Consortium (OGC) introduced various standards specifications covering data sharing, retrieval, processing, content, visualization, and cataloging [9, 11]. These standards are: Web Map Service (WMS) [38], Web Feature Service (WFS) [37], Web Coverage Service (WCS) [39], Catalog Service for the Web (CSW) [40], Web Processing Service (WPS) [42], and Web Coverage Processing Service (WCPS). These standards are built around web services technologies. A web service is a stateless application containing a collection of operations that is accessible through the web and that users can discover and invoke [50]. In other words, interoperable web services aim to provide users with the functionality they need independently of the specific computing platform and programming language. Moreover, the composability and reusability of standard components offered by web services allow for building applications specific to the needs of domain and/or community of users, overcoming disadvantages and inflexibilities of monolithic GIS [53]. OGC web services (OWS) are based on XML (extended markup language) to encode calls and HTTP (hypertext transfer protocol) for communicating.

Several studies [8, 13, 21] have shown that projects that have adopted and implemented geospatial interoperability standards saved around 25% of their time, compared to those who rely on proprietary standards. These reports also showed that using geospatial interoperability standards lowers the transaction costs for sharing data and information. The fact that exchange of data and information is performed on standardized interfaces enhances flexibility and adaptability of projects over time. These benefits provided by geospatial interoperability standards probably contribute to their growing success. The recently published INSPIRE State of Play 2010 [56] highlights that European countries are increasingly focusing their attention on interoperability issues.

View services appear to be very well developed and download services have recently begun to emerge. Consequently, we assume that WMS [38] is now well accepted among different communities to make their data viewable in an interoperable manner, but this is not yet the case for download using WFS [37] and WCS [39] standards. Only in a few cases have data been made available through these standards [4, 10, 12, 14, 27].

Currently SDIs are mainly concerned with data retrieval, data processing, and data visualization [2, 52], allowing data discoverability using Catalog Service for the Web (CSW)

[40], retrieval with WFS and WCS, processing and analysis through WPS [42] and WCPS, and visualization through WMS. Initiatives at the regional and global scale such as INSPIRE (Infrastructure for Spatial Information in the European Community) [17] and the Global Earth Observation System of Systems (GEOSS) [25] are good examples of what can be done in terms of geospatial interoperability standards. These initiatives are seeking to relate environmental data providers to the widest possible audience, with the objective to enhance and improve decision-making. In these two examples, as in many others, OWS are key enablers providing interoperable access to data in an efficient and timely manner. However, having interoperable access to data is only a first step towards data integration. To satisfy users' expectations, it is obviously also required to have services of sufficient quality, especially in terms of performance [36]. To ensure that a service is sufficiently responsive to fulfill users' needs and requirements, performance of a given service must be measured and monitored. This includes tracking latencies, bottlenecks, and errors that may negatively influence its overall quality.

Despite the importance of data retrieval and access, little research has been conducted on benchmarking and evaluating the quality of WFS and WCS services [29, 53, 55, 57, 59]. These studies mostly concentrate on the usability of OGC Web Services (OWS) when integrating distributed data sources, but provide neither a framework nor guidance to measure the performance of data services. Moreover, the only published papers specifically on this topic examine the WMS [26] and the WFS [3] specifications, to our knowledge. Consequently, a framework to assess the usability and performance of download services through a set of quantitative measures is required that allows for the quantification, repeatability, comparability, and understandability of results. Based on these considerations the aims of this paper are to:

- (1) extend the FOSS4G approach for measuring the server-side performance of different WFS and WCS services provided by two widely-used software implementations;
- (2) provide some guidance to data providers aiming at improving the quality of their services; and
- (3) encourage other users to contribute to completing this benchmark by setting up other scenarios and performing other tests (e.g., client-side, different parameters).

All developed material (benchmark scripts, data, and procedures) are freely available by contacting the authors.

## 2 Geospatial data interoperability

Data accessibility, availability and compatibility are among the most frequent difficulties encountered while preparing strategic environmental assessments (SEA) and environment impact assessments (EIA) in Europe [5]. Moreover, it is estimated that up to 50% of users' time is spent in data discovery and transformation to make data compatible and harmonized [32, 58]. These authors emphasize various reasons leading to such problems, including:

- (1) geospatial data is voluminous;
- (2) geospatial data is geographically distributed (e.g., various data collector/providers around the world);
- (3) geospatial data is heterogeneous (e.g., formats, schemas, coordinate systems);

- (4) geospatial data is complex (e.g., geometries, relationships, attributes); and
- (5) institutional arrangements and policies (e.g., copyright) are lacking or restricting access to (meta)data.

All these factors may influence the way data providers store, publish, and deliver geospatial data. Hence, making data interoperable and improving the quality of this interoperability can potentially enhance the above-mentioned situation, allowing data users to spend more time in data analysis than in data discovery, which would enable more people to benefit from using geospatial data. To be fully interoperable a system must be syntactically, semantically, and schematically interoperable. However, OGC specifications concentrate mostly on syntactical interoperability allowing exchanging data with other components or systems. To reach semantic and schematic interoperability, both components of a system must agree on a common reference model providing the possibility to interpret accurately, unambiguously, and meaningfully the information exchanged. These levels of interoperability are important issues that have been extensively studied [22, 28, 31, 60]. Hence, to enable effective and syntactically interoperable access to geospatial data, OGC WFS (for vector data) and WCS (for raster data) specifications are essential components and a prerequisite for testing performance of different software implementations.

The OGC WFS specification [37] defines an interface for accessing feature-based geospatial data, commonly known as vector (e.g., rivers, country borders, and cities), encoded in geography markup language (GML) [50, 58]. A WFS interface is invoked through a URL and can perform a certain number of operations (e.g., retrieving, creating, deleting, updating) allowing a client to handle data [51].

The OGC WCS specification [39] defines a web interface allowing a client to access raster data sets. A raster data set represents a matrix of cells in continuous space organized in rows and columns where each cell contains a value. Thus WCS services give access to different types of space and/or temporal-varying data such as a digital elevation model (DEM) or remote sensing imagery. WCS delivers raw data and does not have transactional capabilities [50, 58].

### 3 Quality of service (QoS)

Satisfaction of users is a major objective of any service provider. However, even if a system is fully interoperable (e.g., syntax, semantics, schema), this satisfaction would not necessarily be guaranteed. Evaluating and predicting users' satisfaction is a complex task because quality can be based on quantitative and qualitative attributes [29]. The overall performance of the service is dependent on the combination of server, network, and client components. In particular, the overall performance will be affected by the slowest component that needs to be identified first. Quality can be interpreted both as a measure that represents accessibility/performance and also as a perceived quality of a service [53]. Despite the importance of qualitative perceptions (e.g., good description, easy access to a service) it is undeniable that unresponsiveness or slowness of a service will negatively affect users' satisfaction [34]. With the expected increasing diffusion of OWS, quality of service (QoS) will be an important factor to distinguish between reliable services and others. Therefore, QoS can be referred as a set of measurable attributes related to the individual behavior of a web service that guarantees sufficient availability and performance of a service [53]. This excludes all qualitative perceptions and concentrates on technical aspects

that are easily measurable and replicable under different conditions or environments. A detailed overview of testing methods both on the server- and client-sides is given in [26]. These authors recognize the importance of testing to evaluate characteristics that identify measurable or quantifiable parameters.

INSPIRE is a legal framework (that entered into force in May 2007 and will be fully operational by 2019) for the establishment of a European Union SDI. The directive provides five sets of implementing rules that set out how the various elements of the system (metadata, data sharing, data specification, network services, monitoring, and reporting) will operate, and ensures that spatial data infrastructures of the member states are compatible and usable in a community and trans-boundary context [6].

To ensure sufficient availability and accessibility to data, the European Commission [19,20] has adopted regulations on network services and specifically on *download* services. Annex I of the regulation on download services sets out QoS criteria and corresponding values that any service must achieve:

- (1) *Performance* refers to how fast a request can be completed. The response time for sending the initial response shall be a maximum of 30 seconds in a normal situation (e.g., periods out of peak load), and shall maintain a sustained response greater than 0.5MB (or 500 spatial objects) per second.
- (2) *Capacity* is the limit of simultaneous requests that a service can handle with guaranteed performance [26]. For a download service, the minimum number of served simultaneous services shall be 10 requests per second. The number of requests processed in parallel may be limited to 50.
- (3) *Availability* measures the probability that a network service is available and shall be 99% at any given time.

These measurements are important as they allow easy quantification, repeatability, comparability, and understandability [26,35]. QoS criteria can be measured either on the server-side (e.g., number of visits, session duration, and average time per page) or the client-side (e.g., load testing, capacity testing). In general, the tested service is considered as a black box receiving requests and providing responses without considering software implementation at all.

## 4 Methodology of testing

Currently, even if regional initiatives such as INSPIRE are highlighting the importance of QoS there is no commonly agreed framework to measure performance of download services. Consequently, it is important to measure performance with approaches that [18] are:

- (1) sufficiently generic to be independent of infrastructure and application design;
- (2) independent of the communication between the service and client; and
- (3) based on request-response pairs and avoid complex transactions.

Our proposed approach is based on the one developed by the FOSS4G (Free and Open Source for Geospatial) community<sup>1</sup> to test WMS services. The aim is to extend this open approach to WFS and WCS services by testing various OWS implementations on a common set of geospatial data located on the same platform. Obviously, to fulfill the requirements

---

<sup>1</sup>[http://wiki.osgeo.org/wiki/FOSS4G\\_Benchmark](http://wiki.osgeo.org/wiki/FOSS4G_Benchmark)

mentioned previously, it is important that these tests are based on realistic usage scenarios. For that purpose, load testing (also known as stress testing) is generally performed simulating multiple concurrent queries to a service. This approach allows analyzing the behavior of a service under various conditions that are beyond normal usage patterns [26].

In our case, WMS testing was achieved through the use of the `GetMap` query, which returns to a client a raster map of selected layers based on parameters including map extent, coordinate system, width and height of the map, and image format. The query was used for the purpose of calibrating our resulting curves with those of FOSS4G<sup>2</sup> 2009 shootout. Regarding WFS and WCS testing, the use of the `GetFeature` (returns a GML result set with geometry and feature attributes) and `GetCoverage` (returns a response to a client that either contains or references the requested coverage) queries placed the emphasis on actual access to the data.

Our general methodology was to take one case as a basic case, and to vary one by one a given set of parameters related to data characteristics (geometry type, data resolution, complexity and number of attributes, field indexation, input, and output formats) and computer memory. Due to their small impact on WFS robustness [3], complexity and completeness of metadata were not tested. Random screen-sized requests were performed, on commonly used scales and extents corresponding to the bounding box of various countries. Concerning the WCS, the proposed methodology focuses on 2D data delivery. However, the WCS specification defines an interface to serve  $n$ -dimensional coverages that are important when going beyond maps into 3D time series or voxel data (e.g., 4D/5D climate data). To represent realistic usage scenarios, two groups of users were simulated: (1) a small team (e.g., people working within the same laboratory) with a workload varying from 1 to 16 concurrent requests, and (2) a larger set of 150 users simultaneously retrieving data such as in an emergency situation (e.g., after Haiti or Japan earthquakes). Each test was performed three times but only the last measure, considered as the most stable one, was retained.

#### 4.1 Technical architecture and software

One of the aims of the proposed approach is to provide some guidance to service providers who are looking to share their data in an interoperable manner. In particular, vast amount of geospatial data is available within institutions, such as universities, research centers, that do not have means and/or do not want to develop complex infrastructures. Consequently, the proposed architecture to serve OWS must be simple, reflecting these conditions and showing that with little effort it is feasible to provide download services of sufficient quality. Therefore, the measurements will reflect performance of the server-side component of the architecture. The technical architecture (Figure 1) is based on a three-tiered model: (1) a data layer, (2) a service layer, and (3) a client layer.

In the data layer, vector and raster data can be stored directly either in a database or in file directories. PostGIS 1.5.1<sup>3</sup> and ArcSDE 9.3.1<sup>4</sup> were used with PostgreSQL 8.4.4<sup>5</sup> database management system (DBMS) to support geospatial data. Interoperable access to data is provided through the service layer using GeoServer 2.1.1<sup>6</sup> and ArcGIS Server 9.3.1<sup>7</sup>,

<sup>2</sup>[http://wiki.osgeo.org/wiki/Benchmarking\\_2009](http://wiki.osgeo.org/wiki/Benchmarking_2009)

<sup>3</sup><http://postgis.refrations.net/>

<sup>4</sup><http://www.esri.com/software/arcgis/arcscde/>

<sup>5</sup><http://www.postgresql.org/>

<sup>6</sup><http://www.geoserver.org>

<sup>7</sup><http://www.esri.com/software/arcgis/arcgisserver/>



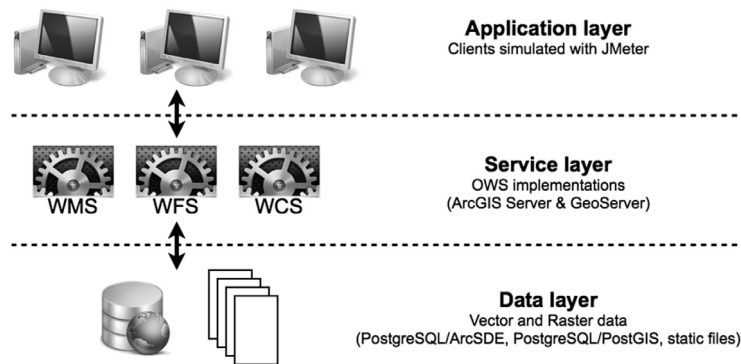


Figure 1: Architecture used for testing different OWS implementations.

which are able to manage different data sources such as PostGIS, ArcSDE, or files in folders. Even though there are many available solutions to publish WFS and WCS services (e.g., MapServer, Deegree) we decided to focus on ArcGIS Server and GeoServer because they are widely used in the GIS community and they allowed us to test “native” solutions (e.g., ArcGIS Server/ArcSDE, GeoServer/PostGIS) as well as “crossed” solutions (e.g., ArcGIS Server/PostGIS, GeoServer/ArcSDE).

The client requesting services were simulated using JMeter<sup>8</sup>, a Java open source software designed to test servers, networks, and services performance under various load conditions on static and dynamic resources (e.g., files, servlets, scripts, databases and queries, FTP servers). JMeter was originally designed to test web applications but has since expanded to other testing functions using plugins. Despite the fact that this is a desktop application, it is important to note that JMeter is not a browser; it does not execute Javascript nor do HTML rendering. Its usage is very simple, requiring only the URL querying a specific service. Users can design their testing plan using variables, counters, logs, or other parameters to be tested. A JMeter test then corresponds to a script that can be easily executed and maintained through the JMeter graphical user interface. Once tests have been executed, performance logs can be accessed through text files providing various types of information such as number of requests, number of threads, or execution time. This information can be used to draw results as graphs or tables. JMeter scripts provided by the FOSS4G WMS 2009 benchmark have been used, extended, and adapted to test our WFS and WCS instances. Components and characteristics of the test environment are summarized in Table 1.

## 4.2 Testing scenarios

Testing scenarios were developed to simulate users’ behavior when downloading data through WFS and WCS services. The testing scenarios are based on:

- (1) common vector and raster input and output formats: shapefile, ESRI geodatabase, ArcSDE, GML, TIFF, GeoTIFF, and JPEG;

<sup>8</sup><http://jakarta.apache.org/jmeter/>

Computer	Operating system	Processor	RAM	Hard disk	Software
Database management system	Suse Linux Enterprise Server 10, 32 bits	2 × Quad Core Xeon E5420 2.5Ghz, 2×6MB cache, 1333MHz FSB	8GB	4 × 1TB 7200rpm, SATA	PostgreSQL 8.3.0, ArcSDE 9.3.1
Geoservices server	Windows server 2003, Enterprise Edition, 32 bits, SP2	1 × Quad Core Xeon X3323, 2.5GHz, 2×6MB cache, 1333MHz FSB	4GB	2 × 1TB, 7200rpm, Near Line SAS,	ArcGIS Server 9.3.1 .NET enterprise SP2, Geoserver 2.1.1 for windows, MS4W 3.0 including Mapserver 5.6.3
Client requesting geoservices	Ubuntu 10.04	Dual Core E6600 2.4GHz, 4MB cache	2GB	1 × 148GB, 7200rpm	JMeter 2.3.4

Table 1: Characteristics of the test environment. All network interfaces between computers were based on 1GB LAN connections.

- (2) popular data sets, such as the Blue Marble New Generation (BMNG) [54] and Global Lakes and Wetlands Database Level 2 (GLWD-2) [30], chosen for their potential use in many fields and their availability at global scale in WGS 84 spatial reference;
- (3) widely used platforms implementing OWS: ArcGIS Server and GeoServer.

Various conditions were defined according to (a) our daily experience using OWS; and (b) past experiments made during the development of the PREVIEW geoportal [23]. The different testing scenarios developed to test WFS services are summarized in Table 2 and those concerning WCS in Table 3. Each scenario is tested both under ArcGIS Server and GeoServer platforms.

Further tests were also performed on memory configuration by varying the allowed memory and by comparing performance of each of the three runs of case #1 with varying memory. An ArcGIS Server endpoint is for WFS:

```
http://<server>/arcgis/services/<service>/MapServer/  
WFSserver?
```

and for WCS:

```
http://<server>/arcgis/services/<service>/MapServer/  
WCSserver?
```

while a GeoServer endpoint corresponds to:

```
<server>/geoserver/ows?
```

As an example, the following WFS request executed under ArcGIS Server allows one to retrieve all rivers of Switzerland (i.e., “cntry\_name” field equal to “Switzerland”) present in the data set “WFS\_GDB:river.” This query uses the OGC filter encoding specification capabilities [37] to extract data based on their attributes:



Case study for WFS testing	Tested parameter	Input format	Input geometry	Input attributes
1	None	ArcSDE	Polygons	Base attributes
2	Input format	Shapefile	Same as case #1	Same as case #1
3	Input format	ESRI FileGeodatabase	Same as case #1	Same as case #1
4	Input geometry	Same as case #1	Polylines	Same as case #1
5	Input geometry	Same as case #1	Points	Same as case #1
6	Complexity and number of attributes	Same as case #1	Same as case #1	No attributes
7	Complexity and number of attributes	Same as case #1	Same as case #1	Additional attributes: text 255; five 8-byte double-precision columns
8	Attribute indexation	Same as case #1	Same as case #1	Indexation of the id field

Table 2: Summary of the WFS testing scenarios. Case #1 is the “base case.”

Case study for WFS testing	Tested parameter	Input format	Output format	Output resolution
1	None	Tiled images in TIFF	TIFF	0.008 decimal degrees (DD)
2	Input format	ESRI File Geodatabase Raster Catalog	Same as case #1	Same as case #1
3	Input format	Case #1 files served by ESRI Image Server	Same as case #1	Same as case #1
4	Output format	Same as case #1	GeoTIFF	Same as case #1
5	Output format	Same as case #1	JPEG	Same as case #1
6	Amount of data	Same as case #1	Same as case #1	Half of the base case resolution
7	Amount of data	Same as case #1	Same as case #1	Case #1 resolution divided by 4

Table 3: Summary of the WCS testing scenarios. Case #1 is the “base case.”

```

service=WFS&styles=&request=GetFeature&version=1.1.0&srs=
EPSG:4326&typename=WFS_GDB:river&Filter=<ogc:Filter><ogc:
PropertyIsEqualTo><ogc:PropertyName>cntry_name</ogc:
PropertyName><ogc:Literal>Switzerland</ogc:Literal></ogc:
PropertyIsEqualTo></ogc:Filter>

```

In GeoServer, the same request shows differences (e.g., “ogc” term mandatory in ArcGIS) in the syntax for the expression of the filter, which already highlights a potential problem in terms of interoperability and universality of semantic:

```
service=WFS&styles=&request=GetFeature&version=1.1.0&srs=
EPSG:4326&typename=owsbenchmark:AQ.SDE.river&filter=
<Filter><PropertyIsEqualTo><PropertyName>cntry_
name</PropertyName><Literal>Switzerland</Literal>
</PropertyIsEqualTo></Filter>
```

Concerning the WCS request, the base case corresponds to the following parameters:

```
SERVICE=WCS&VERSION=1.1.1&REQUEST=GetCoverage&COVERAGE=
1&CRS=EPSG:4326&BBOX=0,0,40,20&WIDTH=9600&HEIGHT=
4800&FORMAT=tiff
```

### 4.3 Data sets

To fit the GIS community’s common users practices, we have selected widely used data sets with a global extent and a large number of features (e.g., attributes, resolution, formats). Therefore vector data sets (used for WFS tests) were extracted from:

- (1) the GLWD-2 as a polygon data set. This very popular data set, used by numerous governmental and non-governmental organizations, contains about 250,000 natural lakes and human-made reservoirs, and 8,500 rivers—respectively 4,000,000 and 350,000 vertices. By default, the polygonal layer includes a dozen textual and numeric attributes.
- (2) the European Space Agency (ESA-ESRIN) World Fires Atlas Program (ATSR) as a point data set. This represents an estimation of fire events around the world between 1997 and 2008, and is available in a compiled form on the PREVIEW Global Risk Data Platform [23]. This data set is composed of around 1,000,090 points.

For the WCS testing, the raster data set we selected as a base case was the BMNG. This data set is provided into 240 GeoTIFF tiles each of 12MB and 15 decimal degrees (DD), spatially referenced on the WGS 84 ellipsoid, with a 0.008DD pixel size, for a total of 2.87GB of data. The BMNG is traditionally used not only in education and research, but also often as a base map, and has a relatively high spatial resolution.

## 5 Results

The tests performed with JMeter consist of sending requests for data with random geographical extent to the different services and measuring the response time. The performance indicator is expressed in data served per second in the case of WFS and in map images served per second in the case of WCS. Different load conditions ranging from 1 to 40 simultaneous threads allow evaluating the server response. Vector data sets were used for testing WMS and WFS, and raster data sets for WCS.

## 5.1 WMS

First, it is important to validate the proposed architecture by comparing results obtained in the FOSS4G Benchmark with the system used in the experiment. For that, the same data set storing edge polylines (roads, rivers) for Texas (shapefile with spatial index) was used.

Threads [map image /s]	Requests	Mapserver CGI SHP	MapServer FCGI SHP	MS4W CGI SHP	MS4W FCGI SHP
1	100	<i>5.0</i>	<i>8.2</i>	3.6	5.7
10	200	<i>11.0</i>	<i>17.1</i>	12.2	17.5
20	400	<i>11.0</i>	<i>16.8</i>	11.0	16.9
40	800	<i>11.5</i>	<i>17.6</i>	11.5	17.3

Table 4: WMS results comparison. FOSS4G benchmark 2009 in italic. CGI: Common Gateway Interface, SHP: shapefile, FCGI: Fast CGI, MS4W: MapServer for Windows.

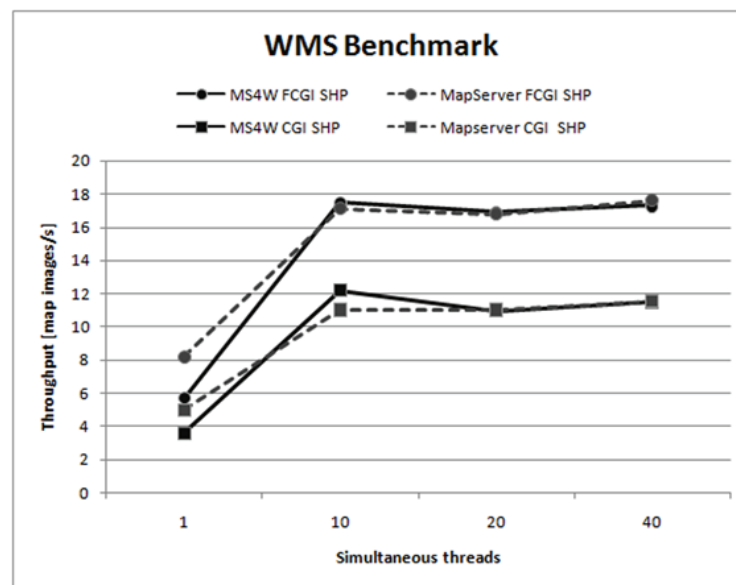


Figure 2: WMS results under various conditions (OWS implementation and file formats).

The performance tests show that the results for shapefiles from FOSS4G WMS benchmark 2009 are similar for Mapserver CGI and FCGI on the Linux platform and for Mapserver CGI and FCGI on Windows<sup>9</sup> (Figure 2, Table 4). This means that the infrastructure used for the benchmark is comparable in terms of performance.

<sup>9</sup>Package Mapserver for Windows - MS4W 3.0 <http://www.maptools.org/ms4w> containing Apache 2.2.15, Mapserver 5.6.3

## 5.2 WFS

WFS 1.1.0 services have been tested through different scenarios on an ArcGIS Server and Geoserver including different data sets (e.g., points and polygons), stored in different formats: ArcSDE, Shapefile, ESRI FGDB, and PostgreSQL/PostGIS.

The fire points data set (in shapefile format) containing about 1,090,000 features for a size of 99MB is about four times slower than the lake data set (shapefile of 117MB) containing about 250,000 polygons and 4,000,000 vertices. This means that the relationship between the storage size of a data set and the performance is not straightforward. “The bigger in file size the slower” is not always true, as performance also depends on the number of features.

On ArcGIS Server, using the “lake polygon” data set, performance was only slightly affected by the storage format, with a small advantage for the ESRI file geodatabase format (Figure 3, Table 5).

[Data /s] Threads	Requests	SDE Fires point	SHP Lake polygon	SDE Lake polygon	FGDB Lake polygon
1	100	0.5	1.7	1.8	1.8
2	200	0.8	3.8	4.4	4.5
4	400	1	4.9	4.1	5.1
8	800	1.1	4.7	4.5	5.3
16	1600	1.1	5	4.6	5.3

Table 5: WFS ArcGIS Server tests results. Refer to Table 4 for acronyms.

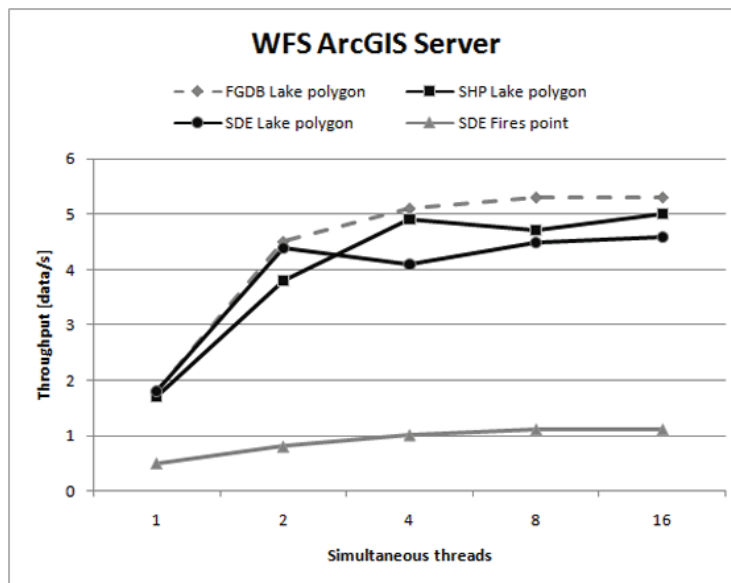


Figure 3: WFS ArcGIS Server tests results.

Compared to ArcGIS Server, Geoserver WFS runs are about three times faster. The polygon data set representing lakes is more efficient when stored in PostGIS. A small decrease

in performance is observed when the number of simultaneous threads exceeds 4, but performance remains good (Figures 4 and 5, Table 6). This phenomenon may come from a disk access bottleneck. An increasing number of service errors have been observed when the simultaneous threads increase.

Threads [Data /s]	Requests	SDE Fires point	SHP Lake polygon	SDE Lake polygon	FGDB Lake polygon
1	100	1.4	6.5	5.1	8.2
2	200	2.5	12.1	9.8	14.7
4	400	3.2	13.5	14	17
8	800	3.5	13.2	14.8	16.9
16	1600	3.3	12.6	13.9	16

Table 6: WFS Geoserver tests results.

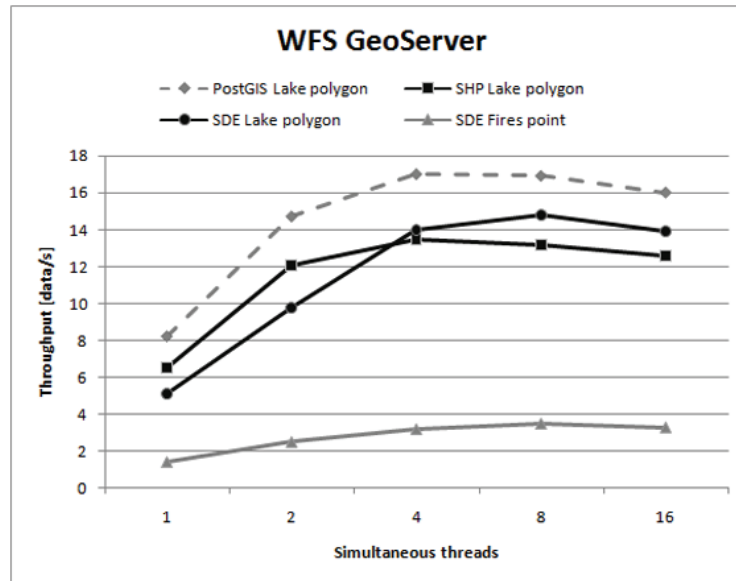


Figure 4: WFS Geoserver test results.

### 5.2.1 Comparison on memory

Comparative tests on memory show that the performance of WFS depends on memory configuration. The variation is significant with ArcGIS Server: by allowing 4 instances instead of 2 (which is the default value), performance improved by about 35% with 4 or more simultaneous threads. Because each separate process uses the same amount of RAM, giving more RAM allow more processes to run independently. With GeoServer, memory configuration shows ambiguous results on WFS performance. Adding 25% of memory, results in gains that are smaller than the variation between the different runs of measures. However, GeoServer appears more stable in term of throughput. The three runs of each

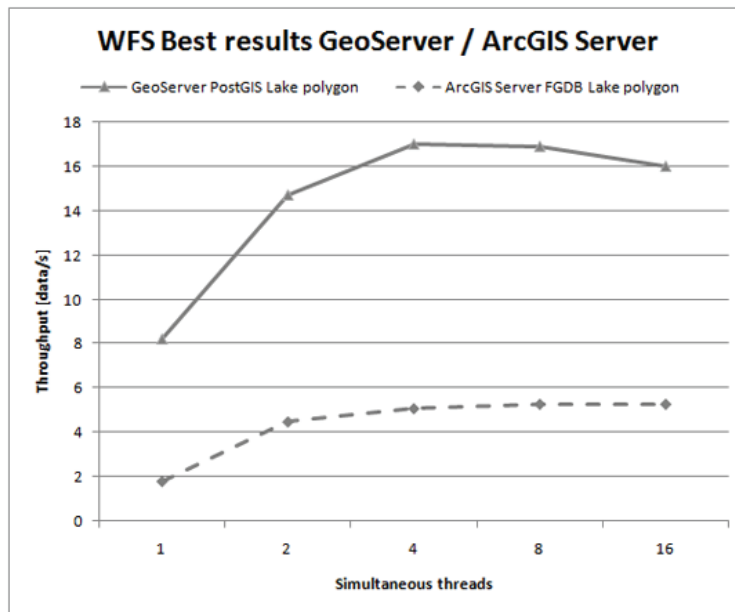


Figure 5: Comparison of best result obtained on each software implementation: GeoServer and ArcGIS Server.

of the previous tests on memory generate comparable throughputs. Variations in performance range from 6% to 9%. First runs are slower than the two others with an increasing number of simultaneous requests.

### 5.3 WCS

The performance of WCS 1.1.1 is tightly linked to the size and resolution of the data set. The BMNG 8-bit color image was stored in low-resolution  $3600 \times 1800$  pixels and high-resolution. Medium resolution was produced by directly resampling the high-resolution image through the WCS instance and requesting a smaller number of pixels in height and width. Different storage options have been tested for the high resolution image consisting of (1) one large GeoTIFF image  $45,688 \times 22,509$  pixels (2.87GB), and (2) 240 tiles of GeoTIFF images  $1876 \times 1876$  pixels (10MB each tile).

Setting up a responsive WCS instance involves efficiently accessing the raster data set. On the ArcGIS Server platform, the WCS is faster when the data set is stored in ESRI file geodatabase raster than in flat TIFF file (Figure 6, Table 7). This is true for low resolution, although in medium and high resolution the file geodatabase is approximately the same as the flat TIFF file. One hypothesis to explain this pattern is that heavy load, the server is limited by other factors such as CPU or memory rather than data access. Overall, the best result are obtained with ArcGIS Image Server with TIFF tiled in high resolution. This result is explained by the overview images created automatically when the ArcGIS Image Service is built. These overview images speed up the service for the WCS requests that cover a large area at small scale.



Threads [Images /s]	Requests	Data (TIFF format)									TIFF tiled	
		AGS lr TIFF	AGS lr FGDB	GS lr TIFF	AGS mr TIFF	AGS mr FGDB	GS mr TIFF	GSwin mr Geo- TIFF	AGS hr TIFF	AGS hr FGDB TIFF	AGS-IS hr Geo- TIFF	GSwin hr Geo- TIFF
1	100	1	1.5	2.2	0.3	0.3	0.8	0.01	0.08	0.1	0.9	0.11
2	200	0.9	1.3	2.3	0.2	0.2	0.2	0.01	0.07	0.1	1.5	0.02
4	400	0.9	1.3	1.8	0.3	0.3	0.1	-	0.11	0.1	1.7	0.04
8	800	0.9	1.4	1.5	0.3	0.3	-	-	0.14	0.1	1.8	0.02
16	1600	0.9	1.4	1.7	0.4	0.4	-	-	0.23	0.2	1.8	-

Table 7: WCS Geoserver (GS) and ArcGIS Server (AGS) tests results. An ArcGIS Server Image Service (IS) has also been tested. Tests have been executed on Linux and Windows (win) operating systems. lr: low-resolution, mr: medium-resolution, hr: high-resolution.

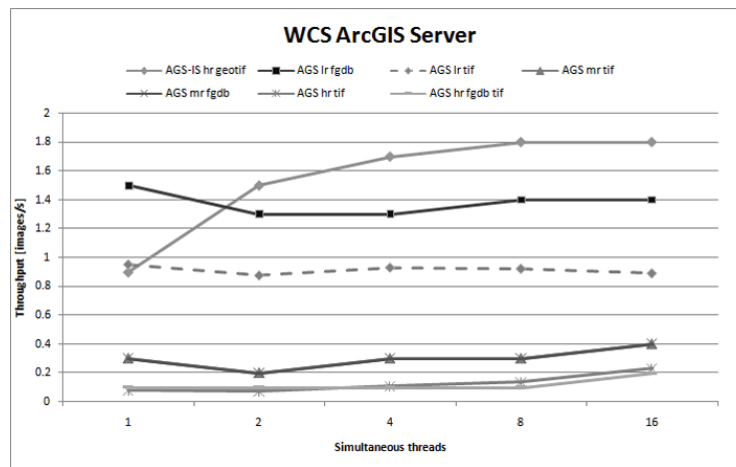


Figure 6: WCS ArcGIS Server tests results.

Geoserver has been tested on Linux with Tomcat and on Windows with Jetty. Tomcat and Jetty are open-source HTTP servers and Java Servlet containers. These two applications provide a pure Java web server environment as well as services such as load balancing, data services, and security. It seems that GeoServer performs faster on Linux. When the load on the server increases in terms of simultaneous threads and with higher resolution, the number of service errors increases and the throughput decreases to very small values. It takes more than one minute to produce each image when the WCS server is overloaded (Figure 7).

## 6 Discussion

Geospatial data is valuable in various domains, particularly in multi-disciplinary research activities that require integrating different data sets from different sources and in different formats. Therefore, having this data readily available and easily accessible is a key require-

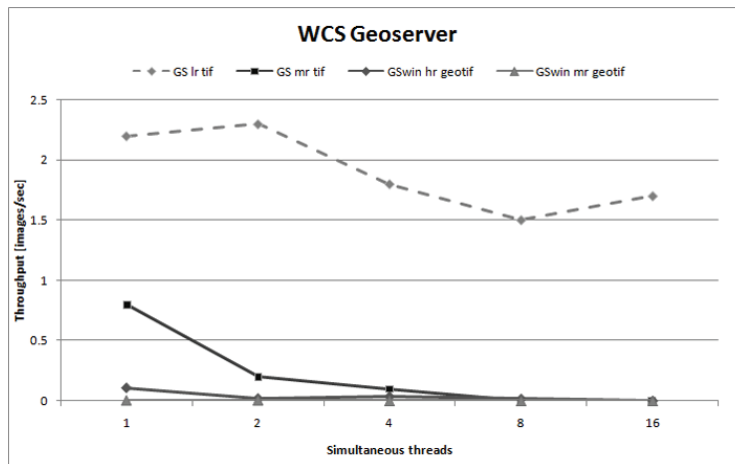


Figure 7: WCS Geoserver tests results.

ment. OGC standards and especially WFS and WCS are of high interest because they can increase interoperability of heterogeneous data sets.

Even though the two tested solutions (GeoServer and ArcGIS Server) show substantial differences in performance, our aim is not to focus on the debate of proprietary versus open source licensed software. Instead we wish to give a first insight concerning server-side WFS and WCS implementations of two of the most widely used solutions within the GI community. We also wish to contribute to: (1) improving the quality of these services independently on the platform used, (2) discussing the possibilities to test performance of these services, and finally (3) sharing our testing scripts and data used with the scientific community (e.g., FOSS4G and other testers) to improve the proposed framework of testing.

The proposed extended approach can be considered as relevant because results obtained from calibration of our architecture against WMS services show similar values as those of the FOSS4G community (Section 5.1). The stability of our results is also positive, meaning that what has been measured is differences in performance caused by the various tested parameters.

Our tests have highlighted that globally these different implementations can provide fairly good performance directly “out of the box,” regarding the vast amount of data (up to 4,000,000 rows), the retrieval extent (worldwide in some cases), and the high number of simultaneous requests (up to 1600). However they clearly show that memory is a critical factor to control and that many elements can influence the response of a given service. Therefore, simplifying the instance (e.g., turning off extraneous services or options), configuring suitable memory parameters (e.g., image rendering, type of data served), and managing the number of requests (e.g., limiting the number of concurrent requests that could prevent timely responses, workload manager) are factors that can improve the overall quality of web services [16, 45]. Various other factors may affect performance, in particular if implementations use containers such as Java virtual machine (JVM)<sup>10</sup> or Jetty<sup>11</sup>. Tuning the

<sup>10</sup><http://www.java.com/en/>

<sup>11</sup><http://jetty.codehaus.org/jetty/>

different options provided by these containers may significantly improve performance of the proposed services, as well as increasing network speed [3].

Serving large amounts of data efficiently requires optimizing data and storage [16,45]. Our results show that ESRI file geodatabase appears a suitable format, both for vector and raster data, as it provides good performance compared to flat files or ArcSDE. Knowing that FGDB could potentially become a common and cross-platform format with the recent release of an open API<sup>12</sup> reveals interesting perspectives for stakeholders. Native solutions like ArcGIS Server/ArcSDE and GeoServer/PostGIS also give globally good results and may be reliable solutions to efficiently serve data using WFS and WCS standards. When storing vector data in databases, indexing geometry and attributes significantly improves performance by accelerating data delivery. This is also particularly important because the relationship between storage size of the data set and overall performance of the download service (e.g., number of features, attributes) is not linear. Regarding raster data, building pyramids, caches, and overviews can improve the performance of the proposed service. These operations will decrease the amount of data sent when panning or zooming by tiling and down-sampling an image to a standardized size and creating overviews as separate image files in a hierarchy [15]. In the same direction, on-the-fly re-projection can be CPU intensive and storing data in the most frequently requested projection may also improve performance. Finally, symbology can also influence the performance of services. In this study we have decided to not test this parameter, because (1) this is not the most sensitive parameter (compared to memory) and (2) lots of factors need to be tested to achieve a reliable symbology (e.g., classes, forms, texture, and size).

In a more general context, WFS and WCS specifications are suitable standards to share and access data in an interoperable manner. However, even if data and storage are tuned and optimized, some bottlenecks may appear when transferring large volumes of data (e.g., geographical extent, attributes, features, or resolution). Indeed, WFS uses GML encoding [1,41,47] to serve data. Due to its verbose nature, transferring large amounts of data might be problematic and lead to high latencies and low performance. Similarly WCS is very sensitive to data resolution. Consequently, these standards are more suited to share local medium-resolution data than global high-resolution data because of the limitation cause by file formats, file size, and network bandwidth. Additionally, differences in OGC specifications have been noted in the various implementations tested (e.g., filter encoding, parameters). This may lead to interoperability problems, especially if clients do not implement the different flavors of these specifications, and may limit seamless data integration capabilities. Although it was not the purpose of this study, we noted differences between clients (e.g., ArcGIS Desktop, QGIS, uDig) when accessing a layer provided by a given service. This can be potentially caused by differences in ways of reading data in the various client software implementations. This should be further investigated as it may influence the way users perceive the quality of a service. Concerning the versions of WFS and WCS, we tested the available implementation of these standards in ArcGIS Server and GeoServer, respectively 1.1.1 and 1.1.0. However these standards are not the current versions specified by the OGC. WFS and WCS are now both available in version 2.0 [43,44]. This is a current limitation, as data providers may be dependent on the version implemented in the selected software. Even in the most recent version of ArcGIS Server (10.1) WCS 2.0 is not currently implemented. This limitation can be tackled by testing other software implementations such as RasDaMan (<http://www.rasdaman.com>) or Deegree (<http://www.deegree.com>).

---

<sup>12</sup><http://resources.arcgis.com/content/geodatabases/10.0/file-gdb-api>

org) thanks to the open approach proposed in our work (by providing access to our testing scripts and data sets). This can allow users to test not recent standard versions.

In this study, we have focused our attention on the performance criteria, which is the most critical factor to test when assessing QoS. As stated by the European Commission [19,20], the two other factors are capacity and availability. Further research should then consider measuring these two factors. Capacity is the limit of simultaneous requests that a service can handle with guaranteed performance. Measuring capacity is possible with the two tested implementations, as they manage threads with queuing functionality. After a certain number of threads, the throughput is stable and the service seems to wait until a worker is free and then handles the request. Availability (i.e., the probability that a network service is available) is more difficult to measure as it is strongly related to the architecture of a system. Having multiple instances, load balancers, and high availability routers will help to ensure reliability of services. Monitoring and measuring the status of these components requires further investigation.

Finally, our approach can be considered a starting point for many other tests to broaden and cover all aspects of QoS for data services. With such an open approach, interesting investigations can be conducted for example on: (1) on-the-fly reprojections, (2) symbology, (3) CPU, (4) new versions of software and/or implementations (e.g., Deegree, MapServer, RasDaMan), (5) different architectural style of services (e.g., representational state transfer REST), (6) WCS serving N-dimensional coverages (e.g., 3D time series, voxel data such as 4D/5D climate data), (7) clients (e.g., ArcGIS Desktop, QGIS, uDig), and (8) network components.

## 7 Conclusions

Spatial data infrastructures seek to facilitate the access and integration of geospatial data coming from various sources. To achieve this objective, systems must be interoperable. OGC specifications are key enablers providing interoperable access to data in an efficient and timely manner. Syntactical, semantic, and schematic interoperability are not enough to ensure that a given service is sufficiently responsive to fulfill users' expectations and requirements. Performance of a given service must also be measured and monitored to track latencies, bottlenecks and errors that may negatively influence its overall quality. The objectives of this study were to (1) extend the FOSS4G approach to measure the performance of different WFS and WCS implementations, (2) provide some guidance to data providers looking to improve the quality of their services, and (3) share our testing scripts and test data to the community (e.g., FOSS4G and others) to improve the proposed framework of testing.

Our work has shown that overall server-side performance of the tested implementations is globally satisfactory already "out-of-the-box" (e.g., without tuning different server-side parameters). This can be interpreted as a positive sign for data providers that may potentially be reluctant (due to too much complexity) to publish their data using OWS with these software. However, our tests have shown that to achieve reliable services, tuning memory is a critical factor, even as default software installations continue to improve usage of memory. Additionally, optimizing data (e.g., attributes indexation and reduction, projection) and storage (e.g., FGDB for flat file, PostGIS for database, Image Server for raster) are factors that can easily increase efficiency of services. Some differences have been

highlighted regarding the various implementations of WFS and WCS specifications. This can potentially limit data integration if clients do not implement the different flavors of these specifications. Finally, by nature these specifications are not well suited to transfer large volumes of data and the current specifications are more appropriate to share local medium-resolution data than global high-resolution data. This may increasingly become an issue in the near future, especially given the ever-increasing volume of high-resolution data available.

## Acknowledgments

The authors would like to acknowledge the University of Geneva and the United Nations Environment Programme (UNEP) for their support. We are grateful to Dr. Nicolas Ray from University of Geneva for reviewing an earlier draft. The views expressed in this paper are those of the authors and do not necessarily reflect the views of the institutions they belong to. We also would like to thank Ian Gunderson, Ron Witt, and four anonymous reviewers for their helpful comments on an earlier draft.

## References

- [1] AMIRIAN, P., AND ALESHEIKH, A. A. Implementation of a geospatial web service using web services technologies and native XML databases. *Middle-East Journal of Scientific Research* 3, 1 (2008), 36–48.
- [2] BARANSKI, B. Grid computing enabled Web Processing Service. In *GI-Days* (Münster, Germany, 2008). <http://www.gi-days.de/archive/2008/downloads/acceptedPapers/Papers/Baranski.pdf>.
- [3] BAUER, J. R. Assessing the robustness of web feature services necessary to satisfy the requirements of coastal management applications. Master's thesis, University of Wisconsin Sea Grant Institute, 2012.
- [4] BERMUDEZ, L., BOGDEN, P., BRIDGER, E., COOK, T., GALVARINO, C., CREA-GER, G., FORREST, D., AND GRAYBEAL, J. Web feature service (WFS) and sensor observation service (SOS) comparison to publish time series data. In *Proc. International Symposium on Collaborative Technologies and Systems* (2009), pp. 36–43. doi:10.1109/CTS.2009.5067460.
- [5] BERNARD, L., AND CRAGLIA, M. SDI—From spatial data infrastructure to service driven infrastructure. In *Research Workshop on Cross-Learning Between Spatial Data Infrastructures and Information Infrastructures* (Eschede, The Netherlands, 2005).
- [6] BERNARD, L., KANELLOPOULOS, I., ANNONI, A., AND SMITS, P. The European geoportal—One step towards the establishment of a European spatial data infrastructure. *Computers, Environment and Urban Systems* 29, 1 (2005), 15–31. doi:10.1016/S0198-9715(04)00049-3.
- [7] BOES, U., AND PAVLOVA, R. Is there a future for spatial data infrastructures? In *GI-Days* (Münster, Germany, 2008), pp. 305–314. <http://www.gi-tage.de/archive/2008/downloads/acceptedPapers/Papers/Boes,Pavlova.pdf>.

- [8] BOOZ, ALLEN, HAMILTON. Geospatial interoperability return on investment. Report for NASA Geospatial Interoperability Office, 2005. [http://egy.org/files/ROI\\_Study.pdf](http://egy.org/files/ROI_Study.pdf).
- [9] BRAUNER, J., FOERSTER, T., SCHAEFFER, B., AND BARANSKI, B. Towards a research agenda for geoprocessing services. In *Proc. 12th AGILE International Conference on Geographic Information Science* (2009), p. 12. [http://www.agile-online.org/Conference\\_Paper/CDs/agile\\_2009/AGILE\\_CD/pdfs/124.pdf](http://www.agile-online.org/Conference_Paper/CDs/agile_2009/AGILE_CD/pdfs/124.pdf).
- [10] BRUNIECKI, K., KULAWIAK, M., AND MOSZYNSKI, M. Concept of web service for real-time satellite imagery dissemination. In *Proc. 2nd International Conference on Information Technology (ICIT)* (2010), pp. 149–152.
- [11] BULATOVIC, V., NINKOV, T., AND SUSIC, Z. Open Geospatial Consortium web service in complex distribution systems. *Geodetski list* 64, 1 (2010), 13–29. [http://hrcak.srce.hr/index.php?show=clanak&id\\_clanak\\_jezik=80885](http://hrcak.srce.hr/index.php?show=clanak&id_clanak_jezik=80885).
- [12] CHRISTENSEN, F., BERNARD, L., KANELLOPOULOS, I., ISO, N., PEEDELL, S., SCHADE, S., AND THORNE, C. Building service oriented applications on top of a spatial data infrastructure: A forest fire assessment example. In *Proc. 9th International AGILE Conference* (2006). [http://plone.itc.nl/agile\\_old/Conference/2006-Visegrad/papers/a119.pdf](http://plone.itc.nl/agile_old/Conference/2006-Visegrad/papers/a119.pdf).
- [13] CRAGLIA, M., AND CAMPAGNA, M. Advanced regional spatial data infrastructures in Europe. Tech. Rep. EUR 23716 EN, European Commission, Joint Research Centre, 2009.
- [14] DA SILVA, J., FIDALGO, R. D. N., AND SOUTO MAIOR DE BARROS, R. Towards a web service for geographic and multidimensional processing. In *Brazilian Symposium on GeoInformatics* (2004), pp. 1–18. <http://www.dpi.inpe.br/geoinfo/geoinfo2004/papers/5835.pdf>.
- [15] ESRI. Services de carte hautes performances: Cas d’utilisation avec les données CORINE land cover. ESRI, 2009. Série ArcGIS Server en pratique.
- [16] ESRI. Performance and throughput tips for ArcGIS Server 9.3.1 cached map services and the Apache HTTP server. ESRI, 2010.
- [17] EUROPEAN COMMISSION. Directive 2007/2/EC of the European Parliament establishing an Infrastructure for Spatial Information in the European Community (INSPIRE), 2007. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:en:PDF>.
- [18] EUROPEAN COMMISSION. INSPIRE network services performance guidelines, 2007. [http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/Network\\_Services\\_Performance\\_Guidelines\\_v1.0.pdf](http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/Network_Services_Performance_Guidelines_v1.0.pdf).
- [19] EUROPEAN COMMISSION. Commission Regulation 976/2009 implementing Directive 2007/2/EC of the European Parliament and of the Council as regards the network services, 2009. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32009R0976:EN:NOT>.



- [20] EUROPEAN COMMISSION. Commission Regulation 1088/2010 amending Regulation (EC) No 976/2009 as regards download services and transformation services, 2010. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:02009R0976-20101228:EN:NOT>.
- [21] EUROPEAN COMMISSION, JOINT RESEARCH CENTRE. Assessing the impacts of spatial data infrastructures. Tech. Rep. LB-NA-22294-EN-C, European Commission, 2006.
- [22] FU, G., JONES, C. B., AND ABDELMOTY, A. I. Building a geographical ontology for intelligent spatial search on the web. In *Proc. IASTED International Conference on Databases and Applications* (2005), pp. 167–172.
- [23] GIULIANI, G., AND PEDUZZI, P. The PREVIEW global risk data platform: A geoportal to serve and share global data on risk to natural hazards. *Natural Hazards and Earth System Sciences* 11, 1 (2011), 53–66. doi:10.5194/nhess-11-53-2011.
- [24] GRANELL, C., DIAZ, L., AND GOULD, M. *Distributed Geospatial Processing Services*. Information Science Reference, 2009, pp. 1186–1193. doi:10.4018/978-1-60566-026-4.ch188.
- [25] GROUP ON EARTH OBSERVATIONS (GEO). Global earth observation system of systems 10-year implementation plan reference document, 2005. <http://preventionweb.net/go/8633>.
- [26] HORAK, J., ARDIELLI, J., AND HORAKOVA, B. Testing of web map services. In *GSDI 11* (2009), pp. 1–25. doi:10.1007/978-3-642-19953-0.26.
- [27] KHOUMERI, E. H., AND BENSLIMANE, D. Web GIS for multiple representation data. In *Proc. 2nd International Conference on Digital Information Management (ICDIM)* (2007), vol. 1, pp. 188–192. doi:10.1109/ICDIM.2007.4444221.
- [28] LACASTA, J., NOGUERAS-ISO, J., BEJAR, R., MURO-MEDRANO, P. R., AND ZARAZAGA-SORIA, F. J. A web ontology service to facilitate interoperability within a spatial data infrastructure: Applicability to discovery. *Data & Knowledge Engineering* 63, 3 (2007), 947–971. doi:10.1016/j.datak.2007.06.002.
- [29] LANCE, K., GEORGIADOU, Y., AND BREGT, A. K. Understanding how and why practitioners evaluate SDI performance. *International Journal of Spatial Data Infrastructures Research* 1 (2006), 65–104.
- [30] LEHNER, B., AND DÖLL, P. Development and validation of a global database of lakes, reservoirs, and wetlands. *Journal of Hydrology* 296, 1–4 (2004), 1–22. doi:10.1016/j.jhydrol.2004.03.028.
- [31] LUTZ, M., SPRADO, J., KLIEN, E., SCHUBERT, C., AND CHRIST, I. Overcoming semantic heterogeneity in spatial data infrastructures. *Computers & Geosciences* 35, 4 (2009), 739–752. doi:10.1016/J.Cageo.2007.09.017.
- [32] MA, X., PAN, Q. H., AND LI, M. L. Integration and share of spatial data based on web service. In *Proc. Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)* (2005), pp. 328–332.

- [33] MASSER, I. The future of spatial data infrastructures. In *ISPRS Workshop on Service and Application of Spatial Data Infrastructure* (2005).
- [34] MENASCE, D. A. Load testing of web sites. *IEEE Internet Computing* 6, 4 (2002), 70–74. doi:10.1109/MIC.2002.1020328.
- [35] MENDES, E., AND MOSLEY, N. *Web Engineering*. Springer, Berlin, 2006. doi:10.1007/3-540-28218-1.
- [36] O’DEA, E., HADDAD, T., DUNNE, D., AND WALSH, K. Coastal web atlas features. In *Coastal Informatics: Web Atlas Design and Implementation* (2011), D. Wright, E. Dwyer, and V. Cummins, Eds., IGI Global, pp. 12–32.
- [37] OPEN GEOSPATIAL CONSORTIUM. Web Feature Service Implementation Specification. Tech. Rep. OGC 04-094, Open Geospatial Consortium, 2005. [http://portal.opengeospatial.org/files/?artifact\\_id=8339](http://portal.opengeospatial.org/files/?artifact_id=8339).
- [38] OPEN GEOSPATIAL CONSORTIUM. OpenGIS Web Map Server Implementation Specification. Tech. Rep. OGC 06-042, Open Geospatial Consortium, 2006. [http://portal.opengeospatial.org/files/?artifact\\_id=14416](http://portal.opengeospatial.org/files/?artifact_id=14416).
- [39] OPEN GEOSPATIAL CONSORTIUM. Web Coverage Service (WCS) Implementation Specification. Tech. Rep. OGC 06-083r8, Open Geospatial Consortium, 2006. [http://portal.opengeospatial.org/files/?artifact\\_id=18153](http://portal.opengeospatial.org/files/?artifact_id=18153).
- [40] OPEN GEOSPATIAL CONSORTIUM. OpenGIS Catalogue Services Specification. Tech. Rep. OGC 07-045, Open Geospatial Consortium, 2007. [http://portal.opengeospatial.org/files/?artifact\\_id=21460](http://portal.opengeospatial.org/files/?artifact_id=21460).
- [41] OPEN GEOSPATIAL CONSORTIUM. OpenGIS Geography Markup Language (GML) Encoding Standard. Tech. Rep. OGC 07-036, Open Geospatial Consortium, 2007. [http://portal.opengeospatial.org/files/?artifact\\_id=20509](http://portal.opengeospatial.org/files/?artifact_id=20509).
- [42] OPEN GEOSPATIAL CONSORTIUM. OpenGIS Web Processing Service. Tech. Rep. OGC 05-007r7, Open Geospatial Consortium, 2007. [http://portal.opengeospatial.org/files/?artifact\\_id=24151](http://portal.opengeospatial.org/files/?artifact_id=24151).
- [43] OPEN GEOSPATIAL CONSORTIUM. OGC WCS 2.0 Interface Standard — Core. Tech. Rep. OGC 09-110r3, Open Geospatial Consortium, 2010. [http://portal.opengeospatial.org/files/?artifact\\_id=41437](http://portal.opengeospatial.org/files/?artifact_id=41437).
- [44] OPEN GEOSPATIAL CONSORTIUM. OpenGIS Web Feature Service 2.0 Interface Standard. Tech. Rep. OGC 09-025r1 and ISO/DIS 19142, Open Geospatial Consortium, 2010. [http://portal.opengeospatial.org/files/?artifact\\_id=39967](http://portal.opengeospatial.org/files/?artifact_id=39967).
- [45] OPENGEO. GeoServer in production. Tech. rep., OpenGeo, 2013. <http://opengeo.org/publications/geoserver-production/>.
- [46] PAUL, M., AND GHOSH, S. K. An approach for service oriented discovery and retrieval of spatial data. In *Proc. International Workshop on Service-Oriented Software Engineering (SOSE)* (2006), pp. 88–94. doi:10.1145/1138486.1138505.

- [47] PENG, Z.-R., AND ZHANG, C. The roles of geography markup language (GML), scalable vector graphics (SVG), and web feature service (WFS) specifications in the development of internet geographic information systems (GIS). *Journal of Geographical Systems* 6, 2 (2004), 95–116. doi:10.1007/s10109-004-0129-0.
- [48] RAJABIFARD, A., AND WILLIAMSON, I. P. Spatial data infrastructures: Concept, SDI hierarchy and future directions. In *Proc. Geomatics'80* (2001).
- [49] RYTTERSGAARD, J. Spatial data infrastructure: Developing trends and challenges. In *International Conference on Spatial Information for Sustainable Development* (2001).
- [50] SAHIN, K., AND GUMUSAY, M. Service oriented architecture (SOA) based web services for geographic information systems. In *XXIst ISPRS Congress* (2008), pp. 625–630.
- [51] SAYAR, A., PIERCE, M., AND FOX, G. C. OGC compatible geographical information systems web services. Tech. Rep. TR610, Indiana University, 2005. <http://www.cs.indiana.edu/cgi-bin/techreports/TRNNN.cgi?trnum=TR610>.
- [52] SCHAEFFER, B. Towards a transactional web processing service (WPS-T). In *GI-Days* (Münster, Germany, 2008), p. 27. <http://www.gi-tage.de/archive/2008/downloads/acceptedPapers/Papers/Schaeffer.pdf>.
- [53] SIMONIS, I., AND SLIWINSKI, A. Quality of service in a global SDI. In *Proc. 8th International Conference for Global Spatial Data Infrastructure* (2005).
- [54] STÖCKLI, R., VERMOTE, E., SALEOUS, N., SIMMON, R., AND HERRING, D. The Blue Marble Next Generation—A true color earth dataset including seasonal dynamics from MODIS. NASA, 2005.
- [55] TU, S., FLANAGIN, M., WU, Y., ABDELGUERFI, M., NORMAND, E., AND MAHADEVAN, V. Design strategies to improve performance of GIS web services. In *Proc. International Conference on Information Technology: Coding and Computing (ITCC)* (2004), vol. 2, pp. 444–448.
- [56] VANDENBROUCKE, D. Spatial data infrastructures in Europe: State of play Spring 2010. K.U.Leuven, 2010.
- [57] VANMEULEBROUK, B., BULENS, J., KRAUSE, A., AND DE GROOT, H. OGC standards in daily practice: Gaps and difficulties found in their use. In *GSDI 11* (2009).
- [58] YAXING, W., SANTHANA-VANNAN, S. K., AND COOK, R. B. Discover, visualize, and deliver geospatial data through OGC standards-based WebGIS system. In *Proc. 17th International Conference on Geoinformatics* (2009), pp. 1–6.
- [59] ZHANG, C. The roles of web feature and web map services in real-time geospatial data sharing for time-critical applications. *Cartography and Geographic Information Science* 32 (2005), 269–283. doi:10.1559/152304005775194728.
- [60] ZHAO, P. S., DI, L. P., YU, G. N., YUE, P., WEI, Y. X., AND YANG, W. L. Semantic web-based geospatial knowledge transformation. *Computers & Geosciences* 35, 4 (2009), 798–808. doi:10.1016/J.Cageo.2008.03.013.